Title	Hybrid video-frame pre-processing architecture for HD-video
Author(s)	Bigioi, Petronel; Corcoran, Peter
Publication Date	2011
Publication Information	Zaharia, C. and Bigioi, P. and Corcoran, P.M. (2011) Hybrid video-frame pre-processing architecture for HD-video Consumer Electronics (ICCE), 2011 IEEE International Conference on
Publisher	IEEE
Link to publisher's version	http://dx.doi.org/10.1109/ICCE.2011.5722927
Item record	http://hdl.handle.net/10379/3629

Hybrid Video-Frame Pre-Processing Architecture for HD-Video

Corneliu Zaharia[1], Petronel Bigioi[1,2], *Senior Member, IEEE* and Peter M. Corcoran[2], *Fellow, IEEE*
1, Smart Imaging Division, Tessera Inc.; 2, College of Engineering & Informatics, National University of Ireland, Galway

***Abstract*--As imaging devices begin to support full frame rate video at 720p and higher resolutions image processing algorithms which operate on full image frames become severely challenged due to both the processing costs of operating on full-frame resolution images and the bandwidth challenges of moving image data to and from system memory. In this paper we outline a novel hybrid hardware architecture which enables a range of image processing primitives to be directly generated from a video frame as it passes through the standard imaging pipeline. These primitives can be either efficiently stored with a sequence of video frames, or depending on the computational capabilities of the device they may be recombined with the original video frame data, concurrently with the acquisition of the following video frame, to implement a range of image processing functions prior to MPEG video data compression. Examples include color segmentation, face tracking, frame-to-frame registration, motion stabilization, skin detection, dynamic range enhancement, global and local histograms and computation of integral image functions.**

I. INTRODUCTION

After images are acquired by the image sensor within a digital imaging system they must be processed before display or storage on the device. The typical image processing chain is illustrated in *Figure 1* below:

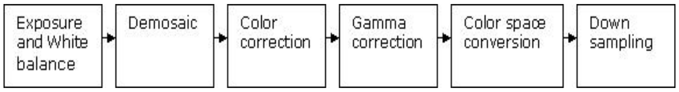

Fig 1: Conventional Image Processing Pipeline

When we wish to implement a real-time video imaging system we run into significant constraints with such an image processing pipeline (IPP) as image data needs to be read from memory on each stage of the IPP and written back after some operations. For HD video the memory bandwidth experiences significant challenges. Thus many elements of the IPP are implemented directly in hardware embodiments in todays video acquisition devices.

Modern digital still cameras (DSC) implement more sophisticated image and scene analysis than can be provided by a basic IPP as shown in Fig 1. In particular these image acquisition devices can detect and track face regions within an image scene, they can analyze and detect blemishes and imperfections within such regions and correct such flaws on the fly; facial enhancement can be applied; image blur and image motion, translational and rotational, can be determined and compensated; facial regions can be recognized and associated with known persons and so on. All of these techniques rely on an analysis of the image scene.

Typically this requires the reading of blocks of image data from a memory store followed by various processing stages of this data. Intermediate data structures must be stored temporarily within the image store to facilitate each scene analysis algorithm. In some cases these data are specific to a single algorithm; in others a data structure may persist across several different scene analysis algorithms. In all cases image data must be moved between image store memory and a CPU to perform various image processing operations. Where multiple algorithms are applied image data must be read several times to perform different image and scene processing operations on each image. This architecture is illustrated in *Figure 2* below.

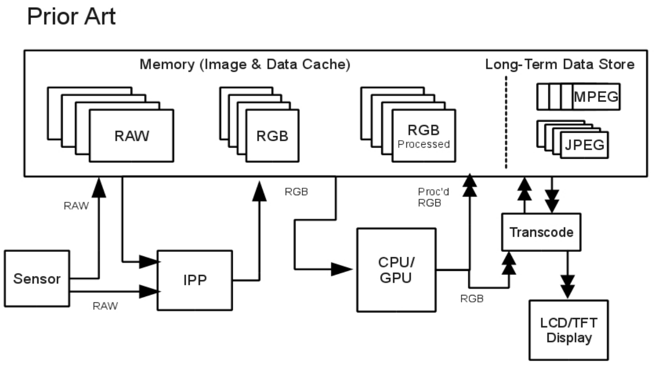

Fig 2: Prior Art Hardware to implement IPP (hardware) and other higher level scene analysis functions (typically in software).

Now, within a digital camera images are typically acquired individually and a substantial time interval, typically of the order of 1 or more seconds, is available between image acquisitions for scene analysis and post processing of individual images.

Within a modern video appliance data must be processed at frame rates of at least 30 fps and due to memory constraints data needs to be digitally compressed and written to a long-term memory store almost immediately. Furthermore, a low-resolution preview stream is not generally available as in the case of a DSC. Finally the requirements of handling a full-HD video stream imply that memory bandwidth management is extremely challenging within such an appliance.

II. The Challenge

In order to achieve the benefits of modern scene analysis techniques such as are presently available within a DSC for a HD video device we identify several key problems: (i) it is not possible to store and perform complex scene analysis on full HD between video frame acquisitions; in fact the size of a full HD images implies it is very challenging simply to move such images through an IPP and into a video compression unit onto long-term storage; (ii) while some limited scene analysis may be possible through hardware additions to the IPP this requires that many settings and configurations must be fixed prior to beginning real-time acquisition of the video stream and cannot be dynamically adapted responsive to ongoing scene analysis; such hardware subsystems are also expensive in terms of development costs and system gate counts; (iii) there is no scope to share image processing data primitives between scene analysis algorithms without introducing very large shared memory buffers into such hardware subsystems;.again system gate counts rise very significantly; in the extreme each scene analysis algorithm, in addition to its own algorithm-specific memory buffers, would need to buffer a full image frame in order to perform full scene analysis.

Thus we can identify a broadly scoped problem that current scene analysis techniques and resulting image enhancement benefits cannot be sensibly applied to real time HD video using current state-of-art techniques.

III. The Hybrid Architecture

Our architecture (AHIP) is based on a one pixel per clock cycle input and generating a number of different types of image processing primitives which provide useful knowledge about the current image/video frame. Each primitive is generated by a processing chain which comprises one or more pixel processing blocks, or modules. These are linked together by a number of internal data buses which many be dynamically switched. Thus multiple modules may share the same input data. Further, the output of individual processing blocks may be combined logically. A simple example is shown in *Figure 3* below.

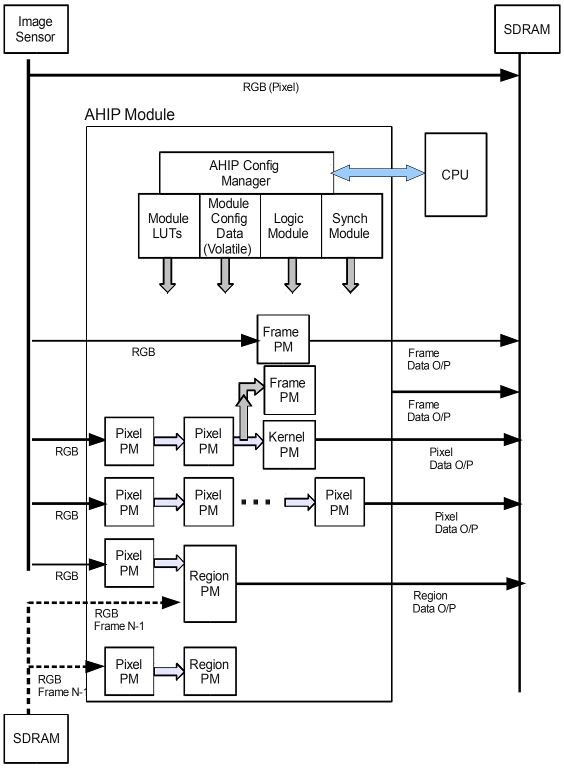

Fig 3: AHIP module illustrating generic processing modules arranged into processing chains for different image primitives.

The individual outputs from multiple processing chains are typically combined into a single data word before being output to external memory (SDRAM) as this facilitates optimal use of memory and external memory buses. Because of the differences in processing time between processing chains a synchronization module is integrated with the logical circuitry to ensure correct alignment of the output data.

In most cases the relevant image primitive(s) and the main image need only be read once in order to analyze and/or apply a particular enhancement algorithm to the image. It is also possible to load primitives from multiple algorithms together with a single read of the main acquired image in order to execute these multiple algorithms on a single image read.

A top-level overview of the AHIP module within the known architectural framework of a video appliance is shown in *Figure 4*. Note that data flows directly, pixel by pixel, to the AHIP from the IPP and primitives are generated at the same time as the image frame is written, with some finite delays due to differences in buffer lengths and the complexity of the technique to generate each image primitive.

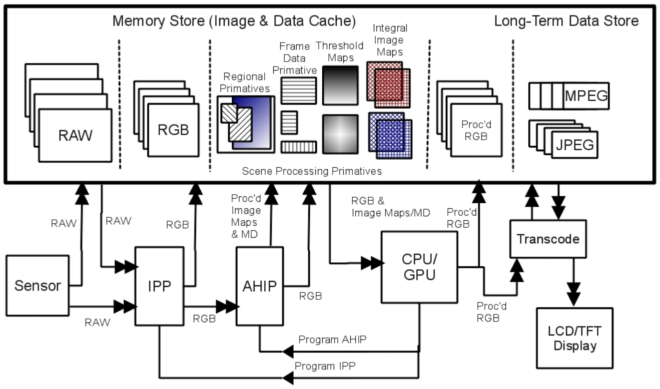

Fig 4: IPP Hardware with AHIP - note pass-through nature of AHIP

To give a practical example, Figure 5 shows an example of an original image frame and an associated 4-bit (16 grayscales) skin map. Such a map can be used to measure the skin quality of different pixels and may be adjusted, using dynamic thresholds, from frame to frame based on variations in global frame data. This dynamic measure of skin quality can be used, for example, to determine the type and strength of skin enhancement which is applied to different regions of a persons face, enabling a real-time face beautification algorithm to be readily implemented.

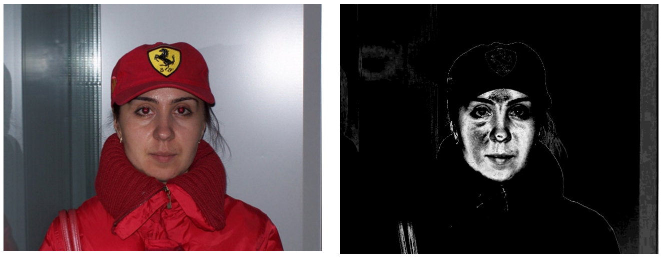

Fig 5: Original image and skin-color map with 16 thresholds (4 bit).

IV. Conclusions

In our conference presentation a number of additional examples of AHIP algorithms will be discussed in more detail. These include real-time face tracking, frame-to-frame image registration, motion stabilization and integral image processing techniques.

Some technical details will be given, including typical gate counts for different complexities of AHIP modules.