Title	Contributions to the measurement of depth in consumer imaging
Author(s)	Javidnia, Hossein
Publication Date	2018-06-12
Publisher	NUI Galway
Item record	http://hdl.handle.net/10379/7398

Contributions to the Measurement of Depth in Consumer Imaging

Hossein Javidnia

College of Engineering and Informatics

National University of Ireland Galway

This dissertation is submitted for the degree of

Doctor of Philosophy

Supervisor: Prof. Peter Corcoran April 2018

“The key to growth is the introduction of higher dimensions of consciousness into our awareness.”

~ *Lao Tzu*

Table of Contents

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 80,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Hossein Javidnia
April 2018

Acknowledgments

Foremost, I would like to express my sincere gratitude to my Ph.D. advisor Prof. Peter Corcoran for the continuous support of my Ph.D. study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D. study. Thanks, Peter!

I am forever indebted to Prof. Christopher Dainty for his constant feedback on my work. I truly appreciate him for being a friend, for providing me with a lot of research insights. I would also like to thank Dr. Petronel Bigioi and Dr. Alexandru Drimbarean for taking me as a part of the FotoNation family and involving me in some exciting projects.

I am grateful to all of those with whom I have had the pleasure to work during this and other related projects.

I would be remiss if I did not thank my friends and colleagues at the C3Imaging group; Shabab Bazrafkan for being a wonderful friend, for all his support during my Ph.D. and the great time we spent together, especially in Las Vegas.

A special thanks to Dr. Claudia Costache for her constant help and support throughout this journey.

Thanks to Adrian Ungureanu for being a supportive friend, especially during the first months of my trip to Ireland.

Thanks to Tudor Nedelcu and Viktor Varkarakis for all the wonderful times and laughs.

Thanks to Asma Khatoon, Anuradha Kar and Aoife McDonagh for all times we had lunch at FotoNation.

Thanks to my unforgettable friends, Joe Lemley and Snail Lemley for their constant help with reading my thesis and papers.

Thanks to Ashkan Parsi and Saba Madani. All the memorable afternoons that we spent together in cafés. Sharing the moments, walks and discussions, made me feel home.

My heartfelt gratitude to Dr. Shejin Thavalengal; a friend, a colleague who was supportive since the day that I started my Ph.D. The only person that I refered to in difficult moments and asked for advice. Thanks for all the moveis and hot chocolates!

I have been so lucky to meet a wonderful friend, Lina. A person who was beside me all the time throughout this journey. Thank you for the never ending support; for all the dances, dinners, lunches and walks. Thank you for everything!

And above everything, throughout the entire journey, I have benefitted immensely from the support of my family and especially my parents for their love, inspiration and always being there when I needed. I never thank you enough for being there for me without a doubt.

I would like to acknowledge Science Foundation Ireland for the generous funding for my Ph.D.

Abstract

This thesis aims to examine and investigate methods that could potentially utilize images captured by consumer cameras such as smartphones to estimate depth and generate a 3D structure.

After more than a century of research in depth sensing and 3D reconstruction, there are still open and unsolved challenges, and ultimately a practical solution for each problem will have to rely on combining a range of techniques as there is no single best solution which can satisfy all the requirements of a depth sensing application.

Based on this, a number of methods and frameworks are presented to take advantage of the existing consumer cameras in depth sensing applications. A method is presented to post-process the depth maps with respect to the geometrical structure of the scene. Later, this method is adopted to evaluate the effectiveness of the deep learning approaches in monocular depth estimation. To utilize the current mono cameras available on smartphones, a framework is presented to use the pre-capturing small motions for 3D reconstruction and depth sensing applications. Similarly, a mono camera can be used to capture a sequence of images in different focal planes known as focal stack. A framework is designed to estimate dense depth map from focal stack in a reasonably fast processing time for high resolution images. Lastly, to investigate the potentials of the current consumer multi-camera arrays, a framework is proposed to estimate dense depth map from these cameras.

The advanced capabilities of today’s smartphones brings hope that we can arrive at a consensual depth sensing imaging system in the next decade or so, and hopefully some of the contributions of this research will contribute in part to this solution.

List of Figures

List of Tables

Chapter 1.

Depth in Images – Its History and Relevance in Contemporary Imaging Technology

In this mobile-first world, having access to high accuracy depth information would introduce great applications on smartphones. However, integrating active depth sensing sensors into mobile phones is not practical due to the size and power limitation. Despite the presence of DLP projectors for smartphones [1] and ToF depth cameras, some practical issues such as energy efficiency and measurement range have to be taken into consideration. The primary objective of the work described in this thesis is to examine methods of estimating depth and 3D reconstruction for lightweight computational devices such as smartphones. The main contributions of this work fall into the following categories:

- Stereo Matching and Depth Estimation
- Depth from Monocular Camera using DNN
- Category of Small Motions:
  1. Structure from Small Motion (SfSM)
  2. Depth from Focal Stack
  3. Depth from Multi-Camera Array

1.1 Human's Visual System and Depth Perception

Human's vision is a unique system with a full range of benefits among all the species. We are equipped with wide-field peripheral vision with an angle of ~175°. Human's vision system contains two frontal eyes which are located side by side and create an area of clear vision with an angle of ~85-95°.

Each eye captures its own view of a scene from a slightly different angle. The brain unifies the images into a single image by matching up the similarities and filling in the small differences. The fused final image is a 3-Dimensional (3D) stereo picture. The word "stereo" comes from the Greek word στέρεος (stéreos) which means firm or solid. With stereo vision, an object is represented as solid in three spatial dimensions including width, height, and depth.

It is believed that depth perception in humans' visual system comes from a cue-based approach, meaning that we identify informative elements that relate to the depth of a scene. Generally, the depth cues are divided into three categories:

1- Monocular - Cues identifiable with one eye.
2- Binocular - Cues identifiable with two eyes.
3- Oculomotor - These are cues based on the ability to sense the position of the objects and the tension in the eye muscles.

1.1.1 Monocular Cues

Monocular cues are the elements of the scene which provide the information for one eye to approximate the depth. These cues are divided into two categories including: Pictorial cues and Movement-based cues.

1.1.1.1 Pictorial Cues

Pictorial cues are the depth information that can be presented in 2-Dimensional (2D) space as a photo, including occlusion, relative height, cast shadows, relative size, familiar size, atmospheric perspective, linear perspective and texture gradient.

1.1.1.2 Movement-Based Cues

As we move in our environment, the objects at different distances seem to move at different rates relative to us. When we are moving, the objects at a closer distance seem to move faster in the opposite direction than the objects located at a further. This dynamic depth cue is known as motion parallax.

Another concept is "deletion and accretion", which is the degree to which a closer object reveals or covers a further object as we pass them. As we move, some parts of the objects get revealed or occluded. The rate of deletion and accretion provides information about the depth of the objects.

1.1.2 Binocular Cues

As mentioned previously, each eye captures its own view of a scene from a slightly different angle. The eyes are separated by a distance of about ~6.3 cm. The brain utilises the differences between the left and right view to perceive depth information. This distance is greater for the objects that are closer to the observer. This process is known as stereopsis, where the information from both eyes is used to derive depth information.

1.1.3 Oculomotor Cues

When we are trying to focus on the objects at different distances, the information sent to the brain from the tension in the eye muscles can provide depth information. These cues are categorised as accommodation and convergence.

1.1.3.1 Accommodation

Accommodation refers to the contraction of the ciliary muscle in the eye, which is what holds the lens allowing the eye to focus. When we are trying to focus on a close object, a contraction happens.

1.1.3.2 Convergence

Convergence helps the eyes focus on close objects. When an object moves away from the observer, the eyes diverge to maintain focus. Muscle movements provide information about the depth of the object.

Good depth perception is important for humans' daily interaction with their environment, as it affects coordination and hand to eye skills, aiding in ease of survival. Without this sense, recognizing a location in space becomes very difficult. Depth perception affects many daily activities, such as jumping, catching, throwing, navigation, collision avoidance, size judgment, and recognition, walking, running etc. Depth perception also helps us to recognise danger from a far distance and enables us to discriminate and identify objects.

1.2 Historical Origin of Stereoscope and 3D Revolution Era

In 1832 at the very early stage of photography, Charles Wheatstone invented a device composed of two mirrors at a 45-degree angle to the viewer's eyes to create the illusion of depth [2], [3]. This device is known as the first invention of the Stereoscope in the world of photography. Wheatstone's stereoscope was designed to show two offset images separately to the left and right eye. The combination of these 2D views in the brain gave the perception of 3D depth data. Later in 1849, David Brewster replaced the mirrors in the stereoscope with lenses and invented the lenticular stereoscope [4].

In 1851, 3D received a historical boost and caught the public's attention. David Brewster's stereoscope was displayed at the Crystal Palace World Fair Exhibition in London. Stereoviews proved to be crowd pleasers at the 1851 Great Exhibition [5]. Brewster presented one to Queen Victoria, who was quite fascinated by this new innovation [6]. At the same exhibition, a picture of Queen Victoria was taken by Louis Jules Duboscq using Brewster's stereoscope which became very well known throughout the world.

By that time, the stereo viewers were heavy wooden boxes, clumsy and expensive to build. In 1859, Oliver Wendell Holmes came up with a revolutionary design for the stereoscopes. He invented a light version of the device which was known as the Holmes stereoscope [7], [8]. His device allowed the viewer to hold the apparatus in one hand while using the other to position the stereograph in one of several grooves carved into a wooden arm extending from the eye tubes [9]. Holmes wrote: "It appeared to me that the box stereoscopes were cumbrous and awkward affairs. I had one of Smith and Beck’s, and one or more of other patterns, but I did not like them; and so one day I cut out a piece of wood in some shape as this, the lines representing slots in which the stereograph was to be placed, stuck an awl in for a handle, and there was my stereoscope." [10].

In 1861, Coleman Sellers patented a stereoscopic moving picture peep show machine which he called Kinematoscope [11], [12]. His kinematoscope was made of a spinning blade inside a cabinet where a series of stereographs were mounted on the blade. Sellers discovered the principle of intermittent motion for moving pictures and his kinematoscope is considered one of the most important precursors to motion pictures and cinema.

From the 1840s through the 1920s, stereoscopes turned into one of the main gadgets for home entertainment, education, and virtual travel. In 1893, William Friese Greene patented a 3D viewing scheme using two side-by-side screens. The images were viewed through a cumbersome stereoscope [11].

In 1915, Edwin S Porter presented his new invention, the first red-green anaglyphic stereoscopic movie projection system. He presented a number of short movies in Astor Theatre, New York City [13].

In 1922, the first public 3D movie, "The Power of Love," was screened, followed by the first 3D colour movie in 1935 [14].

This technology lost its popularity between the late 1920s and early 1930s due to the Great Depression. But the 1950s saw a comeback for the 3D technology and it is known as the golden era of 3D movies. During these times, TVs had become very popular. By 1953, a number of 3D movies were released, such as "Bwana Devil," "House of Wax" and "Man in the Dark". But not all movie theatres had the equipment to screen 3D movies [14]. Many people reported that watching out of sync or unfocused 3D movies caused them nausea or headaches and they preferred watching them in the normal 2D and flat mode. The movies were also expensive to rent for theatres.

The 1960s saw another comeback of the 3D technology when Arch Oboler invented a new technology known as Space-Vision 3D. Oboler's invention did not require any synchronization and it was cheaper and easier to maintain. His new technology utilised a single print solution and removed the need to use two cameras to display 3D movies. The first movie which was screened using this technology was "The Bubble" [15].

In 1970, a new technology called Stereovision was introduced. A special anamorphic lens was used in Stereovision to stretch the picture using a series of polaroid filters. The first movie released using this technology was "The Stewardesses." The movie cost only $100,000 and it earned about $27 million in North America after showing at just 800 theatres. This movie became the most profitable 3D film of all time [15].

By 1980, more movies were screened using the 3D technology, such as "Friday the 13th Part III" and "Jaws 3D."

Later in the 1990s, the first 3D movie, "Echoes of the Sun," was produced in Canada using polarized glasses. IMAX projectors moved the 3D cinema to another level. Some of the most famous movies released in IMAX 3D are "Into the Deep" and "Wings of Courage."

1.2.1 The 21st Century

In early 2000, big 3D movies such as "Spy Kids 3D: Game over," "Aliens of the Deep," "The Adventures of Sharkboy and Lavagirl" and "The Polar Express" were screened using High-Definition (HD) video cameras.

3D TVs and vision became very popular in late 2009 and early 2010 after the big release of the movie "Avatar" [15]. These days many educational shows, sports events, and documentaries are displayed in 3D, and still, the long and interesting story of 3D technology continues.

The success of 3D technology in the movie industry has introduced the so-called "3D revolution" and has facilitated the rapid development of 3D equipment. Stereoscopic and autostereoscopic technology have found their way to home entertainment equipment, vehicles, drones, TVs, mobile devices and PC screens. Nowadays, stereo cameras are available for consumers at a very reasonable price. Capturing and displaying stereoscopic media has been revolutionised by deploying stereoscopic displays and cameras, and we are at the early stage of the new stereoscopic multimedia era. Although stereoscopic hardware has developed rapidly in both acquisition and display technology, there has not been much progress on stereoscopic processing software development, especially in consumer devices.

In the world of consumer electronics, Microsoft Kinect was the world's first consumer depth camera which introduced a new era in depth-sensing technology. Taking advantage of the structured light, Kinect uses pattern projection and relies on parallel computing to capture real-time frames at 30 frames per second (fps). For applications such as robotics and emerging Virtual and Augmented Reality (VR/AR), high-performance depth cameras are required to capture real-time information. Microsoft HoloLens [16] is the recent examples of these cameras. "The displays on HoloLens refresh 240 times a second, showing four separate colour fields for each newly rendered image, resulting in a user experience of 60 FPS (frames per second). To provide the best experience possible, application developers must maintain 60 FPS." [17]

1.3 Digital Camera Technology and Its Challenges

The use of consumer lightweight cameras, specifically smartphones, is growing continuously and the level of expectation around what these cameras can do is increasing yearly. With the growing use of these cameras, deriving the 3D information has become an important challenge in consumer imaging.

The capabilities of imaging sensors deployed in smartphones are limited, and they have a poor performance under low light conditions (lux < ~100). Images are captured blurry and suffer from notable aberration. The small size of the Charge-Coupled Device (CCD) sensors increases the amount of noise in the image and decreases the performance of the computer vision algorithms. A larger sensor could solve this problem, but the size of the components is restricted in smartphones, and consumers are not eager to mount an extra lens on their smartphone camera.

Many specs of the advanced Digital Single-Lens Reflex (DSLR) cameras such as optical zoom, shutter speed, aperture, white balance and ISO settings, are not accessible or are only accessible in limited form in smartphones.

In general, because of all these limitations, it has been very challenging to capture DSLR camera quality images using smartphones. However, most of these challenges have been addressed by deploying advanced mobile processors such as the Image Signal Processor (ISP), which introduced many complex post-processing algorithms to compensate for the shortcomings of the limited optical components of smartphones. The current ISP technology provides many processing capabilities through advanced image processing blocks such as demosaicing, noise filtering, colour correction, tone mapping, auto focus, auto exposure, white balance, lens shading correction etc.

The other limitation in smartphones is the battery power. High resolution and high frame rate cameras require a lot of energy to run. Intensive computer vision algorithms, including Augmented Reality (AR) applications, tend to drain the smartphones' batteries.

All these limitations make 3D information generation a very challenging task in the smartphone industry. Mobile device manufacturers are exploiting depth information to see the world in 3D. Considering all the physical limitations of the cameras, there are still potential features, such as camera motion [18], [19], focal sweeps [20]–[22] and microlenses [23], which could potentially be used to generate 3D information.

1.4 The importance of Depth Information

Most of the conversion methods that generate 3D information from a set of 2D pixels are based on the depth values computed for each 2D point. In a depth map, each pixel is defined not by colour, but by the distance between an object and the camera.

Having the depth and 3D information of a scene enables consumers to infer and understand its semantics and geometric structure as well as enabling many applications in computer vision such as autonomous navigation [24], 3D geographic information systems [25], object detection and tracking [26], medical imaging [27], advanced graphical applications [28], 3D holography [29], 3D television [30], multi-view stereoscopic video compression [31], and disparity-based segmentation [32].

For example, having depth information along with an image in a smartphone allows users to simulate a Bokeh effect [33] and to change the focus points or lighting after the image has been taken. Apps could be designed to entirely remove or add an object to a scene with different lighting, shading and other special effects.

Another example could be Virtual and Augmented Reality (VR/AR) where the depth of a real environment can be captured and reconstructed to enable virtual tours or visits. Faces can be scanned as 3D avatars for teleconferencing and gaming purposes. Virtual maps can be designed for house redecoration [34]. Online shopping can be revolutionised using AR/VR technology [35], [36]. Driving habits and behaviors can be improved using AR/VR glasses with a navigation system.

In another context, depth sensing technologies provide a great opportunity to significantly increase the navigation and interaction capabilities of vehicles. The importance of depth information has been demonstrated in the autonomous car driving. It is essential for an autonomous vehicle to accurately and reliably perceive the geometrical features of the environment, which can potentially lead to obstacle detection.

1.5 Overview of Technical Challenges

After more than a century of research in depth sensing and 3D reconstruction, there are still open and unsolved challenges, and ultimately a practical solution for each problem will have to rely on combining a range of techniques. There are many problems such as depth estimation in an uncalibrated environment, handling occlusion and missing data, real-time performance, estimating depth from single RGB images, fast linear and non-linear solvers, and 3D map compression, which still require an extensive amount of research to be solved.

Acquiring accurate depth information using the minimum computational resources is one of the main open challenges in lightweight imaging systems such as smartphones [37], [38]. Smartphone cameras have limited capabilities and optical properties which make it difficult to understand the geometrical features of the scene. Most of the phones are equipped with only one camera. This problem remains challenging because there are no reliable cues for inferring depth from a single image. For example, temporal information and stereo correspondences are missing from such images. There are low-cost algorithms that can be employed on lightweight devices, but the accuracy of the depth and 3D information estimated by those algorithms is considerably low [39], [40]. Most importantly, they do not preserve the geometrical structure of the scene, which makes it very difficult to distinguish a specific object. Another challenge is discovering how we can utilise features such as the accidental motion of the camera or the optical properties of current cameras, like focal sweeps, to estimate depth information.

In the application of autonomous navigation, it is essential to have real-time accurate depth and 3D data. Autonomous navigation devices can be divided into two categories: battery operated drones, and vehicles. The use of a camera, or sets of cameras, is limited in this application as they can be interfered with different lighting conditions, reflective surfaces, weather conditions etc.

Laser scanners can be used to generate 3D data for autonomous navigation. However, the scanners are expensive and they require a significant power source to operate, which limits their performance on battery operated drones. The more important challenge is how to densify the sparse data generated by a laser scanner while preserving the structure of the scene [41]–[43].

Another important challenge is to consider factors that affect the quality of a depth map while designing an algorithm including occlusion prediction, regularization term, and consistency term.

One of the factors for visually inspecting the quality of a 3D reconstruction is through the incorporation of the colour of every point on the object. The results provided by recent reconstruction methods are not very accurate. The current methods employ a volumetric blending approach that integrates colour samples over a voxel grid [44]–[49] to colour the geometric models produced using consumer depth cameras. The colour maps generated using these methods respect an object's general appearance; however, they suffer from a number of artifacts such as blurring and ghosting, which are visible at close range.

Another important factor that can decrease the quality of depth estimation and 3D reconstruction is the presence of noise, which can lead to a faulty geometrical reconstruction and inaccurate camera pose estimation. For consumer devices that are equipped with a depth and a colour camera, asynchronous shutters commonly cause the misalignment of projected images. In RGB-D consumer cameras, the images mostly suffer from the artifacts introduced by optical distortion [50].

The advanced capabilities of today's smartphones bring hope that we can arrive at a consensual depth sensing imaging system in the next decade or so, and hopefully, some of the contributions of this research will contribute in part to this solution.

The research presented in this dissertation aims to examine and investigate methods that could potentially utilise images captured by consumer cameras to estimate depth and generate a 3D structure.

1.6 Contribution Summary of the Thesis

1.6.1 Generic Post-Processing Method to Refine Depth Map

For the purpose of exploring the challenges of estimating depth from stereo image sets and taking advantage of the growing computational capabilities of embedded imaging systems, a generic post-processing method is proposed which is capable of preserving the structure of the scene in the depth map. The framework starts with a state of the art stereo matching algorithm known as "Adaptive Random Walk with Restart" [51]. To refine the depth map generated by this method, we introduced a form of median solver/filter based on the concept of the mutual structure, which refers to the structural information in both images. This filter is further enhanced by a joint filter. Next, a transformation in image domain is introduced to remove the artifacts, which cause distortion in the image. Note that this algorithm is generic and can be employed to increase the accuracy of the depth maps in any application.

1.6.2 Models to Estimate Depth from Monocular Camera by Employing a Semi-Parallel Deep Neural Network

To overcome the extensive computational power of stereo matching techniques and utilise the monocular camera, the potential of an advanced deep learning technique is investigated which may soon be embedded in the hardware processing chipsets of the image processing pipeline. A Convolutional Neural Network (CNN) model is applied to the problem of determining the depth from a single camera image (monocular depth). Eight different networks are designed to perform depth estimation. After designing a set of networks, these models are combined into a single network topology using graph optimisation techniques. In this study, four Semi Parallel Deep Neural Network (SPDNN) models are trained and evaluated at two stages on a common dataset. To evaluate the performance of the post-processing method presented in Section 4.1, we trained two of the networks using the post-processed depth maps.

1.6.3 A Method to Compute Depth and 3D Structure from Small Motion

This technique utilises sudden motion of the camera to estimate depth and 3D structure. Since consumer cameras usually employ pre-capturing initialization, we propose a novel method by utilising the pre-captured small motions to estimate a depth map and generate a semi-dense 3D structure of a scene. The method uses a sequence of images captured on a narrow baseline in smartphones. The basic idea is to make use of the small movements of the camera such as natural hand-shake to estimate a depth map. In such movements, the baseline between sequences of frames is considered small if it is less than ~8 mm. The proposed framework starts by tracking common features throughout an uncalibrated image sequence and generating a 3D structure which is later optimised using a modified bundle adjustment.

1.6.4 Depth Measurement from a Focal Stack Framework

This technique can take advantage of the faster frame rates and improved focus ranges of some of the most recent smartphones. It utilises the optical properties of consumer cameras to estimate depth from a stack of images captured in different focal planes. Similar to small motions in smartphone cameras, one can record a short sequence of frames with varying focal points. In smartphones, this feature is known as a focal stack which is generated by automatic focal plane sweeps to find the camera's best auto-focus setting while taking photos. A method is proposed to estimate depth from a focal stack for post-capture refocusing purposes. The method initiates by aligning the images in the stack using Epipolar homography alignment. Later, a Modified Laplacian is used to calculate the focus function. The initial depth map is calculated by modeling the focus function using a 3-point Gaussian distribution. The problem of the noisy depth map is reformulated to a convex minimisation problem to be solved by Preconditioned Alternating Direction Method of Multipliers (PADMM) which results in recovering uncertain depth values.

1.6.5 Speed Optimised Depth Measurement from a Multi-Camera Array Framework

The last contribution of this thesis focuses on the new trend of smartphone cameras known as a multi-camera array. These cameras provide multiple views with a small baseline from a scene. One of the known technologies in this category, which has received much attention in the past years, is PiCam [52] where 16 microlenses are placed in an array to generate a camera module which is 3 mm thick. By introducing this camera, Pelican imaging brought light field technology in a compact form to mobile devices. The images captured by this camera can be used for post-capture refocusing, 3D modeling and printing applications, AR and segmentation.

Several methods have been recently proposed to estimate depth and 3D structure from these cameras. However, the computational requirements of these methods make them impractical for many applications. A framework is proposed based on Epipolar Plane Image (EPI) to estimate a high-quality dense depth map from the multi-camera array. The initial depth map is estimated by utilising the local EPI and it is later refined using Total Variation (TV) minimisation based on the Fenchel-Rockafellar duality [53].

1.7 Summary of Structure

Section 4.1 presents a novel post-processing technique to improve the accuracy of the depth map estimated by a state of the art method. In Section 4.2, the challenge of depth estimation from monocular images is tackled by utilising Semi Parallel Deep Neural Network. Section 4.3 investigates general camera motions and introduces a method to estimate depth from a specific type of motion known as "small motions". Section 4.4 analyses the optical properties of the camera and take advantage of its focus/defocus modeling to generate a regularized dense depth map. Section 4.5 starts by studying the light field imaging theory. Next, a novel framework is proposed to estimate a dense depth map from the multi-camera array and light field image sets. Finally, Chapter 5 summarises the contribution of the whole thesis and discusses possible future works.

1.8 List of Published and "Under Review" Publications

1.8.1 Journals & Major Conferences

- **H. Javidnia** and P. Corcoran, "A Depth Map Post-Processing Approach Based on Adaptive Random Walk With Restart," in *IEEE Access*, vol. 4, pp. 5509-5519, 2016. doi: 10.1109/ACCESS.2016.2603220.
- Bazrafkan S, **Javidnia H**, Lemley J, Corcoran P, "Depth from Monocular Images using a Semi-Parallel Deep Neural Network (SPDNN) Hybrid Architecture," arXiv preprint arXiv:1703.03867. 2017 Mar 10. *Under Review*
- **H. Javidnia** and P. Corcoran, "Accurate Depth Map Estimation From Small Motions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (ICCV)*, Venice, Italy, 2017, pp. 2453–2461. doi: 10.1109/ICCVW.2017.289.
- **Hossein Javidnia**, Peter Corcoran, "Application of preconditioned alternating direction method of multipliers in depth from focal stack," *Journal of Electronic Imaging* 27(2), 023019 (6 April 2018). doi: 10.1117/1.JEI.27.2.023019.
- **Javidnia H**, Corcoran P, "Total Variation-Based Dense Depth from Multi-Camera Array," arXiv preprint arXiv:1711.07719. 2017 Nov 21. *In Press - Optical Engineering.*

1.8.2 Filed Patents Applications

- **Javidnia, Hossein**. "Depth Map Post-Processing Approach Based on Adaptive Random Walk with Restart." U.S. Patent Application 15/654,691, filed July 19, 2017.
- **Javidnia, Hossein**. "Depth Map Post-Processing Approach Based on Adaptive Random Walk with Restart." U.S. Patent Application 15/654,693, filed July 19, 2017.

1.8.3 Additional Conference Publications

- **H. Javidnia**, S. Bazrafkan and P. Corcoran, "The application of deep learning on depth from multi-array camera," in *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, USA, 2018, pp. 1-2. doi: 10.1109/ICCE.2018.8326322.
- **H. Javidnia** and P. Corcoran, "Real-time automotive street-scene mapping through fusion of improved stereo depth and fast feature detection algorithms," in *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, 2017, pp. 225-228. doi: 10.1109/ICCE.2017.7889293.
- **H. Javidnia**, A. Ungureanu, C. Costache and P. Corcoran, "Palmprint as a smartphone biometric," in *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, 2016, pp. 463-466. doi: 10.1109/ICCE.2016.7430692.
- A. S. Ungureanu, **H. Javidnia**, C. Costache and P. Corcoran, "A review and comparative study of skin segmentation techniques for handheld imaging devices," in *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, 2016, pp. 530-531. doi: 10.1109/ICCE.2016.7430717.
- **H. Javidnia**, A. Ungureanu and P. Corcoran, "Palm-print recognition for authentication on smartphones," in *Proceedings of the IEEE International Symposium on Technology and Society (ISTAS)*, Dublin, 2015, pp. 1-5. doi: 10.1109/ISTAS.2015.7439441.

Chapter 2.
On the Importance of Depth Estimation and 3D Information

Compact and lightweight mobile technologies such as smartphones have evolved significantly while their prices are constantly diminished. The advancement of consumer devices with high-resolution displays and 3D graphics capabilities has introduced a new generation of mobile Augmented Reality (AR) apps which are known to mix the real environment with the virtual. The first 15 years of the 21st century has seen major and rapid development in AR. Leaders of the consumer electronics industry such as Google [54] and Apple [55] are utilising AR to provide an interactive experience for users. Last year Google introduced Lens [56] which allows users to browse the environment through smart text selection and style match. Apple's ARKit [57] and Google's ARCore [58] are the new examples of the Visual Inertial Odometry (VIO) systems which allow both developers and consumers to take advantage of AR on their smartphones. Both devices can track users' position in space using accelerometer and gyroscope. They generate 3D models based on a sparse 3D point clouds which uses much less memory and CPU time.

Social media are utilising AR to give a better and more creative experience to their users. The tourist industry is taking advantage of the technology to help travelers in choosing their destination and planning their trips. The video game industry is investing in advanced solutions that incorporate AR technology and mobile gaming to improve their customers' experiences. All these applications require accurate depth data to create a realistic augmented scene.

Consumer depth sensing sensors, motion-sensing controllers, and virtual human interfaces have become a part of daily human-computer interaction [59]. Head-mounted displays (HMD) are another example of consumer devices which utilise 3D information for virtual reality purposes and became important for medical, gaming and military applications to view a 3D scene with $360° \times 90°$ angle of view [60], [61].

Autonomous vehicles are revolutionising our mobility behavior by enabling safer and more reliable transportation. Major auto manufacturers have already released, or are soon to release, self-driving features that give the car some ability to drive itself. In order to operate safely, autonomous vehicles will require high precision depth and 3D data which contain significantly detailed information about their surrounding environment [62].

Most of the main smartphone companies [63]–[65] are using depth and 3D information in their products to give a new and different experience of digital photography to their customers. Having access to the depth information on a smartphone enables consumers to virtually interact with the environment by remove objects, isolate single objects, and even remove and replace the entire background of a picture. The rapidly growing use of the depth sensing technologies in smartphones may entirely change the way that people will interact with technology in the near future.

Another area of computer vision that takes advantage of depth sensing sensors and 3D reconstruction methods is 3D facial recognition. Most of the existing recognition systems use 2D images to identify a person; however, many of these systems can be fooled using photographs or video clips known as 2D-media attacks [66]. 3D face modeling provides more accurate details than 2D models for recognition purposes and liveliness detection, such as the iPhone's new Face ID, which employs a dot projector sensor to generate a 3D facial map [67]. However, developing such a technology and its integration with lightweight mobile devices is a costly process.

Considering the current trend of employing depth sensing technologies on handheld and wearable devices, it is expected that 3D cameras will be a standard part of smartphones within the next few years. This technology is not restricted solely to the applications mentioned above. Depth and 3D information introduce new aspects to different fields of computer vision, such as object detection and tracking [68], [69], medical applications [70]–[72], semantic segmentation [73] and micro aerial vehicle navigation [74]–[76].

2.1 Common Depth Estimation Systems and the Challenges

2.1.1 Time of Flight (ToF)

ToF cameras work by calculating the time required for a ray of light to travel from a light source to an object and back to the sensor. Generally, these cameras only perform well at short range (< 2m); increasing the power of the illumination source can, to some degree, increase the sensing range (8-9m). The resolution of a typical ToF camera is usually about 200×200 [77]. The highest resolution ToF sensor currently available in the market is Microsoft Kinect for Xbox One with a 512×424 pixel resolution [78]. Kinect sensor has the capability of being modulated at up to 130 MHz, however, there are not many discrete frequency settings used to resolve the phase wrapping ambiguities [79] which causes the measured distance in the depth map to be much shorter than the actual distance. Another issue is that the raw data from the sensor and the firmware to control the signal generator are complex and the algorithms are not publicly available, so it is difficult to base research work on such devices. Microsoft HoloLens is a recent example of commercial ToF cameras. HoloLens is known as a mixed reality device which combines real and digital worlds. It displays at 60 fps and has the memory limit of 900MB RAM. The goggles are equipped with a camera which helps the device to locate the objects in the environment and project 3D images on top of them [16].

Nevertheless, there is some public research work on ToF cameras. To challenge the phase-based sampling of the current ToF sensors, Kadambi et al. [80] proposed a new ToF architecture for depth estimation purposes inspired by micron-scale microscopic interferometry. Although in this architecture the depth sensing range is increased without the presence of wrapping artifacts (the measured range is equal to the actual range), the prototype is designed on a phase ToF CMOS sensor which is not suitable for frequency sweeps. To challenge the low-resolution issue of the existing ToF cameras, Schuon et al. [81] showed that it is possible to generate high-quality depth maps from ToF cameras using multi-frame super-resolution methods, which are based on [82]. The initial depth map in [81] is regularized using a bilateral filter. This method requires a high number of frames (15 frames) to estimate the initial depth map. It also assumes that the motion in capturing the frames is purely translational. Meaning that the motion does not contain any rotation which is very unlikely for practical applications.

Zhu et al [83] focused on conditions such as highly textured scenes where ToF sensors do not perform well. To solve this, they proposed a method to combine the information from the ToF sensor and the passive stereo matching method. Their proposed method combines the probability distribution functions in depth from each method using a Markov random field (MRF) model. In a similar study, Marin et al. [84] upsampled the ToF depth data using bilateral filter and image segmentation. Parallel to the upsampling process, they estimated a dense depth map using the Semi-Global Matching (SGM) stereo algorithm. Later, a confidence map is calculated for each depth map. The depth information from ToF and SGM are combined at the final step by considering the local consistency of depth data. In another depth fusion study similar to [84], Agresti et al. [85] upsampled the ToF depth data using bilateral filter and image segmentation and they used SGM for stereo matching purposes. However, the confidence map estimation for ToF depth data and SGM is done using a Convolutional Neural Network (CNN). The proposed CNN in [85] takes as input both ToF and stereo clues and outputs the confidence map for each of them. The upsampled ToF data and the stereo disparity are finally fused together to construct the final depth map.

In another study, Noraky et al. [86] studied the power limits in ToF cameras. To minimise the amount of power required for depth estimation, they utilised the motion across images collected alongside the ToF camera to estimate a new depth map without illuminating the scene. They estimate the camera motion using optical flow based on block matching. The paper claims that the implementation of the method on ODROID-XU3 board consumes a total of 678 mW.

2.1.2 Structured Light

The concept of depth estimation using structured light has been studied since the 1990s [87]–[90]. Structured light methods utilise a projector and a camera. The projector illuminates the scene using patterned light such as a single point pattern or coded pattern [91]. The RGB camera captures the pattern reflection, and the depth information is calculated using triangulation. The key challenges of structured light systems are projector calibration and light interference. The difficulty in calibrating the projector comes from the fact that the sensors cannot capture images actively [92]. The projector might be considered an inverse camera [93], however, it is dependent on the camera parameters.

In the context of using structured light for depth estimation purposes, Scharstein et al. [94] proposed a method to automatically capture high-resolution stereo image sets and their corresponding ground truth data. A pair of cameras and light projectors is used to capture the scenes. Each scene is illuminated using a sequence of structured lights, which results in each pixel being illuminated by at least one projector. These image sets are commonly known as Middlebury dataset.

In another study, Zhang et al. [95] addressed the problem of finding the optimum illumination for object surfaces located at different depth levels in the scene by proposing an adaptive illumination framework. The whole system contains a programmable projector, a camera, and a PC. Initially, the camera captures the whole illuminated scene. Afterward, the corresponding depth map is calculated and analysed to generate the next illumination pattern. Similar to many other structured light depth sensing systems, the performance of the method in [95] is limited by the low power of the projector.

Chan et al. [96] proposed an optical system to measure depth by projecting a periodic line pattern and a setting for triangulation purposes. In this design, the captured images are converted to the frequency domain to estimate depth information. The projector in [96] contains an IR laser diode and a computer-generated hologram. The evaluation indicates that the framework proposed in [96] is a valid design, however, it has limited applications. “This method can be only applied to the depth measuring of uniform and monotone surfaces: uneven surfaces will distort the results.” [96]

Fanello et al. [97] proposed an algorithm to estimate depth using structured light based on a learning-based classification-regression. The algorithm is independent of the general window matching that is commonly used in stereo matching methods. Each pixel is classified using a label that corresponds to the subpixel position. An IR projector is also used in the setup to generate random dot patterns on the scene. The algorithm learns to recognise the class of the labels in the input image and its depth.

Gupta et al. [98] introduced the concept of light concentration to overcome the challenge of strong ambient illumination, especially in the outdoor environment. The key idea [98] is to show that by properly distributing the light, acquisition time can be reduced. They also make the illumination adaptive using information determined from ambient light. This method requires a high number of input images and it fails in the presence of reflective surfaces, as they concentrate the ambient light.

Wang et al. [99] presented a new method to deal with the interference of the multiple structured light depth sensors. The method takes advantage of a plane sweeping algorithm. The gaps in the initial depth information, which is obtained from the sensors, are filled by maximizing the likelihood of the projector-camera constraint.

In general, structured light systems have been used for a variety of applications during the past decade such as 3D face recognition [100], plant phenotyping [101], [102], underwater imaging [103]. However, they have a number of limitations such as:

- Ambiguity of measurements.
- Multiple reflections.
- Mixed measurements.
- Sensitivity to material reflectance.
- Sensitive to background lighting.

2.1.3 LIDAR

LIDAR scanners have a similar framework to ToF cameras, however, they use laser scanners to gather depth information [104]. They are generally expensive and bulky devices, as they contain laser scanning hardware. The advantage of these systems is their performance in providing long-range (up to 1000m) depth information. Generally, LIDAR scanners provide a 3D point cloud. The transformation of the point cloud to a 2D image plane generates a sparse distribution of the points. This issue has been addressed in [105] where the 3D point cloud is solely processed using a sliding window and bilateral filter to generate a dense depth map. The same issue has been tackled in [106] where a LIDAR scanner and a CCD camera are synchronously used to capture a scene. It is assumed that the LIDAR and the camera are perfectly calibrated so that each 3D point corresponds to a pixel in the RGB image. Then, the 3D points are projected onto RGB images. The sparse depth map is later upsampled into a dense depth map using a self-adaptive method where the RGB image and the anisotropic diffusion tensor are utilised to guide upsampling. The final result is refined by applying convex minimisation. The same problem is studied in [107] using the similar self-adaptive method. The framework presented in [107] also calculates the normal map using a trilateral filter which is based on the depth map and the RGB image.

Maddern et al. [108] proposed a probabilistic method to combine the sparse 3D data from LIDAR with stereo matching data to generate a dense depth map. The probabilistic approach is based on [39] and pyramid interpolation is employed for upsampling purposes.

Ding et al. [109] claimed that it is not required for the camera and the LIDAR sensor to be deployed dependently. They use a set of RGB images and a Structure from Motion (SfM) method to estimate a rough 3D structure. Using the estimated alignment information from SfM, the 3D LIDAR points are reprojected onto image planes and are used to estimate the dense depth map. The depth maps are later post-processed using a bilateral filter.

To provide a standard benchmark to evaluate the methods which have been developed for depth estimation purposes, Geiger et al. [110] presented a benchmark commonly known as KITTI, which is captured in rural areas. Their recording platform is equipped with high-resolution cameras, a laser scanner to generate the ground truth depth map and a localisation system. To minimise the registration error, the cameras are triggered by the laser scanner.

Premebida et al. [111] proposed a strategy to use the depth data captured by a laser scanner for pedestrian detection. Two feature maps are extracted from RGB and depth images using HOG features. A multiscale deformable part model [112] is later trained on the feature maps to learn the part positions and bounding boxes. Tan et al. [113] proposed a framework to detect curbs for driver assistance systems. They recover the dense depth map using a filter-based fusion system presented in [114]. Based on the normal map of the surface, a Markov chain model is created to capture the consistency property of the curb.

LIDAR scanners have also been used intensively for Simultaneous Localisation And Mapping (SLAM) and navigation purposes during the last decade [115]–[123].

2.1.4 Multi-Camera Approaches

Multi-camera or Multi-view approaches aim to estimate depth information from multiple viewpoints. The cameras are placed in a different position to capture a scene from different angles. These types of depth estimation methods can be divided into several categories such as dual cameras or stereo view, multi-array cameras, depth, and SfM. Generally, the depth estimation in multi-view methods (more than two) is divided into two steps. Initially, a depth map is estimated for each viewpoint and then the depth maps are merged to generate the final depth map or 3D structure.

2.1.4.1 Binocular Stereo Matching

Traditional binocular stereo matching systems utilise the horizontal displacement of a point to estimate disparity information. In human vision, disparity refers to the difference in the location of an object/point in two images (left and right) [124] and is inversely proportional to depth, meaning that an object that is located at a further point has a smaller horizontal displacement or disparity than an object that is closer to the camera (larger disparity.) Considering the focal length of the camera and the distance between two cameras, which is called the baseline, the disparity can be converted to depth.

For left and right images in a binocular stereo, the pixel (x, y) in the disparity map is calculated by finding the distance between the pixel (x, y) in one image (e.g. left image) and the corresponding pixel (x', y') in the second image (e.g. right image.)

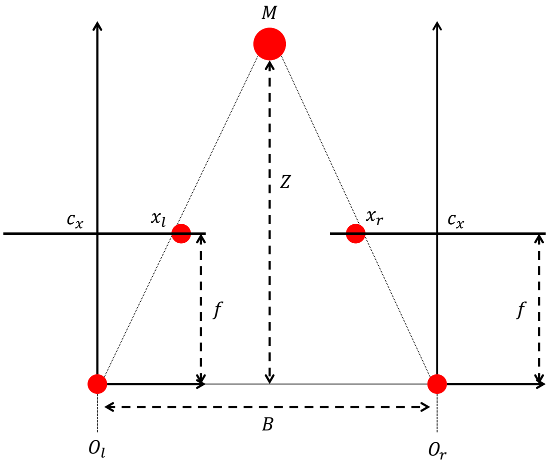


Figure 2.1: The inverse relation of depth and disparity

Figure 2.1 illustrates the inverse geometrical relation between depth and disparity where the disparity value is $d = x_l - x_r$. The left and right pixels are shown as x_l and x_r, respectively. f represents the focal length of the camera, B shows the baseline or the distance between two cameras and M is the object. O_l and O_r represent left and right cameras, respectively. c_x shows the projection of the object M in both cameras. The conversion of disparity to depth can be defined as Equation (2.1):

$$Z = \frac{fB}{x_l - x_r} \tag{2.1}$$

In general, traditional stereo matching methods consist of four steps [124]: computing matching cost, cost aggregation, disparity estimation, and disparity refinement.

To find the closest match in the target image to the reference image in a stereo set, a matching cost has to be calculated for each candidate. The most commonly used matching costs in the state of the art include Sum of Square Difference (SSD), Sum of Absolute Difference (SAD), Absolute Differences (AD), and Census transform [125]. Considering the intensity of the reference pixel in the left image as (x_l, y) and a candidate pixel in right image as $(x_l - d, y)$ then, SSD, SAD and AD can be calculated using the following equations.

$$SSD(x_l, y, d) = \sum_{x,y} \big((x_l, y) - (x_l - d, y)\big)^2 \tag{2.2}$$

$$SAD(x_l, y, d) = \sum_{x,y} |(x_l, y) - (x_l - d, y)| \tag{2.3}$$

$$AD(x_l, y, d) = \big((x_l, y) - (x_l - d, y)\big) \tag{2.4}$$

Note that SSD and SAD matching costs are usually calculated using a window search with the size of $((2w + 1) \times (2w + 1))$ where (x_l, y) is the central pixel and w is the search range.

Census transform relies on the ordering of the intensity values rather than their absolute values. This transform maps the intensity values of the pixels within a square window to a bit string, which results in capturing the structure of the image. Comparisons are done such that if a pixel in a window has an intensity smaller than the center pixel, it is marked as 1. If the intensity of the pixel is larger than or equal to the center pixel, then it is marked as 0. The final matching cost is calculated using the, Hamming distance of the binary vectors.

Based on [124], the matching costs are summed and averaged in a given region at the cost aggregation step. This step results in a smoother matching cost and it reduces disparity estimation mismatches.

Generally, the process of choosing the correct disparity value for a pixel starts by taking the minimum cost within a disparity range. This process is known as Winner-Take-All (WTA.) The main problem in WTA is that the disparity value may be estimated incorrectly due to the presence of strong local minima. It has been suggested to use a confidence matrix to solve this issue [126], [127].

The disparity refinement step aims to enhance the accuracy of the disparity maps using a variety of filters such median, bilateral or left-right, right-left consistency checks.

The depth from motion or SfM methods use common feature points in multiple images to generate a 3D structure. The common features are tracked throughout the images and the corresponding 3D point can be reconstructed by triangulation. In a similar category as SfM, Simultaneous Localisation And Mapping (SLAM) has made astonishing progress over the last 40 years, enabling large-scale real-world applications, and witnessing a steady transition of this technology to industry. Most of the SLAM methods present the 3D models as a set of sparse points corresponding to the features of the scene such as lines and corners [128]–[130]. Unlike the feature based SLAM, dense representations are introduced to provide high resolution models of the 3D geometry [131]–[133]. Generally dense models are visually more pleasant. However, they are usually cumbersome and they require a large amount of storage to save the data.

2.1.4.2 Literature Review

Many disparity and depth estimation methods have been developed since the 1970s with the goal of estimating depth from two or more views [134]–[136]. Due to the large distribution of these methods in different applications and fields, we refer to Middlebury Stereo Vision benchmark [124], [137] where most of the state of the art algorithms are evaluated based on their performance. Although the high accuracy algorithms provide better visual and quantitative results, their iterative refinement processes and computational times require large memory and computational power.

Taniai et al. [37] proposed a stereo matching method based on local expansion moves which utilises the spatial propagation of graph cuts. The per-pixel 3D plane labels are deduced on a pairwise MRF, which is based on the heuristics in PatchMatch. In another attempt, Li et al. [38] proposed a cost-aggregation method that utilises a minimum spanning tree to aggregate 3D costs. To reduce the complexity of computing the cost for every pixel, they developed several minimum spanning tree structures for cost aggregation. Drouyer et al. [138] presented a refinement method to densify sparse or noisy depth maps using hierarchical segmentation, which is accomplished by modeling each noisy or incomplete part of the disparity map using a bivariate linear polynomial. Li et al. [139] challenged the issue of assigning 3D labels to each pixel in stereo images for more accurate depth estimation by proposing an algorithm called PatchMatch-based Superpixel Cut. 3D labelling methods simultaneously estimate the disparity and normal direction of a pixel and they are optimised using superpixels. They also proposed a bilayer matching scheme based on a pre-trained CNN that measures the similarity of 3D labels.

Zhang et al. [140] presented a global method to estimate depth and generate 3D structure from stereo images. The initial idea starts by splitting the images into 2D triangles with joint vertices. Each triangle is modeled by a slanted plane, and a 2-layer MRF optimisation is used to model the depth discontinuities at the object boundaries. One layer is responsible for modeling the splitting properties to properly split the vertices in the 3D model, and the other layer optimises region-based stereo matching.

Wei et al. [141] proposed a method to estimate depth from multi-view stereo images. Their proposed method is based on the PatchMatch algorithm and it starts by initializing a sparse depth map. The images and estimated depth maps are later down-sampled and a propagation-filtering approach is used as part of the hierarchical estimation. Outliers are then eliminated by cross-view filtering which helps in propagating reliable information. A consistency check is applied after upsampling the depth and normal maps, and they are refined using an edge-aware filter.

Very recently, many deep learning approaches have been developed to estimate depth from stereo [142]–[147] and multi-view images [148], [149].

Park at al. [150] proposed a CNN model to learn the stereo matching cost by designing a per-pixel pyramid-pooling layer, which is the modification of spatial-pyramid-pooling presented in [151]. The proposed layer performs multiple pooling, with different window sizes to respect fine details of the scene, and the final feature map is generated by concatenating the outputs of each pooling.

Ye et al. [152] focused on the first and last steps of the stereo matching pipeline. They modeled the matching cost step by proposing a patch-based network, and the refinement step used a regression network architecture. The cost aggregation is based on SGM. Two initial disparity maps (left and right) and the left RGB images are used for refinement purposes. The refinement is based on a probability error map calculated from the fusion of two depth maps. Unlike [152], Liang et al. [153] proposed a CNN architecture which performs all the common steps of stereo matching algorithms. Their CNN model consists of three parts. The first part is responsible for extracting multiscale shared features. The second part calculates the initial disparity estimation, which is later refined by the third sub-network. The disparity estimation sub-network utilises encoder-decoder architecture, and the refinement process in part three is formulated as a Bayesian inference process.

2.1.5 Learning-based Single Camera

Most of the binocular or multi-view methods are able to estimate fairly accurate depth information. However, their computational time and memory requirements are important challenges for many applications. The idea of using the monocular image to capture depth information could potentially solve the memory requirement issue, but it is computationally difficult to capture the global properties of a scene [154] such as texture variation or defocus information. This topic is one of the challenging research fields in computer vision that has been recently tackled utilising deep learning techniques. The current state of the art methods related to the development of deep learning techniques for monocular depth estimation utilise large network architecture with millions of parameters. Although these networks have fast computational time, they are not optimised enough to be implemented on low power hardware. Most of these models also perform on low-resolution images for memory efficiency purposes and to reduce the computational overload.

Saxena et al. [154], [155] proposed a supervised learning approach to estimate depth from a single image. The image is divided into small patches and the depth is estimated for each patch. The depth of each patch is estimated using two types of features which represent the absolute depth of the patch and its difference with another neighboring patch. Later, an MRF is used to model the differences in depth between a patch and its other neighbors.

Liu et al. [156] proposed a deep CNN by formulating the single image depth estimation task as a Conditional Random Fields (CRF) learning problem. The image is divided into superpixels. The depth of each superpixel is represented by its central pixel. Later, the conditional probability distribution of the continuous depth values of all superpixels are modeled based on CRF and the depth is estimated by solving a Maximum A Posteriori Probability (MAP) problem.

Kuznietsov et al. [157] presented a semi-supervised approach to estimate depth from monocular images which takes advantage of both supervised and unsupervised learning. The supervised part of the model is trained on a set of sparse depth maps generated by LIDAR scanner. To complement the trained model, the geometry principles of stereo matching are used to learn depth prediction in an unsupervised manner. The network architecture is based on the encoder-decoder scheme similar to [158].

Garg et al. [159] proposed an unsupervised framework to estimate depth from a single camera. The stereo image sets and the motion between two frames are used for training. A convolutional encoder is trained to learn the transformation from an image to the depth map. The loss used for learning is the photometric difference between the input image (e.g. left image) and the inverse warped target (e.g. right image). Similar to [159], Godard et al. [160] proposed an unsupervised approach to estimate depth from monocular images. However, their method utilises a left-right disparity consistency loss which results in a higher accuracy disparity map.

Eigen et al. [161] proposed a network that consists of two sub-networks to estimate depth from single images. The first network aims to globally estimate the depth of the scene, and the second network is designed to refine the global depth map within its local regions. The concept of augmentation is also used in this paper where the training data is scaled, rotated and flipped. The input data is also down-sampled to half for training purposes, which requires the final depth maps to be upsampled.

2.2 Comparison of the Depth/3D Imaging Methods

All the depth estimation methods described in the previous section have their own strengths and weaknesses. Table 2.1 presents a brief comparison between the common depth estimation methods including their strengths and weaknesses. Ten factors are chosen to describe the general performance of each method in terms of range, depth accuracy, resolution and scanning speed etc. As an example, ToF cameras take advantage of active illumination. Their performance varies under different lighting condition and they have quite high power consumption. ToF cameras usually cover short scanning range unless the power of the pulse emitting source is increased. The accuracy of the depth varies from mm to cm based on the resolution of the sensor. ToF cameras have fast scanning speed but they provide QQVGA and QVGA output. Besides their low overall system cost and real-time capability, they are sensitive to scattered light and sunlight.

Despite all the limitations of the depth sensors, they give computers an entire dimension of data, expanding computer vision applications and their capability. Among all the sensors presented in Table 2.1, ToF camera is the promising technology which could be used for consumer imaging. Especially lately, as imaging sensors have been developing rapidly, the high cost and low resolution are not the problems anymore. With the low cost ToF cameras on consumer devices, a new application area can be expected to emerge.

	Time of flight	Stereoscopic vision	Fixed structured light	Programmable structured light	LIDAR	Learning based	Structure from Motion/SLAM
Operational principle	IR pulse, measure light transit time	Two 2D sensors emulate human eyes	Single pattern visible or IR illumination, detects distortion	Multiple pattern visible or IR illumination, detects distortion	Laser illumination	Trained model using deep learning	2D Feature tracking and triangulation
Point cloud generation	Direct out of chipset	High SW processing	Medium SW processing	SW processing scales with # of patterns	Direct out of chipset	High SW processing	Medium SW processing
Active illumination	Yes	No	Yes	Yes – Customizable spectrum	Yes	No	No
Low light performance	Good	Weak	Good	Good	Good	Weak - Unless trained for that	Medium/Good
Bright light performance	Medium	Good	Medium / weak Depends on illumination power	Medium / weak Depends on illumination power	Good	Medium	Good
Power consumption	Medium / High / Scales with distance	Low	Medium	Medium / Scales with distance	High	Low / Medium	Low / Medium
Range	*Short (0.2m) to long (8m) range *Depends on laser power & modulation	Depends on spacing between cameras	*Depends on spacing between camera and projector *Depends on illumination power	*Depends on spacing between camera and projector *Depends on illumination power	Short (1m) to very long (1000m) range	Depends on training data	Depends on camera motion. Can vary from cm to m
Resolution	QQVGA, QVGA	Camera Dependent	Projected pattern dependent	WVGA to 1080p	Depends on the laser module	QQVGA, QVGA	Camera Dependent
Depth accuracy	mm to cm Depends on resolution of sensor	mm to cm Difficulty with smooth surface	mm to cm	µm to cm	mm	mm to m Depends on the trained model	mm to m Depends on the optimization
Scanning speed	Fast Limited by sensor speed	Medium Limited by SW complexity	Medium Limited by SW complexity	Fast / medium Limited by camera speed	Fast / medium Limited by sensor speed	Medium Limited by SW complexity	Medium / Fast Limited by SW complexity
Other Strengths	*The scene is recorded all at once and doesn't have to be scanned *2D and 3D information in a multi-part image *Compact system without moving components *No structure or contrast required *Large working distances are possible with a sufficiently strong light source *Low overall system costs *High real-time capability	*Possibility to achieve high accuracy at short range *2D area scan cameras can be used	*Possibility to achieve high accuracy at short range *2D area scan cameras can be used *Can be optimized for real-time applications		*Very high accuracy *Difficult lighting conditions are not a problem *No problems with mirroring or highly reflective surfaces *Suitable for real-time applications	*Can be optimized for low resolution real-time applications *There is a potential to estimate depth from one camera	*Possibility to achieve high accuracy at short range *2D area scan cameras can be used *Can be optimized for real-time applications *Depth can be estimated using one camera
Other Weaknesses	*Sensitive to scattered light *Difficulties with sunlight	*Will not work on homogeneous surfaces *High computing load makes real-time capability difficult *Exposure to sunlight is a problem *Will not work with highly reflective surfaces	*High overall system costs due to complex setup and high installation cost *Limited to short scanning range		*Very expensive individual components *High overall system costs due to complex setup and high installation cost	*Highly dependent on graphics processing unit power *High computing load makes real-time capability difficult for high resolution images *There is no guarantee to achieve a specific accuracy	*Exposure to sunlight is a problem *Will not work with highly reflective surfaces *Posibility of failure while encountering with texture-less surfaces

Table 2.1: Comparison of the depth/3D imaging technologies

Chapter 3.
Research Contributions to the Problem of Depth Estimation

There are five primary contributions for this research. First, for the purpose of exploring the challenges of estimating depth from stereo image sets, we develop a novel generic post-processing method to increase the structural accuracy of the depth maps calculated by state of the art techniques. Second, to overcome the extensive computational power of the stereo matching techniques, we tackle the ill-posed problem of depth estimation from monocular images by exploiting the relation between colour pixels and depth using end-to-end trainable CNN architecture. Third, since the consumer cameras usually employ pre-capturing initialization, we propose a novel method by utilising the pre-capturing small motions to estimate a depth map and generate a semi-dense 3D structure of a scene. Fourth, we take advantage of the optical properties of consumer cameras to estimate depth from a stack of images captured in different focal planes. Fifth, since the consumer multi-camera array in smartphones has received much attention in the past years, we develop a framework to generate a high-quality continuous depth map from these cameras.

3.1 A Depth Map Post-Processing Approach based on Adaptive Random Walk with Restart

The copy of the paper published based on this section is presented in Appendix A. This section was published in:

H. Javidnia and P. Corcoran, "A Depth Map Post-Processing Approach Based on Adaptive Random Walk With Restart," in *IEEE Access*, vol. 4, pp. 5509-5519, 2016. doi: 10.1109/ACCESS.2016.2603220

3.1.1 Overview

To increase the structural accuracy of the depth estimation methods, we proposed a generic post-processing method to preserve the structure of the reference image including corners and edges in the depth map. The proposed method utilises the mutual information between the RGB image and the initial depth map, and normalised-interpolated convolution. The method is implemented on top of a state of the art stereo matching algorithm known as

Adaptive Random Walk with Restart [51]. A quantitative evaluation is done based on Middlebury benchmark [124], [137] which indicates the competitive performance of the proposed method against the top stereo matching algorithms. The final depth maps calculated using the proposed post-processing method are later compared qualitatively with the Google Lens Blur application, which is used to generate Bokeh effect in smartphone images.

Based on Table 2.1 the proposed approach has the advantage of generating depth without using active illumination. The method can perform accurately on high resolution images (2864×1924 pixels) as shown in Appendix A. The depth accuracy is high compared to the top state of the art algorithms. However, it has some disadvantages such as high computing load, poor performance while encountering with reflective surfaces and texture-less regions.

3.2 Semi-Parallel Deep Neural Network (SPDNN) Hybrid Architecture, First Application on Depth from Monocular Camera

The copy of the paper published based on this section is presented in Appendix B. This section was published in:

Bazrafkan S, **Javidnia H**[*], Lemley J, Corcoran P. Depth from Monocular Images using a Semi-Parallel Deep Neural Network (SPDNN) Hybrid Architecture. arXiv preprint arXiv:1703.03867. 2017 Mar 10. *Under Review*

The first two authors have contributed equally to this paper.

[*]Contributions include: Data preparation including the stereo depth using the method presented in the previous section, experiments and evaluation, manuscript preparation and literature review.

3.2.1 Overview

Generally, stereo matching methods as mentioned in the previous section are computationally intensive. Although stereo images can provide valuable depth information, they require a significant amount of memory and an accurate calibration. Also, in the consumer industry, most of the devices are already equipped with a single camera which is more convenient for users. To address this issue, we present a Semi Parallel Deep Neural Network (SPDNN) to estimate depth from monocular images. A semi-parallel network topology is developed using a graph theory optimisation of a set of independently optimised Convolutional Neural Networks (CNN.) In this study, four SPDNN models are trained and have been evaluated at two stages on the KITTI dataset [162]. To evaluate the performance of the post-processing method presented in the previous section, we trained two of the networks using the post-processed depth maps.

Based on Table 2.1 the proposed approach is capable of estimating depth from monocular images. It does not require active illumination. It can be optimised for real-time applications. However, the resolution of the output image is low (80×264 pixels), and working on high resolution images introduces a high memory load. Also, this approach is highly dependent on a Graphics Processing Unit (GPU) which requires optimisation to be applicable for embedded platforms.

3.3 Accurate Depth Map Estimation from Small Motions

The copy of the paper published based on this section is presented in Appendix C. This section was published in:

H. Javidnia and P. Corcoran, "Accurate Depth Map Estimation From Small Motions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (ICCV)*, 2017, pp. 2453–2461. doi: 10.1109/ICCVW.2017.289.

3.3.1 Overview

As described in the previous section, using DNNs can significantly decrease the computational time of the depth estimation task. However, it requires images to be down-sampled which results in information loss. Also, due to the high memory requirements of these models, they have to be optimised to be applicable for lightweight devices such as smartphones. This section investigates the use of random and small motions captured by a smartphone camera for depth sensing applications. By small motion, we refer to the pre-initialization of the camera before capturing the image or consumers' natural handshake. Our evaluation shows that the developed method is capable of producing semi-dense 3D point cloud and dense depth map with a higher structural accuracy in comparison to the state of the art. Due to the lack of ground truth, in this case, a separate evaluation is done against the Middlebury benchmark [124], [137] where we proved that the proposed method is also capable of processing images with baselines as large as 400 mm where the state of the art methods fail to provide a depth map.

This approach is in the category of SfM. The advantages and disadvantages of the method can be expressed based on the stereoscopic vision category in Table 2.1. The proposed approach is capable of producing a dense/semi-dense point cloud. It does not require active illumination. It can be used to process high resolution images and it can outperform the state of the art in terms of depth accuracy. However, the method requires an optimisation to be able to perform on real-time applications. There is a high chance of failure with texture-less and reflective surfaces. The scanning range is short, as it is designed for small motions.

3.4 Application of Preconditioned Alternating Direction Method of Multipliers in Depth from Focal Stack

The copy of the paper published based on this section is presented in Appendix D. This section was published in:

Hossein Javidnia, Peter Corcoran, "Application of preconditioned alternating direction method of multipliers in depth from focal stack," *Journal of Electronic Imaging* 27(2), 023019 (6 April 2018). doi: 10.1117/1.JEI.27.2.023019.

3.4.1 Overview

This section focuses on the sequence of frames captured with small baselines in different focal planes known as a focal stack. This optical feature of smartphone cameras can be used to estimate depth information. Initially, the images in the stack are aligned using Epipolar homography alignment. The initial depth is estimated using 3-point Gaussian distribution [163] and it is later refined by employing Preconditioned Alternating Direction Method of Multipliers (PADMM) [164], [165]. The performance of the proposed method is compared against a state of the art method and two commercial softwares. Preliminary results indicate that the proposed method has a better performance in terms of structural accuracy and optimisation in comparison to the current state of the art methods.

The proposed approach does not require active illumination and it is capable of generating a dense depth map. It covers short to mid scanning range. High resolution images can be processed in a fast computational time (~28 seconds). The method has a potential to be used for smartphone applications and it uses only one camera to capture the input sequence. The limitation of this approach is that it requires the image frames to be captured with a small motion (within 2cm) or translation.

3.5 Total Variation-Based Dense Depth from Multi-Camera Array

The copy of the paper published based on this section is presented in Appendix E. This section was published in:

Javidnia H, Corcoran P. Total Variation-Based Dense Depth from Multi-Camera Array. arXiv preprint arXiv:1711.07719. 2017 Nov 21. *Accepted for publication in Optical Engineering.*

3.5.1 Overview

This section studies the concept of depth estimation from a new type of multi-view camera known as a multi-camera array. These camera arrays contain a set of cameras arranged in a grid with a very small baseline. Recently this technology is being used as a replacement for smartphones' conventional camera. Similar to stereo matching methods, the existing algorithms which provide an accurate depth from the multi-camera array are computationally expensive. To address this issue, a framework is presented which utilises analysis of the local Epipolar Plane Image (EPI) to estimate depth from a multi-camera array. Later, the depth map is refined using Total Variation (TV) minimisation based on the Fenchel-Rockafellar duality [53]. Compared to the state of the art algorithms, the proposed method can estimate a dense depth map within a short computational time (~38 seconds).

This method does not require active illumination and it has a low computational overload. All the images are captured in one shot. It can be used to process high resolution images. The disadvantages of these cameras are the production cost and the short scanning range due to the small baseline between the cameras.

Chapter 4.
Overview of Methodologies

This section provides an overview to the technical details of the primary contributions of this research. First, a brief introduction to Adaptive Random Walk with Restart (ARWR) stereo matching method is provided. This method is used as the baseline for the proposed post-processing framework. Later the overview to the technical details of each research contribution is provided. More technical details for each method are provided in Appendices A-E.

4.1 A Depth Map Post-Processing Approach based on Adaptive Random Walk with Restart

4.1.1 A Brief Introduction to Adaptive Random Walk with Restart Stereo Matching

Random walk with restart is defined in [51] as Equation (4.1): Consider a random particle starting from node i. The particle iteratively moves to its neighborhood with the probability that is proportional to their edge weights which is calculated from Equation (4.3).

$$\vec{r}_i = c\overline{W}\vec{r}_i + (1-c)\vec{e}_i \tag{4.1}$$

where $\overline{W}$ is the normalised weighted matrix and $\vec{e}_i$ is $n \times 1$ starting vector with the i-th element 1 and 0 for others. Also at each step, it has some probability c to return to the node i. The relevance score of node j wrt node i is defined as the steady-state probability $r_{i,j}$ that the particle will finally stay at node j.

In [51] this equation has been defined as:

$$X_{t+1}^{d} = c\overline{W}X_t^d + (1-c)X_0^d \tag{4.2}$$

where $X_0^d = [F(s,d)]_{k\times 1}$ represents the initial matching cost. X_t^d denotes the updated matching cost with t as the number of the iteration, k is the number of super-pixels and $(1-c)$ is the restart probability. Note that $F(s,d)$ is the super-pixeling cost function. $\overline{W} = [w_{ij}]_{k\times k}$ which is the weighted matrix, contains the edge weights. These weights are influenced by the intensity similarity between neighboring super-pixels. So we can write:

$$w_{ij} = \exp\left(-\frac{\left(I(s_i)-I(s_j)\right)^2}{\sigma_e}\right) \tag{4.3}$$

where $I(s_i)$ and $I(s_j)$ are the intensities of the i-th and j-th super-pixels and σ_e is a parameter that controls the shape of the function. Note that the intensity of super-pixels is computed by averaging the intensity of the corresponding pixels.

One of the most common ways to solve the random walk is the iterative method, which is iterating the Equation (4.1) until convergence. In [51] this iteration has been applied to Equation (4.2).

This convergence happens when the L_2 norm of successive estimates of X_{t+1}^d is below a threshold ξ, or a maximum iteration step m is reached. Note that L_2 norm of a vector is the square root of the sum of the absolute values squared.

The main contribution of [51] was to integrate the visibility term and fidelity term into the cost function, Equation (4.2). So the new cost function looks like:

$$X_{t+1}^d = c\overline{W}\left((1-\lambda)V_t^d + \lambda\Psi_t^d\right) + (1-c)X_0^d \tag{4.4}$$

where Ψ_t^d is the fidelity term, V_t^d represents the visibility term and λ leverages the visibility and fidelity term. The final disparity map is computed by combining the super-pixel and pixel-wise matching costs:

$$\hat{d} = arg_d \min\left(X_t^d(s) + \gamma P(u,v,d)\right) \tag{4.5}$$

where s is the super-pixel corresponding to the pixel (u,v) and γ represents the weighting of the super-pixels and pixel-wise matching cost.

In the present research, the visibility term is eliminated from the cost function. So the modified cost function looks like:

$$X_{t+1}^d = c\overline{W}\left(\lambda\Psi_t^d\right) + (1-c)X_0^d \tag{4.6}$$

4.1.2 Process of the Proposed Post-Processing Method

The proposed post-processing method is implemented on top of the stereo matching algorithm presented in [51]. The algorithm starts by calculating the local matching cost by combining the Census transform and gradient image matching. Each pixel of a gradient image measures the change in intensity of that same point in the original image, in a given direction. The local cost is aggregated using Simple Linear Iterative Clustering (SLIC) superpixelling [166]. The aggregated depth map is later optimised by integrating the visibility term and fidelity term into the cost function of the random walk with restart. At the next step, the mutual information is calculated, which goes through the joint filtering block. The joint filtering block is based on the weighted median filter. Two new features are added to this filter to make it more robust to noise and respectful to a reference image's structure:

1- The window size of the median filter is adaptive
2- The weights are being allocated dynamically based on joint histogram

The similarity map is obtained from the mutual structure block. Based on the similarity map, a certain window size can be assigned to a specific block of pixels in the image. The filtered image from this stage is mathematically transformed from a 2-dimensional array of pixels into another domain Ψ_{ω} such as spatial frequency to facilitate removal of blocky artifacts within the disparity map where we applied normalised convolution followed by interpolated convolution. In normalised convolution, the box filter in the transformed domain is computed using a Summed Area Table (SAT). SAT is basically calculated by computing the cumulative sum along the specified dimension of the input data. The box filter is computed two times: once on a horizontal domain, and then the result is filtered on the vertical domain. This will give us the result of normalised convolution. In interpolated convolution, the box filter in transformed domain is computed using a SAT, but in this case the SAT is built using the area under the graph (in the transformed domain) of the interpolated signal. Again the same process, similar to normalised convolution, happens here to compute the box filter. Figure 4.1 presents the results of the proposed post-processing method on sample stereo sets from Middlebury database. The initial depth map is computed by ARWR which is later modified using the proposed post-processing framework.

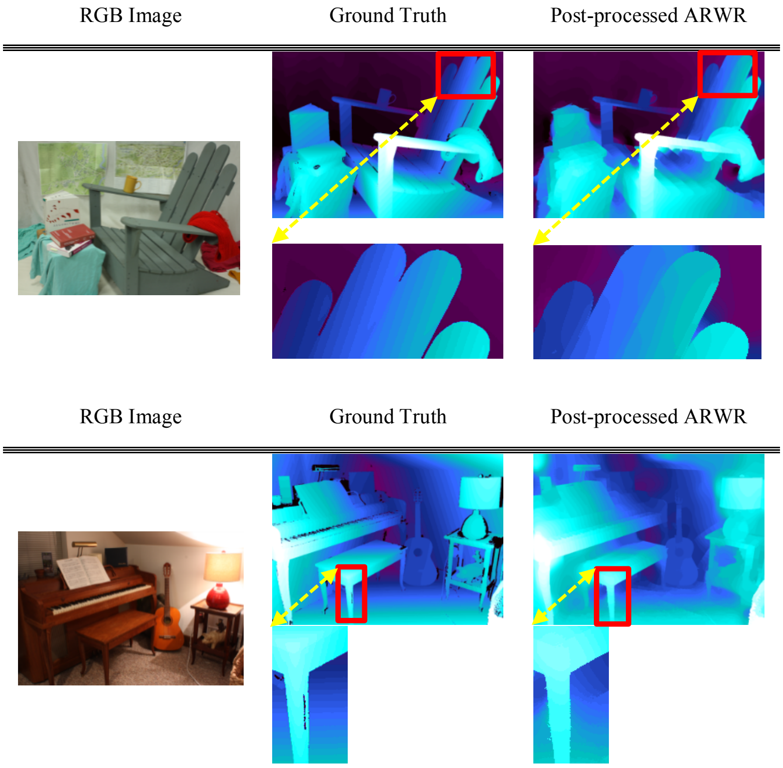


Figure 4.1: The result of the sample images from Middlebury database. Each set of figures denotes the left image, the ground truth and the proposed post-processed depth map

Figure 4.2 illustrates the general overview of ARWR along with the post-processing method. The ARWR depth estimation block can be replaced with any other state of the art depth estimation algorithm. The entire process can be described briefly as follows:

1. Extract the initial depth using the ARWR algorithm.
2. ***A.*** Apply mutual joint weighted median filter to fill the regions of occlusion or depth discontinuity in the initial depth map.

   B. Overwrite the structure of the RGB image on the depth map.
3. Transfer the depth map to a signal and perform normalised interpolated convolution on the domain of the signal to obtain an edges preserved depth map.

More details about the proposed post-processing method can be found in Appendix A. The code for the post-processing method and the stereo matching algorithm [51] is available at: https://github.com/hosseinjavidnia/Post-Processing-ARWR

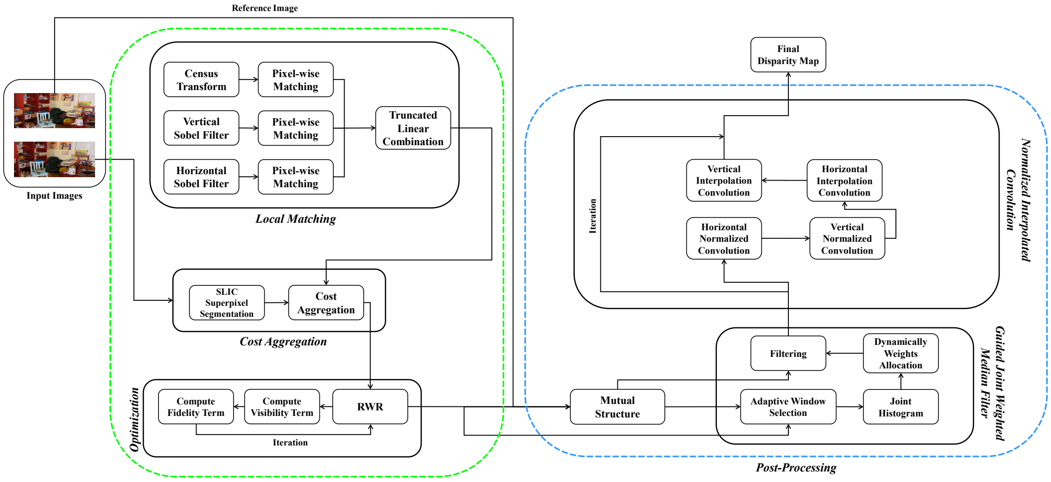
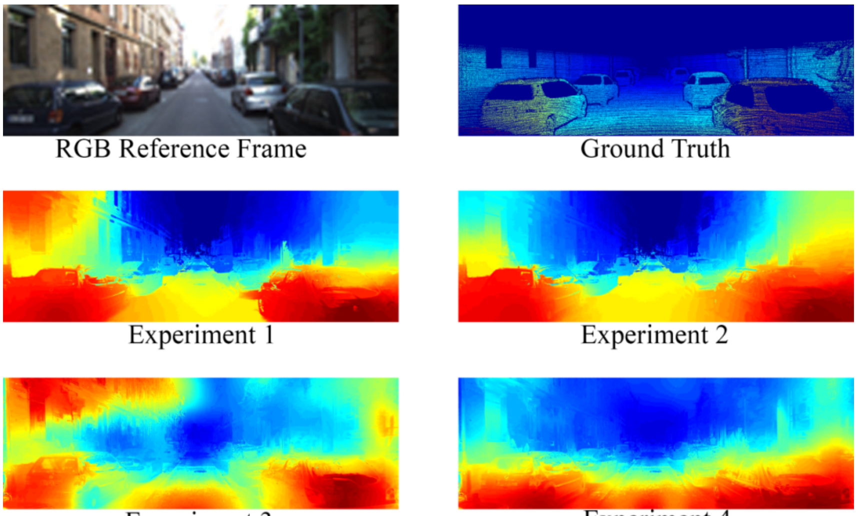


Figure 4.2: Overview of ARWR and the proposed post-processing method

4.2 Semi-Parallel Deep Neural Network (SPDNN) Hybrid Architecture, First Application on Depth from Monocular Camera

The concept of SPDNN is inspired by graph optimisation techniques. In this method, several deep neural networks are parallelized and merged in a novel way that facilitates the advantages of each. The merging of multiple networks using SPDNN is described in detail in the context of the current depth mapping problem in Appendix B. The process of estimating depth from monocular images using SPDNN starts by down-sampling the images to 80×264 pixels resolution. The initial depth is estimated using the trained models and is later upsampled to the original size using Joint Bilateral Upsampling [167]. Figure 4.3 represent the colour-coded depth maps computed by the trained models using the proposed DNN, where the dark red and dark blue parts represent closest and furthest points to the camera respectively. On the top right of each figure, the ground truth given by the benchmark is illustrated. For visualization purposes, all of the images presented in this section are upsampled using Joint Bilateral Upsampling [167]. The results show that using semantic segmentation along with the visible image as input will improve the model marginally. Using the post-processed target in the training stage helps the model to converge to more realistic results.



RGB Reference Frame

Ground Truth

Experiment 1

Experiment 2

Experiment 3

Experiment 4

Figure 4.3: Estimated depth maps from the trained models

Figure 4.4 presents the overview of the proposed framework where the depth map is estimated from a stereo set using the proposed post-processing method. Afterwards, the left RGB image from the stereo set is used as the input and the estimated depth map as the target. The network tries to formulate this problem and provide a model to estimate depth from one image.

In general four models are trained in this project:

1. **First Model:** Input: Left RGB Image + Pixel-wise Segmented Image. Target: Post-Processed Disparity.
2. **Second Model:** Input: Left RGB Image. Target: Post-Processed Disparity.
3. **Third Model:** Input: Left RGB Image + Pixel-wise Segmented Image. Target: Disparity.
4. **Fourth Model:** Input: Left RGB Image. Target: Disparity.

The contributions of this research are as follows:

1- A method to mix and merge several deep neural networks called “Semi Parallel Deep Neural Network (SPDNN)”, described in detail in Appendix B.

2- The application of deep neural networks and SPDNN on estimating depth from a monocular camera.

More details about the application of SPDNN in monocular depth estimation can be can be found in Appendix B. The code for this method is available at: https://github.com/hosseinjavidnia/SPDNN-Depth

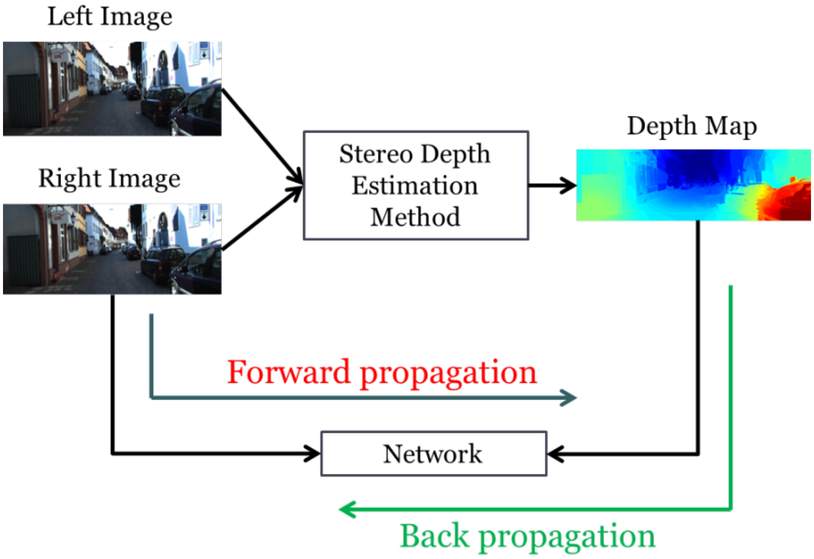


Figure 4.4: Overview of the proposed solution

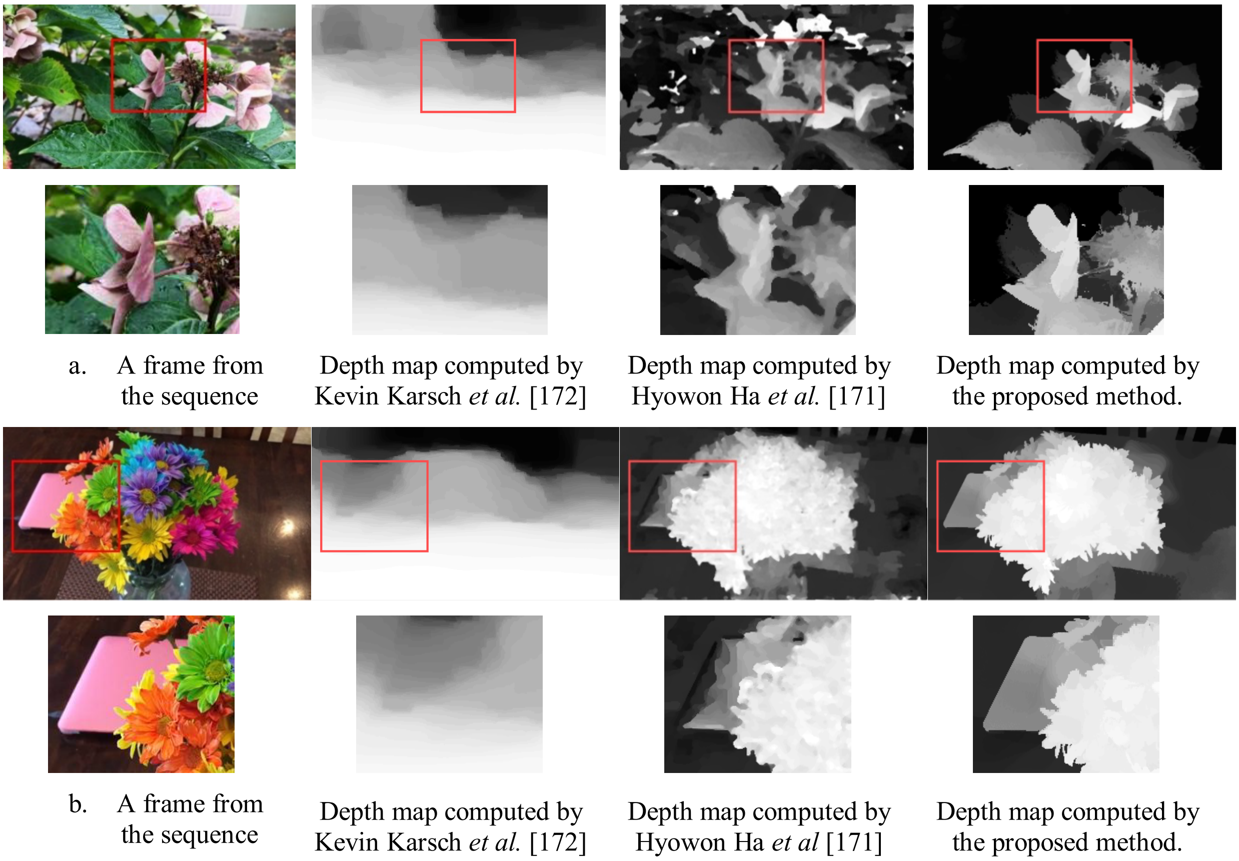
4.3 Accurate Depth Map Estimation from Small Motions

The algorithm starts by computing the relative pose, 3D points of the scene based on ORB features [168] and camera calibration details. Note that the features are tracked using the Kanade-Lukas-Tomashi (KLT) algorithm [169]. After computing the intrinsic and extrinsic details of the camera, the intensity profile is generated using the Plane Sweeping method [170] as presented in Section 3.1 in Appendix B. The pixel-wise matching cost is aggregated to volume cost, based on the colour and similarity of features. The matching cost from the image is weighted by a similarity feature. Once the cost volume is computed, the initial disparity map is obtained by parameterising the plane equation at pixel level with local disparity values. Later the disparity map is refined by defining a smoothness term and a data term as presented in Section 3.2 of Appendix C.

Figure 4.5 shows the depth map computed by Hyowon Ha *et al.* [171], Kevin Karsch *et al.* [172] and our method. These images show the performance of the proposed method in terms of accuracy of the depth along edges and the depth values on the surface of objects in the case of small motions and small baseline.

The results by Hyowon Ha *et al.* [171] and Kevin Karsch *et al.* [172] have inaccurate depth values along the edges and corners of the objects as seen in Figure 4.5.a and Figure 4.5.b. Note that due to the very small baseline between the frames these methods distinguish foreground information better than background information.

In some cases as shown in Figure 4.5.b, the depth map estimated by these methods are suffering from inaccurate depth values on an object's surface or the depth values of the background and foreground objects are mixed together which cause inaccurate performance in segmentation and 3D reconstruction applications.



a. A frame from the sequence

Depth map computed by Kevin Karsch *et al.* [172]

Depth map computed by Hyowon Ha *et al.* [171]

Depth map computed by the proposed method.

b. A frame from the sequence

Depth map computed by Kevin Karsch *et al.* [172]

Depth map computed by Hyowon Ha *et al* [171]

Depth map computed by the proposed method.

Figure 4.5: Comparison of the depth from small motion with state-of-the-art methods

Figure 4.6 shows the overview of the entire framework. Six important contributions have been proposed in this work as follows:

General Contributions:

1. Generally in small motions, the feature tracker can obtain more inliers due to the small difference between the frames. However the number of inliers reduces when the baseline becomes wider and as the result the generated depth map becomes inaccurate. The modified cost function in the proposed method makes it capable of processing sequence of frames with the baseline up to 400 mm while most of the methods in this field fail for the baselines wider than ~12 mm.
2. Performance for $frame \geq 2$.
3. Occlusion handling by respecting the structure of the reference frame.

Technical Contributions:

1. New data and smoothness terms are defined to recondition cost volume and cost aggregation function.
2. Proposed cost propagation is formulated as energy minimiser function for depth on each pixel point.

The proposed method can approximate non-planar surfaces, while being robust against depth outliers and occlusion. The code for this method is available online at: https://github.com/hosseinjavidnia/Depth-Small-Motion

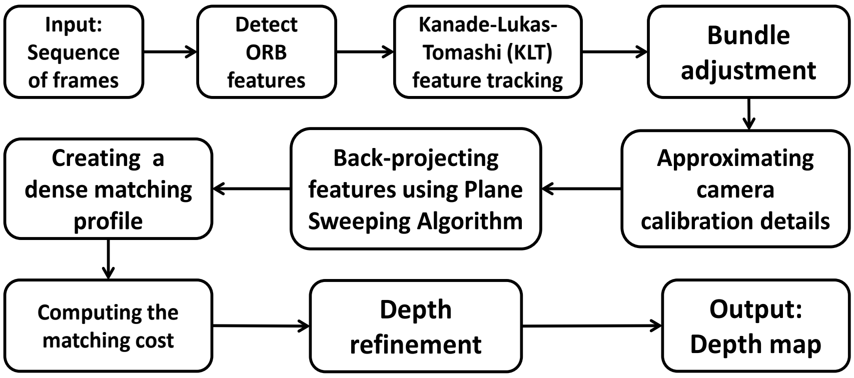

Input: Sequence of frames
Detect ORB features
Kanade-Lukas-Tomashi (KLT) feature tracking
Bundle adjustment
Approximating camera calibration details
Back-projecting features using Plane Sweeping Algorithm
Creating a dense matching profile
Computing the matching cost
Depth refinement
Output: Depth map

Figure 4.6: Overview of the proposed solution

4.4 Application of Preconditioned Alternating Direction Method of Multipliers in Depth from Focal Stack

The pipeline of estimating depth from focal stack initiates by aligning the images in the stack. The reason to apply this alignment is to compensate the misalignment of the input focal stack and minimise the effect of the motion in the sequence. The alignment is done using Epipolar homography alignment as presented in Section 3.1 in Appendix D. The value of the focus factor for each pixel at every frame of the aligned focal stack is later calculated using Modified Laplacian as described in Section 3.2 in Appendix D. The value of the focus factor for a pixel over all the frames in the stack is referred to as focus function. The initial depth map is computed by modeling the focus function using the 3-point Gaussian distribution [163]. That means the initial depth map suffers from uncertain depth values. This condition becomes severe in case of small motions of the camera. This problem is reformulated to a convex minimisation problem to be solved using Preconditioned Alternating Direction Method of Multipliers (PADMM) [164], [165]. Figure 4.7 present the sample visual results of the proposed framework, depth map captured using Lytro camera and the state of the art methods. This figure shows how the proposed framework is capable of generating a depth map with high structural accuracy in images captured under bright lighting condition. By looking at this figure, one might argue that the depth map generated by Moeller, *et al*. [173] looks more pleasant. However, the main concern is the smoothness of the depth map and respecting the structural geometry of the scene with minimum artifacts. By having these two features, the post-capturing functions such as Bokeh can be applied with a much higher quality.

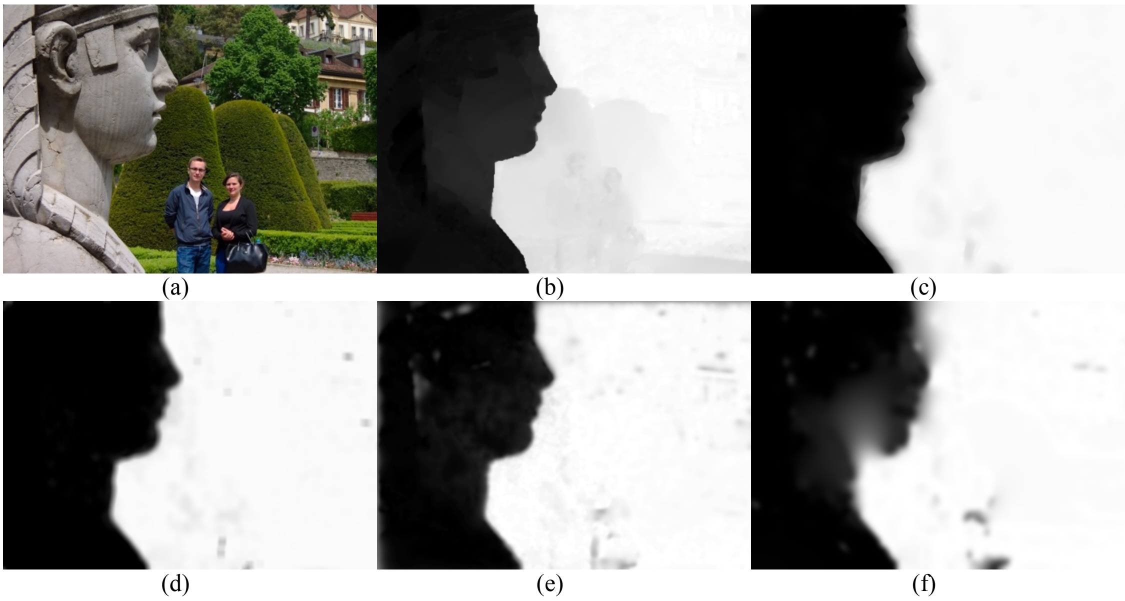

(a) (b) (c)
(d) (e) (f)

Figure 4.7: Sample depth estimation-Bright lighting condition. (a) All in focus image. (b) Ground Truth (Lytro Camera). (c) Proposed framework. (d) Moeller, *et al*. [173]. (e) Helicon Focus [174]. (f) Zerene Stacker [175]

Figure 4.8 shows the overview of the entire framework. The contributions of this research are as follows:

1- The problem of depth refinement is formulated to a convex minimisation problem by employing the vectorial ℓ^1 norm fidelity term.
2- Defining two new proximal terms to precondition ADMM.
3- An alignment framework by utilising Epipolar homography.

The whole framework is presented in detail in Section 3 of Appendix D and the code is available online at: https://github.com/hosseinjavidnia/Depth-Focal-Stack

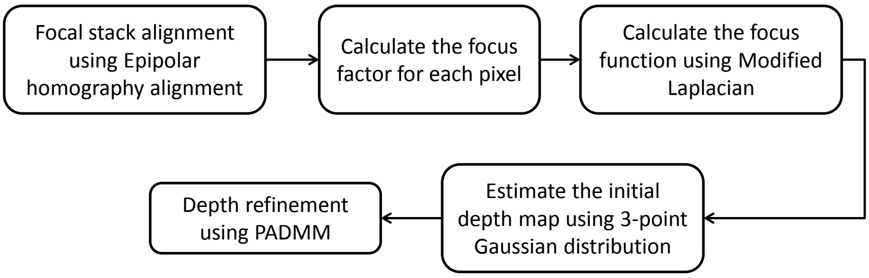

Focal stack alignment using Epipolar homography alignment
Calculate the focus factor for each pixel
Calculate the focus function using Modified Laplacian
Estimate the initial depth map using 3-point Gaussian distribution
Depth refinement using PADMM

Figure 4.8: Overview of the proposed solution

4.5 Total Variation-Based Dense Depth from Multi-Camera Array

In this framework, the initial depth is estimated by analysis of the local Epipolar Plane Image (EPI). To do this, we employed the initial part of the depth estimation algorithm of [176] as presented in Section 3.1 of the Appendix E. The implementation is based on the Cocolib light field suite [177], [178]. This method analyses the orientations of patterns on the EPIs. It takes an analytic approach to compute the orientation of Epipolar lines using the structure tensor. The estimated depth map is then refined using Total Variation (TV) minimisation based on the Fenchel-Rockafellar duality [53] as described in Section 3.2 of Appendix E. Figure 4.9 illustrates the disparity maps, ground truth error map and the median error map of the studied algorithms. Each row in these figures represents an algorithm. For each algorithm per individual image set, there are three columns illustrating the disparity maps, ground truth error map and the median error map. To generate the median error map, the median of the absolute disparity differences of all algorithms with the ground truth is computed for each pixel. Further, the absolute disparity difference of each algorithm is subtracted from the median error. The median error map gives a conceptual understanding of the parts of the image where algorithms perform below or above average performance of all algorithms. Yellow parts in this map represent the average, green above-average and red below-average performance.

The median error maps of the proposed method in Fig. 10 show how competitive the proposed method performs compared to the other algorithms while dealing with slanted planar surfaces and complex scene structure. However, there are still highly textured areas with fine patterns such as box frames in the "Boxes" image set which introduces many challenges to depth estimation algorithms.

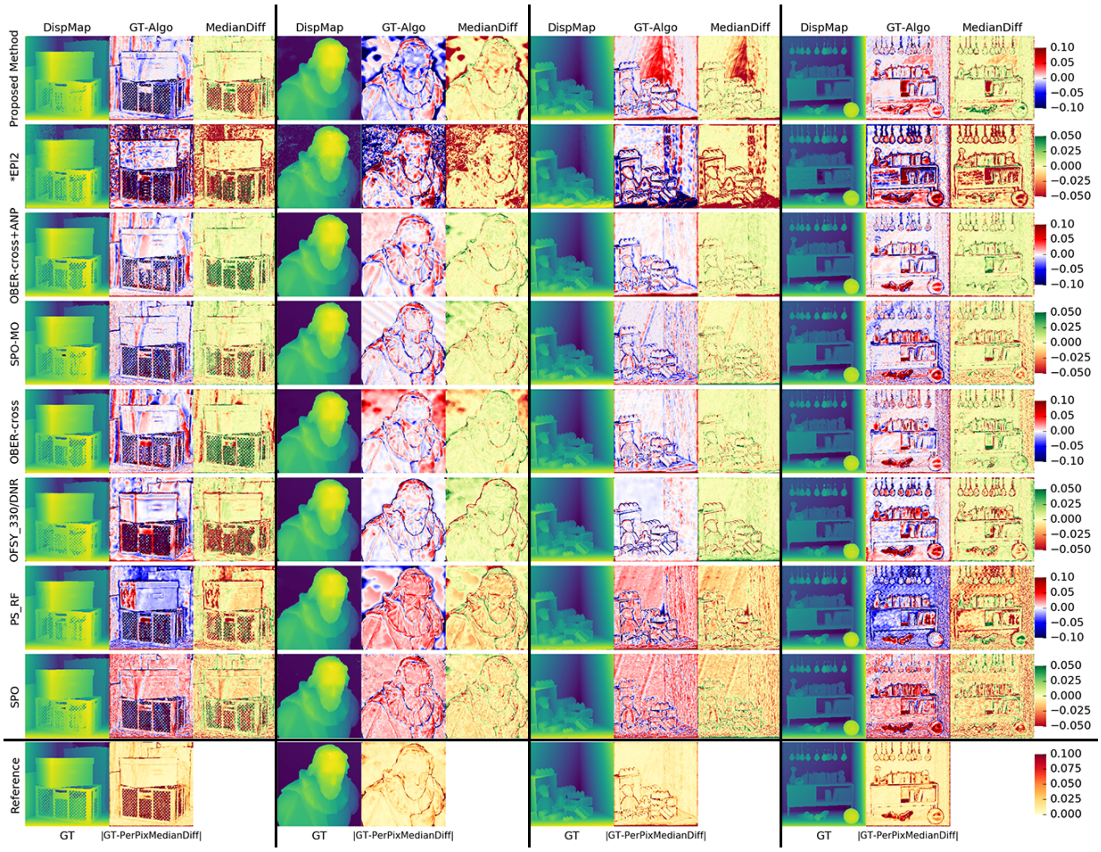


Figure 4.9: Visualization of disparity maps and their differences with ground truth. Each row represents an algorithm. The first column for each training scenes illustrates the disparity maps of the proposed method and the studied algorithms. The second column illustrates the disparity difference to the ground truth. Highly accurate parts are shown in white, too close in blue and too far in red areas. The third column illustrates how algorithms perform relative to the median algorithm performance. Yellow parts show average, green above-average and red below-average performance. The last row of the figure illustrates the ground truth disparity maps and the median absolute disparity difference to the ground truth at each individual pixel among all algorithms.

Figure 4.10 shows the overview of the entire framework. The main contributions of this work are:

1- Introducing a lightweight computational framework to estimate depth from the 4D light field on the EPI. The proposed framework is less sensitive to occlusion, noise, spatial aliasing, angular resolution and more importantly it is 2-100 times faster/more computationally efficient than the studied state of the art methods.

2- Proposing a new computational cost function derived from the Fenchel-Rockafellar duality.

The code for this method is available online at: https://github.com/hosseinjavidnia/Depth-MultiCamera

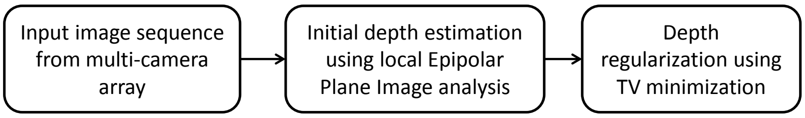

Input image sequence from multi-camera array
Initial depth estimation using local Epipolar Plane Image analysis
Depth regularization using TV minimization

Figure 4.10: Overview of the proposed solution

Chapter 5.
Discussion and Future Work

The need for computer vision systems that can perceive and understand 3D scenes is growing rapidly, motivated by the need for machines to interact with the 3D world in real-time. In this dissertation, the methods of estimating depth and 3D information from consumer cameras such as smartphones were investigated. The goal was to propose methods and frameworks which can provide depth information on lightweight computational devices such as smartphones.

Having access to such information enables consumers to interact with the digital photographs by applying post-capture modification such as objected segmentation, 3D holography and refocusing etc.

The existing methods that perform accurately in this task are computationally intensive; they require an additional sensor such as dot projector or NIR, which is usually expensive, and they can be interfered with different factors such as light sources. The performance of these sensors significantly decreases while being exposed to very bright environment. For these reasons, the existing methods are not efficient enough for lightweight devices with limited computational power.

As the use of stereo cameras became popular in the smartphone industry since 15 years ago, this research is initiated by studying stereo matching methods. Instead of proposing a new stereo matching technique, a generic post-processing algorithm is presented which can be employed along with any depth estimation algorithm to increase its structural accuracy. The proposed post-processing method takes advantage of the mutual information between the reference RGB image and the disparity image. The aim of this process is to respect the structure of the original scene, including object boundaries and corners in the disparity map. The method was implemented along with a state of the art algorithm and was evaluated based on the Middlebury stereo benchmark. The preliminary results indicated that the state of the art stereo matching algorithm with our post-processing technique can be ranked among the top 8 methods in the Middlebury benchmark. Considering the performance of this method in providing highly accurate disparity maps, it still requires some optimisation to perform faster in real-time mode on high-resolution images.

To reduce the computational time of the depth estimation using stereo matching methods, a CNN architecture is proposed which is known as Semi Parallel Deep Neural Network (SPDNN). The goal of this network was to learn to provide depth information from monocular images. Four models were trained using this architecture based on the KITTI benchmark while the target for two of the models was generated using the post-processing algorithm which helped the network to generate more precise models. The semantic segmentation of the input frame is also added to the models, which resulted in preserving the structural information in the output depth map. Our extensive evaluation indicated that the proposed CNN can estimate depth maps from monocular images in ~1.23 sec/MP with very close accuracy to the stereo matching method.

Despite the fast performance of the CNN models, they require input images to be down-sampled for memory purposes. These models are often too large for consumer devices, computational power. However, it is still possible to use a single camera to estimate depth information by utilising its motion. To do this, the present research analysed random and small motions captured by a smartphone camera for depth sensing applications. The term small motion refers to the pre-initialization of the camera before capturing the image or consumers' natural handshake. A framework is proposed to generate semi-dense 3D point cloud and dense depth map from the motions with the baseline less than ~8 mm. A modified bundle adjustment with a new cost function was used for optimisation purposes. The evaluation showed that the proposed framework has a superior performance to the state of the art methods. The second part of the evaluation indicated that the proposed method can be applied on the motions with the baseline up to ~400 mm which makes it comparable with the stereo matching methods.

In another attempt, quite similar to the concept of small motions, we took advantage of the optical properties of the cameras by capturing a short sequence of frames with varying focal points which are known as focal stacks. Initially, the focal stacks were aligned using Epipolar homography alignment. Afterwards, Modified Laplacian was used to calculate the focus function, followed by a 3-point Gaussian distribution method to estimate the initial depth map. Later, Preconditioned Alternating Direction Method of Multipliers (PADMM) was modified by adding two new proximity terms for refinement purposes. The performance of the proposed method was compared against a state of the art algorithm and two commercial software packages. Both quantitative and qualitative evaluations noted the superior performance of the proposed method.

The concept of small motions can be defined in multi-view images where the baseline between different viewpoints is relatively small. This is similar to the newly introduced cameras for smartphones known as a multi-camera array. These camera arrays contain a set of cameras arranged in a grid with a very small baseline. Using Epipolar geometry, it is possible to estimate depth information from the images captured by these cameras. To do so, a method is proposed to estimate depth from the multi-camera array by analysing Epipolar Plane Images (EPI). The image sequences are initially aligned using the Epipolar homography alignment. The initial depth map is estimated using local EPI analysis which is later regularized using Total Variation (TV) minimisation. The proposed minimisation problem takes advantage of the Fenchel-Rockafellar duality to optimise the cost function. The evaluation is done based on HCI, the Heidelberg 4D Light Field Benchmark. The results demonstrate the competitive performance of the proposed framework among the top state of the art methods in terms of accuracy of depth estimation. The fast convergence of the proposed cost function and the method's fast computational time make it a potential method for consumer electronics applications and devices with the aid of parallel technology and GPUs.

In general, the accuracy of the depth and 3D data is not identical to the real world measurements. That strongly depends on the geometrical design of the imaging system, geometrical features of the real world scene, the presence of the depth cues, occlusion, shadow, lighting condition etc. Because of all the limitations, the existing depth estimation methods fail in generating real-world depth and 3D information. Most of the algorithms consist of multiple elements and terms which make it difficult to establish one best algorithm that outperforms in all aspects and metrics.

Similarly, choosing a perfect solution for an application of depth is another challenge. If an application requires a third dimension, then a suitable technology has to be chosen according to the requirements of the application and the limitation of the technology. There are main criteria to be assessed while choosing a suitable technology. These conditions include:

1- How much accuracy does the depth application require?
2- What are the surface conditions of the objects that the technology has to deal with?
3- What are the required scanning range and speed?
4- Is real-time performance essential?
5- Does the depth technology have a complex setup and maintenance?
6- How expensive is the depth technology?
7- What is the cost of 3D reconstruction based on the depth technology?
8- What are the environmental challenges for the depth technology?

Considering all these assessments, there is still no single perfect solution which can satisfy all the requirements for a depth application. All the requirements have to be prioritised based on the use of the application and then the right technology can be chosen. Certainly, the disadvantages of these technologies can be compensated by employing a complementary method with an additional cost.

Another main challenge in this field is quality assessment and lack of ground truth data. Having access to real-world image sequences and the corresponding ground truth data enables us to validate all the methodologies. The existing datasets and benchmarks are mostly synthetic which are not representative of the real world scene. For the real world data, the ground truth information is generated using a sparse LIDAR scanner which requires an extra computational process for densification purposes. The main metrics introduced in the state of the art to evaluate depth estimation methods are pixel-wise. An evaluation based on these metrics cannot clarify the performance of an algorithm in terms of structural accuracy.

One of the most significant future works will be optimising all these methods for implementation on low power computational platforms. There is a good potential for these methods, especially the algorithms presented in Section 4.4 and Section 4.5 to be implemented on consumer devices. Another important aspect to note is that some of the proposed algorithms perform on high-resolution images without any down-sampling which is already one step ahead of the state of the art. As already mentioned, this field of research suffers from a lack of real-world data and proper evaluation metrics. This can be another potential gap to be focused on. Proposing a unified assessment system and global metrics can definitely improve the way that the existing methods are evaluated.

Nowadays, the promises of 3D capabilities in Augmented Reality (AR) are opening up many opportunities. Engineers can bring up 3D models of parts and rotate them in space, getting a view that is more encompassing than what is possible on a computer monitor. Retail customers could see themselves wearing clothing or accessories, turning to view all angles in a mirror, without physically donning the objects. Field service personnel could see how an equipment casing opened and then have the virtual version guide them through a series of repair actions.

Advances in 3D perception are integrating into AR devices over time. The 3D modeling capabilities can combine models generated ahead of time with real-time modeling and mapping. The infrastructure for indoor positioning, both in coverage and precision, is yet to develop. But AR devices will provide operating-system-level support to surface the location and orientation information and make that available to applications and developers. Already, 3D displays are very good and are expected to get even better. However, the evolution of AR/VR without artificially intelligent systems in place would remain incredibly basic. Nowadays, Deep learning is playing a key role in developing AR/VR technologies and will continue to improve their levels of functionality. When all of these capabilities come together in a seamless manner, the technology will indeed merge the 2D digital world with the 3D physical world of work and play.

References

[1] "Innovative products with DLP® technology," *Texas Instruments*. [Online]. Available: http://www.ti.com/dlp-technology/markets.html. [Accessed: 22-May-2018].

[2] "Stereoscope," *Mathematics, North Carolina School of Science and*. [Online]. Available: http://courses.ncssm.edu/gallery/collections/toys/html/exhibit01.htm.

[3] P. Rotter, "Why Did the 3D Revolution Fail?: The Present and Future of Stereoscopy [Commentary]," *IEEE Technol. Soc. Mag.*, vol. 36, no. 1, pp. 81–85, 2017.

[4] D. Brewster, *The Stereoscope; Its History, Theory and Construction, with Its Application to the Fine and Useful Arts and to Education, Etc*. John Murray, 1856.

[5] "Louis Jules Duboscq," *FutureLearn*. [Online]. Available: https://www.futurelearn.com/courses/stereoscopy/0/steps/16696.

[6] I. P. Howard and B. J. Rogers, *Binocular vision and stereopsis*. Oxford University Press, USA, 1995.

[7] O. W. Holmes, "The stereoscope and the stereograph," *Atl. Mon.*, vol. 3, no. 20, 1859.

[8] G. E. Hamilton, *Oliver Wendell Holmes: His pioneer stereoscope and the later industry*, no. 448. Newcomen Society in North America, 1949.

[9] E. Strain, "The History and Mechanics of Stereoscopy." [Online]. Available: http://homes.lmc.gatech.edu/~strain/Stereoscope/history.html.

[10] E. L. and W. J. YOUMANS., *Popular Science Monthly*, Vol. 21. Bonnier Corporation, 1882.

[11] R. Zone, *Stereoscopic Cinema and the Origins of 3-D Film, 1838-1952*. University Press of Kentucky, 2014.

[12] C. Sellers, "Coleman sellers." Google Patents, 05-Feb-1861.

[13] A. Fernando, S. T. Worrall, and E. Ekmekcioðlu, *3DTV: processing and transmission of 3D video signals*. John Wiley & Sons, 2013.

[14] "3-D (Three Dimensional) Movie Craze (1950's)," *MORTALJOURNEY*, 2010. [Online]. Available: http://www.mortaljourney.com/2010/11/1950-trends/3d-movies.

[15] D. Sung, "The history of 3D cinema," *Pocket-lint*, 2009. [Online]. Available: https://www.pocket-lint.com/tv/news/98279-the-history-of-3d-cinema.

[16] "Microsoft HoloLens," *Microsoft*. [Online]. Available: https://www.microsoft.com/en-us/hololens. [Accessed: 22-May-2018].

[17] "Hologram stability," *Microsoft*. [Online]. Available: https://docs.microsoft.com/en-us/windows/mixed-reality/hologram-stability. [Accessed: 22-May-2018].

[18] F. Yu and D. Gallup, "3D Reconstruction from Accidental Motion," in *2014 IEEE*

Conference on Computer Vision and Pattern Recognition, 2014, pp. 3986–3993.

[19] S. Im, H. Ha, G. Choe, H. G. Jeon, K. Joo, and I. S. Kweon, "High Quality Structure from Small Motion for Rolling Shutter Cameras," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 837–845.

[20] M. Strecke, A. Alperovich, and B. Goldluecke, "Accurate Depth and Normal Maps from Occlusion-Aware Focal Stack Symmetry," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2529–2537.

[21] M. Subbarao and G. Surya, "Depth from defocus: A spatial domain approach," *Int. J. Comput. Vis.*, vol. 13, no. 3, pp. 271–294, Dec. 1994.

[22] C. Hazirbas, L. Leal-Taixé, and D. Cremers, "Deep Depth From Focus," *arXiv Prepr. arXiv1704.01085*, 2017.

[23] N. B. Monteiro, J. P. Barreto, and J. Gaspar, "Dense Lightfield Disparity Estimation Using Total Variation Regularization," in *Image Analysis and Recognition: 13th International Conference, ICIAR 2016, in Memory of Mohamed Kamel, Póvoa de Varzim, Portugal, July 13-15, 2016, Proceedings*, A. Campilho and F. Karray, Eds. Cham: Springer International Publishing, 2016, pp. 462–469.

[24] F. Tombari, S. Mattoccia, and L. Di Stefano, "Stereo for robots: Quantitative evaluation of efficient and low-memory dense stereo algorithms," in *2010 11th International Conference on Control Automation Robotics & Vision*, 2010, pp. 1231–1238.

[25] S. Yang, G. Huang, Z. Zhao, and N. Wang, "Extraction of Topographic Map Elements with SAR Stereoscopic Measurement," in *2011 International Symposium on Image and Data Fusion*, 2011, pp. 1–4.

[26] R. Muñoz-Salinas, E. Aguirre, and M. García-Silvente, "People detection and tracking using stereo vision and color," *Image Vis. Comput.*, vol. 25, no. 6, pp. 995–1007, 2007.

[27] P. Haigron *et al.*, "Depth-map-based scene analysis for active navigation in virtual angioscopy," *IEEE Trans. Med. Imaging*, vol. 23, no. 11, pp. 1380–1390, 2004.

[28] G. Yahav, G. J. Iddan, and D. Mandelboum, "3D Imaging Camera for Gaming Application," in *2007 Digest of Technical Papers International Conference on Consumer Electronics*, 2007, pp. 1–2.

[29] M. Grosse, J. Buehl, H. Babovsky, A. Kiessling, and R. Kowarschik, "3D shape measurement of macroscopic objects in digital off-axis holography using structured illumination," *Opt. Lett.*, vol. 35, no. 8, pp. 1233–1235, 2010.

[30] P. Kauff *et al.*, "Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability," *Signal Process. Image Commun.*, vol. 22, no. 2, pp. 217–234, 2007.

[31] P. Merkle *et al.*, “The Effect of Depth Compression on Multiview Rendering Quality,” in *2008 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, 2008, pp. 245–248.

[32] S. R. Malireddi *et al.*, “HandSeg: A Dataset for Hand Segmentation from Depth Images,” *arXiv Prepr. arXiv1711.05944*, 2017.

[33] New York Film Academy, “Bokeh Photography – Capturing The Bokeh Effect,” 2014. [Online]. Available: https://www.nyfa.edu/student-resources/capturing-the-bokeh-effect/. [Accessed: 17-May-2018].

[34] S. R. C. Boga, B. Kansagara, and R. Kannan, “Integration of augmented reality and virtual reality in building information modeling: The next frontier in civil engineering education,” in *Virtual and augmented reality: Concepts, methodologies, tools, and applications*, IGI Global, 2018, pp. 1037–1066.

[35] S. S. Abed, “Opportunities and Challenges of Augmented Reality Shopping in Emerging Markets,” in *Emerging Markets from a Multidisciplinary Perspective: Challenges, Opportunities and Research Agenda*, Y. K. Dwivedi, N. P. Rana, E. L. Slade, M. A. Shareef, M. Clement, A. C. Simintiras, and B. Lal, Eds. Cham: Springer International Publishing, 2018, pp. 107–117.

[36] “Obsess.” [Online]. Available: http://www.obsessar.com/. [Accessed: 17-May-2018].

[37] T. Taniai, Y. Matsushita, Y. Sato, and T. Naemura, “Continuous 3D Label Stereo Matching using Local Expansion Moves,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PP, no. 99, p. 1, 2017.

[38] L. Li, X. Yu, S. Zhang, X. Zhao, and L. Zhang, “3D cost aggregation with multiple minimum spanning trees for stereo matching,” *Appl. Opt.*, vol. 56, no. 12, pp. 3411–3420, Apr. 2017.

[39] A. Geiger, M. Roser, and R. Urtasun, “Efficient Large-Scale Stereo Matching,” in *Computer Vision -- ACCV 2010: 10th Asian Conference on Computer Vision, Queenstown, New Zealand, November 8-12, 2010, Revised Selected Papers, Part I*, R. Kimmel, R. Klette, and A. Sugimoto, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 25–38.

[40] L. Keselman, J. I. Woodfill, A. Grunnet-Jepsen, and A. Bhowmik, “Intel(R) RealSense(TM) Stereoscopic Depth Cameras,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 1267–1276.

[41] N. Schneider, “Simple Network in Network CNN with Mask Concatenated [NiN+Mask CNN],” 2017. [Online]. Available: http://www.cvlibs.net/datasets/kitti/eval_depth_detail.php?benchmark=depth_completion&result=c95cb941ad57e21705c8c0ceaa88d394daf0428e.

[42] N. Schneider, “Simple Network in Network CNN [NiN CNN],” 2017. [Online]. Available: http://www.cvlibs.net/datasets/kitti/eval_depth_detail.php?benchmark=depth_completion&result=ca9247f9e582228c504567c7b91e90ab41123406.

[43] N. Schneider, “Semantically Guided Depth Upsampling [mono] [SGDU],” 2017. [Online]. Available: http://www.cvlibs.net/datasets/kitti/eval_depth_detail.php?benchmark=depth_completion&result=776969e2b4a512f59a35934dd71aadfa4fb06e1b.

[44] S. Izadi *et al.*, “KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera,” in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, 2011, pp. 559–568.

[45] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger, “Real-time 3D Reconstruction at Scale Using Voxel Hashing,” *ACM Trans. Graph.*, vol. 32, no. 6, p. 169:1--169:11, Nov. 2013.

[46] T. Whelan, H. Johannsson, M. Kaess, J. J. Leonard, and J. McDonald, “Robust real-time visual odometry for dense RGB-D mapping,” in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 5724–5731.

[47] E. Bylow, J. Sturm, C. Kerl, F. Kahl, and D. Cremers, “Real-Time Camera Tracking and 3D Reconstruction Using Signed Distance Functions.,” in *Robotics: Science and Systems*, 2013, vol. 2.

[48] J. Sturm, E. Bylow, F. Kahl, and D. Cremers, “CopyMe3D: Scanning and Printing Persons in 3D,” in *Pattern Recognition: 35th German Conference, GCPR 2013, Saarbrücken, Germany, September 3-6, 2013. Proceedings*, J. Weickert, M. Hein, and B. Schiele, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 405–414.

[49] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, “3-D Mapping With an RGB-D Camera,” *IEEE Trans. Robot.*, vol. 30, no. 1, pp. 177–187, 2014.

[50] S. y. Kim, M. Kim, and Y. s. Ho, “Depth image filter for mixed and noisy pixel removal in RGB-D camera systems,” *IEEE Trans. Consum. Electron.*, vol. 59, no. 3, pp. 681–689, 2013.

[51] S. Lee, J. H. Lee, J. Lim, and I. H. Suh, “Robust stereo matching using adaptive random walk with restart algorithm,” *Image Vis. Comput.*, vol. 37, no. Supplement C, pp. 1–11, 2015.

[52] K. Venkataraman *et al.*, “Picam: An ultra-thin high performance monolithic camera array,” *ACM Trans. Graph.*, vol. 32, no. 6, p. 166, 2013.

[53] R. T. Rockafellar, *Convex Analysis*. Princeton University Press, 1970.

[54] “Virtual reality for everyone,” *Google*. [Online]. Available: https://vr.google.com/.

[Accessed: 23-May-2018].

[55] "Augmented Reality for iOS," *Apple*. [Online]. Available: https://www.apple.com/ios/augmented-reality/. [Accessed: 23-May-2018].

[56] R. Patel, "Google Lens: real-time answers to questions about the world around you," *Google Blog*, 2018. [Online]. Available: https://www.blog.google/products/google-vr/google-lens-real-time-answers-questions-about-world-around-you/. [Accessed: 23-May-2018].

[57] "ARKit," *Apple*. [Online]. Available: https://developer.apple.com/arkit/. [Accessed: 23-May-2018].

[58] "ARCore Overview," *Google*. [Online]. Available: https://developers.google.com/ar/discover/. [Accessed: 23-May-2018].

[59] S. Greenwald *et al.*, "Technology and applications for collaborative learning in virtual reality," 2017.

[60] J. Butterworth, A. Davidson, S. Hench, and M. T. Olano, "3DM: A Three Dimensional Modeler Using a Head-mounted Display," in *Proceedings of the 1992 Symposium on Interactive 3D Graphics*, 1992, pp. 135–138.

[61] J. M. Knapp and J. M. Loomis, "Limited Field of View of Head-mounted Displays is Not the Cause of Distance Underestimation in Virtual Environments," *Presence: Teleoper. Virtual Environ.*, vol. 13, no. 5, pp. 572–577, Oct. 2004.

[62] R. Abele, "High-precision maps for self-driving cars," *Mercedes-Benz*, 2016. [Online]. Available: https://www.mercedes-benz.com/en/mercedes-benz/next/connectivity/high-precision-maps-for-self-driving-cars/.

[63] Google, "Tango: It's your turn. Build the future.," 2017. [Online]. Available: https://developers.google.com/tango/.

[64] Apple, "iPhone X," 2017. [Online]. Available: https://www.apple.com/ie/iphone-x/specs/.

[65] Qualcomm, "Qualcomm First to Announce Depth-Sensing Camera Technology Designed for Android Ecosystem," 2017. [Online]. Available: https://www.qualcomm.com/news/releases/2017/08/15/qualcomm-first-announce-depth-sensing-camera-technology-designed-android.

[66] S. Chen, A. Pande, and P. Mohapatra, "Sensor-assisted Facial Recognition: An Enhanced Biometric Authentication System for Smartphones," in *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, 2014, pp. 109–122.

[67] Apple, "Face ID," 2017. [Online]. Available: https://www.apple.com/ie/iphone-x/#face-id.

[68] F. da S. Guizi and C. S. Kurashima, "Real-time people detection and tracking using

3D depth estimation," in *2016 IEEE International Symposium on Consumer Electronics (ISCE)*, 2016, pp. 39–40.

[69] T. K. S. Cheung and K. T. Woo, "Human tracking in crowded environment with stereo cameras," in *2011 17th International Conference on Digital Signal Processing (DSP)*, 2011, pp. 1–6.

[70] K. Bakhtiyari, N. Beckmann, and J. Ziegler, "Contactless heart rate variability measurement by IR and 3D depth sensors with respiratory sinus arrhythmia," *Procedia Comput. Sci.*, vol. 109, no. Supplement C, pp. 498–505, 2017.

[71] T. Tan *et al.*, "Segmentation of malignant lesions in 3D breast ultrasound using a depth-dependent model," *Med. Phys.*, vol. 43, no. 7, pp. 4074–4084, 2016.

[72] M. Field, D. Clarke, S. Strup, and W. B. Seales, "Stereo endoscopy as a 3-D measurement tool," in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2009, pp. 5748–5751.

[73] C. Zhang, L. Wang, and R. Yang, "Semantic Segmentation of Urban Scenes Using Dense Depth Maps," in *Computer Vision -- ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV*, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 708–721.

[74] Z. Li *et al.*, "A 1920 × 1080 30-frames/s 2.3 TOPS/W Stereo-Depth Processor for Energy-Efficient Autonomous Navigation of Micro Aerial Vehicles," *IEEE J. Solid-State Circuits*, vol. 53, no. 1, pp. 76–90, 2018.

[75] M. Faessler, F. Fontana, C. Forster, E. Mueggler, M. Pizzoli, and D. Scaramuzza, "Autonomous, Vision-based Flight and Live Dense 3D Mapping with a Quadrotor Micro Aerial Vehicle," *J. F. Robot.*, vol. 33, no. 4, pp. 431–450, 2016.

[76] S. Zollmann, C. Hoppe, T. Langlotz, and G. Reitmayr, "FlyAR: Augmented Reality Supported Micro Aerial Vehicle Navigation," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 4, pp. 560–568, 2014.

[77] C. Lee, H. Song, B. Choi, and Y. S. Ho, "3D scene capturing using stereoscopic cameras and a time-of-flight camera," *IEEE Trans. Consum. Electron.*, vol. 57, no. 3, pp. 1370–1376, 2011.

[78] Microsoft, "Kinect for Xbox One," 2017. [Online]. Available: https://www.xbox.com/en-US/xbox-one/accessories/kinect.

[79] A. D. Payne, A. P. P. Jongenelen, A. A. Dorrington, M. J. Cree, and D. A. Carnegie, "Multiple frequency range imaging to remove measurement ambiguity," *Optical 3-D Measurement Techniques*. Conference held at Vienna, Austria, 2009.

[80] A. Kadambi, J. Schiel, and R. Raskar, "Macroscopic Interferometry: Rethinking Depth Estimation with Frequency-Domain Time-of-Flight," in *2016 IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 893–902.

[81] S. Schuon, C. Theobalt, J. Davis, and S. Thrun, "High-quality scanning using time-of-flight depth superresolution," in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2008, pp. 1–7.

[82] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super resolution," *IEEE Trans. Image Process.*, vol. 13, no. 10, pp. 1327–1344, 2004.

[83] J. Zhu, L. Wang, R. Yang, and J. Davis, "Fusion of time-of-flight depth and stereo for high accuracy depth maps," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[84] G. Marin, P. Zanuttigh, and S. Mattoccia, "Reliable Fusion of ToF and Stereo Depth Driven by Confidence Measures," in *Computer Vision -- ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11--14, 2016, Proceedings, Part VII*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 386–401.

[85] G. Agresti, L. Minto, G. Marin, and P. Zanuttigh, "Deep Learning for Confidence Information in Stereo and ToF Data Fusion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 697–705.

[86] J. Noraky and V. Sze, "LOW POWER DEPTH ESTIMATION FOR TIME-OF-FLIGHT IMAGING."

[87] K. L. Boyer and A. C. Kak, "Color-Encoded Structured Light for Rapid Active Ranging," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 1, pp. 14–28, 1987.

[88] C.-S. Chen, Y.-P. Hung, C.-C. Chiang, and J.-L. Wu, "Range data acquisition using color structured lighting and stereo vision," *Image Vis. Comput.*, vol. 15, no. 6, pp. 445–456, 1997.

[89] D. Bergmann, "New approach for automatic surface reconstruction with coded light," in *PROCEEDINGS-SPIE THE INTERNATIONAL SOCIETY FOR OPTICAL ENGINEERING*, 1995, p. 2.

[90] B. Carrihill and R. Hummel, "Experiments with the intensity ratio depth sensor," *Comput. Vision, Graph. Image Process.*, vol. 32, no. 3, pp. 337–358, 1985.

[91] T. Jia, Z. Zhou, and H. Gao, "Depth measurement based on infrared coded structured light," *J. Sensors*, vol. 2014, 2014.

[92] H. Luo, B. Gao, J. Xu, and K. Chen, "An approach for structured light system calibration," in *2013 IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems*, 2013, pp. 428–433.

[93] H. Cui, N. Dai, T. Yuan, X. Cheng, and W. Liao, "Calibration Algorithm for

Structured Light 3D Vision Measuring System," in *2008 Congress on Image and Signal Processing*, 2008, vol. 2, pp. 324–328.

[94] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, 2003, vol. 1, p. I-195-I-202 vol.1.

[95] Y. Zhang, Z. Xiong, P. Cong, and F. Wu, "Robust depth sensing with adaptive structured light illumination," *J. Vis. Commun. Image Represent.*, vol. 25, no. 4, pp. 649–658, 2014.

[96] S.-Y. Chan, H.-F. Shih, and J.-S. Chen, "Depth measurement using structured light and spatial frequency," *Appl. Opt.*, vol. 55, no. 19, pp. 5069–5075, Jul. 2016.

[97] S. R. Fanello *et al.*, "HyperDepth: Learning Depth from Structured Light without Matching," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5441–5450.

[98] M. Gupta, Q. Yin, and S. K. Nayar, "Structured Light in Sunlight," in *2013 IEEE International Conference on Computer Vision*, 2013, pp. 545–552.

[99] J. Wang, C. Zhang, W. Zhu, Z. Zhang, Z. Xiong, and P. A. Chou, "3D scene reconstruction by multiple structured-light based commodity depth cameras," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 5429–5432.

[100] F. Tsalakanidou, F. Forster, S. Malassiotis, and M. G. Strintzis, "Real-time acquisition of depth and color images using structured light and its application to 3D face recognition," *Real-Time Imaging*, vol. 11, no. 5, pp. 358–369, 2005.

[101] S. Paulus, J. Behmann, A.-K. Mahlein, L. Plümer, and H. Kuhlmann, "Low-Cost 3D Systems: Suitable Tools for Plant Phenotyping," *Sensors*, vol. 14, no. 2, pp. 3001–3018, 2014.

[102] T. T. Nguyen, D. C. Slaughter, N. Max, J. N. Maloof, and N. Sinha, "Structured Light-Based 3D Reconstruction System for Plants," *Sensors*, vol. 15, no. 8, pp. 18587–18612, 2015.

[103] S. G. Narasimhan and S. K. Nayar, "Structured light methods for underwater imaging: light stripe scanning and photometric stereo," in *Proceedings of OCEANS 2005 MTS/IEEE*, 2005, p. 2610–2617 Vol. 3.

[104] "How does LiDAR work?," *LiDAR UK*, 2018. [Online]. Available: http://www.lidar-uk.com/how-lidar-works/.

[105] C. Premebida, L. Garrote, A. Asvadi, A. P. Ribeiro, and U. Nunes, "High-resolution LIDAR-based depth mapping using bilateral filter," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 2469–2474.

[106] L. Chen, Y. He, J. Chen, Q. Li, and Q. Zou, "Transforming a 3-D LiDAR Point

Cloud Into a 2-D Dense Depth Map Through a Parameter Self-Adaptive Framework," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 1, pp. 165–176, 2017.

[107] Y. He, L. Chen, J. Chen, and M. Li, "A novel way to organize 3D LiDAR point cloud as 2D depth map height map and surface normal map," in *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2015, pp. 1383–1388.

[108] W. Maddern and P. Newman, "Real-time probabilistic fusion of sparse 3D LIDAR and dense stereo," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 2181–2188.

[109] L. Ding and G. Sharma, "Fusing structure from motion and lidar for dense accurate depth map estimation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 1283–1287.

[110] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.

[111] C. Premebida, J. Carreira, J. Batista, and U. Nunes, "Pedestrian detection combining RGB and dense LIDAR data," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 4112–4117.

[112] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, 2010.

[113] J. Tan, J. Li, X. An, and H. He, "Robust Curb Detection with Fusion of 3D-Lidar and Camera Data," *Sensors (Basel).*, vol. 14, no. 5, pp. 9046–9073, May 2014.

[114] J. Dolson, J. Baek, C. Plagemann, and S. Thrun, "Upsampling range data in dynamic environments," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1141–1148.

[115] J. Zhang and S. Singh, "LOAM: Lidar Odometry and Mapping in Real-time.," in *Robotics: Science and Systems*, 2014, vol. 2.

[116] J. Zhang and S. Singh, "Visual-lidar odometry and mapping: Low-drift, robust, and fast," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, 2015, pp. 2174–2181.

[117] J. Zhang and S. Singh, "Low-drift and Real-time Lidar Odometry and Mapping," *Auton. Robot.*, vol. 41, no. 2, pp. 401–416, Feb. 2017.

[118] K. Lenac, A. Kitanov, R. Cupec, and I. Petrović, "Fast planar surface 3D SLAM using LIDAR," *Rob. Auton. Syst.*, vol. 92, no. Supplement C, pp. 197–220, 2017.

[119] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," in *2016 IEEE International Conference on Robotics and Automation*

(ICRA), 2016, pp. 1271–1278.

[120] C. Park, P. Moghadam, S. Kim, A. Elfes, C. Fookes, and S. Sridharan, “Elastic LiDAR Fusion: Dense Map-Centric Continuous-Time SLAM,” *arXiv Prepr. arXiv1711.01691*, 2017.

[121] K. Cho, S. Baeg, and S. Park, “Natural Terrain Detection and SLAM Using LIDAR for UGV,” in *Intelligent Autonomous Systems 12: Volume 1 Proceedings of the 12th International Conference IAS-12, held June 26-29, 2012, Jeju Island, Korea*, S. Lee, H. Cho, K.-J. Yoon, and J. Lee, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 793–805.

[122] J. Tang *et al.*, “LiDAR Scan Matching Aided Inertial Navigation System in GNSS-Denied Environments,” *Sensors (Basel).*, vol. 15, no. 7, pp. 16710–16728, Jul. 2015.

[123] F. Moosmann and C. Stiller, “Velodyne SLAM,” in *2011 IEEE Intelligent Vehicles Symposium (IV)*, 2011, pp. 393–398.

[124] D. Scharstein and R. Szeliski, “A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms,” *Int. J. Comput. Vis.*, vol. 47, no. 1, pp. 7–42, Apr. 2002.

[125] R. Zabih and J. Woodfill, “Non-parametric local transforms for computing visual correspondence,” in *Computer Vision --- ECCV ’94: Third European Conference on Computer Vision Stockholm, Sweden, May 2--6 1994 Proceedings, Volume II*, J.-O. Eklundh, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 151–158.

[126] A. Motten, L. Claesen, and Y. Pan, “Binary confidence evaluation for a stereo vision based depth field processor SoC,” in *The First Asian Conference on Pattern Recognition*, 2011, pp. 456–460.

[127] X. Hu and P. Mordohai, “Evaluation of stereo confidence indoors and outdoors,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1466–1473.

[128] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, “MonoSLAM: Real-Time Single Camera SLAM,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, 2007.

[129] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, “ORB-SLAM: A Versatile and Accurate Monocular SLAM System,” *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, 2015.

[130] G. Klein and D. Murray, “Parallel Tracking and Mapping for Small AR Workspaces,” in *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2007, pp. 225–234.

[131] J. Engel, V. Koltun, and D. Cremers, “Direct Sparse Odometry,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, 2018.

[132] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, “DTAM: Dense tracking and mapping in real-time,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011, pp. 2320–2327.

[133] J. Engel, T. Schöps, and D. Cremers, “LSD-SLAM: Large-Scale Direct Monocular SLAM,” in *Computer Vision -- ECCV 2014*, 2014, pp. 834–849.

[134] P. Fua, “A parallel stereo algorithm that produces dense depth maps and preserves image features,” *Mach. Vis. Appl.*, vol. 6, no. 1, pp. 35–49, 1993.

[135] T. Kanade, A. Yoshida, K. Oda, H. Kano, and M. Tanaka, “A stereo machine for video-rate dense depth mapping and its new applications,” in *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1996, pp. 196–202.

[136] R. Nevatia, “Depth measurement by motion stereo,” *Comput. Graph. Image Process.*, vol. 5, no. 2, pp. 203–214, 1976.

[137] D. Scharstein and R. Szeliski, “Middlebury Stereo Evaluation - Version 3,” 2015. [Online]. Available: http://vision.middlebury.edu/stereo/eval3/.

[138] S. Drouyer, S. Beucher, M. Bilodeau, M. Moreaud, and L. Sorbier, “Sparse Stereo Disparity Map Densification Using Hierarchical Image Segmentation,” in *Mathematical Morphology and Its Applications to Signal and Image Processing: 13th International Symposium, ISMM 2017, Fontainebleau, France, May 15--17, 2017, Proceedings*, J. Angulo, S. Velasco-Forero, and F. Meyer, Eds. Cham: Springer International Publishing, 2017, pp. 172–184.

[139] L. Li, S. Zhang, X. Yu, and L. Zhang, “PMSC: PatchMatch-Based Superpixel Cut for Accurate Stereo Matching,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. PP, no. 99, p. 1, 2017.

[140] C. Zhang, Z. Li, Y. Cheng, R. Cai, H. Chao, and Y. Rui, “MeshStereo: A Global Stereo Model with Mesh Alignment Regularization for View Interpolation,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2057–2065.

[141] J. Wei, B. Resch, and H. P. A. Lensch, “Multi-View Depth Map Estimation With Cross-View Consistency.,” in *BMVC*, 2014.

[142] S. Tulyakov, A. Ivanov, and F. Fleuret, “Weakly Supervised Learning of Deep Metrics for Stereo Reconstruction,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1348–1357.

[143] J. Žbontar and Y. LeCun, “Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches,” *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2287–2318, Jan. 2016.

[144] C. Zhou, H. Zhang, X. Shen, and J. Jia, “Unsupervised Learning of Stereo Matching,” in *2017 IEEE International Conference on Computer Vision (ICCV)*,

2017, pp. 1576–1584.

[145] Y. Zhong, Y. Dai, and H. Li, “Self-Supervised Learning for Stereo Matching with Self-Improving Ability,” *arXiv Prepr. arXiv1709.00930*, 2017.

[146] A. Kendall, H. Martirosyan, S. Dasgupta, and P. Henry, “End-to-End Learning of Geometry and Context for Deep Stereo Regression,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 66–75.

[147] W. Luo, A. G. Schwing, and R. Urtasun, “Efficient Deep Learning for Stereo Matching,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5695–5703.

[148] A. A. Soltani, H. Huang, J. Wu, T. D. Kulkarni, and J. B. Tenenbaum, “Synthesizing 3D Shapes via Modeling Multi-view Depth Maps and Silhouettes with Deep Generative Networks,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2511–2519.

[149] A. Kar, C. Häne, and J. Malik, “Learning a Multi-View Stereo Machine,” in *Advances in Neural Information Processing Systems*, 2017, pp. 364–375.

[150] H. Park and K. M. Lee, “Look Wider to Match Image Patches With Convolutional Neural Networks,” *IEEE Signal Process. Lett.*, vol. 24, no. 12, pp. 1788–1792, 2017.

[151] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition,” in *Computer Vision -- ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part III*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 346–361.

[152] X. Ye, J. Li, H. Wang, H. Huang, and X. Zhang, “Efficient Stereo Matching Leveraging Deep Local and Context Information,” *IEEE Access*, vol. 5, pp. 18745–18755, 2017.

[153] Z. Liang *et al.*, “Learning Deep Correspondence through Prior and Posterior Feature Constancy,” *arXiv Prepr. arXiv1712.01039*, 2017.

[154] A. Saxena, S. H. Chung, and A. Y. Ng, “Learning depth from single monocular images,” in *Advances in neural information processing systems*, 2006, pp. 1161–1168.

[155] A. Saxena, S. H. Chung, and A. Y. Ng, “3-D Depth Reconstruction from a Single Still Image,” *Int. J. Comput. Vis.*, vol. 76, no. 1, pp. 53–69, Jan. 2008.

[156] F. Liu, C. Shen, G. Lin, and I. Reid, “Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2024–2039, 2016.

[157] Y. Kuznietsov, J. Stückler, and B. Leibe, “Semi-Supervised Deep Learning for Monocular Depth Map Prediction,” in *2017 IEEE Conference on Computer Vision*

and Pattern Recognition (CVPR), 2017, pp. 2215–2223.

[158] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper Depth Prediction with Fully Convolutional Residual Networks," in *2016 Fourth International Conference on 3D Vision (3DV)*, 2016, pp. 239–248.

[159] R. Garg, V. K. B.G., G. Carneiro, and I. Reid, "Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue," in *Computer Vision -- ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 740–756.

[160] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised Monocular Depth Estimation with Left-Right Consistency," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6602–6611.

[161] D. Eigen, C. Puhrsch, and R. Fergus, "Depth Map Prediction from a Single Image Using a Multi-scale Deep Network," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, 2014, pp. 2366–2374.

[162] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3061–3070.

[163] S. K. Nayar and Y. Nakagawa, "Shape from focus," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 8, pp. 824–831, 1994.

[164] A. Chambolle and T. Pock, "A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging," *J. Math. Imaging Vis.*, vol. 40, no. 1, pp. 120–145, 2011.

[165] M. Benning, F. Knoll, C.-B. Schönlieb, and T. Valkonen, "Preconditioned ADMM with nonlinear operator constraint," in *IFIP Conference on System Modeling and Optimisation*, 2015, pp. 117–126.

[166] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, 2012.

[167] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, "Joint Bilateral Upsampling," *ACM Trans. Graph.*, vol. 26, no. 3, Jul. 2007.

[168] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 International Conference on Computer Vision*, 2011, pp. 2564–2571.

[169] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *Proceedings of the 7th international joint conference on*

Artificial intelligence - Volume 2. Morgan Kaufmann Publishers Inc., Vancouver, BC, Canada, pp. 674–679, 1981.

[170] R. T. Collins, “A space-sweep approach to true multi-image matching,” in *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1996, pp. 358–363.

[171] H. Ha, S. Im, J. Park, H. G. Jeon, and I. S. Kweon, “High-Quality Depth from Uncalibrated Small Motion Clip,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5413–5421.

[172] K. Karsch, C. Liu, and S. B. Kang, “Depth Transfer: Depth Extraction from Video Using Non-Parametric Sampling,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2144–2158, 2014.

[173] M. Moeller, M. Benning, C. Schönlieb, and D. Cremers, “Variational Depth From Focus Reconstruction,” *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5369–5378, 2015.

[174] “Helicon Focus.” [Online]. Available: http://www.heliconsoft.com/heliconsoft-products/helicon-focus/.

[175] “Zerene Stacker.” [Online]. Available: http://zerenesystems.com/.

[176] S. Wanner and B. Goldluecke, “Globally consistent depth labeling of 4D light fields,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 41–48.

[177] B. Goldluecke, “cocolib.” [Online]. Available: http://cocolib.net.

[178] B. Goldluecke, E. Strekalovskiy, and D. Cremers, “The Natural Vectorial Total Variation Which Arises from Geometric Measure Theory,” *SIAM J. Imaging Sci.*, vol. 5, no. 2, pp. 537–563, 2012.

Appendix A: A Depth Map Post-Processing Approach based on Adaptive Random Walk with Restart

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 22, 2016, accepted August 12, 2016, date of publication August 26, 2016, date of current version September 28, 2016.

Digital Object Identifier 10.1109/ACCESS.2016.2603220

A Depth Map Post-Processing Approach Based on Adaptive Random Walk With Restart

HOSSEIN JAVIDNIA, (Student Member, IEEE), AND PETER CORCORAN, (Fellow, IEEE)
Department of Electronic Engineering, College of Engineering, National University of Ireland, Galway SW4 794, Ireland

Corresponding author: H. Javidnia (h.javidnia1@nuigalway.ie)

This work was supported by the Strategic Partnership Program of Science Foundation Ireland (SFI) and co-funded by SFI and FotoNation Ltd., on Next Generation Imaging for Smartphone and Embedded Platforms under Project 13/SPP/I2868.

ABSTRACT Accurate depth estimation is still an important challenge after a decade, particularly from stereo images. The accuracy comes from a good depth level and preserved structure. For this purpose, a depth post-processing framework is proposed in this paper. The framework starts with the ''Adaptive Random Walk with Restart (2015)'' algorithm. To refine the depth map generated by this method, we introduced a form of median solver/filter based on the concept of the mutual structure, which refers to the structural information in both images. This filter is further enhanced by a joint filter. Next, a transformation in image domain is introduced to remove the artifacts that cause distortion in the image. The proposed post-processing method is then compared with the top eight algorithms in the Middlebury benchmark. To explore how well this method is able to compete with more widely known techniques, a comparison is performed with Google's new depth map estimation method. The experimental results demonstrate the accuracy and efficiency of the proposed post-processing method.

INDEX TERMS Stereo matching, depth map, accuracy, edge preserving.

I. INTRODUCTION

A. STEREO DEPTH MAPS

In 3D computer graphics a depth map is an image or image channel that contains information relating to the distance to the surfaces of scene objects from a viewpoint [1]. The depth information corresponds to luminance in proportion to the distance from the camera. Near surfaces are depicted as lighter while far surfaces are shown as darker. Estimating the depth can be considered an important component of understanding geometric relations within a scene. In turn, such relations help to provide a richer representation of objects and their environment, often leading to improvements in existing recognition tasks, as well as enabling further applications such as robotics. In recent years, many new economical facilities, including time-of-flight [2], [3], structured light [4], and the Kinect were introduced for depth determination from stereo images. Kinect captures pairs of synchronized depth-color images for a scene within a range of several meters. However, the depth map cannot be used directly in scene reconstruction because it has some deficiencies such as gaps due to occlusion, reflection and other optical factors.

In general stereo algorithms or stereo matching algorithms are categorized into two groups based on the taxonomy scheme of Scharstein and Szeliski [5]: i.e. local and global algorithms.

In the local algorithms, the depth value at pixel P is dependent on the intensity and color values of the window W in which P is located. The initial matching cost is pixel-wise which is often noisy with minimum information in parts of the image with smoother texture. Therefore using the cost of the neighboring regions will assign the best depth value to pixel P.

On the other hand global methods consider the overall structure of the scene and smoothen the image and then try to solve the cost optimization problem.

B. STEREO MATCHING ALGORITHMS

In the last decade stereo matching has attracted a lot of attention from researchers and many matching algorithms have been developed. Some of the most well-known and studied algorithms are LIBELAS [6], iSGM [7], DBP [8] and CostFilter [9], LIBELAS [6] has been used since 2010 in different research studies. It is inspired from the observation that despite the fact that many stereo correspondences are highly ambiguous, some of them can be robustly matched.

While the processing speed of the LIBELAS is quite fast, the accuracy of the estimated depth map is poor.

iSGM [7] is an iterative scheme of Semi-global matching (SGM) technique with refined concept of the cost integration of semi-global matching. The gathered buffer is evaluated to a prior disparity map after horizontal and vertical integration.

DBP [8] is a global matching algorithm based on energy-minimization which as all other global methods contains data and smoothness term. The main contribution in data term in this algorithm is that, it is being approximated by a color weighted correlation. Afterwards, the data term is being refined in occluded regions by employing the hierarchical loopy belief propagation algorithm.

CostFilter [9] is a framework for multiple applications such as computing the disparity maps in real-time. It is the technique which aims to be fast and edge-aware. It consists of three steps: constructing a cost volume, fast cost volume filtering and winner-take-all label selection. The estimated depth by this method suffers from blocky artifacts along the edges and corners, especially in the regions with illumination transition. This causes a broken synthetic view along the edges.

There are other methods which tried to obtain better accuracy of depth map based on the combination of Markov Random Field (MRF) and sophisticated global optimization techniques in different researches [10]–[13], but still obtaining a good accuracy in depth estimation remains a challenge, especially in images with sophisticated or very simple texture.

Another approach which has been considered to improve the accuracy of the depth map by mostly preserving the edges was using the Mutual Information (MI) and SIFT features. A multisensor synthetic aperture radar (SAR) image registration method was proposed based on MI [14] and SIFT [15]. In this application, MI was used to estimate the registration parameters which were being used later by conjugate feature selection during the SIFT matching phase to decrease the number of false matches. Following the same idea, a stereo matching method was introduced in [16], based on the combination of MI, SIFT, plane-fitting and log-chromaticity color space.

Generally finding a local matching method which performs well in terms of both speed and accuracy is not easy and straightforward. But recently employing the random walk with restart along with optimizing the matching cost proved that it is possible to have fast matching with pretty accurate estimation. ARWR is a local matching algorithm based on random walk with restart method [17] which is used as the fundamental algorithm in this paper.

At this point it is timely to introduce the field of application, which establishes requirements for a high performance stereo disparity map. This work derives from research on automotive street-scene analysis where it is important to determine small objects in order to evaluate risks in the path of a vehicle – e.g. distant pedestrians, animals, vehicles. As most automotive imaging systems employ relatively small sensors (2-4 MP) compared to consumer devices it is important to be able to run disparity mapping algorithms at full native sensor resolution – in our case 2864 * 1924 pixels.

All current methods, as outlined above, suffer from non-accurate depth around edges and corners, depth discontinuity especially in texture-less areas, depth conflict around the area with similar colors and missing depth in one depth level. By solving these challenges a depth map can present correct and accurate depth information while respecting the structure of the reference image.

C. FEATURES OF THE PROPOSED METHOD

In this paper is presented a method to refine the depth map generated by the *Adaptive Random Walk with Restart* (ARWR) algorithm in order to obtain significant improvements in accuracy. The main features of the proposed method are:

1- A guided joint filter based on the mutual information was designed by diffusing the image domain.
2- Weights are allocated dynamically to the windows as part of the joint filter. The weights are being regenerated every time the window is moving to the other patch of pixels. The pixels count in different bins of a histogram instead of storing the weights directly.
3- The important point about the proposed filter is that it is rotation invariant because of the joint mutual information. Also the filter can be applied repeatedly to remove more noise but the edges and corners will be preserved because of the mutual joint feature.
4- When using this filter, the algorithm works better on high resolution images in comparison with low resolution.
5- This filter can be used for upsampling/downsampling purposes.
6- This method has the advantage of filling the depth map in regions with missing depth values.

The rest of this paper is organized as follows:

In the next section the chosen method, ARWR is presented in detail. Section 3 provides the details of the proposed post-processing filter. The results of the evaluation as well as experimental results are presented in section 4, while conclusions are drawn in section 5. There are also 2 appendices linked to this paper presenting extended numerical and visual results.

II. INTRODUCTION TO ADAPTIVE RANDOM WALK WITH RESTART

In this section we describe the fundamental and technical details of the chosen stereo matching method, ARWR.

ARWR has an acceptable and comparable performance in terms of estimation and speed against other algorithm, but it is still far from the top stereo matching algorithm on Middlebury benchmark in terms of accuracy.

This algorithm has several important advantages which make it a suitable method for a variety of applications. It is

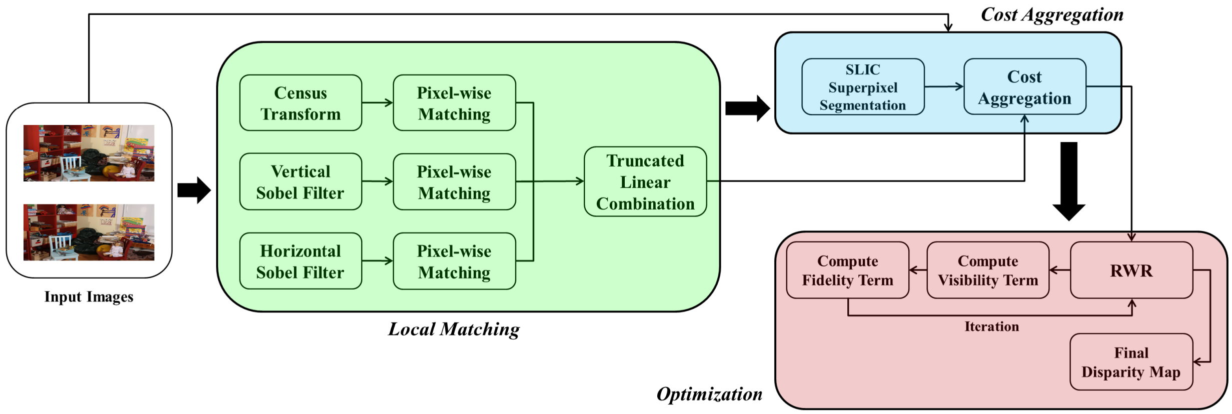


FIGURE 1. Overview of the adaptive random walk with restart.

not affected by illumination variation because of gradient and census transform, the processing time is quite fast in comparison with recently studied methods, has good performance in both outside and inside environment and gives us the option to have a estimation of the depth in low texture scenes.

One important advantage of this algorithm which convinced us to employ it as a part of our approach, is the good performance on high resolution images. A traditional way to speed up stereo computation is to use image pyramids or downsized images which also reduce the disparity range. This down-sampling in disparity computation will cause some small objects to be missed. The full disparity resolution for large distance is vital for long range object detection. The point about the chosen algorithm is that the image doesn't need to be down-sampled to speed up the method.

The comparison of this method with several others methods done in this paper showed that it has acceptable depth estimation in high resolution images, 2864 * 1924 pixels.

Acceptable depth estimation refers to the fact that the algorithm doesn't have the problem of estimating different layers of depth in one object. It respects the depth layers without conflict. This feature along with the fast processing time makes this algorithm suitable for high resolution real-time applications. Also it gives us the ability of making a more accurate filter, which is described later in the paper.

A. ALGORITHM DESIGN

The initial matching cost in ARWR is pixel-wise calculated by employing census transform and gradient image matching. Census-based matching technique or census transform was initially introduced by Zabi in 1994 [18]. It is a form of non-parametric local transform to map the intensity values of the pixels within a square window to a bit string, thereby capturing the image structure. In other words, it computes for every pixel a binary string (census signature) by comparing its grey value with the grey values in its neighborhood.

The census transform is robust to radiometric variations but the noise in the local image structure is being encoded based on the intensity of the pixels. The encoded noise brings some matching doubts especially in the area with repetitive or similar texture patterns.

To overcome this problem gradient image matching is employed as part of the local matching block in ARWR. At this stage gradient images are computed using 5 × 5 Sobel filters. The whole process of the ARWR is shown in Fig. 1.

The green block in Fig. 1 shows the local matching block including the transformation and matching parts.

The usual similarity criteria in stereo matching are only strictly valid for surfaces with Lambertian (diffuse) reflectance characteristics. Specular reflections are viewpoint dependent and may cause large intensity difference at corresponding image points. In the presence of specular reflection, traditional stereo methods are often unable to establish any correspondence, or the calculated disparity values tend to be inaccurate.

In this case using the gradient image matching makes the local matching method more robust on non-Lambertian surfaces.

The noise variation in the local pixel-wise matching methods can be vital in term of the performance. That is why SLIC (Simple Linear Iterative Clustering) algorithm is employed in ARWR, the blue block in Fig. 1. SLIC is one of the common super-pixeling methods [19].

The local measurements in the matching block are more robust to noise variation when the super-pixels are considered as the smallest parts of the image to be matched to the target image. Super-pixeling is considered as an alternative to pixels in pixel-wise matching which leads to a reduction in memory requirements in the whole algorithm.

At the last step of the ARWR which is shown as pink block in Fig. 1, the calculated matching cost is updated using the RWR algorithm to determine the optimum disparity with respect to occluded and discontinuity regions. The standard

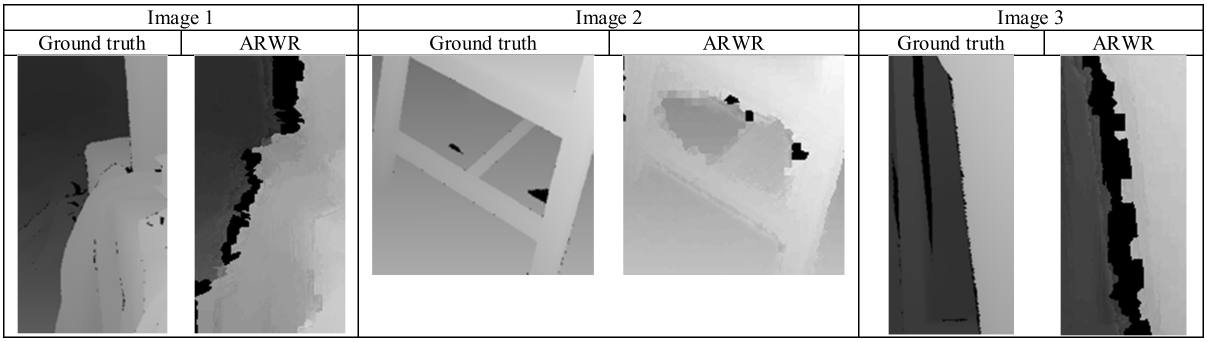


FIGURE 2. Broken edges and corners in the computed depth map by ARWR.

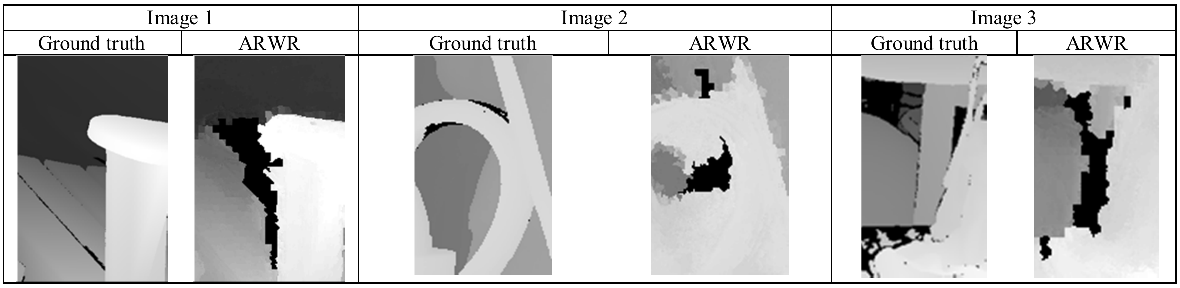


FIGURE 3. Missing patches in the computed depth map by ARWR.

cost update algorithm in RWR is modified in ARWR where the matching cost is updated adaptively by considering the position of the super-pixels in the regions of occlusion or depth discontinuity.

To recover the smoothness failure at occlusion or depth discontinuity regions in ARWR, a visibility constraint is formulated within the RWR algorithm which requires an occluded pixel to have no match on the target image, and a non-occluded pixel to have at least one match.

B. ALGORITHM TRADE OFF

There are some issues with the generated depth map based on ARWR which need to be solved to obtain a clearer and more accurate depth map.

The depth map produced by the ARWR is suffering from speckle noise and inaccurate object edges especially for objects with a detailed geometry. Basically the generated map is not preserving the edges and corners. At some parts of the computed depth map the edges are broken or they are faded into other objects which makes it unsuitable for segmentation purposes and classification. Fig. 2 shows examples of the broken edges and corners in the computed depth and the corresponding patches in the ground truth.

The other issue is the missing parts in the generated map. We demonstrate that each patch of pixels in a depth map can provide us valuable information like the scaling factor and distance to the objects. Fig. 3 represents some samples of the missing parts in the depth map and the corresponding patches in the ground truth.

The samples show that some parts of the depth map were not estimated by ARWR and it brings a false depth level which is not suitable for 3D reconstruction applications.

These issues are generally some of the most challenging problems in the current depth computation and enhancement methods. Having a map which is preserving the right edges and corners while all pixel patches are contributing in the depth level allows us to reconstruct an accurate 3D scene from the camera view point. It also provides an accurate fundamental platform for variety of applications such as classification, segmentation, distance estimation, obstacle detection and autonomous navigation.

In the next section of this paper our approach is presented and shown to provide a suitable solution to the issues mentioned above.

III. PROPOSED POST-PROCESSING FILTER

To solve the issues mentioned in the previous section, mutual information of the reference image and the depth map is used as the input of the joint weighted median filter. By employing the mutual joint filter the problem of the regions of occlusion or depth discontinuity in the initial depth map is solved. To resolve the blocky artifacts from object edges, the depth map is transferred to another domain by convolving it.

The whole process of the ARWR + proposed post-processing method is as follows:

1- Extract the initial depth by using the ARWR algorithm.

2- *A*. Apply mutual joint weighted median filter to fill the regions of occlusion or depth discontinuity in the initial depth map

B. Overwrite the structure of the RGB image on the depth map.

3- Transfer the depth map to a signal and perform normalized interpolated convolution on the domain of the signal to obtain an accurate, edges preserved depth map.

Fig. 4 presents the general overview of the whole process and Fig. 5 shows the detailed view of the ARWR + proposed post-processing method.

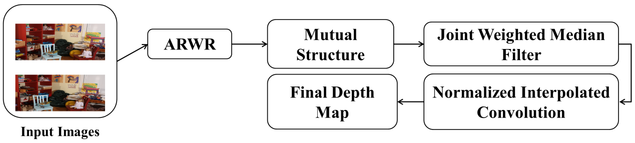

Input Images
ARWR
Mutual Structure
Joint Weighted Median Filter
Normalized Interpolated Convolution
Final Depth Map

FIGURE 4. Overview of the proposed post-processing method.

A. MUTUAL-STRUCTURE

Mutual information has developed into an accurate measure for rigid and affine mono- and multimodality image registration or for two images, it is a combination of the entropy values of the images, both separately and jointly [20]. By measuring the structure similarity of two images, we can let the mutual-structure to guide the joint filtering process. Let's denote D and I as the initial depth map and the reference RGB image respectively. Also D_p and I_p are the pixel intensities in initial depth map and the reference RGB image respectively. To compute the structure similarity between two images, we consider a variety of patches in the images. One common and well-studied method to measure the structure similarity is to use normalized cross covariance (1). If we consider the images as two time series signals, then we can delay D by W samples and then calculate the cross-covariance between the pair of signals,

$$CC\left(W\right) = \frac{1}{M-1}\sum_{k=1}^{M}(D_{k-W}-\mu_D)(I_k-\mu_I), \tag{1}$$

Where μ_D and μ_I are the means of each time series and there are M samples in each. $CC\left(W\right)$ is the cross-covariance function. Normalized cross-covariance is called cross-correlation,

$$N\left(W\right) = \frac{CC\left(W\right)}{\sqrt{\sigma(D_p)\sigma(I_p)}}, \tag{2}$$

$$N\left(D_p, I_p\right) = \frac{cov(D_p, I_p)}{\sqrt{\sigma(D_p)\sigma(I_p)}}, \tag{3}$$

Where $cov(D_p, I_p)$ is the covariance of patch intensity. $\sigma\left(D_p\right)$ and $\sigma(I_p)$ denote the variances of pixel intensities in the initial depth map and RGB image respectively. The maximum value of $N\left(D_p, I_p\right)$ is 1 when two patches are with the same edges, otherwise $\left|N\left(D_p, I_p\right)\right| < 1$. Nonlinear computation makes it hard to use the normalized cross-correlation directly in the process. To solve this problem, making a connection between normalized cross-correlation and least-square regression would be helpful. If we consider $H(p)$ as a patch centered at pixel p, then the least-squared regression function would be:

$$f\left(D, I, \alpha_p^1, \alpha_p^0\right) = \sum_{q\in H(p)}\left(\alpha_p^1 D_q + \alpha_p^0 - I_q\right)^2, \tag{4}$$

Where α_p^1 and α_p^0 are the regression coefficients. This function linearly represent one patch in D corresponding with the one in I. Minimum error with the optimal α_p^1 and α_p^0 can be defined as:

$$e(D_p, I_p)^2 = \frac{min}{\alpha_p^1, \alpha_p^0}\frac{1}{|H|}f\left(D, I, \alpha_p^1, \alpha_p^0\right), \tag{5}$$

By considering the (1) and (5), we can say the mean square error is:

$$e\left(D_p, I_p\right) = \sigma\left(I_p\right)\left(1 - N\left(D_p, I_p\right)^2\right), \tag{6}$$

The relation between the mean square error and normalized cross-correlation is previously proved in [19]. When $\left|N\left(D_p, I_p\right)\right| = 1$, it means that two patches only contain mutual structure and $e\left(D_p, I_p\right) = 0$. So:

$$e(I_p, D_p)^2 = \frac{min}{b_p^1, b_p^0}\frac{1}{|H|}f\left(I, D, b_p^1, b_p^0\right), \tag{7}$$

Therefore $e\left(I_p, D_p\right) = 0$ when $\left|N\left(D_p, I_p\right)\right| = 1$. According to the above analysis, the structure similarity can be defined as:

$$S_s\left(D, I, \alpha, b\right) = \sum_p\left(f\left(D, I, \alpha_p^1, \alpha_p^0\right) + f(I, D, b_p^1, b_p^0)\right), \tag{8}$$

where α and b are the coefficient sets of $\left\{\alpha_p^1, \alpha_p^0\right\}$ and b_p^1, b_p^0 respectively.

Algorithm 1 computes the mutual information of D and I.

Algorithm 1 Mutual Information

```
Input: Image $D$ and $I$
Output: Mutual Information of $D$ and $I$
    Initialize $W, M$ to 0;
    Initialize $\alpha = \beta(\alpha_p)$;
    Initialize $b = \beta(b_p)$;
      $\mu_D \leftarrow mean(D)$;
      $\mu_I \leftarrow mean(I)$;
      $\sigma_W = M / \sum\left(D_{M-W} - \mu_D\right)\left(I_M - \mu_I\right)$;
        foreach $H$ in $D$ do
            $\sum\left(\alpha_p D_{N(p)} + \alpha_p - I_{N(p)}\right)^2$;
        end
     return $S\left(D, I, \alpha, b\right)$;
```

B. JOINT WEIGHTED MEDIAN FILTER

Median filter [21] is a nonlinear operation which runs through an image I and replaces each pixel value V by the median value of neighboring pixels within a $(2j+1)^2$ window W_p:

$$I_{median}\left(p\right) = median\left\{V : p_i \in W_p\right\}, \tag{9}$$

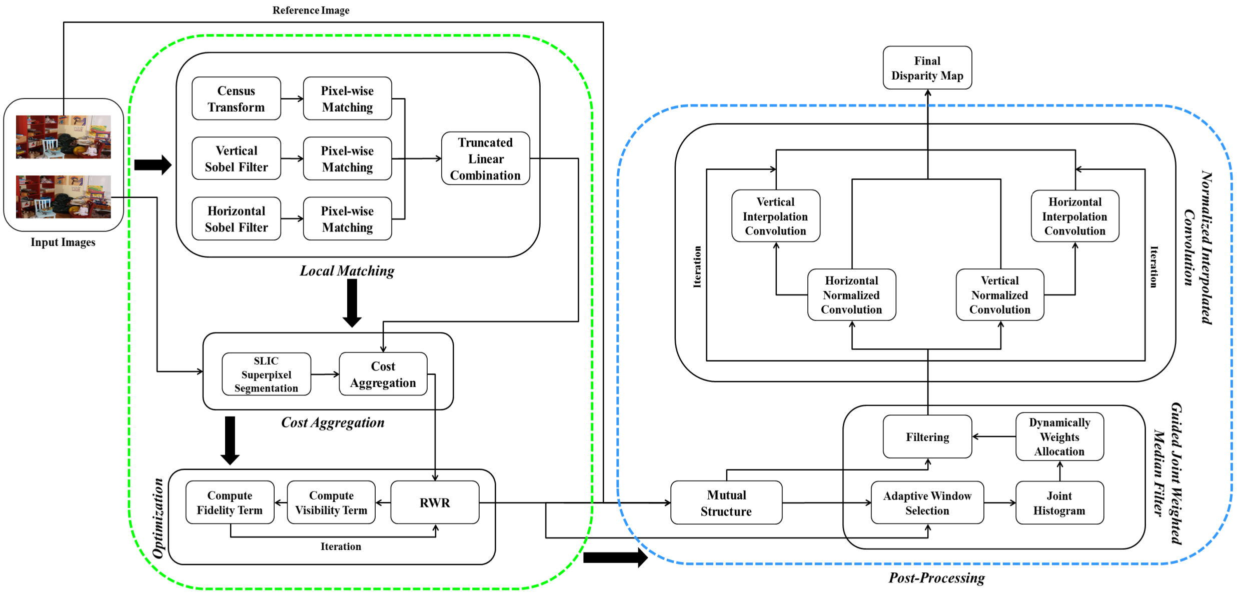


FIGURE 5. Overview of the ARWR + Proposed post-processing method.

Median filter processes all the neighbors equally and may lead to some artifacts like changing the shape of the sharp corners and make them circular or removing thin structures. Weighted median filter [22] was introduced to solve this issue. Considering $\omega(p, p')$ the weight on image I, then:

$$h(p, i) = \sum_{p' \in W_p} \omega(p, p')\delta(V(p') - i), \tag{10}$$

where W_p is a local windows near p, i is the discrete bin index and $\delta(.)$ is the Kronecker delta function which is 1 when the argument is 0, otherwise it is 0. $h(p, .)$ is the local histogram with the weighted pixel in it. By accumulating $h(p, i)$ the weighted median value is obtained.

Joint median filter on a depth map D with a group S of segments as masks is defined as:

$$D_{J_median}(p) = median\left\{D(p_i) : p_i \in W_p \cap S_p\right\}, \tag{11}$$

where $S_p \in S$ is the segment containing pixel p. So the new local histogram for depth map would be:

$$h_D(p, i) = \sum_{p' \in W_p \cap S_p} \delta(D(p') - i), \tag{12}$$

Based on the (10) and (12), the local histogram of the joint weighted median filter on the depth map D would be:

$$h_{D_f}(p, i) = \sum_{p' \in W_p \cap S_p} \omega(p, p')\delta(D(p') - i), \tag{13}$$

Using the mutual structure and joint weighted median filter gives us the capability to transfer the structural information of the reference image to the depth map, instead of transferring the whole pattern. And in addition it contributes greatly to a preservation of the edges in the depth map.

C. NORMALIZED INTERPOLATED CONVOLUTION

Joint weighted median filter based on the mutual structure provides an edge preserved and smooth depth image, but still the depth map is suffering from blocky artifact, especially on the edges. To decrease the blocky effects on the depth map, converting the image to another domain would be helpful. Let's consider a signal:

$$f(t) = [x_1; 0; 0; x_4; x_5; 0; x_7; 0], \tag{14}$$

where x_i are known samples of signals and the missing samples are replaced by 0.

A simple smoothing filter is:

$$g(t) = \left[\frac{1}{3}; \frac{1}{3}; \frac{1}{3}\right], \tag{15}$$

Filling the missing part of the $f(t)$ by applying the $g(t)$ will provide:

$$f(t) \times g(t) = \left[\frac{x_1}{3}; \frac{x_1}{3}; \frac{x_4}{3}; \frac{x_4 + x_5}{3}; \frac{x_4 + x_5}{3}; \frac{x_5 + x_7}{3}; \frac{x_7}{3}; \frac{x_7 + x_1}{3}\right], \tag{16}$$

At this level using the Normalized Convolution appends a component to each signal which expresses the confidence of a signal. This component is equal to 0 for each missed sample. If we consider the map of the component on signal $f(t)$ as $g(t)$, then:

$$c(t) = [1; 0; 0; 1; 1; 0; 1; 0], \tag{17}$$

By considering the convolution of $c(t)$, it is possible to approximate the original signal with the filled gaps. So:

$$f(t)_O = \frac{f(t) \times g(t)}{c(t) \times g(t)}, \tag{18}$$

where the $f(t)_O$ is the original signal without gaps.

This scenario previously has been studied to filter the non-uniform sampled signals [23]. If $T_\omega(ct\,(x))$ is a uniformly sampled signal in Ψ_w, then for a uniform discretization $U(\Psi)$ of the original domain Ψ, normalized convolution generates the smoothed value of a sample $q \in U(\Psi)$ as:

$$Fi\,(q) = \left(\frac{1}{J_q}\right) \sum_{l \in U(\Psi)} T\,(l)\,R\left(t\left(\hat{q}\right), t\left(\hat{l}\right)\right), \tag{19}$$

Where $J_q = \sum_{l \in U(\Psi)} R(t\,(\hat{q})\,, t(\hat{l}))$ is a normalized factor for q and R is an arbitrary kernel. Generally interpolated surfaces in an image are smoother than the corresponding ones generated by normalized convolution. To obtain this, $Fi(q)$ can be filtered by continuous convolution as below:

$$CCF\,(q) = \int_{U_\Psi} Fi\,(x)\,R\left(t\left(\hat{q}\right), x\right) dx, \tag{20}$$

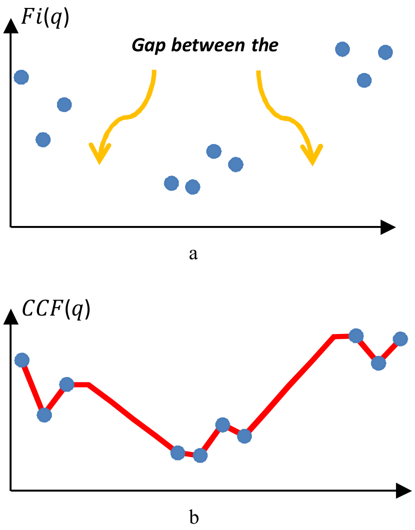


FIGURE 6. Missing samples recovery. (a) Samples of a signal with missing parts. (b) Recovered samples in domain Ψ.

Where R is a normalized kernel. Fig. 6.b shows how the missing samples of signal T are recovered in domain Ψ.

Applying the same process on a depth map generates a smooth and artifact free map by transferring it into the domain Ψ.

IV. EVALUATION

A. MIDDLEBURY BENCHMARK

The Middlebury benchmark has been widely used over the last decade to evaluate the performance of stereo matching algorithms [24]. The ARWR was applied with and without the proposed post processing on 15 standard images from the Middlebury 'dense' training dataset. Based on the average weight on metric 'bad 2.0', the first 8 algorithms from Middlebury were chosen for comparison, including GCSVR [25], INTS [26], MCCNN_Layout [25], MC-CNN+FBS [25], MC-CNN-acrt [27], MC-CNN-fst [27], MeshStereo [28], SOU4P-net [25] and the original ARWR without post-processing. As evaluation metrics we consider the ones presented in Table 1.

TABLE 1. Metrics used in this paper to evaluate the algorithms.

	Metric	Formula
1	MSE (Mean Squared Error)	$\frac{1}{mn}\sum_{0}^{m-1}\sum_{0}^{n-1}\lVert f(i,j)-g(i,j)\rVert^2$
2	RMSE (Root Mean Squared Error)	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}$
3	PSNR (Peak Signal to Noise Ratio)	$20\log_{10}(\frac{MAX_f}{\sqrt{MSE}})$
4	SNR (Signal to Noise ratio)	$\frac{10\log_{10}(P_{signal})}{10\log_{10}(P_{noise})}$
5	MAE (Mean Absolute Error)	$\frac{1}{n}\sum_{i=1}^{n}\lvert f_i-y_i\rvert=\frac{1}{n}\sum_{i=1}^{n}\lvert e_i\rvert$
6	SSIM (Structural Similarity Index)	$\frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{(\mu_x^2+\mu_y^2+c_1)(\sigma_x^2+\sigma_y^2+c_2)}$
7	DSSIM (Structural Dissimilarity Index)	$\frac{1-SSIM(x,y)}{2}$

All the evaluation process in this paper is based on the high quality version of the images and all experiments were done under the same conditions.

All the images were normalized before evaluation and maximum disparity setup was defined for all algorithms. The average value of the 15 images in each metric was considered as the representing value of the corresponding algorithm. Table 2 shows the average value of metric/algorithm. To find the extended tables for each metric/image (color coded to better present relative performance of each algorithm for each evaluation metric) please refer to Appendix 1.

The best algorithm's value in each metric is emboldened. Based on the MSE, PSNR, SNR, SSIM and DSSIM metrics the proposed post-processing method has the best performance. Table 3 represents the ranking within the 10 tested algorithms of the ARWR without post-processing and with post processing applied for each of the evaluated metrics.

Fig.7 presents the results of the proposed post-processing method on three Middlebury database images.

The initial depth map is computed by ARWR. Beside the parametric evaluation, the visual comparison of the generated results and the ground truth clarify the fact that the proposed post-processing method can preserve edges and the structure. For more results of the post-processed ARWR and visual comparison with other methods please refer to Appendix 2.

While the performance of the proposed post-processing method in term of accuracy is good, the processing time is a trade-off. Fig.8 shows the processing time required by each step of the proposed post-processing method on an image with 962×1414 pixels resolution ran on Matlab R2013a. The initial disparity is estimated with a maximum disparity of 256.

TABLE 2. Average values of metric/algorithm.

	MCCNN_ Layout	MC-CNN-acrt	MC-CNN+FBS	SOU4P-net	MC-CNN-fst	MeshStereo	INTS	GCSVR	Original ARWR	Post-processed ARWR
MSE	0.0133	0.0171	0.0194	0.0133	0.0177	0.0195	0.0193	0.0235	0.0277	**0.0126**
RMSE	**0.104**	0.1199	0.1243	0.1069	0.1225	0.1357	0.1339	0.1456	0.1455	0.1041
PSNR	20.2902	19.028	19.2026	19.9836	18.842	17.6704	17.8519	17.2273	17.7517	**20.3136**
SNR	15.3891	14.1269	14.3016	15.0826	13.9409	12.7694	12.9509	12.3262	12.8506	**15.4125**
MAE	**0.0524**	0.0639	0.0915	0.0739	0.0666	0.101	0.1026	0.112	0.0867	0.0644
SSIM	0.99849	0.9981	0.9975	0.99847	0.998	0.9975	0.9976	0.9969	0.9966	**0.9985**
DSSIM	0.0008	0.001	0.0012	0.0008	0.001	0.0011	0.0012	0.0015	0.0017	**0.0007**

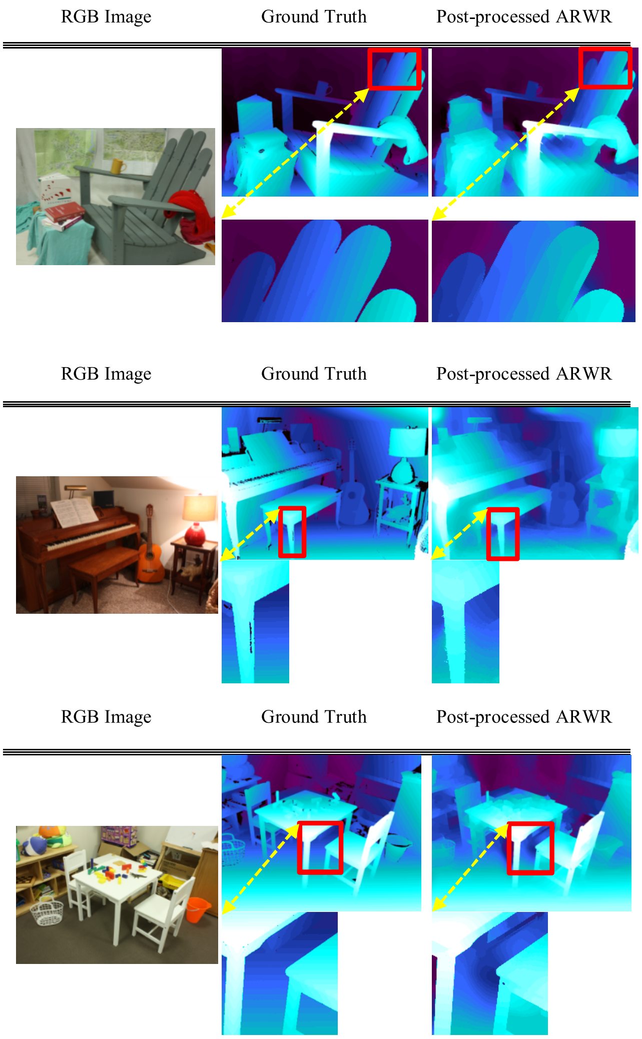


FIGURE 7. The result of the sample images from Middlebury database. Each set of figures denotes the left image, the ground truth and the proposed postprocessed depth map.

Table 4 represents the average performing time of the all algorithms applied on the same high resolution image set as per Middlebury.

The processing time of the studied method is poor, but can be readily improved as much of this work was not optimized for fast computation. The improvement of algorithm efficiency and computational speed is currently the subject of a follow-on research project to optimize for an embedded DSP or GPU implementation.

TABLE 3. Ranking of ARWR without and with post-processing out of 10 algorithms.

	ARWR without post-processing	ARWR with post-processing
MSE	9	1
RMSE	9	2
PSNR	8	1
SNR	8	1
MAE	6	3
SSIM	10	1
DSSIM	10	1

TABLE 4. The processing time of the studied algorithms on same high resolution image set.

	Algorithm	Time/Sec
1	MC-CNN-fst	1.26
2	ARWR without post-processing	21
3	MeshStereo	62
4	INTS	104
5	MC-CNN-acrt	106
6	MC-CNN+FBS	157
7	MCCNN_Layout	300
8	**ARWR with post-processing**	**440**
9	SOU4P-net	688
10	GCSVR	5891

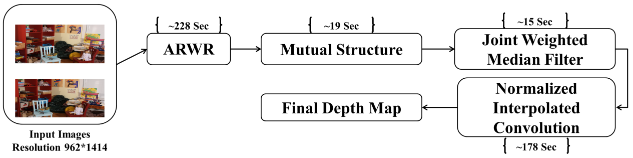


FIGURE 8. Processing time required by each step of the algorithm.

B. COMPARISON WITH GOOGLE'S DEPTH ESTIMATION TECHNIQUE

In the second part of the evaluation we referred to the recent technology which is used by the Google Camera "Lens Blur" feature in Android OS. The basic idea in this technology is to match the stereo images in the bilateral space by avoiding

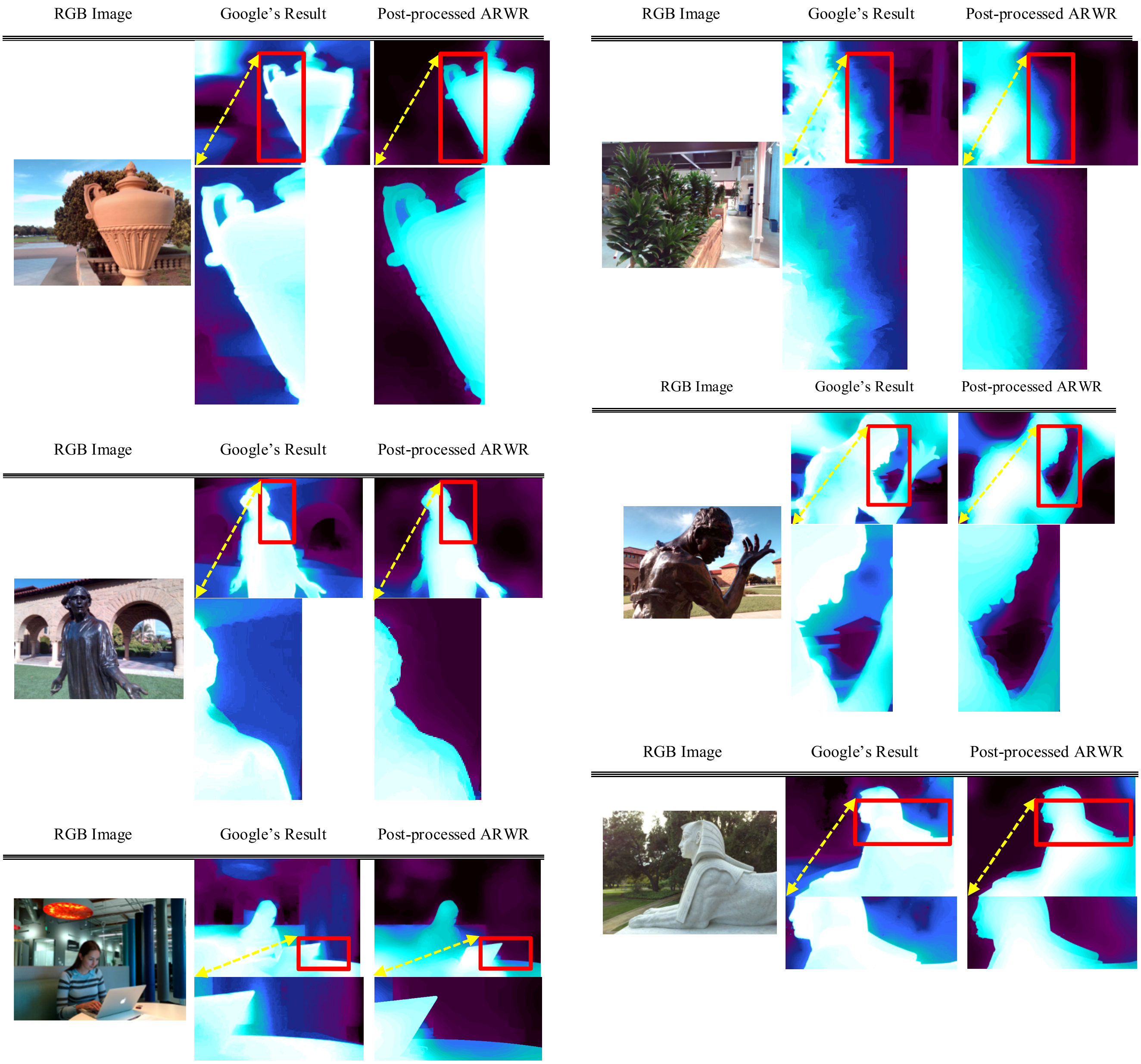


FIGURE 9. **The result of the images from Google's method [29]. Each set of figures denotes the left image, Google's result and the proposed post-processed depth map.**

per-pixel inference using leveraging techniques for fast bilateral filter [29]. This idea is presented in the other form to compute the depth from focus in the handheld devices by using focal stack.

A global approach is employed to generate the depth map by minimizing a cost function related to the pixel disparities. The data matching cost in their method is based on the Birchfield-Tomasi technique [30].

To satisfy the smoothness term of the cost function, the bilateral filter is used which causes a smoother image while the edges are preserved. For each pixel i of an image, one would typically consider a square (kernel) centered at i and perform a convolution.

Minimizing the cost function is extremely slow for higher resolution pictures. This problem is solved by splatting the value of each pixel into a higher dimensional bilateral space. The general idea is to; instead of applying the bilateral filter in pixel space, splat the pixels according to their location and color into a five-dimensional bilateral grid. Then blur the grid using a short range isotropic blur filter, and slice the grid in order to recover the filtered image.

According to the authors of [29], the most instinctive way to evaluate the performance of a stereo algorithm for defocus is to visually inspect the renderings produced using that algorithm. The kind of error that they cared about was related to failing to follow image edges at occlusion boundaries

TABLE 5. Structural similarity and dissimilarity of Google's method and post-processed ARWR.

	SSIM	DSSIM
Image 1	0.9939	0.0031
Image 2	0.9955	0.0022
Image 3	0.9867	0.0066
Image 4	0.9960	0.0020
Image 5	0.9907	0.0047
Image 6	0.9979	0.0011

where errors in disparity can cause rendering errors. The Middlebury error metrics are not considering this type of error. Middlebury error metrics are pixel-wise and Google's method has a poor performance on this benchmark, because their algorithm over- or under-estimate the disparity of flat texture-less regions, has disparity confusion in close shots with different level of brightness, has disparity confusion at the regions with specific pattern and sharp opposite colors.

Unfortunately there is no ground truth and benchmark based on this method. We only had access to a number of images and generated disparities which are published in [29].

To find out the structural similarity of the Google's result and the proposed post-processing method, we employed SSIM and DSSIM metrics. For two identical images the values of SSIM and DSSIM are 1 and 0 respectively. Table 5 shows how close are our results to Google's for each image and with the same disparity level setup. The visual comparison of the Google's technique and the post-processed ARWR is shown in Fig. 9. The visual comparison shows different patches of the estimated disparity by Google's and our method. This visual and numerical comparison show how close the proposed method is to Google's in terms of preserving the structure of the estimated disparity, edges and corners.

V. CONCLUSION

In this paper we proposed and evaluated a post-processing technique to increase the accuracy of the depth map computed by Adaptive Random Walk with Restart method. We demonstrated that keeping the sharp edges and corners along with main structure of the reference image in the depth map is an important factor to increase the accuracy. The proposed method uses the combination of the mutual structure of the RGB image to keep the structure and joint weighted filter to make the depth planes smooth and fill the regions of discontinuity. Transferring the depth map to another domain gave us the option to implement normalized interpolated convolution to remove the blocky artifacts of around the edges and corners. The comparison with the top 8 methods of the Middlebury benchmark and the ARWR without post-processing proved the performance quality of the proposed method. The value of the average structural similarity index which is about 0.9935 with Google's stereo matching method is another confirmation on the performance of the discussed method.

With respect to the performance of the studied method in this paper, there are still a number of open challenges such as reducing the processing time, while maintaining the same accuracy in real-time applications with low processing power. This challenge motivates our future research activity. In follow-on work it is planned to filter each image in 8-16 dimensional bilateral space instead of employing a normalized interpolation. Preliminary experiments indicate this could improve the speed of the enhanced ARWR by as much as an order of magnitude. This refinement would make the post-processed ARWR algorithm competitive in terms of computation time with the top 2-3 algorithms form Middlebury.

ACKNOWLEDGMENT

The authors would like to thank Dr. Claudia Costache for her helpful comments.

REFERENCES

[1] *Computer Arts/3D World Glossary*, Future plc, Somerset, U.K., 2011.
[2] C. Niclass, M. Soga, H. Matsubara, and S. Kato, "A 100m-range 10-frame/s 340×96-pixel time-of-flight depth sensor in 0.18 μm CMOS," in *Proc. ESSCIRC (ESSCIRC)*, Sep. 2011, pp. 107–110.
[3] C. Niclass *et al.*, "Design and characterization of a 256×64-pixel single-photon imager in CMOS for a MEMS-based laser scanning time-of-flight sensor," *Opt. Exp.*, vol. 20, no. 11, pp. 11863–11881, May 2012.
[4] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1. Jun. 2003, pp. I-195–I-202.
[5] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vis.*, vol. 47, nos. 1–3, pp. 7–42, Apr. 2002.
[6] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," in *Proc. 10th Asian Conf. Comput. Vis. Comput. Vis. (ACCV)*, Queenstown, New Zealand, Nov. 2010, pp. 25–38.
[7] S. Hermann and R. Klette, "Iterative semi-global matching for robust driver assistance systems," in *Proc. 11th Asian Conf. Comput. Vis. Comput. Vis. (ACCV)*, Daejeon, South Korea, Nov. 2012, pp. 465–478.
[8] Q. Yang, L. Wang, R. Yang, H. Stewénius, and D. Nistér, "Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 3, pp. 492–504, Mar. 2009.
[9] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 3017–3024.
[10] R. Kozik, "Improving depth map quality with Markov random fields," in *Image Processing and Communications Challenges 3*, R. S. Choraś, Ed. Berlin, Germany: Springer, 2011, pp. 149–156.
[11] K.-H. Lo, K.-L. Hua, and Y.-C. F. Wang, "Depth map super-resolution via Markov random fields without texture-copying artifacts," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2013, pp. 1414–1418.
[12] C. D. Herrera, J. Kannala, P. Sturm, and J. Heikkila, "A learned joint depth and intensity prior using Markov random fields," in *Proc. Int. Conf. 3D Vis. (3DV)*, Jun./Jul. 2013, pp. 17–24.
[13] S. Zheng, P. An, Y. Zuo, X. Zou, and J. Wang, "Depth map upsampling using segmentation and edge information," in *Proc. 8th Int. Conf. Image Graph. (ICIG)*, Tianjin, China, Aug. 2015, pp. 116–126.
[14] H. Hirschmuller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2. Jun. 2005, pp. 807–814.
[15] S. Suri, P. Schwind, P. Reinartz, and J. Uhl, "Combining mutual information and scale invariant feature transform for fast and robust multisensor SAR image registration," presented at the 75th Annu. ASPRS, Baltimore, MD, USA, 2009.

[16] Y. S. Heo, K. M. Lee, and S. U. Lee, ''Joint depth map and color consistency estimation for stereo images with different illuminations and cameras,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 5, pp. 1094–1106, May 2013.

[17] S. Lee, J. H. Lee, J. Lim, and I. H. Suh, ''Robust stereo matching using adaptive random walk with restart algorithm,'' *Image Vis. Comput.*, vol. 37, pp. 1–11, May 2015.

[18] R. Zabih and J. Woodfill, ''Non-parametric local transforms for computing visual correspondence,'' in *Proc. 3rd Eur. Conf. Comput. Vis. Comput. Vis. (ECCV)*, Stockholm, Sweden, May 1994, pp. 151–158.

[19] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, ''SLIC superpixels compared to state-of-the-art superpixel methods,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[20] X. Shen, C. Zhou, L. Xu, and J. Jia, ''Mutual-structure for joint filtering,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3406–3414.

[21] Y. Zhu and C. Huang, ''An improved median filtering algorithm for image noise reduction,'' *Phys. Procedia*, vol. 25, pp. 609–616, Apr. 2012.

[22] G. R. Arce, ''A general weighted median filter structure admitting negative weights,'' *IEEE Trans. Signal Process.*, vol. 46, no. 12, pp. 3195–3205, Dec. 1998.

[23] H. Knutsson and C.-F. Westin, ''Normalized and differential convolution,'' in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 1993, pp. 515–523.

[24] D. Scharstein *et al.*, ''High-resolution stereo datasets with subpixel-accurate ground truth,'' in *Proc. 36th German Conf. Pattern Recognit. (GCPR)*, Münster, Germany, Sep. 2014, pp. 31–42.

[25] *Middlebury Stereo Evaluation—Version 3*. [Online]. Available: http://vision.middlebury.edu/stereo/eval3/

[26] X. Huang, Y. Zhang, and Z. Yue, ''Image-guided non-local dense matching with three-steps optimization,'' in *Proc. ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. III-3. 2016, pp. 67–74.

[27] J. Žbontar and Y. LeCun, ''Stereo matching by training a convolutional neural network to compare image patches,'' *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2287–2318, Jan. 2016.

[28] C. Zhang, Z. Li, Y. Cheng, R. Cai, H. Chao, and Y. Rui, ''MeshStereo: A global stereo model with mesh alignment regularization for view interpolation,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2057–2065.

[29] J. T. Barron, A. Adams, Y. Shih, and C. Hernández, ''Fast bilateral-space stereo for synthetic defocus,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4466–4474.

[30] J. T. Barron, A. Adams, S. YiChang, and C. Hernandez, ''Fast bilateral-space stereo for synthetic defocus—Supplemental material,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015, pp. 1–15.

HOSSEIN JAVIDNIA received the master's degree in information technology engineering from the University of Guilan, Iran, in 2014. He is currently pursuing the Ph.D. degree in electrical engineering with the National University of Ireland, Galway. His current research interests include image processing, machine vision, and automotive navigation.

PETER CORCORAN (F'10) has co-authored over 300 technical publications and co-inventor on more than 250 granted US patents. His research interests include biometrics, cryptography, computational imaging, and consumer electronics. He is the Editor-in-Chief of the *IEEE Consumer Electronics Magazine* and a Professor with a Personal Chair at the College of Engineering & Informatics at NUI Galway. In addition to his academic career, he is also an Occasional Entrepreneur, Industry Consultant, and Compulsive Inventor.

Appendix 1:

At this part we present extended results of the numerical comparison of post-processed ARWR and other methods. These results are based on the 15 standard images of Middlebury training dense set. Each table represents one of the primary metrics which has been used in the evaluation part of the paper.

The cells are colour encoded in each row based on the ranking of the each method per image. The light-green, represents the best performed method on an image and orange shows the last method.

Table 1. The values of PSNR for Image/Algorithm

	ARWR without post-process	ARWR with post-process	GCSVR	INTS	MCCNN_Layout	MC-CNN+FBS	MC-CNN-acrt	MC-CNN-fst	MeshStereo	SOU4P-net
Adirondack	16.47441	23.07429	24.71369	25.97614	19.97654	26.08177	19.46033	17.34919	24.268	21.37629
ArtL	10.86264	13.64302	11.85928	18.83427	11.14089	11.14726	10.79723	10.88494	19.62349	12.05687
Jadeplant	13.67408	16.30873	17.48587	18.53229	17.74748	19.01975	16.04076	16.30057	16.49046	17.99232
Motorcycle	18.11462	17.96316	17.86835	18.13126	20.30191	18.94324	20.10659	19.51334	17.34773	19.7692
MotorcycleE	18.25178	18.20584	17.63625	18.00913	20.26992	18.88366	19.9789	19.47956	16.8263	19.6648
Piano	21.8518	23.35054	14.67729	14.61034	22.83847	14.65069	22.48261	21.80654	16.57978	19.54981
PianoL	8.59595	16.47541	16.19664	14.90191	21.01406	13.71234	18.8214	14.76416	15.37572	17.88242
Pipes	16.56657	17.72871	17.32472	17.86993	19.64706	20.63705	18.84818	18.52577	16.94395	19.42738
Playroom	17.33568	20.85915	16.56368	16.55452	20.30064	19.57448	19.90555	19.81452	16.44838	19.96842
Playtable	20.68156	24.64304	19.65852	18.43558	21.59188	21.72184	19.71762	18.5681	17.17693	19.59922
PlaytableP	20.85154	24.66861	18.9242	18.0407	21.51447	19.92808	19.56796	19.59549	18.69339	21.67471
Recycle	21.93646	20.53941	17.21367	17.50807	19.25616	17.7585	19.26106	20.45713	17.58198	21.27788
Shelves	18.5676	20.70959	14.5391	15.53954	20.39318	14.25186	20.3385	19.47025	16.30708	19.18939
Teddy	22.24437	21.93508	13.3383	14.13207	25.67085	27.44939	24.67957	24.51732	13.69756	27.16605
Vintage	20.26597	24.5988	20.40933	20.70264	22.68876	24.27969	15.41331	21.58288	21.69535	23.15981
AVERAGE	*17.75167*	*20.31356*	*17.22726*	*17.85189*	*20.29015*	*19.20264*	*19.02797*	*18.84198*	*17.67041*	*19.98364*

Last **Middle** **First**

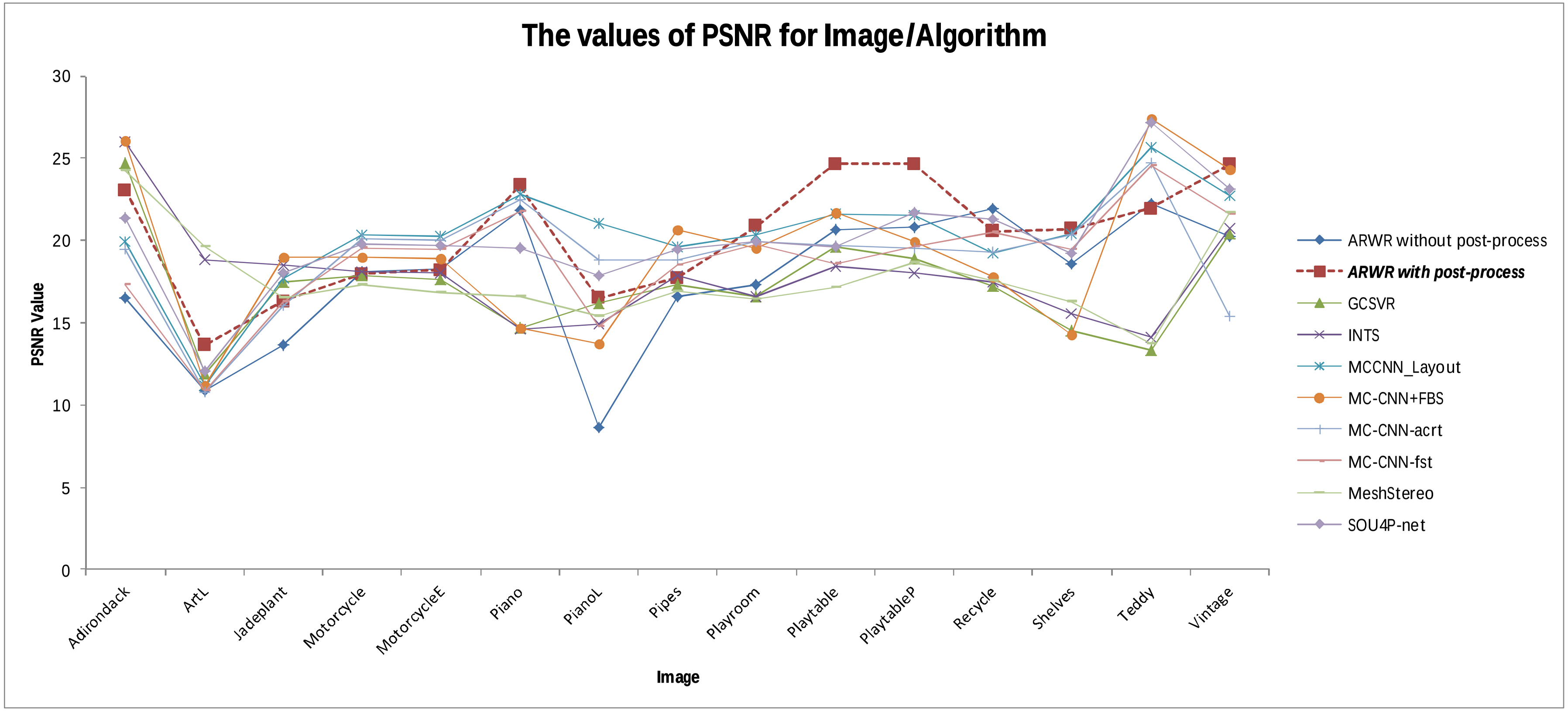


Figure 1. The values of PSNR for Image/Algorithm

Table 2. The values of MSE for Image/Algorithm

	ARWR without post-process	ARWR with post-process	GCSVR	INTS	MCCNN_Layout	MC-CNN+FBS	MC-CNN-acrt	MC-CNN-fst	MeshStereo	SOU4P-net
Adirondack	0.02251	0.00492	0.00337	0.00252	0.01005	0.00246	0.01132	0.01841	0.00374	0.00728
ArtL	0.08198	0.04322	0.06517	0.01307	0.07689	0.07678	0.08322	0.08156	0.0109	0.06227
Jadeplant	0.04291	0.02339	0.01784	0.01402	0.01679	0.01253	0.02488	0.02343	0.02243	0.01587
Motorcycle	0.01543	0.01598	0.01633	0.01537	0.00932	0.01275	0.00975	0.01118	0.01841	0.01054
MotorcycleE	0.01495	0.01511	0.01723	0.01581	0.00939	0.01293	0.01004	0.01127	0.02076	0.0108
Piano	0.00652	0.00462	0.03406	0.03459	0.0052	0.03427	0.00564	0.00659	0.02197	0.01109
PianoL	0.13816	0.02251	0.024	0.03234	0.00791	0.04253	0.01311	0.03338	0.029	0.01628
Pipes	0.02204	0.01687	0.01851	0.01633	0.01084	0.00863	0.01303	0.01404	0.02021	0.0114
Playroom	0.01846	0.0082	0.02206	0.0221	0.00933	0.01102	0.01021	0.01043	0.02265	0.01007
Playtable	0.00854	0.00343	0.01081	0.01433	0.00693	0.00672	0.01067	0.0139	0.01915	0.01096
PlaytableP	0.00822	0.00341	0.01281	0.0157	0.00705	0.01016	0.01104	0.01097	0.01351	0.0068
Recycle	0.0064	0.00883	0.01899	0.01774	0.01186	0.01675	0.01185	0.009	0.01745	0.00745
Shelves	0.0139	0.00849	0.03516	0.02792	0.00913	0.03756	0.00925	0.01129	0.0234	0.01205
Teddy	0.00596	0.0064	0.04636	0.03861	0.0027	0.00179	0.0034	0.00353	0.04268	0.00192
Vintage	0.0094	0.00346	0.0091	0.0085	0.00538	0.00373	0.02875	0.00694	0.00676	0.00483
AVERAGE	*0.02769*	*0.01259*	*0.02345*	*0.01926*	*0.01325*	*0.01937*	*0.01708*	*0.01773*	*0.01953*	*0.01331*

Last **Middle** **First**

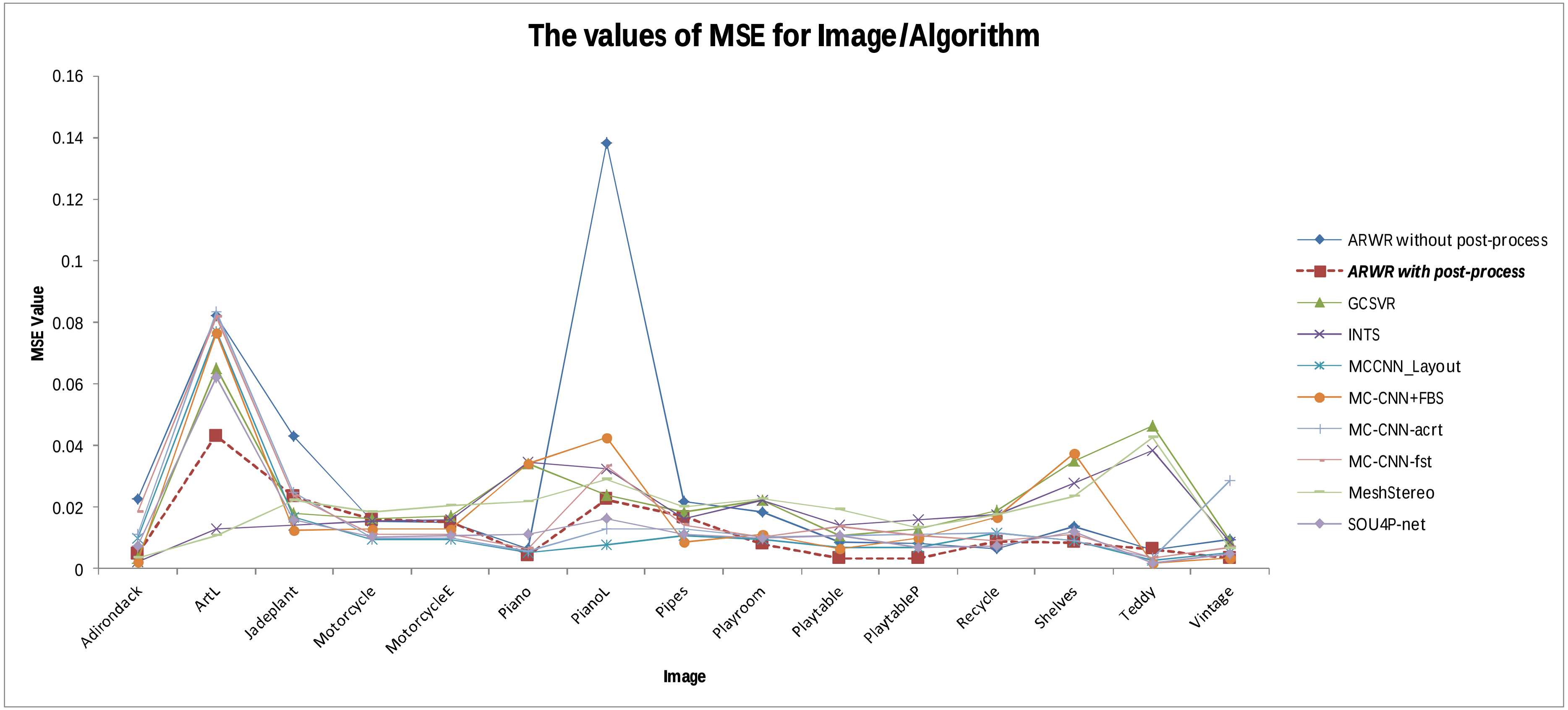


Figure 2. The values of MSE for Image/Algorithm

Table 3. The values of RMSE for Image/Algorithm

	ARWR without post-process	ARWR with post-process	GCSVR	INTS	MCCNN_Layout	MC-CNN+FBS	MC-CNN-acrt	MC-CNN-fst	MeshStereo	SOU4P-net
Adirondack	0.15006	0.07019	0.05811	0.05025	0.10027	0.04964	0.10641	0.13568	0.06117	0.08534
ArtL	0.28633	0.20789	0.25529	0.11436	0.2773	0.2771	0.28849	0.28559	0.10442	0.24954
Jadeplant	0.20715	0.15295	0.13356	0.1184	0.1296	0.11194	0.15774	0.15309	0.14978	0.126
Motorcycle	0.12424	0.12642	0.12781	0.124	0.09658	0.11293	0.09878	0.10576	0.13571	0.10269
MotorcycleE	0.12229	0.12294	0.13127	0.12576	0.09694	0.11371	0.10024	0.10617	0.1441	0.10393
Piano	0.0808	0.06799	0.18455	0.18598	0.07212	0.18512	0.07513	0.08122	0.14825	0.10531
PianoL	0.3717	0.15004	0.15494	0.17984	0.08898	0.20624	0.11453	0.18272	0.17029	0.1276
Pipes	0.14848	0.12988	0.13607	0.12779	0.10414	0.09292	0.11418	0.11849	0.14216	0.10681
Playroom	0.13589	0.09058	0.14853	0.14868	0.09659	0.10502	0.10109	0.10215	0.15051	0.10036
Playtable	0.09245	0.05859	0.104	0.11973	0.08325	0.08201	0.1033	0.11792	0.1384	0.10472
PlaytableP	0.09066	0.05842	0.11318	0.1253	0.08399	0.10083	0.10509	0.10476	0.11623	0.08246
Recycle	0.08001	0.09397	0.13782	0.13322	0.10894	0.12944	0.10887	0.09487	0.13209	0.08631
Shelves	0.11792	0.09215	0.18751	0.16711	0.09557	0.19382	0.09617	0.10628	0.15298	0.10978
Teddy	0.07722	0.08002	0.21532	0.19651	0.05205	0.04241	0.05834	0.05944	0.20659	0.04382
Vintage	0.09698	0.05889	0.09539	0.09222	0.07337	0.06109	0.16956	0.08334	0.08226	0.0695
AVERAGE	*0.14548*	*0.10406*	*0.14556*	*0.13394*	*0.10398*	*0.12428*	*0.11986*	*0.1225*	*0.13566*	*0.10694*

Last **Middle** **First**

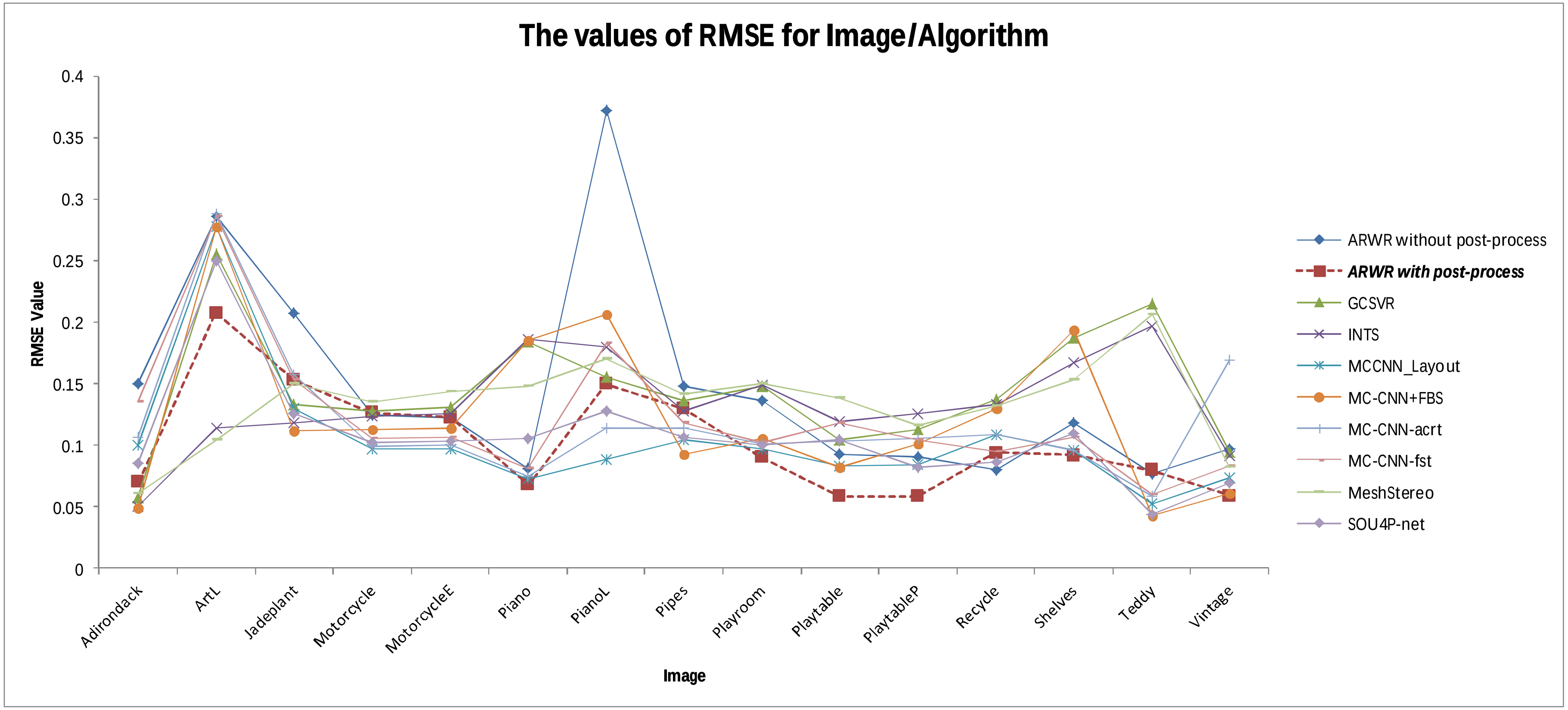


Figure 3. The values of RMSE for Image/Algorithm

Table 4. The values of SNR for Image/Algorithm

	ARWR without post-process	ARWR with post-process	GCSVR	INTS	MCCNN_Layout	MC-CNN+FBS	MC-CNN-acrt	MC-CNN-fst	MeshStereo	SOU4P-net
Adirondack	10.20299	16.80287	18.44227	19.70472	13.70512	19.81034	13.18891	11.07777	17.99658	15.10487
ArtL	4.40799	7.18837	5.40464	12.37963	4.68625	4.69262	4.34259	4.4303	13.16885	5.60222
Jadeplant	8.61219	11.24684	12.42398	13.4704	12.68559	13.95786	10.97887	11.23868	11.42857	12.93043
Motorcycle	14.59278	14.44132	14.34651	14.60942	16.78007	15.4214	16.58475	15.9915	13.82589	16.24736
MotorcycleE	14.72994	14.684	14.11441	14.48729	16.74808	15.36182	16.45706	15.95772	13.30446	16.14297
Piano	16.46209	17.96084	9.28758	9.22064	17.44876	9.26098	17.09291	16.41683	11.19007	14.16011
PianoL	3.20625	11.08571	10.80694	9.51221	15.62435	8.32264	13.4317	9.37446	9.98602	12.49271
Pipes	10.58006	11.7422	11.33821	11.88342	13.66055	14.65054	12.86167	12.53926	10.95744	13.44086
Playroom	11.51384	15.03731	10.74184	10.73268	14.4788	13.75265	14.08371	13.99268	10.62654	14.14658
Playtable	16.54147	20.50294	15.51842	14.29549	17.45179	17.58174	15.57753	14.428	13.03683	15.45913
PlaytableP	16.76749	20.58456	14.84016	13.95666	17.43043	15.84404	15.48392	15.51144	14.60935	17.59067
Recycle	17.17228	15.77523	12.4495	12.7439	14.49198	12.99433	14.49689	15.69295	12.81781	16.51371
Shelves	14.87392	17.01591	10.84541	11.84586	16.6995	10.55818	16.64482	15.77657	12.6134	15.4957
Teddy	17.36426	17.05496	8.45818	9.25195	20.79073	22.56927	19.79945	19.6372	8.81744	22.28593
Vintage	15.73185	20.06468	15.87521	16.16853	18.15464	19.74557	10.8792	17.04877	17.16123	18.62569
AVERAGE	*12.85063*	*15.41252*	*12.32622*	*12.95085*	*15.38911*	*14.3016*	*14.12693*	*13.94094*	*12.76936*	*15.0826*

Last **Middle** **First**

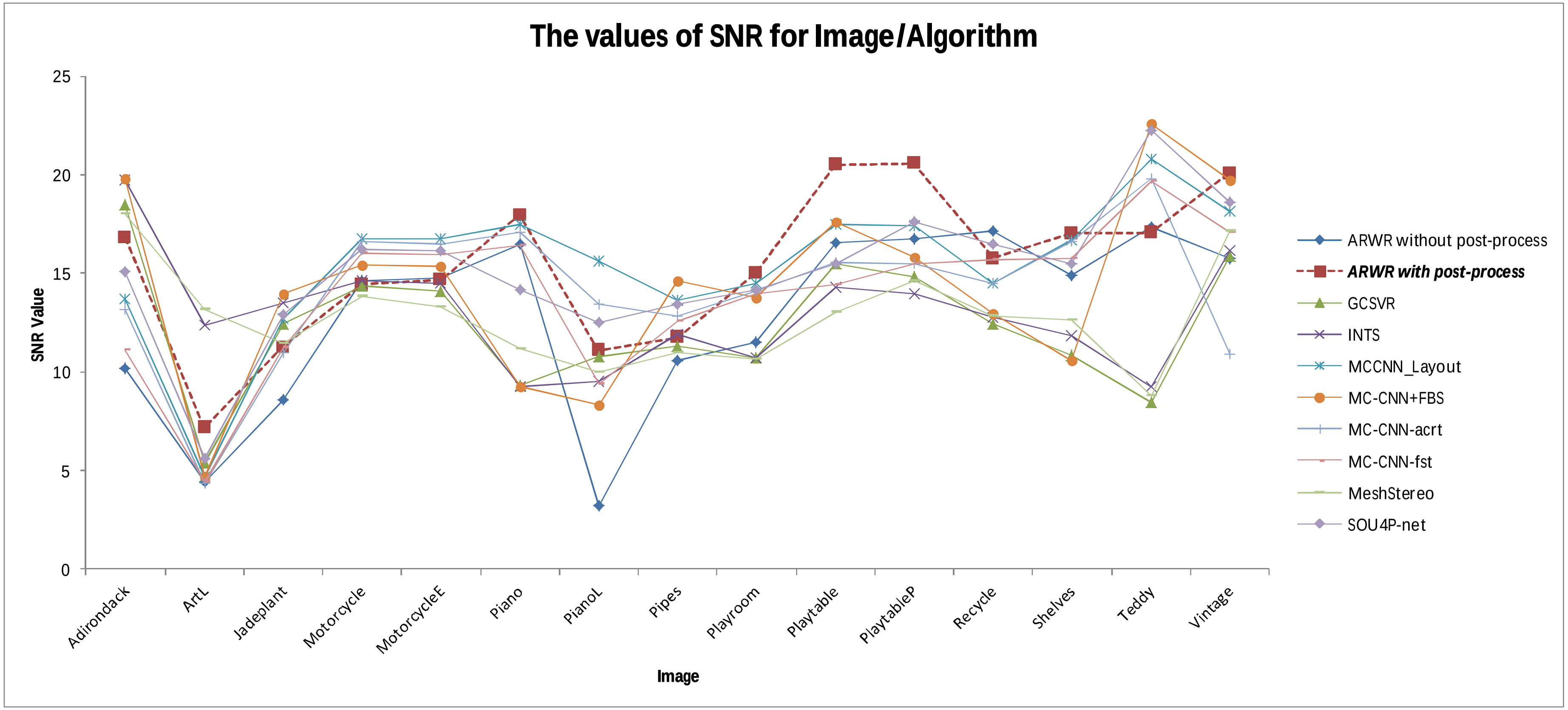


Figure 4. The values of SNR for Image/Algorithm

Table 5. The values of MAE for Image/Algorithm

	ARWR without post-process	ARWR with post-process	GCSVR	INTS	MCCNN_Layout	MC-CNN+FBS	MC-CNN-acrt	MC-CNN-fst	MeshStereo	SOU4P-net
Adirondack	0.11968	0.04489	0.01434	0.01336	0.06557	0.01167	0.06831	0.09021	0.01499	0.03988
ArtL	0.23538	0.16063	0.21648	0.03645	0.23468	0.23491	0.24013	0.23915	0.03494	0.20993
Jadeplant	0.11998	0.09069	0.06368	0.05261	0.06559	0.05191	0.08158	0.08323	0.07529	0.05516
Motorcycle	0.04314	0.0677	0.0887	0.08716	0.0417	0.07875	0.03694	0.03196	0.09212	0.05875
MotorcycleE	0.04247	0.06333	0.09072	0.08675	0.04215	0.07912	0.03946	0.03272	0.09593	0.06028
Piano	0.03205	0.03883	0.17002	0.17214	0.03158	0.17129	0.03389	0.03684	0.13441	0.08404
PianoL	0.34548	0.08117	0.12729	0.15595	0.04248	0.17555	0.06107	0.15168	0.14996	0.10547
Pipes	0.06524	0.06966	0.11144	0.10196	0.03684	0.02998	0.03651	0.0428	0.11222	0.07147
Playroom	0.05314	0.0448	0.124	0.12455	0.0285	0.07222	0.03022	0.03341	0.123	0.06658
Playtable	0.03563	0.03643	0.07765	0.10773	0.02657	0.05979	0.03862	0.06292	0.11786	0.083
PlaytableP	0.03555	0.03596	0.09102	0.11002	0.02624	0.08058	0.03982	0.0533	0.10045	0.06922
Recycle	0.02914	0.08	0.11836	0.11864	0.04801	0.11287	0.04675	0.0348	0.11436	0.06608
Shelves	0.05452	0.05616	0.13038	0.13473	0.04307	0.16252	0.04235	0.04466	0.10873	0.06697
Teddy	0.05988	0.07192	0.19403	0.16549	0.01472	0.02553	0.01244	0.01397	0.17896	0.03042
Vintage	0.02895	0.02344	0.06189	0.07096	0.03783	0.02649	0.15104	0.04784	0.06193	0.04043
AVERAGE	*0.08668*	*0.06437*	*0.112*	*0.10257*	*0.05237*	*0.09154*	*0.06394*	*0.06663*	*0.10101*	*0.07385*

Last **Middle** **First**

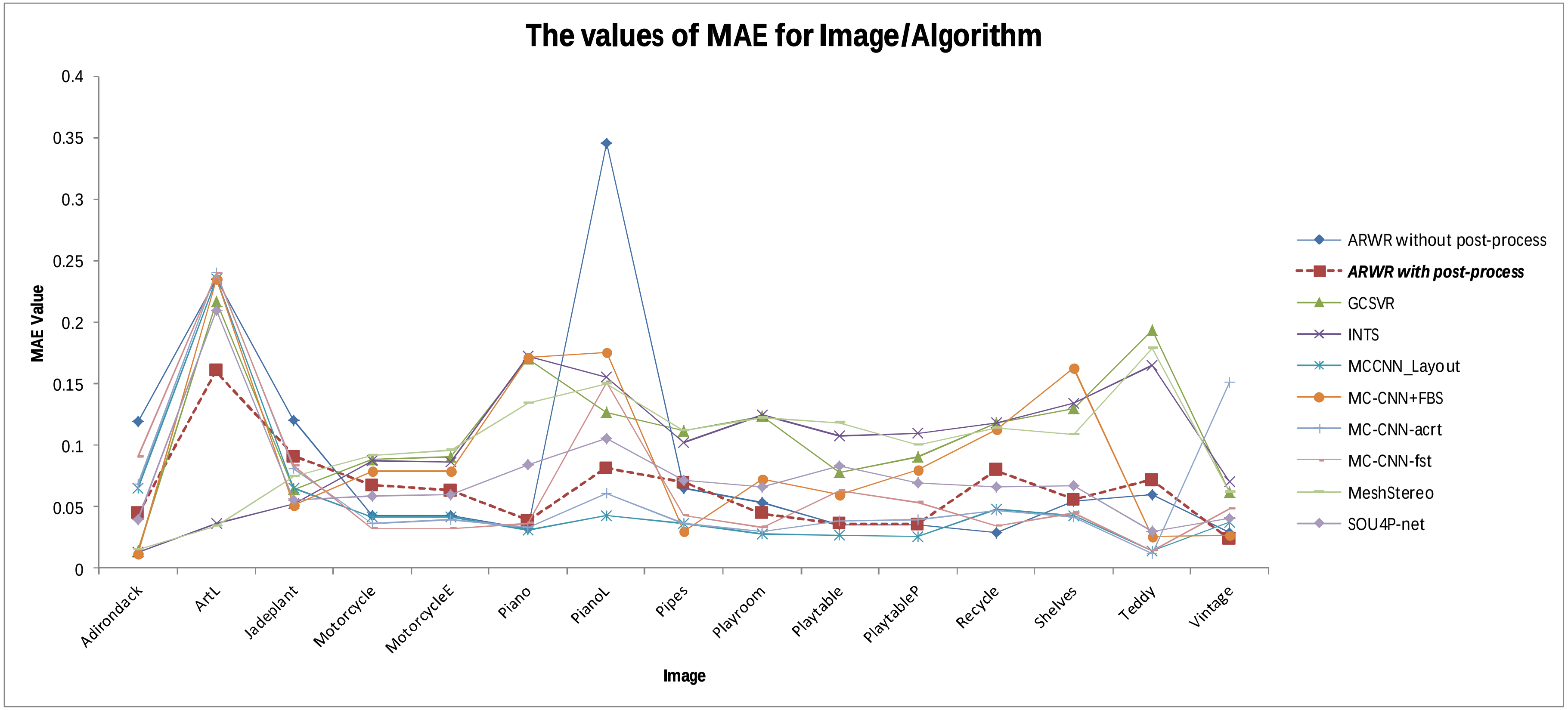


Figure 5. The values of MAE for Image/Algorithm

Table 6. The values of SSIM for Image/Algorithm

	ARWR without post-process	ARWR with post-process	GCSVR	INTS	MCCNN_Layout	MC-CNN+FBS	MC-CNN-acrt	MC-CNN-fst	MeshStereo	SOU4P-net
Adirondack	0.997	0.99942	0.99963	0.99976	0.99871	0.99976	0.99858	0.99761	0.9996	0.99913
ArtL	0.9889	0.99419	0.99088	0.99857	0.98924	0.98918	0.98849	0.98864	0.99883	0.99127
Jadeplant	0.99509	0.9972	0.99812	0.99867	0.99832	0.99882	0.9977	0.99767	0.9976	0.99844
Motorcycle	0.99835	0.99814	0.99806	0.99819	0.9991	0.99855	0.99911	0.99896	0.99781	0.99891
MotorcycleE	0.99843	0.99826	0.99796	0.99814	0.9991	0.99853	0.99907	0.99894	0.99749	0.99888
Piano	0.99947	0.99957	0.99529	0.99523	0.99959	0.99523	0.99956	0.99948	0.99709	0.9988
PianoL	0.98072	0.99722	0.99696	0.99571	0.99926	0.9942	0.99863	0.99608	0.99608	0.99814
Pipes	0.99757	0.99799	0.99771	0.99807	0.99894	0.99917	0.99878	0.99864	0.99754	0.99878
Playroom	0.99828	0.99926	0.9973	0.99729	0.99929	0.99898	0.99923	0.99923	0.99726	0.99913
Playtable	0.99912	0.9996	0.99867	0.99812	0.99925	0.99917	0.99884	0.99849	0.99741	0.99878
PlaytableP	0.99918	0.9996	0.99827	0.99787	0.99922	0.99867	0.9988	0.99882	0.99817	0.99921
Recycle	0.99941	0.99885	0.99746	0.99761	0.99859	0.99779	0.99863	0.99906	0.99767	0.99913
Shelves	0.99883	0.99926	0.99568	0.99666	0.99937	0.99521	0.99939	0.9992	0.99751	0.99901
Teddy	0.9993	0.99916	0.99315	0.99431	0.99966	0.99978	0.99964	0.99959	0.99368	0.99976
Vintage	0.99908	0.99972	0.99899	0.99909	0.9996	0.99975	0.99648	0.99942	0.99931	0.99962
AVERAGE	*0.99658*	*0.9985*	*0.99694*	*0.99755*	*0.99848*	*0.99752*	*0.99806*	*0.99799*	*0.99754*	*0.99847*

Last Middle First

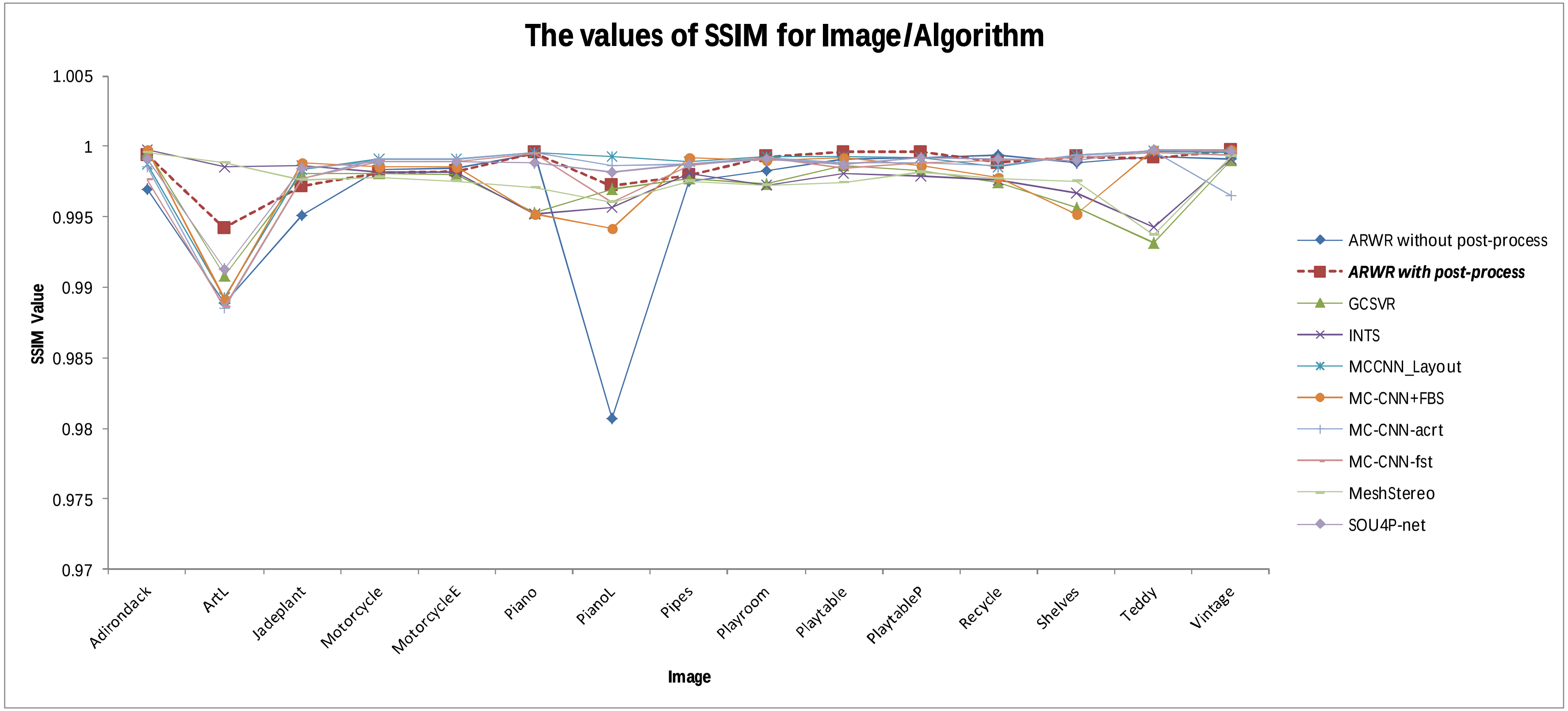


Figure 6. The values of SSIM for Image/Algorithm

Table 7. The values of DSSIM for Image/Algorithm

	ARWR without post-process	ARWR with post-process	GCSVR	INTS	MCCNN_Layout	MC-CNN+FBS	MC-CNN-acrt	MC-CNN-fst	MeshStereo	SOU4P-net
Adirondack	0.00149	0.00028	0.00018	0.00011	0.00064	0.00011	0.0007	0.00119	0.00074	0.00043
ArtL	0.00554	0.0029	0.00455	0.00071	0.00537	0.0054	0.00575	0.00567	0.00046	0.00436
Jadeplant	0.00245	0.00139	0.00093	0.00066	0.00083	0.00058	0.00114	0.00116	0.00106	0.00077
Motorcycle	0.00082	0.00092	0.00096	0.0009	0.00044	0.00072	0.00044	0.00051	0.00066	0.00054
MotorcycleE	0.00078	0.00086	0.00101	0.00092	0.00044	0.00073	0.00046	0.00052	0.00073	0.00055
Piano	0.00026	0.00021	0.00235	0.00238	0.0002	0.00238	0.00021	0.00025	0.0006	0.00059
PianoL	0.00963	0.00138	0.00151	0.00214	0.00036	0.00289	0.00068	0.00195	0.00128	0.00092
Pipes	0.00121	0.001	0.00114	0.00096	0.00052	0.00041	0.0006	0.00067	0.00109	0.0006
Playroom	0.00085	0.00036	0.00134	0.00135	0.00035	0.0005	0.00038	0.00038	0.00122	0.00043
Playtable	0.00043	0.00019	0.00066	0.00093	0.00037	0.00041	0.00057	0.00075	0.00073	0.0006
PlaytableP	0.0004	0.00019	0.00086	0.00106	0.00038	0.00066	0.00059	0.00058	0.00059	0.00039
Recycle	0.00029	0.00057	0.00126	0.00119	0.0007	0.0011	0.00068	0.00046	0.00113	0.00043
Shelves	0.00058	0.00036	0.00215	0.00166	0.00031	0.00239	0.0003	0.00039	0.00292	0.00049
Teddy	0.00034	0.00041	0.00342	0.00284	0.00016	0.0001	0.00017	0.0002	0.00305	0.00011
Vintage	0.00045	0.00013	0.0005	0.00045	0.00019	0.00012	0.00175	0.00028	0.00036	0.00018
AVERAGE	*0.0017*	*0.00074*	*0.00152*	*0.00122*	*0.00075*	*0.00123*	*0.00096*	*0.001*	*0.00111*	*0.00076*

Last **Middle** **First**

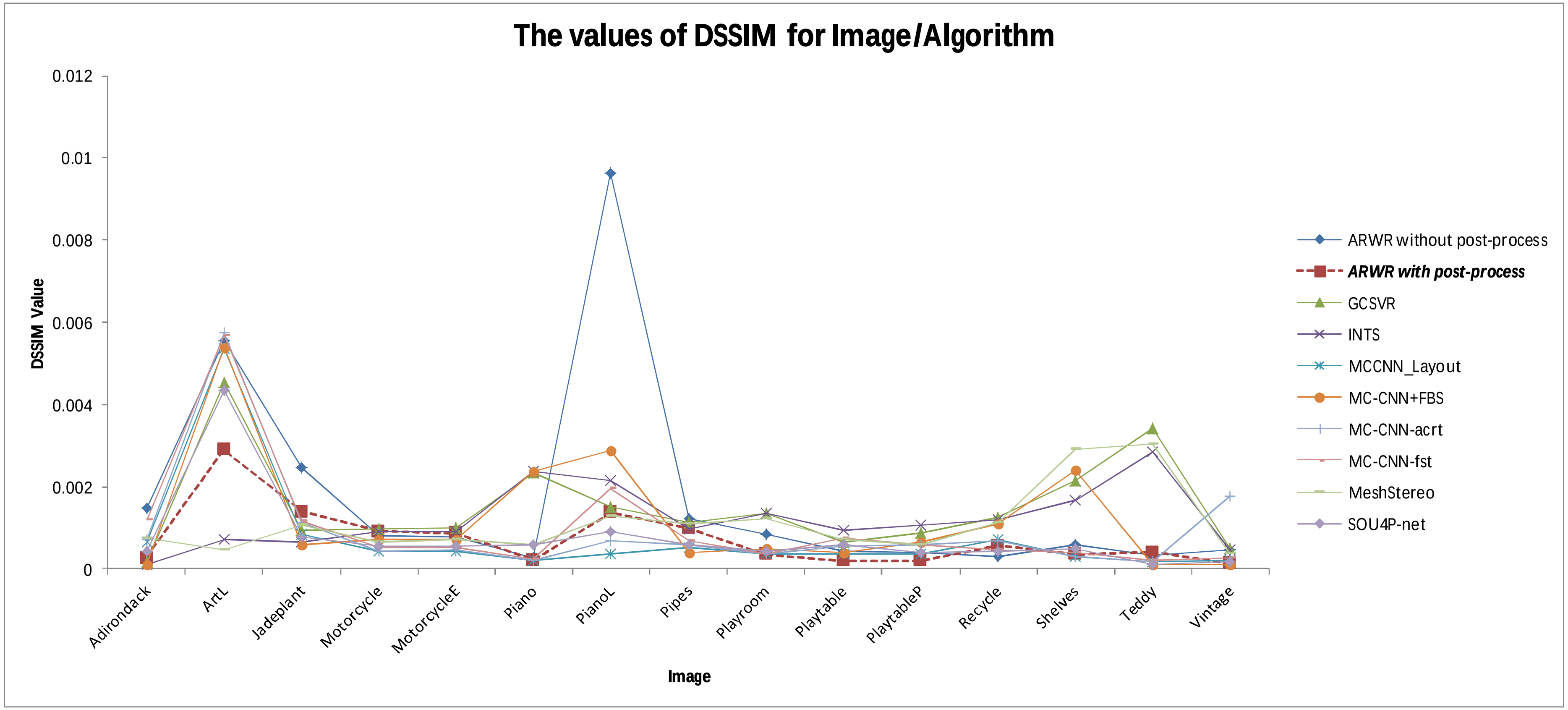
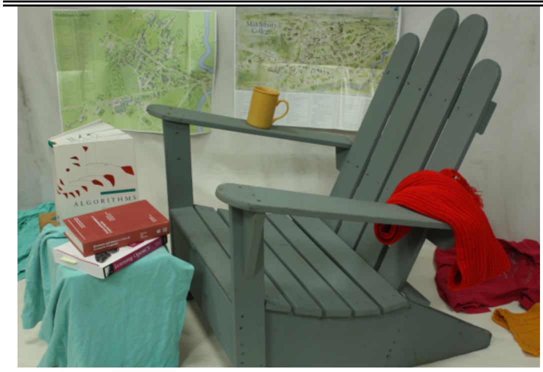
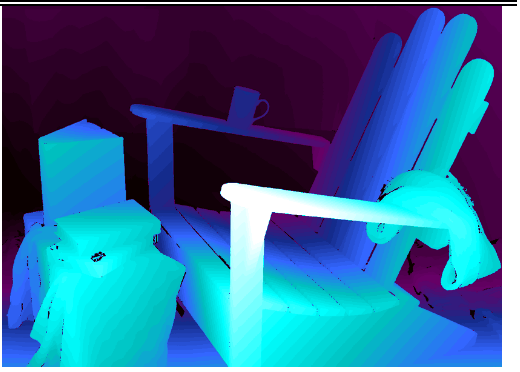
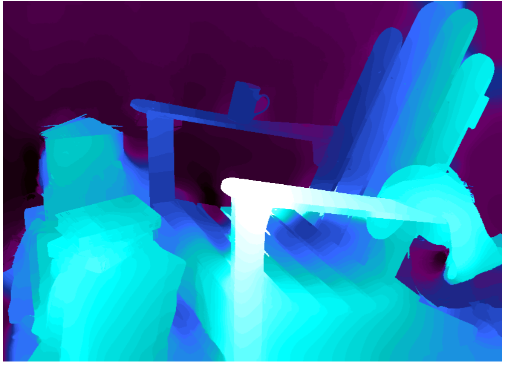
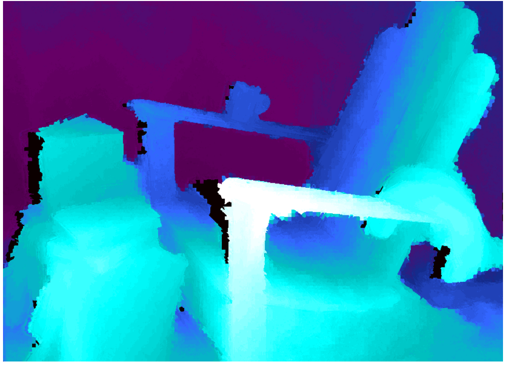
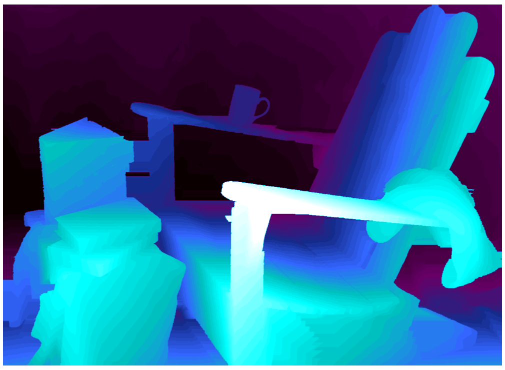
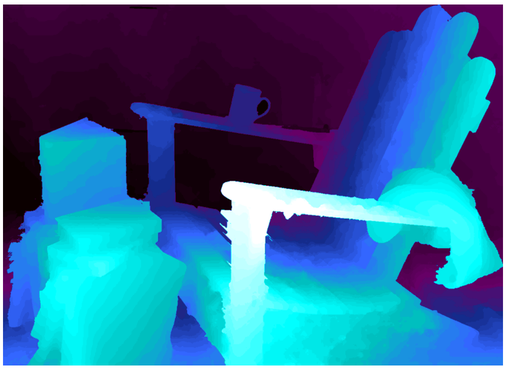
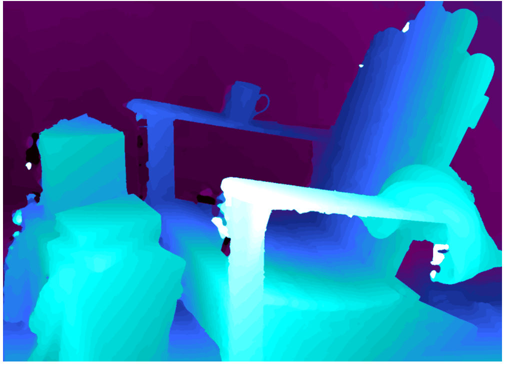
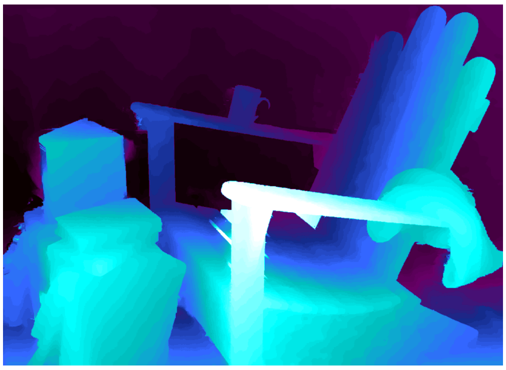
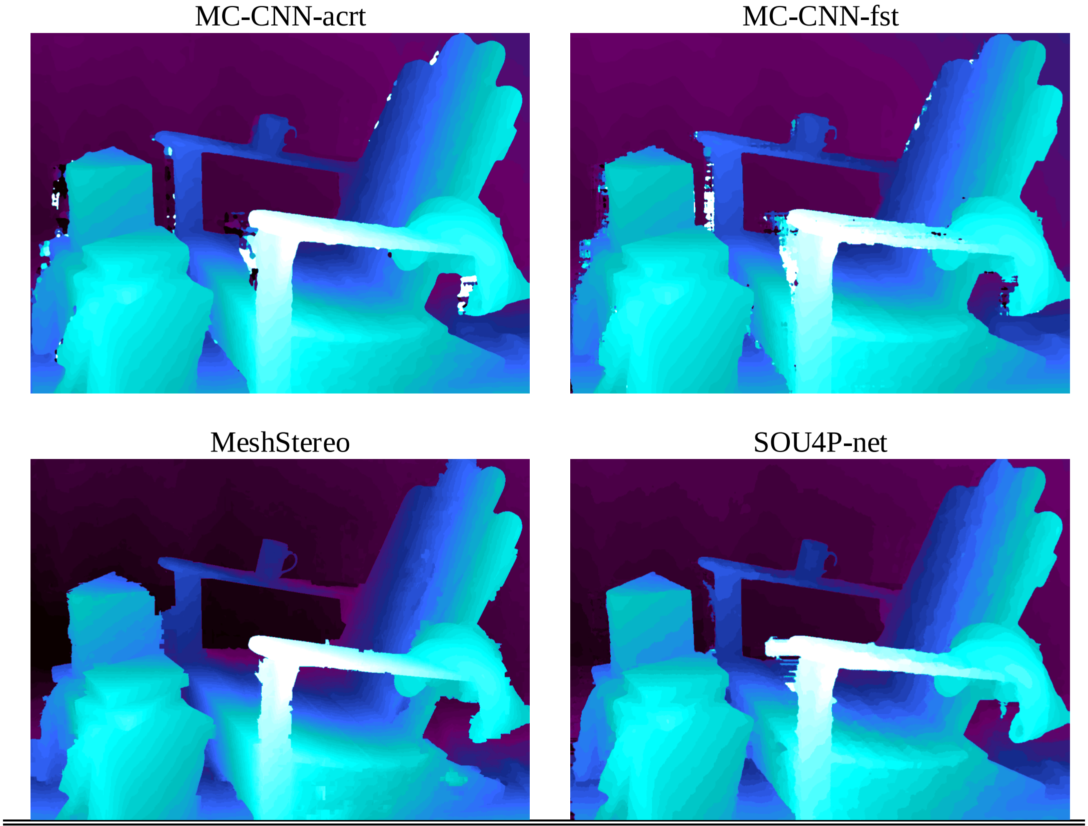


Figure 7. The values of DSSIM for Image/Algorithm

Appendix 2: This section presents extended visual results of the comparison of post-processed ARWR and other methods. These results are based on the standard images of Middlebury training dense set.

Right Image	Ground Truth
	
ARWR with Post-Processing	ARWR without Post-Processing
	
GCSVR	INTS
	
MCCNN_Layout	MC-CNN+FBS
	

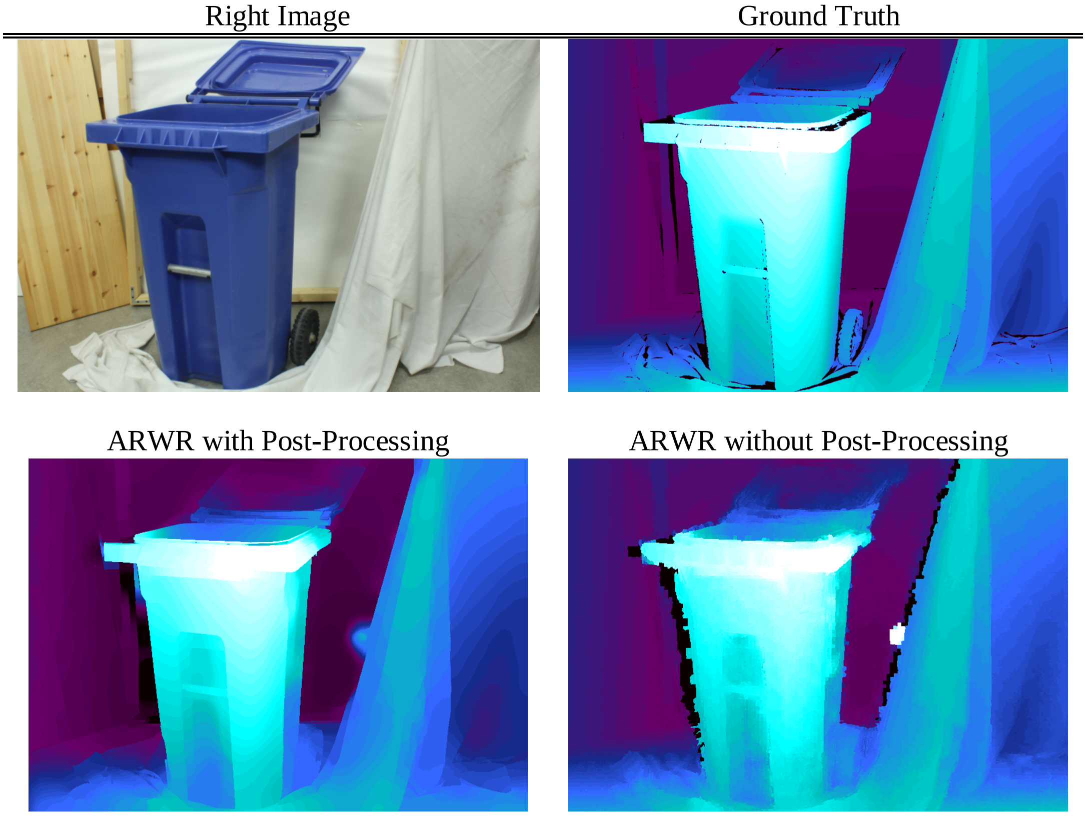


MC-CNN-acrt

MC-CNN-fst

MeshStereo

SOU4P-net



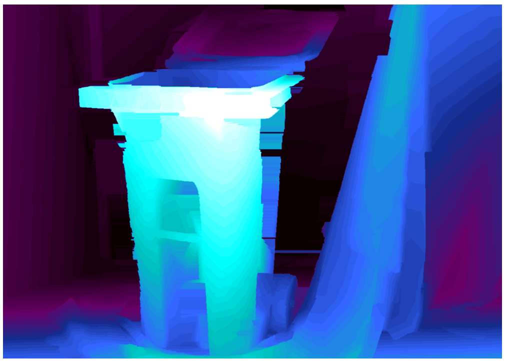
Right Image

Ground Truth

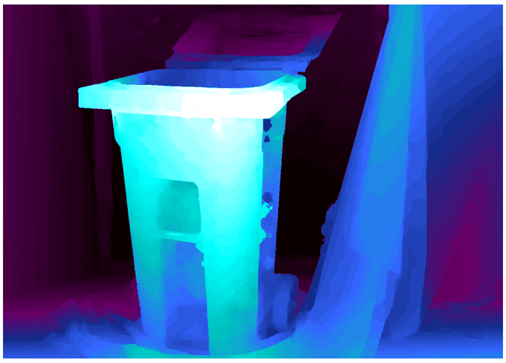
ARWR with Post-Processing

ARWR without Post-Processing

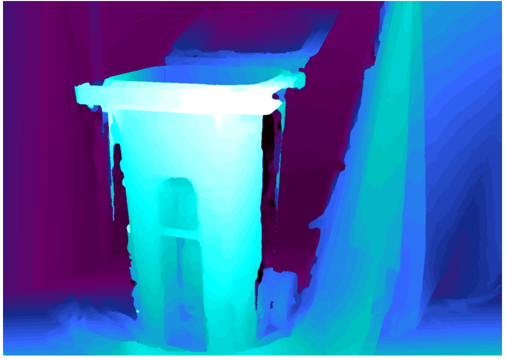
GCSVR



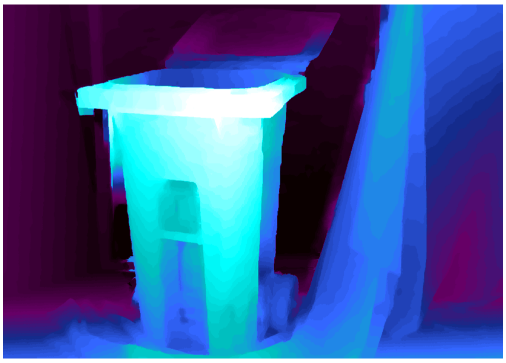
INTS



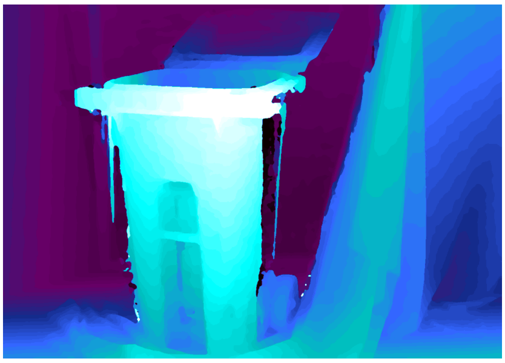
MCCNN_Layout



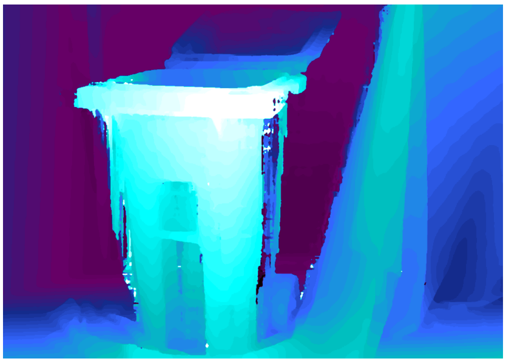
MC-CNN+FBS



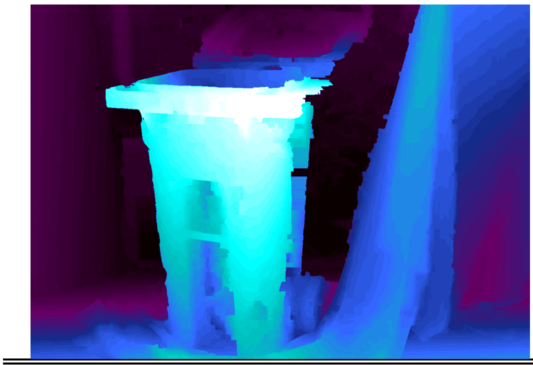
MC-CNN-acrt



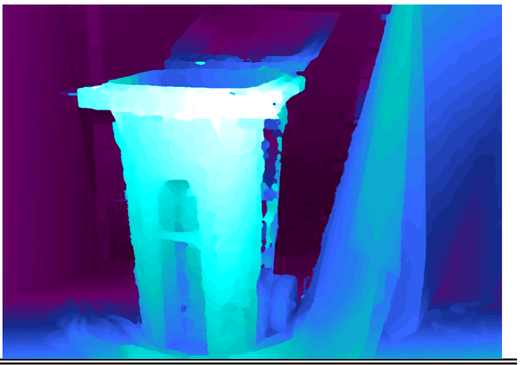
MC-CNN-fst

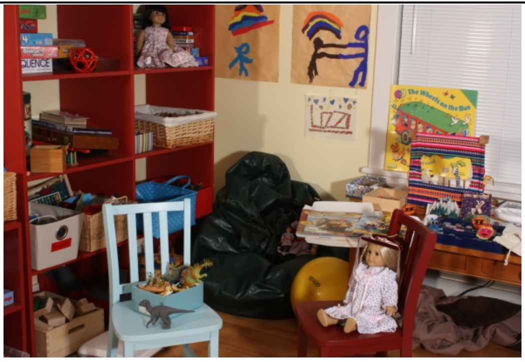
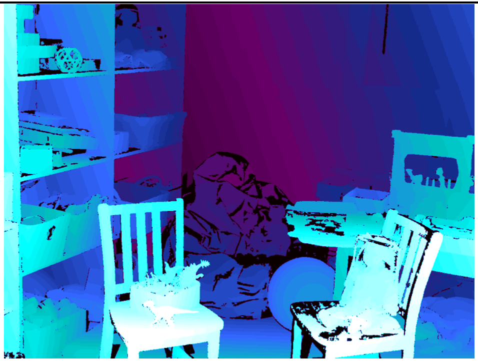
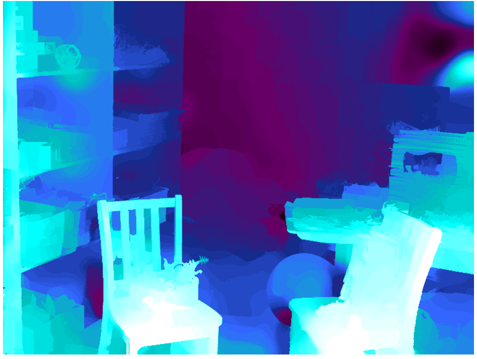
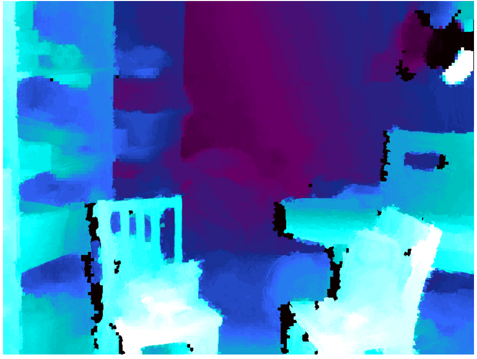
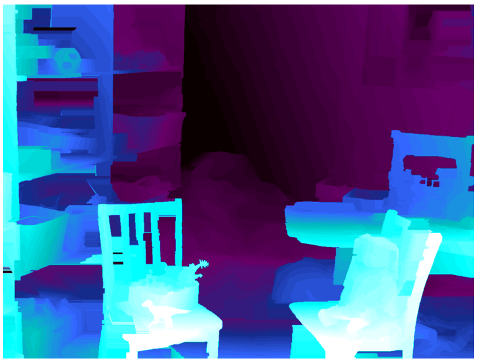
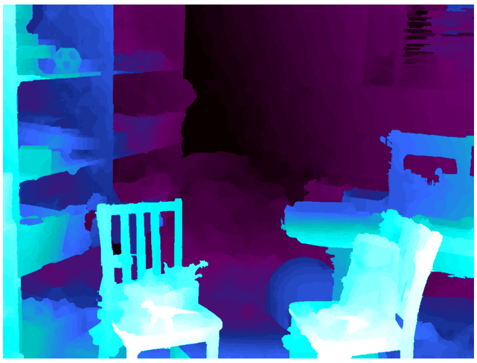
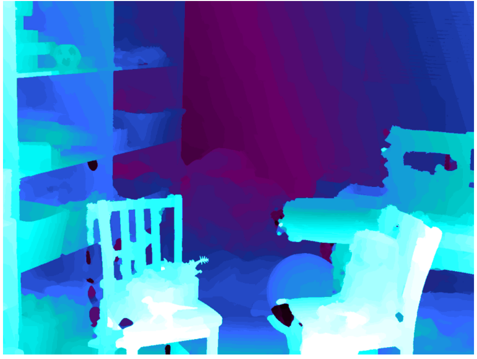
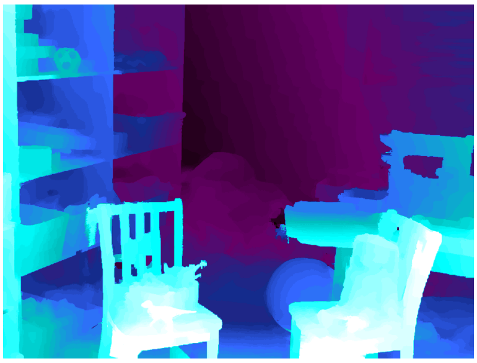


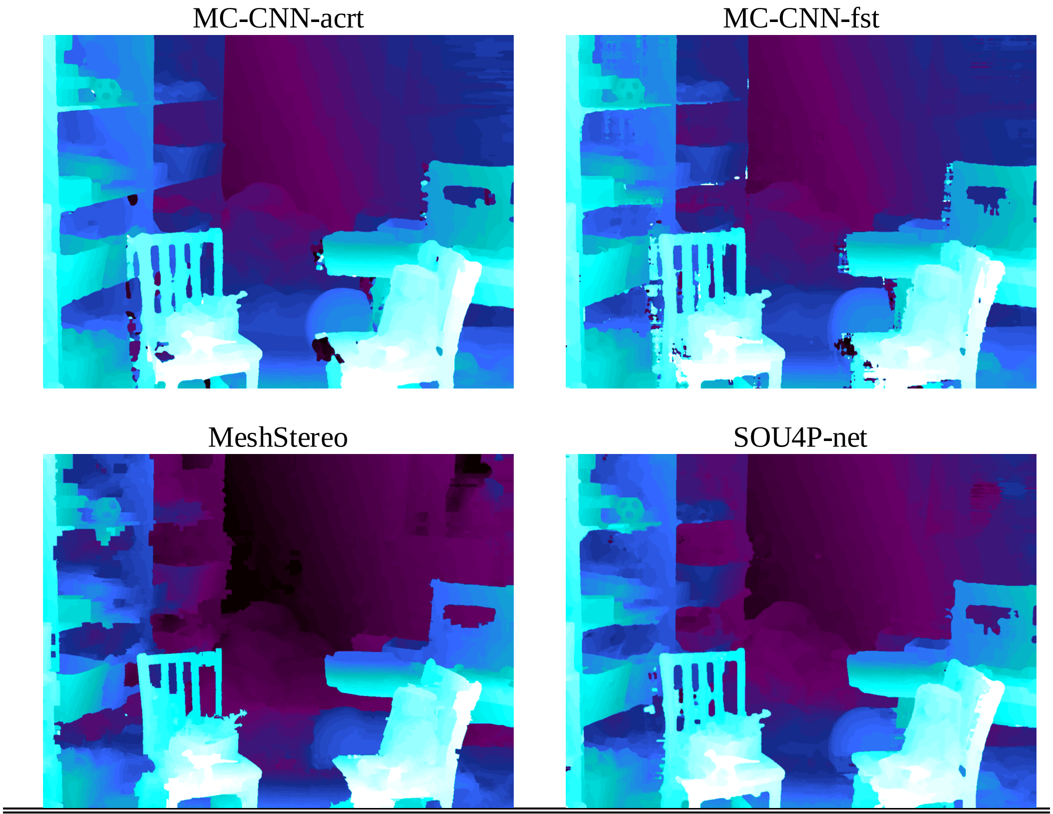
MeshStereo



SOU4P-net



Right Image	Ground Truth
	
ARWR with Post-Processing	ARWR without Post-Processing
	
GCSVR	INTS
	
MCCNN_Layout	MC-CNN+FBS
	

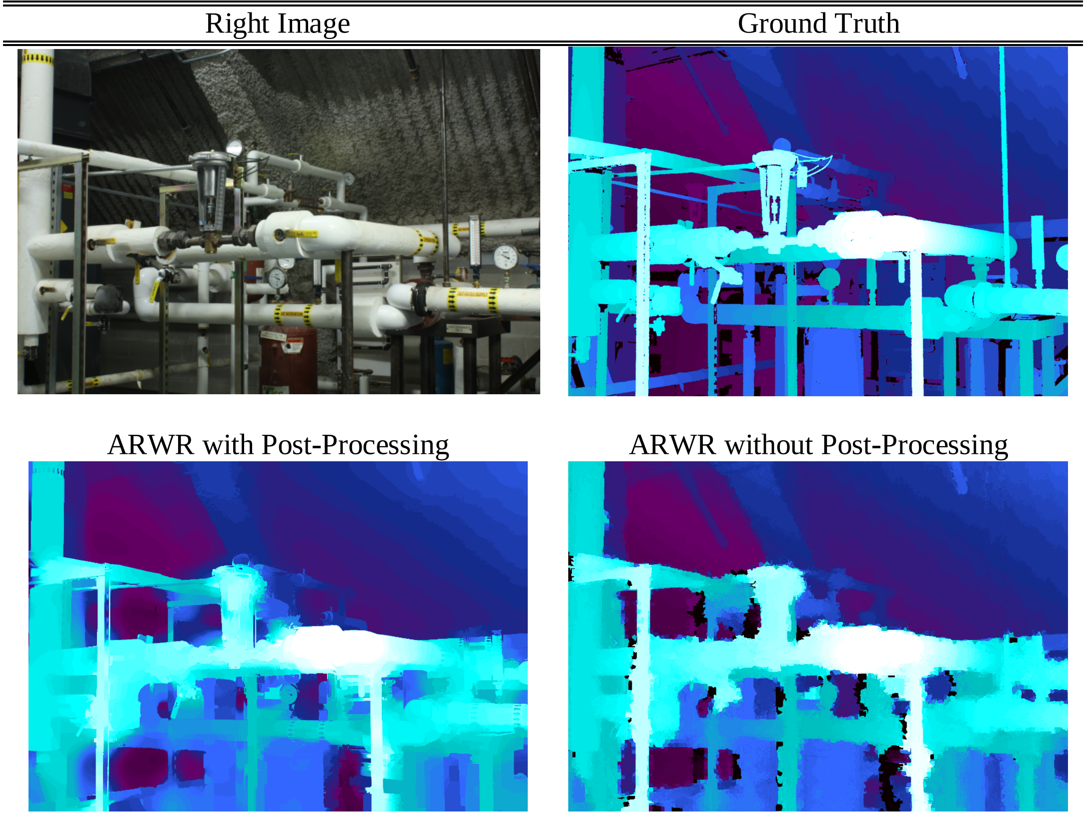


MC-CNN-acrt

MC-CNN-fst

MeshStereo

SOU4P-net



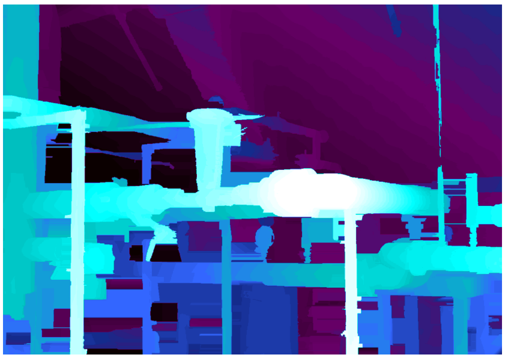
Right Image

Ground Truth

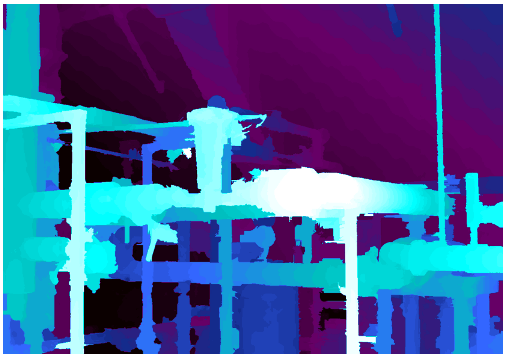
ARWR with Post-Processing

ARWR without Post-Processing

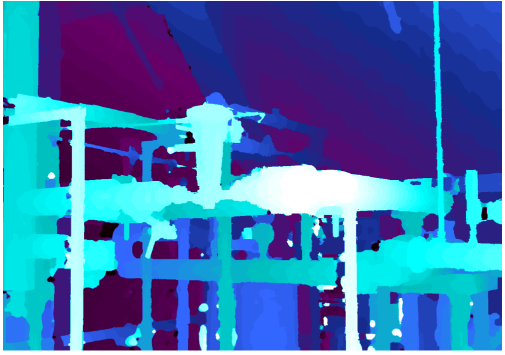
GCSVR



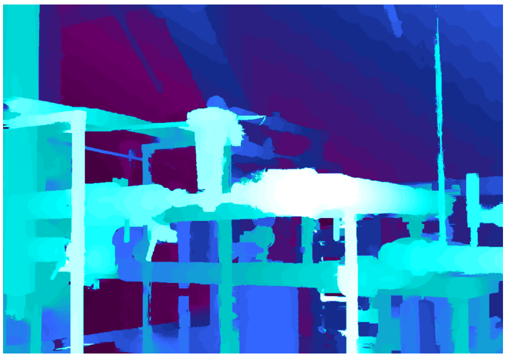
INTS



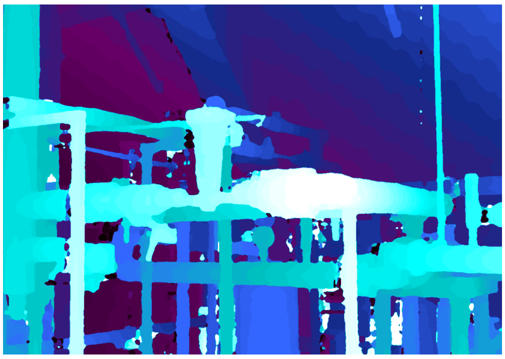
MCCNN_Layout



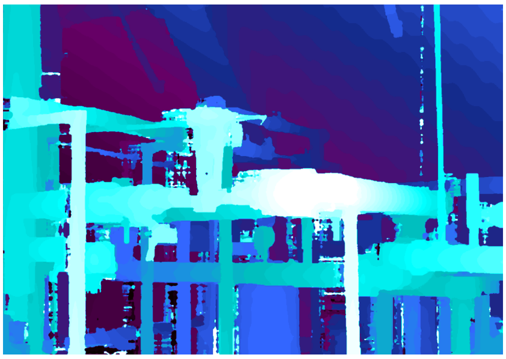
MC-CNN+FBS



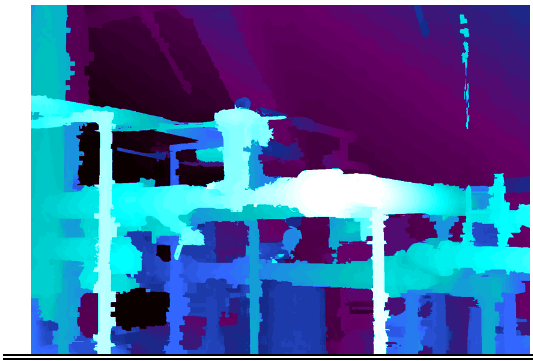
MC-CNN-acrt



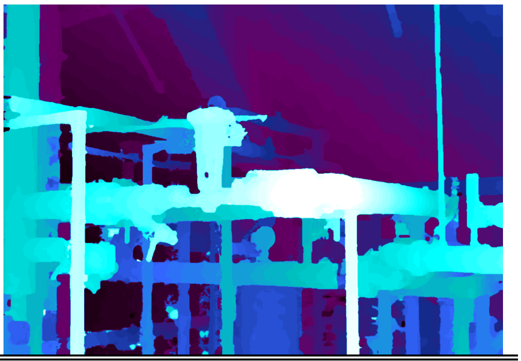
MC-CNN-fst

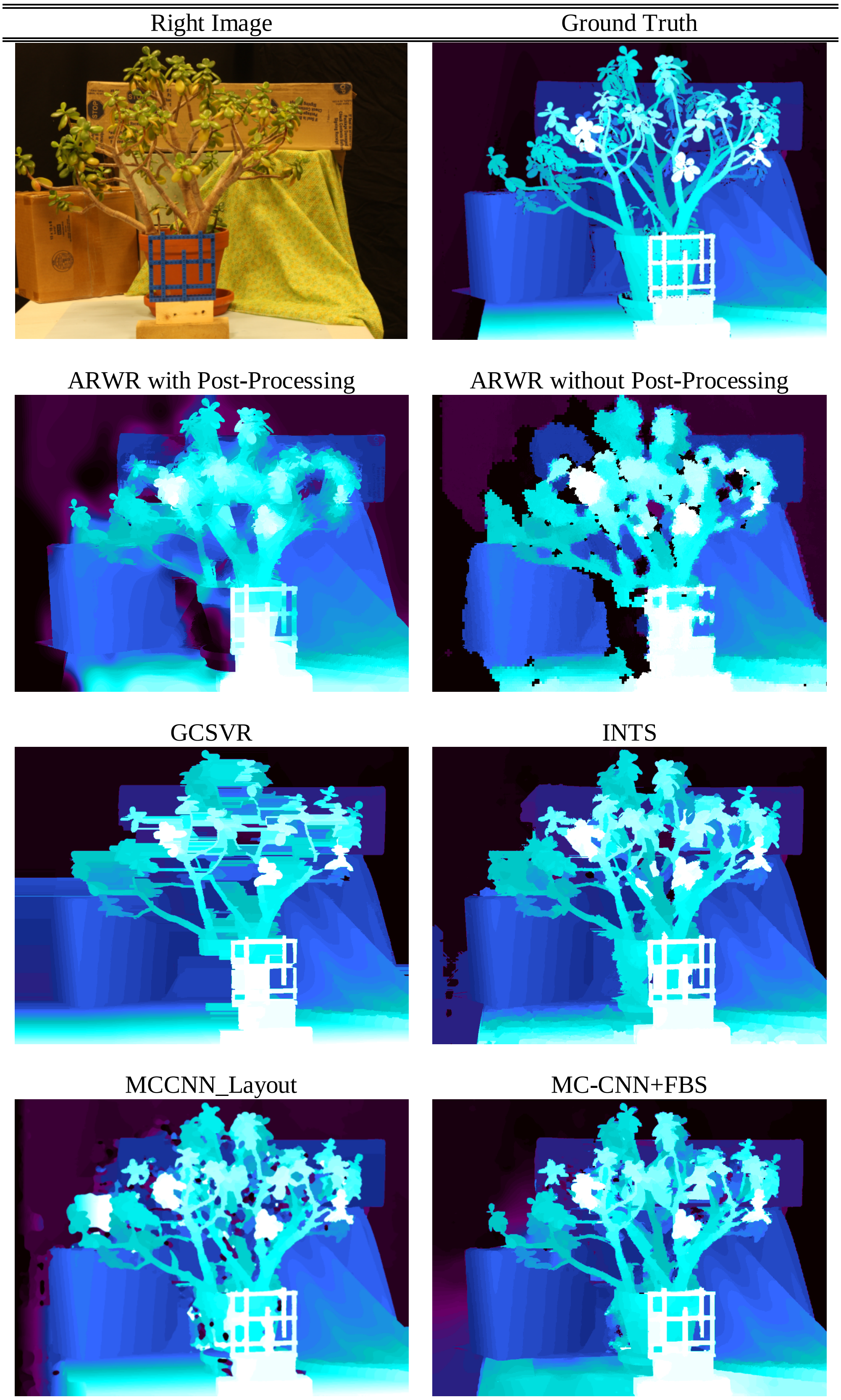


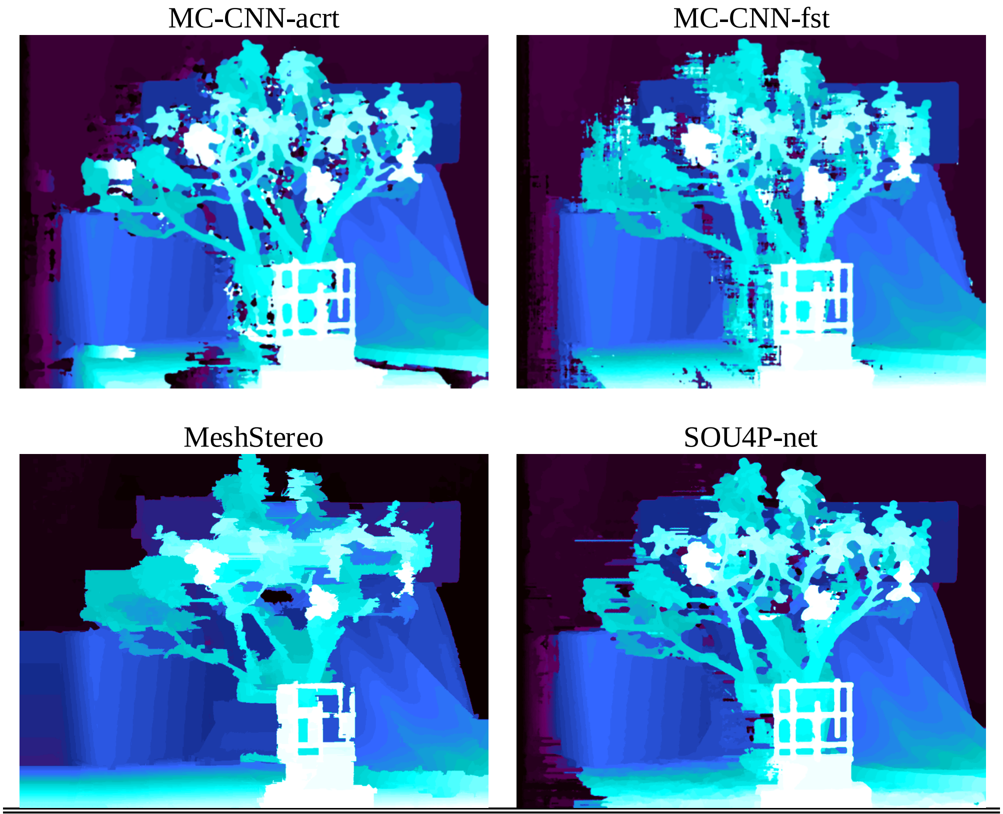
MeshStereo



SOU4P-net




Right Image
Ground Truth
ARWR with Post-Processing
ARWR without Post-Processing
GCSVR
INTS
MCCNN_Layout
MC-CNN+FBS

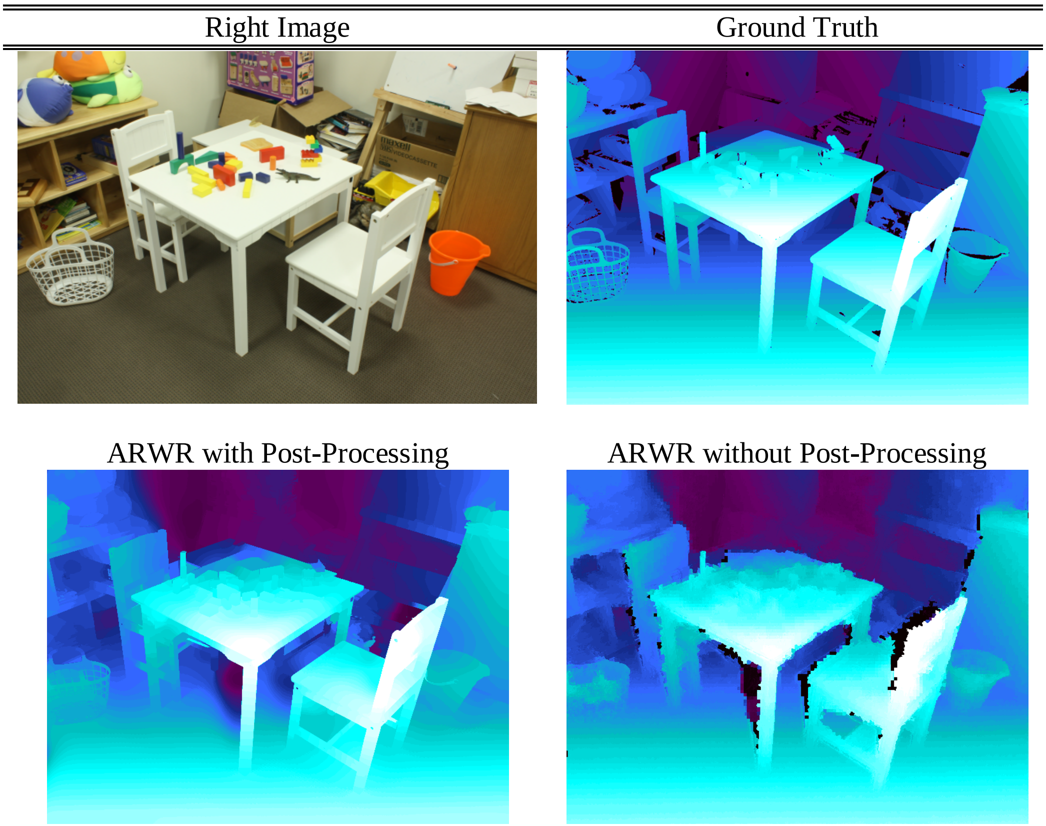


MC-CNN-acrt

MC-CNN-fst

MeshStereo

SOU4P-net



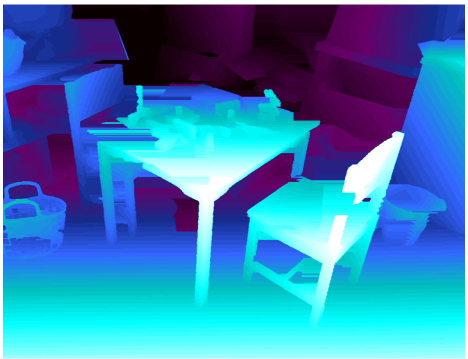
Right Image

Ground Truth

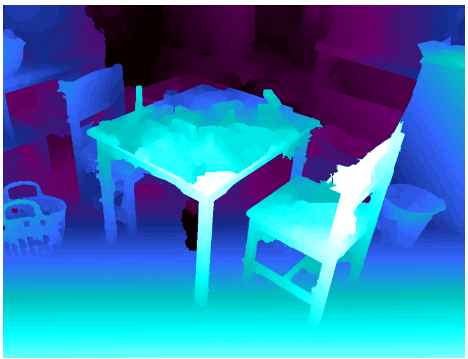
ARWR with Post-Processing

ARWR without Post-Processing

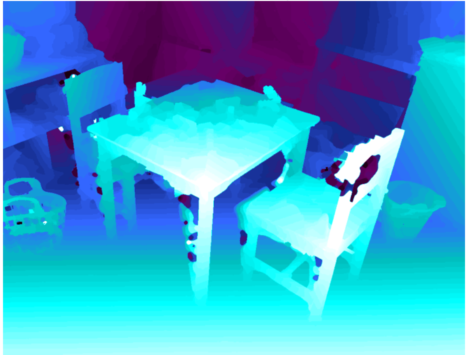
GCSVR



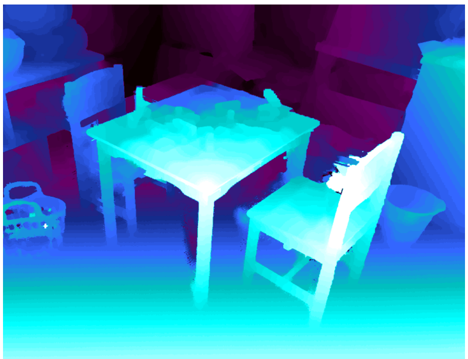
INTS



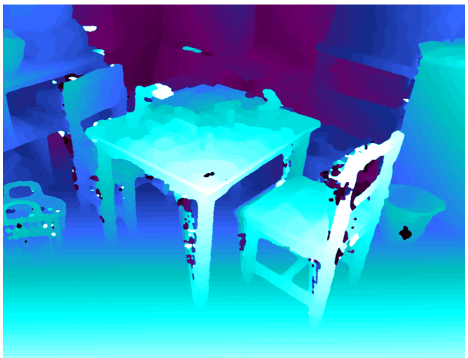
MCCNN_Layout



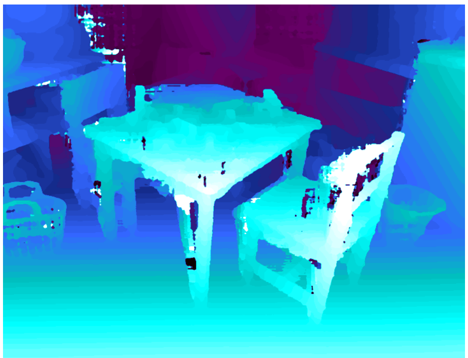
MC-CNN+FBS



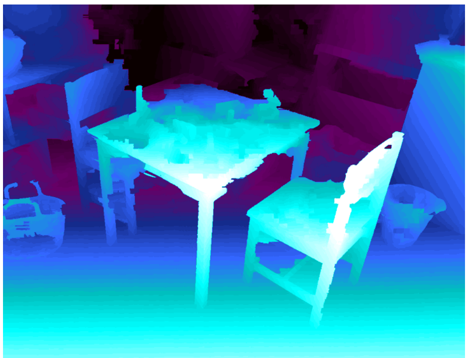
MC-CNN-acrt



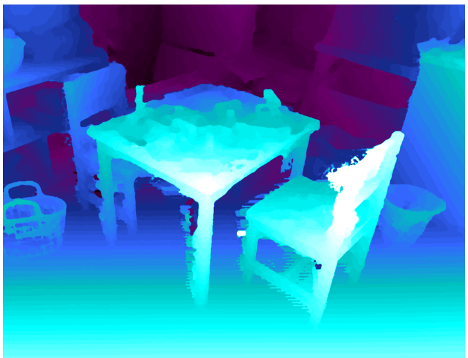
MC-CNN-fst

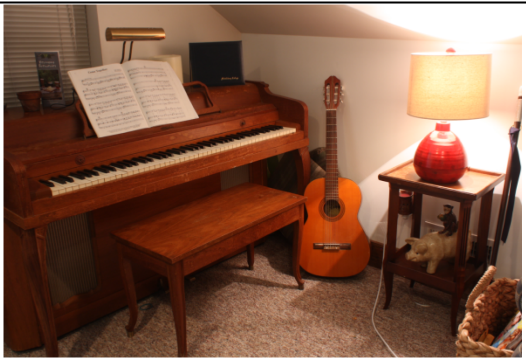
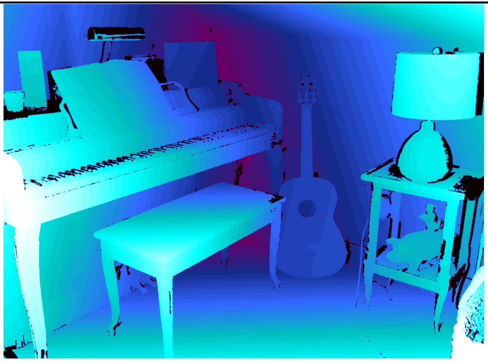
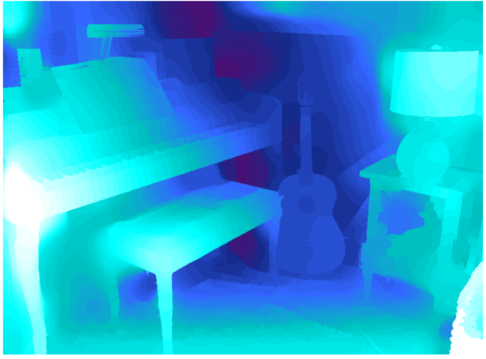
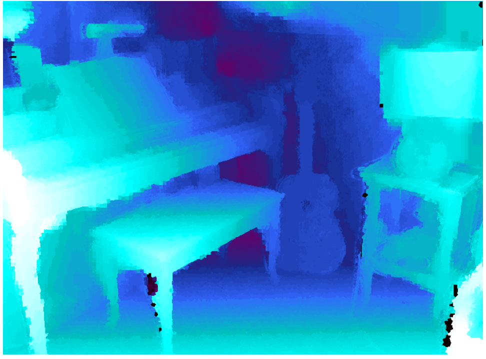
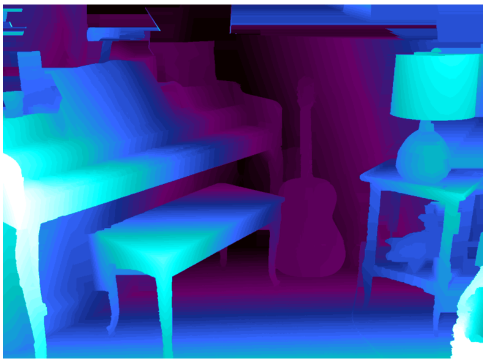
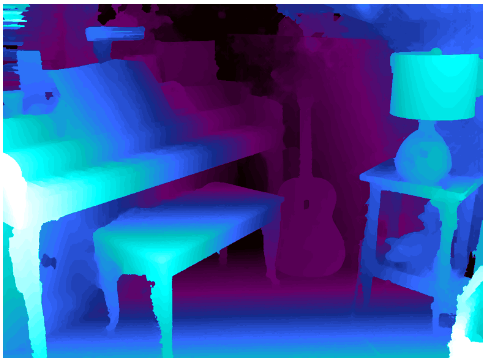
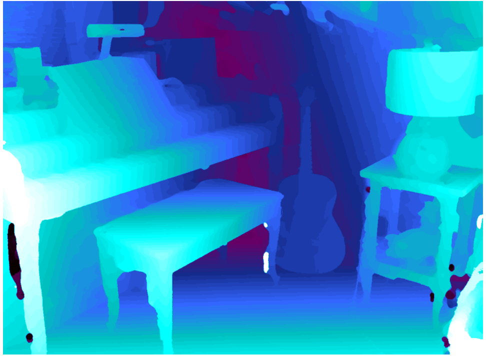
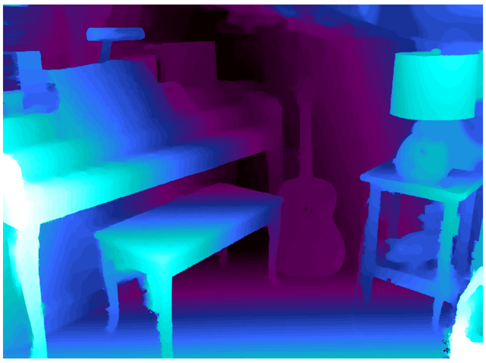


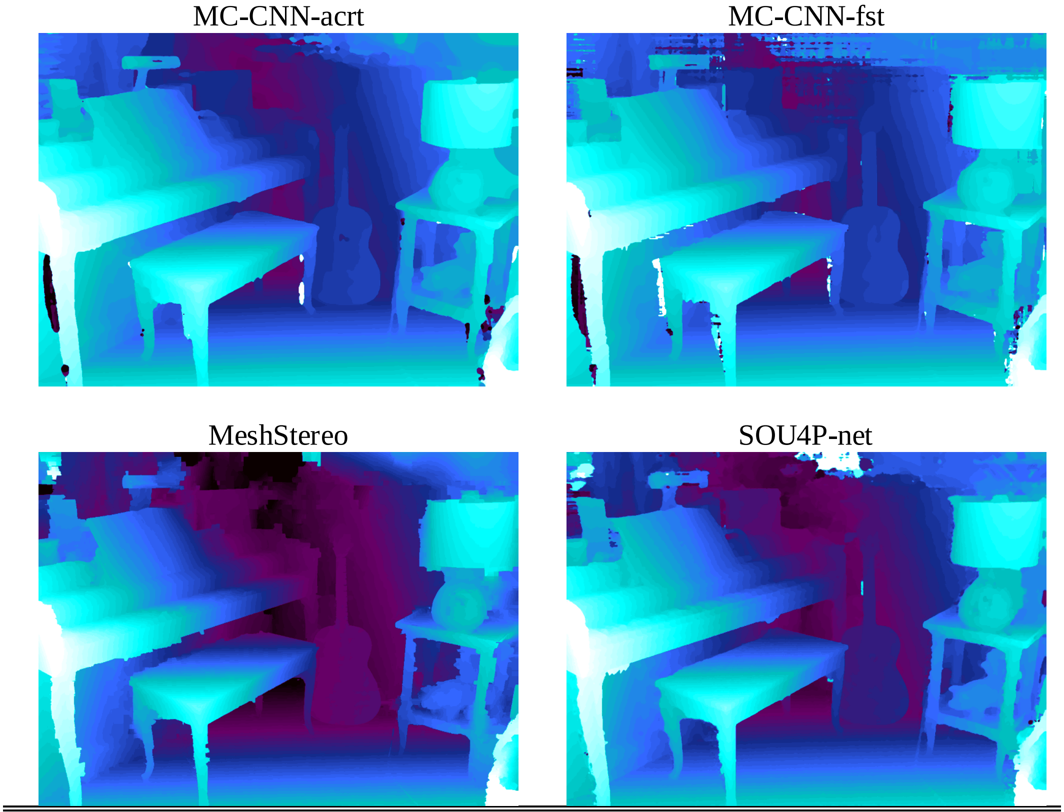
MeshStereo



SOU4P-net



Right Image	Ground Truth
	
ARWR with Post-Processing	ARWR without Post-Processing
	
GCSVR	INTS
	
MCCNN_Layout	MC-CNN+FBS
	



MC-CNN-acrt

MC-CNN-fst

MeshStereo

SOU4P-net

US 20180027224A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2018/0027224 A1**

Javidnia et al. (43) **Pub. Date: Jan. 25, 2018**

(54) **SYSTEMS AND METHODS FOR ESTIMATING AND REFINING DEPTH MAPS**

(71) Applicant: **FotoNation Limited**, Galway (IE)

(72) Inventors: **Hossein Javidnia**, Galway (IE); **Peter Corcoran**, Claregalway (IE)

(21) Appl. No.: **15/654,693**

(22) Filed: **Jul. 19, 2017**

Related U.S. Application Data

(60) Provisional application No. 62/364,263, filed on Jul. 19, 2016.

Publication Classification

(51) **Int. Cl.**

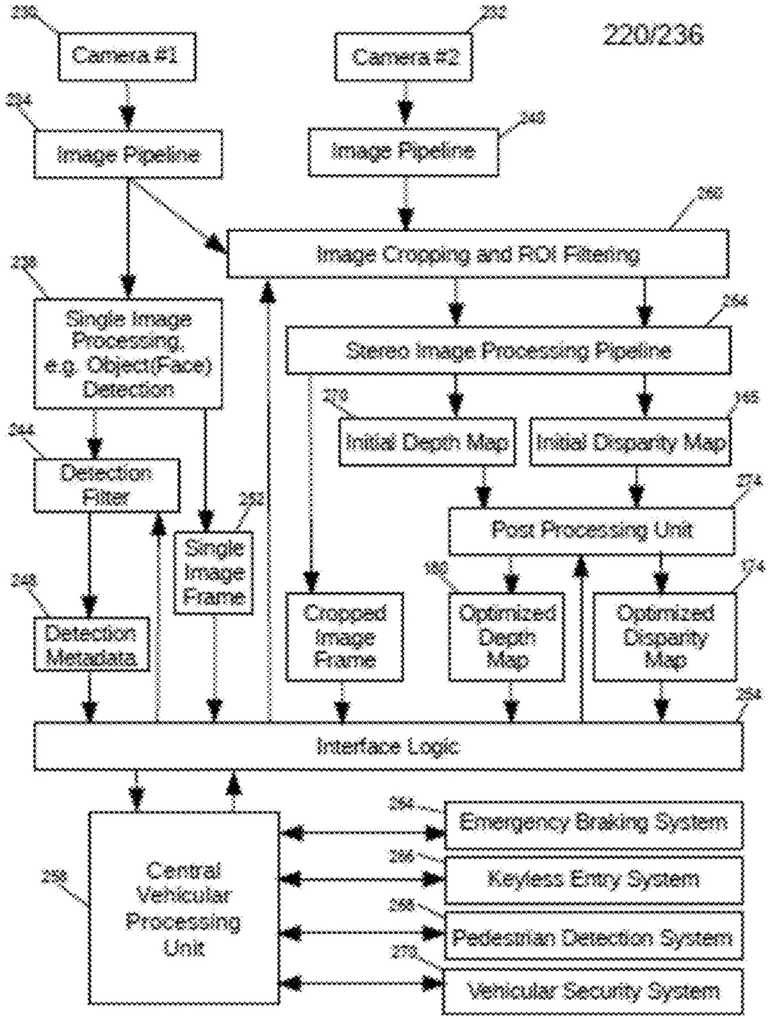
H04N 13/00	(2006.01)
G06T 5/20	(2006.01)
G06T 7/593	(2006.01)
H04N 13/02	(2006.01)

(52) **U.S. Cl.**

CPC ***H04N 13/0022*** (2013.01); ***H04N 13/0239*** (2013.01); ***G06T 5/20*** (2013.01); ***G06T 7/593*** (2017.01); ***H04N 13/0257*** (2013.01); ***H04N 13/0271*** (2013.01); *G06T 2207/10028* (2013.01); *G06T 2207/20032* (2013.01); *G06T 2207/10024* (2013.01); *G06T 2207/30256* (2013.01); *G06T 2207/30261* (2013.01)

(57) **ABSTRACT**

A method for improving accuracy of depth map information derived from image data descriptive of a scene. In one embodiment Mutual Feature Map data are created based on initial disparity map data values and the image data descriptive of the scene. The Mutual Feature Map data are applied to create a series of weighting functions representing structural details that can be transferred to the first disparity values to restore degraded features or replace some of the first disparity values with values more representative of structural features present in the image data descriptive of the scene.



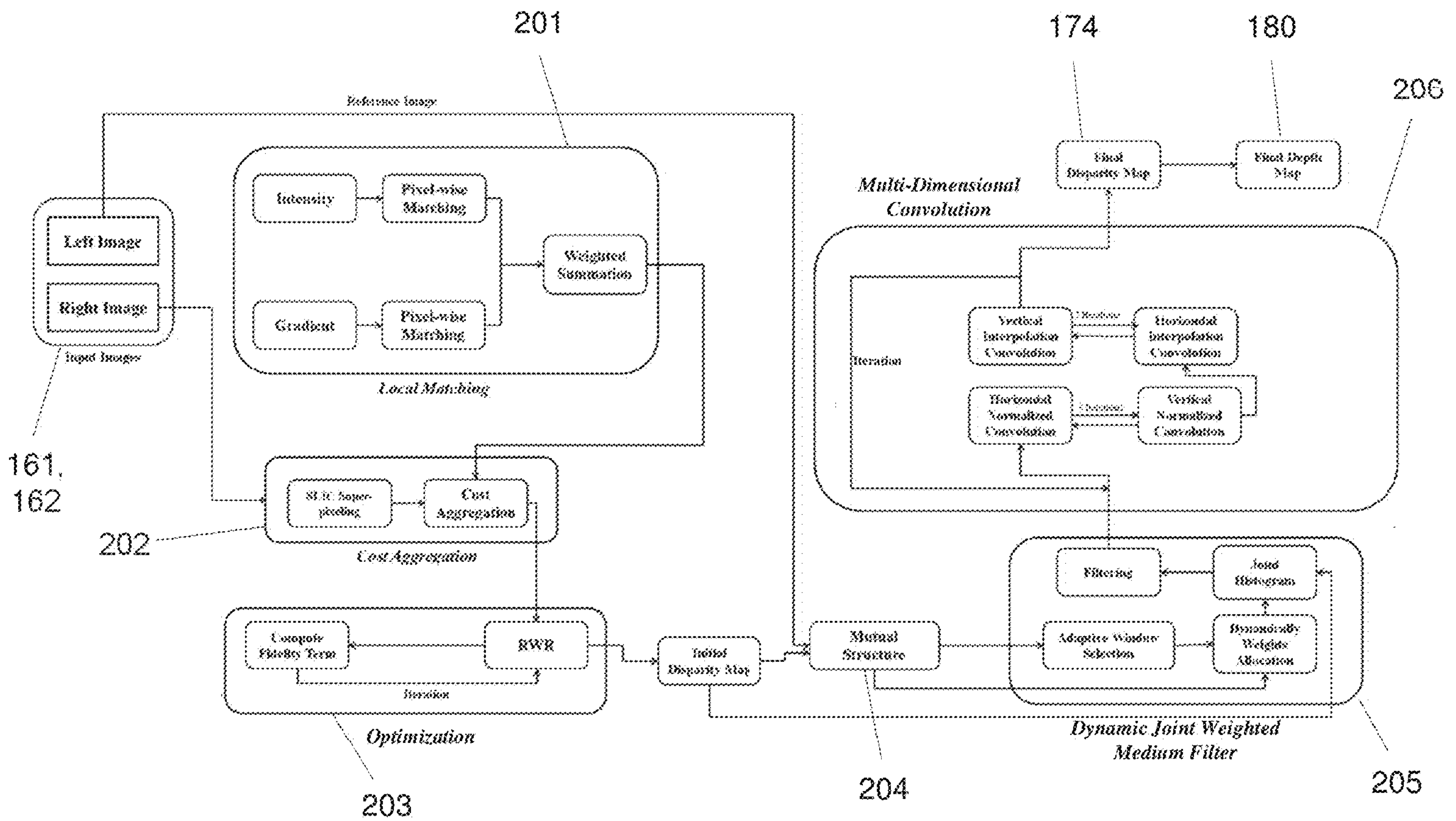

201
174
180
206
Reference Image
Left Image
Right Image
Input Images
161,
162
Intensity
Pixel-wise Matching
Gradient
Pixel-wise Matching
Weighted Summation
Local Matching
202
SLIC Super-pixeling
Cost Aggregation
Cost Aggregation
Compute Fidelity Term
RWR
Iteration
Optimization
203
Initial Disparity Map
Mutual Structure
204
Final Disparity Map
Final Depth Map
Multi-Dimensional Convolution
Iteration
Vertical Interpolation Convolution
Iteration
Horizontal Interpolation Convolution
Horizontal Normalized Convolution
Iteration
Vertical Normalized Convolution
Filtering
Joint Histogram
Adaptive Window Selection
Dynamically Weights Allocation
Dynamic Joint Weighted Medium Filter
205

Fig. 1

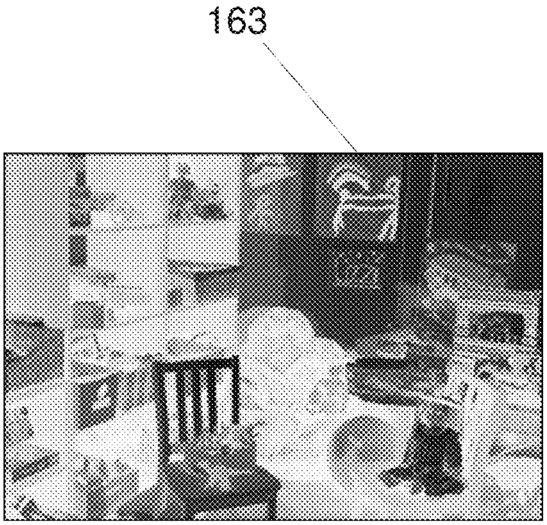


Fig. 2A

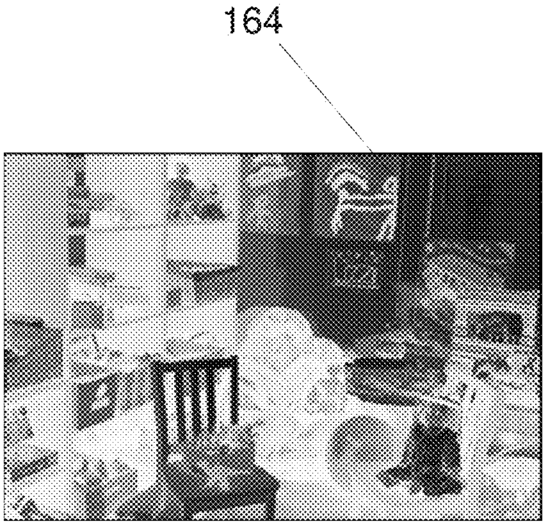


Fig. 2B

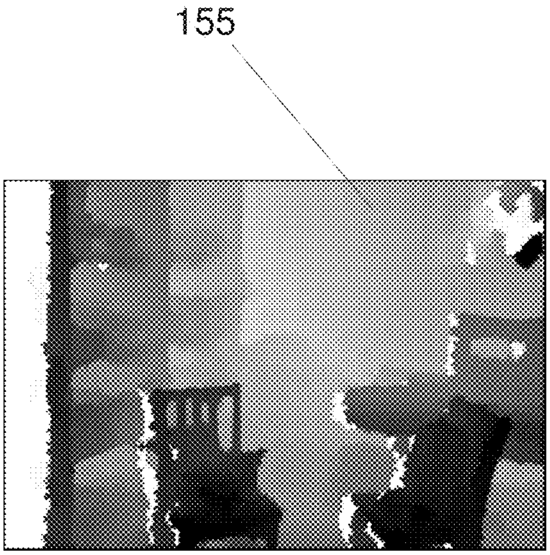


Fig. 2C

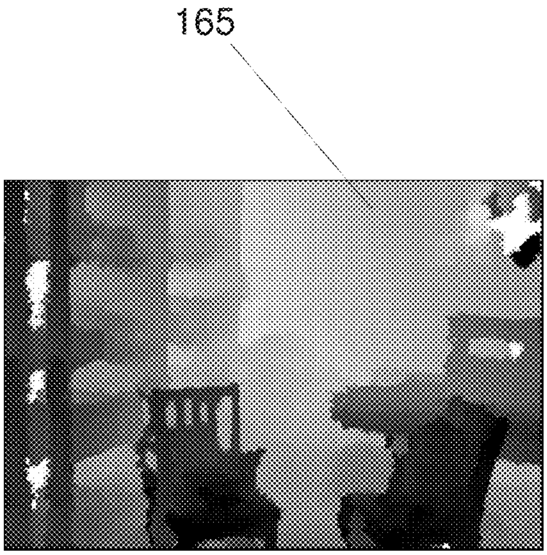


Fig. 2D

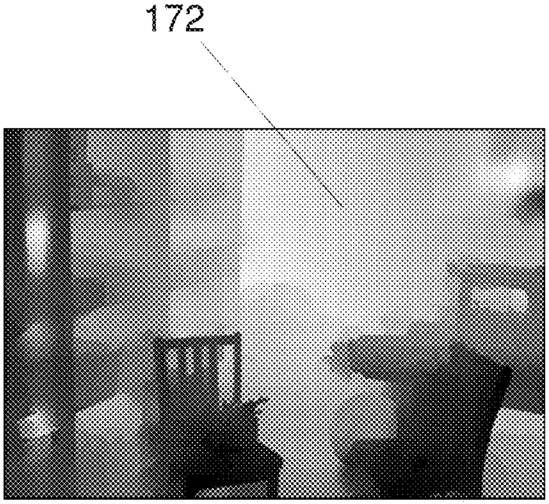


Fig. 2E

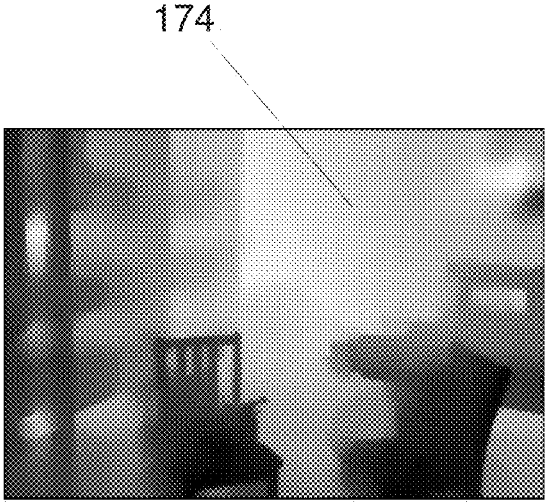


Fig. 2F

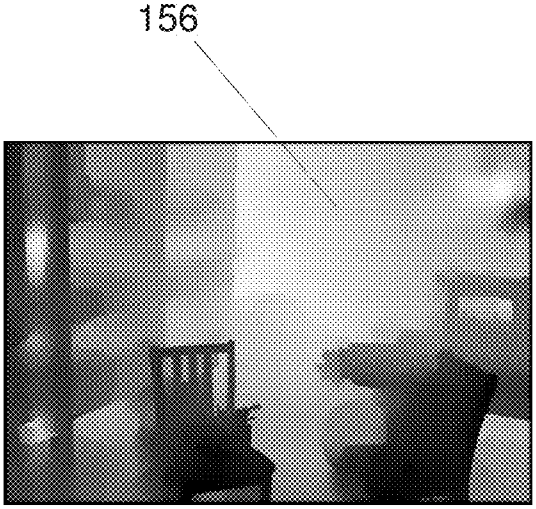


Fig. 2G

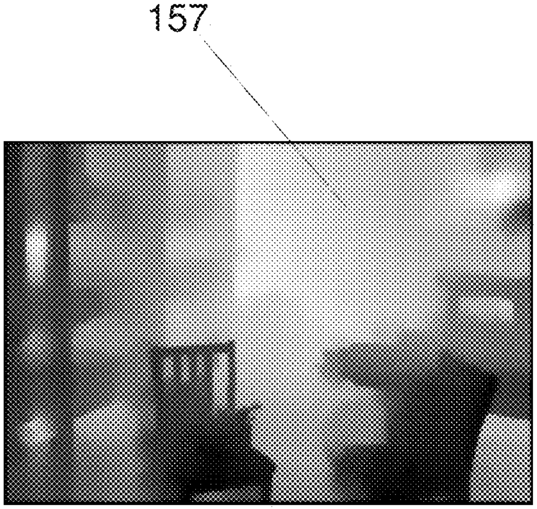


Fig. 2H

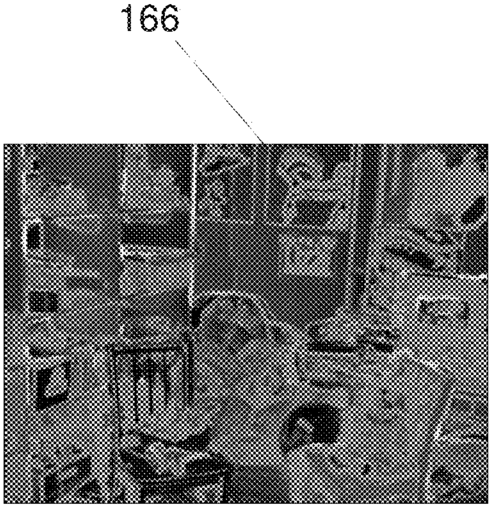


Fig. 2I

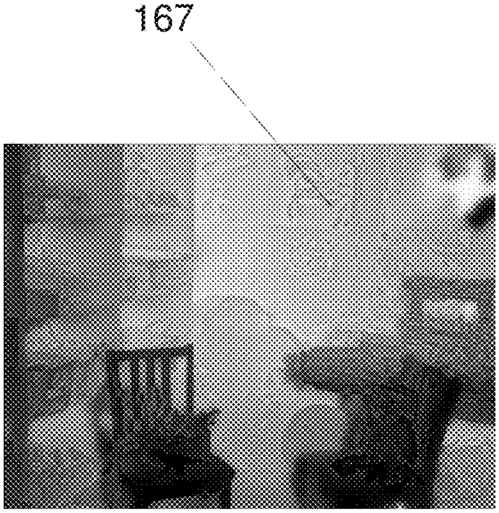


Fig. 2J

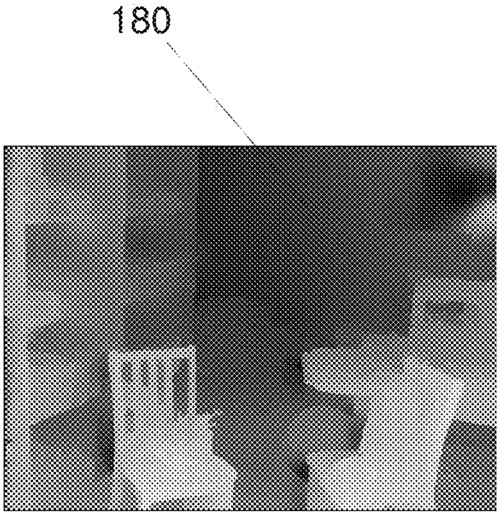


Fig. 2K

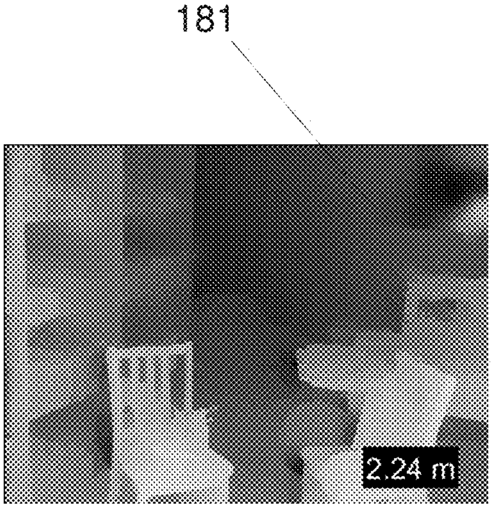


Fig. 2L

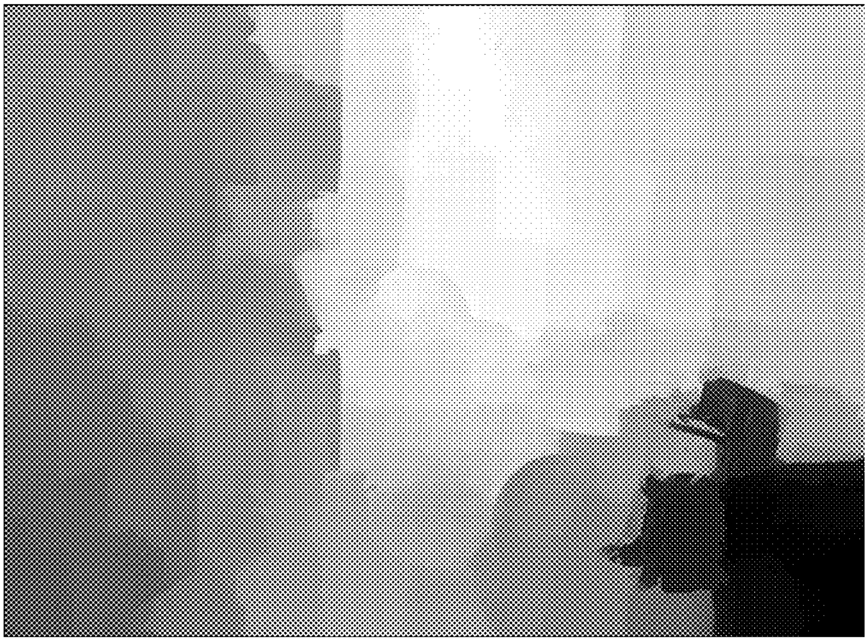


Fig. 3A

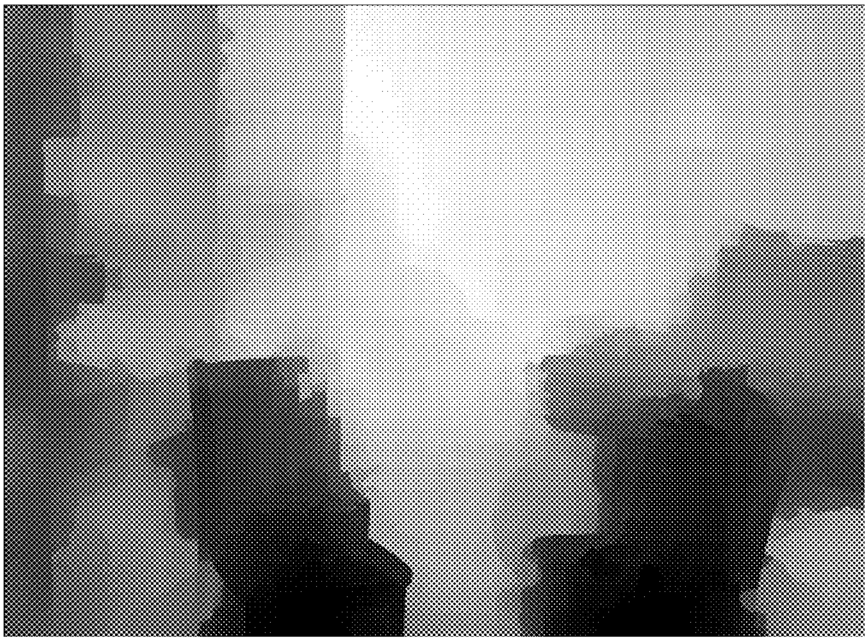


Fig. 3B

Fig. 3C

Fig. 3D

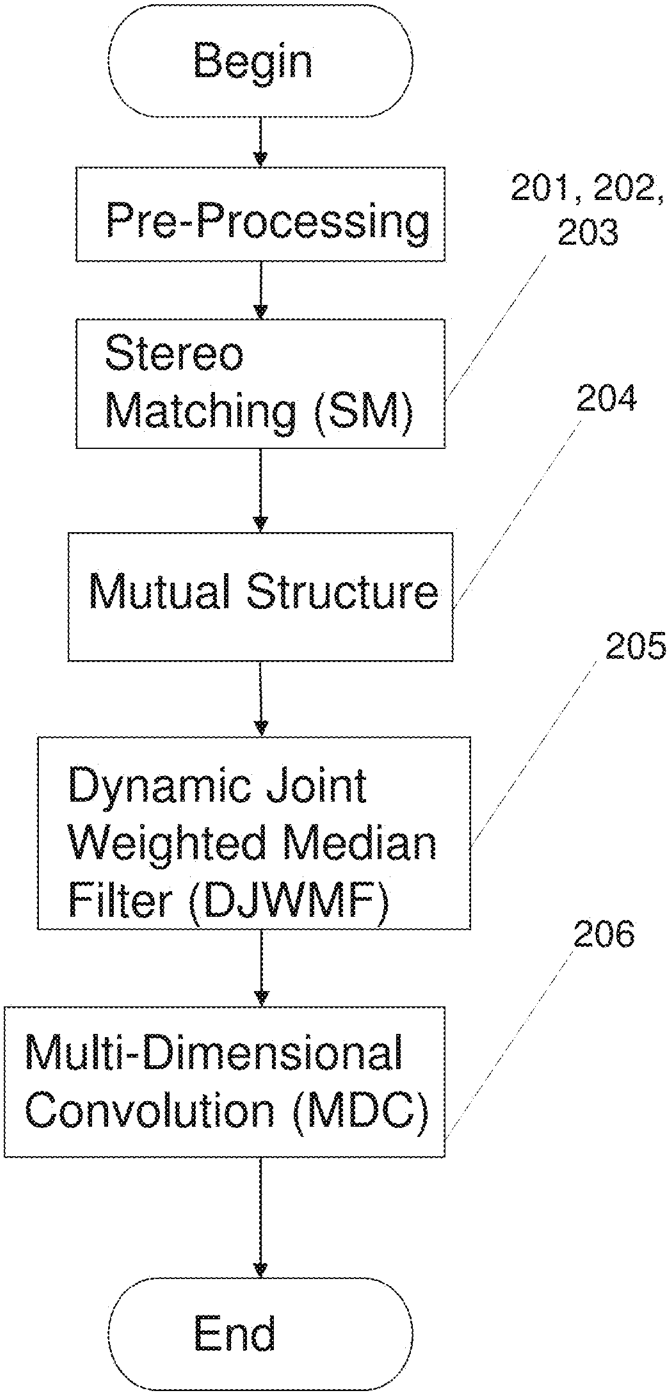

Begin
Pre-Processing
201, 202, 203
Stereo Matching (SM)
204
Mutual Structure
205
Dynamic Joint Weighted Median Filter (DJWMF)
206
Multi-Dimensional Convolution (MDC)
End

Fig. 4

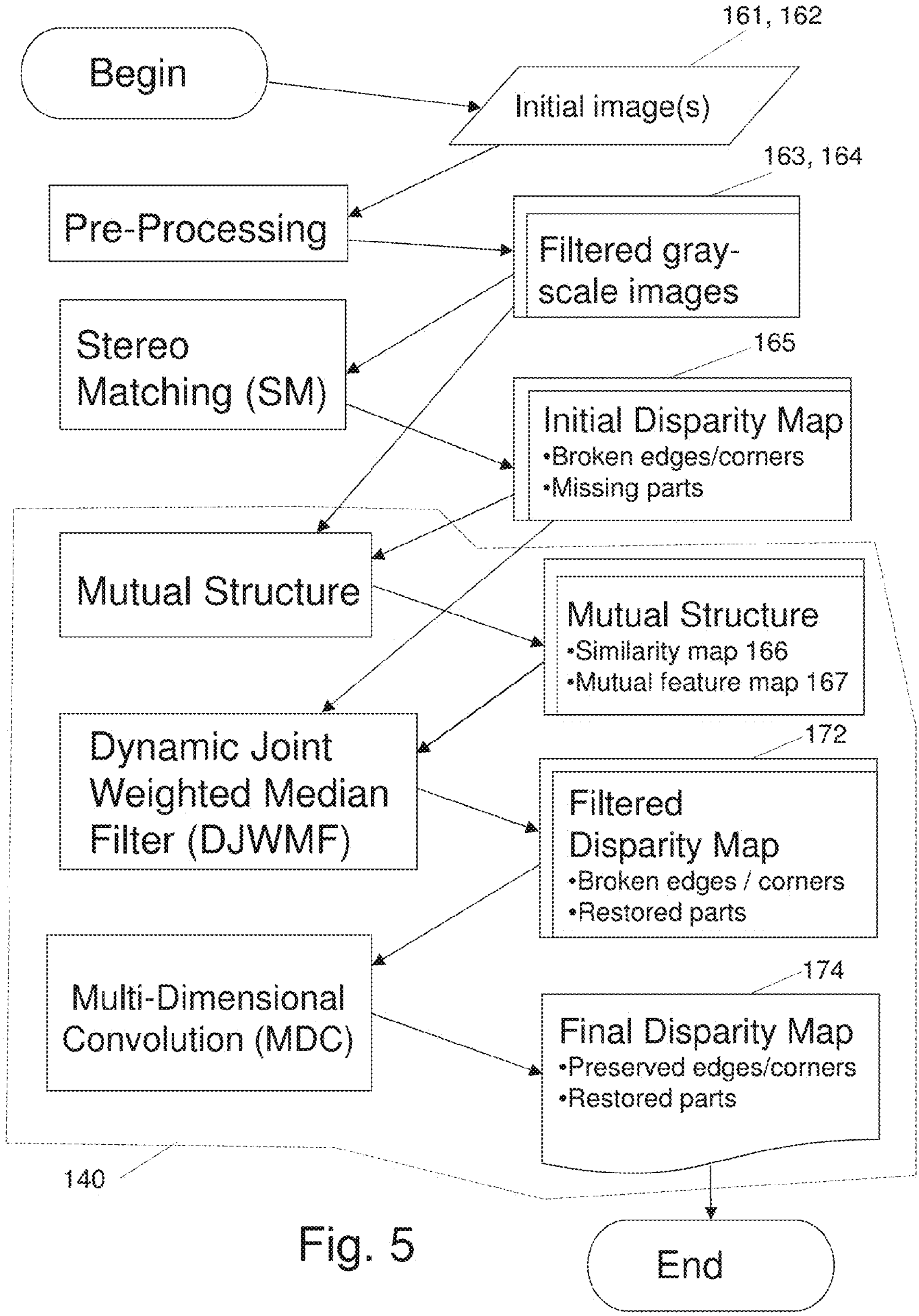

Begin
161, 162
Initial image(s)
163, 164
Pre-Processing
Filtered gray-scale images
Stereo Matching (SM)
165
Initial Disparity Map
•Broken edges/corners
•Missing parts
Mutual Structure
Mutual Structure
•Similarity map 166
•Mutual feature map 167
Dynamic Joint Weighted Median Filter (DJWMF)
172
Filtered Disparity Map
•Broken edges / corners
•Restored parts
Multi-Dimensional Convolution (MDC)
174
Final Disparity Map
•Preserved edges/corners
•Restored parts
140
End

Fig. 5

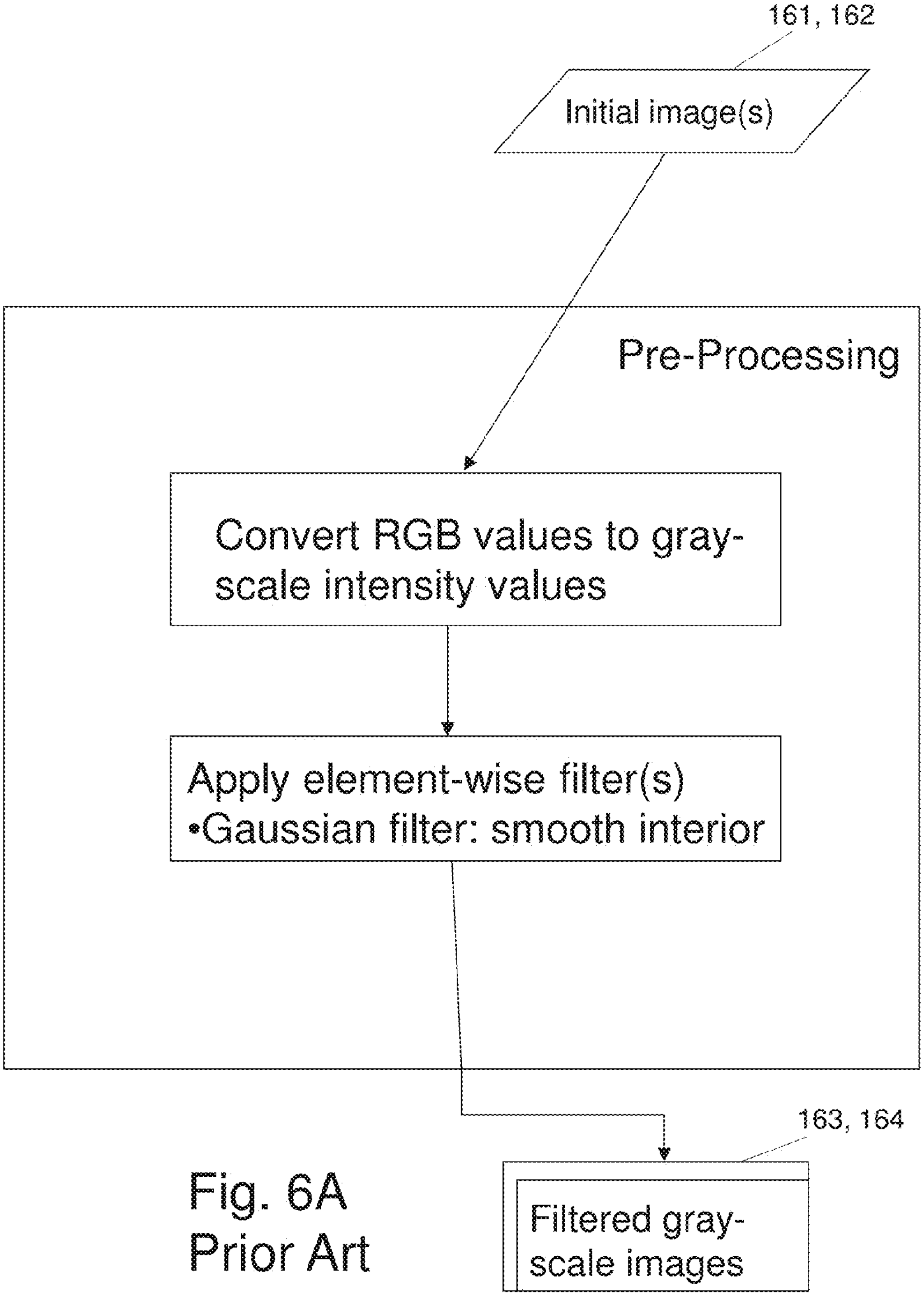

161, 162
Initial image(s)
Pre-Processing
Convert RGB values to gray-scale intensity values
Apply element-wise filter(s)
•Gaussian filter: smooth interior
163, 164
Filtered gray-scale images

Fig. 6A
Prior Art

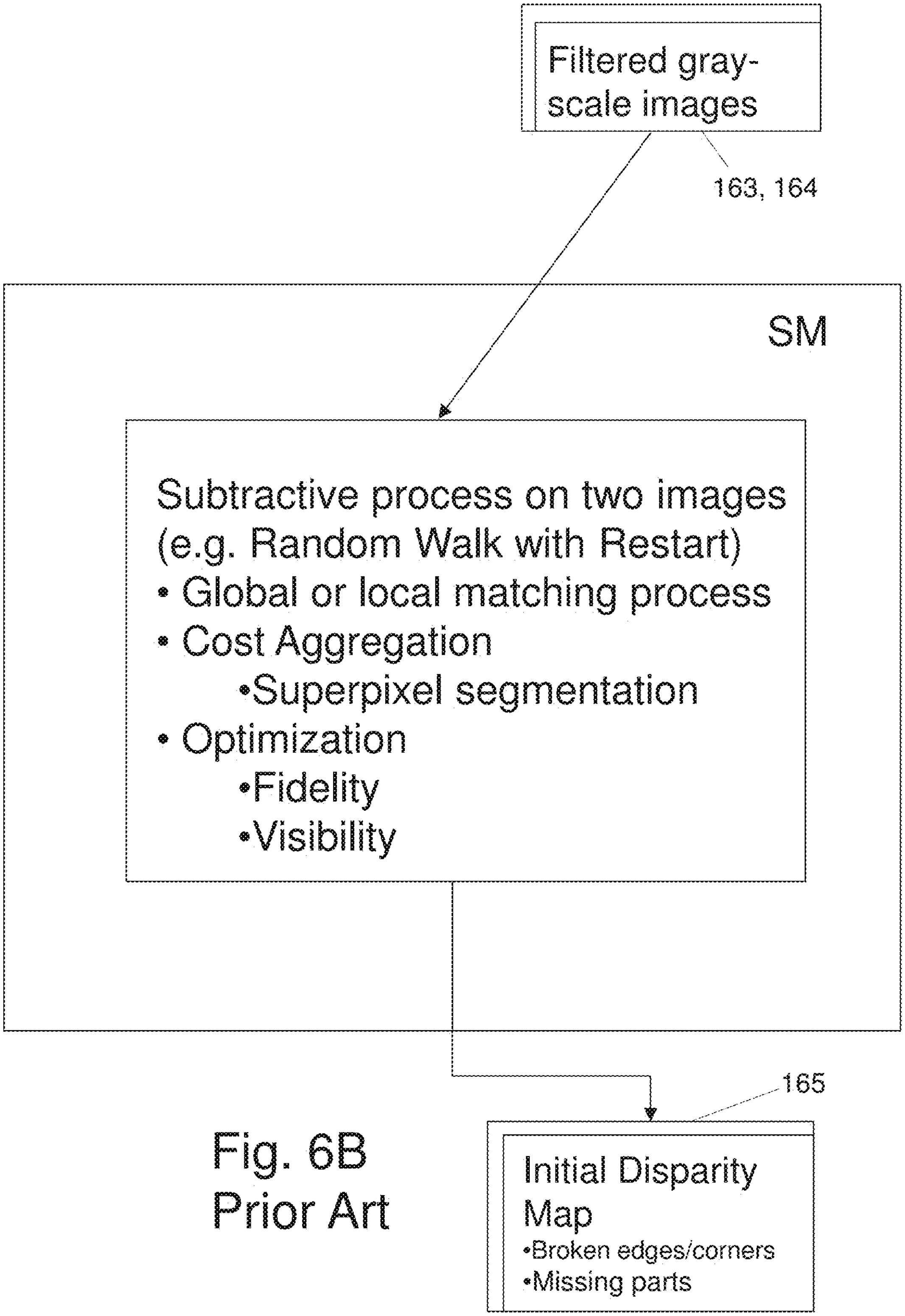


Fig. 6B
Prior Art

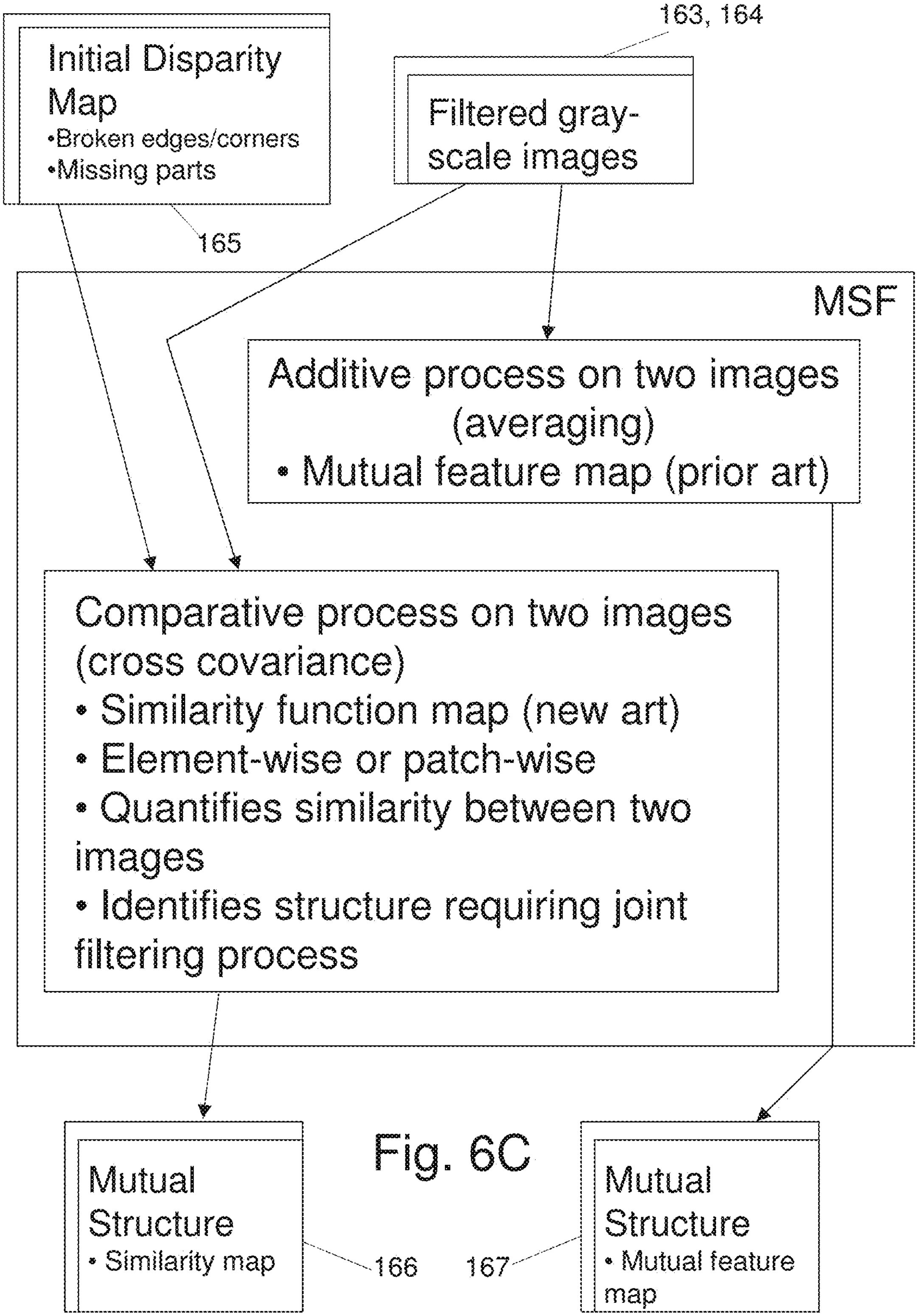

Initial Disparity Map
•Broken edges/corners
•Missing parts
165
163, 164
Filtered gray-scale images
MSF
Additive process on two images (averaging)
• Mutual feature map (prior art)
Comparative process on two images (cross covariance)
• Similarity function map (new art)
• Element-wise or patch-wise
• Quantifies similarity between two images
• Identifies structure requiring joint filtering process
Mutual Structure
• Similarity map
166
167
Mutual Structure
• Mutual feature map

Fig. 6C

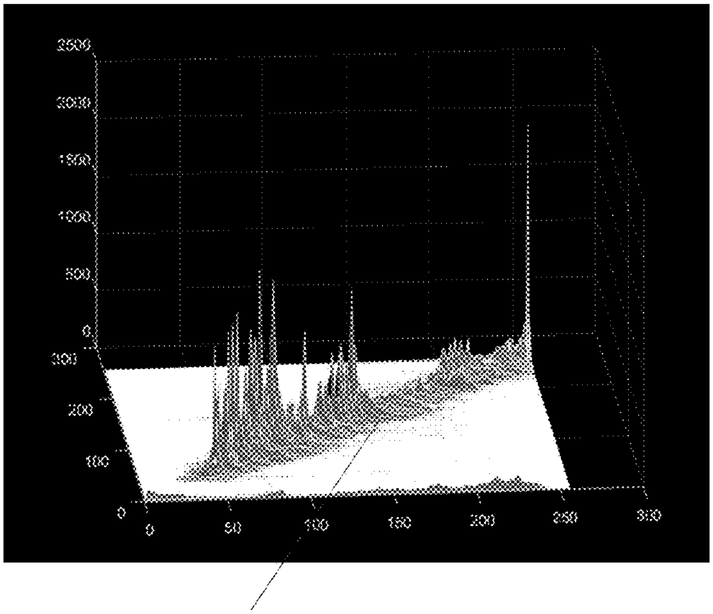

2500
2000
1500
1000
500
0
300
200
100
0
0
50
100
150
200
250
300
168

Fig. 6D

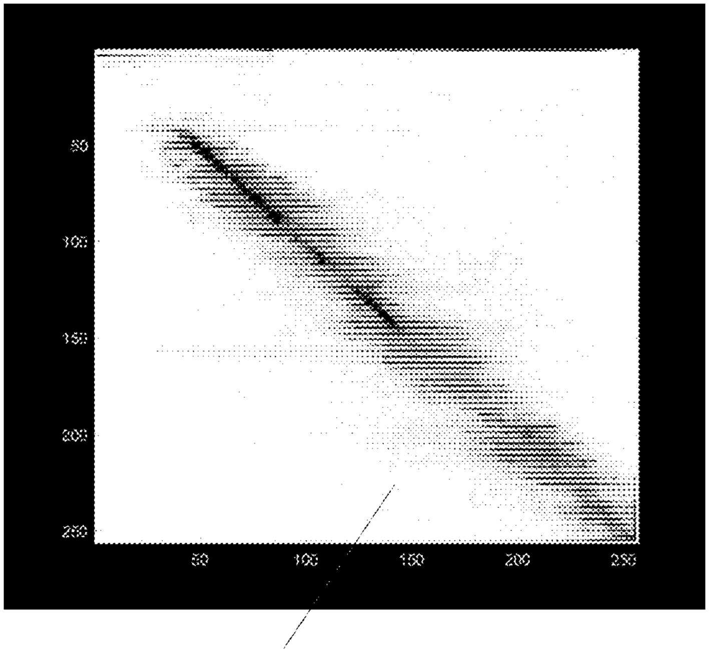

50
100
150
200
250
50
100
150
200
250
168

Fig. 6E

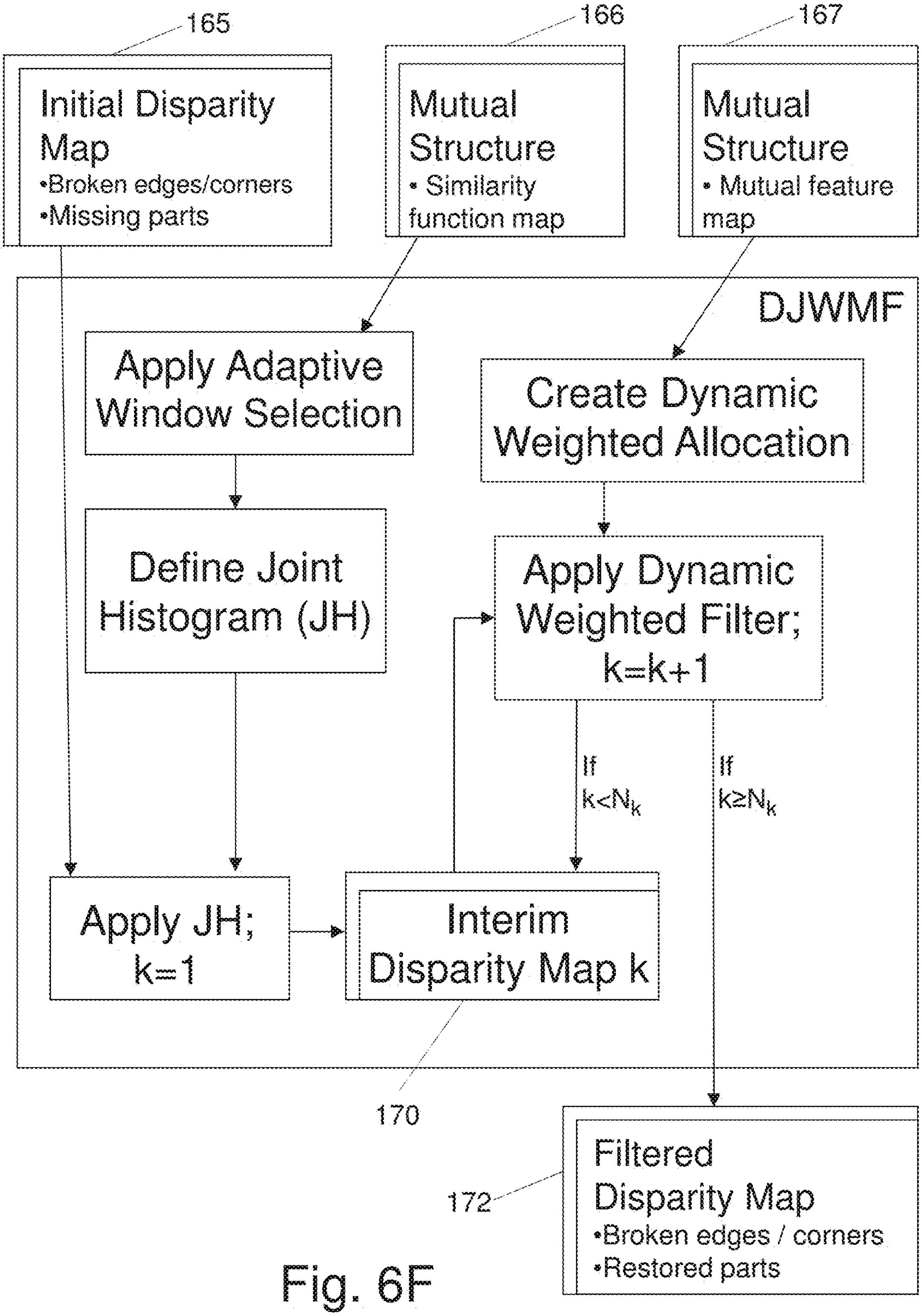

165
Initial Disparity Map
•Broken edges/corners
•Missing parts
166
Mutual Structure
• Similarity function map
167
Mutual Structure
• Mutual feature map
DJWMF
Apply Adaptive Window Selection
Create Dynamic Weighted Allocation
Define Joint Histogram (JH)
Apply Dynamic Weighted Filter; k=k+1
If k<N$_k$
If k≥N$_k$
Apply JH; k=1
Interim Disparity Map k
170
172
Filtered Disparity Map
•Broken edges / corners
•Restored parts

Fig. 6F

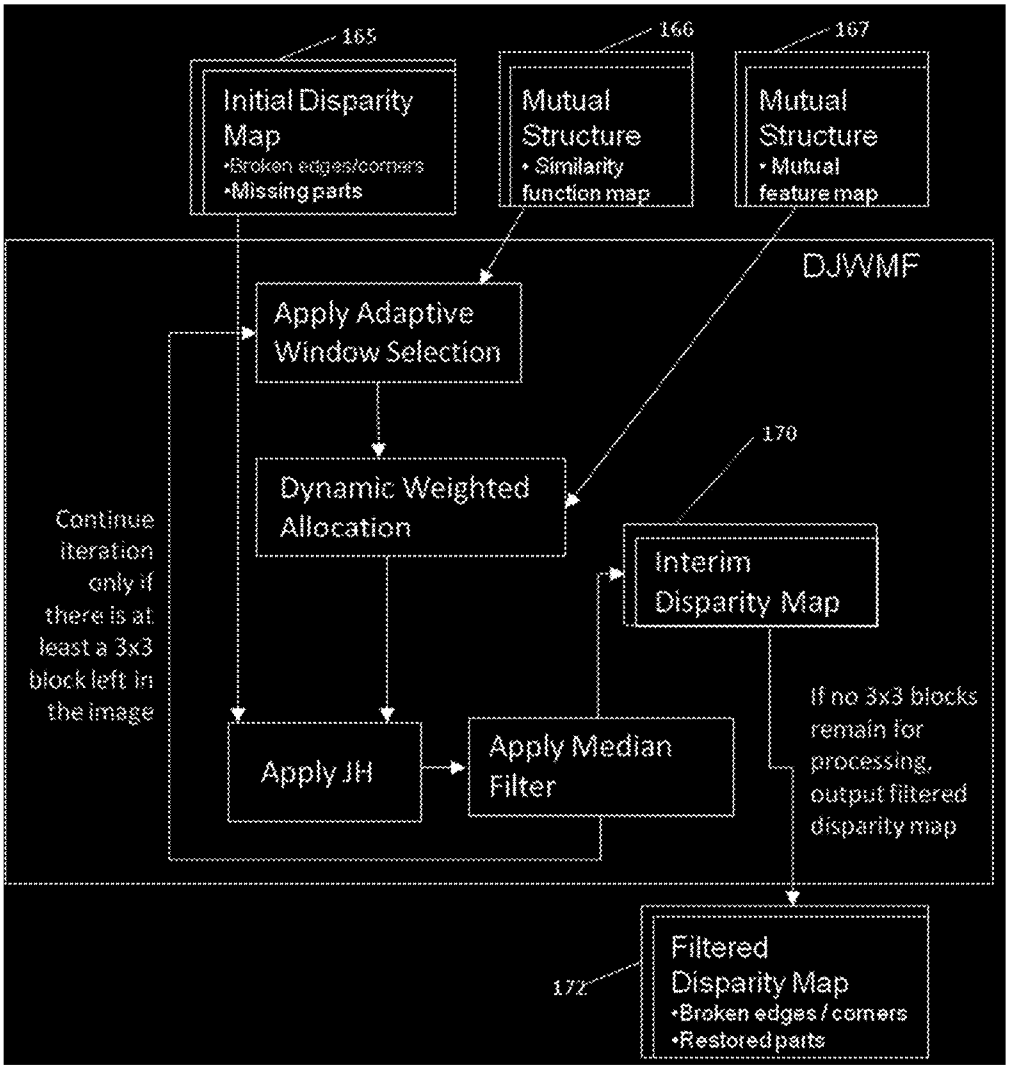

165
Initial Disparity Map
•Broken edges/corners
•Missing parts
166
Mutual Structure
• Similarity function map
167
Mutual Structure
• Mutual feature map
DJWMF
Apply Adaptive Window Selection
Dynamic Weighted Allocation
170
Interim Disparity Map
Continue iteration only if there is at least a 3x3 block left in the image
Apply JH
Apply Median Filter
If no 3x3 blocks remain for processing, output filtered disparity map
Filtered Disparity Map
•Broken edges / corners
•Restored parts
172

Fig. 6G

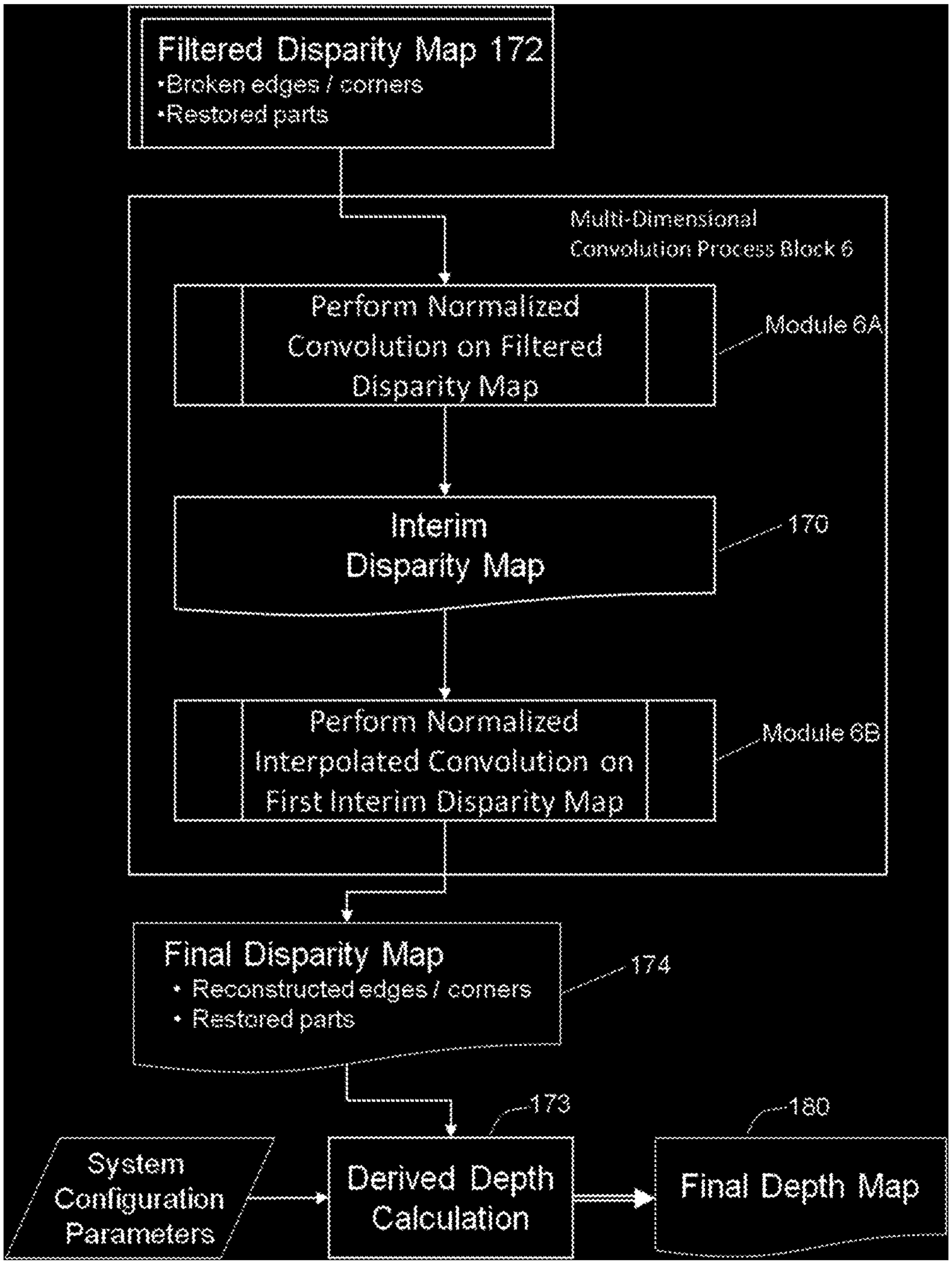

Filtered Disparity Map 172
•Broken edges / corners
•Restored parts
Multi-Dimensional
Convolution Process Block 6
Perform Normalized Convolution on Filtered Disparity Map
Module 6A
Interim Disparity Map
170
Perform Normalized Interpolated Convolution on First Interim Disparity Map
Module 6B
Final Disparity Map
• Reconstructed edges / corners
• Restored parts
174
173
180
System Configuration Parameters
Derived Depth Calculation
Final Depth Map

Fig. 6H

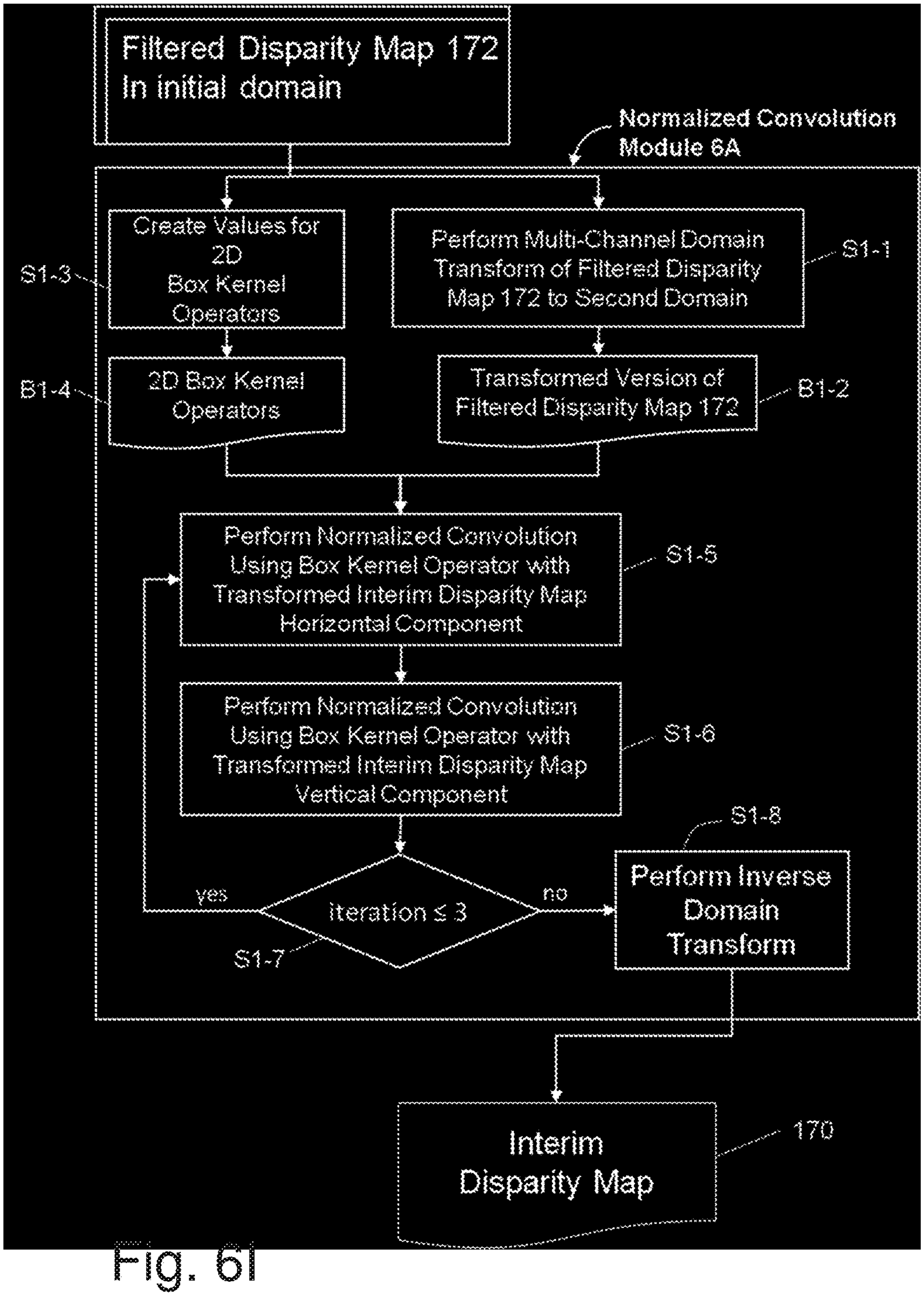

Filtered Disparity Map 172
In initial domain
Normalized Convolution
Module 6A
Create Values for
2D
Box Kernel
Operators
S1-3
Perform Multi-Channel Domain
Transform of Filtered Disparity
Map 172 to Second Domain
S1-1
2D Box Kernel
Operators
B1-4
Transformed Version of
Filtered Disparity Map 172
B1-2
Perform Normalized Convolution
Using Box Kernel Operator with
Transformed Interim Disparity Map
Horizontal Component
S1-5
Perform Normalized Convolution
Using Box Kernel Operator with
Transformed Interim Disparity Map
Vertical Component
S1-6
S1-8
yes
iteration ≤ 3
no
S1-7
Perform Inverse
Domain
Transform
Interim
Disparity Map
170

Fig. 6I

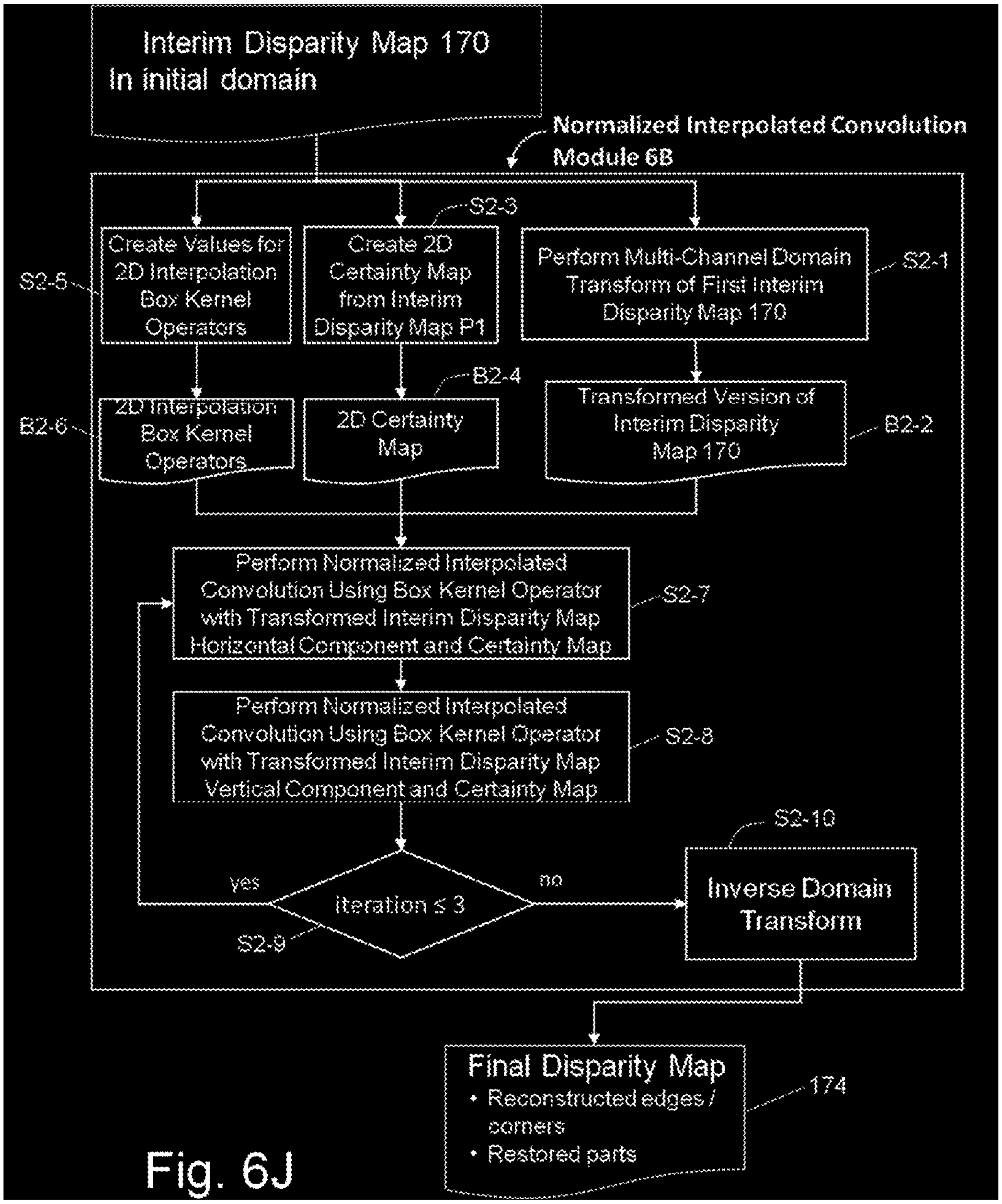

Interim Disparity Map 170
In initial domain
Normalized Interpolated Convolution
Module 6B
S2-3
Create Values for 2D Interpolation Box Kernel Operators
S2-5
Create 2D Certainty Map from Interim Disparity Map P1
Perform Multi-Channel Domain Transform of First Interim Disparity Map 170
S2-1
B2-4
2D Interpolation Box Kernel Operators
B2-6
2D Certainty Map
Transformed Version of Interim Disparity Map 170
B2-2
Perform Normalized Interpolated Convolution Using Box Kernel Operator with Transformed Interim Disparity Map Horizontal Component and Certainty Map
S2-7
Perform Normalized Interpolated Convolution Using Box Kernel Operator with Transformed Interim Disparity Map Vertical Component and Certainty Map
S2-8
S2-10
yes
Iteration ≤ 3
no
Inverse Domain Transform
S2-9
Final Disparity Map
• Reconstructed edges / corners
• Restored parts
174

Fig. 6J

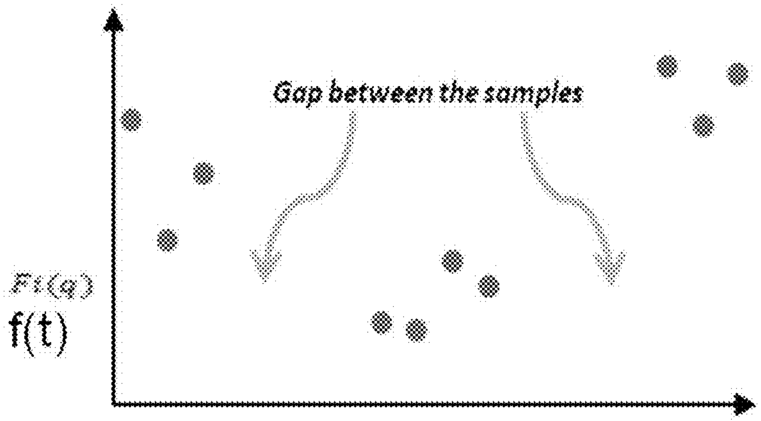


Fig. 7A

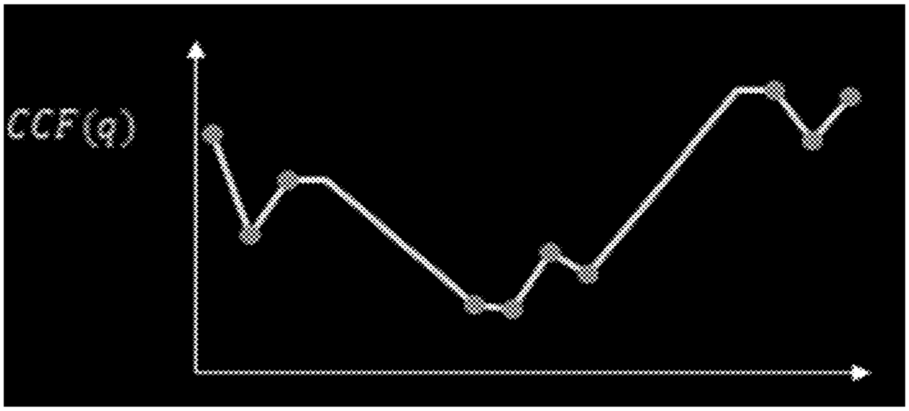


Fig. 7B

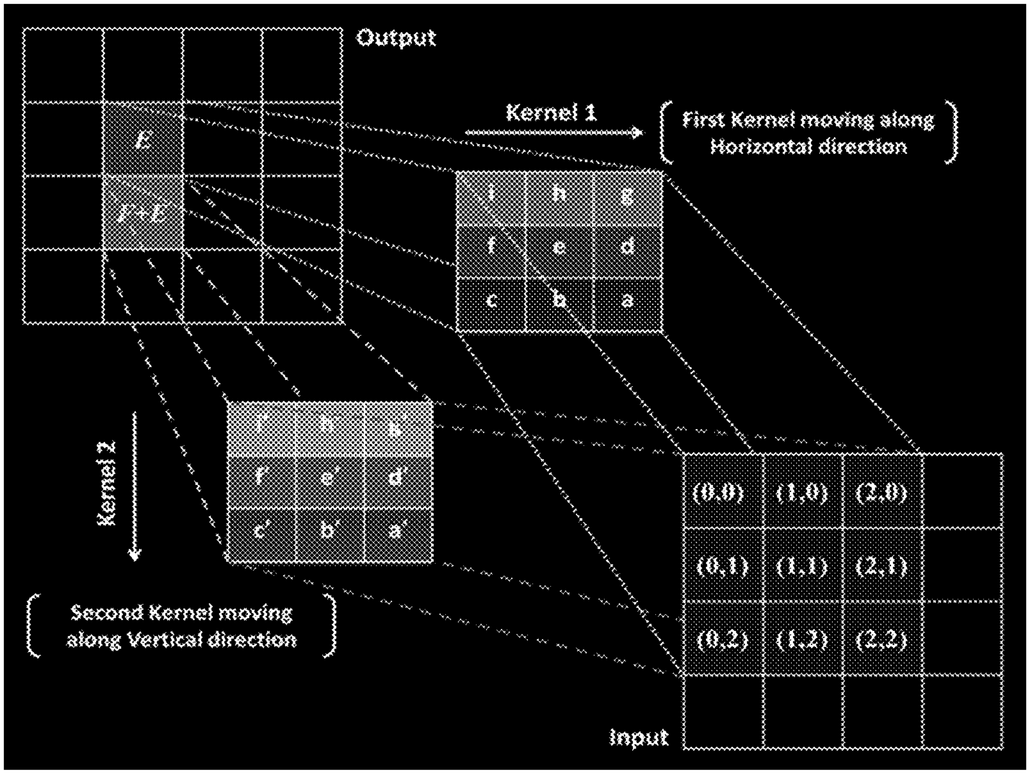
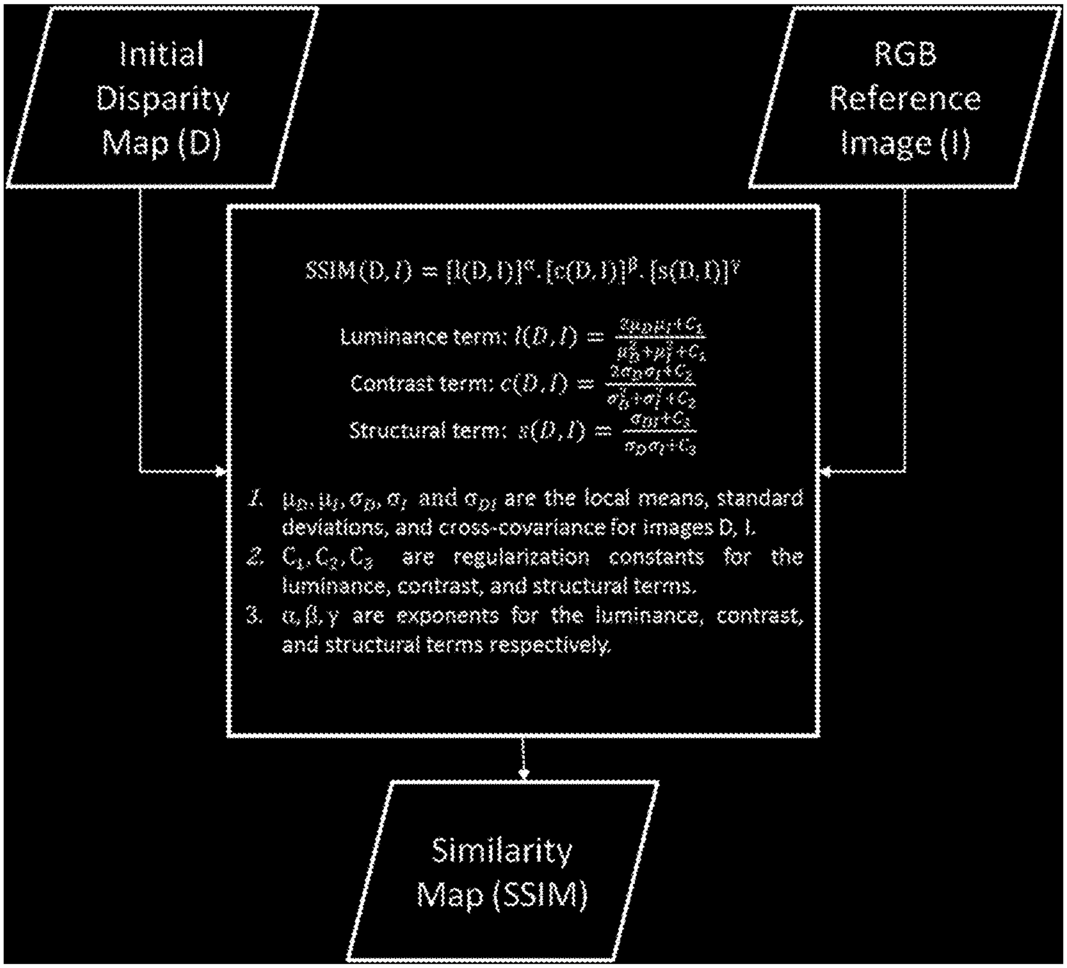

Output
E
F+E
Kernel 1
First Kernel moving along Horizontal direction
i
h
g
f
e
d
c
b
a
i'
h'
g'
f'
e'
d'
c'
b'
a'
Kernel 2
Second Kernel moving along Vertical direction
(0,0)
(1,0)
(2,0)
(0,1)
(1,1)
(2,1)
(0,2)
(1,2)
(2,2)
Input

Fig. 8



$$\mathrm{SSIM}(D,I) = [l(D,I)]^{\alpha} \cdot [c(D,I)]^{\beta} \cdot [s(D,I)]^{\gamma}$$

Luminance term: $l(D,I) = \frac{2\mu_D\mu_I + C_1}{\mu_D^2 + \mu_I^2 + C_1}$

Contrast term: $c(D,I) = \frac{2\sigma_D\sigma_I + C_2}{\sigma_D^2 + \sigma_I^2 + C_2}$

Structural term: $s(D,I) = \frac{\sigma_{DI} + C_3}{\sigma_D\sigma_I + C_3}$

1. $\mu_D, \mu_I, \sigma_D, \sigma_I$ and σ_{DI} are the local means, standard deviations, and cross-covariance for images D, I.
2. C_1, C_2, C_3 are regularization constants for the luminance, contrast, and structural terms.
3. α, β, γ are exponents for the luminance, contrast, and structural terms respectively.

Fig. 9

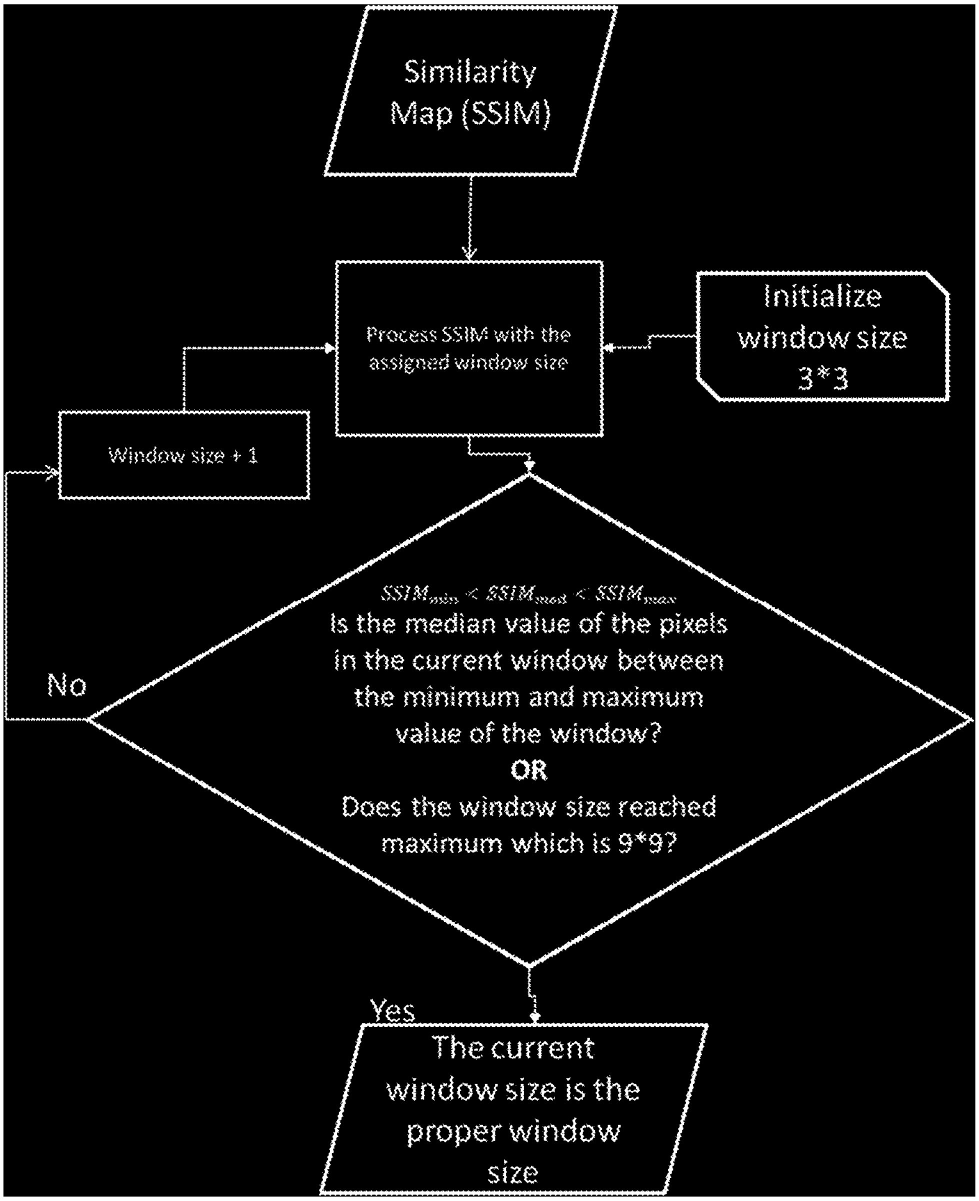

Similarity Map (SSIM)
Process SSIM with the assigned window size
Initialize window size 3*3
Window size + 1
$SSIM_{min} < SSIM_{med} < SSIM_{max}$
Is the median value of the pixels in the current window between the minimum and maximum value of the window?
OR
Does the window size reached maximum which is 9*9?
No
Yes
The current window size is the proper window size

Fig. 10

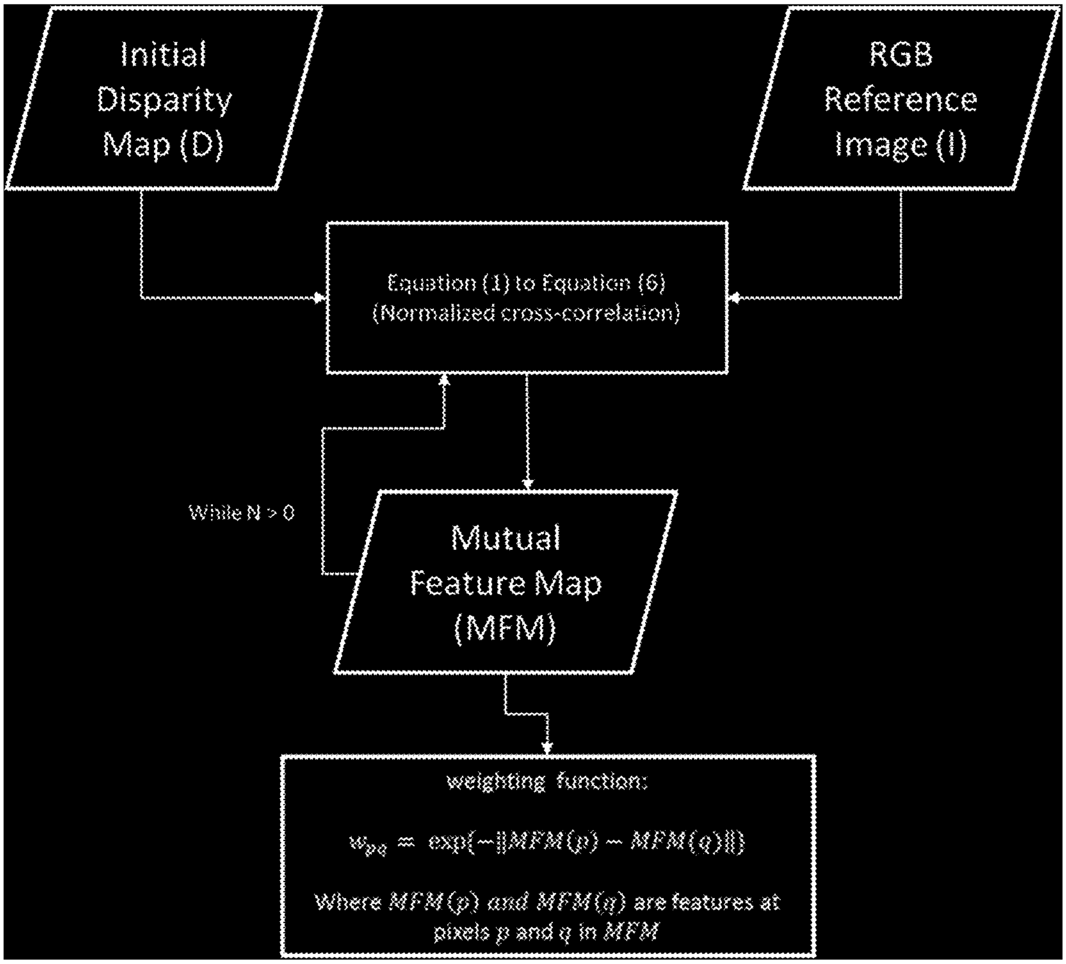

Initial Disparity Map (D)
RGB Reference Image (I)
Equation (1) to Equation (6) (Normalized cross-correlation)
While N > 0
Mutual Feature Map (MFM)
weighting function:
$w_{pq} = \exp\{-\|MFM(p) - MFM(q)\|\}$
Where $MFM(p)$ and $MFM(q)$ are features at pixels p and q in MFM

Fig. 11

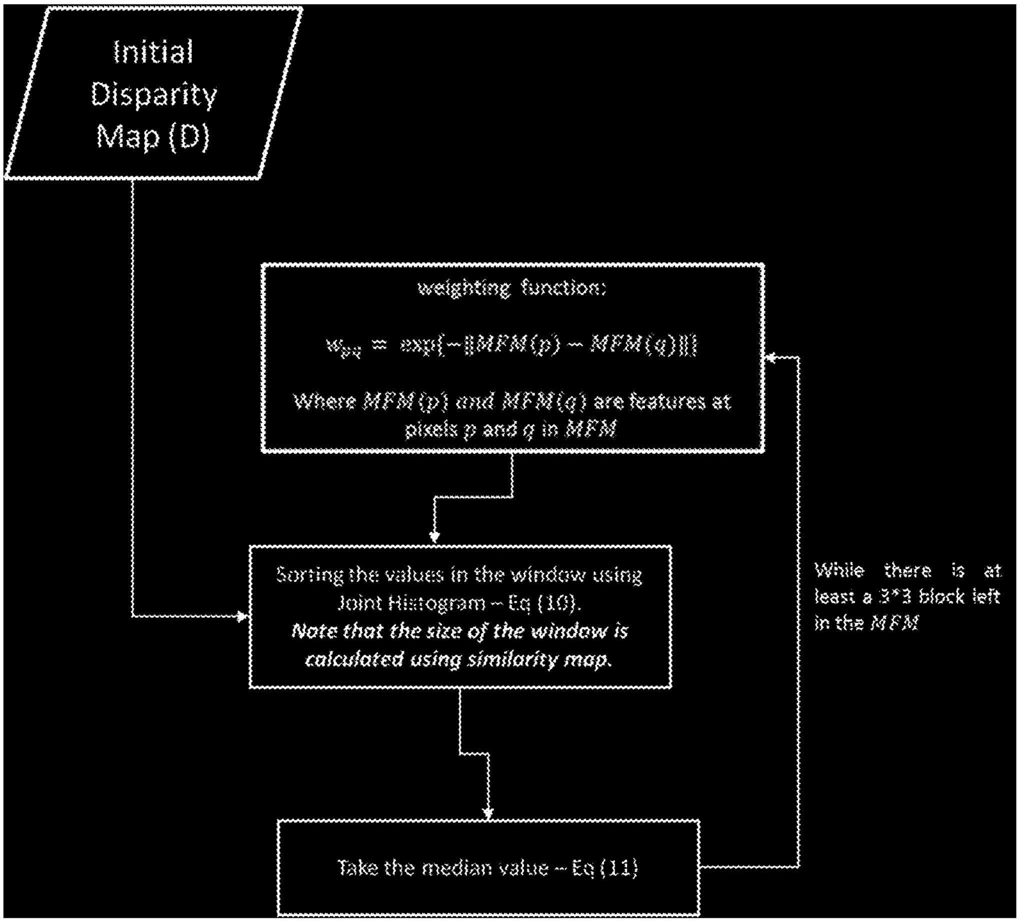

Initial Disparity Map (D)
weighting function:
$w_{pq} = \exp\{-\|MFM(p) - MFM(q)\|\}$
Where $MFM(p)$ and $MFM(q)$ are features at pixels p and q in MFM
Sorting the values in the window using Joint Histogram – Eq (10).
Note that the size of the window is calculated using similarity map.
Take the median value – Eq (11)
While there is at least a 3*3 block left in the MFM

Fig. 12

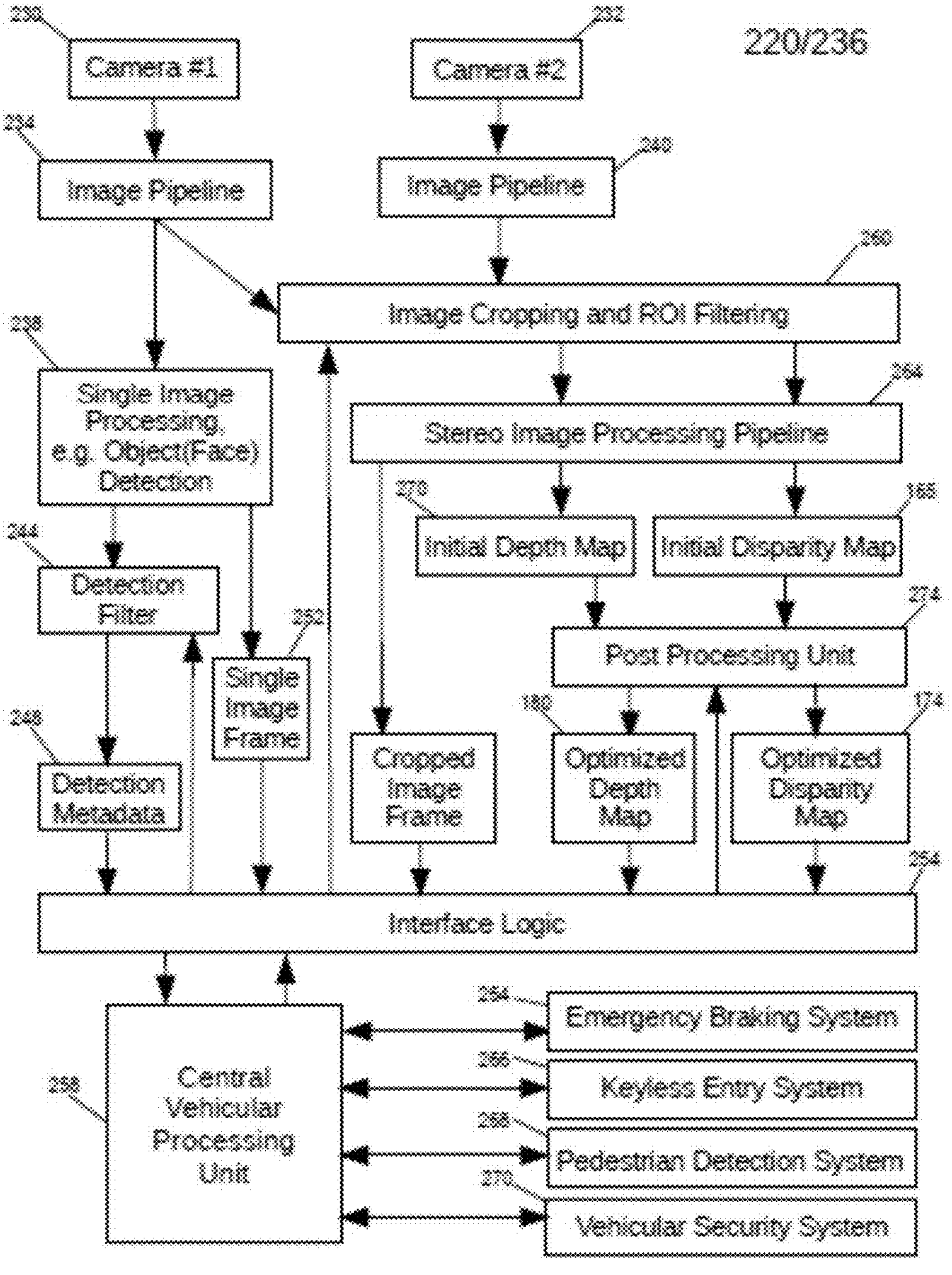

220/236
230
Camera #1
232
Camera #2
234
Image Pipeline
240
Image Pipeline
260
Image Cropping and ROI Filtering
238
Single Image Processing, e.g. Object(Face) Detection
262
Stereo Image Processing Pipeline
272
Initial Depth Map
168
Initial Disparity Map
244
Detection Filter
252
Single Image Frame
274
Post Processing Unit
248
Detection Metadata
Cropped Image Frame
180
Optimized Depth Map
174
Optimized Disparity Map
254
Interface Logic
264
Emergency Braking System
266
Keyless Entry System
258
Central Vehicular Processing Unit
268
Pedestrian Detection System
270
Vehicular Security System

Fig. 13

SYSTEMS AND METHODS FOR ESTIMATING AND REFINING DEPTH MAPS

CLAIM OF PRIORITY AND RELATED PATENTS AND APPLICATIONS

[0001] This application claims priority to provisional patent application Ser. No. 62/364,263, "Depth Map Post-Processing Approach Based on Adaptive Random Walk with Restart" filed 19 Jul. 2016 and is related to U.S. Pat. No. 7,916,897, U.S. Pat. No. 8,170,294, U.S. Pat. No. 8,934,680; U.S. Pat. No. 8,872,887, U.S. Pat. No. 8,995,715, U.S. Pat. No. 8,385,610, U.S. Pat. No. 9,224,034, U.S. Pat. No. 9,242,602, U.S. Pat. No. 9,262,807, U.S. Pat. No. 9,280,810, U.S. Pat. No. 9,398,209 and U.S. patent application Ser. No. 13/862,372, filed Apr. 12, 2013, and U.S. patent application Ser. No. 14/971,725, filed Dec. 16, 2015, and U.S. patent application Ser. No. 15/591,321, filed May 10, 2017, all of which are assigned to the assignee of the present application and hereby incorporated by reference.

FIELD OF THE INVENTION

[0002] This invention relates to processing methods and systems for estimating and refining depth maps and disclosed embodiments relate to processing techniques which bring optimized local matching costs to improved levels of speed and accuracy.

BACKGROUND OF THE INVENTION

[0003] Time critical machine vision applications require high levels of speed and accuracy in the matching algorithms which determine depth. Depth estimation is typically based on stereo correspondence, the difference in coordinates of corresponding pixels in stereo images. The difference in coordinate position between a pair of corresponding pixels is referred to as the disparity, and the assimilation of differences among pairs of corresponding pixels in stereo imagery is referred to as a depth map.

[0004] The accuracy of depth mapping is dependent on accurate identification of corresponding pixels while applications, such as automatic vehicle braking, require rapid execution. Satisfactory accuracy for real time responses can require rapid execution of data intensive, iterative computations.

[0005] Conventionally, estimating depth from imagery normally begins with application of a stereo matching algorithm to construct a disparity map from a pair of images taken of the same scene from different viewpoints. Typically, the two images are acquired at the same time with two cameras residing in the same lateral plane, although a depth map may also be determined from correspondence between images of a scene captured at different times provided that spatial differences occur between corresponding pixels in the lateral plane. Generally, for depth estimations, most of the pixels of interest in one image will have a corresponding pixel in the other image.

SUMMARY OF THE INVENTION

[0006] Embodiments of the present invention employ a stochastic approach comprising a combination of iterative refinements to generate an optimized disparity map. The figures illustrate an exemplary embodiment of a control system **10** which applies improved processing techniques in conjunction with matching algorithms to provide a more accurate disparity map at computation speeds suitable for real time applications.

[0007] While the invention can be practiced with numerous other matching algorithms, FIG. **1** illustrates application of a matching algorithm for pixel wise determination of initial matching costs. In this example, in which an Adaptive Random Walk with Reset (ARWR) algorithm is iteratively applied to optimize stereo matching, processing steps address discontinuities and occlusions, and provide additional filtering steps to enhance image registration. Optimized matching costs bring local matching to improved levels of speed and accuracy.

[0008] ARWR methods have previously been applied to solve the stereo correspondence problem. See, for example, S. Lee, et al., "Robust stereo matching using adaptive random walk with restart algorithm," *Image and Vision Computing*, vol. 37, pp. 1-11 (2015). See, also, Hamzah and Ibrahim, "Literature Survey on Stereo Vision Disparity Map Algorithms" *Journal of Sensors*, Volume 2016 p. 6 (2016); and Oh, Changjae, Bumsub Ham, and Kwanghoon Sohn. "Probabilistic Correspondence Matching using Random Walk with Restart." *BMVC*, pp. 1-10. 2012.

[0009] The system **10** incorporates a sequence of six major process steps, referred to as Processing Blocks: (1) Local Matching Processing Block 1; (2) Cost Aggregation Processing Block 2; (3) Optimization Processing Block 3 which iteratively applies a matching algorithm; (4) Mutual Structure Processing Block 4 which identifies structure common to the images; (5) Dynamic Joint Weighted Median Filter (DJWMF) Processing Block 5; and (6) Multi-Dimensional Convolution (MDC) Processing Block 6.

BRIEF DESCRIPTION OF THE FIGURES

[0010] Other aspects and advantages of the present invention will be more clearly understood by those skilled in the art when the following description is read with reference to the accompanying drawings wherein:

[0011] FIG. **1** illustrates, in flowchart format, an exemplary depth estimation system and an exemplary multi-step depth estimation process, based on pixel-wise stereo matching which incorporates post processing refinements according to the invention;

[0012] FIGS. **2**A-L show an exemplary set of inter-related images resulting from intermediate steps of an exemplary multi-step depth estimation process according to the invention, based on Middlebury Benchmark stereo input RGB images "Playroom":

[0013] Not shown are color images, first input (initial) (reference) RGB image "I" **161**, and second input (initial) (reference) RGB image **162**, where RGB input images **161**, **162** are a stereo pair;

[0014] FIG. **2**A shows a first filtered gray-scale image **163**;

[0015] FIG. **2**B shows a second filtered gray-scale image **164**;

[0016] FIG. **2**C shows a Disparity Map After Cost Aggregation **155**;

[0017] FIG. **2**D shows an Initial Disparity Map "D" **165**;

[0018] FIG. **2**E shows a Filtered Disparity Map **172**;

[0019] FIG. **2**F shows a Final Disparity Map **174**;

[0020] FIG. **2**G shows a Disparity Map (After Convolution) **156**;

[0021] FIG. **2**H shows a Disparity Map (After Interpolation) **157**;

[0022] FIG. **2**I shows a Similarity Map **166**;

[0023] FIG. **2**J shows a Mutual Feature Map **167**;

[0024] FIG. **2**K shows a Depth Map **180**;

[0025] FIG. **2**L shows a Labeled Depth Map **181**;

[0026] In FIG. **3** a series of images illustrates a form of artifact which varies as a function of superpixel size, where FIGS. **3**A, **3**B, **3**C, and **3**D respectively illustrate using N=100, 1000, 5000, and 16000 pixels;

[0027] FIG. **4** summarizes a series of processing steps in a depth estimation process according to the invention;

[0028] FIG. **5** describes a series of information inputs to the processing steps of FIG. **4** to provide a refined depth map, showing Depth Map Refinement Processing **140** according to the invention;

[0029] FIGS. **6**A and **6**B briefly illustrate conventional processing steps incorporated into the exemplary depth estimation process;

[0030] FIG. **6**C summarizes development of Similarity Map data and Mutual Feature Map data according to the invention;

[0031] FIG. **6**D shows an exemplary Joint Histogram (JH) **168** in a 3-dimensional (3D) format);

[0032] FIG. **6**E shows an exemplary Joint Histogram (JH) **168** in a 2-dimensional (2D) format);

[0033] FIG. **6**F, **6**G illustrate alternate embodiments for implementation of a Dynamic Joint Weighted Median Filter Process which applies the map data of FIG. **6**C and the Joint Histogram (JH) **168** of FIGS. **6**D, **6**E to restore, i.e. reconstruct, features in disparity map data;

[0034] FIGS. **6**H, **6**I, **6**J illustrate alternate embodiments for implementation of a multistep convolution process performed on disparity map data generated by the filter process of FIG. **6**F, **6**G to further refine disparity map data for improved depth estimation, by providing details of a Processing Block 6, Normalized Interpolated Convolution (NIC), reference **206**;

[0035] FIGS. **7**A, **7**B illustrate details of an interpolation process performed on disparity map data generated by the convolution process of FIGS. **6**H, **6**I, **6**J to further refine disparity map data for improved depth estimation:

[0036] FIG. **7**A illustrates in 1D exemplary data showing gaps between valid data;

[0037] FIG. **7**B illustrates in 1D an exemplary reconstruction by interpolation, of missing performed on the data of FIG. **124**A;

[0038] FIG. **8** illustrates general features of 2D interpolation according to the invention;

[0039] FIGS. **9**, **10**, **11**, **12** provide details of a Processing Block 5, Dynamic Joint Weighted Median Filter (DJWMF), reference **205**; and

[0040] FIG. **13** illustrates an exemplary depth estimation system systems design suitable application in a vehicle which incorporates depth map refinement techniques, based on pixel-wise stereo matching which incorporates post processing refinements according to the invention.

[0041] Like reference numbers are used throughout the figures to denote like components. Numerous components are illustrated schematically, it being understood that various details, connections and components of an apparent nature are not shown in order to emphasize feature of the invention. Various features shown in the figures are not shown to scale in order to emphasize features of the invention.

DETAILED DESCRIPTION OF THE INVENTION

[0042] Before describing in detail particular methods, components and features relating to the invention, it should be observed that the present invention resides primarily in a novel and non-obvious combination of elements and method steps. So as not to obscure the disclosure with details that will be readily apparent to those skilled in the art, certain conventional elements and steps have been presented with lesser detail, while the drawings and the specification describe in greater detail other elements and steps pertinent to understanding the invention. The following embodiments are not intended to define limits as to the structure or method of the invention, but only to provide exemplary constructions. The embodiments are permissive rather than mandatory and are illustrative rather than exhaustive.

[0043] A method and system are described for constructing a depth map. Estimating depths from imagery commonly begins with application of a stereo matching algorithm to construct a disparity map. Stereo correspondence is determined for pixels in a pair of images taken of the same scene from different viewpoints. Depth estimations are based on differences in coordinate positions of the corresponding pixels in the two images. These differences, each referred to as a disparity, are assimilated and processed to form a depth map. Typically, the two images are acquired at the same time with two cameras residing in the same lateral plane, although a depth map may also be determined from correspondence between images of a scene captured at different times, with spatial differences occurring between corresponding pixels in a lateral plane. Generally, for depth estimations, most of the pixels of interest in one image will have a corresponding pixel in the other image. However, the disclosed systems and methods are not limited to embodiments which process data from multiple images.

[0044] Stereo matching algorithms are commonly described as local and global. Global methods consider the overall structure of the scene and smooth the image before addressing the cost optimization problem. Global methods address disparity by minimizing a global energy function for all values in the disparity map. Markov random Field modeling uses an iterative framework to ensure smooth disparity maps and high similarity between matching pixels. Generally, global methods are computationally intensive and difficult to apply in small real-time systems.

[0045] With local methods the initial matching cost is typically acquired more quickly but less accurately than with global methods. For example, in addition to the presence of noise in the pixel data, relevant portions of the scene may contain areas of relatively smooth texture which render depth determinations in pixel regions of interest unsatisfactory. Advantageously, the pixel-wise depth determinations may be based on computations for each given pixel value as a function of intensity values of other pixels within a window surrounding a given pixel. With local algorithms, the depth value at the pixel P may be based on intensity of grey values or color values. By basing the correspondence determination on the matching cost of pixels in a neighboring region (i.e., a window of pixels surrounding the given pixel P) a more accurate depth value can be determined for the pixel P. For example, with use of a statistical estimation, which only considers information in a local region, noise can be averaged out with little additional computational complexity. The disparity map value assignment may be based on Winner Take All (WTA) optimization. For each pixel, the corresponding disparity value with the minimum cost is assigned to that pixel. The matching cost is aggregated via a sum or an average over the support window.

[0046] The accuracy of depth mapping has been dependent on time intensive processing to achieve accurate identification of corresponding pixels. Many time critical machine vision applications require still higher levels of speed and accuracy for depth determinations than previously achievable. There is a need to develop systems and methods which achieve accurate depth information with rapid execution of data intensive, iterative computations.

[0047] Embodiments of the invention provide improvements in accuracy of local matching approaches, based on area-wide statistical computations. In one embodiment a processing system **10** applies improved processing techniques in conjunction with a matching algorithm to provide a more accurate disparity map at computation speeds suitable for real time applications. An exemplary stochastic approach comprises a combination of iterative refinements to generate an optimized disparity map.

[0048] While the invention can be practiced with numerous other matching algorithms, FIG. **1** illustrates application of an Adaptive Random Walk with Restart (ARWR) algorithm in a processing system which generates disparity maps based on pixel wise determination of minimum matching costs, i.e., the matching cost is a measure of how unlikely a disparity is indicative of the actual pixel correspondence. In this example, the ARWR algorithm is iteratively applied to optimize stereo matching. Image registration is enhanced with processing steps that address discontinuities and occlusions, and which apply additional filtering steps. Resulting matching costs bring local matching to improved levels of speed and accuracy.

[0049] When performing stereo matching with the system **10**, disparity computation is dependent on intensity values within finite windows in first and second reference images of a stereo image pair. The stereo algorithm initially performs pre-processing, followed by a matching cost computation which identifies an initial set of pixel correspondences based on lowest matching costs. This is followed by cost aggregation, disparity computation and a series of disparity refinement steps.

[0050] Pre-processing includes initial filtering or other operations applied to one or both images to increase speed and reduce complexity in generating the disparity map. Example operations which eliminate noise and photometric distortions are a conversion of the image data to grayscale values and application of a 3×3 Gaussian smoothing filter.

[0051] In one embodiment, the system **10** performs a sequence of six major process steps following pre-processing. The major steps are referred to as Process Blocks 1 through 6. Alternate embodiments of the major steps in the system **10** comprise some of the six Process Blocks or replace Process Blocks with variants, referred to as Alternate Process Blocks.

[0052] Local Matching Process Block 1 operates on a stereo image pair comprising first and second images **14**, **14'**, to initially determine pixel-wise correspondence based on the lowest matching cost. Second image **14'** is referred to as a reference image in relation to interim and final disparity maps. This is had by comparing portions of captured image structure in the two images based on pixel intensity values and use of a gradient matching technique. Processing within Cost Aggregation Process Block 2 begins with segmenting the images into superpixels based on the local matching. The superpixels become the smallest features for which the matching cost is calculated. For these embodiments, superpixels are defined about depth discontinuities based, for example, on a penalty function, or a requirement to preserve depth boundaries or intensity differences of neighboring superpixels. On this basis, with the superpixels being the smallest features for which the matching cost is calculated, the local correspondence determinations of Block 1 are aggregated to provide an initial disparity map.

[0053] In Optimization Process Block 3 the exemplary ARWR matching algorithm is iteratively applied as a matching algorithm to calculate an initial disparity map based on a superpixel-wise cost function. Mutual Structure Process Block 4 generates mutual structure information based on the initial disparity map obtained in Processing Block 3 and a reference image, e.g., one of the reference images **14**, **14'**. The mutual structure information is modified with weighted filtering in Filter Process Block 5 that provides pixel values in regions of occlusion or depth discontinuity present in the initial disparity map and over-writes the structure of the reference image on the disparity map.

[0054] To decrease blocky effects in the disparity map, Multi-Dimensional Convolution (MDC) Process Block 6 applies further filter treatment to the disparity map. Pixel information is converted into a two dimensional signal array on which sequences of convolutions are iteratively performed.

[0055] Local matching based on lowest matching cost may be accomplished with a variety of techniques. For the example process illustrated in Block 1, the initial local matching costs are based on a pixel-wise determination of lowest costs. The pixel-wise matching results of a census-based matching operation **22** (also referred to as a census transform operation) are combined with the pixel-wise matching results of a vertical gradient image filter operation **24** and the pixel-wise matching results of a horizontal gradient image filter operation.

[0056] The census-based matching operation **22** is typically performed with a non-parametric local transform which maps the intensity values of neighboring pixels located within a predefined window surrounding a central pixel, P, into a bit string to characterize the image structure. For every pixel, P, a binary string, referred to as a census signature, may be calculated by comparing the grey value of the pixel with grey values of neighboring pixels in the window. The Census Transform relies on the relative ordering of local intensity values in each window, and not on the intensity values themselves to map the intensity values of the pixels within the window into the bit string to capture image structure. The center pixel's intensity value is replaced by the bit string composed of a set of values based on Boolean comparisons such that in a square window, moving left to right,

```
If (Current Pixel Intensity < Centre Pixel Intensity):
    Boolean bit=0
        else
    Boolean bit=1
```

[0057] The matching cost is computed using the Hamming distance of two binary vectors.

[0058] Summarily, when the value of a neighboring pixel $P_{i,j}$ is less than the value of the central pixel, the corresponding value mapped into the binary string is set to zero; and when the value of a neighboring pixel $P_{i,j}$ is greater than the value of the central pixel, the corresponding value mapped into the binary string is set to one. The census transformation performs well, even when the image structure contains radiometric variations due to specular reflections.

[0059] However, the census-based matching can introduce errors, particularly in areas of a scene having repetitive or similar texture patterns. One source of error with stereo correspondence methods is that smoothness assumptions are not valid in depth discontinuity regions, e.g., when the areas contain edges indicative of depth variations. Where disparity values between the foreground and background structure vary, the depth boundaries are difficult to resolve and appear blurred due to perceived smoothness. The absence of texture is not necessarily a reliable indicator of an absence of depth.

[0060] Because image intensity values are not always indicative of changes in depth, pixel intensity values encoded in census transform bit strings can contribute to errors in pixel matching. To overcome this problem, gradient image matching is applied with, for example, vertical and horizontal 3×3 or 5×5 Sobel filters, also referred to as Sobel-Feldman operators. The operators yield gradient magnitudes which emphasize regions of high spatial frequency to facilitate edge detection and more accurate correspondence. Noting that similarity criteria in stereo matching primarily apply to Lambertian surfaces, another advantage of employing gradient image matching is that matching costs estimated with the processing according to Block 1 are less sensitive to the spatial variability of specular reflections and are, therefore, less viewpoint dependent when traditional stereo correspondence methods are unable to accurately calculate disparity values.

[0061] Because the horizontal and vertical gradient image filter operations **24** and **26** indicate directional change in the intensity or color in an image, the resulting gradient images may be used to extract edge information from the images. Gradient images are created from an original image **14** or **14'** by convolving with a filter, such as the Sobel filter. Each pixel of the gradient image **24** or **26** measures the change in intensity of that same point in the original image in a given direction to provide the full range of change in both dimensions. Pixels with relatively large gradient values are candidate edge pixels, and the pixels with the largest gradient values in the direction of the gradient may be deemed edge pixels. Gradient image data is also useful for robust feature and texture matching.

[0062] In one example, the Sobel operator uses two 3×3 kernels which are convolved with the original image to calculate approximations of the derivatives—one for horizontal changes, and one for vertical. Referring to the image to be operated on (e.g., image **14** or **14'**) as I, we calculate two derivatives indicative of horizontal and vertical rates of change in image intensity, each with a square kernel of odd size.

[0063] The horizontal image gradient values are computed by convolving I with a kernel G_x. For a kernel size of 3, G_x would be computed as:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * I \tag{1}$$

The horizontal image gradient values are computed by convolving I with a kernel G_y. For a kernel size of 3, G_y would be computed as:

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * I \tag{2}$$

At each point of the image we calculate an approximation of the gradient magnitude, G, at that point by combining G_x and G_y:

$$G=\sqrt{G_x^2+G_y^2} \tag{3}$$

[0064] The combination of census transform matching **22** and gradient image matching **24**, **26** renders the local matching method more robust on non-Lambertian surfaces. The calculated census transform values and vertical and horizontal gradient values of G are combined with a weighting factor to create a pixel-wise combined matching cost CMC. The weighting factors are selected to balance the influence of the census and gradient components. The result is then truncated to limit influence of outliers. Summarily, the gradient image matching reveals structure such as edges and corners of high spatial frequency to facilitate edge detection and more accurate correspondence.

[0065] Local matching costs are obtainable with other component operations, including the rank transform, normalized cross-correlation, absolute intensity difference, squared intensity difference, and mutual information. In one series of variants of Processing Block 1 initial matching costs are calculated for each pixel, P, with an additive combination of the component operations. For embodiments which iteratively apply an ARWR matching algorithm, the initial matching cost is calculated pixel-wise by employing methods most suitable to accurate local matching as a precursor to deriving the disparity map with, for example, superpixel segmentation.

[0066] Optimizing local matching in Process Block 1 is limited as it is based on a combination of operations which are applied as a weighted sum. That is, neither combination can fully influence the result, and inclusion of additional operations further limits the influence of all operations. The weighted combination of the census transformation and the gradient image filter operations provide improved performance for an image structure that contains both radiometric variations due to specular reflections and edge regions that require detection with a gradient filter operation. However, similarity criteria used in stereo matching are only strictly valid for surfaces exhibiting Lambertian reflectance characteristics. Specular reflections, being viewpoint dependent, can cause large intensity differences in values between corresponding pixels. In the presence of specular reflection, traditional stereo methods are, at times, unable to establish correspondence with acceptable accuracy. Further improvements in correspondence determinations are attained with refinements to or addition of other process steps.

[0067] The aggregation step applies pixel matching costs over a region to reduce correspondence errors and improve the overall accuracy of the stereo matching. Improvements in accuracy and speed are highly dependent on the operations incorporated in the cost aggregation step.

[0068] For example, prior to accumulating matching costs the cost aggregation process may replace the cost of assigning disparity d to a given pixel, P, with the average cost of assigning d to all pixels in a square window centered at the pixel P. This simplistic square-window approach implicitly assumes that all pixels in the square window have disparity values similar to that of the center pixel. The aggregated cost for the pixel, P, may be calculated as a sum of the costs in a 3×3 square window centered about that pixel. Processing with the square-window approach for cost aggregation is time intensive. Although it is based on assigning an average cost to each pixel, the approach may aggregate matching costs among all pixels.

[0069] However, it is well known that when groups of adjoining pixels are clustered into super-pixels, and intensity values of super-pixels are synthesized from the constituent pixel-wise data, cost aggregation based on the super-pixels becomes more robust to variations caused by artifact—including variations due to specular reflections. With the super-pixels becoming the smallest parts of an image to be matched, determining cost aggregation with the super-pixel values is an advantageous extension over pixel-wise aggregation of matched data. Being statistically based, intensity values of adjoining super-pixels do not exhibit undesirable variations to the same extent as individual pixel values of adjoining pixels. Use of super-pixels also results in reduced memory requirements in the remainder of the processing blocks of the system **10**. To this end, Cost Aggregation Process Block 2 employs a Simple Linear Iterative Clustering (SLIC) algorithm to define super-pixels. An exemplary super-pixel wise cost function is the mean of all cost values of the pixels inside the super-pixel S:

$$F_r(S, d) = \frac{1}{n_s} \sum\nolimits_{(u,v)\in S} P_r(u, v, d) \tag{4}$$

where: F_r is the cost of the super–pixel S, n_s is the number of pixels in the super-pixel S, d is the disparity between corresponding pixels and $P_r(u,v,d)$ is the pixel-wise matching cost calculated by the census transform and image gradient matching operations.

[0070] Disparity values are heavily influenced by the manner in which cost aggregation process is performed. Cost aggregation based on super-pixeling provides improved matching costs over larger areas. Yet, even with economies resulting from use of the SLIC algorithm to define super-pixels, the processing requirements for cost aggregation in Process Block 2 are time intensive (e.g., image-matching costs must be combined to obtain a more reliable estimate of matching costs over an image region). The necessary high speed computation and memory bandwidth presents an impediment to deploying stereo matching in real-time applications, including automated braking systems, steering of self-driving cars and 3-D scene reconstruction. With the demand for greater accuracy and speed in disparity maps generated at video frame rates, design of an improved cost aggregation methodology is seen as a critically important element to improving the overall performance of the matching algorithm.

[0071] In the past, processing requirements have been based, in part, on requirements that superpixels be defined with a density which avoids artifact that degrades depth map accuracy. A feature of the invention is based on recognition that cost aggregation may be performed with less regard to limiting the sizes of superpixels, in order to increase computational speed, but without degradation in depth map accuracy.

A. Disparity Map Optimization with an Adaptive Algorithm Block 3

[0072] In the Optimization Process Block 3 of the system **10**, the ARWR algorithm, illustrated as an exemplary matching algorithm, is iteratively applied to calculate an initial disparity map, **165**, based on a superpixel-wise cost function, where superpixel segmentation is determined in Cost Aggregation Processing Block 2 with, for example, the SLIC algorithm or the LRW algorithm. Iterative updating of Processing Block 3 continues until convergence occurs in order to achieve optimum disparity with respect to regions of occlusion and discontinuity. The ARWR algorithm updates the matching cost adaptively by accounting for positions of super-pixels in the regions of occlusion and depth discontinuity. To recover smoothness failures in these regions the ARWR algorithm may, optionally, incorporate a visibility constraint or a data fidelity term.

[0073] The visibility constraint accounts for the absence of pixel correspondence in occlusion regions. The iterative process may include a visibility term in the form of a multiplier, M, which requires that an occluded pixel (e.g., superpixel) not be associated with a matching pixel on the reference image **14**, and a non-occluded superpixel have at least one candidate matching pixel on the reference image **14**. See S. Lee, et al., "Robust Stereo Matching Using Adaptive Random Walk with Restart Algorithm," *Image and Vision Computing*, vol. 37, pp 1-11 (2015).

[0074] The multiplier, M, is zero when a pixel is occluded to reflect that there is no matching pixel in the disparity image; and allows for non-occluded pixels to each have at least one match. That is, for super-pixels having positions in regions containing an occlusion or a depth discontinuity, the cost function is adaptively updated with an iterative application of the algorithm until there is convergence of matching costs. The occluded regions may, for example, be detected by performing consistency checks between images. If the disparity value is not consistent between a reference image and a target image, a superpixel is determined to be occluded. After superpixels are iteratively validated as non-occlusive, the results are mapped into a validation vector and the matching costs are multiplied by the validation vector. See S. Lee, et al, p. 5.

[0075] In regions where disparity values vary, smoothness assumptions can blur boundaries between foreground and background depths, especially when variations in disparity values are substantial. Intensity differences between superpixels along depth boundaries are preserved by reducing the smoothness constraint in regions of depth discontinuity. It has been proposed to do so by modifying the standard Random Walk with Restart (RWR) iterative algorithm with a data fidelity term, Ψ, based on a threshold change in the disparity value. See S. Lee, et al, p. 6. This allows preservation of the depth discontinuity. By so preserving the depth discontinuity, a more accurate or optimal disparity value is identified for each superpixel based on a refined calculation of the updated matching cost. The data fidelity term measures the degree of similarity between two pixels (or regions) in terms of intensity. It preserves depth boundaries and is effective at the boundaries of objects where there is a unique match or there are relatively few likely matches.

[0076] The Random Walk with Restart (RWR) method for correspondence matching is based on determining matching costs between pixels (i.e., the probability that points in different images are in true correspondence). The random walker iteratively transmits from an initial node to another node in its neighborhood with the probability that is proportional to the edge weight between them. Also at each step, it has a restarting probability c to return to the initial node. $\vec{r}_i$, the relevance score of node j with respect to node i, is based on a random particle iteratively transmitted from node i to its neighborhood with the probability proportional to the edge weights $\tilde{W}$:

$$\vec{r}_i = c\tilde{W}\vec{r}_i + (1-c)\vec{e}_i \tag{10)(5}$$

[0077] At each step, there is a probability c of a return to the node i. The relevance score of node j with respect to the node i is defined as the steady-state probability $r_{i,j}$ that the walker will finally stay at node j. The iteratively updated matching cost, $X_{t+1}{}^{d}$, is given as:

$$X_{t+1}{}^{d} = c\overline{W}X_t^d + (1-c)X_0{}^{d} \tag{11)(6}$$

where $X_0{}^{d}=[F(s,d)]_{k\times1}$. represents the initial matching cost, $X_t{}^{d}$ denotes the updated matching cost, t is the number of iterations, k is the number of super-pixels and (1−c) is the restart probability. F(s,d) is the super-pixel wise cost function with $F_r(S,d)$ being the mean of all cost values of the pixels inside a super-pixel S.

$\overline{W}=[w_{ij}]_{k\times k}$, which is the weighted matrix, comprises the edge weights w_{ij}, which are influenced by the intensity similarity between neighboring super-pixels.

$$w_{ij} = \exp\left(-\frac{(I(s_i) - I(s_j))^2}{\sigma_e}\right) \tag{12)(7}$$

where $l(s_i)$ and $l(s_j)$ are the intensities of the i-th and j-th super-pixels and σ_e is a parameter that controls the shape of the function. The intensity of super-pixels is computed by averaging the intensity of the corresponding pixels.

[0078] Equation (6) is iterated until convergence, which is influenced by updating of the weights w_{ij}. Convergence is reached when the L_2 norm of successive estimates of $X_{t+1}{}^{d}$ is below a threshold ξ, or when a maximum iteration step m is reached. The L_2 norm of a vector is the square root of the sum of the absolute values squared.

[0079] Optimization Process Block 3 may incorporates a fidelity term, $\Psi_t{}^{d}$, and a visibility term, $V_t{}^{d}$, into the matching cost, $X_{t+1}{}^{d}$. See Equation (8) which weights the fidelity term, $\Psi_t{}^{d}$, and the visibility term, $V_t{}^{d}$, with respect to one another with a factor λ:

$$X_{t+1}{}^{d} = c\overline{W}((1-\lambda)V_t^d + \lambda\Psi_t^d) + (1-c)X_0{}^{d} \tag{13)(8}$$

[0080] Based on Equation (8), the final disparity map is computed by combining the super-pixel and pixel-wise matching costs:

$$\hat{d} = \arg_d\min(X_t^d(s) + \gamma P(u,v,d)) \tag{14)(9}$$

where s is the super-pixel corresponding to the pixel (u,v) and γ represents the weighting of the super-pixels and pixel-wise matching cost. In another embodiment the visibility term is not included in the cost function, resulting in

$$X_{t+1}{}^{d} = c\overline{W}(\lambda\Psi_t{}^{d}) + (1-c)X_0{}^{d} \tag{15)(10}$$

[0081] The interim disparity map, **165**$_i$, generated by Optimization Processing Block 3 may be given as the combination of superpixel and pixel wise matching costs similar to the prior art approach taken to construct a final disparity map. See S. Lee, et al, p. 6. Summarily, an embodiment of an algorithm for developing the interim disparity map includes the following sequence of steps:

[0082] 1. Computing the local matching cost for each pixel using the truncated weighted sum of the census transform and gradient image matching.

[0083] 2. Aggregating the matching costs inside each superpixel.

[0084] 3. Computing the optional visibility term based on the current matching cost.

[0085] 4. Computing the fidelity term using the robust penalty function.

[0086] 5. Updating the matching costs.

[0087] 6. Iterating Steps 3, 4 and 5 multiple times to determine the final disparity from the minimum cost.

[0088] A first post processing stage of the system **10** generates an interim disparity map using mutual structure information (**166**, **167**) in a DJW Median Filter operation performed on the initial disparity map **165** and the first reference RGB image **161**. The combination of Processing Blocks 4 and 5 transfer structural information from the reference image **161** to the disparity map **165**, in essence guiding the filter to restore edges in the depth map. Registration between two images of the same scene is optimized based on sequential alignment between each array of pixel data. A final disparity map **174** is then developed in a second processing stage by an iterative sequence of vertical and horizontal interpolated convolutions. See Processing Block 6.

[0089] Two forms of a mutual structure calculation are used as input information to the Dynamic Joint Weighted Median Filter (DJWMF) operation **205** of Process Block 5:

[0090] (1) A Similarity Map (SM) **166** provides a measure of similarity between the initial disparity map **165** and the first RGB reference image **161**. FIG. **6**C illustrates creation of SM **166**. FIG. **6**F illustrates an exemplary application of SM **166** to determine values of an adjustable window size during the DJWMF operation **205** on the Initial Disparity Map **165**.

[0091] (2) Mutual Feature Map (MFM) **167** is the result of an additive mutual structure calculation from which a Dynamic Weighted Allocation is created for application in the filter operation **205**. FIG. **6**C summarizes creation of MFM **167** for determination of a weighting function. FIG. **6**F illustrates a portion of a process which applies the weighting function derived from the MFM data to determine each median value. Data in the Mutual Feature Map (MFM) **167** is intensity map data, similar to the initial disparity map data, but which includes structure present in the RGB reference Image **161**, including edges and corner features.

[0092] The Similarity Map (SM) **166** is a map representation of differences and similarities between the Initial Disparity Map **165** and the RGB reference image **161**. SM **166** indicates how structurally similar a disparity map and a reference image are, without including in the SM data the structure of either the disparity map or the reference image. This is to be distinguished from the Mutual Feature Map **167** which contains structure features that can be transferred to a disparity map. The similarity values in SM **166** are used as a basis to determine window sizes to be assigned for each filter operation performed on a disparity map. Thus the Similarity Map determines final specifications for operation of a weighted median filter on a disparity map. A Mutual Feature Map **167** cannot be used for this purpose because it contains structure.

[0093] A Structural SiMilarity (SSIM) method, based on the Similarity Map **166**, provides measures of the similarity between two images based on computation of an index which can be viewed as a quality measure of one of the images relative to the other image. The other image may be regarded as a standard of quality, i.e., corresponding to an accurate representation of the scene from which the image is derived, e.g., the ground truth. The similarity map **166** is applied to identify areas in a disparity map which could be filtered by relatively large window sizes and areas in the disparity map which could be filtered by smaller window sizes. A window sizing process is provided which applies elements of the Similarity Map **166** to selectively process areas of the Initial Disparity Map **165** having higher similarity values with an individual filter operation based on a larger window size, facilitating faster computational time for the filter operation performed over the entire disparity map; while areas with lower similarity values are processed with an individual filter operation based on a smaller window size, contributing to increased computational time over the entire disparity map. Elements of the Similarity Map **166** applied to selectively process areas of a disparity map may be computed for patches of superpixels or for groups of patches. Embodiments of the method discriminate between patch areas with higher similarity values and patch areas with lower similarity values to more optimally reduce overall processing time required for the filter operations to generate a Filtered Disparity Map **172** as an output of Process Block 5.

[0094] Referring to FIG. **6**C, in one embodiment, the Similarity Map **166** is created based on multiple functions which provide measures of the similarity between the initial disparity map **165** and the first RGB reference image **161**. An exemplary Structural SIMilarity (SSIM) index provides a pixel-by-pixel measure of similarity between the two sets of image data. As noted for the SSIM method, the SSIM index may be regarded as a quality measure of the Initial Disparity Map **165** relative to the first RGB reference image **161**, or in comparison to other image data, provided the other image data is of suitable accuracy for achieving acceptable depth map accuracy. The illustrated embodiment of the SSIM index is a function of a luminance term, l, a contrast term, fvc, and a structural term, s, in which D and I are, respectively, the initial disparity map **165** and the first reference RGB image **161**. Also, D_p and I_p (referred to as D_p and I_p in the computer code, respectively) are the pixel intensities in the initial disparity map **165** and the first reference RGB image **161**, respectively. One embodiment of the Structural SIMilarity (SSIM) index is given by

$$\mathrm{SSIM}(D,I)=[l(D,I)]^{\alpha}\cdot[c(D,I)]^{\beta}\cdot[s(D,I)]^{\gamma} \quad \text{(I)(11)}$$

where:

$$l(Dp,I)=\frac{2\mu_D\mu_I+C_1}{\mu_D^2+\mu_I^2+C_1} \quad \text{(II)}$$

-continued

$$fvc(Dp,I)=\frac{2\sigma_D\sigma_I+C_2}{\sigma_D^2+\sigma_I^2+C_2} \quad \text{(III)}$$

$$s(Dp,I)=\frac{\sigma_{DI}+C_3}{\sigma_D\sigma_I+C_3}, \quad \text{(IV)}$$

and where μ_D, μ_I, σ_D, σ_I and σ_{DI} are, respectively, the local means, standard deviations, and cross-covariance for images D and I (e.g., at the level of a patch group, a patch of pixels or at the super pixel level); and C_1, C_2, C_3 are regularization constants for the luminance, contrast, and structural terms, respectively. Terms α, β and γ are exponents for the luminance, contrast, and structural terms, respectively.

[0095] Where N is the number of the patches (e.g., of size 11×11) extracted from each image D and I:

[0096] (1) the local means for the images D and I may be calculated by applying a Gaussian filter (e.g., of size 11×11 with standard deviation 1.5) as follows:

(*a*)

$$\mu_D=\frac{1}{N}\sum_{i=1}^{N}D_i \quad (12)$$

(*b*)

$$\mu_I=\frac{1}{N}\sum_{i=1}^{N}I_i \quad (13)$$

[0097] (2) the standard deviations be calculated as follows:

(*c*)

$$\text{standard deviations in } D=\sigma_D=\left(\frac{1}{N-1}\sum_{i=1}^{N}(D_i-\mu_D)^2\right)^{\frac{1}{2}} \quad (14)$$

(*d*)

$$\text{standard deviations in } I=\sigma_I=\left(\frac{1}{N-1}\sum_{i=1}^{N}(I_i-\mu_I)^2\right)^{\frac{1}{2}} \quad (15)$$

and

[0098] (3) the cross covariance may be calculated as

(*e*)

$$\sigma_{DI}=\frac{1}{N-1}\sum_{i=1}^{N}(D_i-\mu_D)(I_i-\mu_I). \quad (16)$$

[0099] C_2, C_3 are regularization constants for the luminance, contrast, and structural terms. In the luminance term, $C_1=(K_1L)^2$, L is the dynamic range of the pixel values (e.g., 255 for 8-bit grayscale image) and K_1 is a small constant value (e.g., $K_1=0.01$). In the contrast term, $C_2=(K_2L)^2$, L is again the dynamic range of the pixel values and K_2 is a small constant value (e.g., K_2=0.03). In structural term C_3=C_2/2.

[0100] Computation of the above luminance, contrast and structural terms II, III and IV to calculate the SSIM index is also described in the literature. See, for example, Wang, et al. "Image Quality Assessment: From Error Visibility To Structural Similarity" IEEE Transactions On Image Processing 13.4 (2004): 600-612. See, also, Z. Wang, et al., "Multiscale structural similarity for image quality assessment," Invited Paper, IEEE Asilomar Conference on Signals, Systems and Computers, November 2003; and also see Wang et al., "A Universal Image Quality Index," in IEEE Signal Processing Letters, vol. 9, no. 3, pp. 81-84, March 2002.

[0101] In one embodiment for performing a weighted median filter operation, the SSIM index is applied to create a SSIM map as the Similarity Map **166**, comprising a value $SSIM_{ij}$ for each pair of corresponding pixels $p_{i,j}$ present in both the RGB image **161** and the Initial Disparity Map **165**. These SSIM index values are used for adaptive window sizing. Initially the process starts with a relatively small window size W_{min} (e.g., the size 3*3 corresponding to (2j+1) with j=1) as an initial and minimum candidate size for an adaptive window. With $SSICM_{ij}$ denoting the SSIM Index value for one center pixel position inside the current window of the SSIM map, $SSIM_{min}$ is the minimum pixel value inside the current window and $SSIM_{max}$ is the maximum pixel value in the current window. Letting W be the current window size (e.g., 3*3); W_{max} be the maximum size of the adaptive window (e.g., the size 9*9 corresponding to (2j+1) with j=4); and $SSIM_{med}$ be the median pixel value determined for the current window, W, centered about the center pixel position, then: the proper window size is determined using the following steps:

[0102] a) If the inequality statement $SSIM_{min}$<$SSIM_{med}$<$SSIM_{max}$ is true, then the current window size is the chosen window size for performing a weighted median operation about the center pixel position corresponding to the SSIM position $SSICM_{ij}$.

[0103] b) If the inequality statement in step a) is not true, the size of the window is increased to (2j+1) by incrementing j: j=>j+1.

[0104] c) Next, steps a) and b) are repeated as necessary until either: (i) $SSIM_{med}$ is between $SSIM_{min}$ and $SSIM_{max}$; or (ii) the maximum window size is reached, in which case the maximum window size is the chosen window size for performing a weighted median operation about the center pixel position corresponding to the SSIM position $SSICM_{ij}$.

[0105] Reference to the word "median" in the context of $SSIM_{med}$ is not to be confused with the median attribute of the dynamic joint weighted median filter which calculates median values to construct the Filtered Disparity Map **174**. The "median" associated with $SSIM_{med}$ is in the context of adaptive window sizing and steps a), b) and c), i.e., the selection of a size for a window, W, which is later used during the DJWM filter operation **205**.

[0106] An advantageous embodiment has been described in which the Similarity Map **166** is derived by applying SSIM indices to create a SSIM map. Generally, applying a median filter with constant window size on an image might remove structural information which should be retained for improved depth map accuracy. The median filter operation may also retain unnecessary information. By choosing a relatively large window size, processing time for the median operation can be reduced, but some valuable information may be removed by operating the filter over a larger window size. On the other hand, by choosing a relatively small window size, the processing time for the entire median operation may be increased and unnecessary information might be retained.

[0107] Advantageously adaptive window sizing permits a more optimal selection of window sizes as the filter operation progresses over the disparity map. When the window sizes are adaptively chosen in an embodiment which uses the SSIM map as the Similarity Map **166**, the Similarity Map provides a measure of how similar corresponding patches in the Initial Disparity Map **165** and the RGB reference image **161** are in terms of structure. Information in the Similarity Map **166** is used to identify the areas of the data array which can advantageously be filtered by larger window sizes and the areas which should be filtered by smaller window sizes to preserve structural information. Ideally this process optimally determines areas with relatively high similarity values that can be filtered with relatively large window sizes for faster computational time, and optimally identifies only those areas with relatively low similarity values that should be filtered with application of smaller window sizes, at the cost of slower computational speeds, to restore or preserve important structural information. The result is an ability to balance a minimization of overall median filter computational time while keeping necessary information to achieve acceptable depth map accuracy.

[0108] According to another aspect of the invention, structure similarity of two images (i.e., the mutual-structure) is used to guide a median filtering operation, which is why the DJWMF operation **205** is referred to as a joint filtering process. The operation results in a Filtered Disparity Map **172**. With D and I again denoting, respectively, the Initial Disparity Map **165** and the first RGB image **161**, D_p and I_p denote the pixel or superpixel intensities in initial disparity map **165** and the first RGB image respectively. The structure similarity between the two images **165** and **161**, i.e., as embodied in the exemplary Mutual Feature Map (MFM) **167**, may be calculated based on cross covariance, normalized cross correlation, $N(D_p,I_p)$, or least-square regression. See FIG. **503**. The MFM **167** is applied during the DJWMF operation **205** on D, the Initial Disparity Map **165**. See FIG. **504**. This results in transfer of structural information from the reference RGB image **161** to the Initial Disparity Map **165** to restore edges and corners in the Initial Disparity Map **165** with improved efficiency over transferring the entire structure of the RGB image.

[0109] In one embodiment, D **165** and I **161** are treated as two ordered signal data arrays, each comprising M samples of pixels. A measure of the actual similarity between two images, based on patches in the images, may be calculated with the normalized cross covariance.

[0110] With the ordered arrays of signal data treated in like manner to a time series representation of data, a delay of W samples is iteratively imposed between corresponding data in the depth map image, D, and the reference image, I, to determine the cross-covariance between the pair of signals:

(1)

$$CC(W) = \frac{1}{M-1}\sum_{k=1}^{M}(D_{k-W}-\mu_D)(I_k-\mu_I), \tag{17}$$

where μ_D and μ_I are, respectively, for each time series, the mean value of data in the depth map image array and the mean value of data in the reference image array. When normalized, the cross-covariance, CC(W) becomes N(W), commonly referred to as the cross-correlation:

(2)

$$N(W) = \frac{CC(W)}{\sqrt{\sigma(D_p)\sigma(I_p)}}, \quad (18)$$

where $\sigma(D_p)$ and $\sigma(I_p)$ denote the variance of the pixel intensity in D and I, respectively.

[0111] After normalization the cross-correlation between D and I is:

(3)

$$N(D_p, I_p) = \frac{\text{cov}(D_p, I_p)}{\sqrt{\sigma(D_p)\sigma(I_p)}}, \quad (19)$$

where $\text{cov}(D_p,I_p)$ is the covariance of patch intensity between D and I. The variance of pixel intensity in the initial depth map and RGB image **161** are denoted by $\sigma(D_p)$ and $\sigma(I_p)$, respectively. The maximum value of $N(D_p,I_p)$ is 1 when two patches are with the same edges. Otherwise $|N(D_p,I_p)|<1$. Nonlinear computation makes it difficult to use the normalized cross-correlation directly in the process.

[0112] An alternate method of performing the nonlinear computations for normalized cross-correlations is based on the relationship between the normalized cross-correlation and the least-square regression to provide a more efficient route to maximizing similarity between images and accuracy in depth map estimates. If we consider H(p) as a patch of superpixels centered at pixel p, then the least-squared regression function ƒ(D,I,) between pixels in the two images D and I may be expressed as:

$$f(D,I,\alpha_p{}^1,\alpha_p{}^0)=\Sigma_{q\in H(p)}(\alpha_p{}^1 D_q+\alpha_p{}^0-I_q)^2, \quad (4)(20)$$

where q is a superpixel element in the patch of pixels H(p), and $\alpha_p{}^1$ and $\alpha_p{}^0$ are regression coefficients. The function $f(D,I,\alpha_p{}^1,\alpha_p{}^0)$ linearly represents an extent to which one patch of superpixels in D **165** corresponds with one patch of superpixels in I **161**. The minimum error, which corresponds to a maximum value for $N(D_p,I_p)$, based on optimized values of $\alpha_p{}^1$ and $\alpha_p{}^0$, is:

(5)

$$e(D_p, I_p)^2 = \min_{\alpha_p^1,\alpha_p^0} \frac{1}{|H|} f(D, I, \alpha_p^1, \alpha_p^0), \quad (21)$$

[0113] Based on Eqns (17) and (21), the relation between the mean square error and normalized cross correlation is:

$$e(D_p,I_p)=\sigma(I_p)(1-N(D_p,I_p)^2). \quad (6)(22)$$

The relation between the mean square error and normalized cross-correlation is described in Achanta, et al., "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 2274-2282, 2012.

[0114] When $|N(D_p,I_p)|=1$, two patches only contain mutual structure and $e(D_p,I_p)=0$.

[0115] So, using the same procedure as above:

(7)

$$e(I_p, D_p)^2 = \min_{b_p^1,b_p^0} \frac{1}{|H|} f(I, D, b_p^1, b_p^0), \quad (23)$$

where $\alpha_p{}^1$ and $\alpha_p{}^0$ are regression coefficients. Therefore $e(I_p,D_p)=0$ when $|N(D_p,I_p)|=1$.

[0116] The final patch similarity measure is defined as the sum of the functions defined in Eqns (19) and Eq (21) as: $e(D_p,I_p)^2+e(I_p,D_p)^2$. Based on the foregoing, both the Mutual Feature Map **167** and the patch similarity are obtained from application of:

$$S(D_p,I_p)=e(D_p,I_p)^2+e(I_p,D_p)^2 \quad (8A)(24)$$

Based on Eqns. (21) and (22), and considering that $N(D_p, I_p)=N(I_p,D_p)$, the pixel similarity, S, with which the patch similarity is determined, can also be expressed as:

$$S(D_p,I_p)=(\sigma(D_p)^2+\sigma(I_p)^2)(1-N(D_p,I_p)^2)^2 \quad (8B)(25)$$

[0117] When, for superpixels in corresponding patches, $|N(D_p,I_p)|$ approaches one, $S(D_p,I_p)$ approaches zero, indicating that two patches have common edges. When the patches don't clearly contain common edges, then $\sigma(D_p)$ and $\sigma(I_p)$ are relatively small values and the output of $S(D_p,I_p)$ is a small value.

[0118] Based on the above analysis, the Mutual Feature Map **167**, also referred to as S_s, is the sum of pixel or patch level information:

$$S_s(D,I,\alpha,b)=\Sigma_p(f(D,I,\alpha_p{}^1,\alpha_p{}^0)+f(I,D,b_p{}^1,b_p{}^0)), \quad (8)(26)$$

where α and b are the regression coefficient sets of $\{\alpha_p{}^1, \alpha_p{}^0\}$ and $\{b_p{}^1,b_p{}^0\}$, respectively.

[0119] The Mutual Feature Map **167** is used for weight allocation. An exemplary and typical choice for weighting allocation is based on the affinity of p and a in the Mutual Feature Map S expressed as:

$$w_{pq}=g(S(p),S(q)) \quad (9)(27)$$

where S(p) and S(q) are features at pixels p and q in S. A reasonable choice for g is a Gaussian function, a common preference for affinity measures:

$$\exp\{-\|S(p)-S(q)\|\} \quad (10)(28)$$

[0120] The Initial Disparity Map **165**, also referred to as a target image, typically lacks structural details, such as well-defined edges and corner features. The deficiencies may be due to noise or insufficient resolution or occlusions. The DJWMF operation **205** of Process Block 5 restores the missing detail by utilizing the RGB reference image **161** to provide structural guidance by which details missing from the initial disparity map are restored, while avoiding further incorporation of artifact. In the past, filters applied to transfer structural features of a guidance image have considered information in the guidance image without addressing inconsistencies between that information and information in the target image. Hence the operation could transfer incorrect contents to the target image. Prior approaches to avoid transfer of incorrect information have considered the contents of both the target and the reference image used to provide guidance with dynamically changing guidance. For example, in a process that minimizes a global objective function, the guidance signals may be updated during iterative calculations of the objective function to preserve mutually consistent structures while suppressing those structures not commonly shared by both images.

[0121] Weighted median filter operations according to the invention can provide for a more optimal restoration of features. With the first stage of disparity map refinement operations utilizing both the Similarity Map (SM) **166** and the Mutual Feature Map (MFM) **167** generated in the Mutual Structure Processing Block 4, the DJWMF operation **205** can selectively and optimally transfer the most useful structural information from the reference RGB image for depth refinement on a patch level. This enables restoration in the Interim Disparity Map **170** of physically important features, corresponding to object details evident in the reference RGB image **161**. To this end, filter weightings applied in the median filter operation are dynamically allocated, e.g., with patch level or superpixel level weightings w_{pq} based on $S_s(D, I, \alpha, b)$ and advantageously provided in a joint histogram. Similarities or differences in structure on, for example, a patch level, between the disparity and reference maps can play a more effective role in the weighting process to effect restoration of edges and corners without introduction of additional artifact. Select embodiments of the methods which dynamically apply weightings to the median filter can fill regions of occlusion or depth discontinuity. When pixel data includes large erroneous deviations, the values resulting from the dynamic weighting can be less distorted than values which would result from application of a standard median filter which only base the new pixel value on a value in the defined window. Consequently, edge regions processed with the dynamically weighted median filter can restore a level of sharpness which may not be achievable with a standard median filter or a median filter which does not provide variable weight allocations.

[0122] To reduce the relatively expensive computational cost of sorting values in a median filter operation, embodiments of the invention employ histogram techniques to represent the distribution of the image intensities within each adaptive window, i.e., indicating how many pixels in an image have a particular intensity value, V. The histogram is created by incrementing the number of pixels assigned to each bin according to the bin intensity level. Each time a pixel having a particular intensity is encountered, the number of pixels assigned to the corresponding bin is increased by one. For discrete signals the median value is computed from a histogram h(p,.) that provides the population around the position of a center pixel p located at a position (x,y):

$$h_D(p,i)=h_D(i)=\Sigma_{p'\epsilon W_p}\delta(V(p')-i), \qquad (9a)(29)$$

where W_p is a local window of dimensions 2j+1 around p, V is the pixel value and i, the discrete bin index, is an integer number referring to the bin position in the histogram. For example, a bin i,j in the histogram has an index i,j corresponding to one value in a monotonically increasing sequence of intensity values, V, each value mapped to an assigned bin. δ(.) the Kronecker delta function, is one when the argument is zero and is otherwise zero.

[0123] There are 2 main iterations in Eqn (29). By way of example, the first iteration may range from 0 to 255, which corresponds to a range of pixel intensity values. For each such iteration there may be a sub-iteration of $(2j+1)^2$ over pixel values in a window (e.g., from 0 to 8 for a 3*3 window W_p of pixels). The following illustration of a median operation can readily be applied to a dynamically weighted median operation as well by including a weighting factor such as ω(p,p') as noted below.

[0124] For a window W_p, in an image D, the associated histogram h is based on the number of pixels, N, in W_p and with pixel values ranging from 0 to 255. The term O_{Mid} corresponds to the middle pixel in the ordered bin sequence of the N data points:

$$O_{Mid} = \frac{N-1}{2}$$

where N is odd. The median based on the histogram can be computed for a window W_p of exemplary values V

$$W = \begin{array}{|c|c|c|}\hline 156 & 89 & 75 \\ \hline 190 & 204 & 89 \\ \hline 89 & 75 & 255 \\ \hline \end{array}$$

as:

```
Function m= medhist (W)
{
// Input: Window $W_p$ storing N pixels
// Output: Median of the window
csum=0; //csum means the cumulative function of the histogram
for i=0 to 255
    for j=0 to N
        h(i) += δ(W(j)−i);   //δ(•) is 1 when the argument is 0, otherwise
        it is 0
    end for
    If hn[i] > 0 then
    csum += hn[i];
```

[0125] The above median determination first creates the histogram from the input data set. Then, the cumulative function of the histogram is evaluated by incrementing the index over the example range of pixel values from 0 to 255. When the cumulative function reaches the middle order, O_{Mid}, the current index is the median value for the current window data set required.

[0126] However, because a standard median filter processes all pixel values in a window equally the operation may introduce artifact such as the giving of curvature to a sharp corner or the removal of thin structures. For this reason, elements in a median filter according to embodiments of the invention are dynamically weighted so that certain pixel values, V, in the windows are, based on affinity, selectively weighted and the filter operation accumulates a series weighted median pixel intensity value for the Filtered Disparity Map **172**. In one embodiment, given a weighting function ω(p,p'), the weighted local histogram with weighted pixels p' in a selected window W_p within the disparity map data is given by:

$$h(p,i)=\Sigma_{p'\epsilon W_p}\omega(p,p')\delta(V(p')-i) \qquad (10)(30)$$

where i and δ(.) are as previously stated.

[0127] By accumulating h(p,i) the weighted median value is obtained. That is, as described for the unweighted median filter operation, the cumulative function of the histogram h(p,i) in Eqn (30) is evaluated by incrementing the index over the example range of pixel values from 0 to 255. When the cumulative function reaches the middle order, O_{Mid}, the current index is the median value for the current window data set required.

[0128] For the joint median filter on a depth map D with a group S of segments, the local histogram is defined as:

$$h_D(p,i)=h_D(i)=\Sigma_{p'\epsilon W_p\cap S_p}\delta(D(p')-i), \qquad (12)(31)$$

where $S_p\epsilon S$ is the segment containing pixel p. Image segments in S represent, for example, the edge information of the reference image **161**.

[0129] For the joint weighted median filter applied to a disparity map D with a group S of segments and a weighting function ω(p,p'), the local histogram is defined as:

$$h_D(p,i)=h_D(i)=\Sigma_{p'\epsilon W_p\cap S_p}\omega(p,p')\delta(D(p')-i) \qquad (3)(32)$$

In the context of a histogram of a window, the word "segment" corresponds to a group of pixels representing a feature like an edge or corner in the reference image.

[0130] By accumulating the weighted median filter value is obtained, as described for the unweighted median filter operation.

[0131] An exemplary process for applying the Mutual Feature Map **167** in performing the DJWMF operation **205** is described for 3*3 window size operations. When a joint histogram combines colour information with intensity gradient information, a given pixel in an image has a colour (in the discretized range 0 . . . $n_{colour}-1$) and an intensity gradient (in the discretized range 0 . . . $n_{gradient}-1$). The joint histogram for color and intensity gradient contains n_{colour} by $n_{gradient}$ entries. Each entry corresponds to a particular colour and a particular intensity gradient. The value stored in this entry is the number of pixels in the image with that colour and intensity gradient. In the DJWMF operation **205**, values inside a window are sorted in order to determine the median. Sorting these values is an expensive computation in an iterative process. It is necessary to sort the values inside the window (e.g., by considering the repeated numbers) and then counting the values to find the median. This process can be performed faster by applying a Joint Histogram. Assuming a pixel q located inside the window in an image D(q) is associated with a feature S(q), and given N_D different pixel values, the pixel value index for D(q) is denoted as d and the pixel value index for S(q) is denoted as s. The total number of different features is denoted as N_S. In a 2D joint-histogram H, pixel q is put into the histogram bin H(d, s) in the d−th row and s−th column. Thus, the whole joint-histogram is constructed as:

$$H(d,s)=\#\{q\epsilon R(p)|D(q)=D_d,S(q)=S_s\}, \qquad (33)$$

where # is an operator which counts the number of elements and R(p) denotes the local window radius r of centre pixel p. This counting scheme enables fast weight computation even when the window shifts.

[0132] For a disparity map D having a window with values of:

$$D=\begin{array}{|c|c|c|}\hline 5&2&3\\\hline 5&5&6\\\hline 7&8&9\\\hline\end{array}$$

and a Mutual Feature Map, S, having a window with values of:

$$S=\begin{array}{|c|c|c|}\hline 1&2&3\\\hline 4&5&6\\\hline 7&8&9\\\hline\end{array}$$

the joint histogram of D and S has 9*9 size and it contains the number of pixels in the image that are described by a particular combination of feature values. For clarity in reading the matrices, D and S are reshaped as:

$$D=\begin{array}{|c|c|c|c|c|c|c|c|c|}\hline 5&2&3&5&5&6&7&8&9\\\hline\end{array}$$

$$S=\begin{array}{|c|c|c|c|c|c|c|c|c|}\hline 1&2&3&4&5&6&7&8&9\\\hline\end{array}$$

H is the joint histogram of D and S:

$$H=\begin{array}{|c|c|c|c|c|c|c|c|c|}\hline 0&0&0&0&0&0&0&0&0\\\hline 0&1&0&0&0&0&0&0&0\\\hline 0&0&1&0&0&0&0&0&0\\\hline 0&0&0&0&0&0&0&0&0\\\hline 1&0&0&1&1&0&0&0&0\\\hline 0&0&0&0&0&1&0&0&0\\\hline 0&0&0&0&0&0&1&0&0\\\hline 0&0&0&0&0&0&0&1&0\\\hline 0&0&0&0&0&0&0&0&1\\\hline\end{array}$$

For an example interpretation of H, given that the first value of D is 5, for the value 5 in D there are 3 corresponding features in S at locations 1, 4 and 5. So in the joint histogram H, in row 5 there are 3 values of 1 in columns 1, 4 and 5, respectively. The same process is iterated for all the pixels in R. At the end of the process a vector is constructed with the number of occurrences of each pixel based on the feature S. Each cell in the matrix Occurrence is the sum of a row in H:

$$\text{Occurrence}=\begin{array}{|c|c|c|c|c|c|c|c|c|}\hline 0&1&1&0&3&1&1&1&1\\\hline\end{array}$$

[0133] Next, the values in the chosen Gaussian weight kernel are multiplied with the values in the matrix Occurrence: G*Occurrence With S representing MFM **167**, the weight kernel for each pixel position in the window is calculated as:

$$W_{pq}=\exp\{-\|MFM(p)-MFM(q)\mu\} \qquad (34)$$

where p is the centre of the window and q is the value of each pixel. This means that the weight assigned for the pixel position q is the exponential function of the distance between p and q.

Let's consider the distance between p and q as:

$$di=\|MFM(p)-MFM(q)\| \qquad (35)$$

Then the weight assigned for the pixel position q is:

$$W_{pq}=\exp\{-di\}=e^{-di} \qquad (36)$$

The weight for all the pixels inside the window is calculated using the same process. For the foregoing example, the Gaussian weight kernel is:

$$G = \begin{array}{|c|c|c|}\hline 0.094 & 0.11 & 0.094 \\ \hline 0.11 & 0.14 & 0.11 \\ \hline 0.094 & 0.11 & 0.094 \\ \hline \end{array}$$

To make the interpretation simple, the kernel, G, is reshaped as follows:

0.094	0.11	0.094	0.11	0.14	0.11	0.094	0.11	0.094

Next the values in G are multiplied with the values in matrix Occurrence to provide w, the product of G and the Occurrence:

0	0.11	0.094	0	0.42	0.11	0.094	0.11	0.094

The sum of the reshaped matrix w is 1.032 referred to as w_s Next, a cumulative summation is calculated of the weights in the matrix G until a value $\geq w_s/2$ is reached. which in this example is 0.516. That is,

$$0.094+0.11+0.094+0.11+0.14=0.5480\geq 0.516.$$

The location index of the 0.14 in above summation in the matrix G corresponds to the location of the median value in matrix D. This means that the value 0.14 is located in the 5th cell of matrix G. So the median of R is located the 5th cell.

[0134] The Dynamic Joint Weighted Median Filter operation of Process Block 5, referred to as DJWMF process **205**, processes data corresponding to both the Similarity Map **166** and the Mutual Feature Map **167** to produce data values for the Filtered Disparity Map **172**. The Filtered Disparity Map **172** has reconstructed features corresponding to depth map structure in the original stereo-pair RGB reference images **161**, **162** (**14**, **14'**), but also includes processing artifacts near edges and corners introduced, in part, during the Cost Aggregation operation of Process Block 2, referred to in the figures as Cost Aggregation Process **202**. As an example of processing artifacts, neighboring pixels with vastly different pixel intensities, or superpixels of relatively large size, may create an undesired exaggerated “staircase” or “blocky” appearance, commonly known as a “blocky artifacts.” Instances of blocky artifact are especially prone to occur along edges and at corners of image objects within the Initial Disparity Map **165** if the segmentation size criterion of the superpixel-wise cost function results in definition of superpixels that cross intensity boundaries or object boundaries within the scene. Generally, when superpixels are defined for cost aggregation, this can introduce visible artifacts at the boundaries between superpixels. Because the cost aggregation process is applied individually in each superpixel, neighboring superpixels aggregate the cost differently. This leads to discontinuities at the superpixel boundaries, referred to as blocky artifact. The series of disparity map images in FIGS. **6**A-**6**C provides a simple example of the extent to which blocky artifact becomes more prevalent as the superpixel size, defined in the Cost Aggregation Process **202**, increases. The figures illustrate this based on N, the number of superpixels in the image. The example given is for an image having typical pixel length and width dimensions, e.g., 891×597 (maximum Middlebury benchmark) or 1243×375 (maximum KITTI benchmark). With N=100 in FIG. **2**A, the disparity map has relatively few, but very large, superpixels. As N increases, the superpixel size decreases and the extent of the blocky artifact decreases notably. This is evidenced for several values of N: 100, 1,000, 5000 and 16,000. Referring generally to FIGS. **103**E-**103**G, the Multi-Dimensional Convolution (MDC) Process Block 6, reference **206**, mitigates the severity of such processing artifacts, including blocky artifacts, by performing a Normalized Convolution **212** followed by a Normalized Interpolated Convolution (NIC) **214**, as described below.[2]

[0135] A more detailed illustration of an embodiment of the MDC Process Block 6, reference **206**, shown in FIG. **103**E, combines smoothing methods and interpolation methods to reconstruct a patch H(p) about a suspect pixel p having an intensity value for which there is a high likelihood of correspondence with processing artifacts. Specifically, the reconstruction of a suspect artifact pixel p is accomplished by modifying the pixel value via smoothing or filtering and/or replacing the pixel value via interpolation; and replacing the suspect invalid pixel data values based on neighboring pixel data values having a low likelihood of correspondence with processing artifacts. Those pixels having data values assessed as having a low likelihood of correspondence with processing artifacts are referred to as valid pixels or in some references certain samples, and the data values are referred to as valid data values.

[0136] The intensity values at and near a suspect pixel p are modified based on intensities of neighboring valid pixels, giving the closest neighboring valid pixel data values the most significant influence in assigning a modified intensity value to the suspect pixel or to other pixels in a patch about the suspect pixel p. Applicable methods include filtering, smoothing, and/or interpolation methods and may include mathematical weighting, tapering, or windowing factors which, for example, vary smoothly as a function of proximity of each neighboring pixel to the suspect pixel.

[0137] Mathematically, this may be expressed as a convolution ƒ(x)*g(x) of functions ƒ(x) and g(x), which convolution function may be interpreted as a modified, blended, or filtered version of ƒ(x) as modified, blended, or filtered by function g(x). An exemplary expression for the convolution ƒ(x)*g(x) of a function ƒ(x) by a smoothing function g(x) of a discrete variable x is given by

$$f(x)*g(x)=\Sigma_{k=-\infty}^{\infty}f(k)g(x-k),(F-1) \tag{37}$$

where g(x) is referred to as a filter or smoothing operator. The convolution is generally a computationally expensive operation. The convolution ƒ(x)*g(x) for vectors ƒ(x) of length m and g(x) of length n requires on the order of n×m operations per output value. Much research has been performed over the last 30 years with the goal of creating algorithms that take advantage of specific mathematical geometries and operations for a gain in computational efficiency.

[0138] In the image processing literature, often no distinction is made between the terms “normalized convolution” and “normalized interpolated convolution.” The term “normalized convolution” as used herein is an operation based on a standard kernel g(x) which filters or smoothes the function ƒ(x) and which scales the resulting values by the magnitude of g(x). Alternately, g(x) may be normalized prior to the convolution operation. In contrast to the normalized convolution, the normalized interpolated convolution (NIC) is used to fill gaps and replace invalid data samples based on valid data values, and also scales the resulting values by an additional convolutional product.

[0139] According to an embodiment of the invention, the Normalized Interpolated Convolution (NIC) **214** is used in a processing method that allows reconstruction of image data when relatively few valid pixels are available due to, for example, the presence of noise or instrumental error. In the NIC **214**, the convolution operation of equation (37) is extended using a component c(x) in order to express the confidence or certainty of each measurement or sample of the signal, where x represents the index of the data. Elements of c(x) corresponding to missing sample values equal 0; and elements of c(x) corresponding to valid data values in $f(x)$ equal 1. Therefore, the certainty associated with signal $f(x)$ is expressed as a map c(x) with the same dimension as $f(x)$. The normalization factor in the interpolated convolution is created by the convolution of this certainty map c(x) with the filter or window g(x), as noted below in the discussion of equation (42). See, Knutson, et al., "Normalized and Differential Convolution Methods for Interpolation and Filtering of Incomplete and Uncertain Data" Computer Vision and Pattern Recognition, 1993. Proceedings CVPR '93, 1993 IEEE Computer Society Conference.

[0140] In an example embodiment of the invention, the Normalized Convolution **212** and Normalized Interpolated Convolution (NIC) **214** of Multi-Dimensional Convolution (MDC) Process Block 6, reference **206**, are applied to remove image processing artifacts from the Filtered Disparity Map **172** by smoothing the Initial Disparity Map D **165** or Interim Disparity Map **170**, filling gaps, and replacing invalid pixel values. The method includes sequential execution of multi-channel data transform operations, iterative convolution operations, iterative interpolated convolution operations, and then an inverse data transform operation. A feature of the iterative operations is the approximation of a complex Gaussian blur operation by repeated operation of a much simpler function that imparts computational efficiency. This reconstruction results in Final Disparity Map **174**.

[0141] In a simplified example embodiment, an interpolation process reconstructs missing data. In this example, a function $f(x)$ is a short, one-dimensional representation of a portion of a sequence of the pixel data of a disparity map in which some data determined to be invalid pixel values are replaced by zero values:

$$f(x)=[x_1;0;0;x_4;x_5;0;x_7;0], \qquad \text{(F-2)(38)}$$

where x_i are known samples of valid pixel data, and the missing, invalid, or corrupted samples have been replaced with zeros. An exemplary normalized simple smoothing function is:

$$g(x)=[⅓;⅓;⅓]. \qquad \text{(F-3)(39)}$$

[0142] Convolving the smoothing filter g(x) with $f(x)$ results in a modified one-dimensional representation of a portion of a sequence of the pixel data, referred to as function $f_m(x)$:

$$(F\text{-}4)$$

$$f_m(x) = g(x) * f(x) = \begin{bmatrix} \frac{x_1}{3}; \frac{x_1}{3}; \frac{x_4}{3}; \\ \frac{x_4+x_5}{3}; \frac{x_4+x_5}{3}; \frac{x_5+x_7}{3}; \frac{x_7}{3}; \frac{x_7+x_1}{3} \end{bmatrix} \qquad (40)$$

[0143] The modified function $f_m(x)$ demonstrates in its sequence several interpolated, smoothed values in positions where the original function $f(x)$ had missing data.

[0144] As introduced previously, in the traditional Normalized Convolution (NC) process that is found in the literature, the convolution operation of equation (37) is extended using a certainty factor c(x) to express the confidence or certainty of each measurement or sample or pixel. The certainty factor c(x) associated with function $f(x)$, for the exemplary portion of a sequence of pixel data of an interim disparity map is expressed as a map identifying locations of valid samples:

$$c(x)=[1;0;0;1;1;0;1;0], \qquad \text{(F-5)(41)}$$

where elements of c(x) are zero for missing sample values and one for each known element in $f(x)$ containing valid pixel data values. The convolution of the certainty map c(x) with the smoothing filter g(x) is then calculated to arrive at the normalizing factor:

$$(F\text{-}6)$$

$$c(x) * g(x) = \left[\frac{1}{3}; \frac{1}{3}; \frac{1}{3}; \frac{2}{3}; \frac{2}{3}; \frac{2}{3}; \frac{1}{3}; \frac{2}{3}\right]. \qquad (42)$$

The function of equation (40) is next divided by the vector function of equation (42) to arrive at a normalized reconstructed function with missing data replaced by interpolated values, i.e. with filled gaps:

$$(F\text{-}7)$$

$$f(x)_0 = \frac{f(x) * g(x)}{c(x) * g(x)} \qquad (43)$$

where a normalized reconstructed function $f(x)_0$ approximates the original signal $f(x)$ but without corrupt or missing values. For the example given,

$$(F\text{-}8)$$

$$f(x)_0 = \begin{bmatrix} \frac{x_1}{1}; \frac{x_1}{1}; \frac{x_4}{1}; \\ \frac{x_4+x_5}{2}; \frac{x_4+x_5}{2}; \frac{x_5+x_7}{2}; \frac{x_7}{1}; \frac{x_7+x_1}{2} \end{bmatrix} \qquad (44)$$

[0145] The interpolated convolution approach reduces to the normalized convolution when the certainty values in c(x) are identically equal. For gray scale images, this is identical to interpolation by convolution using a kernel that retains local DC-component values, as commonly used when resampling images in scaling operations that increase the size of an image.

[0146] The certainty factor c(x) may be generalized by allowing values of c(x) from a continuous distribution between zero and one, as opposed to only binary values; this generalized certainty factor c(x) is used to indicate how important or applicable the signals in $f(x)$ are for the analysis at a given point. This allows "locality" to be defined by letting the importance or applicability to decrease with the radius from the current given point, e.g., the suspect pixel location. In this case the applicability vector c(x) becomes equivalent to a mathematical window function (for example, Gaussian, Blackman, or Tukey windows). See, again, Knutson, et al., 1993. The applicability vector reverts to the standard certainty map for an exemplary one dimensional vector. In the case of an image, when the mathematical window refers to a rectangle or boxcar, the smoothing function g(x) is two-dimensional. For an example embodiment, the input data is in the form of the Interim Disparity Map **174**, which has the same dimensionality as the image data it is derived from, i.e., the same number of pixels in the same aspect ratio in the same geometrical arrangement. For an image, Equation (38) corresponds to known pixels x_i whose values are the pixel intensity in the Filtered Disparity Map **172**, and where pixels that are lacking information or corrupted (i.e. missing data) have values replaced with zeroes.

[0147] The convolution and interpolation computation examples are described with one-dimensional (1D) examples to provide a simplified illustration of the concepts applicable to convolution and interpolation for image processing, it being understood that two-dimensional (2D) convolution and interpolation are performed for general image processing. However, convolution in the space of the principal domain of the disparity map image is a computationally expensive operation. To perform a convolution of a two-dimensional Gaussian applicability vector function of radius r=10 on an image having total number of n pixels requires approximately $0=n\times4r^2$ or ~400 separate computations for every output value.

[0148] However, a feature of embodiments of the invention realizes computational advantages by transforming the image into an appropriate domain where the complexity and number of data processing computations are reduced. For example, in the case of a time-series transformed into the Fourier domain, the time-domain operation of convolution becomes a simple point by point multiplication; n×m operations may be reduced to O(n log m) operations. The transformation operation may be performed using Fourier, Laplace, Hilbert, wavelet transforms, or other mathematical transforms. After processing, the data is transformed back to its original domain.

[0149] In an exemplary embodiment, the DJWMF image, D, the Filtered Disparity Map **172**, is mathematically transformed from a 2-dimensional array of pixel values to another domain Ψ_w, such as the spatial frequency domain, to facilitate removal of blocky artifacts within the disparity map. For a uniform discretization U(Ψ) of the original domain Ψ, the normalized convolution generates the smoothed function value Fi of a sample q∈U(Ψ) as:

$$(F\text{-}9)$$

$$Fi(q)=\left(\frac{1}{J_q}\right)\sum\nolimits_{l\in U(\Psi)}D(l)R\left(t(\hat{q}),\,t(\hat{l})\right), \tag{45}$$

where $J_q=\Sigma_{l\in U(\Psi)}R(t(\hat{q}),t(\hat{l}))$ is a normalized factor for pixel q such that $t(\hat{q})=ct(q)$ and R is an arbitrary kernel, e.g., a box kernel. Here ct(q) implements the domain transformation $\Psi\rightarrow\Psi_w$ as:

$$ct(q)=\int_0^q 1+\Sigma_{k=1}^{c}|D'_k(x)|dx,q\in\Psi \qquad \text{(F-10)(46)}$$

where D'_k is the k-th channel of the disparity map D, i.e., each channel is a map, while D_k refers to the whole disparity map. For example, D_k can be the value of a color channel in some color definition space such as RGB. Defining the isometric transform t as

$$t(\hat{q})=ct(q)=t(q,D_1(q),\ldots,D_c(q)) \qquad \text{(F-11)(47)}$$

for t: $\mathbb{R}^{c+1}\rightarrow\mathbb{R}$ where $\mathbb{R}$ represents an arbitrary space of dimensionality c+1 channels, then equation (46) defines a warping ct: $\Psi\rightarrow\Psi_\omega$, of the signal's 1D spatial domain Ψ to the domain Ψ_ω by the isometric transform t. For an example of this type of transform, see Tomasi and Manduchi, "Bilateral filtering for gray and color images" Sixth International Conference on Computer Vision, 1998, published in Proceedings of the 1998 IEEE International Conference on Computer Vision, Bombay, India, pp. 839-846. doi:10.1109/ICCV.1998.710815

[0150] The arbitrary kernel R in equation (45) may, by way of example, be the box kernel, defined as:

$$R(t(\hat{q}),t(\hat{l}))=\delta\{|(\hat{q})-t(\hat{l})|\leq r\} \qquad \text{(F-12)(48)}$$

where r is the kernel radius and δ is a Boolean function returning 1 or 0. R then acts as a smoothing function or filter.

[0151] For the normalized convolution, the box filter operation in the transformed domain is performed by the box filter kernel operation on a summed area table (SAT). The SAT is calculated by computing the cumulative sum along the specified dimension of the input data. As shown in FIG. **103**F, in an exemplary embodiment the box filter kernel operation is performed twice: initially in the horizontal row direction (or the rows of the image), and then that result is processed in the vertical column direction (or the columns of the image). The two-step procedure is iterated three times, to approximate a Gaussian blur operation performed simultaneously across both dimensions, and providing a gain in computational efficiency over the standard Gaussian blur.

[0152] Generally, interpolated surfaces created in an image disparity map according to the invention are smoother than the corresponding ones generated by a normalized convolution operation. The interpolation process estimates the intermediate values, i.e., between discrete samples of a continuing process or a continuous function. The result, based on n values is generation of n+z values, with the z new values specified at appropriate intervals (e.g., pixel positions) of the continuous function, and is applied in these embodiments to correct processing artifacts. The illustrated mathematical interpolation function is a special type of mathematical approximation function that is formulated to provide values that coincide with the sampled data at the interpolation nodes, or discrete sample points, i.e., positions of valid pixel values. The interpolation function may be regarded as a continuous mathematical model of the sampled data, e.g., a mathematically continuous function derived from the discrete sample values.

[0153] The interpolation procedure may be placed in the form of a convolution (using the formalism of Keys, 1981, Keys, R. (1981) Cubic Convolution Interpolation for Digital Image Processing. IEEE Transactions on Acoustics, Speech, and Signal Processing, 29, 1153-1160 (https://doi.org/10.

1109/TASSP.1981.1163711), for example). The interpolation can then be written as the convolution of the sampled data and an interpolation kernel based on the sampled data. In the implementation described here, the continuous interpolation function itself is defined in the domain Ψ, is then transformed into the domain Ψ_ω, using the same transformation as equation (46) and then the equivalent of the convolution operation performed in that domain.

[0154] To provide increased continuity and smoothness to the filtered disparity map Fi(q) can be filtered using the continuous interpolation function transformed into the Ψ_ω domain. This means that the interpolated convolution can filter the result of the normalized convolution as:

$$CCF(q)=\int_{U\Psi}Fi(x)R(t(\hat{q})dx, \qquad \text{(F-13)(49)}$$

where R is the normalized interpolation kernel defined as:

$$R(t(\hat{q}),x)=\delta\{|t(\hat{q})-x|\leq r\}/2r \qquad \text{(F-14)(50)}$$

for r defining the kernel radius.

[0155] In the interpolated convolution, the box filter operation in the transformed domain is again computed by the box filter kernel operating on a summed area table (SAT). However, in this case the SAT is built using the area under the graph, in the transformed domain, of the transformed signal. Again the identical process performed for normalized convolution is employed for the normalized interpolation convolution to implement the box filter, with the box filter is executed twice: an initial pass is performed in the horizontal index direction, and then that result is processed in the vertical index direction. The box filter procedure is iterated three times, thereby approximating a Gaussian blur operation performed simultaneously across both dimensions, but with a gain in computational efficiency.

[0156] In the module 6A the normalized kernels are applied in the convolution operation to the input Filtered Disparity Map **172** generating the Interim Disparity Map **174**.

[0157] An embodiment of a method for removing blocky artifacts according to the Multi-Dimensional Convolution (MDC) Process of Block 6 includes steps to implement the Normalized Convolution module 6A followed by steps to implement the Normalized Interpolated Convolution (NIC) Module 6B. See FIGS. **103**F and **103**G. With the Filtered Disparity Map **172** input to the Normalized Convolution module 6A, in Step S**1**-**1** a multi-channel domain transform is performed to transform the Filtered Disparity Map **172** from an initial Domain to create a Transformed Interim Disparity Map in a Second Domain. See sub-block B**1**-**2**.

[0158] In Step S**1**-**4** values for 2D (horizontal row direction and vertical column direction) Box Kernel Operators (block S**1**-**4**) are generated. In Step S**1**-**5** the Transformed Version of the Interim Disparity Map B**1**-**2** and the Box Kernel Operators B**1**-**4** are used to perform a Normalized Convolution in the Second Domain in the horizontal row direction, using the horizontal row box kernels and the Transformed Version of the Filtered Disparity Map B**1**-**2** (horizontal (row) components). The horizontal Box Kernel Operators are applied in the horizontal direction after accumulating the sum of the overlapped input data into a summed area table.

[0159] In Step S**1**-**6** the Transformed Version of the Interim Disparity Map generated in Step 3 and the Box Kernel Operators from process block S**1**-**4** are used to perform a Normalized Convolution in the Second Domain in the vertical column direction using the vertical column box kernels and the Transformed Version of the Filtered Disparity Map of Block B**1**-**2** vertical column component. The vertical Box Kernel Operators are applied in the vertical direction after accumulating the sum of the overlapped input data into a summed area table.

[0160] According to Step S**1**-**7**, Step 3 and Step 4 are repeated two additional times for a total of three iterations of the two step Normalized Convolution procedure. In Step S**1**-**8**, after the three iterations are completed, the resulting modified Transformed Version of the Interim Disparity Map is transformed back into the initial domain, resulting in the First Interim Disparity Map **174**. This approximates a Gaussian blur operation applied to the Filtered Disparity Map **172**, but the approximation requires fewer computational steps and results in reduced processing time compared to that of a Gaussian blur operation.

[0161] In the Normalized Interpolated Convolution Module 6B of MDC Process Block 6, the normalized interpolation kernels are applied in the interpolated convolution operation to the input First Interim Disparity Map **174** to generate the Final Disparity Map **176**. The First Interim Disparity Map **174** generated in Module 6A is provided as an input to the Normalized Interpolated Convolution module 6B. In Step S**2**-**1** a multi-channel domain transform is performed on the transformed First Interim Disparity Map **174** to generate a Transformed Version of the First Interim Disparity Map **174** in the Second Domain. See sub-block B**2**-**2**.

[0162] In Step S**2**-**3** the First Interim Disparity Map **174** is used to create the values for the 2D Certainty Map. See sub block B**2**-**4**. In Step S**2**-**5** the Interim Disparity Map **174** is input to create the values for the 2D (horizontal row direction and vertical column direction) Interpolated Box Kernel Operators. See sub-block B**2**-**6**. In Step S**2**-**7**, using (i) the horizontal component of the Transformed Version of the Interim Disparity Map **174** in sub-block B**2**-**2**, (ii) the horizontal row Interpolated Box Kernel Operators of sub-block B**2**-**6**, and (iii) the Certainty Map of sub-block B**2**-**4**: a Normalized Interpolated Convolution is performed in the second domain in the horizontal row direction. The horizontal Interpolated Box Kernel Operators of sub-block B**2**-**6** are applied in the horizontal direction after accumulating the sum of the overlapped input Transformed First Interim Disparity Map S**2**-**2** and Certainty Map S**2**-**4** into respective summed area tables.

[0163] In Step S**2**-**8**, using (i) the vertical column component of the Transformed Version of the First Interim Disparity Map **174**, (ii) the vertical column Interpolation Box Kernel Operators of sub-block B**2**-**6**, and (iii) the Certainty Map of sub-block B**2**-**4**: a Normalized Convolution is performed in the second domain in the vertical column direction. The vertical Interpolated Box Kernel Operators of sub-block B**2**-**6** are applied in the vertical direction after accumulating the sum of the overlapped input Transformed First Interim Disparity Map S**2**-**2** and Certainty Map S**2**-**4** into respective summed area tables.

[0164] According to Step S**2**-**9**, the Steps S**2**-**7** and S**2**-**8** are repeated two additional times for a total of three iterations of the two step Normalized Interpolated Convolution procedure. In Step S**2**-**10**, after the three iterations are completed, the resulting version of the Interim Disparity Map in the Second Domain is transformed back to the Initial Domain, producing the Final Disparity Map **176**. This approximates a Gaussian blur operation applied to the

Interim Disparity Map **174** where invalid pixels have been replaced with interpolated pixel values. This approximates a Gaussian blur operation applied to the Interim Disparity Map **174**, but the approximation requires fewer computational steps and results in reduced processing time compared to that of a Gaussian blur operation.

[0165] The first interpolated kernel is applied along the horizontal direction and the second interpolated kernel is applied along the vertical direction. The output values represent the result of the interpolation kernel affected by the result of the normalized kernel. Applying the disclosed process on data of the Filtered Disparity Map **172**, e.g., transforming the disparity map into an appropriate domain and performing both the normalized convolution and the normalized interpolation convolution, generates a smooth disparity map essentially free of blocky artifacts. A prior art calculation using two camera parameters: (1) a distance between a pair of stereo cameras and (2) a stereo camera focal length, is used to calculate the Final Depth Map **180** from the Final Disparity Map **176**.

[0166] Comparing the Final Disparity Map **176** created in Multi-Dimensional Convolution Process Block 6 to the predecessor Initial Disparity Map **165** created in Block 3, the Final Disparity Map **174** advantageously exhibits restored edges and corners, and reconstructed parts, whereas the former is characterized by missing structure. Blocky artifacts are not present in the Final Disparity Map **180**. In other words, with the described methods for operating on the Filtered Disparity Map **172** greatly improved quality results in the final disparity map. This improvement in the disparity map directly affects calculation of specific depths of objects of primary interest in the images, reducing the number of false positives in detection of specific depths.

[0167] The depth estimation process is useful in a wide number of system applications, including three dimensional image reconstruction, three dimensional imaging, object segmentation, autonomous navigation (e.g., drones and aircraft generally, road vehicles, and, generally, situations in which it is desirable to estimate distance to an object. A system running the depth estimation process may be embedded with other image based systems such as used for object detection and classification. See U.S. patent application Ser. No. 15/591,321, “Multi-Camera Vision System and Method Of Monitoring”, filed May 10, 2017 and U.S. patent application Ser. No. 15/654,465, filed Jul. 19, 2010 “Portable System Providing Augmented Vision of Surroundings”, each of which is assigned to the assignee of the present application and hereby incorporated by reference. Other applications include augmented reality and integration with consumer electronics products equipped with cameras, like smart phones, to bring artistic effects such as Bokeh.

[0168] The depth estimation process can be embodied within an embedded signal processing unit, or equivalent, and integrated with a digital camera system to provide real-time depth/disparity maps of an automotive scene. Such maps are useful for a range of automotive applications, including pedestrian detection, vehicle detection and road/lane identification. The disclosed depth estimation methods can be embedded to augment these and other automotive vision systems.

[0169] In one example, the afore described depth estimation process is employed to complement an automotive vision system (AVS) based on advanced object detection technology as described in US filed applications. The object detection AVS can provide the location and estimated range of detected objects to a central processing unit which can compare these data with a corresponding depth map. The information from the depth estimation process can confirm or refine precision in distance determinations for detected objects such as pedestrians and other vehicles. Advantageously alert information may be sent to vehicle collision avoidance systems and emergency braking systems when a trigger event is imminent.

[0170] In other embodiments, where the depth estimation process may not be supported with sufficient resources to operate on full size image frames at real-time rates (e.g. when a wide-field, high resolution 4K or 8K camera system is deployed), the algorithm may be applied selectively to regions of interest (ROI) within the imaged frame. These ROIs can be based on the detected objects and surrounding areas, as provided by the advanced object detection AVS.

[0171] In still other applications the depth estimation process refines performance of other in-vehicle systems. For example, in a vehicle with a 360 degree multi-camera surround vision system the depth estimation process provides detailed range information for persons approaching the vehicle. In the context of a keyless entry system it is desirable to unlock the vehicle when a person with a verified wireless keyfob approaches one of the vehicle doors. However, when a person is beyond 1-2 steps from the vehicle then the locking mechanism should also be engaged. Yet it can be difficult to accurately gauge distances based solely on, for example, the wireless signal strength (e.g., if the battery is weak) but the embedded depth estimation process, when coupled with a 360° multi-camera vision system can solve this problem. See US 20120019613 and US20120019614 which are incorporated herein by reference.

[0172] In another embodiment the depth estimation process is combined with an emergency braking/collision warning (EBCW) system. The information generated by the depth estimation process is communicated to the EBCW system when it detects, for example, that a large region of the central portion of the field of vision is very close to the vehicle. The definition of ‘very close’ will depend on the speed of the vehicle, but at typical urban speeds (e.g., 30 mph) the distance might be on the order of ten to fifteen meters; and, if the region comes closer than, for example, 10 meters, then an active audio alarm is sounded and, if the separation distance continues to diminish, then emergency braking procedures are initiated. As the warning level becomes more elevated, the emphasis of the depth estimation processing can shift to the central region of the image frame to provide faster processing time needed to generate updated information on the separation distance for the object of interest.

[0173] A multi-camera 360° surround vision system for a vehicle can stitch images from adjacent cameras to generate a 360° view around the vehicle. The information from two cameras in the system can also provide a stereo image pair suitable for input to the depth estimation process to determine a more accurate range of distance to the subject than may be possible using object detection alone. Also, use of separate image frames from a panoramic sweep may be used to generate a pseudo stereo image pair for the depth estimation process. See, again, US 20120019613 & 20120019614, hereby incorporated by reference, which disclose techniques applicable to assist in generating stereo image pairs suitable for input to the depth estimation process.

[0174] Referring to FIG. **106**, an exemplary system **220** includes first and second cameras **230**, **232** mounted on a vehicle **236**. The first camera **230**, positioned on the left side of the vehicle, provides a first stream of image data through an image pipeline **234** in which it undergoes local processing, e.g., for object detection, in processing unit **238**. This is followed by further processing, including a detection filter stage **244** and a metadata detection stage **248**, which information is provided, with a series of image frames **252**, to interface logic **254**.

[0175] The second camera **232**, positioned on the right side of the vehicle, provides a second stream of image data through a second image pipeline **240**. A stereo pre-processing unit **260** receives image data from pipelines **234** and **240** where it selectively undergoes treatments such as cropping and filtering, e.g., to reduce the field over which object depth estimation is performed. The preprocessed image data is then processed in a stereo image matching unit **264** which comprises processing stages **201**, **202** and **203** as summarized in FIG. **1** to generate an initial disparity map **165** and an initial depth map **270**. Post processing unit **274** comprises processing stages **204**, **205** and **206** to generate an optimized, final disparity map **174** and an optimized final depth map **180**.

[0176] The main outputs from the single-camera processing and stereo processing pipelines are received by interface logic **254** which ports the differing types of data to a central vehicular processing unit **258** for selective dissemination to vehicle subsystems (e.g., emergency braking system **264**, keyless entry system **266**, pedestrian detection system **268**, vehicular security system **270**) in support of multiple functions controlled by the central vehicular processing unit **258**. Depending on monitored conditions (e.g., determining whether an object distance has become less than a predefined distance from the vehicle **236**), the unit sends various commands or updates to the vehicle subsystems. The unit may also send commands to the processing units in the image pipelines. In one example, the central vehicular processing unit **258** may decide to perform processing on a smaller image region in order to enhance depth accuracy or reduce processing time required to send information to the emergency braking system **264**. The central vehicular processing unit **258** may then send a command to the preprocessing unit **260** to crop a particular region of an image prior to depth map processing so that processing is only performed on a smaller portion of the image.

[0177] Generally, input from the central processing unit **258** can adapt the crop region the main image frame for more selective depth map processing. With a smaller selected region more detailed processing is possible and faster processing (higher frame rate) becomes possible. The system **220** may be in implemented with other processing configurations and other filtering and processing blocks may be incorporated on both single-camera and stereo-camera-pair processing workflows.

The claimed invention is:

1. A method for improving accuracy of depth map information derived from image data descriptive of a scene where pixels of such image data, acquired with one or more image acquisition devices, each have an assigned intensity value, the method comprising:

performing a matching cost optimization by iteratively refining disparities between corresponding pixels in the image data and using optimization results to create a sequence of first disparity values for an initial disparity map for the scene based in part on a superpixel-wise cost function;

performing a guided filter operation on the first disparity values by applying other image data containing structural details that can be transferred to the first disparity values to restore degraded features or replace some of the first disparity values with values more representative of structural features present in the image data descriptive of the scene [which have been degraded due to noise or low spatial resolution,]

the guided filtering operation performed by applying a series of weighted median filter operations to pixel intensity values in the sequence of first disparity values so that each median filter operation replaces a member in the sequence of first disparity values with a median intensity value, where each median intensity value is based on intensity values in a group of pixels within a window of pixels positioned about said member in the sequence,

each window being of a variable size to include a variable number of pixels positioned about said member in the sequence, where selections of multiple ones of the window sizes are based on a measure of similarity between the first disparity values and said other image data, and wherein the series of [weighted] median filter operations provides a new sequence of disparity values for a refined disparity map or from which a depth map of improved accuracy can be created.

2. The method of claim **1** where the other image data applied to perform the guided filtering operation comprises a portion of the pixel image data acquired with the one or more image acquisition devices.

3. The method of claim **1** where the depth map information is derived from stereo image data comprising data from first and second RBG reference images descriptive of the scene and said other image data containing structural details applied to perform the guided filtering operation comprises data from one of the RBG reference images.

4. The method of claim **1** where the series of median filter operations are weighted median filter operations and multiple ones of the window sizes are selected based on a similarity index values.

5. The method of claim **1** where multiple ones of the window sizes are selected based on measures of differences and similarities between data in the sequence of first disparity values for the initial disparity map and data in one the RGB reference images.

6. The method of claim **1** where multiple ones of the window sizes are selected based on one or more measures of similarity between two sets of image data.

7. The method of claim **6** where the two sets of image data comprise the first disparity values and said other image data.

8. The method of claim **1** where:

multiple ones of the window sizes are selected based on one or more measures of similarity between the first disparity values and said other image data; and

the one or more measures of similarity are based on at least one feature taken from the group consisting of luminance, contrast and structure.

9. The method of claim **8** where the measures of similarity are expressable as a map providing pixel by pixel measures of similarity between the two sets of image data.

10. The method of claim **8** where the measures of similarity are calculated in accord with

$$\mathrm{SSIM}(D,I)=[l(D,I)]^{\alpha}\cdot[c(D,I)]^{\beta}\cdot[s(D,I)]^{\gamma},$$

where D and I are, respectively, the sequence of first disparity values for the initial disparity map **165** and data values from the image data descriptive of a scene and acquired with said one or more image acquisition devices,

where l, fvc and s are, respectively, luminance, contrast and structural terms

and

where α, β and γ are exponents for the luminance, contrast, and structural terms, respectively.

11. A method for performing a [median] filter operation on a sequence of data [intensity or magnitude] values which are a subset in a larger plurality of data values derived from acquisitions with one or more devices, the method comprising:

performing a series of filter operations on at least some values in the sequence and replacing each in a plurality of the values in the sequence with a new value resulting from the filter operation,

where each new value is based on values in a different group containing a number of other data values in the sequence and where the number of data values in each group of values is variable and definable in terms of a selectable dimension in a range of window dimensions, and where the window dimensions vary among the groups based on a measure of similarity between values in the sequence and other values in the larger plurality of data values.

12. The method of claim **11** where the larger plurality of data values includes first and second RGB images comprising image data descriptive of a scene acquired with multiple image acquisition devices.

13. The method of claim **11** where the sequence of data values comprises disparity map data values for an initial disparity map for the scene created from the RGB images with a matching cost optimization based in part on a superpixel-wise cost function.

14. The method of claim **11** where the filter operation is a median filter operation.

15. The method of claim **11** where the filter operation is a weighted median filter operation.

16. The method of claim **11** where each new value is a median intensity value based on values in the plurality of data values different from data values in the sequence.

17. The method of claim **11** where each new value is a median intensity value based on RBG image data in the plurality of data values different from data values in the sequence.

18. The method of claim **11** where each group of values corresponds to a different set of pixel image data within a two dimensional window positioned about a pixel position for which a value is replaced by a new value based on the filter operation.

19. A method of defining a series of sizes of adaptive windows, each for use in a filter operation performed on data values for a disparity map, D, corresponding to a scene represented by a reference image I, each filter operation replacing a different data value associated with the disparity map with a new value to improve depth map accuracy for the scene, comprising:

calculating data values for the disparity map associated with the scene;

providing data values for the reference image;

providing a plurality of values, S, collectively corresponding to a similarity map, each value, S, indicative of a level of similarity between a portion of the disparity map data values and a corresponding portion of the reference image data values;

applying the values, S, to determine a series of windows of variable sizes, each window in the series for use in a different filter operation performed on the disparity map data values;

with W designating current window size under evaluation:

defining a smallest window size W_{min} of dimension (2j+1) by (2j+1) where $j=j_{min}$;

defining a largest window size W_{max} of dimension (2j+1) by (2j+1) where $j=j_{max}$;

for each pixel position $P_{i,j}$ in the disparity map, performing a test with the inequality statement

$$\mathrm{SSIM}_{min}<\mathrm{SSIM}_{med}<\mathrm{SSIM}_{max}$$

for each corresponding pixel position $S_{i,j}$ in the similarity map beginning with the smallest window size, where:

SSIM_{min} is the minimum pixel value inside the current window, W

SSIM_{max} is the maximum pixel value inside the current window, W and

SSIM_{med} is the median pixel value determined for the current window, W; and

concluding that:

If the inequality statement is true, then the current window size is the chosen window size for performing a weighted median operation about the pixel position $P_{i,j}$ in the disparity map corresponding to the similarity map pixel position $S_{i,j}$; and

If the inequality statement in step a) is not true, increasing the size of the window by incrementing j to j+1; and

repeating steps a) and b) until either:

(i) SSIM_{med} is between SSIM_{min} and SSIM_{max}, in which case the current window size is the chosen window size for performing a weighted median operation about the pixel position $P_{i,j}$ in the disparity map corresponding to the similarity map pixel position $S_{i,j}$, or

(ii) the maximum window size is reached, in which case the maximum window size is the chosen window size for performing the weighted median operation about the pixel position $P_{i,j}$ in the disparity map corresponding to the similarity map pixel position $S_{i,j}$.

20. A method for improving accuracy of depth map information derived from image data descriptive of a scene, comprising:

performing a matching cost optimization by iteratively refining disparities between corresponding pixels in the image data and using optimization results to create a sequence of initial disparity map data values for an initial disparity map for the scene;

creating a Mutual Feature Map data based on initial disparity map data values and the image data descriptive of the scene;

applying the Mutual Feature Map data to create a series of weighting functions representing structural details that can be transferred to the first disparity values to restore degraded features or replace some of the first disparity values with values more representative of structural features present in the image data descriptive of the scene, including applying a series of weighted median filter operations in which a weight kernel based on the Mutual Feature Map data is applied to assign a weighting factor to each pixel position in a weighted median filter operation, whereby each median filter operation replaces a member in the sequence of first disparity values with a median intensity value, where each median intensity value is based on intensity values in a group of pixels within a window of pixels positioned about a data value in the sequence.

* * * * *

Appendix B: Semi-Parallel Deep Neural Network (SPDNN) Hybrid Architecture, First Application on Depth from Monocular Camera

Semi-Parallel Deep Neural Network (SPDNN) Hybrid Architecture, First Application on Depth from Monocular Camera

Shabab Bazrafkan,[a, ¶] Hossein Javidnia,[a,*, ¶] Joseph Lemley,[a] Peter Corcoran[a]
[a]National University of Ireland Galway, College of Engineering, Department of Electronic Engineering, University Road, Galway, Ireland

Abstract. Deep neural networks are applied to a wide range of problems in recent years. In this work, Convolutional Neural Network (CNN) is applied to the problem of determining the depth from a single camera image (monocular depth). Eight different networks are designed to perform depth estimation, each of them suitable for a feature level. Networks with different pooling sizes determine different feature levels. After designing a set of networks, these models may be combined into a single network topology using graph optimization techniques. This "Semi Parallel Deep Neural Network (SPDNN)" eliminates duplicated common network layers, and can be further optimized by retraining to achieve an improved model compared to the individual topologies. In this study, four SPDNN models are trained and have been evaluated at 2 stages on the KITTI dataset. The ground truth images in the first part of the experiment are provided by the benchmark, and for the second part, the ground truth images are the depth map results from applying a state-of-the-art stereo matching method. The results of this evaluation demonstrate that using post-processing techniques to refine the target of the network increases the accuracy of depth estimation on individual mono images. The second evaluation shows that using segmentation data alongside the original data as the input can improve the depth estimation results to a point where performance is comparable with stereo depth estimation. The computational time is also discussed in this study.

Keywords: Deep Neural Networks, Depth Estimation, Monocular Camera, Machine Learning.

*****Corresponding Author**,** E-mail: h.javidnia1@nuigalway.ie
¶These authors contributed equally to this work.

1 Introduction

Computing pixel depth values provides a basis for understanding the 3D geometrical structure of images. As it has been presented in recent research [1], using stereo images provides an accurate depth due to the advantage of having local correspondences; however, the processing times of these methods is still an open issue.

To solve this problem, it has been suggested to use single images to compute the depth values, but extracting depth from monocular images requires extracting a large number of cues from the global and local information in the image. Using a single camera is more convenient in industrial applications. Stereo cameras require detailed calibration and many industrial use cases already employ single cameras – e.g. security monitoring, automotive & consumer vision systems, and camera infrastructure for traffic and pedestrian management in smart cities. These and other smart-vision applications can greatly benefit from accurate monocular depth analysis. This challenge has been studied for a decade and is still an open research problem.

Recently the idea of using neural networks to solve this problem has attracted attention. In this paper, we tackle this problem by employing a Deep Neural Network (DNN) equipped with semantic pixel-wise segmentation utilizing our recently published disparity post-processing method.

This paper also introduces the use of *Semi Parallel Deep Neural Networks* (SPDNN). A SPDNN is a semi-parallel network topology developed using a graph theory optimization of a set of independently optimized CNNs, each targeted at a specific aspect of the more general classification problem. In [2] [3] the effect of SPDNN approach on increasing convergence and improving model generalization is discussed. For the depth from monocular vision problem a fully-connected topology, optimized for fine features, is combined with a series of max-pooled topologies (2×2, 4×4 and 8×8) each optimised for coarser image features. The optimized SPDNN topology is re-trained on the full training dataset and converges to an improved set of network weights.

It is worth mentioning that this network design strategy is not limited to the 'depth from monocular vision' problem, and further application examples and refinements will be developed in a series of future publications, currently in press.

1.1 Depth Map

Deriving the 3D structure of an object from a set of 2D points is a fundamental problem in computer vision. Most of these conversions from 2D to 3D space are based on the depth values computed for each 2D point. In a depth map, each pixel is defined not by color, but by the distance between an object and the camera. In general, depth computation methods are divided into two categories:

1- Active methods

2- Passive methods

Active methods involve computing the depth in the scene by interacting with the objects and the environment. There are different types of active methods, such as light-based depth estimation, which uses the active light illumination to estimate the distance to different objects [4]. Ultrasound and time-of-flight (ToF) are other examples of active methods. These methods use the known speed of the wave to measure the time an emitted pulse takes to arrive at an image sensor [5].

Passive methods utilize the optical features of captured images. These methods involve extracting the depth information by computational image processing. In the category of passive methods, there are two primary approaches a) Multi-view depth estimation, such as depth from stereo, and b) Monocular depth estimation.

1.2 Stereo Vision Depth

Stereo matching algorithms can be used to compute depth information from multiple images. By using the calibration information of the cameras, the depth images can be generated. This depth information provides useful data to identify and detect objects in the scene [6].

In recent years, many applications, including time-of-flight [7,8], structured light [9], and Kinect were introduced to calculate depth from stereo images. Stereo vision algorithms are generally divided into two categories: Local and Global. Local algorithms were introduced as statistical methods that use the local information around a pixel to determine the depth value of the given pixel. These kinds of methods can be used for real-time applications if they are implemented efficiently. Global algorithms try to optimize an energy function to satisfy the depth estimation problem through various optimization techniques [10].

In terms of computation, global methods are more complex than local methods, and they are usually impractical for real-time applications. Despite these drawbacks, they have the advantage in being more accurate than local methods. This advantage recently attracted considerable attention in the academic literature [11,12].

For example, the global stereo model proposed in [11] works by converting the image into a set of 2D triangles with adjacent vertices. Later, the 2D vertices are converted to a 3D mesh by computing the disparity values. To solve the problem of depth discontinuities, a two-layer Markov Random Field (MRF) is employed. The layers are fused with an energy function allowing the method to handle the depth discontinuities. The method has been evaluated on the new Middlebury 3.0 benchmark [12] and it was ranked the most accurate at the time of the paper's publication based on the average weight on the bad 2.0 index.

Another global stereo matching algorithm, proposed in [13], makes use of the texture and edge information of the image. The problem of large disparity differences in small patches of non-textured regions is addressed by utilizing the color intensity. In addition, the main matching cost function produced by a CNN is augmented using the same color-based cost. The final results are post-processed using a 5×5 median filter and a bilateral filter. This adaptive smoothness filtering technique is the primary reason for the algorithm’s excellent performance and placement in the top of the Middlebury 3.0 benchmark [12].

Many other methods have been proposed for stereo depth, such as PMSC [12], GCSVR [12], INTS [14], MDP [15], ICSG [16], which all aimed to improve the accuracy of the depth estimated from stereo vision, or to introduce a new method to estimate the depth from a stereo pair. However, there is always a trade-off between accuracy and speed for stereo vision algorithms.

Table 1 Comparison of the performance time between the most accurate stereo matching algorithms

Algorithm	Time/MP (s)	W × H (ndisp)	Programming Platform	Hardware
PMSC [12]	453	1500 × 1000 (<= 400)	C++	i7-6700K, 4GHz-GTX TITAN X
MeshStereoExt [11]	121	1500 × 1000 (<= 400)	C, C++	8 Cores-NVIDIA TITAN X
APAP-Stereo [12]	97.2	1500 × 1000 (<= 400)	Matlab+Mex	i7 Core 3.5GHz, 4 Cores
NTDE [13]	114	1500 × 1000 (<= 400)	n/a	i7 Core, 2.2 GHz-Geforce GTX TITAN X
MC-CNN-acrt [17]	112	1500 × 1000 (<= 400)	n/a	NVIDIA GTX TITAN Black
MC-CNN+RBS [18]	140	1500 × 1000 (<= 400)	C++	Intel(R) Xeon(R) CPU E5-1650 0, 3.20GHz, 6 Cores- 32 GB RAM-NVIDIA GTX TITAN X
SNP-RSM [12]	258	1500 × 1000 (<= 400)	Matlab	i5, 4590 CPU, 3.3 GHz
MCCNN_Layout [12]	262	1500 × 1000 (<= 400)	Matlab	i7 Core, 3.5GHz
MC-CNN-fst [17]	1.26	1500 × 1000 (<= 400)	n/a	NVIDIA GTX TITAN X
LPU [12]	3523	1500 × 1000 (<= 400)	Matlab	Core i5, 4 Cores- 2xGTX 970
MDP [15]	58.5	1500 × 1000 (<= 400)	n/a	4 i7 Cores, 3.4 GHz
MeshStereo [11]	54	1500 × 1000 (<= 400)	C++	i7-2600, 3.40GHz, 8 Cores
SOU4P-net [12]	678	1500 × 1000 (<= 400)	n/a	i7 Core, 3.2GHz-GTX 980
INTS [14]	127	1500 × 1000 (<= 400)	C, C++	i7 Core, 3.2 GHz
GCSVR [12]	4731	1500 × 1000 (<= 400)	C++	i7 Core, 2.8GHz-Nvidia GTX 660Ti
JMR [12]	11.1	1500 × 1000 (<= 400)	C++	Core i7, 3.6 GHz-GTX 980
LCU [12]	9572	750 × 500 (<= 200)	Matlab, C++	1 Core Xeon CPU, E5-2690, 3.00 GHz
TMAP [19]	1796	1500 × 1000 (<= 400)	Matlab	i7 Core, 2.7GHz
SPS [12]	49.4	3000 × 2000 (<= 800)	C, C++	1 i7 Core, 2.8GHz
IDR [20]	0.36	1500 × 1000 (<= 400)	CUDA C++	NVIDIA GeForce TITAN Black

Table 1 shows an overview of the average normalized time by the number of pixels (sec/megapixels) of the most accurate stereo matching algorithms as they are ranked by the Middlebury 3.0 benchmark, based on the “bad 2.0” metric. The ranking is on the test dense set. This comparison illustrates that obtaining an accurate depth from a stereo pair requires significant processing power. These results demonstrate that today, these methods are too resource intensive for real-time applications like street sensing or autonomous navigation due to their demand for processing resources.

To decrease the processing power of stereo matching algorithms, researchers recently began to work on depth from monocular images. Such algorithms estimate depth from a single camera while keeping the processing power low.

1.3 Deep Learning

DNN (Deep Neural Networks) are among the most recent approaches in pattern recognition science that are able to handle highly non-linear problems in classification and regression. These models use consecutive non-linear signal processing units in order to mix and re-orient their input data to give the most representative results. The DNN structure learns from the input and then it generalizes what it learns into data samples it has never seen before [21]. The typical deep neural network model is composed of one or more convolutional, pooling, and fully connected layers accompanied by different regularization tasks. Each of these units is as follows:

Convolutional Layer: This layer typically convolves the 3D image I with the 4D kernel W and adds a 3D bias term b to it. The output is given by:

$$P = I * W + b \tag{1}$$

where * operator is nD convolution and P is the output of the convolution. During the training process, the kernel and bias parameters are updated in a way that optimizes the error function of the network output.

Pooling Layer: The pooling layer applies a (usually) non-linear transform (Note that the average pooling is a linear transform, but the more popular max-pooling operation is non-linear) on the input image which reduces the spatial size of the data representation after the operation.

It is common to put a pooling layer after each convolutional layer. Reducing the spatial size leads to less computational load and also prevents over-fitting. The reduced spatial size also provides a certain amount of translation invariance.

Fully Connected Layer: Fully connected layers are the same as classical Neural Network (NN) layers, where all the neurons in a layer are connected to all the neurons in their subsequent layer. The neurons give the summation of their input, multiplied by their weights, passed through their activation functions.

Regularization: Regularization is often used to prevent overfitting of a neural network. One can train a more complex network (more parameters) with regularization and prevent over-fitting. Different kinds of regularization methods have been proposed. The most important ones are weight regularization, drop-out [22], and batch normalization [23]. Each regularization technique is suitable for specific applications, and no single technique works for every task.

1.4 Monocular Vision Depth

Depth estimation from a single image is a fundamental problem in computer vision and has potential applications in robotics, scene understanding, and 3D reconstruction. This problem remains challenging because there are no reliable cues for inferring depth from a single image. For example, temporal information and stereo correspondences are missing from such images.

As the result of the recent research, deep Convolutional Neural Networks (CNN) are setting new records for various vision applications. A deep convolutional neural field model for estimating depths from a single image has been presented in [24] by reformulating the depth estimation into a continuous conditional random field (CRF) learning problem. The CNN employed in this research was composed of 5 convolutional and 4 fully-connected layers. At the first stage of the algorithm, the input image was over-segmented into superpixels. The cropped image patch centered on its centroid was used as input to the CNN. For a pair of neighboring superpixels, a number of similarities were considered and were used as the input to the fully connected layer. The output of these 2 parts was then used as input to the CRF loss layer. As a result, the time required for estimating the depth from a single image using the trained model decreased to 1.1 seconds on a desktop PC equipped with NVIDIA GTX 780 GPU with 6GB memory.

It has been found that the superpixelling technique of [24] is not a good choice to initialize the disparity estimation from mono images because of the lack of the monocular visual cues such as texture variations and gradients, defocus or color/haze in some parts of the image. To solve this issue an MRF learning algorithm has been implemented to capture some of these monocular cues [25]. The captured cues were integrated with a stereo system to obtain better depth estimation than the stereo system alone. This method uses a fusion of stereo + mono depth estimation.

At small distances, the algorithm relies more on stereo vision, which is more accurate than monocular vision. However, at further distances, the performance of stereo degrades; and the algorithm relies more on monocular vision.

The problem of depth estimation from monocular images has been also studied in [26] where a network is designed with two components. First, the global structure of the scene is estimated and later refined using local information. Although this approach enables the early idea of estimating monocular depth using CNNs, the output depth maps do not clearly represent the geometrical structure of the scene.

In another approach [27], an unsupervised convolutional encoder is trained to estimate the depth from monocular images. The depth is estimated considering the small motion between two images (stereo set as input and target). Later, the inverse warp of the target image is generated using the predicted depth and the known displacement between cameras which results in reconstructing the source image. In a similar research [28], an unsupervised CNN is trained by exploiting Epipolar geometry constraints to estimate disparity from single images. The idea is to learn a function that is able to reconstruct one image from the other, by utilizing a calibrated pair of binocular cameras. A left-right disparity consistency loss is also introduced which combines smoothness, reconstruction, and left-right disparity consistency terms and keeps the consistency between the disparities produced relative to both the left and right images.

1.5 Paper Overview

In this paper, a DNN is presented to estimate depth from monocular cameras. The depth map from the stereo sets are estimated using the same approach as [29] and they are used as the target to train the network while using information from a single image (the left image in the stereo set) as input. Four models are trained and evaluated to estimate the depth from single camera images. The network structure for all the models is same. In the first case, the input is simply the original image. In the second case, the first channel is the original image and the second channel is its segmentation map. For each of these two cases, one of two different targets are used; specifically, these targets were the stereo depth maps with or without post-processing explained in [29]. Fig. 1 shows the overview of the general approach used in this paper.

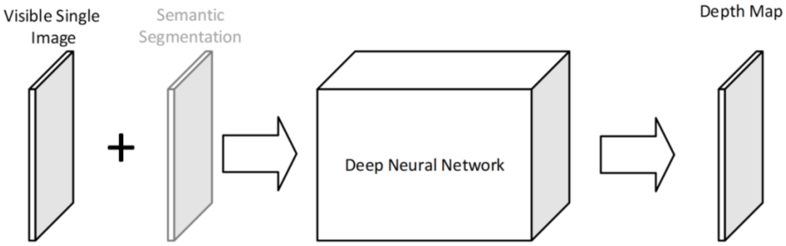


Fig. 1 The overview of the trained models in this paper. The semantic segmentation is just used in two experiments

1.6 Contributions

In this paper two major contributions are presented:

1- A method to mix and merge several deep neural networks called “Semi Parallel Deep Neural Network (SPDNN)”, described in detail in Appendix A.

2- The application of deep neural networks and SPDNN on estimating depth from a monocular camera.

The rest of the paper is organized as follows: In the next section the network structure, database preparation, and the training process are presented. Sec 3 discusses the results and evaluation of the proposed method. The conclusion and discussions are presented in the last section.

2 Methodology

2.1 Network Structure

2.1.1 Semi-Parallel Deep Neural Network (SPDNN)

This paper introduces the SPDNN concept, inspired by graph optimization techniques. In this method, several deep neural networks are parallelized and merged in a novel way that facilitates the advantages of each. The final model is trained for the problem. [2] [3] show that using this method increases the convergence and generalization of the model compared to alternatives.

The merging of multiple networks using SPDNN is described in the context of the current depth mapping problem. In this particular problem, eight different networks were designed for the depth estimation task. These are described in detail in Appendix A. None of these networks on their own gave useful results on the depth analysis problem. However, it was noticed that each network tended to perform well on certain aspects of this task while failing at others. This led to the idea that it would be advantageous to combine multiple individual networks and train them in a parallelized architecture. Our experiments showed that better output could be achieved by merging the networks and then training them concurrently.

2.1.1.1 The Combined Model/Architecture

The process of the network design is discussed in detail in Appendix A. In the final model presented in Fig. 2, the input image is first processed in four, parallel fully convolutional sub-networks with different pooling sizes. This provides the advantages of different networks with different pooling sizes at the same time. The outputs of these four sub-networks are concatenated in two different forms; one to pool the larger images to be the same size as the smallest image in the previous part, and the other one is to un-pool the smaller images of the previous part to be the same size as the largest image.

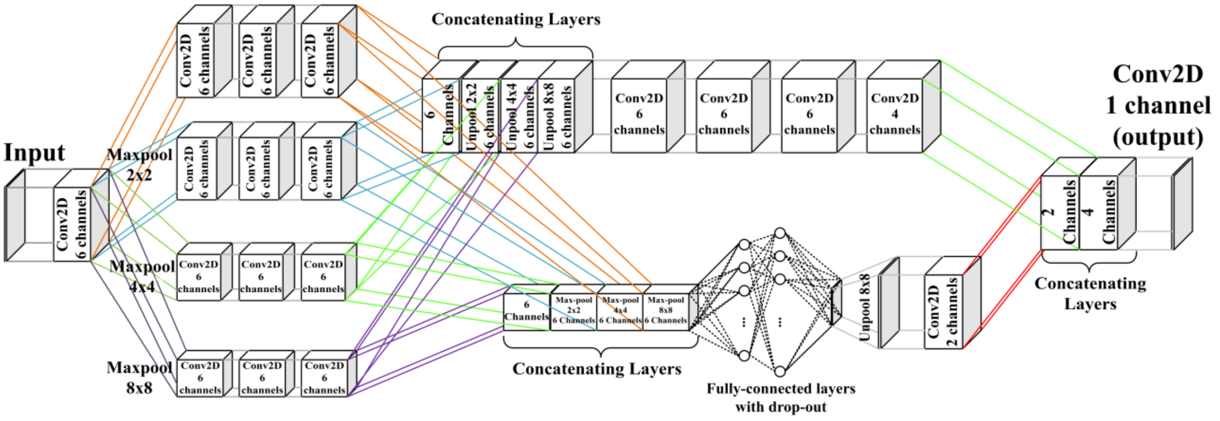


Fig. 2 The model designed for the depth estimation from monocular images.

After merging these outputs, the data is led to 2 different networks. One is the fully convolutional network to deepen the learning and release more abstract features of the input, and the other network is an auto-encoder network with different architecture for encoder and decoder.

It is mentioned in the network design section in Appendix A that, having a fully connected layer in the network is crucial for the reasonable estimation of the image's depth which is provided in the bottleneck of the autoencoder. The results from the autoencoder and the fully convolutional sub-network are again merged in order to give a single output after applying a one channel convolutional layer.

In order to regularize the network, prevent overfitting and increase the convergence, batch normalization [23] is applied after every convolutional layer, and the drop-out technique [22] is used in fully connected layers. The experiments in this paper show that using weight regularization in the fully connected layers gives slower convergence; therefore, this regularization was eliminated from the final design. All the nonlinearities in the network are the ReLU nonlinearity, which is widely used in deep neural networks, except the output layer, which took advantage of the sigmoid nonlinearity. The value repeating technique was used in the un-pooling layer due to non-specificity of the corresponding pooled layer in the decoder part of the auto-encoder sub-network.

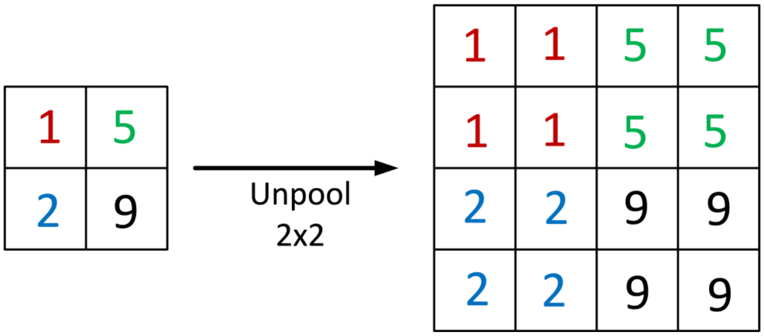


Fig. 3 The repeating technique used in un-pooling layers.

The value repeating technique, illustrated in Fig. 3, involves repeating the value from the previous layer in order to obtain the un-pooled image. The figure shows the 2×2 un-pooling, and the process is the same for other un-pooling sizes.

2.2 Database

In this paper, the KITTI Stereo 2012, 2015 datasets [30] are used for training and evaluation of the network. The database is augmented by vertical and horizontal flipping to expand the total size to 33,096 images. 70% of this dataset is used for training, 20% for validation and 10% for testing. Each model is trained for two sets of input samples and two sets of output targets. The input and target preparation are explained in the following sections.

2.2.1 Data Preparation

2.2.1.1 Input Preparation

Two different sets have been used as the input of the network. The first set includes the visible images given by the left camera. The second set is the visible image + the semantic segmentation of the corresponding input. This gives the opportunity of investigating the segmentation influence on the depth estimation problem. The segmentation map for each image is calculated by employing the well-known model "SegNet" [31,32]. This model is one of the most successful recent implementations of DNN for semantic pixel-wise image segmentation and has surpassed other configurations of Fully Convolutional Networks (FCN) both in accuracy and simplicity of implementation. A short description on SegNet is given in Appendix B.

In our experiments, SegNet was trained using Stochastic Gradient Descent (SGD) with learning rate 0.1 and momentum 0.9. In this paper, the Caffe implementation of SegNet has been employed for training purposes [33]. The gray-scale CamVid road scene database (360×480) [34] has been used in the training step.

2.2.1.2 Target Preparation

The targets for training the network are generated from the stereo information using the Adaptive Random Walk with Restart algorithm [35]. The output of the stereo matching algorithm suffers from several artifacts which are addressed and solved by a post-processing method in [29]. In the present experiments, both depth maps (before post-processing and after post-processing) are used independently as targets. The post-processing procedure is based on the mutual information of the RGB image (used as a reference image) and the initial estimated depth image. This approach has been used to increase the accuracy of the depth estimation in stereo vision by preserving the edges and corners in the depth map and filling in the missing parts. The method was compared with the top 8 depth estimation methods in the Middlebury benchmark [12] at the time the paper was authored. Seven metrics, including Mean Square Error (MSE), Root Mean Square Error (RMSE), Peak Signal-to-Noise Ratio (PSNR), Signal-to-Noise Ratio (SNR), Mean Absolute Error (MAE), Structural Similarity Index (SSIM) and Structural Dissimilarity Index (DSSIM) were used to evaluate the performance of each method. The evaluation ranked the method as 1st in 5 metrics and 2nd and 3rd in other metrics

2.3 Training

As described in Sec 2.2.1.1 and 2.2.1.2 there are two separate sets as input and two separate sets as targets for the training process. This will give four experiments in total as follows:

1- **Experiment 1**: Input: Left Visible Image + Pixel-wise Segmented Image. Target: Post-Processed Depth map

2- **Experiment 2**: Input: Left Visible Image. Target: Post-Processed Depth map.

3- **Experiment 3**: Input: Left Visible Image + Pixel-wise Segmented Image. Target: Depth map.

4- **Experiment 4**: Input: Left Visible Image. Target: Depth map.

The images are resized to 80×264 pixels during the whole process. Training is done on a standard desktop with an NVIDIA GTX 1080 GPU with 8GB memory.

In the presented experiments, the mean square error value between the output of the network and the target values have been used as the loss function, and the Nestrov momentum technique [36] with learning rate 0.01 and momentum 0.9 has been used to train the network. The Training and Validation Loss for each of these experiments are shown in Fig. 4 and Fig. 5 respectively.

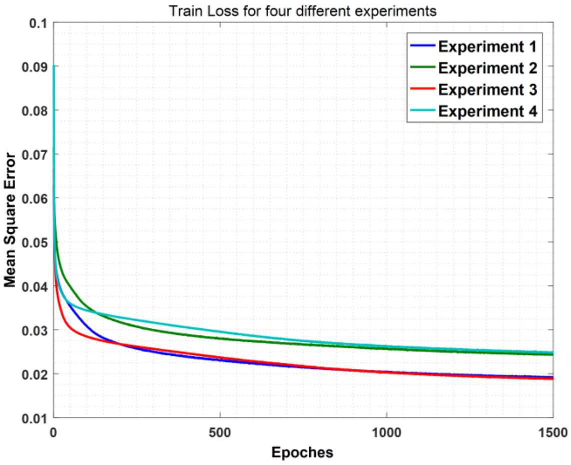


Fig. 4 Train loss for each experiment

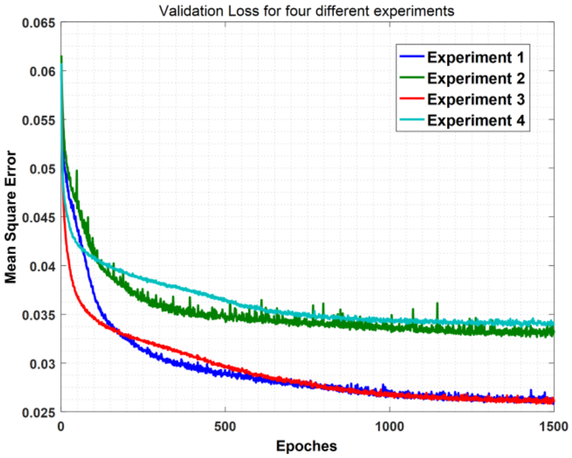


Fig. 5 Validation loss for each experiment

These figures show that using the Post-Processed Depth map as the target results in lower loss values, which means that the network was able to learn better features in those experiments, while semantic segmentation decreases the error only marginally

3 Results and Evaluations

The evaluation in this paper has been done in 4 parts. In the first two parts, the four experiments given in Sec 2.3 are compared to each other, given different ground truths. The third part compares the proposed method to a stereo matching method and the last part shows the comparison against the state of the art monocular depth estimation method. For evaluation purposes, 8 metrics including PSNR, MSE (between 0 and 1), RMSE (between 0 and 1), SNR, MAE (between 0 and 1), Structural Similarity Index (SSIM)(between 0 and 1) [37], Universal Quality Index (UQI) (between 0 and 1) [38] and Pearson Correlation Coefficient (PCC) (between -1 and 1) [39] are used. For the metrics PSNR, SNR, SSIM, UQI, and PCC the larger value indicates better performance, and for MSE, RMSE, and MAE, the lower value indicates better performance. PSNR, MSE, RMSE, MAE, and SNR represent the general similarities between two objects. UQI and SSIM are structural similarity indicators and PCC represents the correlation between two samples. To the best of our knowledge, there have been no other attempts at estimating depth from a mono camera on the KITTI benchmark.

3.1 Comparing Experiments Given Benchmark Ground Truth

The KITTI database came with a depth map ground truth generated by a LIDAR scanner.

Table 2 Numerical comparison of the models given the benchmark's ground truth

	Exp. 1	**Exp. 2**	**Exp. 3**	**Exp. 4**
PSNR	**14.3424**	13.7677	13.8333	13.8179
MSE	**0.0382**	0.0436	0.0435	0.0439
RMSE	**0.1937**	0.2069	0.206	0.2066
SNR	4.4026	3.8279	**6.1952**	6.1798
MAE	**0.1107**	0.1212	0.1236	0.1234
SSIM	**0.9959**	0.9955	0.9955	0.9955
UQI	0.9234	**0.9252**	0.9053	0.9064
PCC	0.7687	**0.8485**	0.7702	0.7729

The test set has been forward propagated through the four different models trained in the four experiments, and the output of the networks has been compared to the benchmark ground truth. The results are shown in Table 2. The best value for each metric is presented in bold.

Figs. 6-8 represent the color-coded depth maps computed by the trained models using the proposed DNN, where the dark red and dark blue parts represent closest and furthest points to the camera respectively. On the top right of each figure, the ground truth given by the benchmark is illustrated. For visualization purposes, all of the images presented in this section are upsampled using Joint Bilateral Upsampling [40]. The results show that using semantic segmentation along with the visible image as input will improve the model marginally. Using the post-processed target in the training stage helps the model to converge to more realistic results.

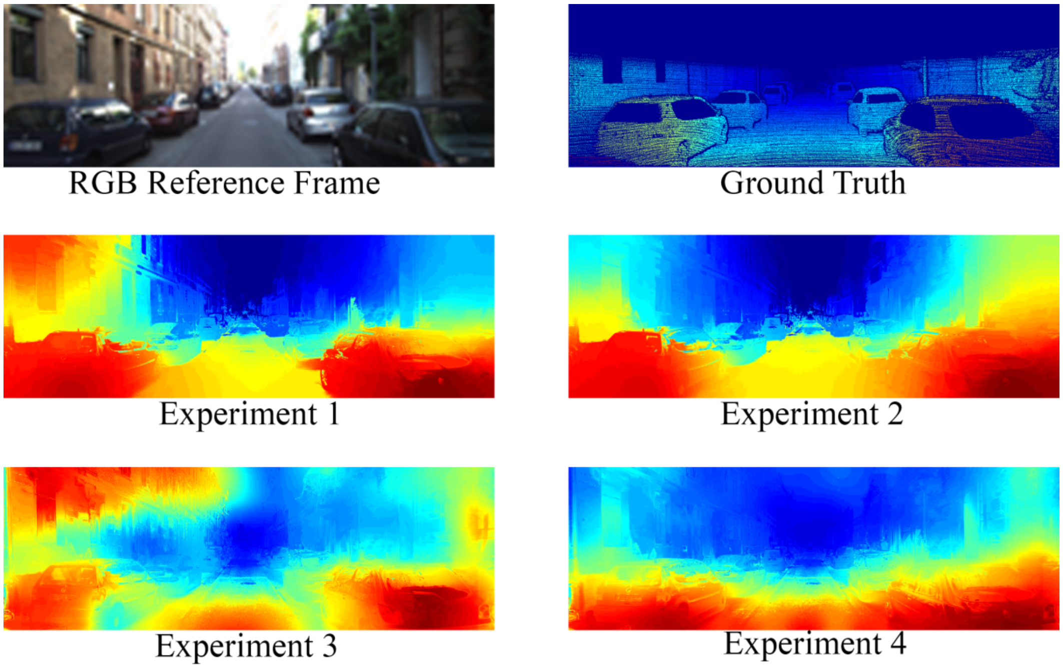


Fig. 6 Estimated depth maps from the trained models – example 1

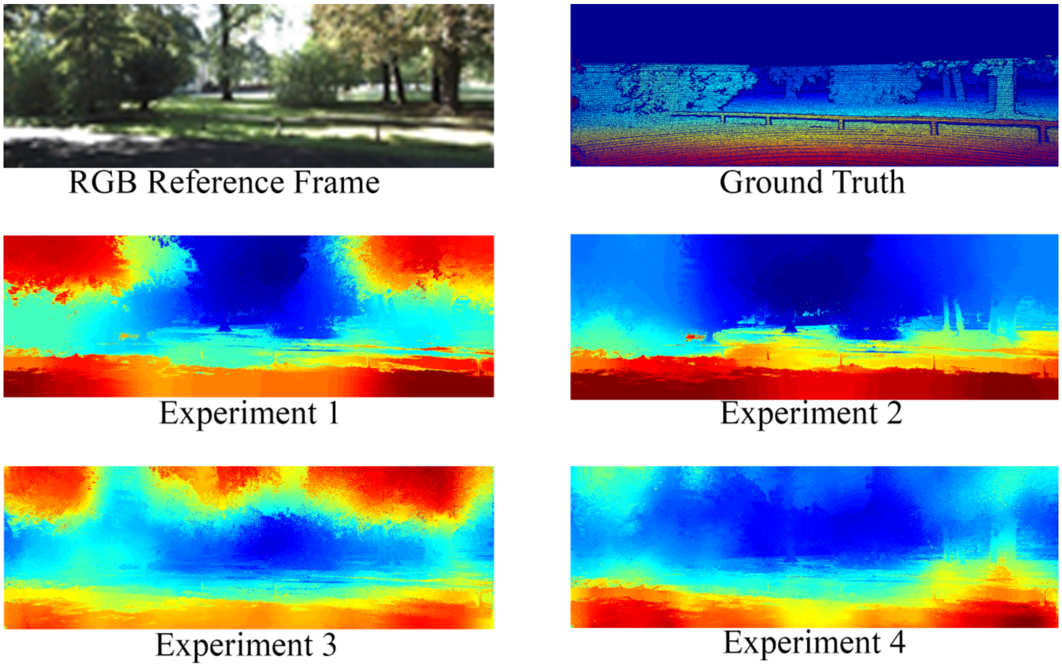


Fig. 7 Estimated depth maps from the trained models – example 2

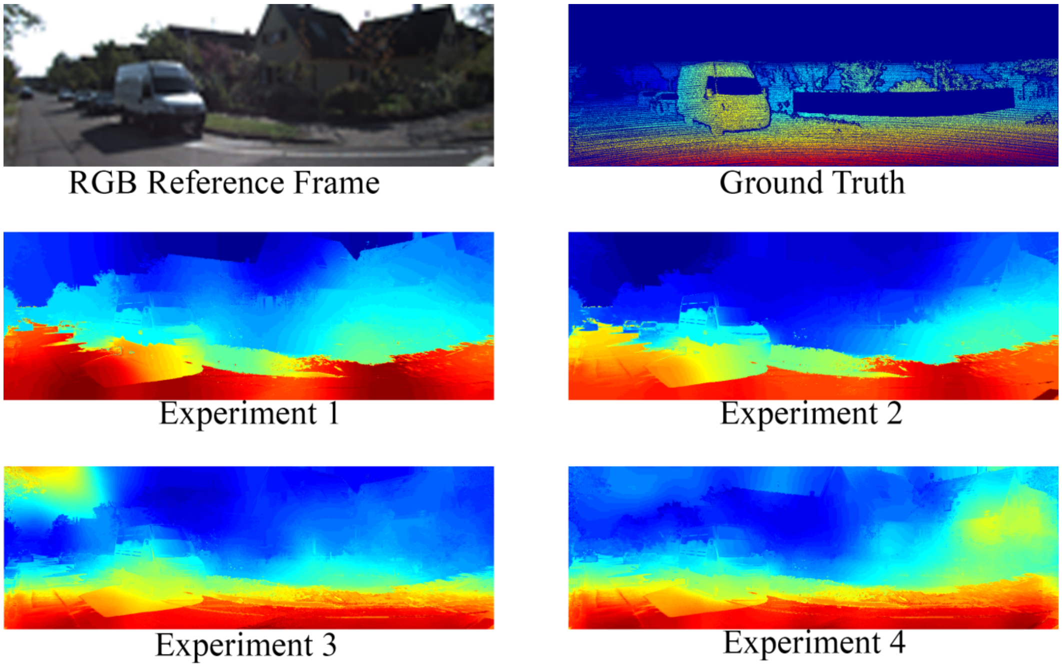


Fig. 8 Estimated depth maps from the trained models – example 3

As it is illustrated in Figs. 6–8, the depth map generated in experiment 1 contains more structural details, and more precise, less faulty depth levels compared with the other experiments.

In general, the presented models in this paper are able to handle occlusions and discontinuities at different depth levels.

3.2 Comparing Experiments Given the Ground Truth from Stereo Matching

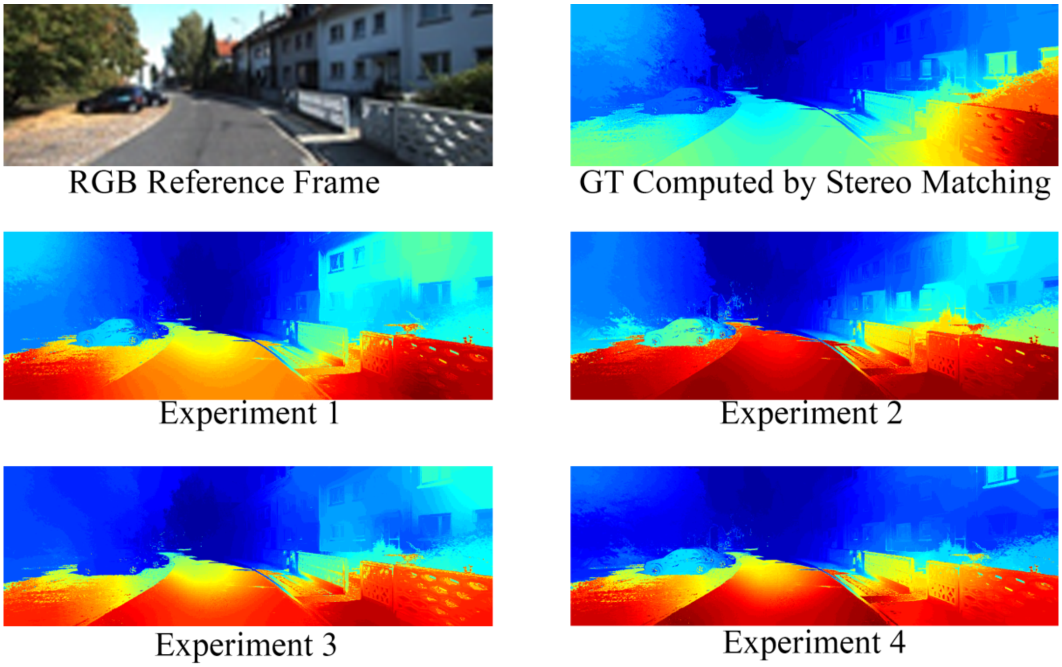
In this section, proposed models are compared to see which one produces closer results to the target value. This gives an idea whether using deep learning techniques on the mono camera can produce reasonable results or not.

Table 3 Numerical comparison of the models given the ground truth from stereo matching

	Exp. 1	**Exp. 2**	**Exp. 3**	**Exp. 4**
PSNR	**15.0418**	14.1895	13.3819	14.0491
MSE	**0.0378**	0.0447	0.0535	0.0441
RMSE	**0.1854**	0.203	0.2223	0.2039
SNR	**8.822**	7.9696	5.4271	6.0943
MAE	**0.1442**	0.1581	0.1673	0.153
SSIM	**0.9952**	0.9943	0.994	0.9951
UQI	**0.8401**	0.8369	0.7951	0.8178
PCC	**0.8082**	0.795	0.704	0.6919

Images in the test set have been forward propagated through the models trained in Sec 2.3, and the outputs are compared with the depth map generated by [29]. The numerical results are shown in Table 3.

The best value for each metric is presented in bold. Figs. 9-11 represent the color-coded depth maps computed by the trained models using the proposed DNN, where the dark red and dark blue parts represent closest and furthest points to the camera respectively. On the top right of each figure, the ground truth calculated by [29] is illustrated. For visualization purposes, all of the images presented in this section are upsampled using Joint Bilateral Upsampling [40]. The results show that using semantic segmentation along with the visible image as input will improve the model marginally. Using the post-processed target in the training stage helps the model to converge to more realistic results.



RGB Reference Frame

GT Computed by Stereo Matching

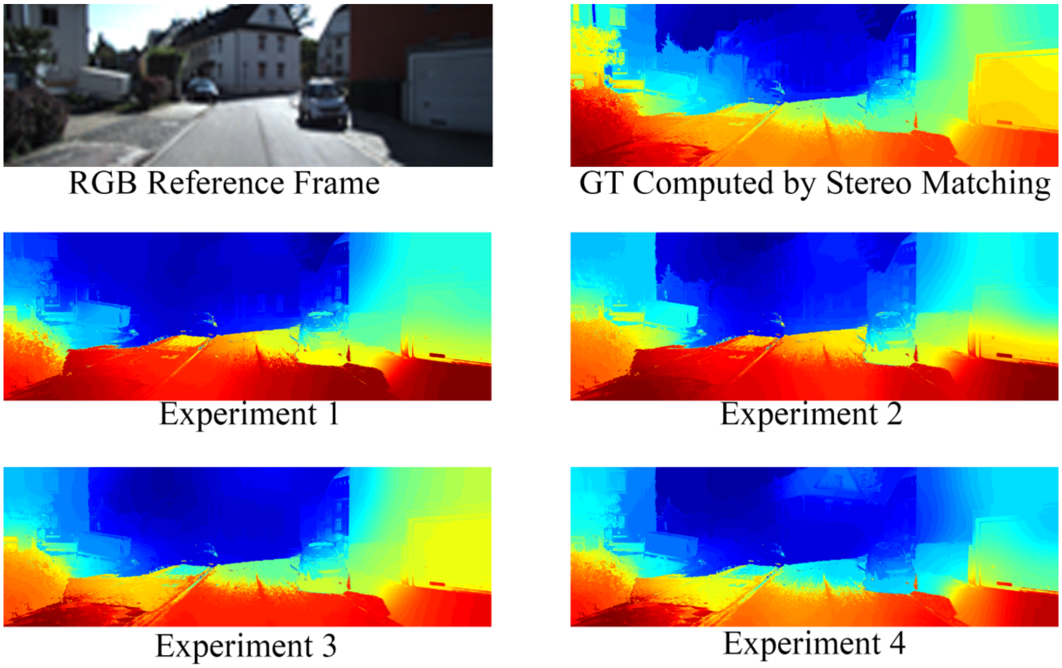
Experiment 1

Experiment 2

Experiment 3

Experiment 4

Fig. 9 Estimated depth maps from the trained models – example 1



RGB Reference Frame

GT Computed by Stereo Matching

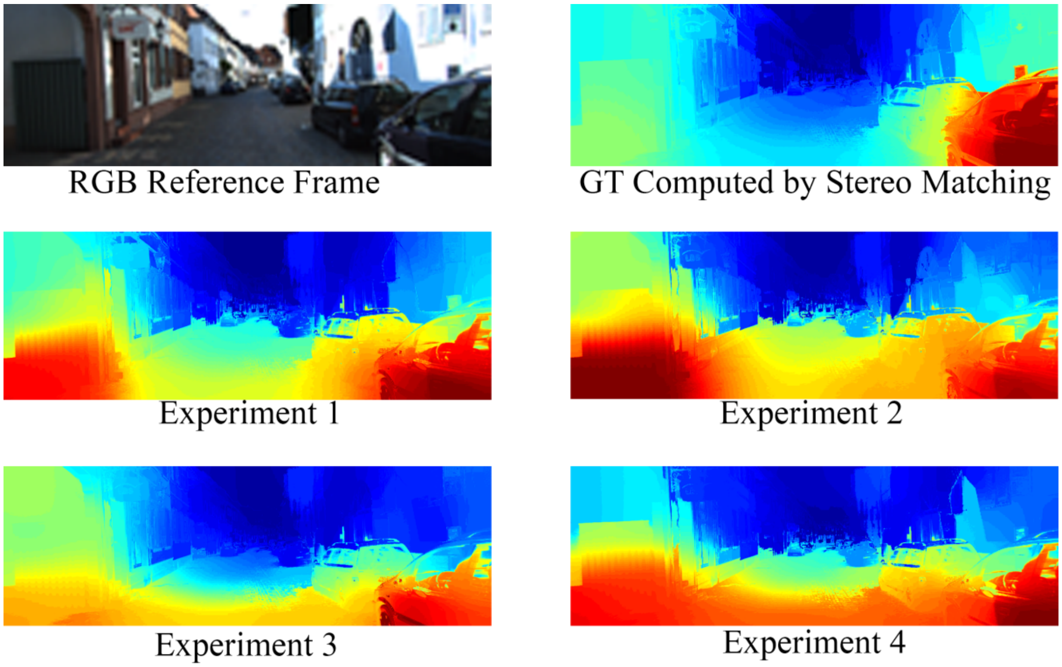
Experiment 1

Experiment 2

Experiment 3

Experiment 4

Fig. 10 Estimated depth maps from the trained models – example 2



RGB Reference Frame

GT Computed by Stereo Matching

Experiment 1

Experiment 2

Experiment 3

Experiment 4

Fig. 11 Estimated depth maps from the trained models – example 3

Figs. 9-11 indicate that the trained models in this paper are able to estimate depth maps comparable to state-of-the-art stereo matching with structural accuracy and precise depth levels. This is also a result of using the semantic segmentation data and injecting the structural information into the network.

3.3 Comparing Mono Camera Results with Stereo Matching

In this section, the results from the mono camera depth estimation given by the proposed method are compared with one of the top-ranked stereo matching methods given in [29]. The ground truth for this comparison is the set of depth maps provided by the KITTI benchmark.

The test images have been forward propagated through the models trained in Sec 2.3 and the best results are compared with the stereo matching technique. The results are shown in Table 4.

The results indicate that using mono camera images and deep learning techniques can provide results which are comparable to stereo matching techniques. As shown in Table 4, the mono camera DNN method was able to provide depth maps similar to the stereo matching methods, represented by PSNR, MSE, MAE, RMSE, and SNR.

Table 4 Numerical comparison between stereo matching and the proposed mono camera model

	Stereo Matching [29]	**Mono Camera DNN**
PSNR	14.8234	14.3424
MSE	0.0351	0.0382
RMSE	0.1845	0.1937
SNR	4.8836	4.4026
MAE	0.1017	0.1107
SSIM	0.9966	0.9959
UQI	0.9353	0.9234
PCC	0.823	0.7687

Having close values for SSIM (0.9966 and 0.9959 in the range [0,1]) and UQI (0.9353 and 0.9234 in the range [0,1]) shows how the mono camera DNN method is able to preserve the structural information, as compared to the Stereo Matching method.

3.4 Comparison against Other Monocular Depth Estimation Methods

In this section, the proposed network is compared again the method presented in [24,26-28]. Table 5 represents the performance of the proposed network compared to the state of the art methods based on seven metrics including Absolute Relative difference, Squared Relative difference, and RMSE/RMSE log. These numbers indicate that the unsupervised CNN proposed by Godard et al. [28] outperforms the others because of the left-right disparity consistency term which allows the network to optimize the disparity values based on both left and right images. However, we believe that the proposed network has a competitive performance compared to the studied methods considering the fact that our models are trained using only left image without taking into account the influence of the right disparity values.

Table 5 Results on the KITTI 2015 stereo 200 training set disparity images.

Method	Supervised	Dataset	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Eigen et al. [26] Coarse	Yes	KITTI	0.361	4.826	8.102	0.377	0.638	0.804	0.894
Eigen et al. [26] Fine	Yes	KITTI	0.203	1.548	6.307	0.282	0.702	0.890	0.958
Liu et al. [24] DCNF-FCSP FT	Yes	KITTI	0.201	1.584	6.471	0.273	0.68	0.898	**0.967**
Garg et al. [27] L12 Aug 8× cap 50m	No	KITTI	0.169	1.080	5.104	0.273	0.740	0.904	0.962
Godard et al. [28]	No	KITTI	**0.148**	1.344	5.927	**0.247**	**0.803**	**0.922**	0.964
Ours	Yes	KITTI	0.288	**1.065**	**4.071**	0.401	0.51	0.77	0.893

Lower is better Higher is better

3.5 *Comparing Running Times*

In this section, the computational time of the proposed method is compared against the stereo matching methods provided in Table 1. The evaluations indicate that the proposed method is able to perform at a rate of ~1.23 sec/MP on a desktop computer equipped with i7 2600 CPU @ 3.4 GHz and 16GB of RAM.

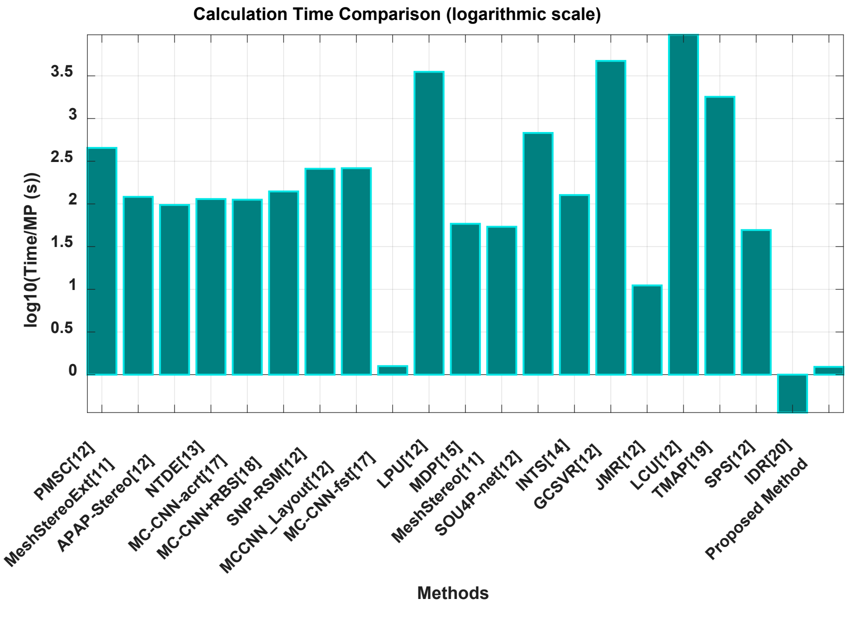


Fig. 12 Comparison of computational time in logarithmic scale

Fig. 12 shows the comparison of the computational times. The comparison is done in a logarithmic scale due to the large range of computational times between different methods.

4 Conclusion and Discussion

In this paper, we have introduced the use of the *Semi Parallel Deep Neural Networks* (SPDNN) method. An SPDNN is a network topology developed using a graph theory optimization of a set of independently optimized CNNs, each targeted at a specific aspect of the more general classification problem. For depth estimation from a monocular set up, a model including fully-connected topology optimized for fine features is combined with a series of max-pooled topologies. The optimized SPDNN topology is re-trained on the full training dataset and converges to an improved set of network weights. Here we used this design strategy to train an accurate model for estimating depth from monocular images.

In this work, 8 different deep neural networks have been mixed and merged using the SPDNN method in order to take advantage of each network's qualities. The mixed network architecture was then trained in four separate scenarios wherein each scenario uses a different set of inputs and targets during training. Four distinct models have been trained. The pixel-wise segmentation and depth estimations given in [29] were used to provide samples for use in the training stage. The KITTI benchmark was used for training and experimental purposes.

Each model was evaluated in two sections, first against the ground truth provided by the benchmark, and secondly against the disparity maps computed by the stereo matching method (Sec 3.1 and 3.2). The results show that using the post-processed depth map presented in [29] for training the network results in more precise models and adding the semantic segmentation of the input frame to the input helps the network preserve the structural information in the output depth map. The results in Sec 3.2 show how close the proposed depth estimation using mono camera can be to the stereo matching method. The semantic segmentation information helps the network converge to the stereo matching results, although the improvement is marginal in this case. The results of the third comparisons in Sec 3.3 show a slightly higher accuracy obtained by employing the stereo matching technique, but our results demonstrate that there is not a big difference between the depths from the models trained by proposed DNN and the values computed by stereo matching. The numerical results of this evaluation show the similarity between the mono camera using DNN method and the stereo matching method, and also the power of the presented method in preserving the structural information in the output depth map.

An important advantage of these models is the processing time of ~1.23 sec/MP. This is equal to 38 fps for an input image of size (80×264) on an i7 2600 CPU @ 3.4 GHz and 16GB of RAM. This makes the model suitable for providing depth estimation in real time. This performance is comparable to the stereo methods MC-CNN-fst [17] and JMR [12], which are 37 fps and 4 fps respectively for the same size of the image, taking advantage of GPU computation power (NVIDIA GTX TITAN X and GTX 980 respectively). The IDR method [20] can give up to 131 fps for the same image size by using an NVIDIA GeForce TITAN Black GPU and CUDA C++ implementation, but the performance on CPU is not given by the authors, so any comparisons with this method would be unfair.

Using pixel-wise segmentation as one of the inputs of the network slightly increased the accuracy of the models, and also helped the model preserve the structural details of the input image. However, it also brought some artifacts, such as wrong depth patches on the surfaces. The evaluation results also illustrate the higher accuracy of the models where a post-processed depth map was used as the target in the training procedure.

4.1 Future Works and Improvements

The model presented in this work is still a big model to implement in low power consumer electronic devices (e.g., handheld devices). Future work will include a smaller design which is able to perform as well as the presented model. The other consideration for the current method is the training data size (which is always the biggest consideration with deep learning approaches). The amount of stereo data available in the databases is usually not big enough to train a deep neural network. The augmentation techniques can help to expand databases, but the amount of extra information they provide is limited. Providing a larger set with accurate depth maps will improve the results significantly.

The SPDNN approach is currently being to other problems and is giving promising results on both classification and regression problems. Those results will be presented in future publications.

Acknowledgments

The research work presented here was funded under the Strategic Partnership Program of Science Foundation Ireland (SFI) and co-funded by SFI and FotoNation Ltd. Project ID: 13/SPP/I2868 on "*Next Generation Imaging for Smartphone and Embedded Platforms*". This work is also supported by an Irish Research Council Employment Based Programme Award. Project ID: EBPPG/2016/280.

References

1. Weber M, Humenberger M, Kubinger W. A very fast census-based stereo matching implementation on a graphics processing unit. Paper presented at: 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops; Sept. 27 2009-Oct. 4 2009, 2009.
2. Bazrafkan S, Corcoran P. Semi-Parallel Deep Neural Networks (SPDNN), Convergence and Generalization. *arXiv:171101963.* 2017;abs/1711.01963.

3. Bazrafkan S, Corcoran PM. Pushing the AI Envelope: Merging Deep Networks to Accelerate Edge Artificial Intelligence in Consumer Electronics Devices and Systems. *IEEE Consumer Electronics Magazine.* 2018;7(2):55-61.

4. Freedman B, Shpunt A, Machline M, Arieli Y. Depth mapping using projected patterns. In: Google Patents; 2013.

5. Govari A, Altmann AC, Ephrath Y, Gliner V. Tissue depth estimation using gated ultrasound and force measurements. In: Google Patents; 2016.

6. Nair D. 3D Imaging with NI LabVIEW. Aug 24, 2016; http://www.ni.com/white-paper/14103/en/.

7. Niclass C, Soga M, Matsubara H, Kato S. A 100m-Range 10-Frame/s 340×96-Pixel Time-of-Flight Depth Sensor in 0.18-µm CMOS. Paper presented at: ESSCIRC (ESSCIRC), 2011 Proceedings of the; 12-16 Sept. 2011, 2011.

8. Niclass C, Ito K, Soga M, et al. Design and characterization of a 256x64-pixel single-photon imager in CMOS for a MEMS-based laser scanning time-of-flight sensor. *Opt Express.* 2012;20(11):11863-11881.

9. Scharstein D, Szeliski R. High-accuracy stereo depth maps using structured light. Paper presented at: Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on; 18-20 June 2003, 2003.

10. Gupta RK, Cho S-Y. A Correlation-Based Approach for Real-Time Stereo Matching. In: Bebis G, Boyle R, Parvin B, et al., eds. *Advances in Visual Computing: 6th International Symposium, ISVC 2010, Las Vegas, NV, USA, November 29 – December 1, 2010, Proceedings, Part II.* Berlin, Heidelberg: Springer Berlin Heidelberg; 2010:129-138.

11. Zhang C, Li Z, Cheng Y, Cai R, Chao H, Rui Y. MeshStereo: A Global Stereo Model with Mesh Alignment Regularization for View Interpolation. Paper presented at: 2015 IEEE International Conference on Computer Vision (ICCV); 7-13 Dec. 2015, 2015.

12. Middlebury Stereo Evaluation - Version 3. http://vision.middlebury.edu/stereo/eval3/.

13. Kim KR, Kim CS. Adaptive smoothness constraints for efficient stereo matching using texture and edge information. Paper presented at: 2016 IEEE International Conference on Image Processing (ICIP); 25-28 Sept. 2016, 2016.

14. Huang X, Zhang Y, Yue Z. Image-Guided Non-Local Dense Matching with Three-Steps Optimization. *ISPRS Ann Photogramm Remote Sens Spatial Inf Sci.* 2016;III-3:67-74.

15. Li A, Chen D, Liu Y, Yuan Z. Coordinating Multiple Disparity Proposals for Stereo Computation. Paper presented at: IEEE Conference on Computer Vision and Pattern Recognition (CVPR); June 2016, 2016; Las Vegas, USA.

16. Shahbazi M, Sohn G, Théau J, Ménard P. Revisiting Intrinsic Curves for Efficient Dense Stereo Matching. *ISPRS Ann Photogramm Remote Sens Spatial Inf Sci.* 2016;III-3:123-130.

17. Žbontar J, LeCun Y. Stereo matching by training a convolutional neural network to compare image patches. *J Mach Learn Res.* 2016;17(1):2287-2318.

18. Barron JT, Poole B. The Fast Bilateral Solver. *arXiv:151103296.* 2016;abs/1511.03296.

19. Psota ET, Kowalczuk J, Mittek M, L. C P, rez. MAP Disparity Estimation Using Hidden Markov Trees. Paper presented at: 2015 IEEE International Conference on Computer Vision (ICCV); 7-13 Dec. 2015, 2015.

20. Kowalczuk J, Psota ET, Perez LC. Real-Time Stereo Matching on CUDA Using an Iterative Refinement Method for Adaptive Support-Weight Correspondences. *IEEE Transactions on Circuits and Systems for Video Technology.* 2013;23(1):94-104.

21. Yegnanarayana B. *Artificial neural networks.* PHI Learning Pvt. Ltd.; 2009.

22. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res.* 2014;15(1):1929-1958.

23. Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv:150203167.* 2015;abs/1502.03167.

24. Liu F, Shen C, Lin G, Reid I. Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 2016;38(10):2024-2039.

25. Saxena A, Schulte J, Ng AY. Depth estimation using monocular and stereo cues. Paper presented at: Proceedings of the 20th international joint conference on Artifical intelligence2007; Hyderabad, India.

26. Eigen D, Puhrsch C, Fergus R. Depth map prediction from a single image using a multi-scale deep network. Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2; 2014; Montreal, Canada.

27. Garg R, B.G. VK, Carneiro G, Reid I. Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue. 2016; Cham.

28. Godard C, Aodha OM, Brostow GJ. Unsupervised Monocular Depth Estimation with Left-Right Consistency. Paper presented at: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 21-26 July 2017, 2017.

29. Javidnia H, Corcoran P. A Depth Map Post-Processing Approach Based on Adaptive Random Walk With Restart. *IEEE Access.* 2016;4:5509-5519.

30. Menze M, Geiger A. Object scene flow for autonomous vehicles. Paper presented at: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 7-12 June 2015, 2015.

31. Badrinarayanan V, Kendall A, Cipolla R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *arXiv:151100561.* 2015;abs/1511.00561.

32. Kendall A, Badrinarayanan V, Cipolla R. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:151102680.* 2015.

33. Kendall A, Badrinarayanan V, Cipolla R. Caffe Implementation of SegNet. https://github.com/alexgkendall/caffe-segnet.

34. Brostow GJ, Shotton J, Fauqueur J, Cipolla R. Segmentation and Recognition Using Structure from Motion Point Clouds. In: Forsyth D, Torr P, Zisserman A, eds. *Computer Vision – ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part I.* Berlin, Heidelberg: Springer Berlin Heidelberg; 2008:44-57.

35. Lee S, Lee JH, Lim J, Suh IH. Robust stereo matching using adaptive random walk with restart algorithm. *Image and Vision Computing.* 2015;37:1-11.

36. Sutskever I, Martens J, Dahl GE, Hinton GE. On the importance of initialization and momentum in deep learning. *ICML (3).* 2013;28:1139-1147.

37. Zhou W, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing.* 2004;13(4):600-612.

38. Zhou W, Bovik AC. A universal image quality index. *IEEE Signal Processing Letters.* 2002;9(3):81-84.

39. Pearson K. Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London.* 1895;58:240-242.

40. Kopf J, Cohen MF, Lischinski D, Uyttendaele M. Joint bilateral upsampling. *ACM Trans Graph.* 2007;26(3):96.

41. Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions. Paper presented at: Proceedings of the IEEE conference on computer vision and pattern recognition2015.

Appendix A: Network Design

A.1: Individual Networks for Depth Analysis

The network shown in Fig. 13 is a deep fully convolutional neural network (A fully convolutional neural network is a network wherein all the layers are convolutional layers) with no pooling and no padding. Therefore, no information loss occurs inside the network, as there is no bottleneck or data compression; this network is able to preserve the details of the input samples. But the main problem is that this model is unable to find big objects and coarse features in the image. In order to solve this problem, three other networks have been designed as shown in Figs. 14-16. These three networks take advantage of the max-pooling layers to gain transition invariance and also to recognize bigger objects and coarser features inside the image. These networks use 2×2, 4×4, and 8×8 max-pooling operators, respectively. Larger pooling kernels allow coarser features to be detected by the network. The main problem with these networks was that the spatial details vanished as a result of data compression in pooling layers.

After several attempts of designing different networks, the observations showed that in order to estimate the depth from an image, the network needed to see the whole image as one object. To do that it requires the kernel to be the same size as the image in at least one layer that is equivalent to a fully connected layer inside the network.

In fully connected layers each neuron is connected to all neurons in the previous/next layer. Due to the computationally prohibitive nature of training fully connected layers, and their tendency to cause overfitting, it is desirable to reduce the number of these connections. Adding fully connected layers results in a very tight bottleneck, which seems to be crucial for the depth estimation task, but also causes the majority of the details in the image to be lost. In Figs. 17-20 the networks with fully connected layers are shown. These networks correspond to networks in

Figs. 13-16 but with convolutional layers replaced with fully connected layers on the right-hand side of the network. Using different pooling sizes before the fully connected layer will cause the network to extract different levels of features, but all these configurations introduce loss of detail.

Each of these eight configurations has its own advantages and shortcomings, from missing the coarse features to missing the details. None of these designs converged to a reasonable depth estimation model.

The main idea of the SPDNN method is to mix and merge these networks and generate a single model which includes all the layers of the original models in order to be able to preserve the details and also detect the bigger objects in the scene for the depth estimation task.

A.2: The SPDNN Parallelization Methodology

A.2.1: Graph Contraction

A consideration while parallelizing neural networks is that having the same structure of layers with the same distance from the input, might lead all the layers to converge to similar values. For example, the first layer in all of the networks shown in Figs. 13-20 is a 2D convolutional layer with a 3×3 kernel.

The SPDNN idea uses graph contraction to merge several neural networks. The first step is to turn each network into a graph in which it is necessary to consider each layer of the network as a node in the graph. Each graph starts with the input node and ends with output node. The nodes in the graph are connected based on the connections in the corresponding layer of the network. Note that the pooling and un-pooling layers are not represented as nodes in the graph, but their properties will stay with the graph labels, which will be explained later.

Figs. 13-20 presents the networks and their corresponding compressed graphs. Two properties are assigned to each node in the graph. The first property is the layer structure, and the second one is the distance of the current node to the input node. To convert the network into a graph, a labeling scheme is required. The proposed labelling scheme uses different signs for different layer structures, C for convolutional layer (for example 3C mean a convolutional layer with 3×3 kernel), F for fully connected layer (for example 30F means a fully connected layer with 30 neurons) and P for pooling property (for example 4P means that the data has been pooled by the factor of 4 in this layer).

Some properties, like convolutional and fully connected layers, occur in a specific node, but pooling and un-pooling operations will stick with the data to the next layers. The pooling property stays with the data except when an un-pooling or a fully connected layer is reached. For example, a node with the label (3C8P, 4) corresponds to a convolutional layer with a 3×3 kernel, the 8P portion of this label indicates that the data has undergone 8×8 pooling and the 4 at the end indicates that this label is at a distance of 4 from the input layer. The corresponding graphs, with assigned labels for each network, are illustrated in Figs. 13-20.

The next step is to put all these graphs in a parallel format sharing a single input and single output node. Fig. 21 shows the graph in this step.

In order to merge layers with the same structure and the same distance from the input node, nodes with the exact same properties are labeled with the same letters. For example, all the nodes with properties (3C, 1) are labeled with letter A, and all the nodes with the properties (3C2P, 4) are labeled K, and so on.

The next step is to apply graph contraction on the parallelized graph. In the graph contraction procedure, the nodes with the same label are merged to a single node while saving their connections to the previous/next nodes. For instance, all the nodes with label A are merged into one node, but its connection to the input node and also nodes B, C, D, and E are preserved. The contracted version of the graph in Fig. 21 is shown in Fig. 22.

Afterwards, the graph has to be converted back to the neural network structure. In order to do this process, the preserved structural properties of each node are used. For example node C is a 3×3 convolutional layer which has experienced a pooling operation. Note that the pooling quality will be recalled from the original network.

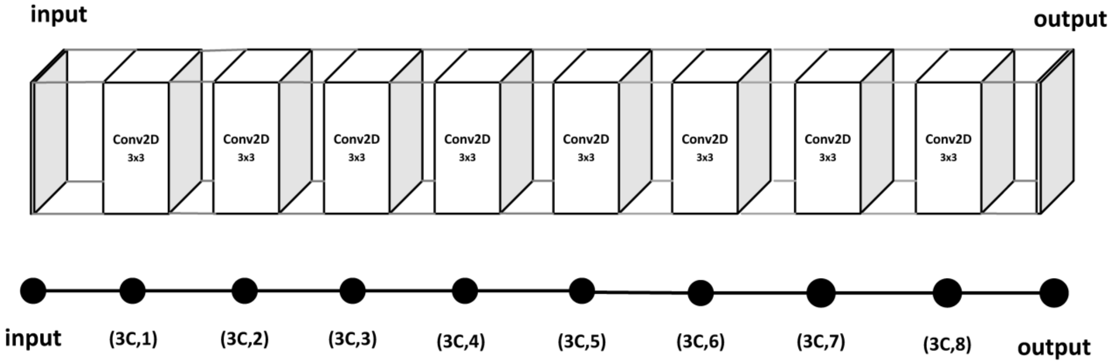


Fig. 13 Top row: network 1, Bottom row: graph corresponds to network1.

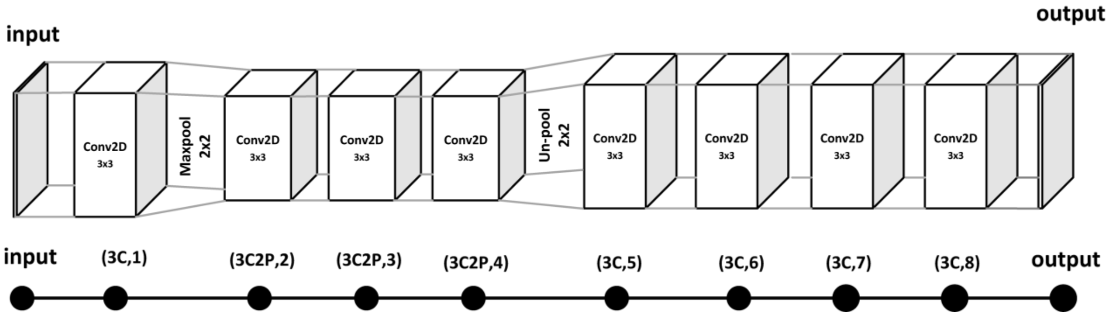


Fig. 14 Top row: network 2, Bottom row: graph corresponds to network2.

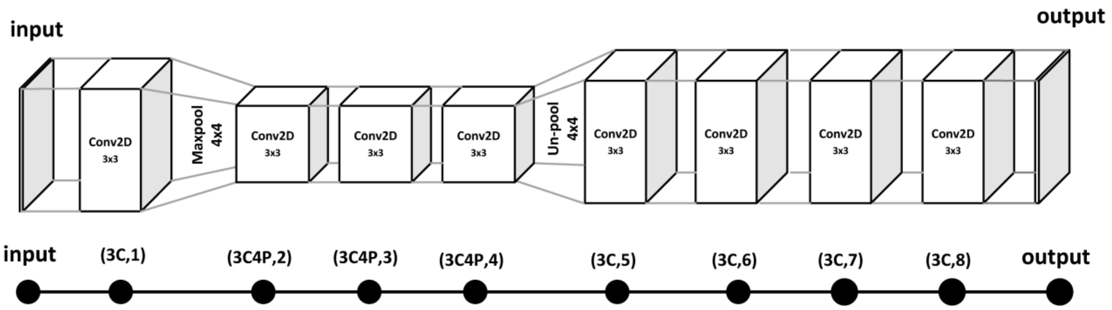


Fig. 15 Top row: network 3, Bottom row: graph corresponds to network3.

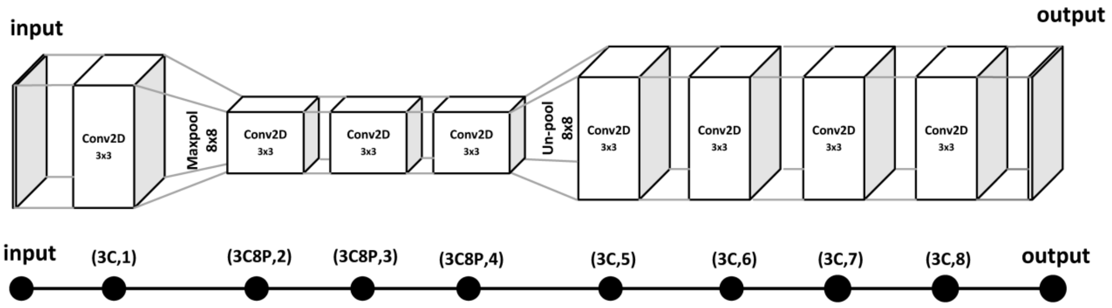


Fig. 16 Top row: network 4, Bottom row: graph corresponds to network4.

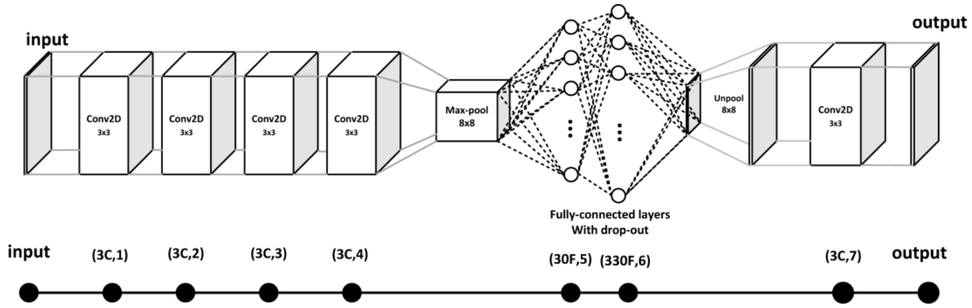


Fig. 17 Top row: network 5, Bottom row: graph corresponds to network5.

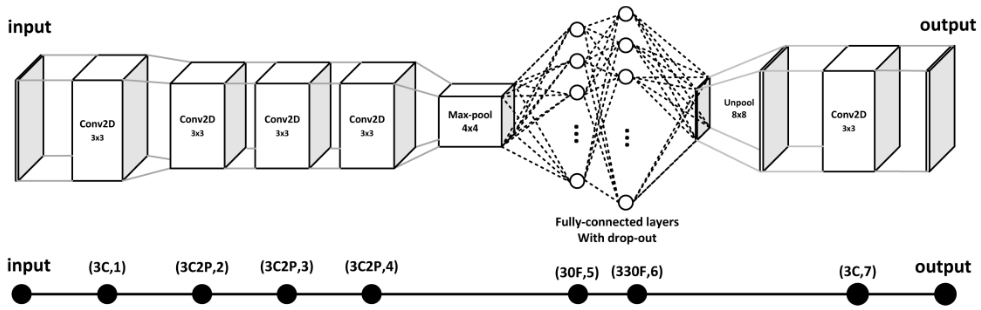


Fig. 18 Top row: network 5, Bottom row: graph corresponds to network6.

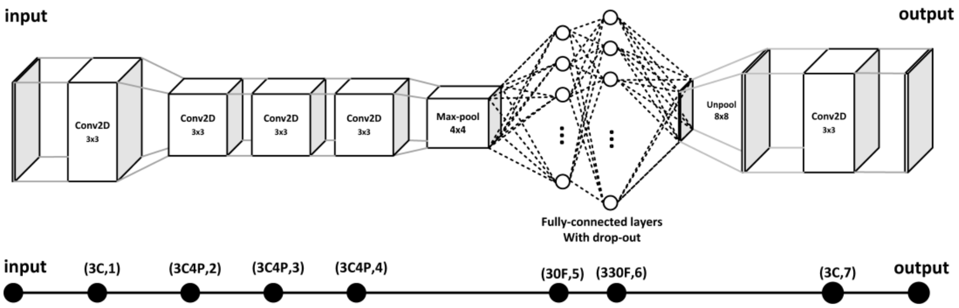


Fig. 19 Top row: network 7, Bottom row: graph corresponds to network7.

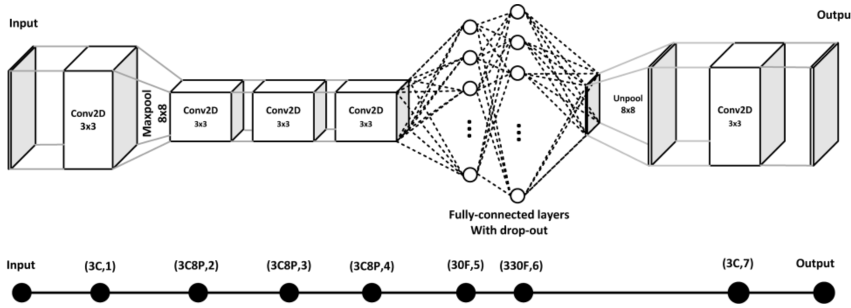


Fig. 20 Top row: network 8, Bottom row: graph corresponds to network8.

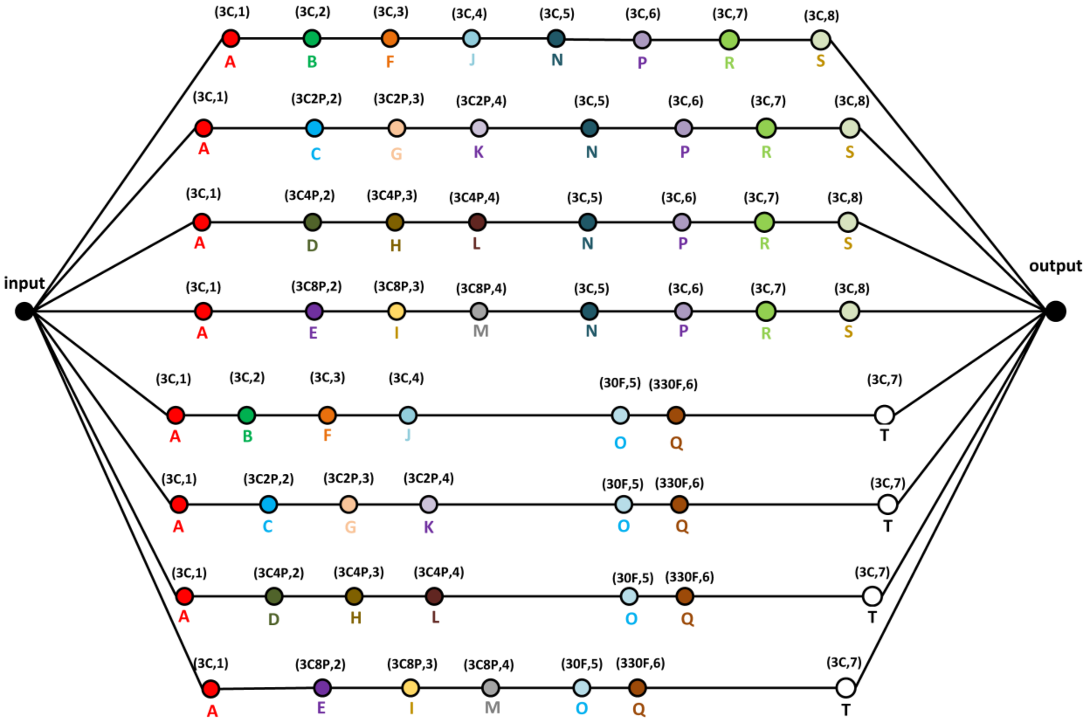

input
output
(3C,1) (3C,2) (3C,3) (3C,4) (3C,5) (3C,6) (3C,7) (3C,8)
A B F J N P R S
(3C,1) (3C2P,2) (3C2P,3) (3C2P,4) (3C,5) (3C,6) (3C,7) (3C,8)
A C G K N P R S
(3C,1) (3C4P,2) (3C4P,3) (3C4P,4) (3C,5) (3C,6) (3C,7) (3C,8)
A D H L N P R S
(3C,1) (3C8P,2) (3C8P,3) (3C8P,4) (3C,5) (3C,6) (3C,7) (3C,8)
A E I M N P R S
(3C,1) (3C,2) (3C,3) (3C,4) (30F,5) (330F,6) (3C,7)
A B F J O Q T
(3C,1) (3C2P,2) (3C2P,3) (3C2P,4) (30F,5) (330F,6) (3C,7)
A C G K O Q T
(3C,1) (3C4P,2) (3C4P,3) (3C4P,4) (30F,5) (330F,6) (3C,7)
A D H L O Q T
(3C,1) (3C8P,2) (3C8P,3) (3C8P,4) (30F,5) (330F,6) (3C,7)
A E I M O Q T

Fig. 21 Parallelized version of the graphs shown in Figs. 13-20 sharing a single input node and single output node

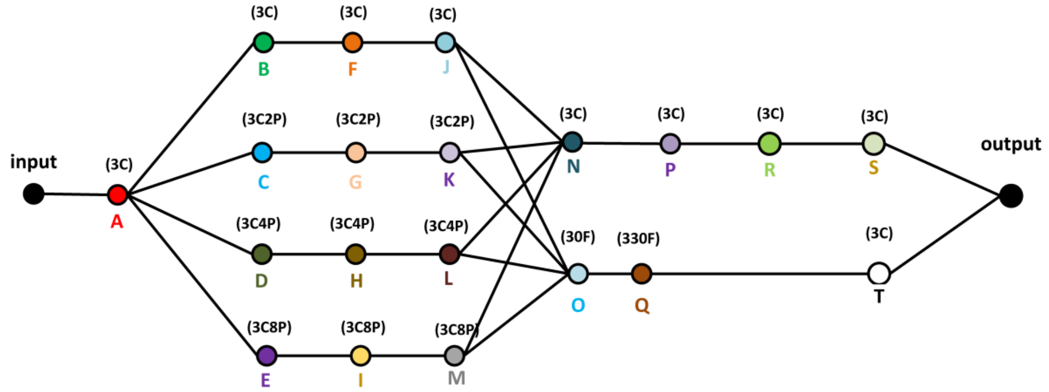


Fig. 22 Contracted version of the big graph shown in Fig. 21

The concatenation layer is used in the neural network in order to implement the nodes wherein several other nodes lead to one node. For example, in nodes N and O the outputs of nodes J, K, L, and M are concatenated with the pooling qualities taken from their original networks.

The graph is translated back to a deep neural network. The network correspond to the graph in Fig. 22 is shown in Fig. 2.

A.3: SPDNN: How it Works and why it is Effective?

One might ask why the SPDNN approach is effective and what the difference is between this approach and other mixing approaches. Here the model designed by the SPDNN scheme is investigated in the forward and back propagation steps. The key component is in the back-propagation step where the parameters in parallel layers influence each other. These two steps are described below:

Forward propagation: Consider the network designed by the SPDNN approach shown in Fig. 23. This exemplary network is made of five sub-networks. Just the general view of the network is shown in this figure and the layers' details are ignored since the main goal is to show the information flow within the whole network.

When the input samples are fed into the network, the data travels through the network along three different paths shown in Fig. 24.

At this stage the parallel networks are blind to each other, i.e., the networks placed in parallel do not share any information with each other. As shown in Fig. 24 the data traveling in Sub-Net 1 and Sub-Net 2 are not influenced by each other since they do not share any path together, as in Sub-Net 3 and Sub-Net 4.

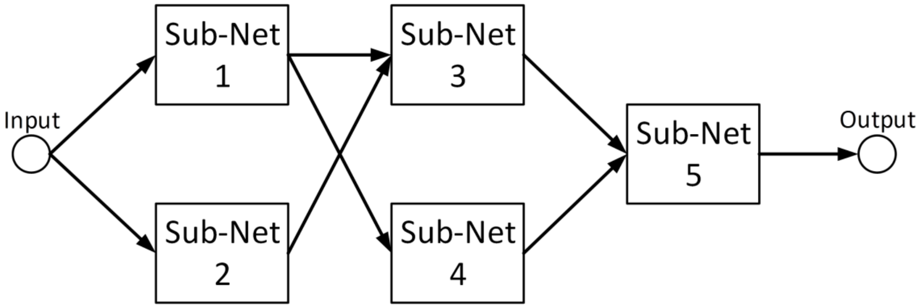


Fig. 23 A network designed using the SPDNN approach. It contains 5 sub-networks placed in parallel and semi-parallel form.

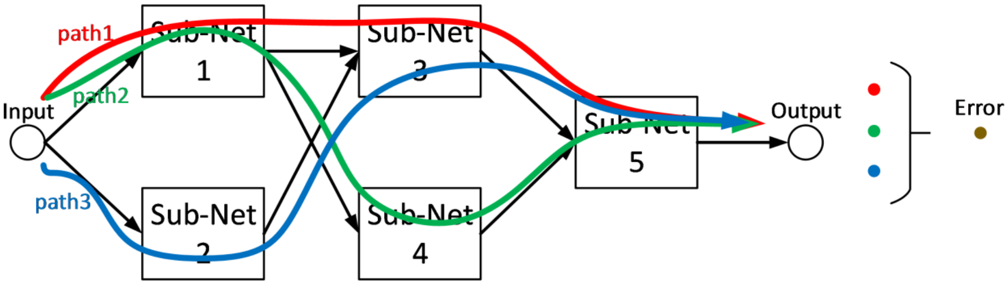


Fig. 24 Forward propagation inside the SPDNN. There are three different paths on which the information can flow inside the network

Backpropagation: while training the network, the loss function calculated based on the error value at the output of the neural network is a mixed and merged function of the error value corresponding to every data path in the network. In the backpropagation step the parameters inside the network update based on this mixed loss values. i.e., this value back-propagates throughout the whole network as it is shown in Fig. 25. Therefore, at this stage of training, each subnetwork is influenced by the error value from every data path shown in Fig. 25. This illustrates the way each subnetwork is trained to reduce the error of its own path and also the error from the mixture of all paths.

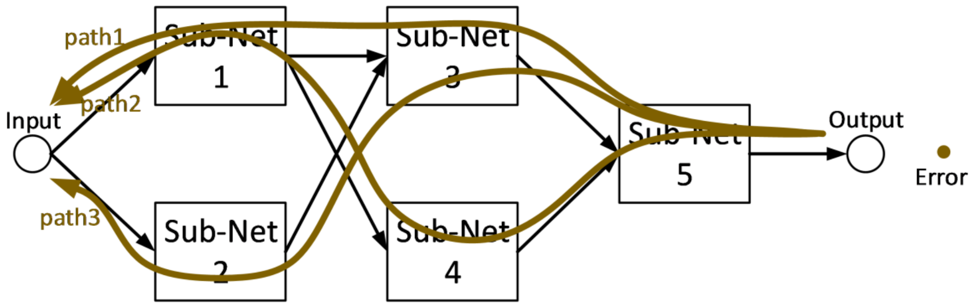


Fig. 25 Backpropagation for SPDNN. The mixed error is back propagated throughout the network while updating parameters.

The main difference between the SPDNN approach and other mixing approaches, like the voting approach, lies in the back propagation step where different sub-nets are influenced by the error of each other and try to compensate for each other's shortcomings by reducing the final mixed error value. In the voting approach, different classifiers are trained independently of each other and they do not communicate to reduce their total error value.

A.3.1: SPDNN vs. Inception

One of the approaches that has superficial similarities to SPDNN is the Inception technique [41]. For clarity, and to aid the reader in understanding, the authors list four significant points of difference between SPDNN and Inception with regard to mixing networks.

1. The main idea in SPDNN is to maintain the overall structure of the networks, but to mix them in a reasonable way. For example, if there is a big kernel such as 13×13 in one of the configurations, the SPDNN method always preserves the structure (13×13 kernel)

inside the final network. This contrasts with inception [41], which reduces larger kernels into smaller ones.

2. In the inception method, all the layers are merged into one final layer, which does not happen with the SPDNN approach.
3. The number of the layers in the SPDNN architecture is less than or equal to the number of the layers in the original networks. In contrast, the inception idea aims to increase the number of layers in the network by (it breaks down each layer into several layers with smaller kernels).

The SPDNN idea is to design a new network from existing networks that perform well at some task or subtask while the idea in inception is to design a network from scratch.

Appendix B: SegNet

SegNet is fully convolutional semantic image segmentation framework presented in [31,32]. This model uses the convolutional layers of the VGG16 network as the encoder of the network and eliminates the fully connected layers, thus reducing the number of trainable parameters from 134M to 14.7M, which represents a reduction of 90% in the number of parameters to be trained. The encoder portion of SegNet consists of 13 convolutional layers with ReLU nonlinearity followed by max-pooling (2×2 window) and stride 2 in order to implement a non-overlapping sliding window. This consecutive max-pooling and striding results in a network configuration that is highly robust to translation in the input image but, has the drawback of losing spatial resolution of the data.

This loss of spatial resolution is not beneficial in segmentation tasks where it is necessary to preserve the boundaries of the input image in the segmented output. To overcome this problem, the following solution is given in [31]. As most of the spatial resolution information is lost in the max-pooling operation, saving the information of the max-pooling indices and using this information in the decoder part of the network preserves the high-frequency information.

Note that for each layer in the encoder portion of the network there is a corresponding decoder layer. The idea of SegNet is that wherever max-pooling is applied to the input data, the index of the feature with the maximum value is preserved. Later these indices will be employed to make a sparse feature space before the de-convolution step, applying the un-pooling step in the decoder part. A batch normalization layer [23] is placed after each convolutional layer to avoid overfitting and to promote faster convergence. Decoder filter banks are not tied to corresponding encoder filters and are trained independently in the SegNet architecture.

Appendix C: Accurate Depth Map Estimation from Small Motions

Accurate Depth Map Estimation from Small Motions

Hossein Javidnia
National University of Ireland, Galway
University Road, Galway, Ireland, H91 TK33
{h.javidnia1}@nuigalway.ie

Peter Corcoran
National University of Ireland, Galway
University Road, Galway, Ireland, H91 TK33
{peter.corcoran}@nuigalway.ie

Abstract

With the growing use of digital lightweight cameras, generating 3D information has become an important challenge in computer vision. Despite several attempts presented in the literature to solve this challenge, it remains an open problem when it comes to the structural accuracy of the depth map and the required baseline (distance between the first and the last frames) to capture a sequence of images. In this paper, a novel approach is proposed to compute a high quality dense depth map together with a semi-dense/dense 3D structure from a sequence of images captured on a narrow baseline. Computing the depth information from small motions has been a challenge for decades because of the uncertain calculation of depth values when using a small baseline – up to 12mm. The proposed method can, in fact, perform on a much wider range of baselines from 8 mm up to 400 mm while respecting the structure of the reference frame. The evaluation has been done on more than 10 sets of recorded small motion clips and for the wider baseline, on 7 sets of stereo images from Middlebury benchmark. Preliminary results indicate that the proposed method has a better performance in terms of structural accuracy in comparison with the current state of the art methods. Also, the performance of the proposed method remains stable even when only a low number of frames are available for processing.

1. Introduction

The use of consumer cameras, specifically smart-phones is growing continuously nowadays and the level of expectation around what these cameras can do is increasing year by year. Consumers and photographers generally prefer to have advance features such as shallow depth of field in their images. This effect requires a large aperture like the ones used in DSLR cameras. Lightweight cameras like those in smartphones are equipped with small apertures which are not capable of reproducing this effect.

Because these types of cameras are equipped with only one lens, this feature is commonly implemented by using a focal stack to compute the depth map [1, 2, 3, 4]. An alternative approach is to compute the 3D structure of the scene and the corresponding depth map.

The 3D structure can be computed using the frame-to-frame movements of the handheld camera. Movements of the camera can occur for several reasons, such as natural hand-shake, or when the user moves the camera slightly to capture a better scene. Generally, this effect is considered as an issue to be solved with image stabilization methods or stabilization gear such as tripods. However these types of movements can be used advantageously in a variety of applications for instance synthetic defocus [5, 6].

The baseline between sequences of frames captured as a sudden motion is considered to be small if it's less than ~8 mm. This restricts the viewing angle of a 3D point to less than 0.2° [7]. Due to this limitation the general Structure from Motion (SfM) methods fails [8, 9, 10] and the computed depth map will be highly penalized.

Several works addressed the challenges of the Structure from Small Motion (SfSM) [5, 7, 11, 12] and proposed a number of algorithms. But there are still a couple of open challenges remaining for these methods such as:

1- These methods fail for baselines wider than ~12 mm. In wide-baseline motions, local image deformations cannot be realistically approximated by translation or translation with rotation and a full affine model is required. Also larger baselines increase the observed disparities, but increase the difficulty of finding corresponding points due to a larger change in viewpoint. This statement is specifically targeting the close scenes with shorter depth ranges.
2- These methods return false results when the number of the input frames is less than 15 frames.
3- The structure of the depth map is not respected properly based on the reference frames. More specifically the depth maps generated by these methods suffer from the lack of accuracy along the edges and corners of structures within the imaged scene.
4- Some of these methods suffer from missing/undefined patches in the depth map, especially along the boundaries of the image.

In this paper, we propose an approach to estimate the depth from small motion clips that addresses each of the challenges mentioned above. In addition to its ability to provide high structural accuracy and occlusion handling, the proposed method has 2 important additional advantages:

1- It is able to process a sequence of image frames with baselines as large as 400 mm.
2- There is no restriction on the minimum number of frames in the proposed method. The evaluation shows that it can perform accurately for $frames \geq 2$.

In the next section, we review the previous works done in this area. Section 3 presents the details of the proposed approach and the evaluation and comparison results are presented in section 4.

2. Related Works

The first step in the process of the SfSM is to build a dense 3D model from the sequence of images. This step is widely studied in several SfM research works [13, 14, 15].

In SfM, bundle adjustment [16] is used to find the optimal estimation of the sparse 3D structure of the scene and positions of camera poses. Nonlinear least square is used as the basic cost function to evaluate the reprojection error from undistorted to distorted image domain. There are several issues that must be solved for this method to be successful:

1. The accuracy of the estimated 3D structure is highly dependent on proper initialization of the cost function. To solve this problem factorization methods have been widely adopted in SfM literature as a means for initializing the bundle adjustment [17, 18, 19].
2. When encountering continues texture-less surface, the method is not capable of producing 3D points due to the lack of features and the failure of the feature tracking.
3. The feature tracking is also an issue in case of rapid movements.
4. Complex computation of the reprojection error for inverse depth representation because of mapping the projected 3D points from the undistorted image domain to the distorted image domain [20]. This issue makes the normal bundle adjustment improper for small motions.

To overcome the problems of the common bundle adjustments with small baseline motions, a modified bundle adjustment is presented in [11]. In this case the reprojection error is calculated from distorted to undistorted image domain [21]. This solves the inverse depth representation problem. The method presented in [11] also employs the idea that in small motion clips the cost function can be initialized better as long as the camera poses or the distance between frames are closer to each other. The idea used in [11] was initially introduced in [7] to find the trajectory of the camera from small motion. The density profile in [7] is created by random depth initialization and plane sweeping based image matching [22, 23]. It employs Markov Random Field [24] to regularize the estimated depth effectively.

The method presented in this paper appears to be the first to deal successfully with wider baselines and low frame-rate motion clips. This work presents evaluation and comparisons with other methods in both small and large baseline motions. The results demonstrate that the method proposed here performs better in terms of accuracy of the depth estimation and respecting the structure of the reference image frame.

Fig. 1 illustrates the general overview of the proposed SfSM approach.

3. Proposed Method

The main steps of the proposed SfSM approach are detailed and explained in this section.

The feature detection of the 3D reconstruction block in the proposed method is equipped with ORB features [25]. The correspondence features location to the initial features is found by Kanade-Lukas-Tomashi (KLT) method [26].

The bundle adjustment presented in [11] is used for 3D structure optimization based on the Huber loss function [27]. The reason for employing this bundle adjustment is the different way of measuring the reprojection error than the usual SfM methods.

The reprojection error is computed by mapping the points in the distorted domain to the points in the undistorted domain. The point of this change is to make the reprojection error computation less complex for inverse depth estimation. Using this technique enables the proposed method to perform on uncalibrated motion clips.

Fig. 2 shows the 3D reconstruction by our method and Hyowon Ha *et al.* [11].

3.1. Dense Matching Profile

The basic idea of the dense matching in the current paper is based on the Plane Sweeping method [22]. Different from plane sweeping based stereo matching methods, we estimate the $\boldsymbol{k}$-th plane directly from the set of ORB matches. If $(\boldsymbol{u_i}, \boldsymbol{v_i})$ and $(\boldsymbol{u_i} - \boldsymbol{k}, \boldsymbol{v_i})$ represents the pixel i in the left image and the correspondence match in the right image respectively, then the set of $\mathbb{M} = \{\boldsymbol{u_i}, \boldsymbol{v_i}, \boldsymbol{k}\}$ denotes the match of the two pixels.

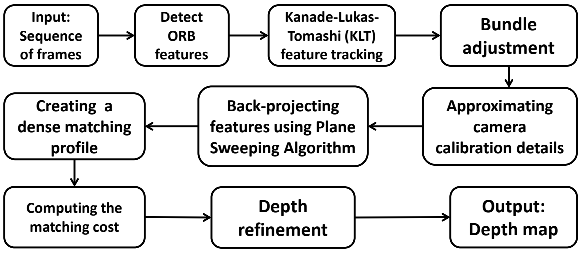

Input: Sequence of frames
Detect ORB features
Kanade-Lukas-Tomashi (KLT) feature tracking
Bundle adjustment
Approximating camera calibration details
Back-projecting features using Plane Sweeping Algorithm
Creating a dense matching profile
Computing the matching cost
Depth refinement
Output: Depth map

Figure 1. General overview of the proposed SfSM

At the pixel $(\boldsymbol{u_i}, \boldsymbol{v_i})$, the disparity is $\mathfrak{D}(\boldsymbol{u}, \boldsymbol{v}) = \mathfrak{p} \dotplus (\boldsymbol{u_i}, \boldsymbol{v_i}, \mathbf{1})$, where $\mathfrak{p} = (\mathfrak{p}_1, \mathfrak{p}_2, \mathfrak{p}_3)$ represents the plane.

To compute the sequence of disparity planes, a segmentation tree [28] is used. The overall objective function in the proposed method which is being minimized by the segmentation tree is:

$$\sum_i^n \mathcal{E}(\boldsymbol{m}_i, \mathfrak{p}_{j_i}) \qquad (1)$$

where $\boldsymbol{m_i} \in \mathbb{M}$ and $\boldsymbol{m_i}$ is part of the plane $\boldsymbol{j_i}$. The goal of this function is to measures the error between the true disparity at $\boldsymbol{m}$ and the disparity generated by the plane. $\boldsymbol{k}$-th disparity plane is computed by minimizing this function using a graph $\boldsymbol{G}$. The graph $\boldsymbol{G}$ is constructed by connecting each node $\boldsymbol{m}$ to its ten nearest neighbours computed by Euclidean distance.

3.2. Matching Cost and Plane Sweeps

At the first step, the frame $\boldsymbol{k}$ is resampled into an $[\boldsymbol{x}, \boldsymbol{y}]$ area from frame $\boldsymbol{k} + \mathbf{1}$ using B-Spline interpolation. The correlation score of $\boldsymbol{\mathcal{N}}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{k})$ is obtained over 5×5 patches. The score is turned into the pixel-wise matching cost as:

$$\boldsymbol{\mathcal{C}}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{k}) = \mathbf{1} - \boldsymbol{\mathcal{N}}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{k}) \qquad (2)$$

where $\boldsymbol{\mathcal{N}}$ refers to Normalized Cross Correlation and $\boldsymbol{\mathcal{C}}$ refers to the matching cost.

The raw cost is converted from pixel cost to the aggregated volume cost using adaptive cost aggregation [29].

As it is common in most of the stereo algorithms, the cost volume is computed as:

$$\boldsymbol{\mathcal{C}}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{k}) = \sum_{(x,y)} \boldsymbol{\mathcal{C}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{k}) \qquad (3)$$

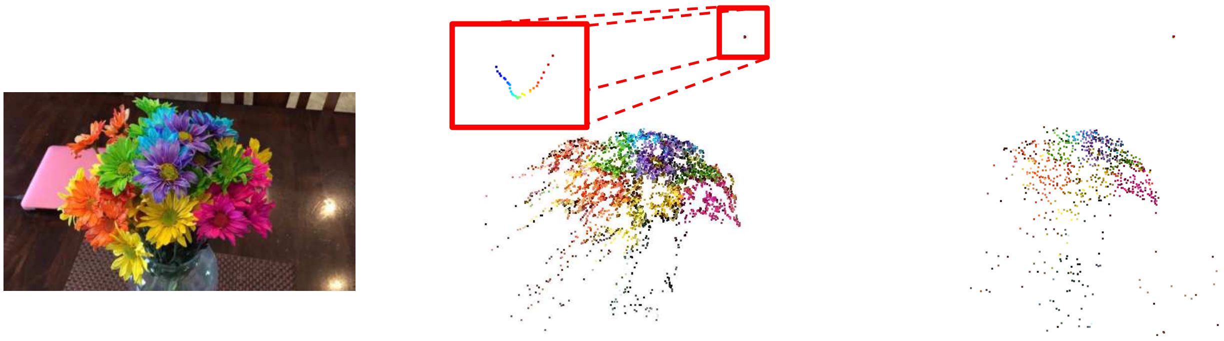
But this assumption has a requirement that the surface has to be facing the camera and this makes the pixels surrounding a patch to have almost the same disparity value. The restriction for this assumption arises from the common and important challenge of handling the occlusions along the boundaries in stereo matching methods. To resolve this issue, the cost volume is computed by aggregating the cost based on the color and similarity features. The matching cost from the resampled image is weighted by a similarity feature, in this paper the $\boldsymbol{\Delta\mathcal{C}}$ and the color difference between $\boldsymbol{p} = (\boldsymbol{u}, \boldsymbol{v})$ and $\boldsymbol{r} = [\boldsymbol{x}, \boldsymbol{y}]$ as $\boldsymbol{\Delta}\mathbb{C}$.

The weighting function $\boldsymbol{w}$ can be defined as:

$$\boldsymbol{w}(\boldsymbol{p}, \boldsymbol{r}) = \mathbf{exp}\left(\frac{-\boldsymbol{\Delta}\mathbb{C} - \boldsymbol{\Delta\mathcal{C}}}{\boldsymbol{t}}\right) \qquad (4)$$

where $\boldsymbol{t}$ is the weighting constant. The basic idea in Eq. 4 is to aggregate the matching cost based on color and feature similarity (geometric proximity). Considering a pixel $\boldsymbol{p}$ and pixel $\boldsymbol{r}$, the matching cost from $\boldsymbol{r}$ is weighted by the color difference between $\boldsymbol{p}$ and $\boldsymbol{r}$, and the Euclidian distance between $\boldsymbol{p}$ and $\boldsymbol{r}$ on the image plane. The computed aggregated cost from the pixel-wise cost is:

$$\mathfrak{C}(\boldsymbol{p}, \boldsymbol{k}) = \frac{\sum_{r,r'} \boldsymbol{w}(\boldsymbol{p},\boldsymbol{r})\boldsymbol{w}'(\boldsymbol{q},\boldsymbol{r}')\boldsymbol{\mathcal{C}}(\boldsymbol{p},\boldsymbol{k})}{\sum_{r,r'} \boldsymbol{w}(\boldsymbol{p},\boldsymbol{r})\boldsymbol{w}'(\boldsymbol{q},\boldsymbol{r}')} \qquad (5)$$



a. A frame of the sequence

b. **Our** reconstructed 3D point cloud and the estimated camera trajectory – Side view

c. Reconstructed 3D point cloud by **Hyowon Ha *et al.***[11] – Side view

Figure 2. Comparison of our 3D reconstruction with Hyowon Ha *et al.* [11]

where $\boldsymbol{p}$ and $\boldsymbol{q}$ are the matching pixels and $\boldsymbol{r}'$ is the support region of $\boldsymbol{q}$. Eq. 5 represents the weighted sum of per pixel which is used as cost aggregation.

Once the cost volume is computed, the initial disparity map $\boldsymbol{\mathcal{D}}$ is obtained by parametrizing the plane equation in pixel level with local disparity values. The condition for choosing the local disparity is minimizing the total aggregation cost.

Although $\boldsymbol{\mathcal{D}}$ has a quite reasonable depth values but it still can be noisy and the structure of the depth map can suffer from inaccurate edges and corners. To solve this issue and handle the probable occlusions, we define 2 terms as the smoothness term and the data term.

The smoothness term for pixels $\boldsymbol{p}$ and $\boldsymbol{q}$ and the displacement vector $\boldsymbol{v}$ is defined as:

$$\boldsymbol{S}(\boldsymbol{v_p}, \boldsymbol{v_q}) = \boldsymbol{w_{pq}} \, \mathbf{min}\left(\left\|\boldsymbol{v_p} - \boldsymbol{v_q}\right\|_2^2, \boldsymbol{t}\right) \qquad (6)$$

where $\boldsymbol{w_{pq}}$ is the weighting variable computed by the color similarity of the patch surrounding $\boldsymbol{p}$ and pixel $\boldsymbol{q}$. $\boldsymbol{t}$ is the reduction threshold.

This term has the most influence on occlusion handling by propagating the cost from the non-occluded pixels to occluded pixels based on their similarity.

Following the smoothness term, the data term is defined as:

$$\boldsymbol{d_p}(\boldsymbol{v_p}) = \begin{cases} \left\|\boldsymbol{v_p} - \boldsymbol{v_p^*}\right\|_2^2 & \boldsymbol{p \; is \; non-occluded} \\ \mathbf{0}, & \boldsymbol{otherwise} \end{cases} \qquad (7)$$

where $\boldsymbol{v_p^*}$ is the initial displacement vector.

Defining these 2 terms handle more than 96% of the occlusions but still, there are some missing parts, specifically around the boundaries of the objects which cause an inaccurate edge structure in the depth map. Considering a pixel $\boldsymbol{p}$ located in the occluded area. We try to estimate its disparity value by using a small patch $\boldsymbol{\mathcal{H}(p)}$ with known disparity values, centered at $\boldsymbol{p}$.The disparity value of $\boldsymbol{p}$ can be estimated by the following equation:

$$\boldsymbol{D_p} = \boldsymbol{D_q} + \langle \boldsymbol{g}\boldsymbol{D_q}, \boldsymbol{p} - \boldsymbol{q} \rangle \qquad (8)$$

where $\in \boldsymbol{\mathcal{H}(p)}$, $\boldsymbol{D_q}$ and $\boldsymbol{g}\boldsymbol{D_q}$ are the disparity value and gradient respectively. $\langle , \rangle$ represents the inner product operation.

This estimation is done for all the pixels in $\boldsymbol{\mathcal{H}(p)}$ and at the end the final disparity map of $\boldsymbol{p}$ is obtained by:

$$\boldsymbol{D_p} = \frac{\sum_{q \in \mathcal{H}(p)} \omega_{pq}[D_q] + \langle \boldsymbol{g}D_q, p - q \rangle}{\sum_{q \in \mathcal{H}(p)} \omega_{pq}} \qquad (9)$$

where $\boldsymbol{\omega_{pq}} = \boldsymbol{w_{pq}}$ is the weighting function and it is defined as:

$$\boldsymbol{\omega_{pq}} = \boldsymbol{\omega_{ds(pq)}} \boldsymbol{\omega_{cl(pq)}} \qquad (10)$$

where $\boldsymbol{\omega_{ds(pq)}}$ denotes the distance term and $\boldsymbol{\omega_{cl(pq)}}$ color similarity term.

$$\boldsymbol{\omega_{ds(pq)}} = \mathbf{exp}\left(-\frac{\|p-q\|^2}{2\alpha^2}\right) \qquad (11)$$

$$\boldsymbol{\omega_{cl(pq)}} = \mathbf{exp}\left(-\frac{\left\|r_p - r_q\right\|^2}{2\beta^2}\right) \qquad (12)$$

where $\boldsymbol{r_p}$ and $\boldsymbol{r_q}$ are the color values of the pixels $\boldsymbol{p}$ and $\boldsymbol{q}$ respectively. $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are constant values specified experimentally.

When corresponding matching pixels have dissimilar colors because of illumination variations, the inaccurate disparity map is generated. Adding the color similarity term to the weighting function helps to handle this issue.

To treat the probable artifacts caused by plane sweeping algorithm due to the over/under sampling, the inter frame motion estimation problem is reformulated to be optimized over image intensity function for sequence of frames. The formulation computes the cost over all pixels of the reference frame. Through this formulation a geometrical fidelity is checked for patch of pixels. The fidelity check is based on consistency of the normal directions between neighboring pixels to make sure they have similar surface normal vector. The correlation between the normal vectors of the center pixel and neighboring pixels can lead optimization to refine the depth map.

3.3. Final Depth Refinement

After computing the final depth map from the previous step, it is refined by the guided joint filter presented in [30]. The filter in [30] is based on the mutual information. The mutual information guides the weighted median filter to follow the structure of the RGB image while filtering the correspondence depth map. To keep the valid depth values and just filtering the false ones, window selection step of the median filter is designed to be adaptive using the joint histogram. The probable remaining artifacts after the adaptive weighted median filter are being eliminated by normalized interpolated convolution in diffused image domain.

Beside the performance of this filter in occlusion handling, it helps the depth map to follow the image structure more precisely. Without defining any limitations, for small parallax including slow-enough motion, or far-enough objects, or fast-enough temporal sampling, occluded areas are small. Our experiments show that the mentioned filter guarantees intra-object occlusion handling accurately even in wide baseline motions. The failure of the filter might occur in the case of large inter-object occlusion. Generally in small motions the main occlusion to deal with is intra-object. Although it is worth pointing out that the filter is able to handle relatively good amount of inter-object occlusions unless there is a considerable displacement or off axis parallax.

4. Experiments and Evaluation

In this paper, the experiment and evaluation is done in 2 parts. First, the proposed method is evaluated for small motions and in the second part, it is evaluated for stereo image sets. The first comparison is done against Hyowon Ha *et al.* [11] and Kevin Karsch *et al.* [31] and the second comparison is done against 3DMST [32] and APAP-Stereo [33] stereo matching algorithms ranked in Middlebury stereo benchmark [34], training dense section.

For the first part, the dataset from [11] is used and we also provided 10 other small motion clips using the devices shown in Table 1. The motion clips are available to download at (`goo.gl/m5QohE`).

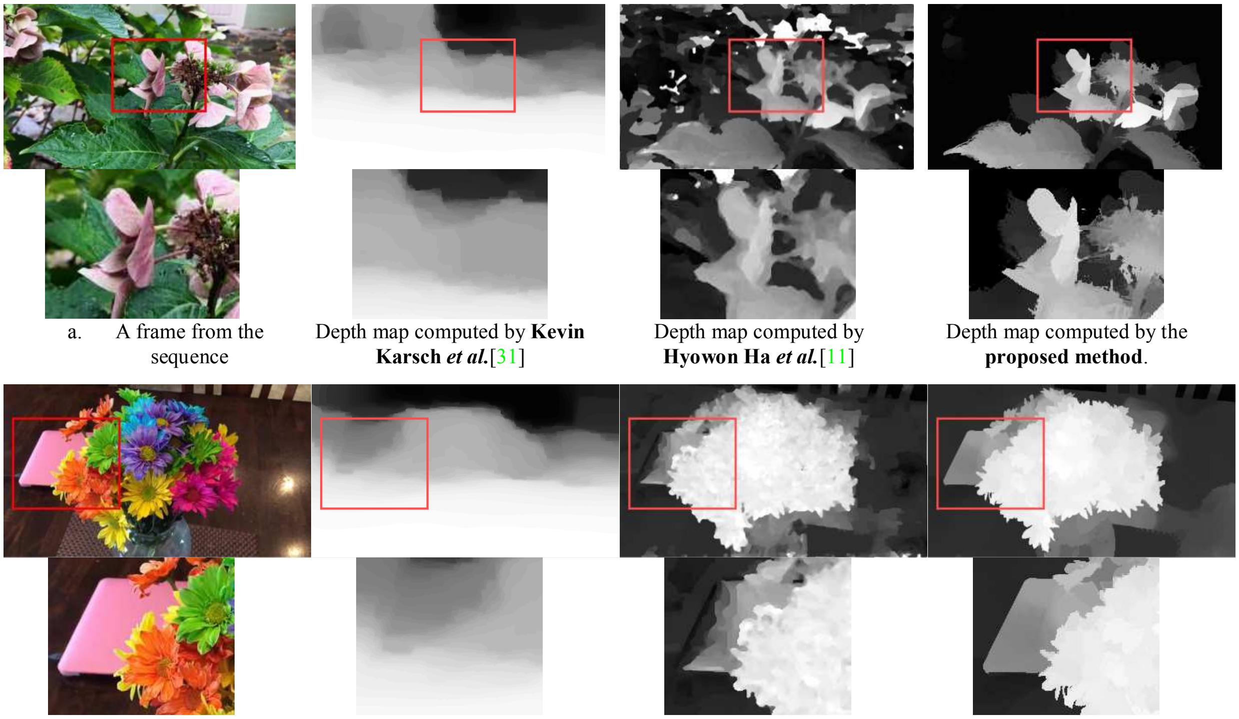
There is no ground truth in this form of evaluation, but the performance of the proposed method makes it possible to show the visual comparison with 2 other methods.

Fig. 3 shows the depth map computed by Hyowon Ha *et al.* [11], Kevin Karsch *et al.* [31] and our method. These images show the performance of the proposed method in terms of accuracy of the depth along edges and the depth values on the surface of objects in the case of small motions and small baseline.

The results by Hyowon Ha *et al.* [11] and Kevin Karsch *et al.* [31] have inaccurate depth values along the edges and corners of the objects as seen in Fig.3.a and Fig.3.b. Note that due to the very small baseline between the frames these methods distinguish foreground information better than background information.

In some cases as shown in Fig.3.b, the depth map estimated by these methods are suffering from inaccurate depth values on an object's surface or the depth values of the background and foreground objects are mixed together which cause inaccurate performance in segmentation and 3D reconstruction applications.

Fig. 4 shows how the inaccurate depth values along the edges can generate a faulty 3D structure. The highlighted patches show a part of the 3D textured mesh generated based on the reference frame and the corresponding depth map.



a. A frame from the sequence

Depth map computed by **Kevin Karsch *et al.***[31]

Depth map computed by **Hyowon Ha *et al.***[11]

Depth map computed by the **proposed method**.

b. A frame from the sequence

Depth map computed by **Kevin Karsch *et al.***[31]

Depth map computed by **Hyowon Ha *et al.***[11]

Depth map computed by the **proposed method**.

Figure 3. Comparison of the depth from small motion with state-of-the-art methods

Table 1. Devices used for making our own dataset

	Device	Resolution	fps
1	iPhone6 Plus	1080p	60
2	iPhone6 Plus	1080p	30
3	iPhone7	1080p	30
4	iPhone7	720p	30
5	iPhone7 Plus	1080p	30
6	iPhone7 Plus	4K	30

The 3D mesh generated based on the depth map by Hyowon Ha *et al.* [11] is suffering from missing parts on objects' surfaces which is caused by inaccurate depth values on reference patches.

For the second part of the comparison, we evaluated the performance of the proposed method for a set of stereo images. In this case, we considered the left and right images as a sequence of frames, 2 frames instead of processing 30 frames by considering the fact that the method is designed to perform on small baseline motions while the higher number of frames provides the higher number of inliers at the feature matching step. Note that more experiments are done on ordinal camera motions recorded by authors [11]. The depth map in Fig.1.d and Fig.1.c in the *Appendix_1* (goo.gl/fqqUxk) is generated using only 2 frames of the real camera motion which is captured by users. That's why the result of the method [11] in Fig.1.d and Fig.1.c in *Appendix_1* is different from what is published in the main paper [11]. The depth map in [11] is computed using 30 frames, but in this paper only 2 frames are used. That shows the superior performance of the proposed method.

To have an accurate evaluation at this part, we used 7 pairs of stereo images from Middlebury stereo benchmark with the corresponding ground truth depth maps. Fig. 5 represents the visual comparison of this evaluation. Fig.5.a and Fig.5.b show how the proposed method is capable of keeping the structure of the reference image in the depth map, especially important features like edges and corners in comparison with top stereo matching algorithms.

The accuracy of the estimated depth by each method has been evaluated against the ground truth which is provided by the benchmark and the numerical results are presented in Table 2. These results illustrate the competitive performance of the proposed method in terms of accuracy of the depth along edges and the depth values on the surface of the objects against top algorithms in Middlebury benchmark. Although there is still the potential for this method to be improved as it is not performing perfectly in some cases.

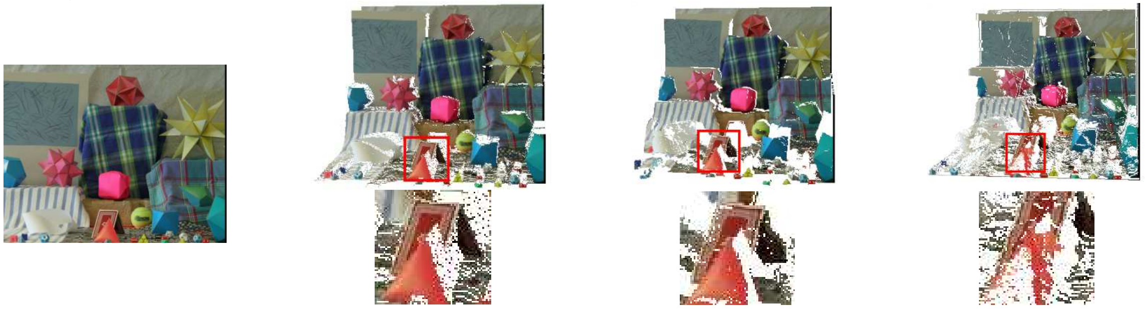
To find more visual/extended numerical results and the higher resolution version of the images presented in Fig. 3 and Fig. 5 please refer to *Appendix_1.*

For evaluation purposes, 4 metrics including PSNR, RMSE, Universal Quality Index (UQI) [35] and Structural Similarity Index (SSIM) [36] are used. Table 2 presents the average numerical comparison of the methods per metric on the chosen stereo sets from the benchmark. The extended numerical results are presented in *Appendix_1.*

Fig. 6 represents the SSIM and UQI maps of the depth map generated by each method from the images in Fig. 5. The SSIM map show how similar is the structure of the computed depth map to the ground truth. The lighter and darker pixel values show more and less structural similarity to the ground truth respectively.

The general quality of the generated depth maps in comparison with the ground truth is shown as UQI map. The lighter and darker pixel values show more and less similarity to the ground truth respectively.

As it is illustrated in Fig. 6, the proposed method is estimating depth maps relatively close to the ground truth in both structural and quality indices as there are larger areas covered with lighter values. The areas presented in dark show how far the depth values are from ground truth based on SSIM and UQI maps.



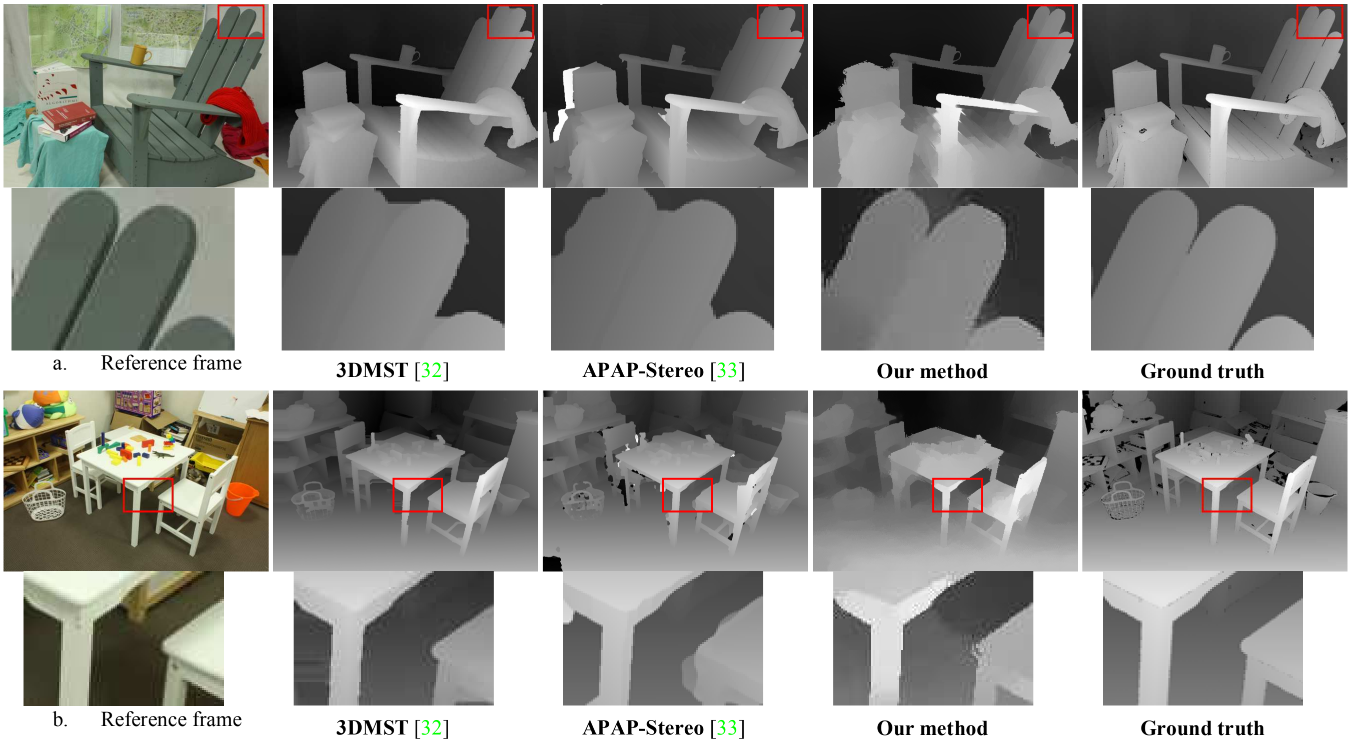
Reference frame

3D textured mesh – **Ground truth**

3D textured mesh – **Our method**

3D textured mesh – **Hyowon Ha *et al.*[11]**

Figure 4. Comparison of the 3D textured mesh based on the depth maps generated by the proposed method and Hyowon Ha *et al.* [11]



a. Reference frame

3DMST [32]

APAP-Stereo [33]

Our method

Ground truth

b. Reference frame

3DMST [32]

APAP-Stereo [33]

Our method

Ground truth

Figure 5. Comparison with 3DMST [32] and APAP-Stereo [33] based on Middlebury benchmark

There are still considerable parts in the depth maps generated by the proposed technique which look far from ground truth but the results are reasonably close to the top stereo matching algorithms.

SSIM maps in Fig. 6 show that the structure of the reference frames including the sharp edges and corners, is respected in the estimated depth map and this is one of the advantages of the proposed method.

Occluded regions are important features in depth extraction methods [37][38]. Unlike most of the current algorithms that are not able to handle this issue, the proposed method can estimate the information on invisible scene components. Fig. 6 illustrates another important advantage of the proposed technique which is the acceptable performance on lower fps motions such as 2 frame stereo images. The presented cost function makes the algorithm capable of processing motions with wider baseline.

The robustness of the proposed method is also evaluated by considering the magnitude of the baseline and number of the frames. The result illustrates that the algorithm can generate depth with the similarity of ~75% to the ground truth as long as the magnitude of the baseline is greater than ~6% of the nearest scene depth and the number of frames captured exceeds 2 frames.

Table 2. Numerical comparison of the methods/average per metric for seven stereo set

	PSNR	**RMSE**	**UQI**	**SSIM**
Ours	17.281	35.491	0.87	0.70
3DMST [32]	18.315	29.975	0.89	0.82
APAP-Stereo [33]	18.734	28.672	0.95	0.85

5. Conclusion

This paper has presented an accurate approach for computing the depth map from narrow baseline motion clips.

Six important contributions have been proposed in this work as follows:

General Contributions:

1. Generally in small motions, the feature tracker can obtain more inliers due to the small difference between the frames. However the number of inliers reduces when the baseline becomes wider and as the result the generated depth map becomes inaccurate. The modified cost function in the proposed method makes it capable of processing sequence of frames with the baseline up to 400 mm while most of the methods in this field fail for the baselines wider than ~12 mm.
2. Accurate performance for $frame \geq 2$

3. Occlusion handling by respecting the structure of the reference frame.

Technical Contributions:

1. New data and smoothness terms are defined to recondition cost volume and cost aggregation function.
2. Proposed cost propagation is formulated as energy minimizer function for depth on each pixel point.
3. The proposed method can approximate non-planar surfaces, while being robust against depth outliers and occlusion.

This practical application has the potential to be used in smartphone cameras. These cameras are designed to gather image frames before and after a user initiate a capture sequence. The 3D information obtained by this method can be used for synthetic defocus applications, object detection and segmentation purposes and scene analyses and understanding.

Unlike other techniques, the 3D points generated by the proposed method at the background of a scene don’t have high uncertainty. This gives a uniform and continuous shape to the point cloud from the closest to the furthest point visible to the camera.

A range of different experiments on both wide and narrow baselines have been conducted which proved that the proposed method exhibits improved performance over state of the art methods. In addition this method is sufficiently robust to perform adequately at low frame rates and with a small number of input images.

With respect to the performance and accuracy of the studied method, there is still the computational time of this technique which has to be considered as a trade-off. The method has been tested on a device equipped with Intel i7-5600U @ 2.60GHz CPU and 16 GB RAM. The whole process of computing the 3D structure and depth map take about 6-8 minutes. The most expensive part of the method is the bundle adjustment optimization which is takes around 4-5 minutes on high resolution images and motivates our future research activities to make this method suitable for real-time applications. The full evaluation of this method requires a dataset of video sequences with valid ground truths which at the moment is not publicly available. As part of our future work on this topic we would like to provide a dataset of video sequences with the ground truths for close-range scenes using ToF cameras.

Acknowledgement

The research work presented here was funded under the Strategic Partnership Program of Science Foundation Ireland (SFI) and co-funded by SFI and FotoNation Ltd. Project ID: 13/SPP/I2868 on “*Next Generation Imaging for Smartphone and Embedded Platforms*”.

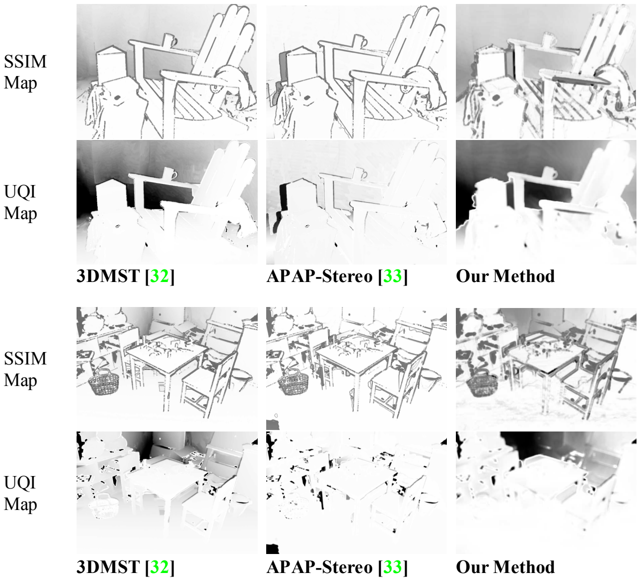


Figure 6. Comparison of SSIM and UQI maps

References

[1] David E. Jacobs, Jongmin Baek, Marc Levoy, “Focal Stack Compositing for Depth of Field Control”, *Stanford Computer Graphics Laboratory Technical Report* 2012-1. October, 2012.

[2] Lin, Haiting, Can Chen, Sing Bing Kang, and Jingyi Yu, "Depth recovery from light field using focal stack symmetry", *In Proceedings of the IEEE International Conference on Computer Vision*, pp. 3451-3459, 2015.

[3] Suwajanakorn, Supasorn, Carlos Hernandez, and Steven M. Seitz, "Depth from focus with your mobile phone", *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3497-3506, 2015.

[4] Bailey, Stephen W., Jose I. Echevarria, Bobby Bodenheimer, and Diego Gutierrez, "Fast depth from defocus from focal stacks", *The Visual Computer,* 31, no. 12, pp. 1697-1708, 2015.

[5] S. Im, H. Ha, G. Choe, H. G. Jeon, K. Joo, and I. S. Kweon, "High Quality Structure from Small Motion for Rolling Shutter Cameras," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 837-845, 2015.

[6] Barron, Jonathan T., Andrew Adams, YiChang Shih, and Carlos Hernández. "Fast bilateral-space stereo for synthetic defocus." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4466-4474. 2015.

[7] F. Yu and D. Gallup, "3D Reconstruction from Accidental Motion," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3986-3993, 2014.

[8] Tron, Roberto. "A Factorization Approach to Inertial Affine Structure from Motion." *arXiv preprint arXiv*:1608.02680, 2016.

[9] Agudo, Antonio, Francesc Moreno-Noguer, Begoña Calvo, and José María Martínez Montiel. "Sequential non-rigid structure from motion using physical priors." *IEEE transactions on pattern analysis and machine intelligence*, 38, no. 5 , pp. 979-994, 2016.

[10] Schonberger, Johannes L., and Jan-Michael Frahm. "Structure-from-motion revisited." *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4104-4113, 2016.

[11] H. Ha, S. Im, J. Park, H. G. Jeon, and I. S. Kweon, "High-Quality Depth from Uncalibrated Small Motion Clip," in *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5413-5421, 2016.

[12] N. Josh and L. Zitnick, "Micro-Baseline Stereo," *Microsoft Research Technical Report,* vol. MSR-TR-2014-73, May 2014.

[13] G. Zhang, H. Liu, Z. Dong, J. Jia, T. T. Wong, and H. Bao, "Efficient Non-Consecutive Feature Tracking for Robust Structure-From-Motion," *IEEE Transactions on Image Processing,* vol. 25, pp. 5957-5970, 2016.

[14] H. Guan and W. A. P. Smith, "Structure-From-Motion in Spherical Video Using the von Mises-Fisher Distribution," *IEEE Transactions on Image Processing,* vol. 26, pp. 711-723, 2017.

[15] H. Zhou, K. Ni, Q. Zhou, and T. Zhang, "An SfM Algorithm With Good Convergence That Addresses Outliers for Realizing Mono-SLAM," *IEEE Transactions on Industrial Informatics,* vol. 12, pp. 515-523, 2016.

[16] Triggs, Bill, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. "Bundle adjustment—a modern synthesis." *In International workshop on vision algorithms*, Springer Berlin Heidelberg, 1999, pp. 298-372. 1999.

[17] Y. Dai, H. Li, and M. He, "Projective Multiview Structure and Motion from Element-Wise Factorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 35, pp. 2238-2251, 2013.

[18] P. Sturm and B. Triggs, "A factorization based algorithm for multi-image projective structure and motion," in *Computer Vision — ECCV '96: 4th European Conference on Computer Vision Cambridge, UK, April 15–18, 1996 Proceedings Volume II*, B. Buxton and R. Cipolla, Eds., ed Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 709-720, 1996.

[19] B. Triggs, "Factorization methods for projective structure and motion," in *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 845-851, 1996

[20] Ma, Lili, YangQuan Chen, and Kevin L. Moore. "Rational radial distortion models of camera lenses with analytical solution for distortion correction." *International Journal of Information Acquisition* 1, no. 02, pp. 135-147, 2004.

[21] Tamaki, Toru, Tsuyoshi Yamamura, and Noboru Ohnishi. "Unified approach to image distortion." In *Pattern Recognition, Proceedings. 16th International Conference on*, vol. 2, pp. 584-587. IEEE, 2002.

[22] R. T. Collins, "A space-sweep approach to true multi-image matching," in *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 358-363, 1996.

[23] S. N. Sinha, D. Scharstein, and R. Szeliski, "Efficient High-Resolution Stereo Matching Using Local Plane Sweeps," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1582-1589, 2014.

[24] N. Komodakis and N. Paragios, "Beyond pairwise energies: Efficient optimization for higher-order MRFs," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2985-2992, 2009.

[25] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 International Conference on Computer Vision*, pp. 2564-2571, 2011.

[26] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," presented at the Proceedings of the 7th international joint conference on Artificial intelligence - Volume 2, Vancouver, BC, Canada, 1981.

[27] P. J. Huber, "Robust Estimation of a Location Parameter," in *Breakthroughs in Statistics: Methodology and Distribution*, S. Kotz and N. L. Johnson, Eds., ed New York, NY: Springer New York, pp. 492-518, 1992.

[28] X. Mei, X. Sun, W. Dong, H. Wang, and X. Zhang, "Segment-Tree Based Cost Aggregation for Stereo Matching," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 313-320, 2013.

[29] L. Wang, M. Liao, M. Gong, R. Yang, and D. Nister, "High-Quality Real-Time Stereo Using Adaptive Cost Aggregation and Dynamic Programming," in *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, pp. 798-805, 2006.

[30] H. Javidnia and P. Corcoran, "A Depth Map Post-Processing Approach Based on Adaptive Random Walk With Restart," *IEEE Access,* vol. 4, pp. 5509-5519, 2016.

[31] K. Karsch, C. Liu, and S. B. Kang, "Depth Transfer: Depth Extraction from Video Using Non-Parametric Sampling," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 36, pp. 2144-2158, 2014.

[32] L. Li, X. Yu, S. Zhang, X. Zhao, and L. Zhang, "3D cost aggregation with multiple minimum spanning trees for stereo matching," Applied Optics, vol. 56, pp. 3411-3420, 2017.

[33] vision.middlebury.edu/stereo/eval3/

[34] D. Scharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," *International Journal of Computer Vision,* vol. 47, pp. 7-42, 2002.

[35] W. Zhou and A. C. Bovik, "A universal image quality index," *IEEE Signal Processing Letters,* vol. 9, pp. 81-84, 2002.

[36] W. Zhou, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing,* vol. 13, pp. 600-612, 2004.

[37] A. Humayun, O. Mac Aodha and G. J. Brostow, "Learning to find occlusion regions," In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2161-2168, 2011.

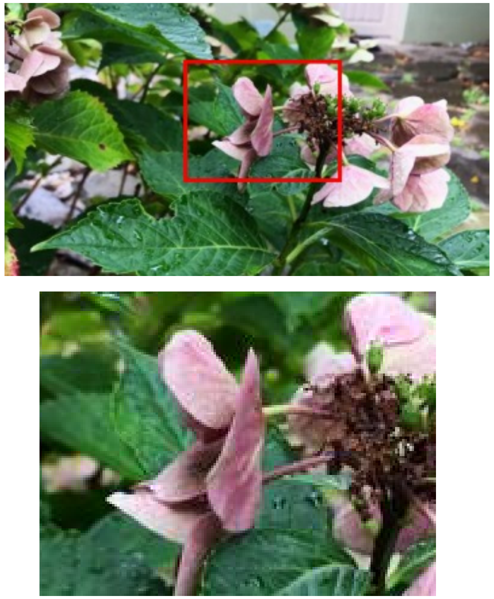
[38] H. Fu, C. Wang, D. Tao and M. J. Black, "Occlusion Boundary Detection via Deep Exploration of Context," In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 241-250, 2016.

Accurate Depth Map Estimation from Small Motions – Appendix 1

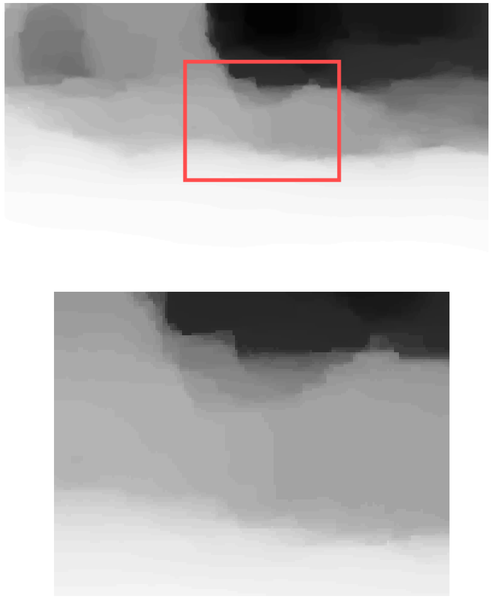
Hossein Javidnia
National University of Ireland, Galway
University Road, Galway, Ireland, H91 TK33
{h.javidnia1}@nuigalway.ie

Peter Corcoran
National University of Ireland, Galway
University Road, Galway, Ireland, H91 TK33
{peter.corcoran}@nuigalway.ie

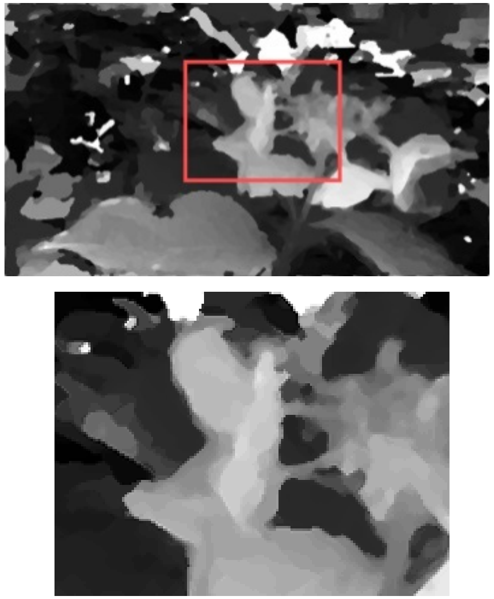
A: Narrow baseline motion (Extended version of Fig. 3 in the paper):



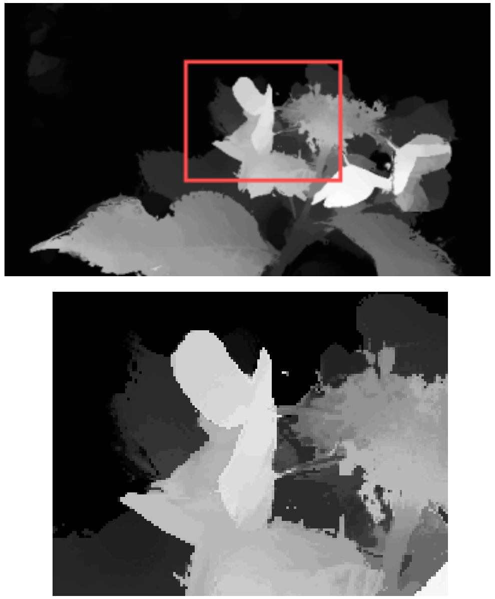
a. A frame from the sequence



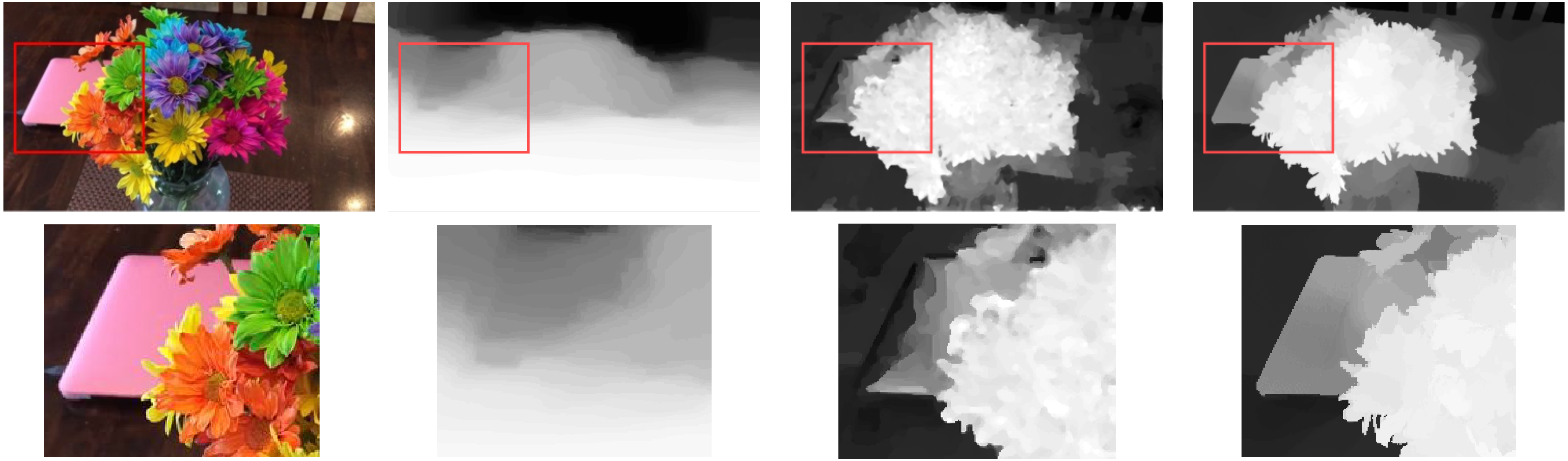
Depth map computed by Kevin Karsch *et al*.



Depth map computed by Hyowon Ha *et al*.



Depth map computed by the proposed method.

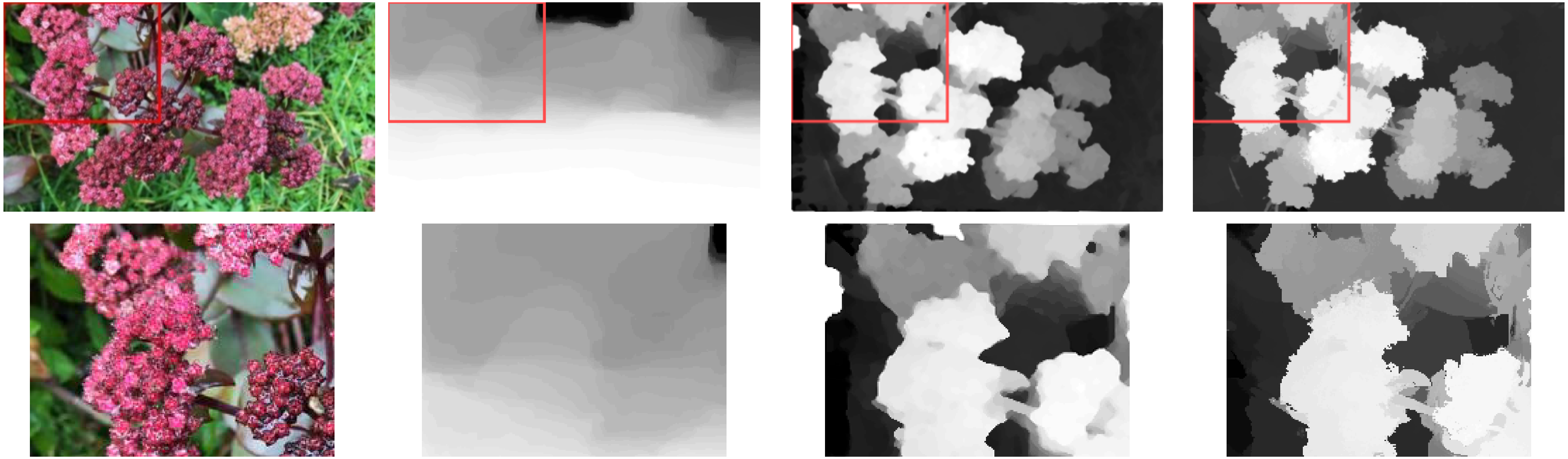


b. A frame from the sequence

Depth map computed by Kevin Karsch *et al*.

Depth map computed by Hyowon Ha *et al*.

Depth map computed by the proposed method.

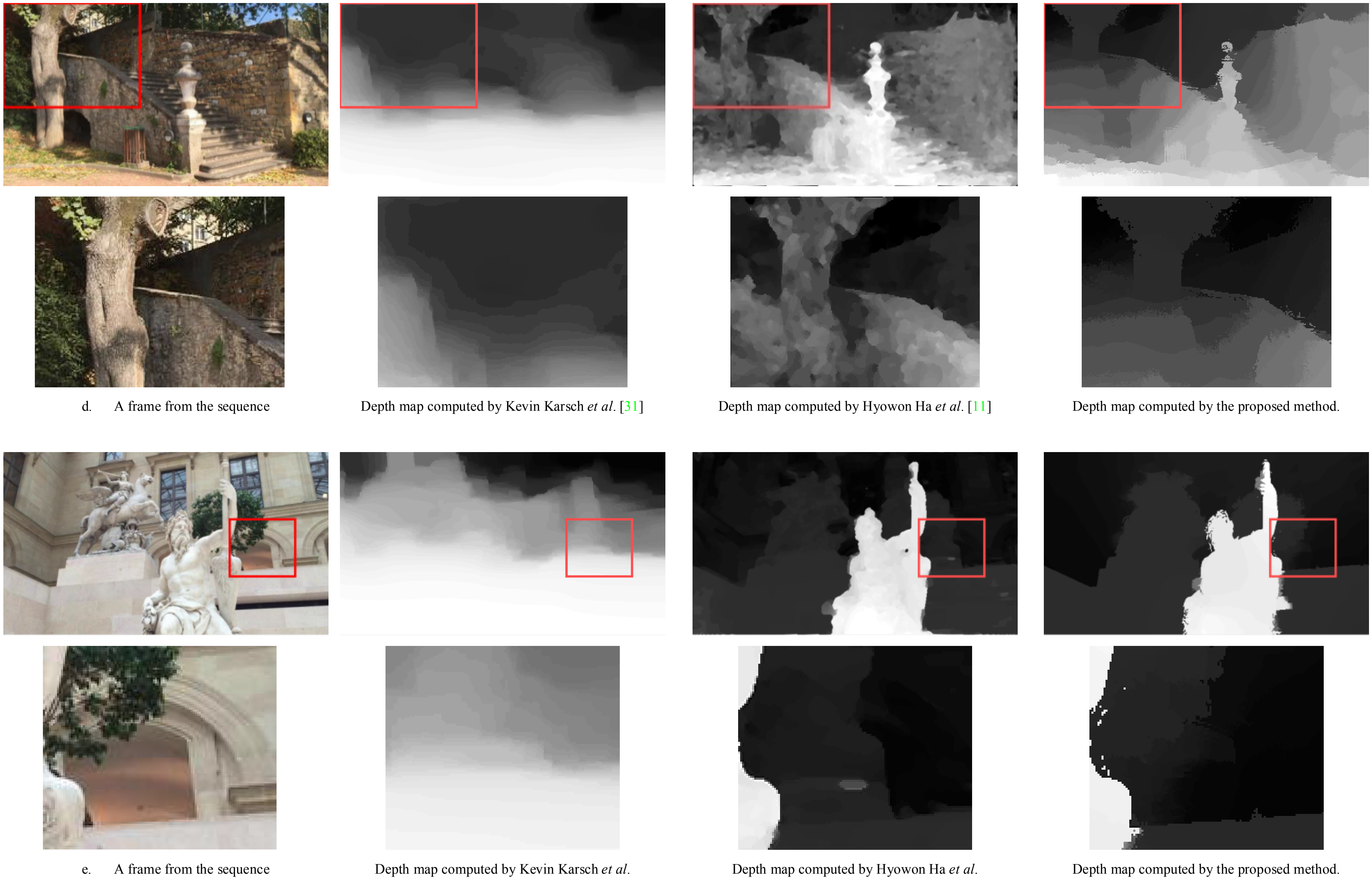


c. A frame from the sequence

Depth map computed by Kevin Karsch *et al*.

Depth map computed by Hyowon Ha *et al*.

Depth map computed by the proposed method.



d. A frame from the sequence

Depth map computed by Kevin Karsch *et al.* [31]

Depth map computed by Hyowon Ha *et al.* [11]

Depth map computed by the proposed method.

e. A frame from the sequence

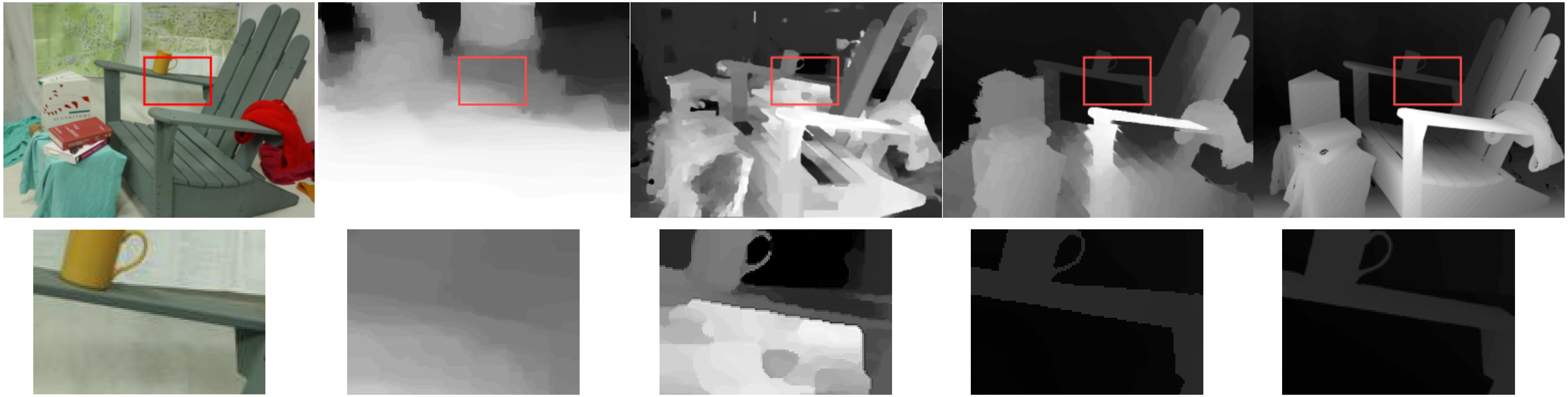
Depth map computed by Kevin Karsch *et al.*

Depth map computed by Hyowon Ha *et al.*

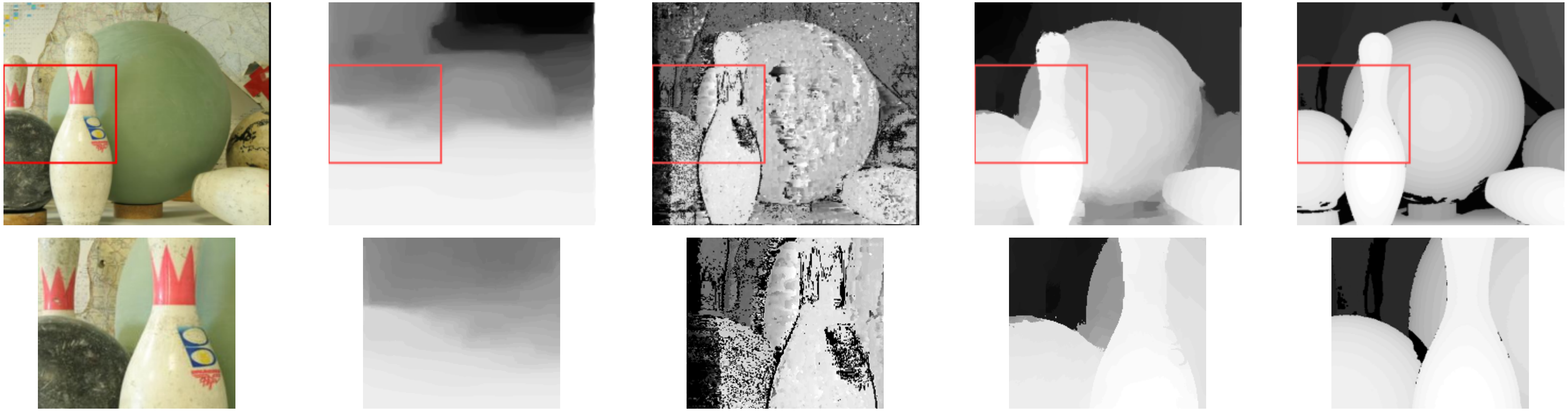
Depth map computed by the proposed method.

Figure 1. Comparison of the depth from small motion with state-of-the-art methods

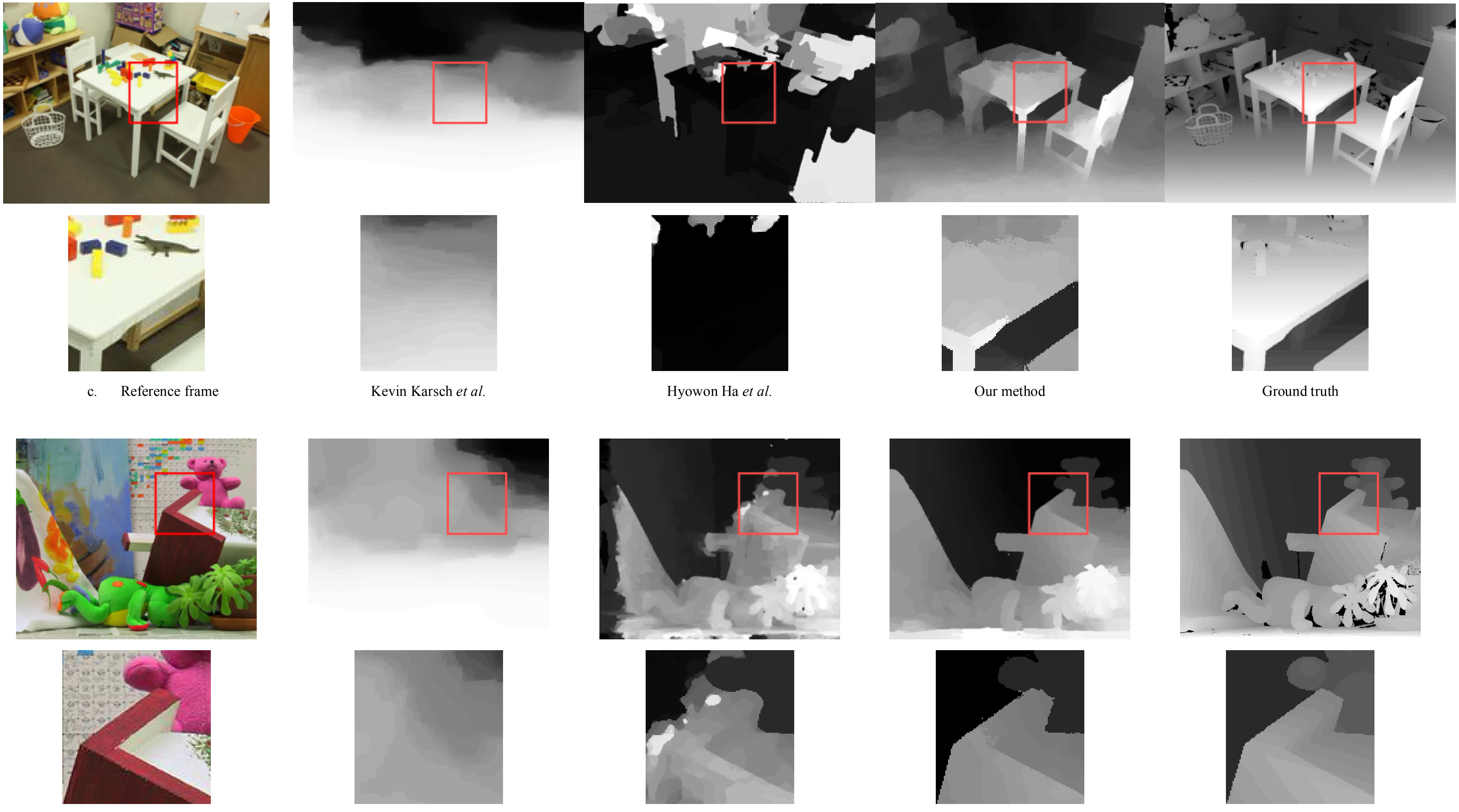
B: Wider baseline motion (Extended version of Fig. 5 in the paper):



a. Reference frame Kevin Karsch *et al.* Hyowon Ha *et al.* Our method Ground truth



b. Reference frame Kevin Karsch *et al.* Hyowon Ha *et al.* Our method Ground truth



c. Reference frame

Kevin Karsch *et al.*

Hyowon Ha *et al.*

Our method

Ground truth

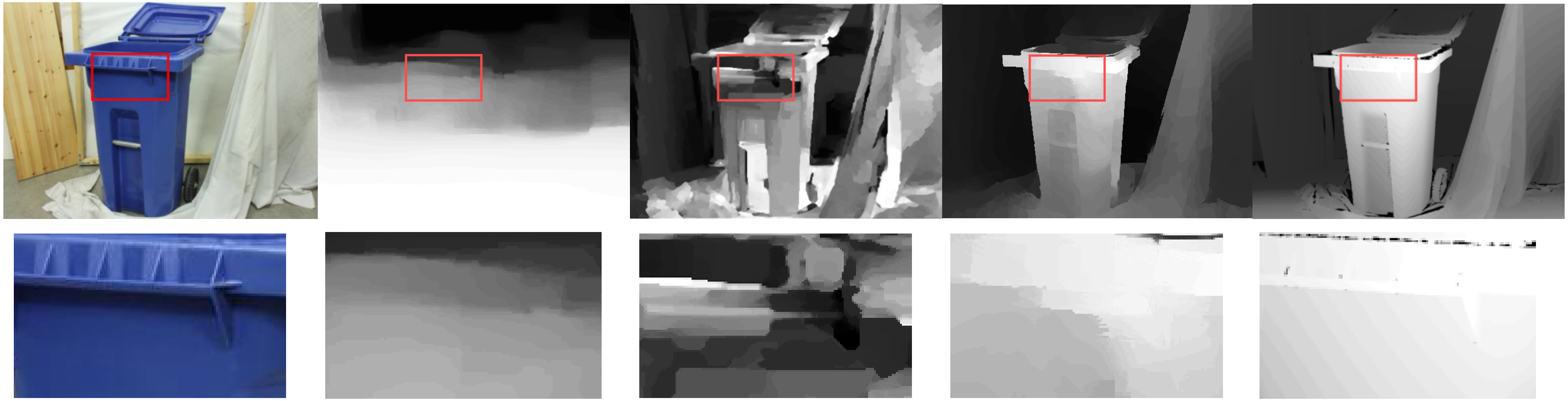
d. Reference frame

Kevin Karsch *et al.*

Hyowon Ha *et al.*

Our method

Ground truth



e. Reference frame

Kevin Karsch *et al.*

Hyowon Ha *et al.*

Our method

Ground truth

Figure 2. Comparison with Kevin Karsch et al. and Hyowon Ha et al. based on Middlebury benchmark

C: Extended numerical results of Table. 2 in the paper:

Table 1. Numerical comparison of the methods/stereo set (Colour coded)

a. PSNR values method/stereo set

	Adirondack	ArtL	Motorcycle	Piano	Playtable	Recycle	Teddy
Ours	19.33	19.4	16.31	14.62	18.27	16.94	16.1
3DMST	20.3976	17.8	20.2455	16.2395	18.6883	18.2445	16.6159
APAP-Stereo	18.0996	21.2	19.2963	16.9351	19.3516	19.6014	16.6411

b. RMSE values method/stereo set

	Adirondack	ArtL	Motorcycle	Piano	Playtable	Recycle	Teddy
Ours	27.53	27.3	38.99	47.37	31.09	36.26	39.91
3DMST	24.3591	29.2	24.7893	39.3156	29.6568	31.2116	31.2996
APAP-Stereo	31.7366	19.6	27.6518	36.2897	27.4763	26.6974	31.2092

c. UQI values method/stereo set

	Adirondack	ArtL	Motorcycle	Piano	Playtable	Recycle	Teddy
Ours	0.95	0.89	0.89	0.85	0.9	0.84	0.77
3DMST	0.80332	0.97	0.97527	0.88361	0.88142	0.80165	0.96016
APAP-Stereo	0.94389	0.99	0.96679	0.94349	0.95932	0.9606	0.92711

d. SSIM values method/stereo set

	Adirondack	ArtL	Motorcycle	Piano	Playtable	Recycle	Teddy
Ours	0.80	0.75	0.66	0.61	0.72	0.72	0.65
3DMST	0.80994	0.87	0.82063	0.77499	0.80362	0.79859	0.90701
APAP-Stereo	0.88369	0.88	0.81608	0.81688	0.83617	0.87918	0.87959

Appendix D: Application of Preconditioned Alternating Direction Method of Multipliers in Depth from Focal Stack

Journal of
Electronic Imaging

JElectronicImaging.org

Application of preconditioned alternating direction method of multipliers in depth from focal stack

Hossein Javidnia
Peter Corcoran

Hossein Javidnia, Peter Corcoran, "Application of preconditioned alternating direction method of multipliers in depth from focal stack," *J. Electron. Imaging* **27**(2), 023019 (2018), doi: 10.1117/1.JEI.27.2.023019.

Application of preconditioned alternating direction method of multipliers in depth from focal stack

Hossein Javidnia* **and Peter Corcoran**
National University of Ireland Galway, College of Engineering, Department of Electronic Engineering, Galway, Ireland

Abstract. Postcapture refocusing effect in smartphone cameras is achievable using focal stacks. However, the accuracy of this effect is totally dependent on the combination of the depth layers in the stack. The accuracy of the extended depth of field effect in this application can be improved significantly by computing an accurate depth map, which has been an open issue for decades. To tackle this issue, a framework is proposed based on a preconditioned alternating direction method of multipliers for depth from the focal stack and synthetic defocus application. In addition to its ability to provide high structural accuracy, the optimization function of the proposed framework can, in fact, converge faster and better than state-of-the-art methods. The qualitative evaluation has been done on 21 sets of focal stacks and the optimization function has been compared against five other methods. Later, 10 light field image sets have been transformed into focal stacks for quantitative evaluation purposes. Preliminary results indicate that the proposed framework has a better performance in terms of structural accuracy and optimization in comparison to the current state-of-the-art methods. *© 2018 SPIE and IS&T* [DOI: 10.1117/1.JEI.27.2.023019]

Keywords: focal stack; depth; regularization; synthetic defocus.

Paper 180012 received Jan. 5, 2018; accepted for publication Mar. 19, 2018; published online Apr. 6, 2018.

1 Introduction

The compact design of mobile cameras does not allow users access to lens properties such as the aperture. By having the control over the aperture in a camera, one can control the camera's depth of field (and the light flux entering the camera). This means the user can decide how much of an image remains in focus around an object. Figure 1(a) shows schematically the relation between the depth of focus (in image space) and depth of field (in object space). As shown in Fig. 1(c), small depth of field will make the main object in focus while the rest of the image will be less sharp. A large depth of field will keep the entire image sharp throughout its depth; this concept is shown in Fig. 1(d). When light rays from an out of focus point source enter a lens, the point on the object is focused into a circle on the image plane. This circle is called the circle of confusion (CoC), which is shown as C in Fig. 1(a).

The size of the CoC is used to measure the sharpness of an image. The bigger CoC shows that the point on the object is more out of focus. The diameter of CoC depends on focal length f, object distance ℓ_n (near point), the distance between the object point and the lens ℓ, and aperture diameter d. Therefore, the diameter of the CoC can be calculated using

$$C = \frac{df|\ell - \ell_n|}{\ell(\ell_n - f)}. \tag{1}$$

Figure 1(b) shows the relationship between the CoC and the object distance for an aperture $f/2.8$ mm for a specific camera model. The X axis in Fig. 1(b) represents the distance of the object points in focus and the Y axis shows how far the object would be in focus. For instance, if an object is located at the distance 1000 mm the value of CoC is 0, this means the object is fully in focus. If an object is located at the distance beyond 1200 mm then the 1.5 m setting of the camera should be used. If an object is located at the distance beyond 2000 mm then the 3 m setting of the camera should be used, and for the objects that are located at the distance beyond 6000 mm, the infinite setting of the camera should be used.

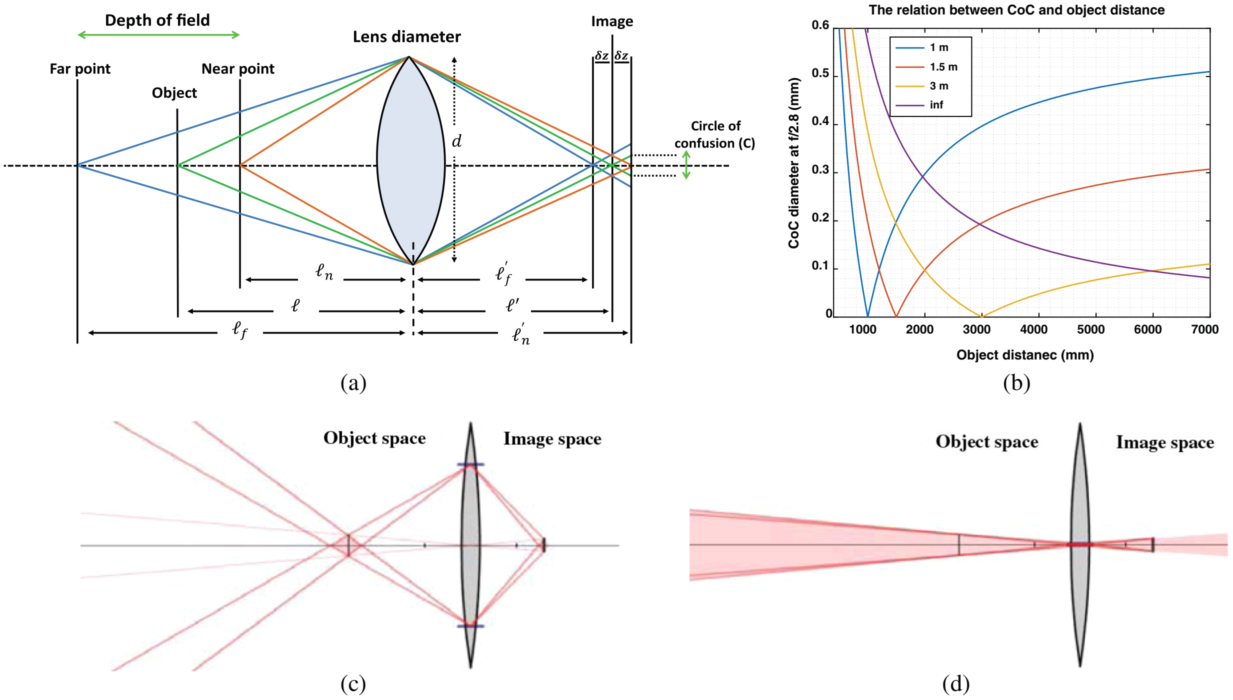
The adjustable aperture feature is available in DSLR cameras but smartphone cameras have a fixed aperture as they are designed for ease of portability, robustness, and low cost.

To overcome this shortcoming, postcapture image refocusing can be employed by using depth from focus (DfD)[1,2] and focal stack. The focal stack is a collection of images with different focus points, which correspond to different depth layers. The focal setting presenting the maximum sharpness of pixel p corresponds to the depth of the pixel or its distance to the camera. The combination of these images can generate the extended depth of field similar to the range being generated by optical properties of the camera. The accuracy of this effect is highly dependent on the accuracy of the corresponding depth map.

In handheld devices such as smartphones, a focal stack is generated by automatic focal plane sweeping to find the camera's best autofocus setting while taking photos. However, in dynamic scenes, the slight translation of the camera by users and their handshake can introduce motion parallax. The experiments indicate that it takes about $1/2$ to $1/3$ s for a camera to capture the full extent of its focus setting.

*Address all correspondence to: Hossein Javidnia, E-mail: h.javidnia1@nuigalway.ie

1017-9909/2018/$25.00 © 2018 SPIE and IS&T



(a)

(b)

(c)

(d)

Fig. 1 Demonstration of the relation between *F*-number and depth of field, CoC, and object distance. (a) Depth of field and CoC for a lens representing light collection optics in an imaging system. (b) The relation between CoC and the object distance/focus position. (c) Small depth of field with *F*-number 0.4. (d) Large depth of field with *F*-number 32.

This means that the local parallax met within this short time frame can be dismissed but yet an alignment procedure is required to compensate the parallax between the first and last frame of the focal stack. Generally, an aligned focal stack should be similar to a stack captured by a telecentric camera.

In this paper, we present a framework to compute and optimize the dense depth map from the high-resolution focal stack, which can be used to produce an accurate synthetic defocus. (The code is available in a Github repository: https://github.com/hosseinjavidnia/Depth-Focal-Stack). The framework initiates by taking a focal stack from a moving camera as the input and generating a stabilized image sequence. At the second step, the initial depth map is estimated from the stabilized focal stack. At the end, preconditioned alternating direction method of multipliers (PADMM) with a new cost function is applied to refine depth discontinuities and generate a noise-free depth map.

The proposed framework has several advantages in comparison to the state-of-the-art methods, such as

1. Fast and better convergence of the optimization function
2. High structural accuracy of the depth map
3. High performance in texture-less scenes
4. Accurate depth information along objects' boundaries and surface

The rest of this paper is organized as follows: Sec. 2 outlines the previous research. The proposed framework is explained in detail in Sec. 3. The evaluation results are presented in Secs. 4 and 5, which include conclusion and feature work.

2 Previous Work

A considerable amount of research focused on DfD control for decades.[3–8] Most of these methods concentrated on DfD/defocus or depth recovery from the focal stack on light field cameras.[5,8–10] Using light field cameras has an advantage of capturing simultaneous multiple views with variable focal points, which provide more accurate information about the depth of the scene; however, the images are captured in low resolution and in small aperture the value of signal-to-noise ratio (SNR) is significantly low.[8] The size of these cameras along with the mentioned challenges makes them inapplicable for handheld devices such as smartphones. Another disadvantage of light field cameras is the disability in handling occlusion due to the lack of lateral variation being captured in different viewpoints.[9]

A framework to recover depth from the focal stack is presented in Ref. 7 to handle images captured on smartphones. The focal stack is being aligned to make it as similar as a focal stack captured by a telecentric camera. Multilabel Markov random field optimization is used to generate all in focus image from the aligned stack. This method works quite well for the Lambertian scenes; however, the optimization problem during the calibration process is highly nonconvex, which makes this process considerably slow. The other problem with this method is the processing time of the nonlinear least squares minimization to jointly optimize the initially estimated aperture size, focal depths, focal length, and the depth map. The whole processing time of this method is ~20 min for 25 frames with 640 × 360 pixels resolution, which make this algorithm almost inapplicable as a smartphone application. The depth maps generated by this method suffer from inaccurate depth values on objects surface, especially on reflective surfaces. In some cases, the depth information along the boundaries of the foreground object is mixed with the values on the background and that might result in an inaccurate synthetic defocus.

The complexity of the nonconvex optimization is reformulated in Ref. 6 where DfD is presented as a variational problem by introducing a nonconvex data fidelity term and a convex nonsmooth regularization. The nonconvex minimization problem in Ref. 6 is aimed to be solved by a linearized alternating directions method of multipliers. This method has a superior performance in comparison to state-of-the-art methods, but the convergence of the optimization function happens very slowly and in a high number of iterations. Also, the depth map generated by this method suffers from inaccurate depth values on objects surface and missing edges and corners. The present research falls into the similar category where the problem of a noisy depth map is reformulated to a convex minimization problem to be solved by PADMM.

Some other approaches in this field have been proposed to facilitate the DfD applications by introducing coded focal stack photography[11] or coded aperture photography.[12,13] These methods require physical changes in the structure of the camera, and yet the generated depth maps suffer from lack of structural quality.

Persch et al.[14] proposed a variational approach for the problem of depth from defocus based on modeling of the image formation by featuring the thin lens model and preserving the crucial physical properties such as maximum-minimum principle for the intensity values. Later, the variational model is minimized using the multiplicative Euler–Lagrange. The proposed solution in Ref. 14 appears to generate false depth levels in relatively close scenes and in general, the depth profiles are likely to be affected by the color information as the robustification method employed in Ref. 14 uses the full-color information of the focal stack.

Pérez et al.[15] proposed a focal stack frequency decomposition algorithm from light field images based on the trigonometric interpolation principle as the discrete focal stack transform. The proposed method in Ref. 15 utilizes fast discrete Fourier transform to generate refocus planes in a reasonably fast computational time. The reverse of this transformation in studied in Ref. 16 where a focal stack is used to obtain a four-dimensional (4-D) light field image set using discrete focal stack transform.

Differently,[15] Mousnier et al.[17] presented an approach to reconstruct 4-D light field image sets from a stack of images taken by a fixed camera at different focal points. The algorithm initiates by calculating the focus map by utilizing region expansion with graph cut. Later, the depth map is estimated based on the calibration details of the camera, and it is used to reconstruct the epipolar images. The reconstructed epipolar images are used for refocusing purposes.

Bailey et al.[18] proposed a method to calculate depth from the focal stack by estimating the blur level for each pixel. The method initiates by applying a focus measure to each pixel in the stack. A normalized convolution is proposed to extrapolate the invalid blur estimates. Afterward, the per-pixel depth is calculated based on the blur estimations.

Jeong et al.[19] presented a postprocessing approach to refine the estimated depth map from two images captured with different focal points. The initial depth map is calculated using a depth from defocus algorithm. To improve the quality of the depth map, mean-shift clustering is applied to the first input image to obtain the segmented image. A single depth value is assigned to each segment of the image by averaging all depth values in the corresponding segment.

Surh et al.[20] presented a focus measure to determine how in focus a point is on an image. The shape of the focus measure introduced in Ref. 20 contains a disk that focuses on the pixel of interest and the ring that surrounds the disk. To estimate the depth map, the initial calculated cost volume is aggregated by employing tree-based cost aggregation method. Afterward, the depth discontinuity and unreliable depth labels are filtered based on the median of absolute deviation map and using tree-based cost aggregation method.

Focal stacks are also used to handle some of the optical features such as postcapture perspective shift and aperture reshaping. Alonso[21] developed a method in the Fourier domain for postcapture aperture reshaping in focal stacks. This allows users to change the blur shape for the out of focus points. This study came to the conclusion that by utilizing domain transformation methods such as Fourier it is possible to manipulate the optical setting of the camera. In another study, Alonso et al.[22] proposed a method for postcapture perspective shift reconstruction of a 3D scene from a focal stack. Unlike the computational approaches that estimate the depth map, the method in Ref. 22 takes advantage of depth-variant point-spread function to introduce the lateral [(x, y) plane) and axial (z plane] shifts.

3 Proposed Framework

3.1 *Focal Stack Alignment*

To compensate the misalignment of the input focal stack, we refer to epipolar homography alignment. To do that, we merge all the homographies into epipolar geometry. Considering there are j plane patches in an image and their corresponding maps in the second image are characterized as

$$\begin{aligned} H_1 &= s_1\mathcal{R}(I-\mathcal{T}N_1^T),\\ H_2 &= s_2\mathcal{R}(I-\mathcal{T}N_2^T),\\ &\dots\\ H_j &= s_j\mathcal{R}(I-\mathcal{T}N_j^T), \end{aligned} \tag{2}$$

where s is a scale factor, $\mathcal{R}$ is a 3×3 rotation matrix, I is the identity matrix, $\mathcal{T}$ is the second camera's translation from first camera's point of view, and $N(\mathfrak{n}_1, \mathfrak{n}_2, \mathfrak{n}_3)$ is the normal vector of the plane surface. Therefore, we can write

$$\begin{aligned} &\frac{s_1}{s_i}H_i - H_1 = s_1\mathcal{R}\mathcal{T}N_1^T - s_1\mathcal{R}\mathcal{T}N_i^T = \mathcal{K}\Delta N_i^T,\\ &\mathcal{K} = (\kappa_1\kappa_2\kappa_3)^T = \mathcal{R}\mathcal{T}, \end{aligned} \tag{3}$$

where $\Delta N_i = (\Delta\mathfrak{n}_1\,\Delta\mathfrak{n}_2\,\Delta\mathfrak{n}_3)^T = s_1(N_1 - N_i)$. Consequently, it can be concluded that

$$d_iH_i = H_1 + \mathcal{K}\Delta N_i^T \qquad i = 2{,}3, \dots, j, \tag{4}$$

where $d = \frac{1}{\|N\|}$ is the distance of the plane from the origin and H_1 represents the correlation between the basis homography and all the other homographies. The important feature of

Eq. (4) is that it reduces the number of independent parameters of a homography and makes them equal to the degree of freedom (dof) of a system with j planar surface. Generally, a homography includes 5 dof indicating the camera motion and 3 dof representing the plane surface normal. Assuming more than one plane between two images, then j homographies will have $8j$ parameters. Equation (4) decreases the number of the parameters to $5 + 3j$, which is equivalent of the total dof in a system with j planar surface.

Using Eq. (4) the motion estimation can break down into two parts:

First, considering that H_1 and $\mathcal{K}$ are fixed, it is possible to characterize ΔN_i and H_i by utilizing least square algorithm for each plane patches. To estimate ΔN_i we define two vectors as

$$\mathcal{V}_1 = (\kappa_1 x - \kappa_3 x x' \, \kappa_1 y - \kappa_3 y x' \, \kappa_1 - \kappa_3 x'),$$
$$\mathcal{V}_2 = (\kappa_2 x - \kappa_3 x y' \, \kappa_2 y - \kappa_3 y y' \, \kappa_2 - \kappa_3 y'). \tag{5}$$

So ΔN_i can be estimated using least squares method as

$$\mathcal{V}_1 \Delta N_i = x'(h_7 x + h_8 y + 1) - (h_1 x + h_2 y + h_3),$$
$$\mathcal{V}_2 \Delta N_i = y'(h_7 x + h_8 y + 1) - (h_4 x + h_5 y + h_6), \tag{6}$$

where h_{1-8} are the parameters of the homography matrix. (x, y, z) and (x', y', z') are the coordinates of the point $P = (X, Y, Z)$ in two camera frames as

$$x = X/Z \, y = Y/Z \, z = 1, \quad x' = X'/Z' \, y' = Y'/Z' \, z' = 1. \tag{7}$$

The second part is somehow the inverse process of the first part. Assuming ΔN_i is fixed, H_1 and $\mathcal{K}$ can be updated by utilizing another least squares process. To estimate H_1 and $\mathcal{K}$, we define three vectors as

$$E_i = (x \; y \; 1 \; 0 \; 0 \; 0 \; -xx' \; -yx' \; \Delta N_i P \; 0 \; -x'\Delta N_i P),$$
$$F_i = (0 \; 0 \; 0 \; x \; y \; 1 \; -xy' \; -yy' \; 0 \; \Delta N_i P \; -y'\Delta N_i P),$$
$$G = (h_1 \; h_2 \; h_3 \; h_4 \; h_5 \; h_6 \; h_7 \; h_8 \; k_1 \; k_2 \; k_3), \tag{8}$$

where $P = (x, y, z)$ is a point on the plane surface and $\Delta N_i P = (\Delta \mathfrak{n}_{i1} x + \Delta \mathfrak{n}_{i2} y + \Delta \mathfrak{n}_{i3} z)$. Therefore, it can be concluded that $E_i G^T = x'$ and $F_i G^T = y'$. Then, G can be estimated using least square process as

$$G^T = (Q^T Q)^{-1} Q^T \mathcal{B}, \tag{9}$$

where $\mathcal{B} = \begin{pmatrix} x'_{11} \\ y'_{11} \\ \vdots \\ x'_{j\mathfrak{n}} \\ y'_{j\mathfrak{n}} \end{pmatrix}$ and $Q = \begin{pmatrix} E_{11} \\ F_{11} \\ \vdots \\ E_{j\mathfrak{n}} \\ F_{j\mathfrak{n}} \end{pmatrix}$ are obtained by stacking $\mathfrak{n}$ feature matches to give an overdetermined linear system. j refers to the index of the plane patch in the image. (x, y, z) and (x', y', z') are the coordinates of the point P in two camera frames. Each point correspondence gives two independent equations as $E_i G^T = x'$ and $F_i G^T = y'$. Given that H is defined by 11 unknown entries, a set of two point correspondences allows to determine the homography up to a scale factor by solving a linear system. By estimating ΔN_i from Eq. (6) and H_1 and $\mathcal{K}$ using Eq. (9), one can construct the global homography from Eq. (4). The alignment process will be over when the average reprojection error is smaller than a threshold. Figure 2 shows the flowchart of the algorithm with required steps for alignment.

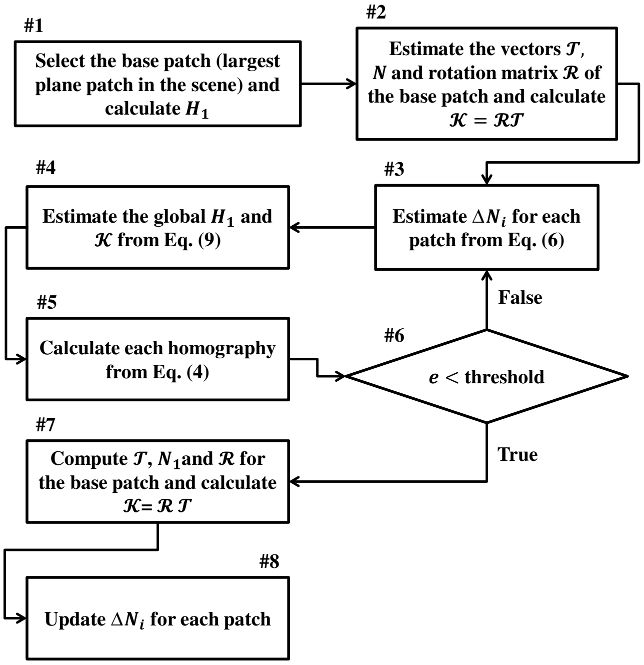

#1
Select the base patch (largest plane patch in the scene) and calculate H_1
#2
Estimate the vectors $\mathcal{T}$, N and rotation matrix $\mathcal{R}$ of the base patch and calculate $\mathcal{K} = \mathcal{R}\mathcal{T}$
#3
Estimate ΔN_i for each patch from Eq. (6)
#4
Estimate the global H_1 and $\mathcal{K}$ from Eq. (9)
#5
Calculate each homography from Eq. (4)
#6
$e <$ threshold
False
True
#7
Compute $\mathcal{T}$, $N_1$and $\mathcal{R}$ for the base patch and calculate $\mathcal{K}$= $\mathcal{R}\mathcal{T}$
#8
Update ΔN_i for each patch

Fig. 2 Flowchart of the alignment algorithm. The parameter *e* refers to the average reprojection error.

Generally, it only takes a few iterations for the algorithm to converge. The purpose of block #7 and #8 in the flowchart is to unify the final homographies precisely into a single epipolar geometry. Two sets of solutions can be obtained as Longuet–Higgins' algorithm[23] is used in block #2. In most cases, the real solution can be picked out by checking the relationship between the camera points and the surface normal vectors. The best solution can be also picked by running the alignment method using each solution and choose the one with smaller reprojection error.

3.1.1 *Alignment evaluation*

The performance of the alignment method is compared against MATLAB® R2017a "estimateGeometricTransform" function.[24–26] Eight focal stack sets from Ref. 7 are used for evaluation purposes and peak signal-to-noise ratio (PSNR) and structural similarity (SSIM)[27] metrics are employed to quantitatively evaluate the performance. Figure 3 shows the visual performance of the proposed alignment method compared to MATLAB.[24] Figure 3(b) shows the image which has to be aligned with the reference image in Fig. 3(a). The initial difference of the images and the SSIM map before alignment is shown in Figs. 3(c) and 3(d), respectively. The brighter SSIM map means a better alignment in terms of SSIM.

As it is shown in Figs. 3(g) and 3(h), the proposed alignment method has a superior performance to MATLAB's geometric transformation function.[24] The numerical evaluation of the proposed framework is shown in Fig. 4 based on

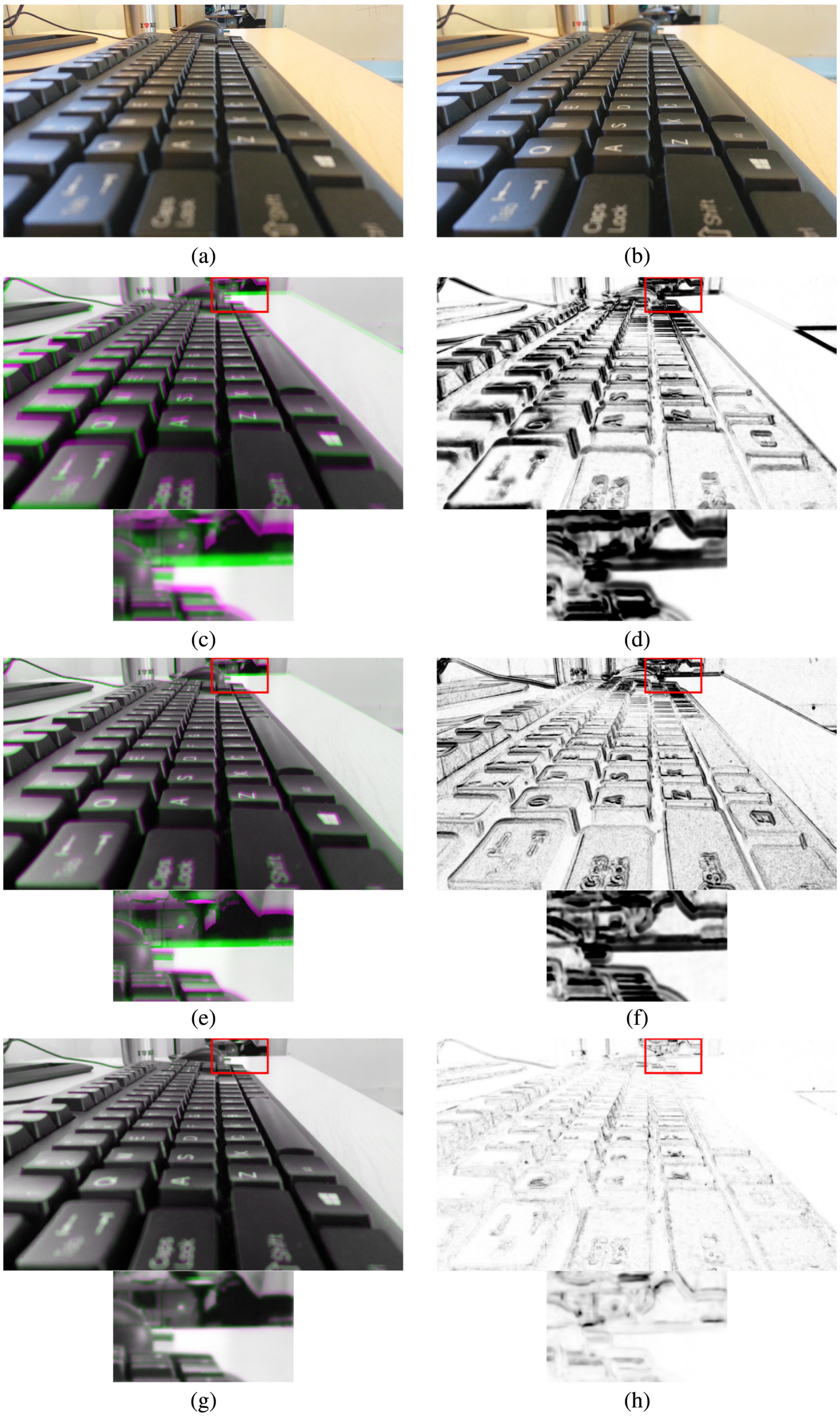
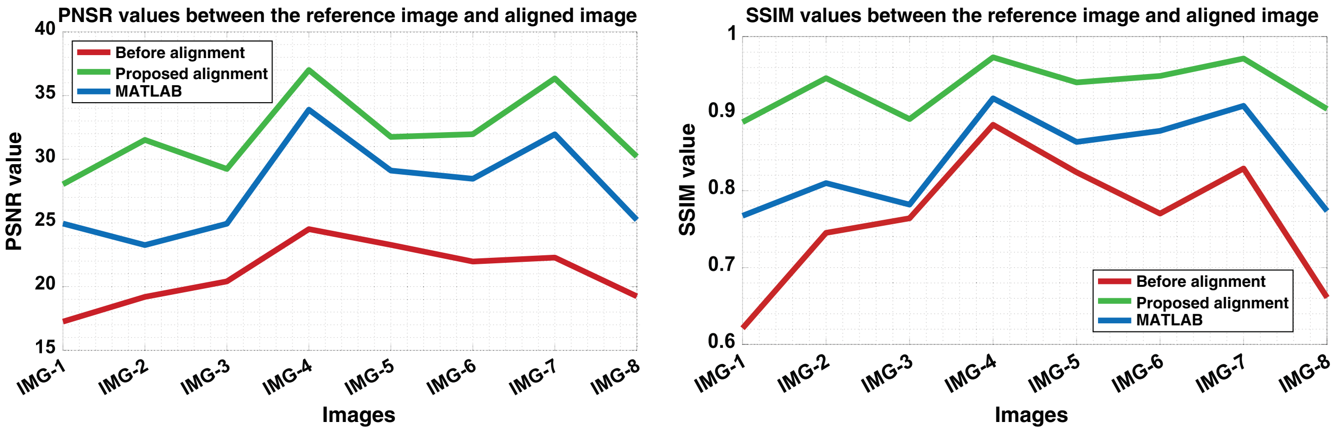


Fig. 3 Performance of the proposed alignment compared to MATLAB[24] geometric transformation function. (a) Image #1 (reference image). (b) Image #2. (c) Differences before alignment. (d) SSIM map before alignment. (e) Differences after alignment by MATLAB.[24] (f) SSIM map after alignment by MATLAB.[24] (g) Differences after alignment by proposed framework. (h) SSIM map after alignment by proposed framework.

PSNR and SSIM values of eight image sets. The values are calculated between the images before and after alignment. Clearly, the proposed framework has a superior performance to MATLAB's function[24] based on PSNR and in terms of SSIM.

3.2 *Depth Estimation and Regularization*

The depth estimation process starts with calculating the value of the focus factor for each pixel at every frame of the aligned focal stack. The value of the focus factor for a pixel (i, j) over all the frames in the stack is referred as focus function.



(a) (b)

Fig. 4 Numerical evaluation of the proposed framework using PSNR and SSIM compared to MATLAB.[24] (a) PSNR values for each method/image set. (b) SSIM values for each method/image set.

The modified Laplacian is used in this case to compute the focus function of I as

$$\mathcal{F} = (|I * C_x| + |I * C_y|) * \alpha_r, \tag{10}$$

where the convolution masks on x and y domains are $C_x = [-1,2,-1]$ and $C_y = C_x^T$, respectively. $I(i,j)$ represents the image intensity at the pixel (i,j). Equation (10) is the convolution between the mask and images in X axis and Y axis. The mean filter mask is used as α by the radius r. The initial depth map is computed by modeling the focus function using the three-point Gaussian distribution.[28] The algorithm relies on three focus factors $\mathcal{F}_{m-1}$, $\mathcal{F}_m$, and $\mathcal{F}_{m+1}$ where m denotes the index of the focus measure based on the number of the images in the stack. In theory, the algorithm requires at least three images to estimate the depth map. This will result in the following focus function:

$$\mathcal{F} = \mathcal{F}_m \exp\left\{-\frac{(M-S)^2}{2\sigma_{\mathcal{F}}^2}\right\}, \tag{11}$$

where S and $\sigma_{\mathcal{F}}$ are the mean standard deviation of the Gaussian distribution and M is the displacement of the object plane. The values $\mathcal{F}_{m-1}$ and $\mathcal{F}_{m+1}$, as well as the maximum focus factor $\mathcal{F}_m$, are used to interpolate a Gaussian function. The estimated depth values correspond to the location of $\mathcal{F}_m$ which is the maximum value of the Gaussian function. This process can be described briefly, as for every pixel, the image with the highest focus measure is identified and the depth corresponding to that pixel is estimated by interpolating a Gaussian function around its position.

As long as there is a good correlation between the Gaussian model and the focus function, the depth values get more authentic. But this situation is not constant and it can be interrupted by a variety of reasons such as noise. The presence of noise in the image domain can cause the focus function not to fit on the Gaussian model. That means the initial depth map is suffering from uncertain depth values. This condition becomes severe in case of small motions of the camera. Figures 5(b), 5(e), and 5(h) show the initial estimated depth map.

This problem is reformulated to a convex minimization problem to be solved by PADMM.[29,30] To define the formulation of the convex problem we refer to regularization method proposed by Rudin, Osher, and Fatemi (ROF),[31] which introduces a minimization problem to generate the restored image t for a noisy image I (which is an element of $L^2(\mathrm{F})$) as

$$\min_{t\in BV(\mathrm{F})}\{|t|_{BV} + \lambda\|t - I\|^2_{L^2}\}, \tag{12}$$

where $\lambda > 0$ is the regularization parameter and $\mathrm{F} \to \mathbb{R}$ is bounded open subset of $\mathbb{R}^2$ and denotes the image domains. $|t|_{\mathrm{BV}}$ is the bounded variation (BV)-seminorm defined as

$$|t|_{\mathrm{BV}} = \sup_{|g|_\infty \le 1, g\in C_c^1(\mathrm{F})^2} \int_{\mathrm{F}} t(x)\,\mathrm{div}\,g(x)\mathrm{d}x, \tag{13}$$

where $|g| = \sqrt{g_1^2 + g_2^2}$ and $C_c^1(\mathrm{F})$ presents the class of continuously differentiable functions of compact support in F.

The ROF model Eq. (12) has certain limitations such as loss of contrast, which happens due to the use of ℓ^2 fidelity and is vulnerable in presence of impulse noise.

To overcome this issue, the ROF function is changed to a unique global minimizer by employing the vectorial ℓ^1 norm fidelity term[32] as a measure of fidelity between the observed and denoised images.

$$\min_{t\in\mathrm{BV}(\mathrm{F})}\{|t|_{\mathrm{BV}} + \lambda\|t - I\|_{L^1}\}, \tag{14}$$

where I is equal to $\mathcal{F}$ in Eq. (11). The model in Eq. (14) is more effective than the ROF model in removing impulse noise. This model is contrast invariant and has a strong geometrical meaning.

Using the ℓ^1 norm also allows solving nonconvex optimization problems using convex optimization methods. The important advantage of using the convex optimization is that the global optimum is achievable with a high precision in a shorter computational time. It is also independent from the initialization.

Since the problem can be solved using convex optimization, we attempt to solve Eq. (14) by utilizing PADMM

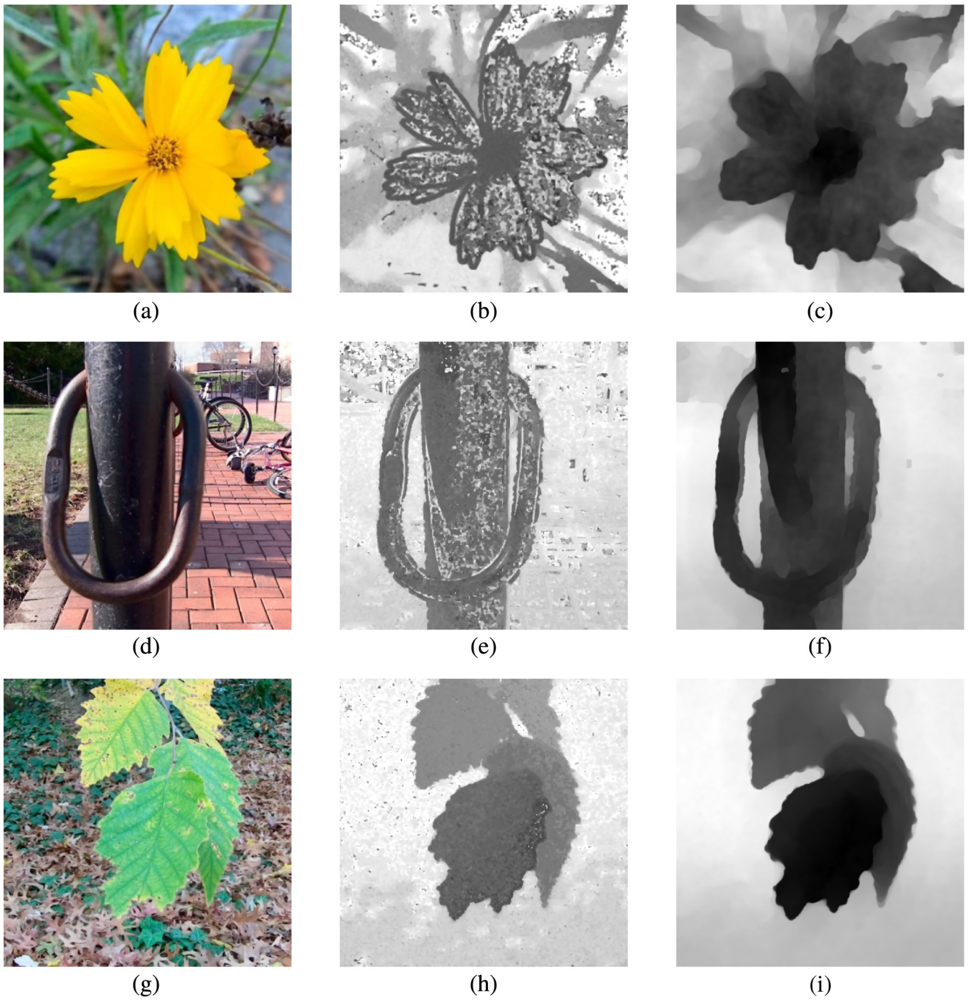


Fig. 5 The performance of the PADMM on filtering the initial depth map. (a), (d), (g) A frame from the focus stack. (b), (e), (h) Initial depth map. (c), (f), (i) Filtered depth map using PADMM.

constrained convex minimization method. Consider a generic constrained minimization problem as

$$\min_{(p,q)}\{R(p)+S(q) \text{ subject to } T(p,q)=l\}, \tag{15}$$

where R and S are proper, closed convex functions, T denotes a nonlinear operator, and l is the specified function. This constraint could be a data constraint based on the local depth confidence values or a smoothness constraint to keep the propagation of the data with high local confidence. In this application, we prefer to use the shading constraint to preserve the fine-scale shape information. To define this, we refer to the model of the Lambertian shading as a quadratical function of the surface normal[33,34]

$$\mathcal{S}(\alpha)=\alpha^T E\alpha, \tag{16}$$

where $\alpha^T=(n_x,n_y,n_z,1)$ for surface normal n and E is a symmetric 4×4 matrix that depends on the lighting environment, which is measured using a sphere placed in the scene as showed in Ref. 33. This model is solved for each pixel. Three-dimensional (3-D) coordinates of each point is calculated by reprojecting each pixel into the scene based on its image coordinates and the initial depth value d. Each pair of pixels i, j has the depth values d_i, d_j, 3-D positions v_i, v_j, and normals n_i, n_j. The vector $\overrightarrow{v_i v_j}$ has to be perpendicular to the normal direction n_i+n_j. Therefore, the shading constraint can be formulated as

$$\sum_{i,j}\left[(v_j-v_i)^T\frac{n_i+n_j}{\|n_i+n_j\|}\right]^2. \tag{17}$$

To put the problem of Eq. (14) in the form of Eq. (15), we take $R(p)=\lambda\|p-I\|_{L^1}$ where $p=t$ and $S(q)=|q|$. The auxiliary variable $q=Rp$ is discarded after optimization. Assume Rp is sparse for some sparsifying transform R. Often R is a "tall" matrix, such as finite differences along horizontal and vertical directions. The only difference of ADMM from the general linear equality-constrained problem is that the initial variable, p here, has been split into two parts, called p and q, with the objective function separable across this splitting.

In general, the advantage of the ADMM lies in the splitting scheme of two subproblems, which are relatively easy to solve.[35,36] Most of the variants of ADMM, including the classic ones, only focus on linear constraints,[37–43] in reality many practical problems require nonlinear constraints. The conditions for the convergence of the linear ADMM are presented in Ref. 44. In the present research, the splitting is performed on the nonlinear operator T in Eq. (15). The behavior of ADMM on these types of problems has been unpredictable as the convergence of the function does not hold anymore, especially when the nonlinear operator results in a nonconvex optimization function and when there are nonsmooth functions and nonconvex sets in the problems. However, it has been shown that ADMM works in some applications and in fact in practice it often exhibits great performance.[45–53] In this paper, the effect of the nonlinear constraint is eliminated due to the use of Taylor linearization as presented in Eqs. (22) and (23). This allows solving convex optimization problems with nonlinear operator constraints (by simultaneous linearization of the nonlinear operator constraint).

Equation (15) is solved by alternating minimization of the augmented Lagrange function

$$\mathcal{L}_{\daleth} = R(p) + S(q) + \langle \rho, T(p,q) - l \rangle + \frac{\daleth \| T(p,q) - l \|_2^2}{2}, \tag{18}$$

where (p, q) are the solution vectors, ρ is a sequence of estimates of the Lagrange multipliers of the constraints $T(p,q) = l$, and $\daleth > 0$ is a predefined penalty parameter. $\| \cdot \|$ and $\langle \cdot, \cdot \rangle$ denote the Euclidean norm and standard inner product, respectively.

Given the residuals as $r = T(p,q) - l$, we can express the ADMM problem as

$$p^{k+1} \in \arg\ \min_p \left\{ R(p) + \langle \rho^k, T(p,q^k) \rangle + \frac{\daleth \| T(p,q^k) - l \|_2^2}{2} \right\}, \tag{19}$$

$$q^{k+1} \in \arg\ \min_q \left\{ S(q) + \langle \rho^k, T(p^{k+1}, q) \rangle + \frac{\daleth \| T(p^{k+1}, q) - l \|_2^2}{2} \right\}, \tag{20}$$

$$\rho^{k+1} = \daleth [T(p^{k+1}, q^{k+1}) - l] + \rho^k, \tag{21}$$

where k is the iteration number. Equations (19) and (20) iteratively minimize p and q, respectively. By finding the linear approximation of $T(p^{k+1}, q^k)$ and $T(p^{k+1}, q^{k+1})$ around p^k and q^k using the Taylor linearization, we can reduce the nonlinearity computation overhead of Eqs. (19) and (20). So

$$T(p, q^k) \cong T(p^k, q^k) + \vartheta_p T(p^k, q^k)(p - p^k), \tag{22}$$

$$T(p^{k+1}, q) \cong T(p^{k+1}, q^k) + \vartheta_q T(p^{k+1}, q^k)(q - q^k). \tag{23}$$

Given the function T, $\vartheta_p T(p)$ denotes its subdifferential at p and ϑ is a subgradient operator.

To convert ADMM to a preconditioned solver, Eqs. (19) and (20) are modified by adding the following proximal terms:

$$\frac{\| p^{k+1} - p^k \|_{Z_1^k}^2}{2}, \tag{24}$$

$$\frac{\| p^{k+1} - p^k \|_{Z_1^k}^2}{2}, \tag{25}$$

$$\| \varpi \|_Z = \sqrt{\langle Z_{\varpi}, \varpi \rangle}, \tag{26}$$

where Z is the positive definite matrix. (A positive definite matrix is a symmetric matrix A for which all eigenvalues are positive.[54]) Therefore, the modified Eqs. (19) and (20) are

$$p^{k+1} \in \arg\ \min_p \left\{ \frac{\lambda \| p - p^k \|_2^2}{2} + R(p) + \langle \rho^k, \mathrm{W}_k p \rangle + \frac{\daleth \| \mathrm{W}_k p - l + \mathrm{W}_k p^k - T(p^k, q^k) \|_2^2}{2} \right\}, \tag{27}$$

$$q^{k+1} \in \arg\ \min_q \left\{ \frac{\lambda \| q - q^k \|_2^2}{2} + S(q) + \langle \rho^k, \mathrm{T}_k q \rangle + \frac{\daleth \| \mathrm{T}_k q - l + \mathrm{T}_k q^k - T(p^{k+1}, q^k) \|_2^2}{2} \right\}, \tag{28}$$

where $\mathrm{W}_k = \vartheta_p T(p^k, q^k)$, $\mathrm{T}_k = \vartheta_q T(p^{k+1}, q^k)$, and $\vartheta_p = \vartheta T / \vartheta p = dT/dp$.

The general idea underlying any preconditioning process for iterative solvers is to modify the (ill-conditioned) system in such a way that we obtain an equivalent system for which the iterative method converges faster. Such a preconditioner is necessary in order to enable practical computation at all of large-scale problems within reasonable time on any given computational platform. When Z is a diagonal matrix with positive diagonal entries, each element of the split variable may be penalized differently, which means the algorithm can take larger steps for those entries that are still far from the solution by increasing the corresponding penalty element. However, such diagonal matrices have often been used for other inverse problems because the diagonal weighting matrix can impede the use of fast computation methods. There are many viable choices for the matrix Z. The choice of Z affects only the convergence rate.

As shown in Ref. 55, one could benefit from efficient preconditioners for solving the implicit problems approximately with only one, two, or three cheap preconditioned iterations without controlling the errors, to guarantee the (weak) convergence of the ADMM iterations.

To obtain the proximity, Han et al.[56] defined the following matrices and proved that the global linear rate convergence of PADMM can be established using the positive definite matrix Z in a convex problem

$$Z_1^k = \zeta_1^k I - \daleth W_k^* \mathrm{W}_k \left(\zeta_1^k < \frac{1}{\daleth \| \mathrm{W}_k \|^2} \right), \tag{29}$$

$$Z_2^k = \zeta_2^k I - \daleth T_k^* T_k \left(\zeta_2^k < \frac{1}{\daleth \| \mathrm{T}_k \|^2} \right), \tag{30}$$

where I is a self-adjoint and positive definite operator, and p^{k+1} and q^{k+1} are updated as

$$p^{k+1} = (I + \zeta_1^k \vartheta R)^{-1} [p^k - \zeta_1^k W_k^* (2\rho^k - \rho^{k-1})], \tag{31}$$

$$q^{k+1} = (I + \zeta_2^k \vartheta S)^{-1}(q^k - \zeta_2^k T_k^* \{\rho^k + \daleth[T(p^{k+1}, q^k) - l]\}). \tag{32}$$

This approach ensures the linear convergence rate of the solver. We set the parameters ζ and $\daleth$ fixed to $1/3$, which experimentally appears to guarantee convergence. Based on Eqs. (30) and (31), the proximity is defined as[57]

$$(I + \alpha\vartheta R)^{-1}(\varpi) = \arg\,\min_p \left\{ \alpha R(p) + \frac{1}{2}\|p - \varpi\|_2^2 \right\}. \tag{33}$$

Figures 5(c), 5(f), and 5(i) shows the filtered depth map by using the PADMM.

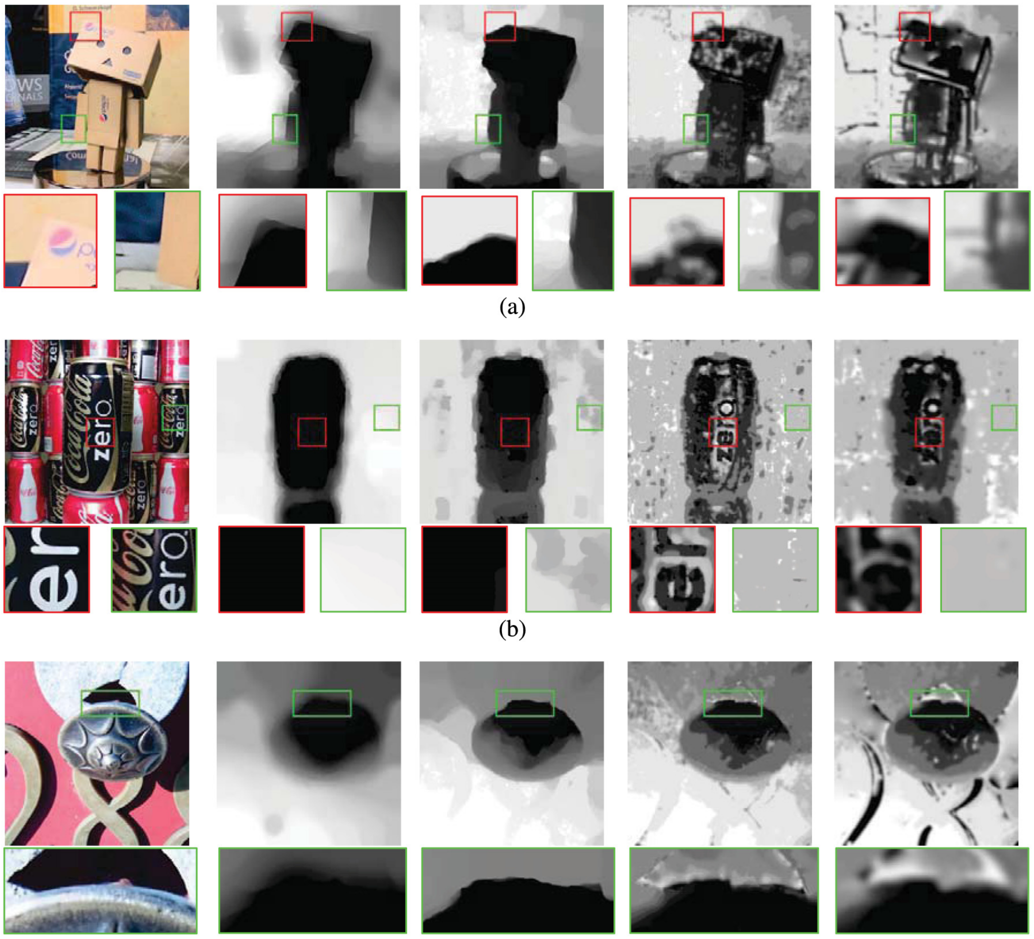
4 Experiments and Evaluation

4.1 *Qualitative Evaluation*

For qualitative evaluation purposes, 21 sets of focal stack images by Ref. 58 are used. The focal stacks are captured using a Lytro camera, which is equipped with an array of 360×360 microlenses (upsampled to 1080×1080) mounted on an 11 MP sensor.

The depth maps generated by the proposed framework are compared against the method presented by Moeller et al.,[6] Helicon Focus,[59] and Zerene Stacker.[60] Numerical comparison of these results is a challenging task as there is no ground truth and publicly available dataset, so the depth maps are compared visually. Figure 6 shows the generated depth maps by the proposed framework, Moeller et al.,[6] Helicon Focus,[59] and Zerene Stacker.[60] Figure 6(a) shows the case that the depth maps computed by Moeller et al.,[6] Helicon Focus,[59] and Zerene Stacker[60] are missing a corner of an object and some parts of the background depth information are mixed with foreground depth values. Figure 6(b) shows the scenario where the depth maps by Moeller et al.,[6] Helicon Focus,[59] and Zerene Stacker[60] are suffering from inaccurate depth values on an object's surface.

Also similar to the previous example, the background depth information is mixed with foreground depth values. Figure 6(c) shows the case where the depth maps by Moeller et al.,[6] Helicon Focus,[59] and Zerene Stacker[60] are not following the edges on the object's boundary. This might cause a problem in segmentation and synthetic defocus applications. To determine the performance of the generated


(a)

(b)

(c)

Fig. 6 The comparison of the depth maps computed by the proposed framework and Moeller et al,[6] Helicon Focus,[59] and Zerene Stacker.[60] Columns left to right: all in focus image, proposed framework, Moeller et al.,[6] Helicon Focus,[59] and Zerene Stacker.[60] (a) Test image #1. (b) Test image #2. (c) Test image #3.

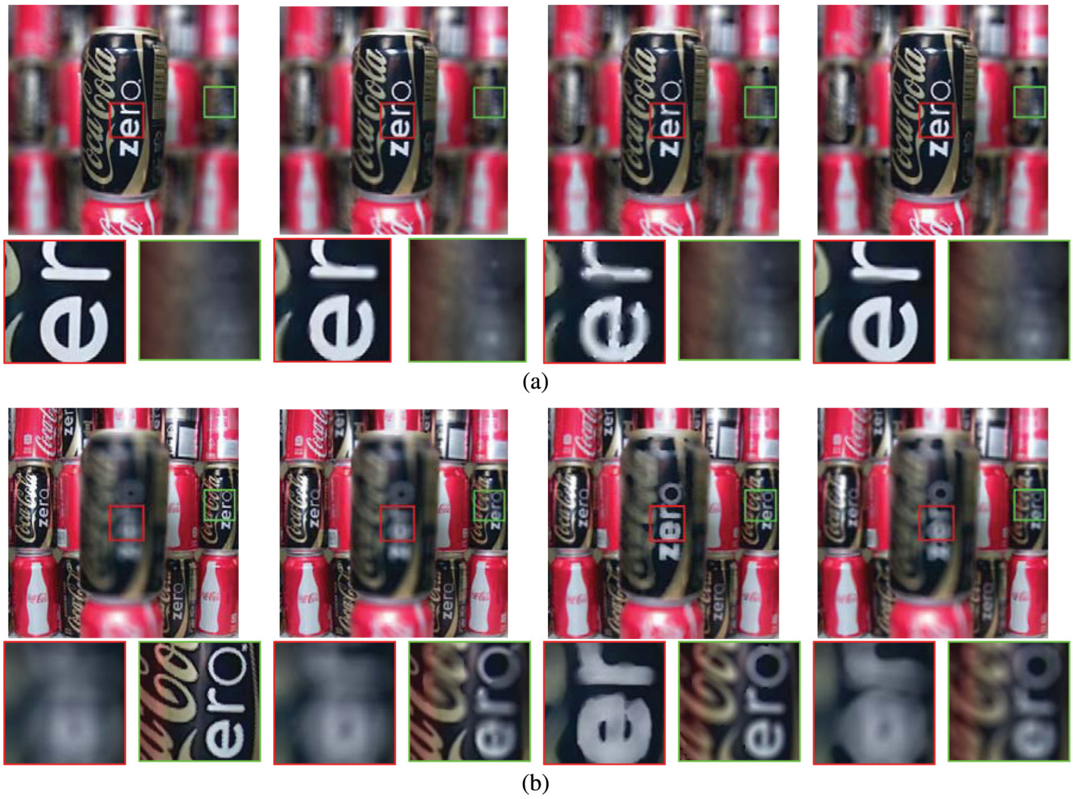


Fig. 7 Refocusing using the recovered depth map and all in focus image. (a) Left to right: proposed framework: front in focus, Moeller et al.[6]: front in focus, Helicon Focus:[59] front in focus, and Zerene Stacker:[60] front in focus. (b) Left to right: proposed framework: background in focus, Moeller et al.:[6] background in focus, Helicon Focus:[59] background in focus, and Zerene Stacker:[60] background in focus.

depth maps for synthetic defocus applications, we applied hexagon-shaped uniform distributed blur to the all in focus images based on the depth layers.

Figure 7 shows the synthetic defocus generated based on the depth maps presented in Fig. 6(b). Frontal object and the background are chosen as two focal points for each sample. As it is shown in Fig. 7, faulty depth values can cause artifacts in applications such as synthetic defocus and postcapture refocusing.

Figure 8 shows the analysis of the 3-D model generated based on the depth map from the proposed framework. Figures 8(a)–8(c) show the all in focus image, the depth map estimated from the corresponding focal stack and the 3-D color mesh generated based on the depth map, respectively. Figure 8(d) represents the rasterized color-coded 3-D model from the proposed framework. The color-coded model indicates how accurate the proposed framework is in terms of establishing depth levels. The transition from red to blue presents the areas, which are closer and far from the camera.

Figure 8(e) shows the 3-D normal of the reconstructed surface calculated using the method presented in Ref. 61. By looking at 3-D normal one can determine the smoothness of the depth values estimated by the proposed framework.

4.2 *Evaluation on the Performance of the Optimization Function*

At the second part of the experiment, the performance of the proposed PADMM is compared against five other optimization methods including fast iterative shrinkage-thresholding algorithm,[62] classical forward-backward,[63] forward-backward splitting (FBS),[64] accelerated FBS + restart,[65–67] and adaptive stepsize selection FBS.[67,68] The mean and standard deviation of the residual norm for each optimization method are shown in Fig. 9. The maximum number of iterations and the regularization parameter are set to 300 and 0.7 for all the methods, respectively. Note that all these methods are already proven to work for the same type of optimization problem as we are dealing with the convex minimization with linearized constraints. Although the application of ADMM has received a lot of attention in different fields, there is a lack of theoretical support for how to set the algorithm parameters, and its step size is typically tuned experimentally. We have found that it is not particularly difficult to choose adequate values for penalty parameters since the algorithm is not overly sensitive to such values as long as they fall into some appropriate but reasonably wide range. A few trial-and-error attempts are usually needed to find good penalty parameter values, judged by observed convergence speed. There have been some studies where different formulations were proposed to set the step size in ADMM, but they are all focused on linear constraints[69] or quadratic problems.[70,71]

As shown in Fig. 9, the presented PADMM optimization method results in lower convergence error in comparison to other methods.

The numerical information related to the convergence of the PADMM and Moeller et al.[6] is presented as the decay of energy in a logarithmic form in Figs. 10(a) and 10(b),

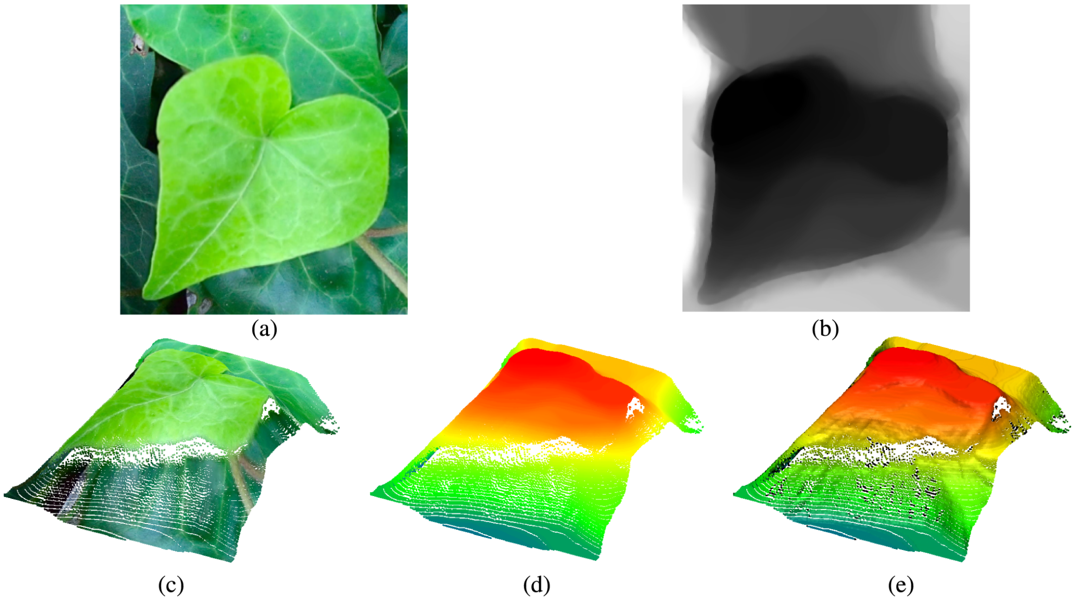


Fig. 8 3-D visualizations of the depth map estimated by the proposed framework. (a) All in focus frame of a focal stack containing 12 images. (b) Estimated depth map. Dark pixels represent closer regions to the camera. (c) Proposed framework, 3-D color mesh. (d) Proposed framework, rasterized 3-D color-coded depth. (e) Proposed framework, 3-D normals.

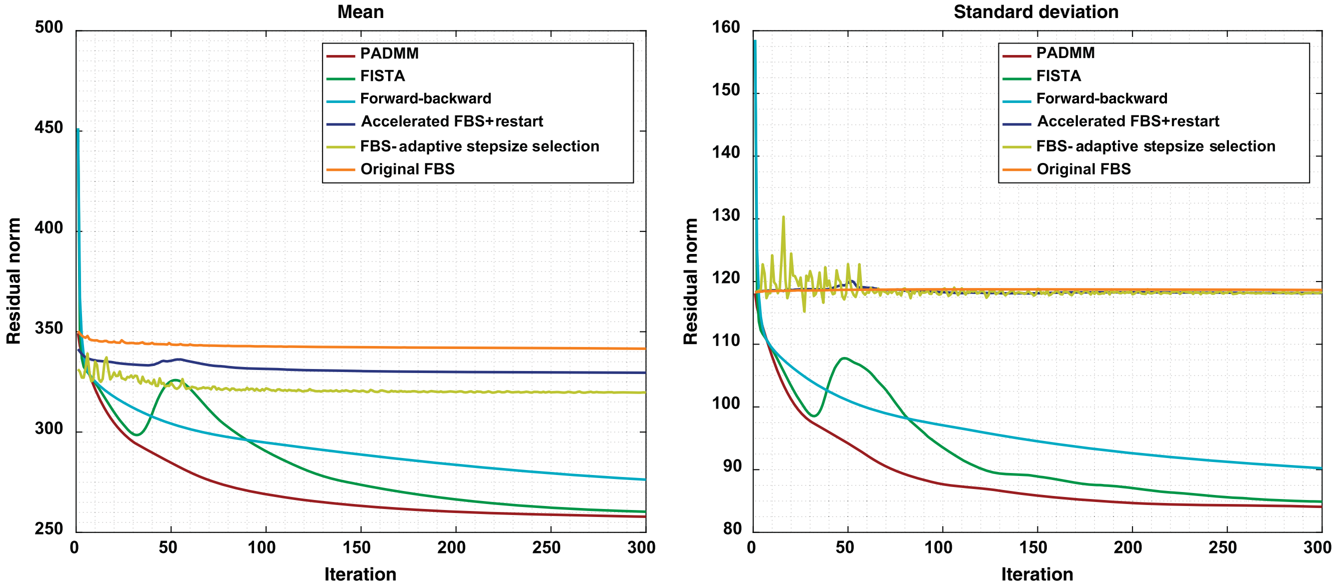
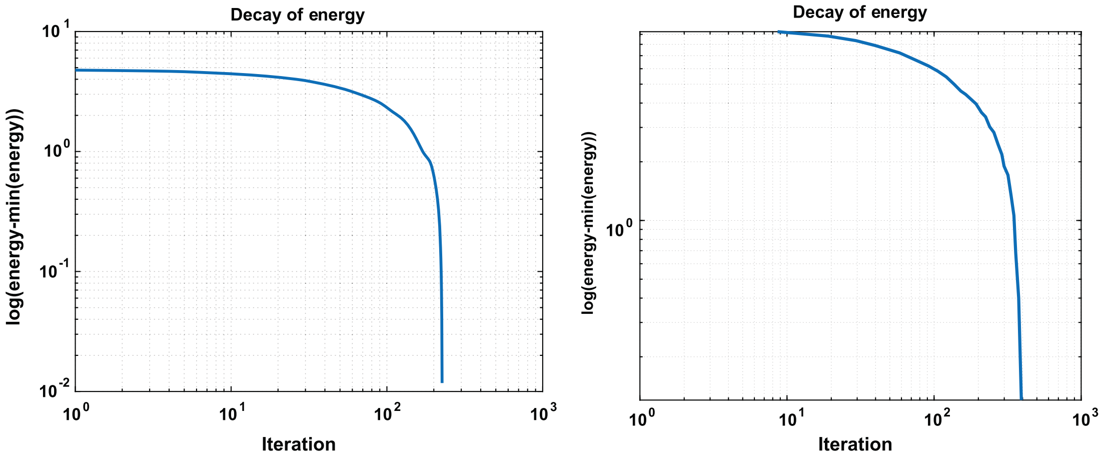


Fig. 9 Mean and standard deviation of the residual norm for PADMM and other five optimization methods.

respectively. As it is shown in Fig. 10(a), the convergence of PADMM happens around the iteration 226 and it reaches 0.01 as the decay of energy, whereas the function presented by Moeller et al.[6] around the same iteration reaches to the decay of 3.6, and it is still not converged. The better value of the decay of energy within the low number of iterations shows the superior performance of the proposed PADMM.

4.3 *Comparison with Suwajanakorn et al.*

The third part of the evaluation is done against the method proposed by Suwajanakorn et al.[7] The reason that we performed a separate comparison against this method is not having access to the code of the algorithm. The authors of Ref. 7 kindly provided the focal stacks and the depth results published in their paper. Figure 11 shows the comparison of the depth maps computed by the proposed framework and Suwajanakorn et al.[7] Figure 11(a) represents the case where the depth map computed by Suwajanakorn et al.[7] is suffering from inaccurate depth values on a reflective surface and some other objects' surface while the depth map by the proposed framework covered these issues. The depth map by Suwajanakorn et al.[7] in Fig. 11(b) shows a similar issue to the previous example, uncertain depth values along



(a)

(b)

Fig. 10 Numerical comparison: convergence of PADMM against Moeller et al.[6] (a) Convergence of PADMM as the decay of energy. (b) Convergence of Moeller et al.[6] as decay of energy.

an object's edges and surface. Figure 11(c) shows the similar issues of the reflective surfaces and inaccurate edges which have been solved by the proposed framework. However, the blue highlighted part in Fig. 11(c) shows the case where the proposed framework computed a patch of uncertain depth values on the background level.

It is also worth pointing out the advantage of the method by Suwajanakorn et al.[7] in computing longer depth range than the proposed framework.

4.4 *Quantitative Evaluation*

To numerically evaluate the performance of the proposed framework, it is required to have focal stack sets with their corresponding depth maps. As there is no available dataset that provides the ground truth, we refer to the light field dataset provided by Rerabek and Ebrahimi.[72] This dataset provides images in light field raw format with their corresponding depth maps, which are captured using Lytro Illum camera.[73] The images are captured in a variety of categories and settings such as close range scenes, lighting conditions, mirrors, transparency, etc. We randomly selected 10 light field image sets from this dataset for the evaluation purposes. Table 1 presents the optical properties at which each light field set was captured. These details are extracted using Lytro Desktop Software.[74]

The advantage of using light field images is that the post-capture refocusing can be done digitally.[75] Furthermore, 4-D light field sets can be transformed into focal stacks as proposed in Ref. 5. In this research, we used Lytro Desktop Software[74] animation module to generate focal stacks. Using this module, one can generate a sequence of motion from the light field set by decreasing the depth of field to isolate the objects in focus. As the result, we generated 61 frames per light field set with different focal points. In other words, the light field image sets are converted to focal stacks with 61 frames with 1080×720 pixels resolution. All the images are processed in their original resolution without any downsampling. The evaluation of the proposed framework is done based on four metrics including PSNR, root mean square error (RMSE), SSIM,[27] and universal quality index (UQI).[76] Similar to Sec. 4.1, the depth maps generated by the proposed framework are compared against the method presented by Moeller et al.,[6] Helicon Focus,[59] and Zerene Stacker.[60] However, in this part, we assume the depth map captured by the Lytro camera to be the ground truth.

Figure 12 shows the performance of the proposed framework compared to Moeller et al.,[6] Helicon Focus,[59] and Zerene Stacker[60] for 10 focal stack sets. As shown in Fig. 12, the proposed framework has a superior performance to the state of the art in all metrics. The SSIM plot in Fig. 12(c) indicates the SSIM of the computed depth map to the ground truth, and the general quality of the depth maps in comparison to the ground truth is shown as UQI plot in Fig. 12(d).

Figures 13–16 show the sample visual results of the proposed framework, depth map captured using Lytro camera, and the studied methods. These figures show how the proposed framework is capable of generating a depth map with high structural accuracy in highly detailed scenes, images captured under different lighting condition and in presence of the transparent and reflective surface. By looking at these figures, one might argue that the depth map generated by Moeller et al.[6] looks more pleasant. However, what we are concerned about is the smoothness of the depth map and respecting the structural geometry of the scene with the minimum artifacts. By having these two features, the post-capturing functions such as Bokeh can be applied with a much higher quality. As clearly shown in Figs. 6 and 13–16, the depth maps estimate by Moeller et al.[6] suffer from artifacts, broken edges and boundaries, missing corners, and mixed depth planes at some parts. Due to all these problems, we cannot consider Moeller et al.[6] depth map as a pleasant one, and it is certainly not suitable for a consumer application.

The analysis of source of the error influencing the other methods is not possible as the information about the methods used in Helicon Focus[59] and Zerene Stacker[60] softwares is not available. Moeller et al.[6] utilized ADMM for the refinement purposes; as it is already mentioned in the paper, there is a lack of theoretical support for how to set the ADMM parameters, and its step size is typically tuned experimentally. It is difficult to particularly mention the source of the error; however, one potential source of error could be the weak performance of the initial alignment, which applies

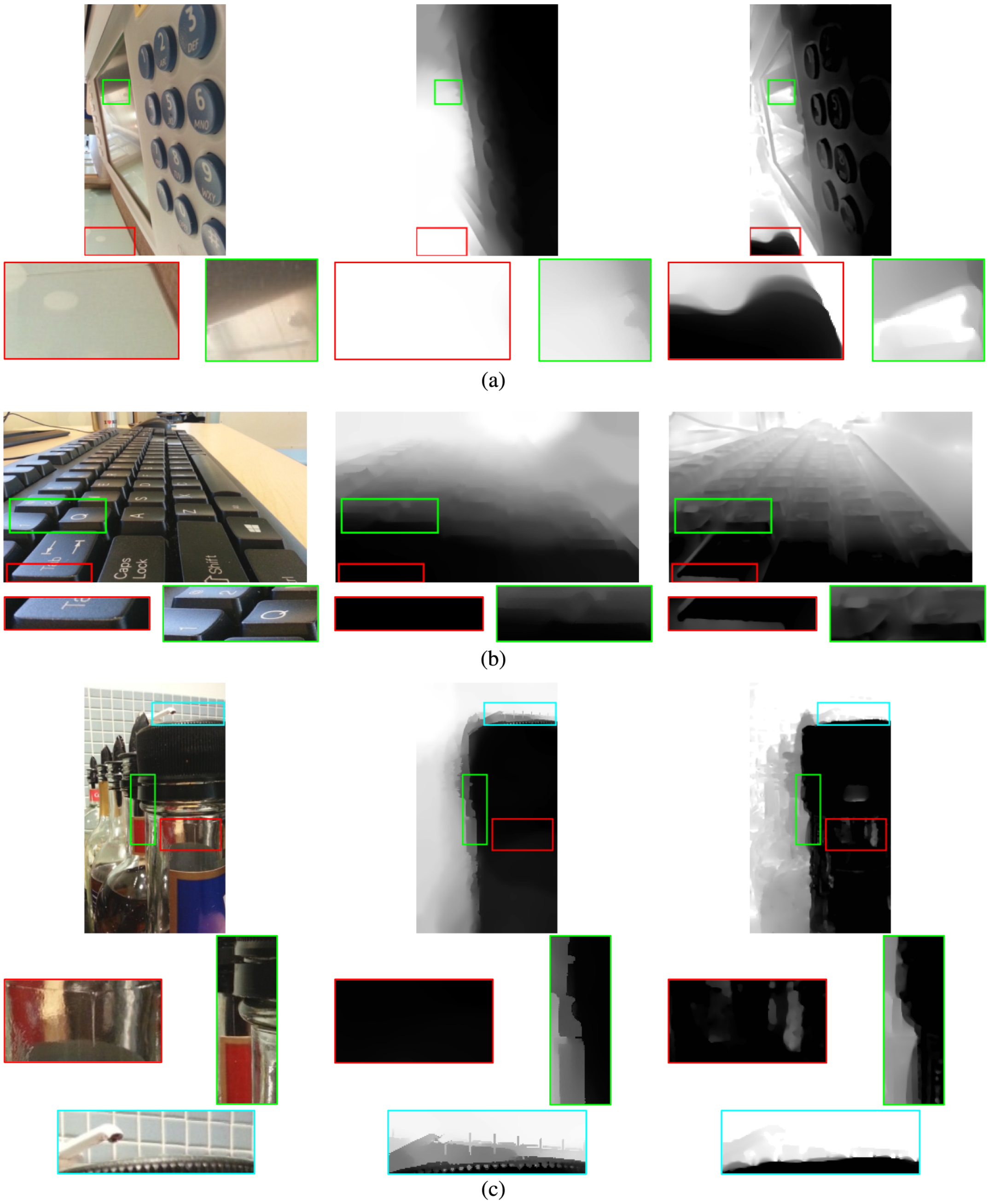


Fig. 11 The comparison of the depth maps computed by the proposed framework and Suwajanakorn et al.[7] Columns left to right: all in focus image, proposed framework, and Suwajanakorn et al.[7] (a) Test image #1. (b) Test image #2. (c) Test image #3.

Table 1 Optical properties of the light field image sets used for evaluation purposes.

	LF #1	LF #2	LF #3	LF #4	LF #5	LF #6	LF #7	LF #8	LF #9	LF #10
Shutter	1/640	1/2000	1/250	1/320	1/1000	1/2500	1/2000	1/800	1/1000	1/1600
ISO	80	80	80	80	80	80	80	80	80	80
F-stop	f/2	f/2	f/2	f/2	f/2	f/2	f/2	f/2	f/2	f/2
Focal length (mm)	66	89	34	51	83	40	40	82	47	60

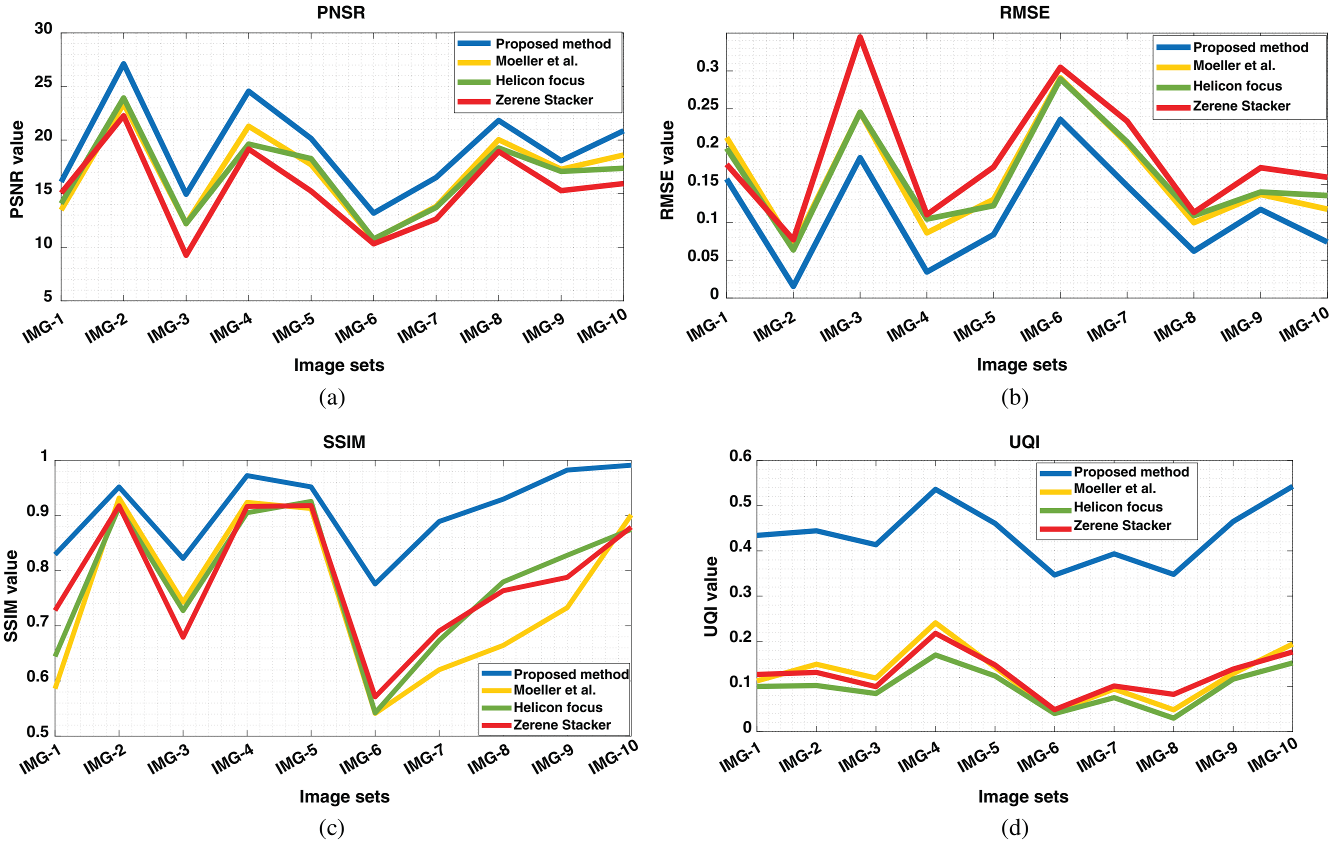


Fig. 12 Performance of the proposed framework based on PSNR, RMSE, SSIM, and UQI. (a) PSNR values for each method/image set. (b) RMSE values for each method/image set. (c) SSIM values for each method/image set. (d) UQI values for each method/image set.

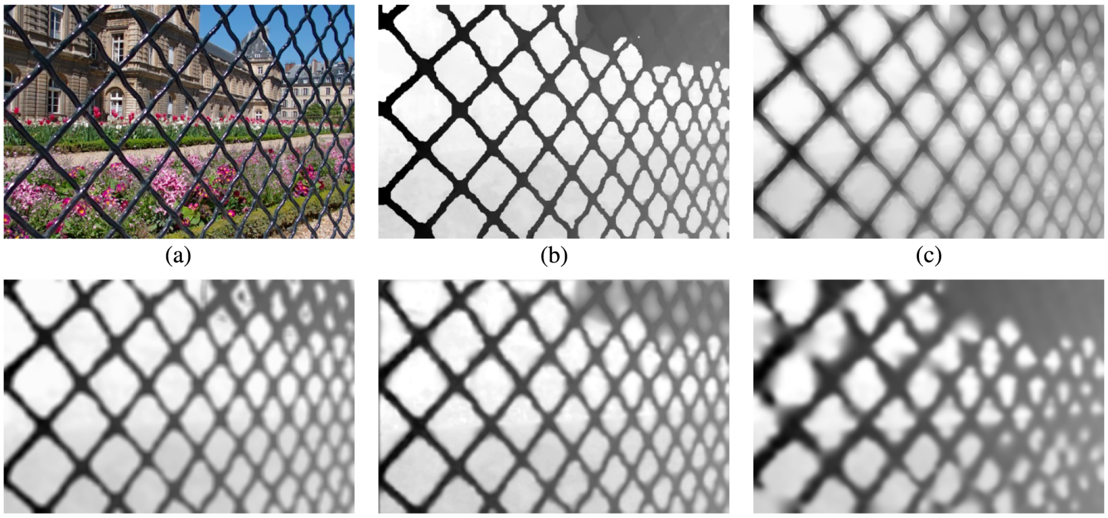


Fig. 13 Sample depth estimation: highly detailed scene. (a) All in focus image. (b) Ground truth (Lytro camera). (c) Proposed framework. (d) Moeller et al.[6] (e) Helicon Focus.[59] (f) Zerene Stacker.[60]

to the images in the stack. The presence of vectorial ℓ^1 norm fidelity term also helps the proposed framework to be more effective than others in removing impulse noises.

Table 2 shows the comparison of the proposed framework's average computational time with the other methods. All the methods are tested on the sequences of 61 images with 1080 × 720 pixels resolution. In general, the higher number of the images provides a better result. Calculating the focus function and the initial depth map is not a very time consuming process. For example, Fig. 6(b) has only five frames (1080 × 1080 resolution) in the stack. The time to calculate the focus function and the initial depth map for

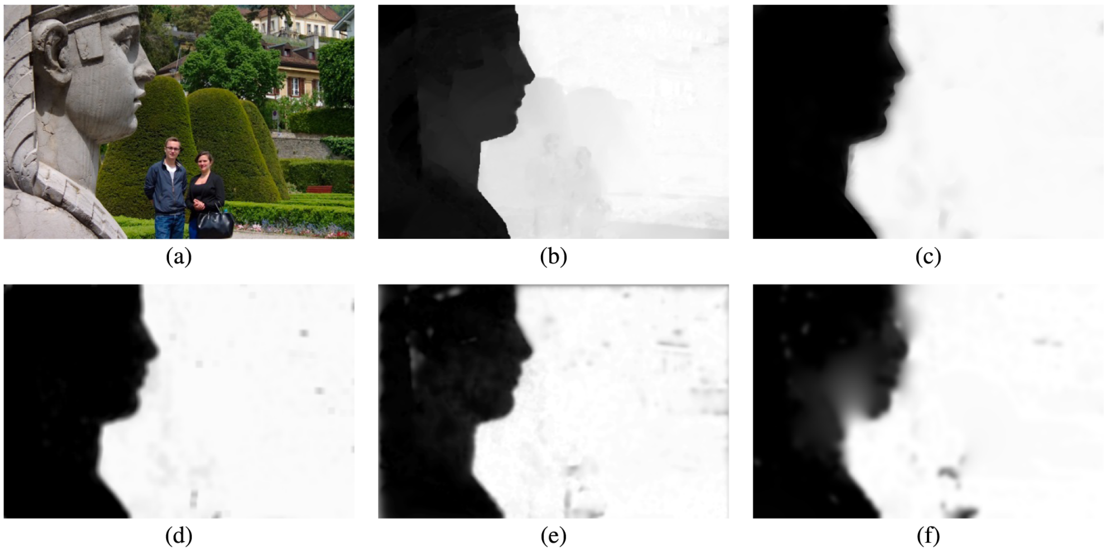


Fig. 14 Sample depth estimation: bright lighting condition. (a) All in focus image. (b) Ground truth (Lytro camera). (c) Proposed framework. (d) Moeller et al.[6] (e) Helicon Focus.[59] (f) Zerene Stacker.[60]

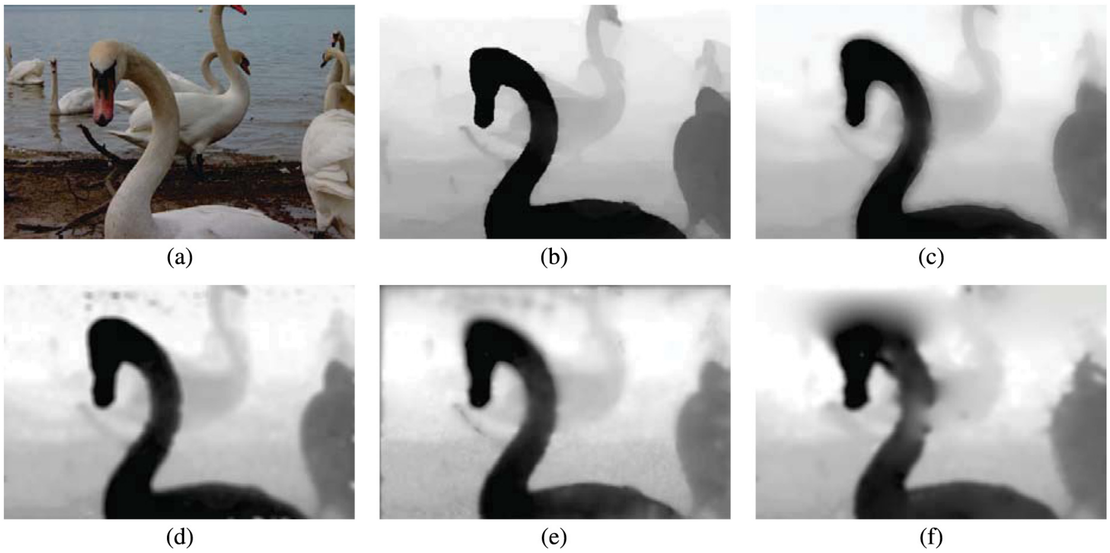


Fig. 15 Sample depth estimation: Dark lighting condition. (a) All in focus image. (b) Ground truth (Lytro camera). (c) Proposed framework. (d) Moeller et al.[6] (e) Helicon Focus.[59] (f) Zerene Stacker.[60]

this stack is ∼1.8 s. For a stack with 61 frames (1080 × 720 resolution), this time increases to ∼7 s. The expensive part is the PADMM optimization, which is independent from the number of the images in the focal stack. PADMM only deals with the initial depth map.

In a camera that captures sequences at 30 fps, it takes 2 s to capture 60 frames. In real world mobile application, it requires more than 2 s to utilize the focal sweep feature of the camera. By employing the evolving GPU and parallelism technology, the proposed framework can process the focal stacks in a considerably faster time, which makes it a potential method for consumer devices.

5 Conclusion

In this paper, a PADMM optimization method is employed to perform on depth from the focal stack and synthetic defocus application. The proposed framework is tested on a sequence of images captured by a camera with hypothetical focus and aperture values to generate the depth map. The proposed technique satisfies the constraint of the state-of-the-art method such as uncertain depth values on objects' surface, mixed depth values on different layers of background and foreground, missed depth information on an object's boundaries, which cause faulty edges, and corners in the depth map.

The proposed framework is evaluated in three parts. First, the generated depth maps with the corresponding defocused images are qualitatively compared against a recent studied method and two commercial softwares. Twenty-one sets of focal stack images are used in this comparison and all the parameters are set equally in both methods.

The second part of the evaluation is done to determine the performance of the proposed optimization technique in

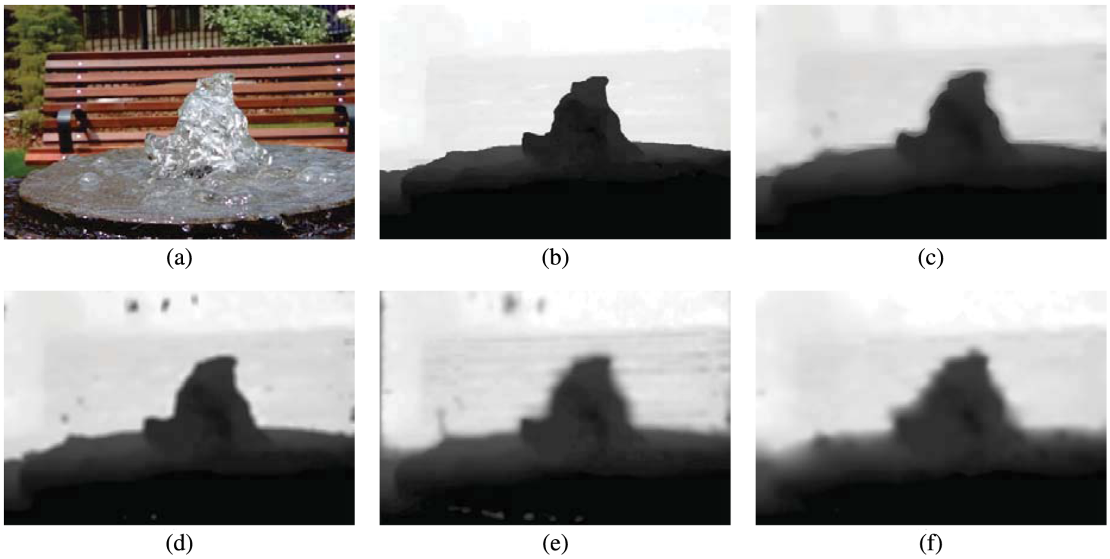


Fig. 16 Sample depth estimation: transparent and reflective surface. (a) All in focus image. (b) Ground truth (Lytro camera). (c) Proposed framework. (d) Moeller et al.[6] (e) Helicon Focus.[59] (f) Zerene Stacker.[60]

Table 2 Average computational time of the proposed framework and the state-of-the-art in seconds.

	Helicon focus[59]	Moeller et al.[6]	Proposed framework	Zerene stacker[60]
Time (s)	5.3	12.8	28.2	47.3
Programming platform	C/C++	Cuda/GPU	Matlab	Java

comparison to five other algorithms. In the third part, a light field dataset is used to generate focal stacks and then the results of the proposed framework are compared against the depth maps from the Lytro camera.

The results of the evaluation show that the proposed framework and PADMM has a superior performance to the studied depth from the focal stack and optimization methods.

The high structural accuracy of the depth map generated by the proposed framework gives the smartphone users the ability to refocus postcapture images accurately without the need to change the aperture size. The method has been implemented in MATLAB® R2017a on a device equipped with Intel i7-5600U at 2.60 GHz CPU and 16 GB RAM. The computational time of the whole framework from initializing the focal stack to final refined depth map takes ∼28.2 s on a focal stack with 61 images with 1080×720 pixels resolution. In our future work, we plan to implement the proposed algorithm as a smartphone application. However, despite the performance and accuracy of the studied method, there is still the computational time of this technique that has to be considered as the trade-off.

Acknowledgments

The research work presented here was funded under the Strategic Partnership Program of Science Foundation Ireland (SFI) and cofunded by SFI and FotoNation Ltd. Project ID: 13/SPP/I2868 on “Next Generation Imaging for Smartphone and Embedded Platforms.”

References

1. Y. Xiong and S. A. Shafer, “Depth from focusing and defocusing,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (1993).
2. M. Subbarao and G. Surya, “Depth from defocus: a spatial domain approach,” *Int. J. Comput. Vision* **13**(3), 271–294 (1994).
3. J. Ens and P. Lawrence, “An investigation of methods for determining depth from focus,” *IEEE Trans. Pattern Anal. Mach. Intell.* **15**(2), 97–108 (1993).
4. P. Grossmann, “Depth from focus,” *Pattern Recognit. Lett.* **5**(1), 63–69 (1987).
5. H. Lin et al., “Depth recovery from light field using focal stack symmetry,” in *IEEE Int. Conf. on Computer Vision (ICCV)* (2015).
6. M. Moeller et al., “Variational depth from focus reconstruction,” *IEEE Trans. Image Process.* **24**(12), 5369–5378 (2015).
7. S. Suwajanakorn, C. Hernandez, and S. M. Seitz, “Depth from focus with your mobile phone,” in *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)* (2015).
8. M. W. Tao et al., “Depth from combining defocus and correspondence using light-field cameras,” in *IEEE Int. Conf. on Computer Vision* (2013).
9. T. Broad and M. Grierson, “Light field completion using focal stack propagation,” in *ACM SIGGRAPH Posters*, Anaheim, California (2016).
10. F. P. Nava, J. G. Marichal-Hernández, and J. M. Rodríguez-Ramos, “The discrete focal stack transform,” in *16th European Signal Processing Conf.* (2008).
11. X. Lin et al., “Coded focal stack photography,” in *IEEE Int. Conf. on Computational Photography (ICCP)* (2013).
12. A. Levin et al., “Image and depth from a conventional camera with a coded aperture,” *ACM Trans. Graphics* **26**(3), 70 (2007).
13. C. Zhou, S. Lin, and S. K. Nayar, “Coded aperture pairs for depth from defocus and defocus deblurring,” *Int. J. Comput. Vision* **93**(1), 53–72 (2011).
14. N. Persch et al., “Physically inspired depth-from-defocus,” *Image Vision Comput.* **57**(Suppl. C), 114–129 (2017).
15. F. Pérez et al., “A fast and memory-efficient discrete focal stack transform for plenoptic sensors,” *Digital Signal Process.* **38**(Suppl. C), 95–105 (2015).
16. F. Pérez et al., “Lightfield recovery from its focal stack,” *J. Math. Imaging Vision* **56**(3), 573–590 (2016).
17. A. Mousnier, E. Vural, and C. Guillemot, “Partial light field tomographic reconstruction from a fixed-camera focal stack,” arXiv:150301903 (2015).
18. S. W. Bailey et al., “Fast depth from defocus from focal stacks,” *Visual Comput.* **31**(12), 1697–1708 (2015).
19. K. Jeong et al., “Digital shallow depth-of-field adapter for photographs,” *Visual Comput.* **24**(4), 281–294 (2008).
20. J. Surh et al., “Noise robust depth from focus using a ring difference filter,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (2017).
21. J. R. Alonso, “Fourier domain post-acquisition aperture reshaping from a multi-focus stack,” *Appl. Opt.* **56**(9), D60–D65 (2017).

22. J. R. Alonso, A. Fernández, and J. A. Ferrari, "Reconstruction of perspective shifts and refocusing of a three-dimensional scene from a multi-focus image stack," *Appl. Opt.* **55**(9), 2380–2386 (2016).
23. H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," in *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*, A. F. Martin and F. Oscar, Eds., pp. 61–62, Morgan Kaufmann Publishers Inc., San Francisco, California (1987).
24. MATLAB, "Estimate geometric transform," https://uk.mathworks.com/help/vision/ref/estimategeometrictransform.html (24 February 2018).
25. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge (2003).
26. P. H. Torr and A. Zisserman, "MLESAC: a new robust estimator with application to estimating image geometry," *Comput. Vision Image Understanding* **78**(1), 138–156 (2000).
27. W. Zhou et al., "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.* **13**(4), 600–612 (2004).
28. S. K. Nayar and Y. Nakagawa, "Shape from focus," *IEEE Trans. Pattern Anal. Mach. Intell.* **16**(8), 824–831 (1994).
29. A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *J. Math. Imaging Vision* **40**(1), 120–145 (2011).
30. M. Benning et al., "Preconditioned ADMM with nonlinear operator constraint," in *System Modeling and Optimization*, L. Bociu et al., Eds., 117–126, Springer International Publishing, Cham (2016).
31. L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D* **60**(1), 259–268 (1992).
32. Y. Meyer, *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations: The Fifteenth Dean Jacqueline B. Lewis Memorial Lectures*, American Mathematical Society, Boston, Massachusetts (2001).
33. M. K. Johnson and E. H. Adelson, "Shape estimation in natural illumination," in *IEEE Conf. on Computer Vision and Pattern Recognition* (2011).
34. R. Ramamoorthi and P. Hanrahan, "An efficient representation for irradiance environment maps," in *Proc. of the 28th Annual Conf. on Computer Graphics and Interactive Techniques* (2001).
35. P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, H. H. Bauschke et al., Eds., pp. 185–212, Springer, New York (2011).
36. A. Repetti, E. Chouzenoux, and J.-C. Pesquet, "Proximal primal-dual optimization methods," in *Proc. of Int. Biomedical and Astronomical Signal Processing (BASP) Frontiers Workshop* (2015).
37. M. A. T. Figueiredo and J. M. Bioucas-Dias, "Deconvolution of Poissonian images using variable splitting and augmented Lagrangian optimization," in *IEEE/SP 15th Workshop on Statistical Signal Processing* (2009).
38. M. A. T. Figueiredo and J. M. Bioucas-Dias, "Restoration of Poissonian images using alternating direction optimization," *IEEE Trans. Image Process.* **19**(12), 3133–3145 (2010).
39. J.-F. Giovannelli and A. Coulais, "Positive deconvolution for superimposed extended source and point sources," *Astron. Astrophys.* **439**(1), 401–412 (2005).
40. T. Goldstein and S. Osher, "The split Bregman method for L1-regularized problems," *SIAM J. Imaging Sci.* **2**(2), 323–343 (2009).
41. Q. Tran-Dinh and V. Cevher, "A primal-dual algorithmic framework for constrained convex minimization," arXiv:14065403 (2014).
42. X. Wang et al., "Robust subspace discovery via relaxed rank minimization," *Neural Comput.* **26**(3), 611–635 (2014).
43. L. Zhao et al., "Multi-task learning for spatio-temporal event forecasting," in *Proc. of the 21th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining* (2015).
44. N. Komodakis and J.-C. Pesquet, "Playing with duality: an overview of recent primal? Dual approaches for solving large-scale optimization problems," *IEEE Signal Process. Mag.* **32**(6), 31–54 (2015).
45. Y. Shen, Z. Wen, and Y. Zhang, "Augmented Lagrangian alternating direction method for matrix separation based on low-rank factorization," *Optim. Methods Software* **29**(2), 239–263 (2014).
46. D. L. Sun and C. Févotte, "Alternating direction method of multipliers for non-negative matrix factorization with the beta-divergence," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)* (2014).
47. Y. Xu et al., "An alternating direction algorithm for matrix completion with nonnegative factors," *Front. Math. China* **7**(2), 365–384 (2012).
48. L. Yang, T. Pong, and X. Chen, "Alternating direction method of multipliers for a class of nonconvex and nonsmooth problems with applications to background/foreground extraction," *SIAM J. Imaging Sci.* **10**(1), 74–110 (2017)..
49. A. P. Liavas and N. D. Sidiropoulos, "Parallel algorithms for constrained tensor factorization via alternating direction method of multipliers," *IEEE Trans. Signal Process.* **63**(20), 5450–5463 (2015).
50. R. Lai and S. Osher, "A splitting method for orthogonality constrained problems," *J. Sci. Comput.* **58**(2), 431–449 (2014).
51. O. Miksik et al., "Distributed non-convex ADMM-inference in large-scale random fields," in *British Machine Vision Conf. (BMVC)* (2014).
52. S. Bouaziz, A. Tagliasacchi, and M. Pauly, "Sparse iterative closest point," in *Proc. of the Eleventh Eurographics/ACMSIGGRAPH Symp. on Geometry Processing*, Genova, Italy (2013).
53. S. You and Q. Peng, "A non-convex alternating direction method of multipliers heuristic for optimal power flow," in *IEEE Int. Conf. on Smart Grid Communications (SmartGridComm)* (2014).
54. E. W. Weisstein, "Positive definite matrix," *From MathWorld—A Wolfram Web Resource*, http://mathworld.wolfram.com/PositiveDefiniteMatrix.html (2003).
55. K. Bredies and H. Sun, "A proximal point analysis of the preconditioned alternating direction method of multipliers," *J. Optim. Theory Appl.* **173**(3), 878–907 (2017).
56. D. Han, D. Sun, and L. Zhang, "Linear rate convergence of the alternating direction method of multipliers for convex composite quadratic and semi-definite programming," arXiv:150802134 (2015).
57. H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Vol. **2011**, Springer, New York (2017).
58. N. Li et al., "Saliency detection on light field," *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(8), 1605–1616 (2016).
59. "Helicon focus," http://www.heliconsoft.com/heliconsoft-products/helicon-focus/ (6 March 2018).
60. "Zerene stacker," http://zerenesystems.com/ (6 March 2018).
61. R. B. Rusu, "Semantic 3D object maps for everyday manipulation in human living environments," PhD Thesis, Computer Science Department, Technische Universität München, Germany (2009).
62. A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.* **2**(1), 183–202 (2009).
63. F. Jelinek, "Continuous speech recognition by statistical methods," *Proc. IEEE* **64**(4), 532–556 (1976).
64. P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Model. Simul.* **4**(4), 1168–1200 (2005).
65. H. Raguet, J. Fadili, and G. Peyré, "A generalized forward-backward splitting," *SIAM J. Imaging Sci.* **6**(3), 1199–1226 (2013).
66. S. Villa et al., "Accelerated and inexact forward-backward algorithms," *SIAM J. Optim.* **23**(3), 1607–1633 (2013).
67. T. Goldstein, C. Studer, and R. Baraniuk, "A field guide to forward-backward splitting with a FASTA implementation," arXiv:14113406 (2014).
68. P. Tseng, "A modified forward-backward splitting method for maximal monotone mappings," *SIAM J. Control Optim.* **38**(2), 431–446 (2000).
69. E. Ghadimi et al., "On the optimal step-size selection for the alternating direction method of multipliers," *IFAC Proc. Vol.* **45**(26), 139–144 (2012).
70. E. Ghadimi et al., "Optimal parameter selection for the alternating direction method of multipliers (ADMM): quadratic problems," *IEEE Trans. Autom. Control* **60**(3), 644–658 (2015).
71. A. U. Raghunathan and S. Di Cairano, "Optimal step-size selection in alternating direction method of multipliers for convex quadratic programs and model predictive control," in *Proc. of Symp. on Mathematical Theory of Networks and Systems* (2014).
72. M. Rerabek and T. Ebrahimi, "New light field image dataset," in *8th Int. Conf. on Quality of Multimedia Experience (QoMEX)* (2016).
73. Lytro, "ILLUM—user manual," (2016), https://s3.amazonaws.com/lytro-corp-assets/manuals/english/illum_user_manual.pdf (5 August 2015).
74. Lytro, "Lytro software," (2017), https://support.lytro.com/hc/en-us/categories/202651357-Software (1 January 2018).
75. C. Hazirbas, L. Leal-Taixé, and D. Cremers, "Deep depth from focus," arXiv:170401085 (2017).
76. W. Zhou and A. C. Bovik, "A universal image quality index," *IEEE Signal Process. Lett.* **9**(3), 81–84 (2002).

Hossein Javidnia received his master's degree in information technology engineering from the University of Guilan, Iran, in 2014. He has started his PhD since 2015 in electrical engineering at the National University of Ireland, Galway. His current research interests include image processing, machine vision and automotive navigation.

Peter Corcoran (F'10) is a fellow of IEEE, editor-in-chief of *IEEE Consumer Electronics Magazine*, and a professor with a personal chair at the College of Engineering and Informatics at NUI Galway. His research interests include biometrics, cryptography, computational imaging, and consumer electronics. He is coauthor on 300+ technical publications and coinventor on more than 250 granted US patents. In addition to his academic career, he is also an occasional entrepreneur, industry consultant, and compulsive inventor.

Appendix E: Total Variation-Based Dense Depth from Multi-Camera Array

Total Variation-Based Dense Depth from Multi-Camera Array

Hossein Javidnia,[a,*] **Peter Corcoran**[a]
[a]Ireland, Galway, University Road, National University of Ireland Galway, College of Engineering, Department of Electronic Engineering

Abstract. Multi-Camera arrays are increasingly employed in both consumer and industrial applications, and various passive techniques are documented to estimate depth from such camera arrays. Current depth estimation methods provide useful estimations of depth in an imaged scene but are often impractical due to significant computational requirements. This paper presents a novel framework that generates a high-quality continuous depth map from multi-camera array/light field cameras. The proposed framework utilizes analysis of the local Epipolar Plane Image (EPI) to initiate the depth estimation process. The estimated depth map is then refined using Total Variation (TV) minimization based on the Fenchel-Rockafellar duality. Evaluation of this method based on a well-known benchmark indicates that the proposed framework performs well in terms of accuracy when compared to the top-ranked depth estimation methods and a baseline algorithm. The test dataset includes both photorealistic and non-photorealistic scenes. Notably, the computational requirements required to achieve an equivalent accuracy are significantly reduced when compared to the top algorithms. As a consequence, the proposed framework is suitable for deployment in consumer and industrial applications.

Keywords: Multi-Camera; Depth; Regularization; Light Field.

***First Author, E-mail:** h.javidnia1@nuigalway.ie

1 Introduction

The use of consumer light field cameras such as Lytro [1], Raytrix [2] and multi-camera array in smart-phones [3-5] has received much attention in the past decade. A light field camera contains multiple viewpoints and captures the intensity of each light ray and sufficient angular information which can reveal important information about the structure of the scene.

These types of cameras have been adapted in a wide range of applications such as saliency detection [6], depth estimation [7-11], digital refocusing [12,13], super-resolution [14] and scene reconstruction [15]. Recent advances in light field imaging technology enable reconstruction of scene depth in a more effective way than with conventional cameras; however, acquiring an accurate *and* dense depth map from these cameras has presented a new challenge for researchers in recent years. One of the important features of light field cameras is the ability to differentiate the rays passing through the lens which makes it easy to provide both monocular and stereo depth cues. A light field camera can extract stereo cues by capturing both magnitude and angular direction of each ray passing through the microlens while recording a scene [16]. However, in such a camera, the maximum stereo baseline is equal to the lens diameter, meaning it is often rather small.

One of the most common techniques for estimating depth from light field data is to exploit the Epipolar Plane Image (EPI) [17]. This has the advantage of being both simple to execute and fast to compute, but the accuracy is limited by the small camera baseline that is typical of these array cameras and most importantly by the illumination variation while capturing Lambertian scenes. An EPI based approach is employed in this paper to initiate the depth estimation framework. In the same way that depth estimation is performed in simple stereo image pairs, the depth from a light field set is computed from a set of rectified[*] images. In EPI, every pixel can be projected into a slope line which represents the depth of the corresponding scene point. The performance of applications that employ light field imaging technology is influenced by the precision of the estimated depth map. However, using only EPI to estimate depth from light field cameras introduces many challenges arising from noise in the depth map, statistical uncertainties in depth values and structural inaccuracies. We tackle these challenges by taking advantage of the Fenchel-Rockafellar duality [18] and Total Variation (TV) minimization.

Fig. 1 illustrates the schematic of a “type 1” light field camera [12] where the object is located in position (A), the camera aperture is shown in position (B) and the camera array (D) is aligned on a regular 2D grid between the main lens (C) and the image sensor (E). Each microlens located on

[*] By applying rectification, all the images from the light field set are projected onto a common image plane.

the camera array (D), diverges the incoming light ray based on its direction. This enables the pixels underneath it to record the original rays coming from different areas of the main lens (C).

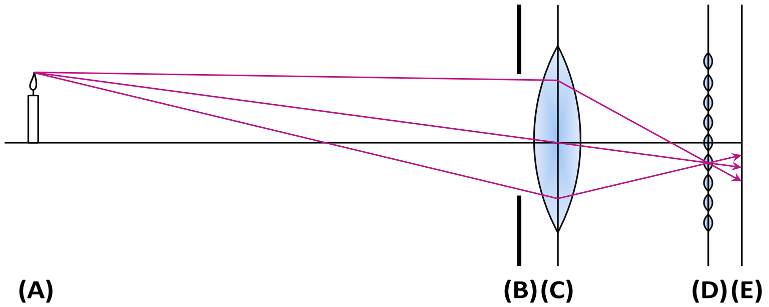
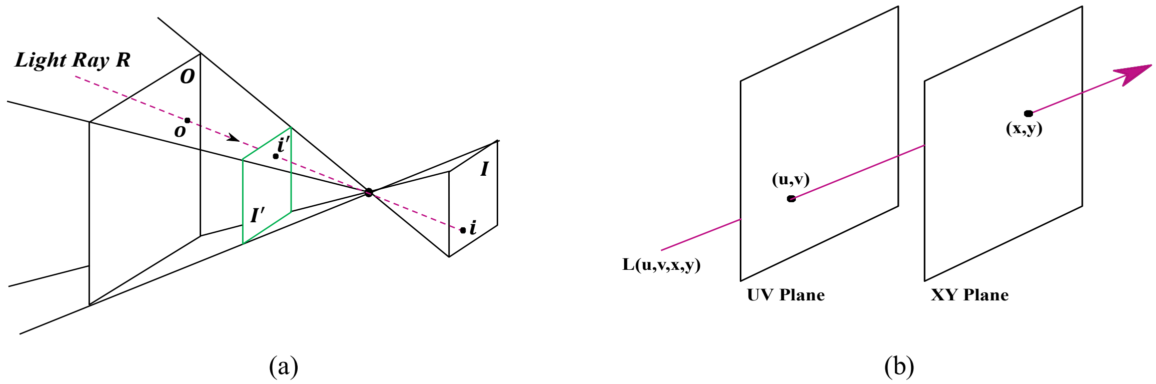


Fig 1 The schematic of a light field camera. (A): Object. (B): Camera aperture. (C): Main lens. (D): Camera array. (E): Image sensor.

Generally, a conventional pinhole camera generates an image by creating a 2D projection of a 3D scene inside a polyhedral shape as presented in Fig.2.a.



(a)

(b)

Fig 2 (a) A conventional pinhole camera model with image plane ***I*** projecting the 3D world inside a polyhedronal shape. (b) Two-plane parameterization of the 4D light field.

The intensity of the pixel i in the image plane I is the intensity of the unique ray R passing through the image plane and the plane containing the viewing points or the corresponding point o in the object plane O. Whilst a pinhole camera defines a unique ray direction R, it is impractical because of light flux and resolution limitations. A camera with a finite lens diameter collects more light and has higher angular resolution but the intensity at any point in the image plane is now the incoherent sum of intensities from many ray directions. This drawback has been tackled by employing light field imaging techniques.

Light field rendering theory explains the 4D light field data as a collection of pinhole views parallel to an image plane [19]. Commonly the position of the 2D image plane is considered as (x, y) and the position of each viewing point as (u, v). Fig.2.b illustrates the two-plane parameterization of the light field, proposed in [19] to simplify the plenoptic function to a 4D function. In multi-camera arrays the image sensor plane of each camera indicates (x, y) and the position of the lens indicates (u, v). In other words (x, y) can be referred to as a pixel in the image and (u, v) as the position of the camera in the array. Each pixel of the 4D light field data represents the intensity of the ray passing through image plane and the plane containing the viewing points. The light field data is stored as a 4D object as $L = (U, V, X, Y)$. Any point in the L can be identified by its coordinates $[u, v, x, y]$.

In this paper, a depth estimation framework is proposed based on local EPI analysis and Total Variation (TV) minimization. The proposed minimization problem takes advantage of the Fenchel-Rockafellar duality [18]. The point in using Fenchel-Rockafellar duality [18] is that the lower bound of the minimum value will be obtained by solving the dual problem [20]. The solution of the primal one can be found much faster by taking advantage of the information on lower bound of the minimum value. Rockafellar has proved that in convex minimization problems, a dual problem can be allied to the primal problem by enclosing the problem in a set of perturbed problems and using the theory of conjugate convex functions [18,21]. More specifically, assume that $\mathcal{S}$ and $\mathcal{Z}$ are convex, proper, and lower semi-continuous functions. $\mathscr{s}^*$ and $\mathscr{z}^*$ represent the infimum of the functions which are minimized in the primal and dual problems, respectively. In this case, the solution to the primal (minimization) problem is always greater than or equal to the solution to the associated dual problem. In other words $\mathcal{s}^* \geq -\mathcal{z}^*$. In the case where $\mathcal{s}^*$is finite, $\mathcal{s}^* + \mathcal{z}^*$ is considered as the duality gap which the difference between the primal and dual solutions. $\mathcal{S}$ and $\mathcal{Z}$ are considered to be convex, proper, and lower semi-continuous functions. Therefore, under a proper condition, there is always a solution for the dual problem and the difference between the primal and dual solutions vanishes. When this difference is equal to zero, the primal optimal objective and the dual optimal objective are equal.

The main contributions of this work are:

1- Introducing a lightweight computational framework to estimate depth from the 4D light field on the EPI. The proposed framework is less sensitive to occlusion, noise, spatial aliasing, angular resolution and more importantly it is 2-100 times faster/more computationally efficient than the studied state of the art methods.

2- Proposing a new computational cost function derived from the Fenchel-Rockafellar duality [18].

The rest of this paper is organized as follows. Section 2 summarizes the state of the art technology for light field depth estimation. The proposed framework is presented in detail in Section 3. The evaluation and benchmarking details are outlined in Section 4 and finally, the analyses are discussed in Section 5

2 Previous Work

A lot of efforts have been made in the context of depth estimation from light field cameras including multi-view stereo matching methods [7,22] or tensor-based methods [23,24]. The following categories summarize different approaches on depth estimation from light field.

2.1 Depth from Light Field using EPI Analysis

Zhang *et al.* [25] proposed an algorithm for light field depth estimation by utilizing the linear structure of EPI [17]. The optimal slope of each line in EPI is selected from a set of candidate angles. The intensity pixel value, gradient pixel value and spatial smoothness consistency are used to aggregate the cost volume. Reliability of each pixel's disparity is identified by analyzing the matching cost curve and locally linear embedding method is used to estimate the disparity of unreliable pixels. Ma *et al.* [26] obtain the sparse depth information of the edges by exploiting local EPI analysis which is used to generate the global depth map by using regional interpolation. Yang *et al.* [27] estimate the disparity map by analyzing the EPI and detecting the slopes using the multi-label technique. Later a linear calibration method is proposed to compensate the error between the disparity values and the actual distances. Zhang *et al.* [28] proposed a spinning parallelogram operator to calculate the orientation of the EPI lines for local depth estimation. The depth estimation is based on the measurement of the slopes in EPI by maximizing the distribution distances of two parts of the parallelogram window. Further, a confidence metric is defined to reduce the effect of the occlusions. From the approaches taken by these researchers, it is evident that the objects reflectance properties are not considered by these methods. Generally, in real scenes, the illumination is not constant over time and that introduces many challenges to depth estimation methods.

The initial step of the depth estimation framework presented in this paper falls into this category. The state of the art depth estimation methods which take advantage of EPI analysis are computationally expensive due to the cost aggregation, densification of sparse depth map or depth confidence measurement. However, the presented method utilizes a regularizer to refine the initial depth map which computationally outperforms the state of the art methods.

2.2 Occlusion-Aware Depth Estimation from Light Field

Wang *et al.* [29] proposed an occlusion aware light field depth estimation algorithm by modifying the photo-consistency condition on angular pixels. This modification along with the means or variances in the angular domain and spatial domain are used to estimate the occlusion-aware depth. In a similar approach [30] a novel data cost volume is introduced based on the correspondence and defocus cues followed by graph cut optimization to handle the occlusion in depth from the 4D light field. However, these methods struggle in handling heavy occlusions. They are mainly focused on the points which are visible in the reference view and invisible in other views. The present research does not fall into this category as handling occlusion is not the main goal of the proposed framework.

2.3 Light Field Depth Estimation and Optimization

Liu *et al.* [31] tackled the light field depth estimation challenge by approaching it as an optimization problem. The objective function includes three terms as fidelity, gradient and classification. The mismatching pixels are corrected iteratively by minimizing the objective function which results in a more accurate depth map. In a similar attempt, Monteiro *et al.* [32] employed Alternating Direction Method of Multipliers to regularize the 2D EPIs and generate a dense disparity map. Unfortunately the computational time of these methods are not reported, however, their objective function contains different pixel-wise terms which introduce high computational demands and a high number of iterations to minimize.

The second part of the depth estimation framework presented in this paper falls into this category. The proposed framework has three main advantages compared to the studied methods in this category:

1- Faster computational time.

2- Better convergence rate of optimization function.

3- Lower residual norm.

2.4 Light Field Depth Estimation and Stereo Framework

Kim *et al.* [33] proposed a framework to generate stereo images from a set of light field data. Their framework is based on 3D light field and its corresponding 3D disparity volume and defines each stereo image as continues cuts through that. Graph cut optimization is also used to calculate the multi-perspective cuts. Basha *et al.* [34] used the multi-camera array for 3D reconstruction purposes by capturing a scene at two different time intervals. A 3D volume is reconstructed for each image set and the corresponding scalar volume is calculated using a nonlinear filter. The final 3D structure and motion are estimated by matching the two scalar volumes. Navarro *et al.* [35] used multi-scale and multi-window stereo method [36] to estimate disparity from two views of the light field image array. The disparity is estimated from the central view and the views in the same row and column. Later, an interpolation method is introduced based on the optical flow approach in [37] to combine the estimated disparity maps and generate the final depth map. These methods have complex disparity constraints and they require a high number of viewpoints. In graph-based methods, the size of the constructed graph increases significantly by adding more viewpoints to the light field set and that is a computationally intense process. On the other hand, reducing the number of viewpoints introduces notable artifacts to the depth map. In interpolation based methods, depth refinement based on optical flow formulation requires significant computational time which goes up to 1 hour and 30 minutes to process one light field image set.

The present research does not fall into this category. The efficient optimization of the proposed framework does not contain complex disparity constraints. Therefore, the proposed framework is significantly more efficient than the methods presented in this category.

2.5 Light Field Depth Estimation and Focal Stack Framework

Pérez *et al.* [38] proposed a focal stack frequency decomposition algorithm from light field images based on the trigonometric interpolation principle as the discrete focal stack transform. The proposed method in [38] utilizes fast discrete Fourier transform to generate refocus planes in a reasonably fast computational time. The reverse of this transformation in studied in [39] where a focal stack is used to obtain a 4D light field image set using discrete focal stack transform. Unlike [38], Mousnier *et al.* [40] presented an approach to reconstruct 4D light field image sets from a stack of images taken by a fixed camera at different focal points. The algorithm initiates by calculating the focus map by utilizing region expansion with graph cut. Later, the depth map is estimated based on the calibration details of the camera and it is used to reconstruct the Epipolar images. The reconstructed Epipolar images are used for refocusing purposes. Lee *et al.* [41,42] proposed a depth estimation method by separating foreground and background of the focus plane. The separated parts are converted to a binary map using the Lambertian assumption and gradient constraint. The final disparity map is estimated by accumulating the binary maps. Using the focal stack symmetry for the application of depth estimation from the 4D light field has the advantage of fast computational time, however, the final estimated depth maps suffer from sever puzzling artifact, false depth values on objects' surface, and false depth values on the non-Lambertian region. Generally, the methods which are based on focal stack symmetry are highly affected by the lack of angular resolution which mostly causes false depth values on regions with a repeated pattern. Unlike these methods, the proposed framework is not affected by the lack of angular resolution and it's capable of estimating wider depth planes with much less faulty depth values.

2.6 Physical Changes in Light Field Depth Estimation

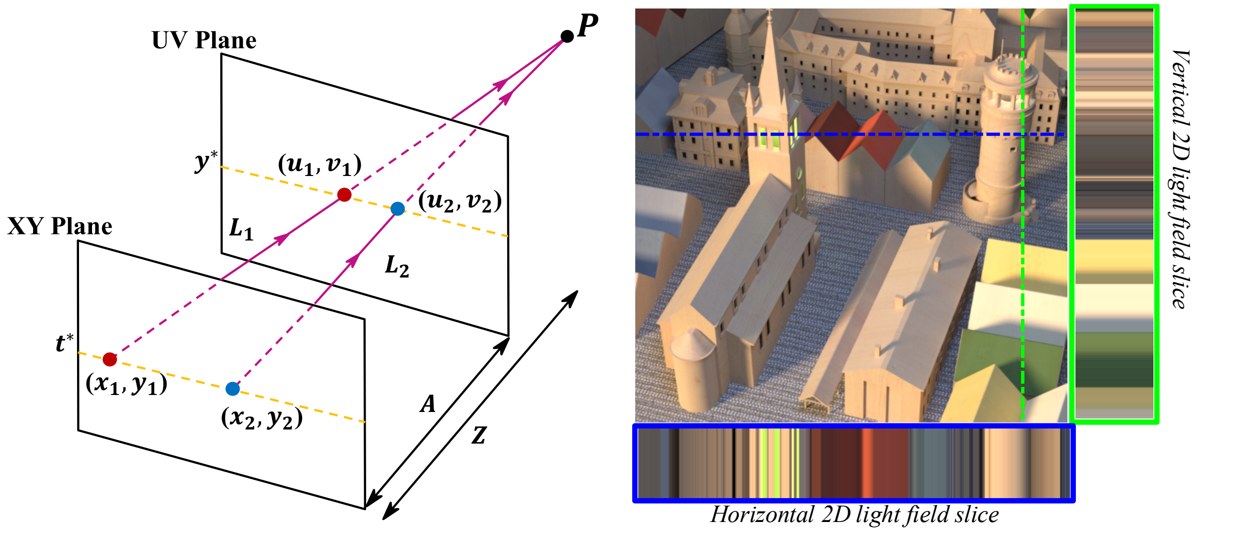
Besides all the state of the art computational approaches for depth estimation from light field, Diebold *et al.* [43] studied the effect of light field imaging system's setup and design on the accuracy of the depth estimation. They concluded that variation in focal length and baseline of the micro cameras in an array can result in depth precision loss. It was recommended to use a precise translation stage as a good alternative for light field cameras. The goal of the present study is to propose a computationally efficient and accurate depth estimation framework for light field image sets. We do not consider any physical changes in the imaging system.

3 Method

3.1 Initial Depth Estimation

Capturing a sequence of images using a multi-camera array is similar to capturing the same sequence by linearly translating one camera. Changing the camera position causes the positional changes in the image plane. Drawing out a horizontal line of constant y^* in the image plane and a constant t^* as the camera coordinate results in a map called EPI which can be used to visualize the positional changes in image plane. Fig.3.b shows a sample of horizontal and vertical Epipolar slices. An important feature of EPI is representing a point which is visible to all sub-aperture images by mapping it to a straight line. This feature has been used in variety of applications such as segmentation [44] and depth estimation [23]. The approach here takes advantage of this feature.

To estimate the depth map, we employed the initial part of the depth estimation algorithm of [23]; the local depth estimation on EPIs where the initial depth is constructed by using a structure tensor on each EPI E_{y,v^*} $(y = 1, \dots, Y)$ and E_{x,u^*} $(x = 1, \dots, X)$ to estimate the slope of the EPI lines. Two slopes are estimated for each pixel in a sub-aperture image I_{u^*,v^*}, one for each EPI.


UV Plane
$\boldsymbol{P}$
y^*
$(\boldsymbol{u_1}, \boldsymbol{v_1})$
$(\boldsymbol{u_2}, \boldsymbol{v_2})$
XY Plane
$\boldsymbol{L_1}$
$\boldsymbol{L_2}$
$\boldsymbol{t^*}$
$(\boldsymbol{x_1}, \boldsymbol{y_1})$
$(\boldsymbol{x_2}, \boldsymbol{y_2})$
$\boldsymbol{A}$
$\boldsymbol{Z}$
Vertical 2D light field slice
Horizontal 2D light field slice

(a)

(b)

Fig 3 The concept of depth calculation from the light field using EPI analysis. (a) The depth of point ***P*** can be estimated by calculating either the vertical or horizontal slope from the light field 2D slices. (b) Visualization of Epipolar image. The central view of a 9×9 camera array with horizontal and vertical 2D light field slices.

As illustrated in Fig.3.a the light rays L_1 and L_2 converge at point P so, the following geometrical relations can be defined with regards to the depth of the point:

$$\begin{cases} x_1 + \dfrac{u_1 - x_1}{A} Z = x_2 + \dfrac{u_2 - x_2}{A} Z \Longrightarrow \dfrac{\Delta u}{\Delta x} = \dfrac{Z - A}{Z} \\ y_1 + \dfrac{v_1 - y_1}{A} Z = y_2 + \dfrac{v_2 - y_2}{A} Z \Longrightarrow \dfrac{\Delta v}{\Delta y} = \dfrac{Z - A}{Z} \end{cases} \qquad (1)$$

where A indicates the distance between two planes. Z represents the depth of the point P from the plane XY. The disparity of angular (x, y) and spatial (u, v) coordinates are $\Delta x = x_2 - x_1$ and $\Delta u = u_2 - u_1$, respectively. Either vertical slope $\Delta u / \Delta x$ or horizontal slope $\Delta v / \Delta y$ can be used to estimate the depth Z. The value of the slope is defined as the maximum pixel disparity of an object point (in pixel) among all views divided by the number of views.

The estimated slopes are combined by minimizing the following global energy function [23]:

$$\mathcal{A}(e) = H(e) + \sum_{i=1}^{N} \int_{\sum_{y^*,t^*}} c_i e_i d(u,x) \qquad (2)$$

where $e = (e_1, \dots, e_N)$ is a vector of indicator functions, N is the number of discrete depth labels, c_i is the local cost function and H is the regularizer. y^*, t^* are the fixed horizontal lines in the image and camera planes, respectively which helps to consider the structure of the light field as a set of 2D slices $\sum_{y^*,t^*}$.

Despite the fast performance of this method, it results in a noisy and unreliable depth values especially at smoother regions [45]. To tackle this issue we propose a regularization framework based on TV minimization. In the rest of this paper, the initial depth map is denoted by D.

3.2 Regularization Framework

The proposed approach can be formulated within a discrete framework. Consider a weighted graph $\mathcal{g} = (V, E)$ with vertices $v \in V$ and edges $e \in E \subseteq V \times V$, with cardinalities $n = |V|$ and $m = |E|$.

An edge passing over two vertices v_i and v_j is declared as $e_{i,j}$. The present paper focuses on weighted graphs which imply weights on both edges and nodes. A value assigned to each edge and node is known as edge weight $w_{i,j}$ and node weight $\mathcal{n}_i$, respectively.

The goal is to deduce a restored vector J in close proximity to the rough vector D considering smooth variations of intensities inside the object.

Let's consider λ as a positive regularization parameter. In a continuous framework, considering a planar domain Ω and its two arbitrary points u, u', the weighted TV [46] can be defined as:

$$\min_{J}\int_{\Omega}\left(\int_{\Omega}\omega(u,u')\big(J(u')-J(u)\big)^{2}\,du'\right)^{1/2}du+\frac{1}{2\lambda}\int_{\Omega}\big(J(u)-D(u)\big)^{2}\,du \qquad (3)$$

where ω is a non-negative valued function on Ω^2. Based on [47] the weighted TV minimization problem in Eq. (3) can be redefined as a min-max problem:

$$\min_{J}\left(\max_{\|\mathrm{P}\|_{\infty}\leq 1}\int_{\Omega^2}\omega(u,u')^{1/2}\big(J(u')-J(u)\big)\mathrm{P}(u,u')\,du\,du'\right.$$

$$\left.+\frac{1}{2\lambda}\int_{\Omega}\big(J(u)-D(u)\big)^{2}\,du\right) \qquad (4)$$

where P is a two variable vector field $\|\mathrm{P}\|_{\infty}=\sup_{u\in\Omega}\left(\int_{\Omega}\mathrm{P}(u,u')^{2}\,du'\right)^{1/2}$.

To establish the discrete framework we define W as the weighted incidence matrix of $\mathcal{g}$ which is used to characterize the discretized gradient and is a fundamental operator for defining combinatorial formulations of variational problems. W is defined for each vertex v_k and edge $e_{i,j}$ as:

$$W_{e_{i,j}v_k}=\begin{cases}-1 & if\ i=k,\\ +1 & if\ j=k,\\ 0 & otherwise,\end{cases} \qquad (5)$$

For any arbitrary matrix A, $|\mathrm{A}|$ represents the matrix constructed from the absolute value of each entry individually, $\cdot$ denotes the Hadamard product while A^2 shows the product $\mathrm{A}\,.\,\mathrm{A}$.

The discrete version of Eq. (3) and its dual Eq. (4) tend to approximate the continuous version the step size of the graph $\mathcal{g}$ moves towards 0. The discrete weighted TV model defined in [46,48,49] is:

$$\min_{J}\sum_{i=1}^{n}\left(\sum_{j\in N_i}\omega_{i,j}\big(J_j-J_i\big)^{2}\right)^{1/2}+\frac{1}{2\lambda}\sum_{i=1}^{n}(J_i-D_i)^{2} \qquad (6)$$

where $N_i = \{j \in \{1, \dots, n\} | e_{i,j} \in E\}$. The dual formulation of the problem which is optimized by [46] is defined as:

$$\min_{J} \max_{\|\mathrm{P}\|_\infty \leq 1} \mathrm{P}^T((WJ).\sqrt{\omega}) + \frac{1}{2\lambda}\|J - D\|^2 \qquad (7)$$

where $\|\mathrm{P}\|_\infty = \max_{i\in\{1\dots n\}}\left(\sum_{j\in N_i} \mathrm{P}_{i,j}^2\right)^{1/2}$. Eq. (7) is the combinatorial primal-dual formulation of Eq. (6). If we define the vector $F = (F_{i,j})_{i,j}$ given that for every $i \in \{1, \dots, n\}$, $j \in N_i$ and $F_{i,j} = \mathrm{P}_{i,j}\sqrt{\omega_{i,j}}$ then the problem can be reformulated as:

$$\min_{J} \max_{F\in B} F^{\mathrm{T}}WJ + \frac{1}{2\lambda}\|J - D\|^2 \qquad (8)$$

$$B = \left\{ \left(F_{i,j}\right)_{i,j} | (\forall i \in \{1, \dots, n\}) \sum_{j\in N_i} \frac{F_{i,j}^2}{\omega_{i,j}} \leq 1 \right\} \qquad (9)$$

The following minimization problem is proposed to extend the discrete weighted TV formulation in Eq. (8).

$$min_{J\in\mathbb{R}^n}\left(sup_{F\in B} F^{\mathrm{T}}WJ + \frac{1}{2}(MJ - D)^{\mathrm{T}}(MJ - D)\mathcal{V}^{-1}\right) \qquad (10)$$

$M \in \mathbb{R}^{b\times n}$ and $\mathcal{V}$ is a symmetric positive-definite weighting matrix. The vector F is a vector representing the edges of $\mathcal{g}$. B is the intersection of closed convex defined with weighted semi-norms as:

$$B = \left\{ F \in \mathbb{R}^m \middle| (\forall i \in \{1, \dots, n\}) \left\|\theta^i . F\right\|_\alpha \leq \mathcal{n}_i \right\} \qquad (11)$$

where $\| . \|_\alpha$ is the ℓ_α norm of $\mathbb{R}^m$ with $\alpha \in [1, +\infty]$. $\mathcal{n} = (\mathcal{n}_i)_{1\leq i\leq n}$ is a vector of $[0, +\infty[^n$. θ^i is a vector of multiplicative constants.

Any solution for Eq. (10) is parametrized as the optimal value corresponds to each node of the weighted graph $\mathcal{g}$.

To solve the minimization problem expressed in Eq. (10) we define the support function of B as ϱ_B assuming B is a nonempty closed convex subset of $\mathbb{R}^n$ as:

$$\varrho_B: \mathbb{R}^n \longrightarrow]-\infty, +\infty]: a \mapsto sup_{F\in B} F^T a \qquad (12)$$

This lower semi-continuous convex function is the conjugate of the indicator function ι_B:

$$\iota_B = F \mapsto \begin{cases} 0, & if\ F \in B, \\ +\infty, & otherwise. \end{cases} \qquad (13)$$

which leads to the modified version of the optimization function in Eq. (10):

$$min_{J\in\mathbb{R}^n} \left(\varrho_B(WJ) + \frac{1}{2}(MJ - D)^T (MJ - D)\mathcal{V}^{-1} + \frac{\daleth\|ZJ\|^2}{2} \right) \qquad (14)$$

where $\daleth \in]0, +\infty[$ and $Z \in \mathbb{R}^{n\times n}$ is the projection matrix onto the nullspace of the M.

Eq. (14) can become equivalent to Eq. (10) where M is injective. The term $J \mapsto \frac{\daleth\|ZJ\|^2}{2}$ vanishes when M is injective. However, it helps the objective function to stay convex by bringing an additional regularization term when M is not injective. Assuming B is a nonempty closed convex then Eq. (14) acknowledges a distinctive solution. In this case, Eq. (14) can be redefined based on Fenchel-Rockafellar duality theorem [18] as:

$$min_F \phi(F) + \iota_B(F) \qquad (15)$$

where $\phi: F \mapsto \frac{F^T W\gamma W^T F}{2} - F^T W\gamma M^T D\mathcal{V}^{-1}$ and $\gamma = (\mathcal{V}^{-1}M^T M + \daleth Z)^{-1}$. The optimum solution $\hat{J}$ of Eq. (14) is concluded from each optimum solution $\hat{F}$ of the duality problem in Eq. (15) by the following relation:

$$\hat{J} = \gamma\left(\mathcal{V}^{-1}M^T D - W^T\hat{F}\right) \qquad (16)$$

The indicator function ι_B can be broken down into the sum of indicator functions of the convex subsets. Consequently, the Fenchel-Rockafellar duality [18] in Eq. (15) can be re-written as:

$$min_{F\in\mathbb{R}^m}\sum_{q=1}^{e}\iota_{B_q}(F)+\ \phi(F) \qquad (17)$$

where for each set $B_q,\ q \in \{1,\dots,e\}$. The above convex function is optimized by employing Parallel Proximal algorithm [50] as shown in Algorithm 1.

Algorithm 1: General form of Parallel proximal algorithm

```
Set $\lambda \in [0,+\infty]$, $\lambda_\ell \in [0,2]$
For $q \in \{1,\dots,e\}$ set $(\mathscr{w}_q)_{1\le q\le e} \in [0,1]$
Set $(y_{q,0})_{1\le q\le e} \in (\mathbb{R}^m)^e$
For $i\ =\ 0:\dots$
  For $q\ =\ 1:e$
     $\mathscr{p}_{q,i} = prox_{\lambda D_q/\mathscr{w}_q} y_{q,i} + \alpha_{q,i}$
  $\mathscr{p}_i = \sum_{q=1}^{e} \mathscr{w}_q \mathscr{p}_{q,i}$
  For $q\ =\ 1:e$
     $y_{q,i+1} = y_{q,i} + \lambda_\ell(\mathscr{p}_i - F_i - \mathscr{p}_{q,i})$
  $F_{i+1} = F_i + \frac{\lambda_\ell(\mathscr{p}_i - F_i)}{2}$
```

λ and $\mathscr{w}_q$ are the positive regularization parameter and a positive constant, respectively. Beside the possible error term α, a relaxation parameter λ_ℓ is defined in each iteration.

4 Evaluation

The evaluation of the technique proposed in this paper is performed in two parts and is based on HCI, the Heidelberg 4D Light Field Benchmark [51]. The first part presents the evaluation of the optimization function and the second part compares the accuracy of the estimated depth map against the state of the art methods which are ranked in HCI, Heidelberg benchmark. The evaluation is performed using the standard "evaluation package/toolkit" provided by the benchmark to assess the performance of the proposed framework.

This benchmark is the first in the state of the art which provides light field image sets with ground truth data and standardized the evaluation framework. The light field image sets are designed to challenge accuracy and reliability of different algorithms in different aspects such as occlusion handling, performance on convex versus concave geometry, keeping fine structure and etc.

The current version of the benchmark provides $9 \times 9 \times 512 \times 512 \times 3$ light field images along with corresponding camera configuration files. The benchmark contains 3 sets including Stratified, Test and Training. These categories are pre-defined in the benchmark. The Stratified set contains 4 light field image sets as shown in Fig.4.a-Fig.4.d. The goal of the Stratified set is to introduce challenges which can lead to fine-tuning algorithm parameters and performance metrics for real-world images. The Training set contains 4 light field photorealistic image sets as illustrated in Fig.4.e-Fig.4.h. The goal of this set is to evaluate the performance of algorithms on respecting scene structures, handling complex occlusions, slanted planar surfaces, and continuous non-planar surfaces.

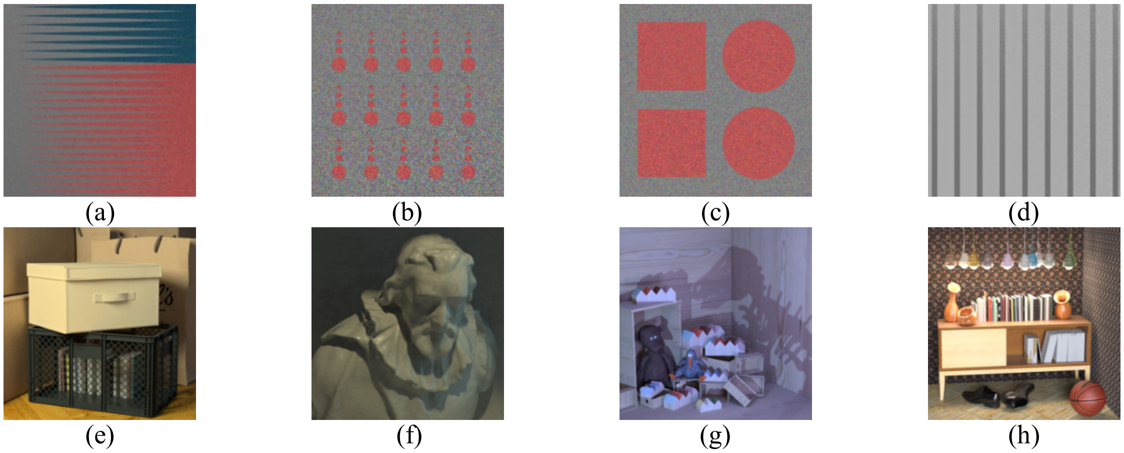


Fig 4 4D Light field image sets used for evaluation. First row shows the four stratified scenes and the second row shows the four photorealistic training scenes. (a) Stratified–Backgammon. (b) Stratified–Dots. (c) Stratified–Pyramids. (d) Stratified–Stripes. (e) Training–Boxes. (f) Training–Cotton. (g) Training–Dino. (h) Training–Sideboard.

In this paper, Stratified and Training sets are used for comparison purposes which include 8 different light field image sets with different configurations. The ground truth data for all these image sets is provided by the HCI benchmark.

4.1 Residual Norm Evaluation

This section presents the convergence analysis of the optimization function. The maximum number of iterations i, the regularization parameter λ and ٦ are set to 300, 0.5 and 0 respectively for light field sets used in this paper. These values are chosen experimentally and for the evaluated image sets, they provide the average best results. Note that, the regularization parameter varies based on the type of the data. Similar to TV minimization, the bigger value of λ generates a smoother result with lower convergence error (in close range scenes and small number of depth planes). However, based on the experiments it is not recommend to set $\lambda >$ 0.8 .

Fig. 5 illustrates the residual norms of the optimization function for each light field image set in Stratified and Training sets.

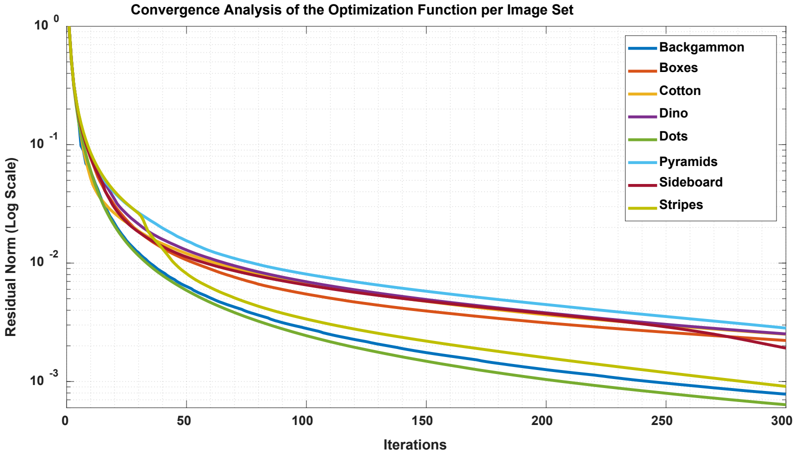


Fig 5 Residual norm analyses of the minimization function for stratified and training sets. The maximum iteration number is set to 300 for all scenes.

As shown in this figure, the presented optimization method, results in a considerably low convergence error after ~50 iterations. The average convergence error of 0.01 at iteration 50 outlines the fast performance of the optimization function. Residual norms are used to verify a solution to the optimization function by substituting it into the function. The residual vanishes when the optimal solution is found [52,53].

4.2 Disparity Estimation Evaluation

In this section, the accuracy of the estimated disparity maps are compared against the provided ground truth, 6 top algorithms ranked by the benchmark and one baseline algorithm. The best estimated disparity maps from all these algorithms are provided by their authors and the benchmark. The top algorithms are chosen based on the average value of the BadPix(0.03) metric and include OBER-cross+ANP [54], SPO-MO [54], OBER-cross [54], OFSY_330/DNR [55], PS_RF [54] and SPO [28]. The baseline algorithm is EPI2 [23] (the local depth estimation on EPIs) which is used to provide the initial depth map in this paper. As this research is more focused on generating accurate depth map and increasing its accuracy, three metrics including BadPix(0.07), MSE and Q25 are chosen for comparison purposes. These metrics are categorized as "High accuracy metrics" [56] in this benchmark. The BadPix(0.07) is quantified as:

$$BadPix_M(0.07) = \frac{|\{x \in M: |d(x) - gt(x)| > 0.07\}|}{|M|} \qquad (18)$$

where d is the estimated disparity map, gt is the ground truth disparity map and M is the evaluation mask. BadPix(0.07) shows the percentage of pixels at the given mask with $|d - gt| > 0.07$. The error threshold "0.07" is the default value defined by the benchmark.

MSE shows the mean squared error over all pixels at the given mask, multiplied with 100:

$$MSE_M = \frac{\sum_{x \in M}(d(x) - gt(x))^2}{|M|} \times 100 \qquad (19)$$

Q25 represents the maximum absolute disparity error of the best 25% of pixels for each algorithm, multiplied by 100.

Fig. 6 and Fig. 7 visualize the distribution of the accurate pixels and mean square error of the proposed method and the top state of the art algorithms for Stratified and Training sets, respectively. Each column in these figures shows the result of an algorithm and each row visualizes a metric. For all the metrics the lower values show a better result. These figures illustrate the performance of the proposed method based on the high accuracy metrics while dealing with different noise level, complex occlusions, slanted planar surfaces and complex scene structure.

In BadPix(0.07) metric, the good pixels are shown in green and the faulty ones are presented in red. In MSE, the correct pixel values are shown in white, the pixels with too close values are illustrated in blue and the pixels with too far values are shown in red. In Q25, the white/yellow parts indicate the good and the red parts indicate relatively bad pixels.

The fluctuation in the ranking of the algorithms based on each metric raise from their differences in terms of data and final optimization term. Some of these algorithms such as EPI2, SPO and OBER-cross estimate the disparity based on EPI analysis. OFSY_330/DNR utilizes the focal stack symmetry for disparity estimation and PS_RF uses the multi-view stereo approach by building individual cost volumes for its data terms.

According to the evaluations and as shown in Fig. 6 and Fig. 7 there is no single best algorithm to be considered as the superior one. As it is challenging from Fig. 6 and Fig. 7 to understand the relative merits of each technique we provide Table 1 and Table 2 which outline the average numerical values per metric per algorithm for the images in each set in both stratified and photorealistic scenes, respectively.

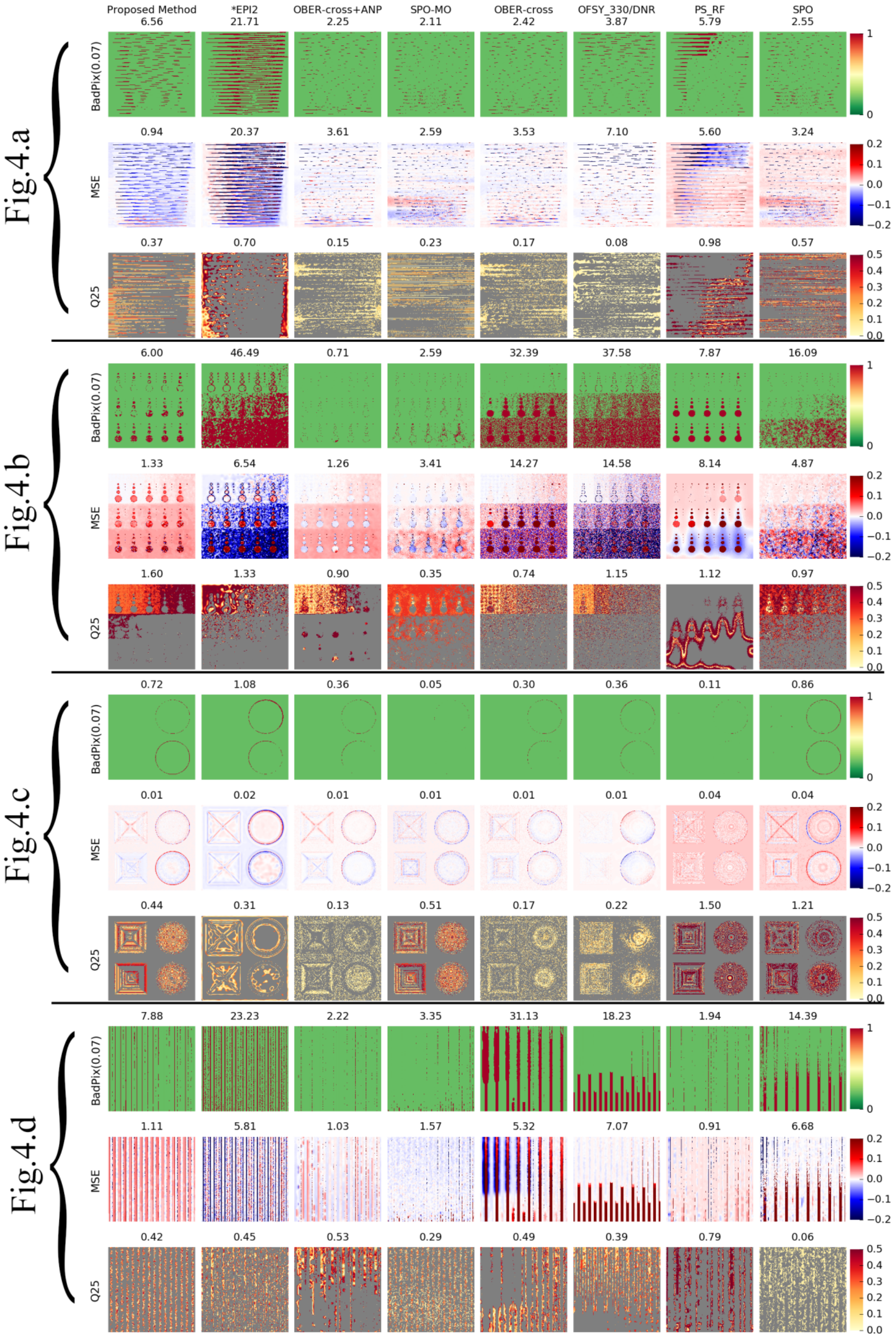


Fig 6 Stratified Scenes-Visualizations of BadPix(0.07), MSE and Q25 error metrics per algorithm are shown for the proposed method, a baseline and six most accurate algorithms on the Stratified set. Each column represents an algorithm. The rows with BadPix(0.07) show the percentage of pixels at the given mask with $|\boldsymbol{d} - \boldsymbol{gt}| > \mathbf{0.07}$. The row with MSE label show the mean square error map and the row with Q25 label shows the absolute error of the 25% of the best pixels for each algorithm

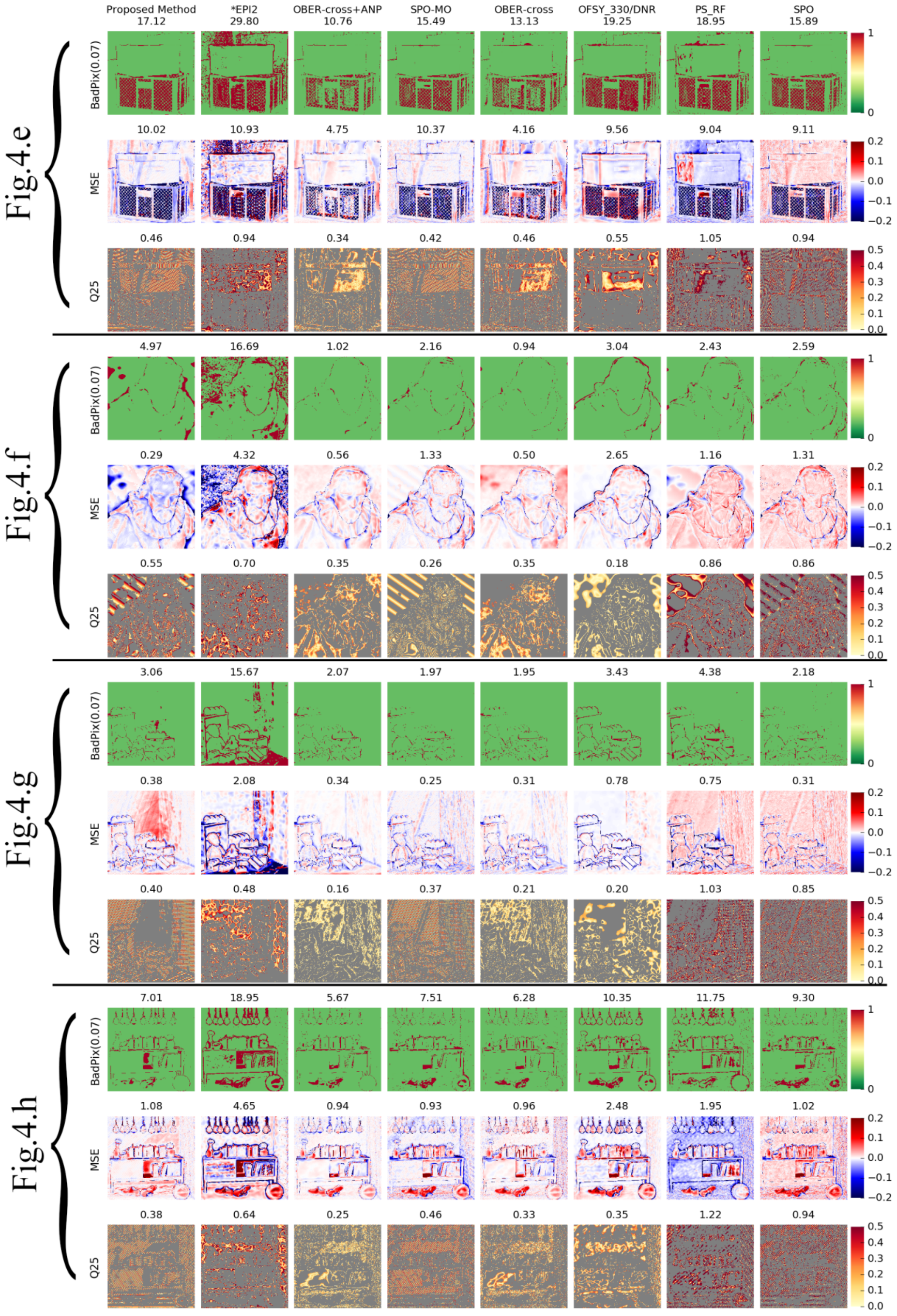


Fig 7 Training Scenes-Visualizations of BadPix(0.07), MSE and Q25 error metrics per algorithm are shown for the proposed method, a baseline and six most accurate algorithms on the Training set. Each column represents an algorithm. The rows with BadPix(0.07) show the percentage of pixels at the given mask with $|\boldsymbol{d} - \boldsymbol{gt}| > \mathbf{0.07}$. The row with MSE label show the mean square error map and the row with Q25 label shows the absolute error of the 25% of the best pixels for each algorithm

The values presented in these tables outline the close performance of the proposed method in comparison to the top state of the art algorithms. The cells are color encoded in each row based on the ranking of each method per metric. The green represents the best performing method and red shows the poorest performing one. These values indicate that the method proposed in this work can estimate depth maps with accuracy very close to the most accurate methods in the benchmark. The proposed method provides a reduction of ~56.5% averaged across all the error metrics when compared to the baseline algorithm EPI2.

Table 1 Average values of metric per algorithm for the images in Stratified set

	Proposed Method	EPI2	OBER-cross+ANP	SPO-MO	OBER-cross	OFSY_330/DNR	PS_RF	SPO
BadPix(0.07)	5.29	23.12	1.38	2.02	16.56	15.01	3.92	8.47
MSE	0.84	8.18	1.47	1.89	5.78	7.19	3.67	3.7
Q25	0.707	0.69	0.42	0.31	0.39	0.46	1.09	0.702

Table 2 Average values of metric per algorithm for the images in Training set

	Proposed Method	EPI2	OBER-cross+ANP	SPO-MO	OBER-cross	OFSY_330/DNR	PS_RF	SPO
BadPix(0.07)	8.04	20.27	4.88	6.78	5.57	9.017	9.37	7.49
MSE	2.94	5.49	1.64	3.22	1.48	3.86	3.22	2.93
Q25	0.44	0.69	0.27	0.37	0.33	0.32	1.04	0.89

Fig. 8 represents the difficulty of each scene type as a heatmap. Each pixel in the heatmap represents the percentage of pixels with the disparity error > 0.07 pixels averaged across all of the algorithms. Thus more than 90% of these algorithms struggle in detecting the correct disparity value for the pixels inside the box in the “Boxes” image set as shown in the first image in Fig. 8.

Another example is the “Sideboard” image set where 80-95% of the algorithms struggle with estimating the correct disparity on the surface of the shoes. The brighter parts in this figure indicate challenging areas. The "Backgammon" scene challenges the algorithms in occlusions and keeping fine structure and the "Stripes" set evaluates the methods for handling textured occlusion boundaries.

Using "Dots" image set, the robustness of each algorithm is evaluated against camera noise. The heatmap for the “Dots” image set shows that almost all the algorithms are sensitive to noise. The bottom row of this image set indicates that about 40-50% of the algorithms have problems in detecting the correct disparity for the background objects while 70-80% of the algorithms struggle in detecting correct disparity values for the foreground objects in presence of noise. The performance of these algorithms is challenged in terms of processing convex, concave, rounded and planar geometry in the "Pyramids" image set.

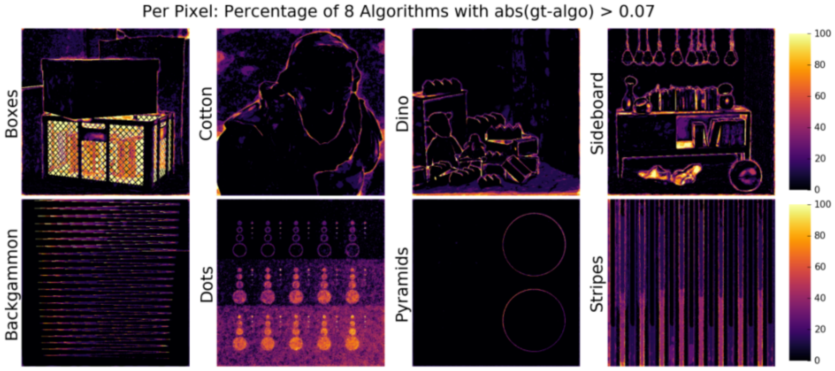


Fig 8 Scene difficulties visualized as heatmaps.

Fig. 9 and Fig. 10 illustrate the disparity maps, ground truth error map and the median error map of the studied algorithms for Stratified and Training sets, respectively. Each row in these figures represents an algorithm. For each algorithm per individual image set, there are three columns illustrating the disparity maps, ground truth error map and the median error map. To generate the median error map, the median of the absolute disparity differences of all algorithms with the ground truth is computed for each pixel. Further, the absolute disparity difference of each algorithm is subtracted from the median error. The median error map gives a conceptual understanding of the parts of the image where algorithms perform below or above average performance of all algorithms. Yellow parts in this map represent the average, green above-average and red below-average performance.

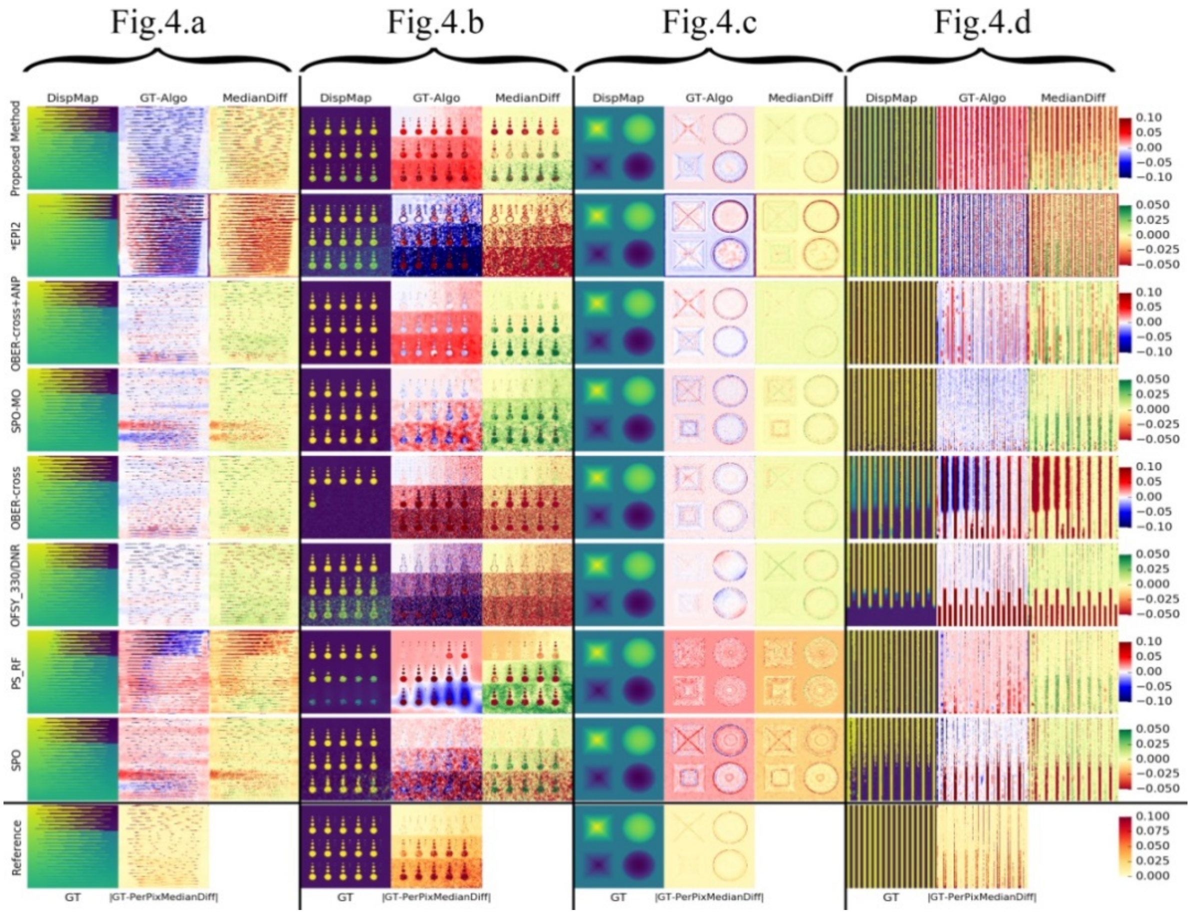


Fig 9 Stratified Scenes-Visualizations for disparity maps and their differences with ground truth. Each row represents an algorithm. The first column for each stratified scenes illustrates the disparity maps of the proposed method and the studied algorithms. The second column illustrates the disparity difference to the ground truth. Highly accurate parts are shown in white, too close in blue and too far in red areas. The third column illustrates how algorithms perform relative to the median algorithm performance. Yellow parts show average, green above-average and red below-average performance. The last row of the figure illustrates the ground truth disparity maps and the median absolute disparity difference to the ground truth at each individual pixel among all algorithms.

The median error maps of the proposed method in Fig. 9 indicate its close performance to the average performance of all algorithms in well-structured scenes, complex occlusions and different noise levels. The same maps in Fig. 10 show how competitive the proposed method performs compared to the other algorithms while dealing with slanted planar surfaces and complex scene structure. However, there are still highly textured areas with fine patterns such as box frames in the "Boxes" image set which introduces many challenges to depth estimation algorithms. For instance, OBER-cross+ANP and OBER-cross algorithms estimated the disparity of the pixels beyond the box frames in “Boxes” image set above-average of the median algorithm. On the other side, EPI2 estimated the disparity level for the same area highly below-average of the median algorithm.

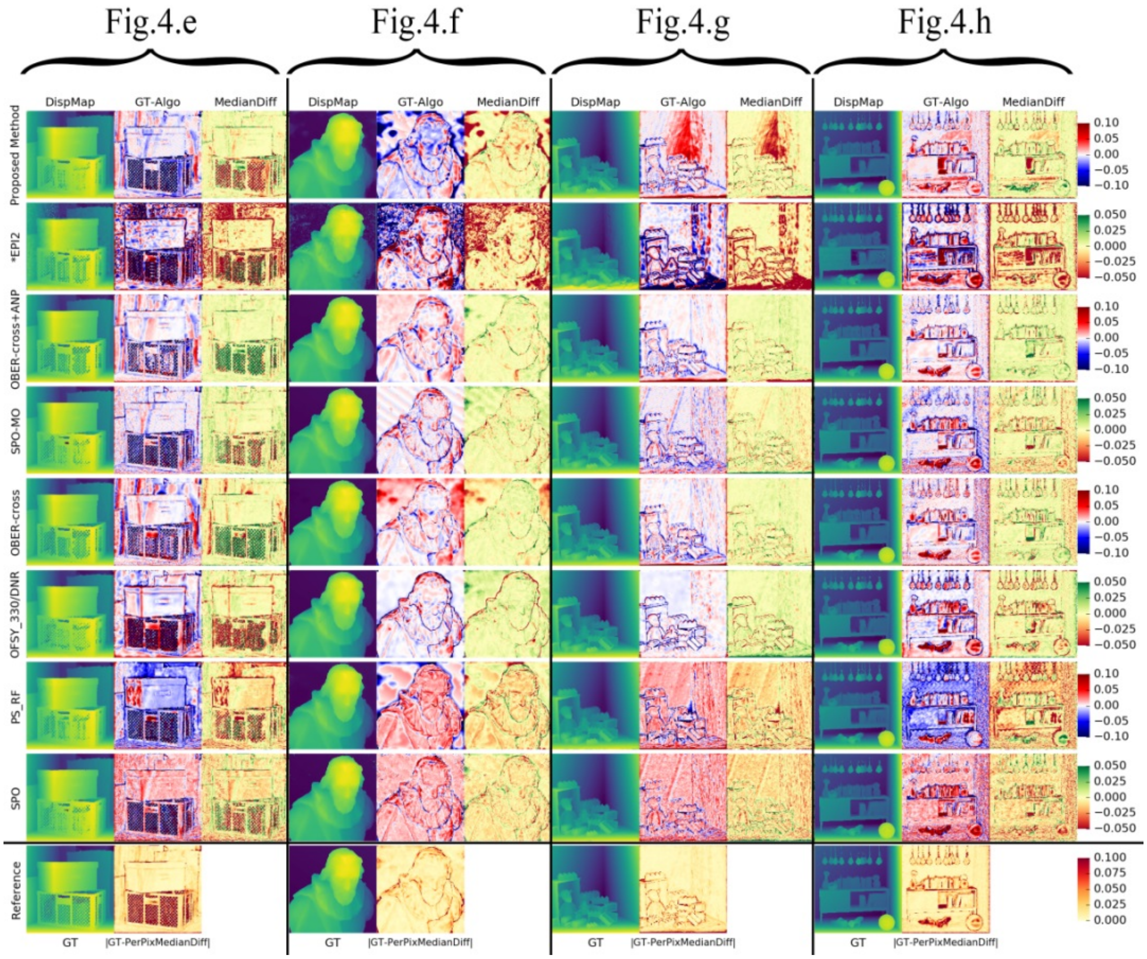


Fig 10 Training scenes-Visualizations for disparity maps and their differences with ground truth. Each row represents an algorithm. The first column for each training scenes illustrates the disparity maps of the proposed method and the studied algorithms. The second column illustrates the disparity difference to the ground truth. Highly accurate parts are shown in white, too close in blue and too far in red areas. The third column illustrates how algorithms perform relative to the median algorithm performance. Yellow parts show average, green above-average and red below-average performance. The last row of the figure illustrates the ground truth disparity maps and the median absolute disparity difference to the ground truth at each individual pixel among all algorithms.

Table 3 and Fig. 11 present the average performance time of the proposed method compared to the studied algorithms in non-logarithmic and logarithmic scale mode, respectively. The computational times for each of these algorithms have been reported by their authors.

A faster performance time and the competitive accuracy of the proposed method make it applicable for deployment in practical embedded systems and Internet-of-Things (IoT) appliances. Using the method proposed in this paper, one could transmit a compressed depth map in an IoT device, rather than the full image stream or analyze the depth map at the edge level and use it to trigger corresponding actions [57]. Table 4 shows how much faster/slower and more/less accurate the proposed method is compared to the other algorithms. For example, the proposed method is ~4.5 times slower than the baseline algorithm EPI2; however, the accuracy of the estimated depth maps has increased ~21% or the proposed method is ~111.8 times faster than SPO-MO but its accuracy decreased ~2.3%.

The estimations in Table 4 are based on the percentage of the pixels with correct disparity values above 0.07 error threshold. Note that, the same metric is initially used to choose these algorithms for comparison purposes.

Table 3 Computational time of the proposed framework and the state of the art in seconds.

Algorithms	EPI2	**Proposed Method**	OBER-cross+ANP	SPO-MO	OBER-cross	OFSY_330/DNR	PS_RF	SPO
Time (s)	8.4	**38.5**	182.9	4304.3	96.4	200.2	1412.6	2115.4

Table 4 Comparison of the proposed method and the state of the art. Factors of computational time improvement and percentage of disparity accuracy variation.

	EPI2	**OBER-cross+ANP**	**SPO-MO**	**OBER-cross**	**OFSY_330/DNR**	**PS_RF**	**SPO**
Proposed Method vs.	~4.5× slower ~21% inc.	~4.7× faster ~3.5% dec.	~111.8× faster ~2.3% dec.	~2.5× faster ~4.95% inc.	~5.2× faster ~6% inc.	~36.7× faster ~0.01% dec.	~54.9× faster ~1.43% inc.

* inc: Increased accuracy dec: Decreased accuracy

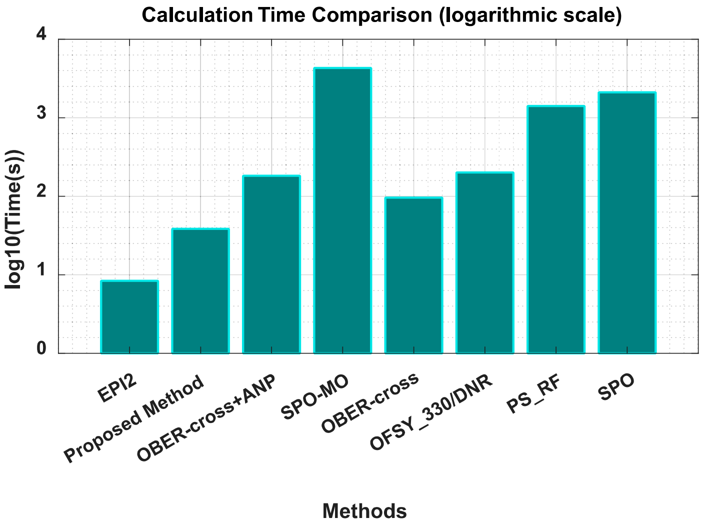


Fig 11 Computational time of the proposed framework and the state of the art in logarithmic scale.

5 Conclusion and Discussion

In this paper, a new framework is proposed based on EPI analysis and TV minimization to estimate depth from the multi-camera array. A new cost function is proposed and analyzed based on Fenchel-Rockafellar duality [18]. Our approach consists of two steps. First, a rough initialization of the depth map is computed using local depth estimation on EPIs. Later, this initialization is refined by applying a TV minimization based on Fenchel-Rockafellar duality [18].

We demonstrate the benefits of the proposed framework on a synthetic dataset including Stratified and photorealistic light field image sets. The method has been implemented in Matlab R2017a on a device equipped with Intel i7-5600U @ 2.60GHz CPU and 16 GB RAM.

The evaluation reveals that most algorithms consist of multiple elements and terms which make it difficult to establish one best algorithm that outperforms in all categories. Also, the high computational time of the studied methods makes almost inapplicable for consumer devices. The results demonstrate the competitive performance of the proposed framework among the top state of the art methods in terms of accuracy of depth estimation. Even though the accuracy of the estimated depth maps using proposed framework varies based on each metric, it still remains in the list of high accuracy methods and the fast convergence of the proposed cost function and its fast computational time make it a potential method for consumer electronics applications and devices with the aid of parallel technology and GPUs. The new generation of GPUs contains a high number of programmable parallel cores (up to 4k). This evolution makes this technology an efficient choice for computationally intensive processes in machine vision applications such as depth estimation. We aim to explore the effect of parallelism on depth estimation from the multi-camera array in the future works

Acknowledgments

The research work presented here was funded under the Strategic Partnership Program of Science Foundation Ireland (SFI) and co-funded by SFI and FotoNation Ltd. Project ID: 13/SPP/I2868 on "*Next Generation Imaging for Smartphone and Embedded Platforms*".

Appendix 1:

Fig. A1 presents the percentage of pixels with correct disparity on the Stratified and Training scenes for the increasing error thresholds. The PerPixBest [56] in Fig. A1 is an artificial algorithm made for evaluation purposes. “The lowest absolute disparity difference to the ground truth at each individual pixel among all algorithms” [56] is used to create the PerPixBest metric.

The algorithms ranking change by varying the thresholds for absolute disparity error. The difference in performance of the algorithms for high error thresholds is relatively small. The lower thresholds show a more apparent difference in the performances and the ranking of the algorithms change significantly for the thresholds between 0.010 and 0.032. Despite the weak tolerance in the performance of the proposed method in lower thresholds between 0 and 0.035, it is ranked among the top three methods from the threshold 0.048 onwards. OFSY_330/DNR has the best performance in lower error thresholds between 0 and 0.012 and its performance reduces for the thresholds higher than 0.012. OBER-cross+ANP has the second best performance up to the threshold 0.022 and it achieves the best performance from the error threshold 0.034 onward while competing very closely with SPO-MO.

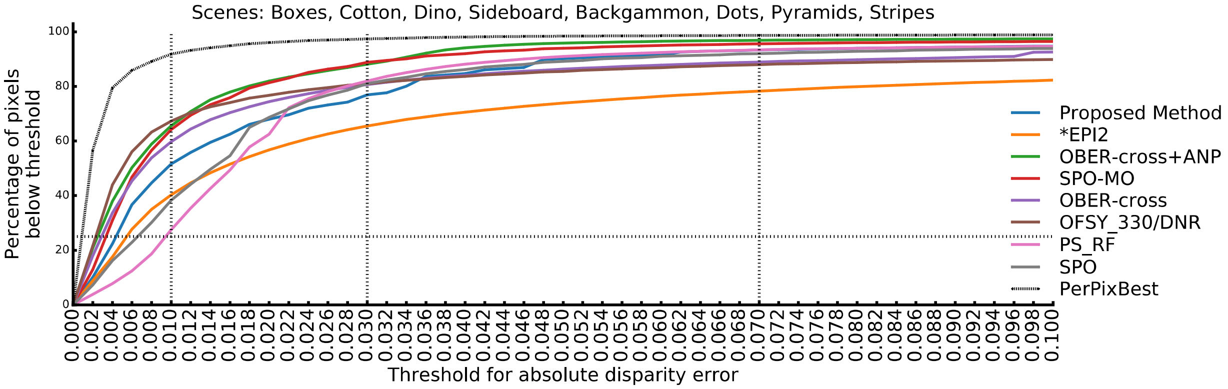


Fig A1 The percentage of pixels with correct disparity on the photorealistic and non- photorealistic scenes with increasing error thresholds.

Fig. A2 illustrate the analysis of the 3D models for the “Cotton” image set in Fig.4.f, generated based on the depth maps from the proposed framework, the ground truth and OBER-cross+ANP which is the best algorithm ranked in the benchmark.

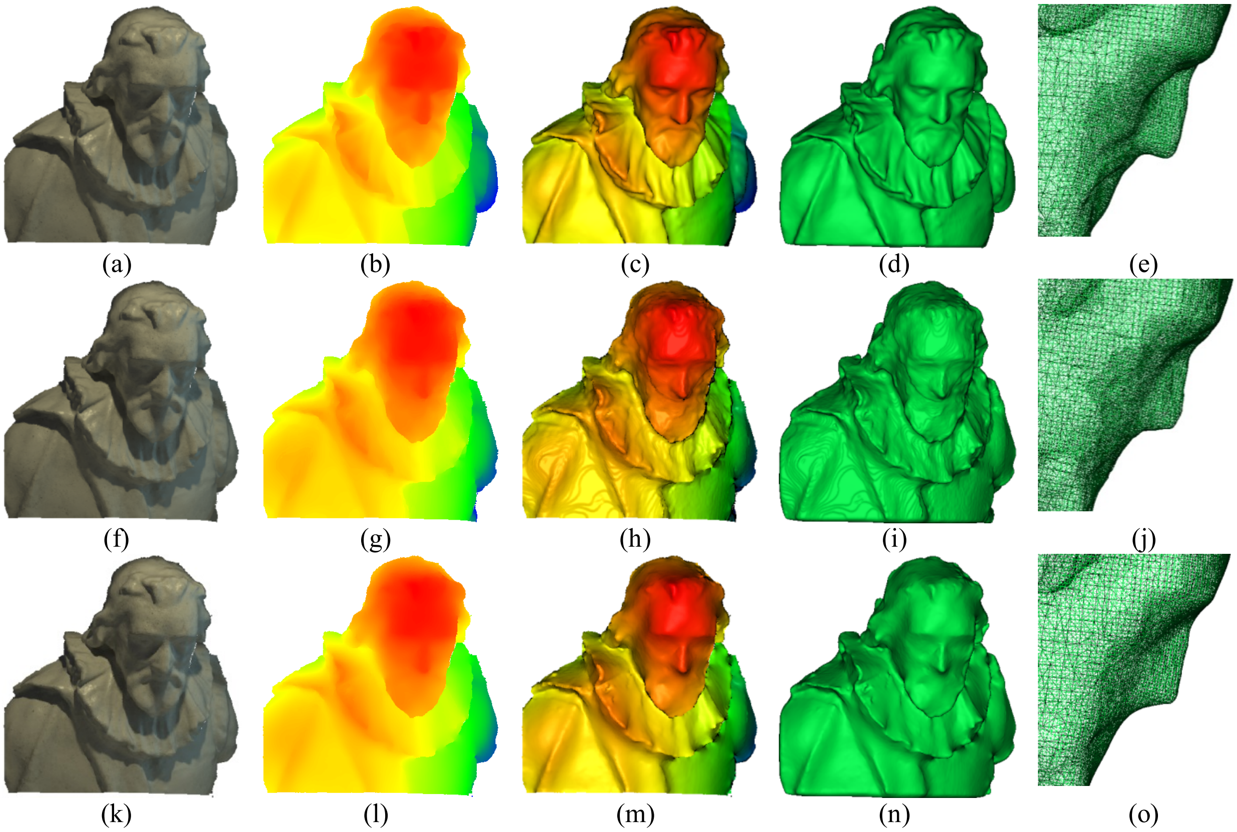


Fig A2 3D visualizations of the “Cotton” image set for proposed method, ground truth and OBER-cross+ANP algorithm. First row shows the 3D models based on the ground truth, the second row presents the model generated based on the proposed method and the third row illustrates the 3D models generated based on OBER-cross+ANP algorithm. First column shows the 3D color mesh, second column shows the rasterized 3D color-coded depth, third column shows the 3D normals, fourth column shows the Poisson surface reconstruction and last column shows the wireframe model of the reconstructed surface for the area around the face. (a) GT–3D color mesh. (b) GT–Rasterized 3D color-coded depth. (c) GT–3D Normals. (d) GT–Poisson surface reconstruction. (e) GT–wireframe model of the reconstructed surface for the face area. (f) Proposed method–3D color mesh. (g) Proposed method–Rasterized 3D color-coded depth. (h) Proposed method–3D Normals. (i) Proposed method–Poisson surface reconstruction. (j) Proposed method–wireframe model of the reconstructed surface for the face area. (k) OBER-cross+ANP–3D color mesh. (l) OBER-cross+ANP–Rasterized 3D color-coded depth. (m) OBER-cross+ANP–3D Normals. (n) OBER-cross+ANP–Poisson surface reconstruction. (o) OBER-cross+ANP–wireframe model of the reconstructed surface for the face area.

The purpose of this comparison is to find out how accurate and close the 3D reconstructed data from the estimated depth map is to the ground truth and the most accurate method in the state of the art. Fig.A2.b, Fig.A2.g and Fig.A2.l represent the rasterized color-coded 3D model from the ground truth, proposed method and OBER-cross+ANP, respectively. The color-coded model indicates how close the proposed method is in terms of establishing depth levels to ground truth and OBER-cross+ANP. The transition from red to blue present the area which are closer and far from the camera. By looking at 3D normals in Fig.A2.c, Fig.A2.h and Fig.A2.m one can determine the smoothness of the disparity values estimated by the proposed framework in comparison to the ground truth and OBER-cross+ANP. Note that the visible line artifacts in Fig.A2.h are the boundaries of each individual depth plane which is the result of the discrete calculation. This issue can be simply solved by applying 3D inpainting methods [58] to make the surface continuous. The Poisson surface reconstruction [59] and the wireframe model which are shown in Fig.A2.i and Fig.A2.j, outline the capability of the proposed framework in dealing with non-uniform surfaces and following fine structures.

References

1. The Lytro Camera. https://www.lytro.com/.
2. Raytrix. 3D Light Field Camera Technology. http://raytrix.de/.
3. Venkataraman K, Lelescu D, Duparré J, et al. *PiCam*: an ultra-thin high performance monolithic camera array. *ACM Trans Graph.* 2013;32(6):1-13.
4. Attar Z, Aharon-Attar C, Wolterink EM, Inventors; Google Patents, assignee. System and Method for Imaging and Image Processing. U.S. Patent Application 13/881,039, Filed April 24, 2011.
5. Tanida J, Kumagai T, Yamada K, et al. Thin observation module by bound optics (TOMBO): concept and experimentalverification. *Appl Opt.* 2001;40(11):1806-1813.
6. Li N, Ye J, Ji Y, Ling H, Yu J. Saliency Detection on Light Field. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 2017;39(8):1605-1616.
7. Chen C, Lin H, Yu Z, Kang SB, Yu J. Light Field Stereo Matching Using Bilateral Statistics of Surface Cameras. Paper presented at: 2014 IEEE Conference on Computer Vision and Pattern Recognition; 23-28 June 2014, 2014.
8. Jeon HG, Park J, Choe G, et al. Accurate depth map estimation from a lenslet light field camera. Paper presented at: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 7-12 June 2015, 2015.
9. Lin H, Chen C, Kang SB, Yu J. Depth Recovery from Light Field Using Focal Stack Symmetry. Paper presented at: 2015 IEEE International Conference on Computer Vision (ICCV); 7-13 Dec. 2015, 2015.
10. Tao MW, Hadap S, Malik J, Ramamoorthi R. Depth from Combining Defocus and Correspondence Using Light-Field Cameras. Paper presented at: 2013 IEEE International Conference on Computer Vision; 1-8 Dec. 2013, 2013.

11. Tao MW, Srinivasan PP, Malik J, Rusinkiewicz S, Ramamoorthi R. Depth from shading, defocus, and correspondence using light-field angular coherence. Paper presented at: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 7-12 June 2015, 2015.

12. Ng R, Levoy M, Brédif M, Duval G, Horowitz M, Hanrahan P. Light field photography with a hand-held plenoptic camera. *Computer Science Technical Report CSTR.* 2005;2(11):1-11.

13. Wang X, Li L, Hou G. High-resolution light field reconstruction using a hybrid imaging system. *Appl Opt.* 2016;55(10):2580-2593.

14. Bishop TE, Favaro P. The Light Field Camera: Extended Depth of Field, Aliasing, and Superresolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 2012;34(5):972-986.

15. Kim C, Zimmer H, Pritch Y, Sorkine-Hornung A, Gross M. Scene reconstruction from high spatio-angular resolution light fields. *ACM Trans Graph.* 2013;32(4):1-12.

16. Tao MW. *Unified Multi-Cue Depth Estimation from Light-Field Images: Correspondence, Defocus, Shading, and Specularity.* University of California, Berkeley; 2015.

17. Bolles RC, Baker HH, Marimont DH. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision.* 1987;1(1):7-55.

18. Rockafellar RT. *Convex Analysis.* Princeton University Press; 1970.

19. Levoy M, Hanrahan P. Light field rendering. Proceedings of the 23rd annual conference on Computer graphics and interactive techniques; 1996.

20. Komodakis N, Pesquet JC. Playing with Duality: An overview of recent primal?dual approaches for solving large-scale optimization problems. *IEEE Signal Processing Magazine.* 2015;32(6):31-54.

21. Rockafellar RT. Conjugate convex functions in optimal control and the calculus of variations. *Journal of Mathematical Analysis and Applications.* 1970;32(1):174-222.

22. Yu Z, Guo X, Ling H, Lumsdaine A, Yu J. Line Assisted Light Field Triangulation and Stereo Matching. Paper presented at: 2013 IEEE International Conference on Computer Vision; 1-8 Dec. 2013, 2013.

23. Wanner S, Goldluecke B. Globally consistent depth labeling of 4D light fields. Paper presented at: 2012 IEEE Conference on Computer Vision and Pattern Recognition; 16-21 June 2012, 2012.

24. Wanner S, Goldluecke B. Variational Light Field Analysis for Disparity Estimation and Super-Resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 2014;36(3):606-619.

25. Zhang Y, Lv H, Liu Y, et al. Light-Field Depth Estimation via Epipolar Plane Image Analysis and Locally Linear Embedding. *IEEE Transactions on Circuits and Systems for Video Technology.* 2017;27(4):739-747.

26. Ma Z, Cen Z, Li X. Depth estimation algorithm for light field data by epipolar image analysis and region interpolation. *Appl Opt.* 2017;56(23):6603-6610.

27. Yang P, Wang Z, Yan Y, et al. Close-range photogrammetry with light field camera: from disparity map to absolute distance. *Appl Opt.* 2016;55(27):7477-7486.

28. Zhang S, Sheng H, Li C, Zhang J, Xiong Z. Robust depth estimation for light field via spinning parallelogram operator. *Computer Vision and Image Understanding.* 2016;145(Supplement C):148-159.

29. Wang TC, Efros AA, Ramamoorthi R. Depth Estimation with Occlusion Modeling Using Light-Field Cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 2016;38(11):2170-2181.

30. Williem W, Park IK. Robust Light Field Depth Estimation for Noisy Scene with Occlusion. Paper presented at: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 27-30 June 2016, 2016.

31. Liu C, Qiu J, Zhao S. Iterative reconstruction of scene depth with fidelity based on light field data. *Appl Opt.* 2017;56(11):3185-3192.

32. Monteiro NB, Barreto JP, Gaspar J. Dense Lightfield Disparity Estimation Using Total Variation Regularization. In: Campilho A, Karray F, eds. *Image Analysis and Recognition: 13th International Conference, ICIAR 2016, in Memory of Mohamed Kamel, Póvoa de Varzim, Portugal, July 13-15, 2016, Proceedings.* Cham: Springer International Publishing; 2016:462-469.

33. Kim C, Hornung A, Heinzle S, Matusik W, Gross M. Multi-perspective stereoscopy from light fields. *ACM Trans Graph.* 2011;30(6):1-10.

34. Basha T, Avidan S, Hornung A, Matusik W. Structure and motion from scene registration. Paper presented at: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on2012.

35. Navarro J, Buades A. Robust and Dense Depth Estimation for Light Field Images. *IEEE Transactions on Image Processing.* 2017;26(4):1873-1886.

36. Buades A, Facciolo G. Reliable Multiscale and Multiwindow Stereo Matching. *SIAM Journal on Imaging Sciences.* 2015;8(2):888-915.

37. Brox T, Bruhn A, Papenberg N, Weickert J. High Accuracy Optical Flow Estimation Based on a Theory for Warping. In: Pajdla T, Matas J, eds. *Computer Vision - ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part IV.* Berlin, Heidelberg: Springer Berlin Heidelberg; 2004:25-36.

38. Pérez F, Pérez A, Rodríguez M, Magdaleno E. A fast and memory-efficient Discrete Focal Stack Transform for plenoptic sensors. *Digital Signal Processing.* 2015;38(Supplement C):95-105.

39. Pérez F, Pérez A, Rodríguez M, Magdaleno E. Lightfield Recovery from Its Focal Stack. *Journal of Mathematical Imaging and Vision.* 2016;56(3):573-590.

40. Mousnier A, Vural E, Guillemot C. Partial light field tomographic reconstruction from a fixed-camera focal stack. *arXiv preprint arXiv:150301903.* 2015.

41. Lee JY, Park R-H. Depth Estimation from Light Field by Accumulating Binary Maps Based on Foreground-Background Separation. *IEEE Journal of Selected Topics in Signal Processing.* 2017.

42. Lee JY, Park R-H. Separation of foreground and background from light field using gradient information. *Appl Opt.* 2017;56(4):1069-1078.

43. Diebold M, Blum O, Gutsche M, et al. Light-field camera design for high-accuracy depth estimation. Paper presented at: SPIE Optical Metrology2015.

44. Berent J, Dragotti PL. Segmentation of Epipolar-Plane Image Volumes with Occlusion and Disocclusion Competition. Paper presented at: 2006 IEEE Workshop on Multimedia Signal Processing; 3-6 Oct. 2006, 2006.

45. Lin P-H, Yeh J-S, Wu F-C, Chuang Y-Y. Depth Estimation for Lytro Images by Adaptive Window Matching on EPI. *Journal of Imaging.* 2017;3(2):17.

46. Gilboa G, Osher S. Nonlocal linear image regularization and supervised segmentation. *Multiscale Modeling & Simulation.* 2007;6(2):595-630.

47. Chan TF, Golub GH, Mulet P. A nonlinear primal-dual method for total variation-based image restoration. *SIAM journal on scientific computing.* 1999;20(6):1964-1977.

48. Elmoataz A, Lezoray O, Bougleux S. Nonlocal Discrete Regularization on Weighted Graphs: A Framework for Image and Manifold Processing. *IEEE Transactions on Image Processing.* 2008;17(7):1047-1060.

49. Bougleux S, Elmoataz A, Melkemi M. Discrete Regularization on Weighted Graphs for Image and Mesh Filtering. In: Sgallari F, Murli A, Paragios N, eds. *Scale Space and Variational Methods in Computer Vision: First International Conference, SSVM 2007, Ischia, Italy, May 30 - June 2, 2007. Proceedings.* Berlin, Heidelberg: Springer Berlin Heidelberg; 2007:128-139.

50. Combettes PL, Pesquet J-C. Proximal Splitting Methods in Signal Processing. In: Bauschke HH, Burachik RS, Combettes PL, Elser V, Luke DR, Wolkowicz H, eds. *Fixed-Point Algorithms for*

Inverse Problems in Science and Engineering. New York, NY: Springer New York; 2011:185-212.

51. Honauer K, Johannsen O, Kondermann D, Goldluecke B. A Dataset and Evaluation Methodology for Depth Estimation on 4D Light Fields. In: Lai S-H, Lepetit V, Nishino K, Sato Y, eds. *Computer Vision – ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part III.* Cham: Springer International Publishing; 2017:19-34.

52. Paige CC, Saunders MA. Solution of Sparse Indefinite Systems of Linear Equations. *SIAM Journal on Numerical Analysis.* 1975;12(4):617-629.

53. Barrett R, Berry M, Chan T, et al. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods.* Society for Industrial and Applied Mathematics; 1994.

54. Honauer K, Johannsen O, Kondermann D, Goldluecke B. 4D Light Field Dataset. 2016; http://hci-lightfield.iwr.uni-heidelberg.de/.

55. Strecke M, Alperovich A, Goldluecke B. Accurate Depth and Normal Maps From Occlusion-Aware Focal Stack Symmetry. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR); July 2017, 2017.

56. Johannsen O, Honauer K, Goldluecke B, et al. A Taxonomy and Evaluation of Dense Light Field Depth Estimation Algorithms. Paper presented at: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW); 21-26 July 2017, 2017.

57. Corcoran P. Beyond stream processing - A distributed vision architecture for the Internet of Things. Paper presented at: 2016 IEEE International Conference on Consumer Electronics (ICCE); 7-11 Jan. 2016, 2016.

58. Caselles V, Haro G, Sapiro G, Verdera J. On geometric variational models for inpainting surface holes. *Computer Vision and Image Understanding.* 2008;111(3):351-373.

59. Kazhdan M, Bolitho M, Hoppe H. Poisson surface reconstruction. Proceedings of the fourth Eurographics symposium on Geometry processing; 2006; Cagliari, Sardinia, Italy.