Real-Time Stereo Vision Hardware Architecture Suitable for Multiple Platforms

Andy Motten, Luc Claesen
Expertise Centre for Digital Media
Hasselt University – tUL – IBBT
Wetenschapspark 2, 3590 Diepenbeek, Belgium
{firstname.lastname}@uhasselt.be

Abstract— **This paper presents a real-time stereo vision System-on-Chip (SoC) architecture for a depth-field generation processor as required in 3D TV applications. This architecture is fully scalable and parameterizable to allow for custom SoC implementations, as well as rapid prototyping on FPGAs. An on-chip memory block architecture is used that allows parallel access to all pixels located in a chosen window of the image. A real-time stereo matching calculation at a frame rate of 56 Hz with a resolution of 800x600, a disparity of 80 and a window size of 11x11 has been realized using this architecture without the need for external memories.**

Keywords: real-time stereo matching; adaptable window; computer vision; Parallel memory architecture; system-on-chip; FPGA

I. Introduction

Stereo matching has long been an important research topic in computational video. Many stereo matching algorithms have been investigated and published. An extensive comparison between different algorithms can be found in [1,2]. Dense stereo matching algorithms can be divided in local (area-based) and global (energy-based) algorithms.

Local and window based methods calculate the differences between the left and right images from a small part of the images. They produce a decent depth image result and are suitable for real-time applications. Selection of the ideal window size is a delicate trade-off. It should be large enough to contain distinguished features, but small enough to keep depth discontinuities. Global methods produce an accurate depth image, but are more time consuming. An example is the segmentation based technique [3] which segments the image first and afterwards labels each region with a disparity.

Implementations on hardware are mainly based on a local window based stereo matching architecture. The matching cost computation often used is the sum of absolute differences (SAD) or the census transform [14]. The support window can be square, rectangular or adaptable. [4] proposed a low cost stereo vision system on a FPGA based on the census transform algorithm. A window-parallel pixel-serial architecture-based VLSI processor to maximize the utilization percentage of processing elements with an adaptable window is presented in [5]. [6] proposed a FPGA based stereo matching architecture that uses SAD and a tree based minima calculation. The architecture is highly pipelined which allows to achieve a frame rate of 600 fps using 450x375 input images with a disparity of 150 pixels. [7] compares the resource requirements and performance on a FPGA of a SAD based stereo matching implementation. Different shapes of windows are evaluated to check their influence on the generated depth image. A reduction of the amount of adders used is achieved by decomposing the block SAD calculation in a column and a row SAD calculation. A real-time FPGA-based stereo vision system is presented in [8] that makes use of the census transform. Their system includes all the pre- and post-processing functions as rectification, LR-check and uniqueness test in a single FPGA.

Recently more advanced local based methods make use of color information to select the optimal support window. A good overview of these methods can be found in [9].

The adaptive-weight algorithm proposed in [10] adjusts the support weight of each pixel in a fixed sized window. The support weights are depending on the color and spatial difference between each pixel in the window and the center pixel. Dissimilarities are computed based on the support weights and the plain similarity scores. Their experiment indicates that a local based stereo matching algorithm can produce depth maps similar to global algorithms. A hardware implementation can is published in [11].

[12] extends the adaptive-weight algorithm of [10] by using information from segmentation. It allows inclusion of connectiveness of pixels and segment shapes, instead of relying only on color and spatial distance.

This paper presents a real-time stereo matching System-on-Chip (SoC) architecture for a depth-field generation processor as required in 3D TV applications. While choosing the architecture, particular attention has been given to pipelining, possible exploitation of parallelism and limiting the number of external components. The architecture consists of a multiple parallel on-chip memory architecture, a segmentation based SAD matching cost computation and a tree based minima calculation with registers to store intermediate results. This architecture is fully scalable and parameterizable to allow for custom SoC implementations, as well as rapid prototyping on FPGAs. The current implementation of this architecture allows for a real-time stereo matching calculation at a frame rate of 56

Hz with a resolution of 800x600, a disparity of 80 and a window size of 11x11.

The remainder of the paper is organized as follows: Section two describes the real-time stereo matching architecture including several sub-blocks. Section three shows and discusses the hardware and the different configuration possibilities. Section four draws the conclusions.

II. Stereo Matching Architecture

A. Basics and Requirements

The stereo matching algorithm takes two undistorted and rectified images that have been taken by two cameras that have a vertical alignment and a horizontal offset (Fig. 1). Objects will appear on both images on the same horizontal line (the epipolar line). The horizontal distance between the same objects on the left and right images is called the disparity. Objects that are close to the cameras will have a larger horizontal disparity than objects that are far away. The goal of the stereo matching algorithm is to measure the disparity between all pixels in the image.

The main advantage of dedicated VLSI or SoC architectures in comparison to general purpose CPUs or GPUs is its inherent parallelism and freedom of architecture. The major focus of the architecture presented in this paper is on the maximization of the parallel calculations needed for stereo vision processing. Memory usage will be of a particular importance since it is not feasible to calculate data in parallel if there is no timely access to parallel data to be processed. A good balance is needed. In this context, a parallel memory architecture is used that makes use of multiple on-chip memory blocks [13].

The architecture that is presented has been developed in a scalable and parameterized way. In this way a custom depth field processor SoC module can be generated and tuned to the available resources, the implementation technology and application at hand. The main parameters are the window size and the disparity depth. They directly influence the amount of hardware generated in an ASIC or FPGA.

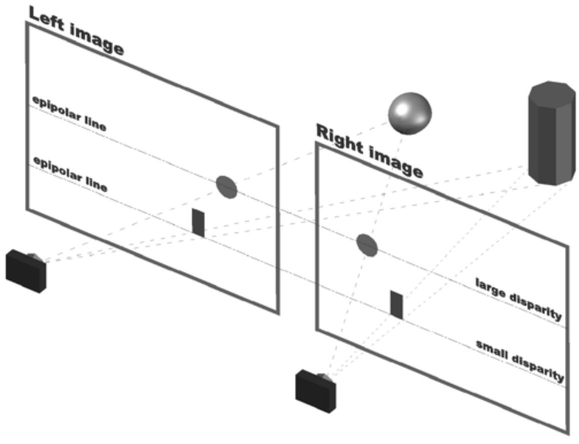


Figure 1. Stereo Vision Setup

B. General Structure

One of the goals of this architecture is to limit the number of external components. Instead of the commonly used frame buffer to capture the pixels coming from the camera, this architecture processes the pixel data without using a large buffer. One line buffer for each camera is needed to resolve the difference in clock speed between the camera and the memory write module. To reduce noise, a median filter is placed just before the pixels entering the memory. Since the bit width of the data bus of the memory is commonly wider than the bit width of the pixel data, a multiplexer is used to combine successive pixel data's in one memory write.

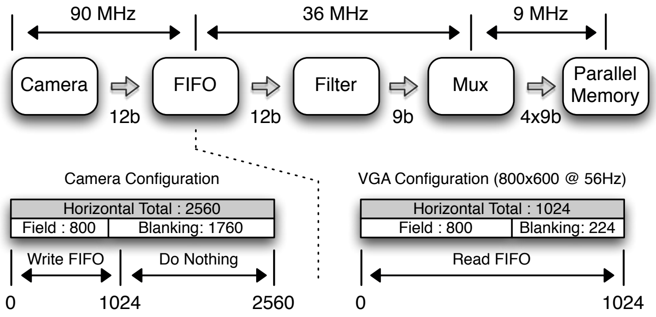


Figure 2. Input Processing Module

Fig. 2 shows an example of an input processing module. The Pixel clock of the camera is configured to run at 90 MHz. Only the first 1024 pixels of each line are written to the FIFO. In this way the FIFO can be read out at 36 MHz without creating an underflow or overflow of the FIFO. The part of the blanking period that is written to the FIFO is the same size as the horizontal blanking period of the VGA specifications for a resolution of 800x600 with a frame rate of 56 Hz. In this way, the output of the FIFO can directly be connected to a VGA screen. The last step of this module stores four successive pixels in the parallel memory at a clock rate of 9MHz.

The two input streams of the cameras are compared with each other using a Sum of Absolute Differences (SAD) calculation (see Fig. 3). Every clock cycle a window of the right camera is compared with four windows of the left camera. Since four successive pixels are stored in one memory location, one memory read accesses four pixels. Which means that four comparison modules need to be implemented in parallel.

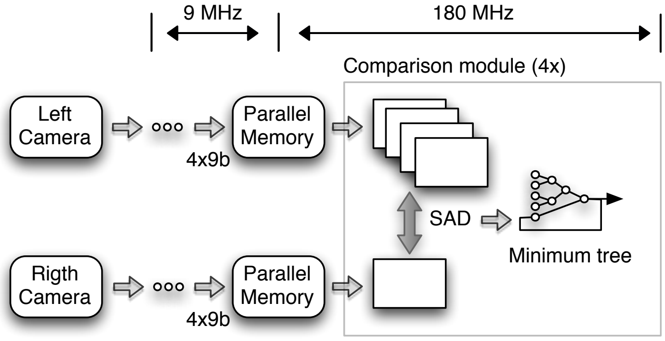


Figure 3. Comparison Module

The frequency of the comparison module controls directly the possible depth range of the stereo matching architecture and can be adapted to the available resources. In the example on Fig. 3 a frequency of 180 MHz is chosen so that depth module runs twenty times faster than a memory write. This leads to a depth range of eighty when the comparison module has a depth of four.

The depth range of a single comparison module is limited to a small depth range. It can be increased to accommodate a larger depth range if the maximum operating frequency is reached.

On each clock cycle (CC), the comparison module compares the reference window with four other windows (Fig. 4). The lowest SAD score and index are saved in a register and reused on the comparison of the next four windows.

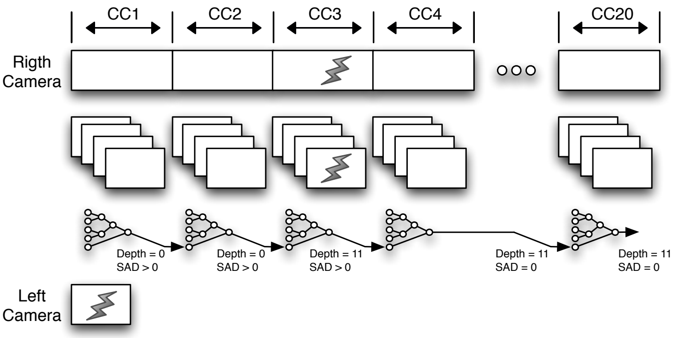


Figure 4. Comparison Module Operating at a Higher Frequency

C. Memory Architecture

Due to the sequential way in which digital video data are presented, video signal processing architectures are traditionally built around line buffers. In the line buffers a number of the most recent scan lines are kept on-chip. Line buffers could be implemented as shift registers, but are currently efficiently implemented as on-chip memory blocks with dedicated addressing logic, such that they are used as FIFOs, typically one FIFO per scan line. At the outputs of the FIFOs, corresponding column pixels for the recent scan lines are accessed. These can then be stored in a shift register array with the size of the area of interest for window based video operations such as filtering, edge detection, sharpening, resampling etc.

However, the traditional scan line based FIFO architecture does not allow for a complete window refresh on each clock cycle. It also does not fully exploit the parallelism that is available with multiple on-chip memory blocks. It.

In [13] the authors present a parallel System-on-Chip (SoC) memory architecture for a stereo vision system. It allows for a parallel access to all pixels located in a chosen window of the image. Using this architecture a complete window refresh on each clock cycle is possible, which can be used to increase the depth range of a stereo matching algorithm. The proposed architecture allows random memory access, which is used in this paper to substantially increase the disparity range.

D. Cost aggregation

The Sum of Absolute Differences (SAD) calculates the differences between two selected windows. It is a measurement of similarity between two parts of an image. Fig. 5 shows the calculation of the main building block of the calculation; the absolute difference (AD).

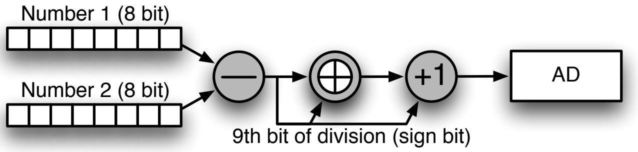


Figure 5. Absolute Difference Calculation

The summation part of the SAD uses a window and depth parallel approach to calculate the sum of absolute differences for all depths and all windows in one clock cycle. Two main methods are integrated in this architecture. The first method uses a fixed window SAD block calculation (1). When using a fixed window shape, implicitly depth continuity across this window is assumed. This assumption is not correct at depth edges, where the center pixel depth is different from some (or the majority) of the surrounding pixels depths.

$$SAD = \sum_{i=1}^{window\ size} absolute\ difference(i) \qquad (1)$$

A more conservative assumption is to assume depth continuity across pixels with similar color [11]. The second method uses a fully adaptable window. For each window a binary mask window is generated which selects the supporting pixels in the cost aggregation phase of the SAD algorithm. This selection is performed using color similarity and spatial distance metrics (2).

$$w = \begin{cases} 0\ if\ (\Delta chroma * \Delta distance) > threshold \\ 1\ otherwise \end{cases} \qquad (2)$$

$$SAD = \sum_{i=1}^{window\ size} w * absolute\ difference(i) \qquad (3)$$

Since the weights (w) in this architecture are ‘0’or ‘1’, the multiplication in (3) can be replaced by an AND operator. This will accommodate for an efficient hardware implementation.

E. Minima Selection

The minima selection is based on an iterative minima tree calculation (Fig. 6). The SAD results are pair wise compared, while each time the lowest value is stored in a register. Afterwards, these registers are pair wise compared and stored in registers. These steps are repeated until one value remains. Storing of the intermediate results in registers makes this method interesting for pipelining.

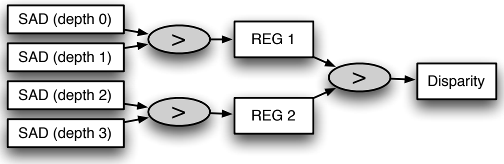


Figure 6. Minima Selection Tree

III. Implementation and Results

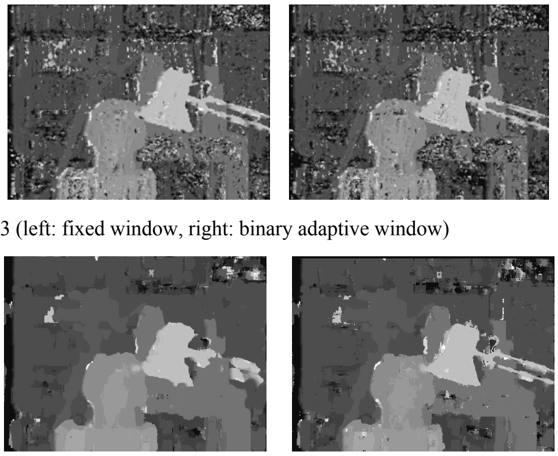
Matlab has been used to generate, out of the chosen parameters and the high level architecture, a complete stereo matching architecture for both simulation and hardware generation from a high level description. This allows an initial check of the stereo matching architecture in Matlab before implementation on the actual hardware. Using this framework, comparison between different stereo matching parameters and architectures can be rapidly performed.

The architecture and methods presented in this paper have been implemented on an FPGA system, based on an Altera Cyclone II with 68.416 logic elements and 250 memory blocks. The sources of the input streams are two cameras with a resolution of 800x600 and a frame rate of 56 Hz. Both fixed window as well as binary adaptive window SAD has been implemented in hardware.

The stereo matching architecture has three main tunable parameters. First the window width and height can be adapted to the application needs. The number of memory blocks used will depend on the size of this window. The second parameter is the operating frequency of the comparison module, the larger the operating frequency, the greater the disparity range can be.

Preliminary results indicate that a real-time stereo matching architecture with a depth range of 80, a resolution of 800x600, a frame rate of 56 Hz and a window size of 11x11 can easily be achieved on a low-cost Cyclone II FPGA device without the need to use external memories.

Fig. 7 shows the depth maps of the Tsukuba stereo pair [2] with a binary adaptive subwindow compared with a squared fixed window [11]. The results indicate that the quality of the resulting depth map increases when using a binary adaptive subwindow. Even with smaller window sizes, small details around the edges are noticeable improved. With larger window sizes the smoothing effect stays while preserving small details around the edges.



a. 3x3 (left: fixed window, right: binary adaptive window)

b. 11x11 (left: fixed window, rigth: binary adaptive window)

Figure 7. Depth map quality of the Tsukuba stereo pair [2] in function of window size and aggregation window type

IV. Conclusions

This paper presents a real-time stereo matching System-on-Chip (SoC) architecture for a depth-field generation processor. The architecture consists of a multi parallel on-chip memory architecture, a segmentation based SAD matching cost computation and a tree based minima calculation. By increasing the operational frequency of the matching module, a twenty-fold increase of disparity range is achieved. It is shown that this architecture can be implemented efficiently into a SoC design without the need for external memories.

References

[1] N. Lazaros, G. C. Sirakoulis, and A. Gasteratos, “Review of stereo vision algorithms: From software to hardware,” International Journal of Optomechatronics, vol. 2 (4), 2008, pp. 435-462.

[2] D. Scharstein, and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” International Journal of Computer Vision, vol. 47 (1), 2002, pp. 7-42.

[3] R. Szeliski, Computer vision: algorithms and applications, unpublished

[4] C. Murphy, D. Lindquist, A. M. Rynning, T. Cecil, S. Leavitt, and M. L. Chang, “Low-cost stereo vision on an FPGA,” International Symposium on Field-Programmable Custom Computing Machines, 2007, pp. 333-334.

[5] M. Hariyama, and M. Kameyama, “Pixel-serial and window-parallel VLSI processor for stereo matching using a variable window size,” Interdisciplinary Information Sciences, vol. 7 (2), 2001,pp. 289-297.

[6] K. Ambrosch, M. Humenberger, and W. Kubinger, “SAD-based stereo matching using FPGAs,” in Embedded Computer Vision, Advances in Pattern Recognition, K. Branislav, ed. London: Springer-Verlag, 2009, pp. 121-138.

[7] J. Yi, J. Kim, L. Li, J. Morris, G. Lee, and P. Leclercq, “Real-time three dimensional vision,” in Advances in Computer Systems Architecture, Lecture Notes in Computer Science, Berlin: Springer-Verlag, vol. 3189 2004, pp. 309-320.

[8] S. Jin, J. Cho, X. D. Pham, K.M. Lee, S. –K. Park, and J. W. Jeon, “FPGA Design and Implementation of a Real-Time Stereo Vision System,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 20 (1), 2010, pp. 15-26

[9] L. Wang, M. Gong, M. Gong, and R. Yang, “How far can we go with local optimization in real-time stereo matching,” Proc. Third Int. Symp. On 3D Data Processing, Visualization, and Transmission, 2006, pp. 129-136.

[10] K.J. Yoon, and I.S. Kweon, “Adaptive support-weight approach for correspondence search,” IEEE Trans. PAMI, vol. 28 (4), 2006, pp. 650-656.

[11] A. Motten, L. Claesen, “A Binary Adaptable Window SoC Architecture for a Stereo Based Depth Field Processor,” in Proceedings IEEE VLSI-SOC-2010, 18th IEEE/IFIP International Conference on VLSI and System-on-Chip, Madrid, 27-29 September 2010, pp. 25 - 30.

[12] F. Tombari, S. Mattoccia, and L. Di Stefano, “Segmentation-based adaptive support for accurate stereo correspondence,” in Advances in Image and Video Technology, Lecture Notes in Computer Science, Berlin: Springer-Verlag, vol. 4872, 2007, pp. 427-438.

[13] A. Motten, L. Claesen, “An On-Chip Parallel Memory Architecture for a Stereo Vision System,” in Proceedings IEEE ECECS-2010, 17th IEEE International Conference on Electronics, Circuits, and Systems, Athens, 12-15 December 2010, 4 pages (accepted for publication).

[14] R. Zabih, J. Woodfill, “Non-parametric Local Transforms for Computing Visual Correspondence”, in Proc. European Conference on Computer Vision, Stockholm, Sweden, May 1994, pp. 151-158.