



Provided by the author(s) and NUI Galway in accordance with publisher policies. Please cite the published version when available.

Title	An FPGA-based stereoscopic camera - electronic design tools and techniques
Author(s)	Andorko, Istvan; Corcoran, Peter; Bigioi, Petronel
Publication Date	2010-08-05
Publication Information	Andorko, I., Corcoran, P., & Bigioi, P. An FPGA-based stereoscopic camera - electronic design tools and techniques. Paper presented at the Roedunet International Conference (RoEduNet), 2010 9th.
Publisher	IEEE
Link to publisher's version	http://www.ieeexplore.ieee.org/search/srchabstract.jsp?tp=&arnumber=5541604
Item record	http://hdl.handle.net/10379/1438

Downloaded 2017-10-18T02:02:34Z

Some rights reserved. For more information, please see the item record link above.



An FPGA-based Stereoscopic Camera - Electronic Design Tools and Techniques

Istvan Andorko, *Student Member IEEE* and Peter Corcoran, *Fellow IEEE*
College of Engineering and Informatics
National University of Ireland, Galway
(NUIG)
i.andorko1@nuigalway.ie; peter.corcoran@nuigalway.ie

Petronel Bigioi, *Senior Member IEEE*
Tessera Ireland; College of Engineering and Informatics
National University of Ireland, Galway
Galway, Ireland
pbigioi@tessera.com; petronel.bigioi@nuigalway.ie

Abstract – Electronic design tools and techniques for the implementation of a stereoscopic camera based on an FPGA (Field Programmable Gate Array) are presented. The stages of an IPP (Image Processing Pipeline) are presented together with the development tools and languages used to implement a stereoscopic camera in hardware. In a further development of the basic system, aspects of the implementation of a 3D camera are presented.

Keywords – stereoscopic camera, FPGA, 3D imaging, embedded development tools

I. INTRODUCTION

Recently there has been a great deal of interest in 3D video and imaging applications driven by a combination of the recent successes of 3D Hollywood movies and the introduction of new 3D display technologies for flat-screen TVs [1].

Of course 3D cinema and imaging applications have been available for many years [2]. Nevertheless there have been significant improvements and commoditization of the underlying display technologies. As 3D displays become increasingly available to the public we can see new consumer needs arising, in particular a requirement for consumer imaging devices which can capture 3D compatible images and video sequences.

Modern electronic systems based on FPGA arrays are now sufficiently powerful to implement an entire image processing pipeline (IPP) within such a device [3]. We are currently working on a project to implement a dual-IPP which enables two images of the same camera scene to be captured at the same time. Amongst the potential applications for such a dual-imaging system it provides a highly flexible enabling technology for 3D imaging.

Some details of our underlying work have been presented elsewhere [4, 5, 6] but in this paper we wish to focus on the electronic design tools and methods that we have used to realize our dual-IPP.

This paper is organized as follows: firstly, in section II we provide some background information on the main elements of a typical IPP from a conventional digital camera. We then review 3D imaging techniques in section III. In the first part of

section IV our core design tools are introduced and a number of examples of the design process are given. In the second part of section IV we discuss about the implementation aspects of a stereo image processing pipeline... Finally in section V we discuss some aspects of our 3D camera design and provide some pointers for other researchers who are interested in building their own 3D imaging devices.

II. IMAGE PROCESSING PIPELINE

The color spectral response of the image sensor needs to match that of a typical human eye, as defined by Commission Internationale de l'Eclairage (CIE) [7]. Real image sensors, however cannot meet this requirement. This is why it is required to do reproducing and enhancing processing to the acquired image such as color interpolation, white balancing, color correction, gamma correction and color conversion [3].

A. Color Interpolation

In color imaging, charge-coupled device (CCD) and complementary metal oxide semiconductor (CMOS) image sensors are covered with a color filter array (CFA) that samples only one color per clock cycle. Because an image pixel consists of three color components red, green and blue (R, G, B), we need to use color interpolation to define the missing color component from each pixel [8]. There are a number of methods available for color interpolation or demosaicking. The simplest method of interpolation is the ideal interpolation. The second interpolation strategy is based on neighborhood considerations where it can be expected to get better estimates for the missing sample values by increasing the neighborhood of the pixel, but this way the computation cost increases too. The third available strategy is the bilinear interpolation. The fourth method is the constant hue-based interpolation. This is one of the first methods used in commercial cameras [9]. An example of the Bayer CFA can be found in figure 1.

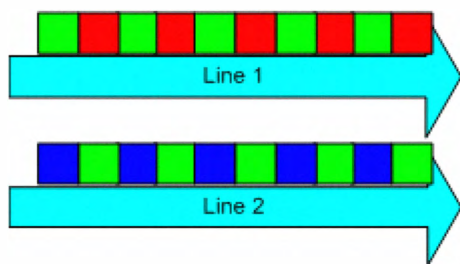


Figure 1. Bayer CFA

B. Automatic White Balance

The aim of the auto white balance is to guess the illumination under which the image is taken and compensate the color shift affected by the illuminate [7]. The white balance problem is usually solved by adjusting the gains of the three primary colors R, G or B of the sensors to make a white object to appear as white under different illuminants. There are a number of methods that were proposed for the white balance operation and we are going to present some of the most important ones.

The first group of methods would be the “Gray World Methods”. These algorithms simply assume that the scene average is identical to the camera response to the chosen “gray” under the scene illuminant. [10].

The second method is the “Illuminant Estimation by the Maximum of Each Channel”. This algorithm estimates the illuminant (R, G, and B) by the maximum response in each channel [10].

The third group of methods is the “Gamut Mapping methods”. These are based on Forsyth’s gamut-mapping approach [11].

The fourth method is the “Color by Correlation”. The basic idea of this approach is to pre-compute a correlation matrix which describes the extent to which proposed illuminants are compatible with the occurrence of image chromaticity’s. [10].

The fifth type of method is the “Neural Net” method. The neural net is a multilayer Perception with two hidden layers. The general structure is pyramidal. In one of the examples the input layer consists of 2500 nodes, the first hidden layer has 400 nodes, the second hidden layer 30 nodes and the output layer has 2 nodes. The chromaticity space is divided into discrete bins. The input to each neuron is a binary value representing the presence or absence of a scene chromaticity falling in the corresponding bin. Thus, a histogram of the image is formed and then the histogram is binaries [10].

C. Color correction

Accurate color reproduction can be a challenge due to the fact that the images captured by the digital cameras are affected by many environmental contents such as illumination, objective settings and the color properties of the object. Therefore we need a transformation which maps the captured RGB colors to correct tri-stimulus values in the device-

independent color space. These are also known as camera characterization [12]. The most common solution for a color correction system is the color compensation chart. The color correction framework consists of the following stages. The first one is the brightness compensation that flattens the brightness of every color sample.

The next stage is the estimation of the tone reproduction curve. This is the process of maintaining the device with a fixed characteristic color response. After the calibration, the raw image data obtained directly from the sensor has linear characteristic.

The third step is the color correction performed in the CIELAB color space. During this stage, the RGB values are transformed into the device independent colors of the profile connection space [13]. The final stage is the encoding of the relationship between the captured color and the reference data in a color profile. Once the profile is created, it is then possible to perform the color correction of other images [12].

D. Gamma Correction

Real-time gamma correction is an essential function in display devices such as CRT, plasma and TFT LCDs. The Gamma Correction controls the overall brightness of the images. The images that are not appropriately revised can look pale or too dark. The gamma correction not only changes the brightness of the images, but also can change the ratios of the RGB [14]. The term gamma is a constant which is related with the quality of the monitors from the relationship between the brightness and the input voltage of the monitors. More exactly the value of gamma is determined experimentally by the characteristics of the display [14].

III. 3D IMAGING

Similarly with the Image Processing Pipeline, the 3D image acquisition and display process has its own well defined steps. These steps are the acquisition of 3D images, the coding and transmission of the images and finally the display of the 3D images which is one of the most important aspects of this process. The depth perception of a scenery can be provided by the systems that ensure that the user sees a specific different view with each eye [15].

A. 3D image acquisition

The 3D image acquisition can be made on two ways. One way is to take 2D images and convert them into 3D images using different tools. This approach could be computationally expensive but is cheaper. The second approach is to acquire 3D images using cameras specially designed for this task. The cameras designed to acquire 3D images in most of the cases have two image sensors placed one next to the other. In some rare cases, these cameras have more than 2 image sensors placed one next to the other.

The first 3D images were made with the help of the anaglyph cameras. These cameras had lenses placed to a specific distance from each other. This distance is the distance between the eyes of a person. The camera lenses were special lenses in the way that one of them only allowed the Red (R)

component of an image to go through whereas the other only allowed the Blue (B) component of the image to go through. After this initial processing, the images were placed overlapped into the same frame. An example of an anaglyph image is presented in figure 2.

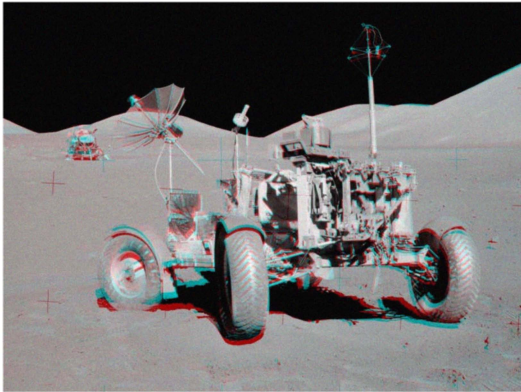


Figure 2. Example of anaglyph image

Nowadays, 3D images are acquired using 2 or more standard image sensors placed one next to the other. We have two IPPs delivering information into the processing device. From here on, it's up to the processing module to generate viewable 3D images.

Another approach is to generate 3D images using 2D images and depth maps of those specific images. One of these approaches can be found in Pourazad et al. [16] where a 2D to 3D video conversion scheme is presented. This method utilizes the motion information between consecutive frames to approximate the depth map of a scene. Another approach is presented in Cheng et al. [17] where they convert 2D images into 3D based on edge information. The system groups the blocks into regions using edge information. A prior hypothesis of depth gradient is used to assign depth of regions. Then a bilateral filter is used to diminish the block effect and to generate depth map for 3D.

B. 3D image coding and transmission

There are no 3D image coding standards available yet, but MPEG are working on it forced by the industry. Although, some of the standards used in 2D imaging, are starting to be used in the coding of 3D video as well [18].

Digital cable systems deliver content to a wide range of audiences. They use MPEG2 transport over QAM to carry video streams encoded as MPEG2, H.264/AVC or as VC-1 [19]. There are nine choices for spatial resolution, six choices for frame rates, two aspect ratios and either progressive or interlaced scanning [19]. Today, the cable systems deliver stereoscopic 3D content using various anaglyph coding [20], but this is not enough to deliver high quality 3D videos for home entertainment. The need exists for the development of other types of transmission. The delivery of 3D data can be done over the existing infrastructure or through the gradual deployment of new hardware and infrastructure. However, the new hardware and infrastructure needs to be compatible with the existing one, so this limits the range of new possible

services. The left and right images can be frame-packed into a single video frame and delivered using existing MPEG encoders and systems as if it were conventional 2D signals. The frame-packing can take various forms like side-by-side, top-bottom, line-interleaved, frame-interleaved, column-interleaved, checkerboard-interleaved and others. The receiver decodes these frames and sends the images to the display system accordingly [20].

C. 3D image display

There are a variety of display technologies available on the market at the moment that support 3D video, each having different input requirements and each having its own technology and offering different viewing experience. To realize high quality auto-stereoscopic displays, multiple views of the video must either be provided as input to the display, or these views must be created locally at the display [15]. The condition for the user to be able to see the 3D images is that he sees the corresponding left and right views with each eye. There are two main categories of displays for 3D images. The first one uses a standard display system, which with the help of special glasses (anaglyph, polarized or shutter) produces the 3D effect. The second types of displays are autostereoscopic displays which can produce the 3D effect without the help of special glasses, but they have certain disadvantages that we are going to present.

The very first method used in theatres was the anaglyph method. This required the display to show the anaglyph image and the user to wear special anaglyph glasses. The anaglyph glasses have different types of lenses. One of them is red and it allows only the red (R) component of the image to go through, the other one is blue (B) or cyan (B + G) and it allows only the blue or cyan component to go through. This way the user sees different views and this creates the desired 3D effect.

The second type of glasses is the polarized glasses. These types of glasses are mainly used in the modern theatres and they provide high quality images. Actually, the newly developed polarized glasses have set the new trend for 3D movies. Polarized glasses permit the kind of full-color reproduction not possible with anaglyph systems. In a typical setup, two projectors are required with different polarized filters and a special nondepolarizing screen is required to ensure that polarization is maintained during projection [2].

The third types of glasses used in 3D technology are the shutter glasses. These are mainly used in the home entertainment industry. This is one of the main reasons why we have chosen this type of display for our system. The purpose of these glasses is the same as in the case of other two, to provide different perspectives of the same image and this way to generate the depth sensation. In this case, light blocking is applied with the help of fast switching lenses which are synchronized with the vertical synchronization signal of the screen and become opaque when the unintended view is rendered on the screen [2]. In this case, to avoid the flickering sensation, the refresh rate of the display needs to be above 120 Hz. The newly developed technologies as the one presented by Kawahara et al in [1] allow the transmission and

display of Full HD images in 3D and this is what makes this approach the best candidate for the future home entertainment 3D systems.

The second main category of display devices are the autostereoscopic devices. In this case there is no need to use any kind of special glasses. This approach is based on the spatial multiplexing of left and right images in combination with a light-directing mechanism that presents those views to the viewer's eyes [21]. The two main multiplexing techniques are the parallax barrier displays and the microlens displays [2, 22]. These techniques suffer from the loss of horizontal resolution, half of the pixels in a horizontal row being delivered to the left eye and half to the right [2, 23].

IV. VERILOG HDL AND XILINX DEVELOPMENT TOOLS

A. Description of the development tools

The VerilogHDL is a hardware description language used to model digital electronic systems. It should not be confused with VHDL which is a different type of hardware description language. Both of these languages are being used currently in the industry and the choice is based on personal preferences.

Three different Xilinx development softwares were used. The first one was Xilinx ISE which was used for the development of the custom made IP, the second one was Xilinx Platform Studio which was used for the development of the hardware section of the design and the third one was Xilinx Platform Studio SDK which was used for the software development of the design. For the simulation of our design we had used Modelsim simulation software.

The Xilinx ISE development tool can be used by the user to create custom made hardware design. There are three different ways to create the design. The first option is to create using schematic design. This allows the user to select different components from a large number of available symbols like logic gates, LUTs etc. and to make the necessary interconnections between these elements. The second option is to create the design by using VHDL hardware description language. The third option is to create the design using Verilog HDL hardware description language. In our case the third option was used. Xilinx ISE also allows the user to generate the user constraint file and the testbench file. It also has an incorporated simulation software but it is mostly suitable for small designs and for academic purposes. For simulation purposes Modelsim can be selected as the primary simulation tool and the user is able to run the Modelsim from within the Xilinx ISE.

The Xilinx Platform Studio is a powerful tool provided by Xilinx which allows the user to add a large variety of peripherals to the design. The interconnection of different modules of the design can be done using a graphical interface, helping the user to have a better view over the design. This development tool also allows the user to generate simulation files not only of the custom made IP but also of the entire design which in our case includes the PowerPC microprocessor and the Processor Local Bus (PLB) and the interconnections and communication protocol between them. A screenshot of the structure of the system can be found in figure 3.

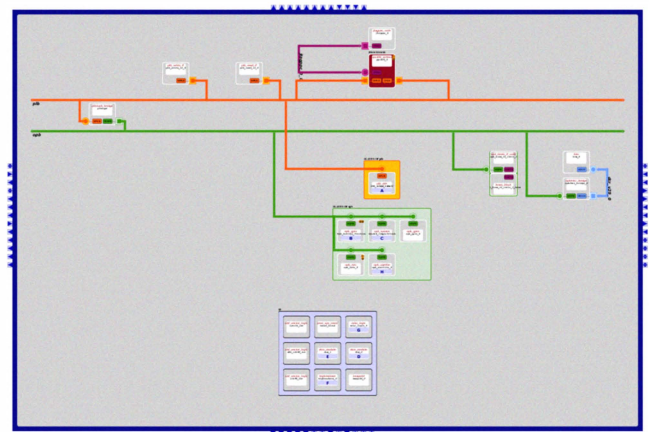


Figure 3. Screenshot from Xilinx Platform Studio

The Xilinx Platform Studio SDK is a software development tool. It allows the user to develop design specific C programs to control the functionality of the hardware or to split the jobs between the hardware part of the design and the microprocessor. This development tool also allows the user to work with a UNIX based operating system that allows him to develop multithread programs and also supports different types of scheduling and synchronization techniques.

The Modelsim simulation software is the most powerful hardware simulation software currently available on the market. It allows the user to simulate designs that have some modules written in Verilog HDL and others written in VHDL and it also allows the simulation of the microprocessor model. For the Modelsim to be able to simulate the entire design (including the microprocessor and other Xilinx specific peripherals), the simulation libraries need to be precompiled both from the Xilinx ISE and Xilinx Platform Studio.

B. Stereo Image Processing Pipeline Implementation aspects

The sensors used in our implementation are Micron's MT9M011 1 Megapixel sensors. These sensors were provided by Terasic, mounted on the same board (TRDB-DC2) one next to the other. The working frequency of the sensors is 25 MHz and it needs to be provided from the hardware design. The sensors use a Bayer CFA (Color Filter Array) and can be controlled using the I2C bus. The values of the pixels are represented on 10 bits.

The monitor we used in our implementation is an off-the-shelf standard CRT monitor with a refresh rate of 60 Hz. The synchronization signals for this monitor are generated in our design but we will discuss this in more detail in the following paragraphs. An example of the internal architecture of our design is presented in figure 4.

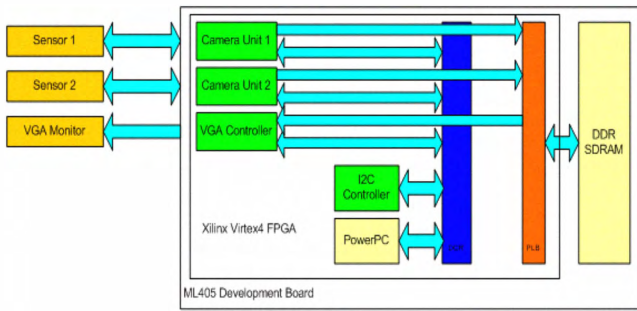


Figure 4. Internal Architecture

The *Camera Unit* is the interface between the image sensors and the DDR SDRAM. All the elements of the IPP (Image Processing Pipeline) are implemented in the Camera Unit. The white balance is controlled by setting the gain values for each colour component in the sensor. This is done by changing the values of certain registers of the sensors using the I2C bus. The demosaicking is done using one of the known algorithms. Then the color correction is done using the characteristics of the sensor. Finally, the gamma correction is implemented in the VGA controller. The camera unit is communicating with the DDR SDRAM via the PLB (Processor Local Bus). The interface protocol for this bus is also implemented in the camera unit. There is one camera unit available for both image pipelines, so they are seen by the PLB bus arbiter as two different clients. After the demosaicking operation, the data is bundled in half-word of 32 bits (R, G, B, Dummy) and prepared to be sent to the DDR SDRAM. An example of the Schematic and Verilog models of a Flip-Flop can be found in figures 5a and 5b.

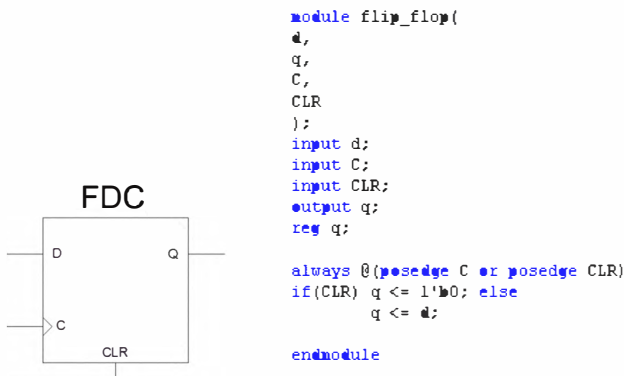


Figure 5.a. Schematic of a Flip-Flop and 5.b. Verilog model of a Flip-Flop

The *VGA Controller* is the interface between the DDR SDRAM and the CRT monitor. The gamma correction is implemented in this controller. This controller also has the PLB bus protocol implemented to be able to read the data from the DDR SDRAM. After reading the data, it unbundles the 32-bit half-word and sends the data to the monitor. The vertical and horizontal synchronization signals are also generated in this controller. The pixel values are being sent to the monitor synchronized with these signals. The VGA Controller is clocked on 25 MHz and the necessary signals are generated using this clock.

The *I2C Controller* is written both in Verilog HDL and C languages. It is based on a hardware-software interface. The

I2C protocol is implemented in software on the PowerPC microprocessor. The C function specifies the sensor that needs to be controlled together with the command. The hardware module, based on this specification from the software sends the command to the corresponding sensor. The sensors can be controlled in real-time independently or simultaneously. The hardware-implemented I2C controller communicates with the processor through the DCR (Device Control Register) bus.

The *PowerPC 405* microprocessor is embedded in the FPGA. It is clocked on 300 MHz and it belongs to one of the fastest RISC microprocessor family available on the market. It has a 5-stage pipeline, separate instruction and data caches and a MMU (Memory Management Unit). It is recommended to be used in custom logic applications [24].

A system BUS is a set of wires. The components of a system are connected to the buses. To send information from one component to another, the source component outputs data onto a bus. The destination component then inputs this data from the bus [25]. The width of the bus defines the size of the data in bits that can be sent on each clock cycle, each of the wires being used to transmit the value of one bit.

The *PLB* bus is a 64-bit bus and has a bus control unit, a watchdog timer and separate address, write, and read data path units with a three-cycle only arbitration feature. The Xilinx PLB bus is based on the IBM PLB bus but there are certain differences between them. The bus has address and data steering support for up to 16 masters and arbitration support for the same amount of masters [26].

The *DCR* bus is a 32-bit bus and is a soft IP core designed for Xilinx FPGAs. It provides support for one DCR master and a variable number of DCR slaves configurable via the design parameters [26]. The main difference between the DCR and the PLB bus is that the PLB bus was designed for large amount of data transfer, this is why it's 64 bits wide and has a complex control unit. The DCR bus was designed mostly for sending control data and this is why it is only 32 bits wide and the complexity of the control unit is smaller.

The *DDR SRAM Controller* is a soft IP core designed for Xilinx FPGAs. It connects to the PLB and provides the control interface for the DDR SDRAMs. It supports 16, 32 and 64-bit DDR SDRAM data widths. The controller supports single-beat and burst transactions. The size of the DDR SDRAM memory on the ML405 development board is 64 MB [26].

V. ASPECTS OF 3D CAMERA DESIGN

As presented in the third section of our paper, one of the conditions in the generation of 3D images is to have separate views over the same image frame. This condition was already fulfilled by using two image sensors with a horizontal displacement between them. Our design supports two types of 3D data transmission. The first one is checkerboard-interleaved which is used in the 3D DLP format [27] and is presented in figure 6.a., and the second one is the frame-interleaved which is presented in figure 6.b.

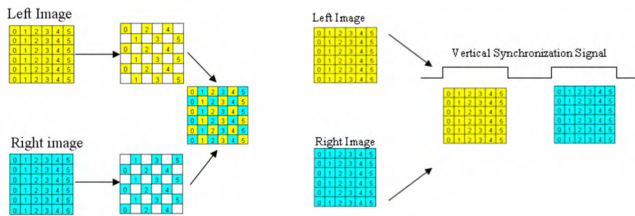


Figure 6.a. Checkerboard-interleaved transmission format and 6.b. Frame-interleaved transmission format

As a display system we used a standard CRT monitor and a pair of shutter glasses. The display technique we used was the frame-interleaved technique where the frames were synchronized with the shutter glasses and this allowed us to control the view for each eye. The checkerboard-interleaved transmission format can be displayed on the CRT monitor as well. The only problem is that by using this format, the horizontal resolution of the original image is reduced to half.

In our design we were using two VGA sensors and by combining the images this way, their quality dropped significantly. The purpose of this implementation was to prove that this format can easily be implemented on a FPGA and by using higher quality sensors we could display high-quality 3D images.

CONCLUSIONS

In our paper we presented the tools and techniques necessary to implement a stereo image processing pipeline on a FPGA and the theory behind the standard image processing pipeline. As an application of this, we presented the aspects of a 3D camera implementation based on the most recent requirements in this area.

ACKNOWLEDGEMENT

The project was financed by the Irish Research Council for Science, Engineering and Technology (IRCSET) and Tessera (Ireland).

REFERENCES

[1] I. Kawahara and K. Suetsugi, "Basic Concept of 3D Full HD plasma theatre system", *Digest of Technical Papers, IEEE International Conference on Consumer Electronics*, pp. 131-132, Las Vegas, January 2010.

[2] J. Konrad, M. Halle, "3-D displays and signal processing", *IEEE Signal Processing Magazine*, pp. 97-111, November 2007.

[3] K. Kim and I. C. Park, "Combined image signal processing for CMOS image sensors", *Proc. of the IEEE International Symposium on Circuits and Systems*, 2006.

[4] I. Andorko, P. Corcoran and P. Bigioi, "FPGA based stereo imaging system with applications in computer gaming", *IEEE International Gaming Innovation Conference*, pp. 239-245, London, 2009.

[5] M. Ionita, I. Andorko and P. Corcoran, "Enhanced real-time face models from stereo imaging for gaming applications", *IEEE International Gaming Innovation Conference*, pp. 17-24, London, 2009.

[6] I. Andorko, P. Corcoran and P. Bigioi, "Hardware Implementation of a real-time 3D video acquisition system", *Submitted to International Conference on Optimization on Electrical and Electronic Equipment*, Brasov 2010.

[7] W. C. Kao, S. H. Wang, L. Y. Chen and S. Y. Lin, "Design considerations of color image processing pipeline for digital cameras", *IEEE Transactions on Consumer Electronics*, vol.54, no.4, pp. 1144-1152, November 2006.

[8] S. C. Hsia, M. H. Chen and P. S. Tsai, "VLSI implementation of low-power high-quality color interpolation processor for CCD camera", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol.14, no.4, April, 2006.

[9] R. Ramanath, W. E. Snyder, G. L. Bilbro and W. A. Sander III, "Demosaicking methods for Bayer color arrays", *Journal of Electronic Imaging*, vol. 11, no.3, pp. 306-315, July 2002.

[10] K. Barnard, V. Carei and B. Funt, "A comparison of computational color constancy algorithms-Part I: Methodology and experiments with synthesized data", *IEEE Transactions on Image Processing*, vol.11, no.9, September 2002.

[11] D. Forsyth, "A novel algorithms for color constancy", *International Journal in Computer Vision*, vol.5, pp. 5-36, 1990.

[12] S. H. Lee and J. S. Choi, "Design and implementation of color correction system for images captured by digital camera", *IEEE Transactions on Consumer Electronics*, vol.54, no.2, May 2008.

[13] G. Hong, M. R. Luo and P. A. Rhodes, "A study of digital camera colorimetric characterization based on polynomial modelling", *Color Research and Application*, vol. 26, no. 1, pp. 76-84, February 2001.

[14] H. Jeong, J. Kim, W. Choi and B. Kang, "Image quality enhancement by real-time gamma correction for a CMOS image sensor in mobile phones", *International SoC Design Conference*, vol.2 pp. 121-124, 2008.

[15] A. Vetro, S. Yea and A. Smolic, "Towards a 3D video format for auto-stereoscopic displays", *Proc. SPIE Conference on Applications of Digital Image Processing XXXI*, vol. 7073, September 2008.

[16] M. T. Pourazad, P. Nasiopoulos and R. K. Ward, "An H.264-based scheme for 2D to 3D video conversion", *IEEE Transactions on Consumer Electronics*, vol.55, no.2 May, 2009.

[17] C. C. Cheng, C. T. Li and L. G. Chen, "A 2D-to-3D conversion system using edge information", *Digest of Technical Papers, IEEE International Conference on Consumer Electronics*, pp. 377-378, Las Vegas, January 2010.

[18] P. Merkle, K. Muller and T. Wiegand, "3D Video: Acquisition, Coding and Display", *Digest of Technical Papers, IEEE International Conference on Consumer Electronics*, pp. 127-128, Las Vegas, January 2010.

[19] D. K. Broberg, "Considerations for stereoscopic 3D video delivery on cable", *Digest of Technical Papers, IEEE International Conference on Consumer Electronics*, pp. 129-130, Las Vegas, January 2010.

[20] J. Baumgartner, "Comcast tries on 3D VoD", *Light Reading's Cable Digital News*, June 2009.

[21] T. Okoshi, "Three dimensional imaging techniques", *Academic Press*, 1976.

[22] I. Sexton, "Parallax barrier display systems", *IEE Colloquium on Stereoscopic Television*, pp. 5/1-5/5, 1992.

[23] N. Dodgson, "Autostereoscopic 3-D displays", *IEEE Computer*, vol. 38, no. 8, pp. 31-36, August 2005.

[24] <http://www-01.ibm.com/chips>

[25] J. D. Carpinelli, "Computer Systems Organization & Architecture", *Addison Wesley Longman*, June, 2001.

[26] www.xilinx.com

[27] http://dlp.com/hdtv/3-d_dlp_hdtv.aspx