FPGA-based Architecture for Real-Time IP Video and Image Compression

D. Maroulis, N. Sgouros, D. Chaikalis
Dept. of Informatics and Telecommunications
National and Kapodistrian University of Athens
Panepistimiopolis, Ilisia, Athens 15784, GREECE
{rtsimage, nsg}@di.uoa.gr

Abstract –**Three-dimensional imaging applications require high resolution images that finally result in high data volumes. Due to bandwidth and storage restrictions, an efficient and robust compression scheme must be developed in order to overcome these limitations. This work presents a hardware implementation of a real-time disparity estimation scheme targeted but not limited to Integral Photography (IP) 3D imaging applications. The proposed system demonstrates an efficient architecture which copes with the increased bandwidth demands that 3D imaging technology requires. Moreover, the system can successfully process high resolution IP video sequences in real-time.**

I. INTRODUCTION

The demand for three-dimensional imaging applications is continuously rising covering a wide variety of specialized to everyday visual communications. A special category of 3D imaging system that provides enhanced sense of depth, full colour support and in most cases multidirectional parallax, functions on the principles of Integral Photography (IP), first introduced by Lippman [1] back in 1908. In Fig. 1, a typical IP capturing and display setup is illustrated.

A portion of a type of an IP image is depicted in Fig. 2. A set of elemental sub-images is formed with the use of a microlens array in the capturing stage. As it is evident, neighbouring sub-images exhibit a high degree of correlation, which results in high volumes of redundant information.

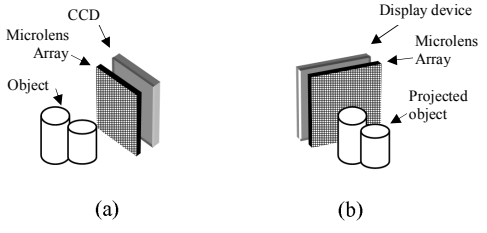

Figure 1. An autostereoscopic capturing and display setup based on the principles of Integral Photography, (a) the capturing setup, (b) the display setup

This work was realized under the framework of the Reinforcement Programme of Human Research Manpower ("PENED 2003" – 03ED656), cofunded by the General Secretariat for Research and Technology, Greece, and the European Social Fund.

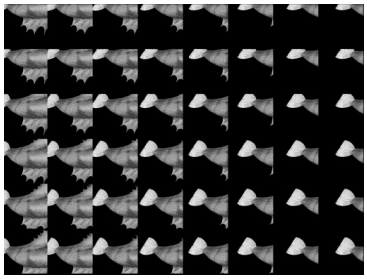

Figure 2. Part of an IP image

One of the main issues when developing applications based but not restricted to the IP principle, is the necessity to handle high resolution images that result in high bandwidth and storage requirements. Consequently, a high-efficiency compression scheme of the associated data is crucial. The inherent data properties of 3D images gave rise to a number of new techniques [2-4] for coding highly correlated data, in order to achieve high compression ratios over traditional methodologies.

Hardware implementations can exploit pipelined and massively parallel processing, thus being an efficient solution for accelerating time-critical and high complexity tasks, such as motion estimation used in various compression schemes, especially for real-time applications.

In this paper, an efficient hardware system of a real-time motion estimation scheme specially modified for IP images is presented, targeted for real-time 3D displays and capturing applications, where high compression ratios must be combined with real time performance of the whole process.

The rest of the paper is organized in four sections. In section 2, we describe the algorithm on which the compression architecture is based, and we refer to the block matching criteria used. In section 3 we analyze the proposed hardware design, the results of which are apposed in section 4. Finally, the discussion on the innovation and the results of the proposed system are summarized in the last section.

II. DISPARITY ESTIMATION

A. Algorithm Description

The implemented algorithm [2] provides high compression rates for all kinds of IP applications while exhibiting a high degree of robustness and feasibility. The technique utilizes the fact that the elemental sub-images can be treated as a spatial sequence of subsequent frames having a predetermined motion pattern. This knowledge allows for developing highly parallel and thus time efficient hardware implementations. The advantages of the method derive from the use of standard 2D-DCT and the relatively low complexity disparity compensation modules that replace a full motion estimation scheme of a typical MPEG implementation. Additionally, standard quantization and coding techniques allow for easy hardware prototyping and scalability of the algorithm. In detail, a sequence of reference (I) and predicted (P) sub-images is formed taking into account the additional constraints that are imposed from the properties of the IP image data. Fig. 3 is an illustration of an IP image segmentation in spatial blocks along with the search method followed.

B. Block Matching Criteria

Three key issues are associated with motion estimation, namely the size and type of traversal of the search area and the metric used for determining the "best match". For these issues, many methods have been proposed [5] in order to reduce the number of computations. The most commonly used metric which is also used in our context is the Sum of Absolute Differences (SAD), which adds up the absolute differences between corresponding elements in the blocks. For an 8x8 block, the SAD is calculated as follows:

$$SAD(U,V) = \sum_{x=0}^{7}\sum_{y=0}^{7}\left|U(x,y) - V(x,y)\right| \quad (1)$$

where $x,y = 0,1,2,\ldots 7$ are spatial coordinates in the pixel domain and U,V are arbitrary 8x8 blocks in adjacent image frames. The actual coordinates of these blocks in the frames are chosen by the search algorithm used.

Concerning the search area, a unidirectional block search method is performed, ensuring an optimal block match. Moreover, the exhaustive block search method is selected, since the proposed technique targets to compressed images

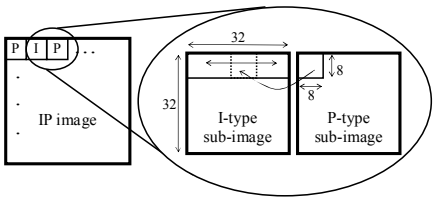

Figure 3. (a) The I-type and P –type sub-images in an IP image, (b) block search area outline

of high quality. The computational cost of the exhaustive search is not $O(p^2)$ as in the 2D search but reduced to $O(p)$, where p is the size of the unidirectional search area, due to a priori knowledge that allows for unidirectional search. Moreover, the method retains the merits of exhaustive search producing optimal matching.

III. HARDWARE DESIGN

The proposed algorithm contains modules that require intensive computational operations. To this end our work targets to accelerate these modules and provide real-time performance that is required in many time critical applications. This is achieved by implementing in hardware the modules that perform disparity estimation and represent the bottleneck of the encoding process (Motion Estimation is a great percentage of the overall complexity - typically about 80%). The block diagram of the aforementioned compression scheme is presented in Fig. 4. In this figure, the shaded components represent the part that is implemented in hardware, which we name Disparity Vectors Matrices' Generator (DVMG). For the proposed hardware design, VHDL is used, and the design is implemented on a Virtex XCV-2000E FPGA device, fitted on a Celoxica RC1000-PP PCI board [6].

A. System overview

The DVMG is comprised of the following modules: the two-dimensional Discrete Cosine Transform (2D-DCT) Unit, the Quantizer and the Inverse Quantizer, the 2D Inverse-DCT (2D-IDCT) Unit, the Disparity Vectors' Creation Unit (DVC Unit or DVCU), an Address Generation Unit, and the Control Unit along with the necessary FPGA memory modules. The first four units that are mentioned above are grouped to one module responsible for compressing and decompressing the I-type sub-images, and are referred to as the I-type sub-image Compression module. Fig. 5 presents a block diagram of the hardware system.

Before the DVMG starts its operation cycle, an image part of up to 2 Mbytes in size is downloaded to the board memory, in a way that pixels of three of the sub-images forming the P-I-P sub-image triplets can be accessed on each clock cycle by the FPGA, aiming to increase the parallelization of the overall procedure. The DVMG starts its operation cycle by transferring 16 P-I-P sub-image triplets to the FPGA memory modules. From thereon, a series of operations are executed in order to calculate and write to the results' memory the final data, which are the disparity vector matrices for the 16 sub-image triplets. Having used all the data available in the FPGA memory, the next 16 P-I-P sub-image triplets are transferred from the board memory to the FPGA memory and the disparity vector matrices are created in the same fashion. The operation cycle of the DVMG ends when all the image data residing in the board memory has been processed and the results are transferred from the results' memory module to the board memory. After the completion of the operation cycle, the host is notified and uploads the final data, and the FPGA system returns to its

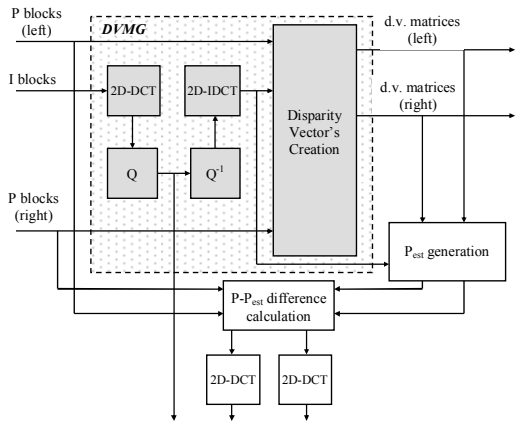

Figure 4. Block diagram of the IP image compression scheme as detailed in [2]. The Disparity Vector Matrices' Generation (DVMG) represents the part that is implemented in hardware.

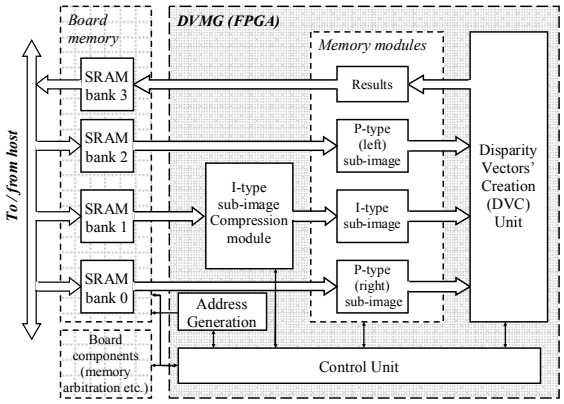

Figure 5. An overview of the DVMG

initial state, waiting for a new start signal from the host. Due to page restrictions, we describe the functionality of each sub-system, which is illustrated in Fig. 5, rather than going deep into implementation details.

B. I-type sub-image Compression Module

In order to achieve more efficient compression rates, the I-type sub-images, used as the reference type image in the encoding procedure, are initially transformed using a 2D-DCT transform, then appropriately quantized to the preferred quality level using standard MPEG quantization tables, and finally inversely transformed. This is done in order to assure that the image used as reference frame in the encoding procedure is actually the input at the decoder, producing the optimal results.

C. DVC Unit

The Disparity Vectors' Creation Unit is responsible for generating the disparity vector matrices for the P-type sub-images by calculating the minimum SAD value for each 8x8 block (see Fig. 3). The SAD value calculations could be performed in a completely parallel manner. Such an approach though, poses the problem of high bandwidth and area demands, which requires a multitude of FPGAs [7].

In the proposed system, the SAD unit is comprised of 8 Absolute Difference units. The addition part of the SAD is performed using an adder tree, which, as demonstrated in [7], presents several advantages in terms of parallelization capabilities and area usage when compared to a systolic array or a sequential addition. By successive repetition, each SAD unit produces a SAD value for an 8x8 block in 8 clock cycles, by processing in parallel 8 pixel pairs from one P-type and I-type sub-image block in every clock cycle. The SAD values for the P blocks are calculated by partitioning the P-type and I-type sub-image in 8x32 pixel areas (referred to as four-block areas hereafter) and traversing these areas columnwise in a left-to-right direction, as shown in Fig. 6.

By introducing delay registers (R) before the SAD units, as shown in Fig. 6, the sub-image pixel values are transferred only once from the FPGA memory to the DVC Unit. Using this technique, the DVC performs the SAD values calculations in approximately 92% less clock cycles, compared to the simplest approach of reading each pixel for every calculation in which it is needed.

In total, 51 SAD values are computed for the 3 P blocks of the P-type four-block area. According to the unidirectional exhaustive search method used, no SAD values are computed for the rightmost P block in the four-block area. Aiming to minimise memory access and the time during which each SAD unit stays idle, it was concluded that we can use only 25 units, by inserting a "clear" cycle each 8 clock cycles, during which a synchronous clear signal is rippled through the SAD units, starting with SAD unit #0. The minimum SAD value for each P block derives from the comparison of the SAD values calculated for it and takes place in the Comparison module, which is illustrated in Fig. 6.

D. Address Generation and Control Units

An Address Generation Unit is necessary for the proper arrangement of the data in the memory modules. The difference in the read and write sequence between the P-type and I-type sub-images lies in the need for compressing and decompressing the I-type sub-images. It must also be noted that, since the 2D-IDCT output is in the same form as the 2D-DCT input, the output of the I-type sub-image Compression module needs to be rearranged according to the P-type sub-image address sequence before written to the FPGA memory modules.

Each module needs an appropriate sequence of enable, clear and other signals for the DVMG to function properly. Given the volume of the design and the multitude of signals and addresses to be driven on each clock cycle, it was deemed preferable to implement a dedicated control in each unit for the generation of its internal signals, while designing a main Control Unit for the enable, start and clear signals of the units. In this fashion, the inputs and outputs of the units are limited to small numbers and focus on the data processed.

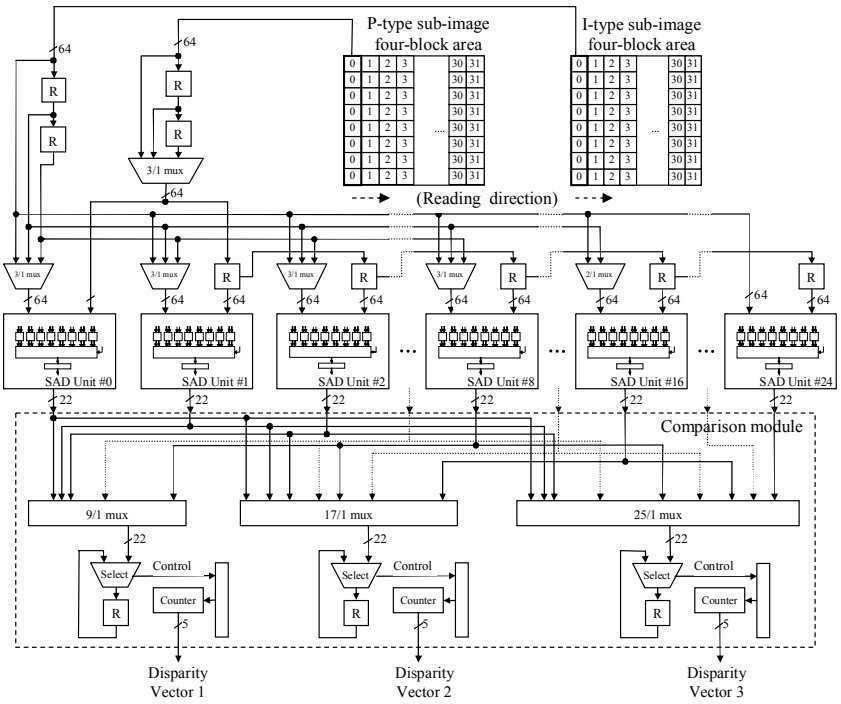

Figure 6. Block diagram of the DVC Unit, showing the additional data buses and the delay registers in order to feed each SAD unit with the appropriate image data.

IV. RESULTS

Our system can be clocked with a maximum frequency of 11,29 MHz and occupies 16,876 FPGA slices, which correspond to 87% of the Virtex XCV-2000E area. The number of operations required for processing an entire IP image is proportional to the image size. Table 1 summarizes our results for different IP image sizes, using representative IP video sequences. The hardware's processing time reveals that the DVMG architecture can successfully process IP images suitable for common types of broadcast applications, in a rate equal or greater of 30 images per second, satisfying real time conditions.

TABLE I. THE PROCESSING TIME OF THE DVMG AND THE NUMBER OF IP IMAGES THAT IT CAN PROCESS PER SECOND

Image dimension (width) x (height) (pixels)	**Processing time (ms)**	**Images/sec**
640x480	12,94	77
720x576	16,64	60
800x600	20,33	49
1024x768	31,42	31

V. DISCUSSION

This paper presents a hardware architecture of a disparity estimation scheme for Integral Photography images. Targeting to a single Virtex-E FPGA with an area capacity equivalent to 2 million logic gates, we succeeded in implementing a novel, real-time disparity estimation hardware system, which demonstrates minimized memory operations and bandwidth requirements. This system manages to deliver IP image and video content for use in broadcast-type applications in real time, as well as coping with increased demands for desktop 3D applications.

REFERENCES

[1] 1 G. Lippman, "La photographie integrale", C. R. Acad. Sci, 146, pp. 446-451, 1908

[2] N. Sgouros, A. Andreou, M. Sangriotis, P. Papageorgas, D. Maroulis, N. Theofanous, "Compression of IP images for autostereoscopic 3D imaging applications", 3rd International Symposium on Image and Signal Processing and Analysis (ISPA03), pp. 223-227, Vol. 1, Rome, Italy, September 18-20, 2003

[3] D. Shah and N. A. Dodgson, "Issues in multi-view autostereoscopic image compression," in *Proc. IS&T/SPIE Symp. on Stereo. Disp. and Appl. XII,* San Jose, Jan. 2001, vol. 4297, pp. 307-316.

[4] R.Zaharia, A. Aggoun,M.McCormick, "Adaptive 3D-DCT compression algorithm for continuous parallax 3D integral imaging," *Signal Processing: Image communication*, Elsevier Science, 2002, vol. 17, Iss. 3,pp. 231-242

[5] S. Wong, S. Vassiliadis, S. Cotofana, "A sum of absolute differences implementation in FPGA hardware", 28th Euromicro Conference (EUROMICRO'02), pp. 183-186, Dortmund, Germany, September 04 - 06, 2002

[6] Celoxica RC1000-PP development board: Hardware reference, www.celoxica.com

[7] S. Wong, B. Stougie, S. Cotofana, "Alternatives in FPGA-based SAD implementations", IEEE I.C. on Field Programmable Technology 2002 (FPT '02), Hong Kong, December 2002