VoC: A Reconfigurable Matrix for Stereo Vision Processing

Ricardo P. Jacobi[1], Renato B. Cardoso[2] and Geovany A. Borges[2]

[1]University of Brasilia
Dept. of Computer Science
Brasilia, Brazil
ricardo@exatas.unb.br

[2]University of Brasilia
Dept. of Electrical Engineering
Brasilia, Brazil
renbarcar@ig.com.br, gaborges@ene.unb.br

Abstract

This paper presents a reconfigurable matrix VoC that can be applied to stereo vision computation. VoC accelerates block pixel matching by providing a highly parallel implementation of the Sum of Absolute Differences metric. Reconfigurability allows VoC to deal with different block sizes, ranging from a single 7x7 SAD computation to 9 simultaneous 5x5 block computations. The pipelined version mapped to Xilinx FPGA could be simulated at 158 MHz, producing 1,42 billion matchings per second.

1. Introduction

The term ”stereo” comes from the greek word ”stereos”, which means solid or hard. Stereo or binocular vision is a technique to compute the distance of elements based on at least two images that capture different views of objects much like the human eyes do. The depth perception provided by stereoscopic vision have several applications, such as robotic vision, virtual reality, remote operation and video conference, among others.

Several computer vision systems are real time applications that deal with large amount of data and perform computational intensive tasks. Among them, mobile and autonomous systems can not rely on conventional parallel architectures. New solutions based on dedicated hardware should be addressed in order to tackle its performance requirements.

Most stereo methods operate on two frames under known camera geometry [9]. The main problem in stereo vision is to find the pixels associated to the same point in the scene in each image. Local area correlation methods compare blocks of pixels to check it. The size of the block is a tradeoff between computing time and quality of the result. Smaller blocks are faster to process while larger ones result in more confidence in the matching.

A block from one image is matched against blocks inside a region in the other image. The cost of the matching computation is one of the main issues in stereo vision. Several methods were developed to accelerate this procedure. One alternative is to develop dedicated hardware to speed up critical tasks.

Dedicated hardware can be implemented either with ASICs (Application Specific Integrated Circuits), ASIPs (Application Specific Instruction Processor) or with PLDs (Programmable Logic Devices). While ASICs provide the highest speed up, they are very expensive and lack flexibility. ASIC fabrics are continuously increasing, demanding high production volume to be economically viable. ASIPs also suffer the same drawback when implemented in dedicated silicon. On the other hand, PLDs, and more specially FPGAs (Field Programmable Gate Arrays) are reaching logic complexity in the order of millions of gates and include dedicated blocks to speed up some critical operations like addition, multiplication, and signal processing. Although less optimized than ASICs, FPGAs are moving from a prototype platform to a final solution for a large class of problems.

The adoption of hardware description languages (HDLs) associated to design automation tools is the framework for dedicated hardware design. With FPGAs, one design can be more easily adapted to a new device by recompiling its HDL description for the new target. Moreover, changing, updating or fixing bugs in hardware can be much more easily done with programmable devices.

In the same way, adapting the hardware to differ-

1-4244-0054-6/06/$20.00 ©2006 IEEE

ent problems can be accomplished by reconfiguring the FPGA. If the configuration is held for all along the application lifetime then the reconfiguration is called static. Dynamic reconfiguration occurs when the circuit in the FPGA is changed during the application.

In this work, we present a reconfigurable architecture to accelerate the matching of blocks of pixels to speed up area correlation methods. The hardware system scans a region comparing a reference block against all the blocks in the region. Area correlation uses the sum of the absolute differences (SAD) to compare the blocks. The VoC structure can be seem as a pyramid where the base is composed by the blocks of pixels to be compared and the body is build with the operators (modulus, subtracter, adder).

The paper is organized as follows. Section 2 presents basic concepts about stereo vision and review related works. Section 3 describes VoC architecture. Section 4 discusses the implementation of the VoC in FPGA and section 5 is the conclusion.

2. Background

2.1 Area Correlation

Area correlation methods are usually applied to compute dense disparity maps. The term disparity refers to the difference in the projection of a point seen by the left and the right eye. In computer vision the term is usually interpreted as the inverse of the depth. Thus, a disparity map display the depth as seen by the stereoscopic vision. Area correlation relies on several assumptions about the physical world and the image formation process. The model adopted usually is based on epipolar geometry and suppose that the cameras are correctly calibrated.

Epipolar Geometry: An epipolar curve is obtained as illustrated on figure 1. Given point P in image plane L, the points on the line that passes through C_L and P that are projected in image plane R with focus C_R produce an epipolar curve e_R. Considering that the left and right images are obtained by a pinhole camera, the epipolar curve becomes a straight line.

The epipolar curve defines the region where the candidates projections can be found. Supposing that the cameras are aligned then the curve reduces to a line, i. e., the corresponding point in the other image lies on the same line, which reduces the search region simplifying the matching process.

Block matching: the sum-of-the-squared differences (SSD) and the sum-of-the-absolute differences (SAD) are the most used methods to compute the similarity of two fragments of the scene.

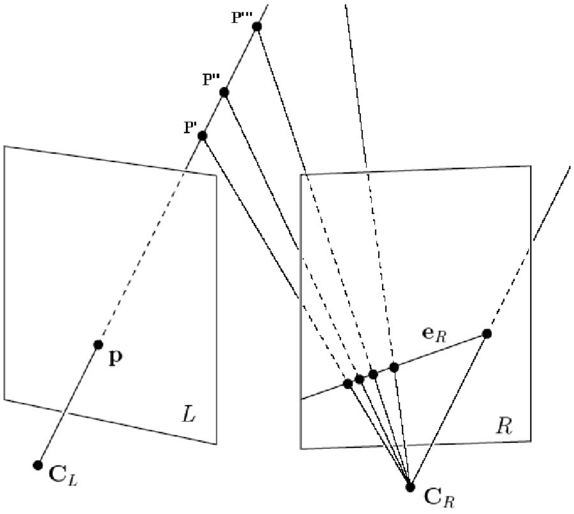


Figure 1. Epipolar curve obtained by the projection of p in several depths.

$$SSD = \sum_{x=1}^{n}\sum_{y=1}^{n}(I'_{x,y} - I_{x,y})^2 \qquad (1)$$

$$SAD = \sum_{x=1}^{n}\sum_{y=1}^{n}|I'_{x,y} - I_{x,y}| \qquad (2)$$

SAD requires less computation effort and provides a good approximation of the result. It can be implemented with simple operators, and the summation can be implemented with a regular structure in hardware.

Disparity Map: to build the disparity map the distance Z between the object and the cameras can be computed by $Z =$ - fb/d, where f is the focal distance in pixels, b is the distance between the cameras and d is the disparity in pixels of one point of the scene in the two images (figure 2).

2.2 Related Works

Several works focusing the acceleration of computer vision systems have been presented in the literature. There are some solutions based on reconfigurable hardware, such as [2, 10, 1]. Other alternatives focus on pure software solutions [9, 8, 5], (among several others), or even commodity graphic cards [13]. Some works try to exploit parallel architectures [4] or specialized libraries [6].

Since FPGA technology is evolving very fast, new solutions that exploit its growing capabilities must be

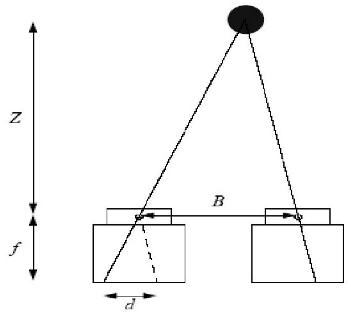


Figure 2. Camera and parameters in disparity computation.

developed and discussed. In [10] an FPGA architecture is proposed to accelerate computer vision algorithms, including SAD calculation. Its core is a coarse grain reconfigurable unit that includes a multiplier, a register bank and a ALU in each node, organized as an array of processing units. It was presented as an ongoing work, implemented on a FPGA board connected through a PCI slot in a PC.

The SAZAN architecture [2] uses 9 cameras and a polynocular stereo based image fusion method for 3D reconstruction. At the time of the publication (1999), it achieved the highest performance among real-time stereo systems, producing a dense disparity map at 20 MDPS (Million Depth-pixels Per Second). One of its main features was the use of weighted windows, where each pixel could be assigned a different weight.

Another reconfigurable hardware used to accelerate stereo image processing is the PARTS reconfigurable engine [12]. It is not dedicated to stereo image, but its architecture, composed by 16 Xilinx 4025 FPGAs and 16 one-megabyte SRAMs could attain a throughput of over 70 million point x disparity measurements per second. Those FPGAs are of course outdated today, but the proposed architecture was shown well adapted to stereo vision problem.

More recently, a commercial stereo vision system was presented [11] which attains very high disparity rates based on Census Stereo algorithm, with a dedicated silicon implementation of a stereo matching processor. Tyzx can attain 2.6 billion pixel disparities per second.

Moreover, a large number of research groups from academia and industry are working on computer vision. An internet search on the keywords *Computer Vision* returns a huge amount of information, which illustrates the importance and interest of this field of research.

3. VoC Architecture

VoC must scan two images to find the pixel correlations. It take one pixel in one image and must look for the corresponding one in the second image. The search region in the second image can be defined by the user, according to the calibration of the camera. Matching is performed taking the block that contains the pixel and computing the SAD against the blocks in the search regions. The system structure is shown in figure 3.

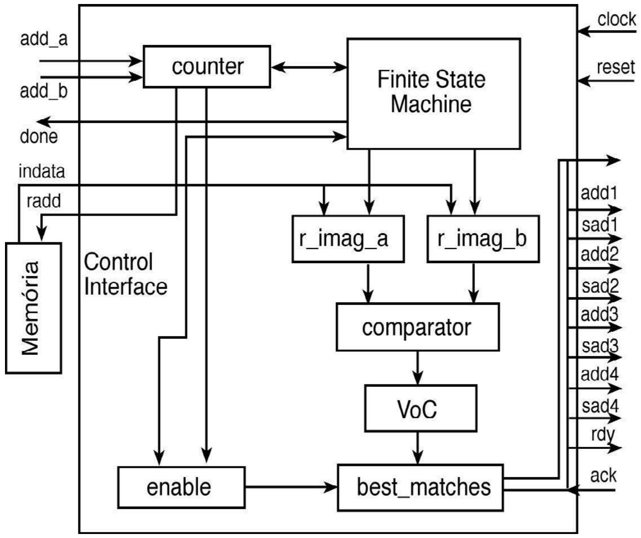


Figure 3. VoC Structure.

It presents three main interfaces: control interface; memory interface and output interface.

- **The control interface** provides the memory starting addresses, control signals and the indication that the image was processed.
- **The memory interface** fetches the data in memory to be processed.
- **The output interface** provides four addresses with the best matches for the given region.

The core of the system is a reconfigurable matrix to process blocks of 7 x 7 pixels. It can be configured to compute SAD of two blocks of 7 x 7 pixels or to compute 9 simultaneous SAD of 5 x 5 pixels block pairs. It is organized as a pipeline matrix of configurable nodes that can yield one 7 x 7 or nine 5 x 5 SADs per clock cycle. The distribution of the nine 5 x 5 blocks inside a single 7 x 7 is shown in figure 4.

An example of the structure of the VoC, reduced to a 3 x 3 matrix for simplification, is shown in figure 5.

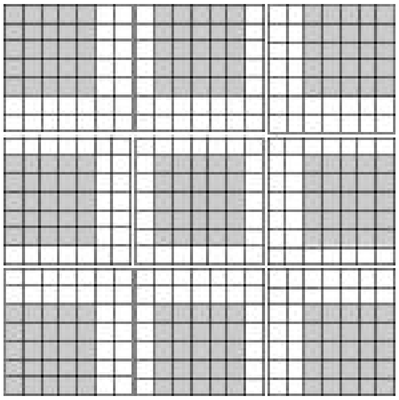


Figure 4. Distribution of 5x5 blocks inside a single 7x7 block.

It follows the basic dataflow structure of the odd-even transposition sorting algorithm [3]. It was shown that this structure can be used to implement some classes of nonlinear filter [7], like nonlinear statistical mean and homomorphic filters, order statistics and median filters and morphological filters. The inputs of the matrix are the modulus of the difference between the corresponding pixels.

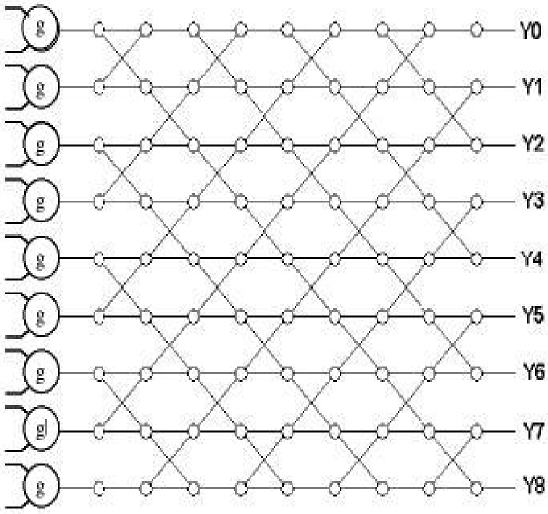
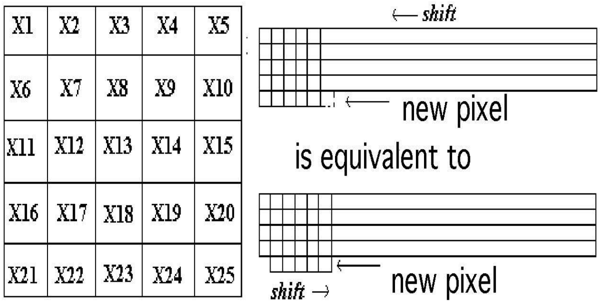


Figure 5. Example structure of a 3x3 VoC.

Memory access: The memory interface is responsible for fetching the blocks of pixels in the second image in order compute the SAD. Ideally, considering a pipeline implementation that delivers one SAD per cycle, it should be able to fetch one block of 7 x 7 pixels per cycle. Fetching 49 pixels simultaneously from the memory would be too complex. A simplification can be obtained using a simple arrangement that stores all the lines where the scanning window must traverse. This way, the displacement of the window can be obtained by reading a single pixel and shifting the window over the pixel lines, as illustrated in figure 6.



X1	X2	X3	X4	X5
X6	X7	X8	X9	X10
X11	X12	X13	X14	X15
X16	X17	X18	X19	X20
X21	X22	X23	X24	X25

Figure 6. Sliding window.

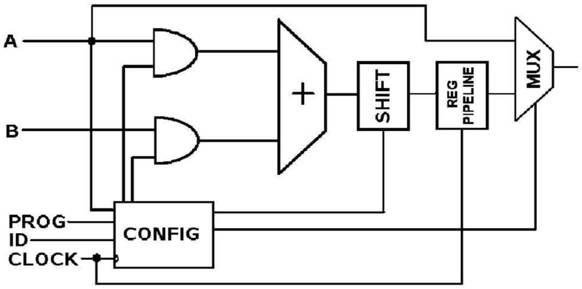


Figure 7. Basic cell.

Basic cell: The operations that can performed by the nodes are dedicated to SAD computation. Nodes operate on 8 bit data. The output of a node can be configured to be one of the following functions:

$$f(x,y) \in \{x, y, x+y, x/2, y/2, (x+y)/2\} \qquad (3)$$

where x and y are the node inputs and x comes from the upper left neighbor and y comes from the lower left neighbor. The modulus of the difference between the pixels are computed by an initial g function, as shown in figure 5. Each node has a registered output in order to pipeline the SAD computation. The structure of a basic node is shown in figure 7. The results of produced by the nodes are normalized to be kept within 8 bits range by discarding least significant bits.

Configuring the basic cells: The configuration of the basic cells is done by using a special signal PROG that puts all nodes in a row in configuration mode. All the nodes in a row become connected to the same data line. Then, a configuration word is put on the data line, which consists of the node ID and the node configuration code ($cfgcode$). The node that matches the node ID is then written with the $cfgcode$.

4. Results

To evaluate the size and performance of VoC, a matrix with 49 entries and 30 columns was designed in Verilog. It was synthesized for a Xilinx XC6000 FPGA using ISE Tools, provided by Xilinx University Program. This matrix is able to compute 9 simultaneous SAD for blocks of 5 x 5 pixels at each cycle, once the pipeline latency if filled. Synthesis results are as follows:

- Device: Virtex II xq2v6000cf1144-4
- Slices: 32459 (80% device)
- Flip-flops: 18196
- LUTs: 60433 (74% device)
- Frequency: 158 MHz

The throughput of VoC, considering that it yields 9 SAD per cycle, is

$$SAD_{rate} = 158Mhz \times 9SAD/cycle = 1.4GSAD/s \quad (4)$$

Thus, VoC is able to compute a peak rate of 1.4 billion matchings per second, which, depending on the search region size, could be used in real time distance computation for computer vision. The influence of the pipeline latency also depends on the size of the search region. The pipeline must be filled with $8 \times line\_length + 8$ pixels. For instance, considering a QCIF (176 x 144 pixels) search region the VoC latency is $8 \times 144 + 8 = 1160$ pixels. At 158 MHz, the latency would take $7.34\mu s$. However, once the latency is filled VoC computes 9 simultaneous matchings. Thus, if a complete search is undertaken, it would need $(176 \times 144) \times (178 \times 146) = 658639872$ comparisons, which results in $0.46s$ of processing time. Usually, however, we may suppose the cameras are aligned and the search may be reduced to horizontal blocks in the same row of the other image. In this case, the complete process could be approximated by

$$comparisons = \frac{area}{9} \times \frac{line\_length}{2} + latency \quad (5)$$

where the latency in this case is given by the time to fill the shift registers for the first time, as indicated above, and the time to displace the matrix (jump 3 lines each time), which is repeated $number\_lines/3$. In this case, the time for a complete scan is approximately $2.8ms$. This would allow for a real-time processing with small images.

To verify the result of applying VoC on a practical example, the disparity map of two images was computed with VoC. In this map, pixels are represented by 8 bits values. The disparity map gives an idea of the depth of the points in the objects with respect to the camera. The two original images are shown in the upper part of the figure 8, while below is presented the disparity map produced by VoC from the upper images. The left area on the disparity map shows a region that is closer to the camera than the area in the center of the map.

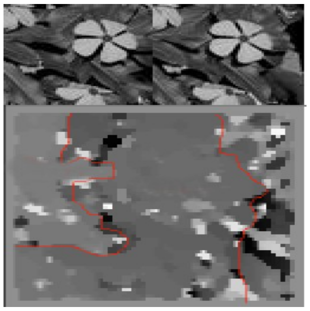


Figure 8. Disparity map example.

5. Conclusions

A parallel reconfigurable architecture for stereo video vision is proposed in the work. It implements the SAD computation either for blocks of 7x7 pixels or for blocks of 5x5 pixels. In this case, the matrix is able to compute 9 simultaneous SAD.

The structure of the matrix is conceived as a reconfigurable parallel processing unit to be used in image processing algorithms. The matrix could be optimized to stereo vision computation, reducing the number of nodes required, but loosing its flexibility. This matrix is to be integrated into a reconfigurable DSP which is part of a image processing SoC under development in our university. VoC allows the reconfiguration of the node operations and steering of signals through the matrix. This way, some classes of filter algorithms can also be implemented with the same structure.

In its general form, the matrix took a lot of FPGA resources. However, its configurability is not tied to

FPGA resources and its implementation in silicon will considerably reduce the area it requires and increase its speed.

The FPGA implementation could produce a SAD rate of 1.4 billion matchings per second at peak rate. This does not take into account the time to fill the pipeline. However, the larger is the search region, the closer the real rate is to this peak.

References

[1] A. Darabiha, J. Rose, and J. W. Maclean. Video-Rate Stereo Depth Measurement on Programmable Hardware. In *Computer Vision and Pattern Recognition (CVPR). Proceedings. 2003 IEEE Computer Society Conference on*, pages 203–210. IEEE, 2003.

[2] S. Kimura, T. Shinbo, H. Yamaguchi, E. Kawamura, and K. Nakano. A convolver-based real-time stereo machine (SAZAN). In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on*, pages 457–463, 1999.

[3] D. E. Knuth. *The Art of Computer Programming*. Addison/Wesley, Reading, Massachusets, 1973.

[4] A. F. Laine and G. C. Roman. A parallel algorithm for incremental stereo matching on SIMD machines. *IEEE Transactions on Robotics and Automation*, 13(1):123–134, 1991.

[5] G. V. Meerbergen, M. Vergauwen, M. Pollefeys, and L. V. Gool. A Hierarchical Symmetric Stereo Algorithm Using Dynamic Programming. *International Journal of Computer Vision*, 1-3(47):275–285, 2002.

[6] J. Mulligan, V. Isler, and K. Daniilidis. Trinocular stereo: A real-time algorithm and its evaluation. In *Stereo and Multi-Baseline Vision, 2001. IEEE Workshop on*, pages 1–8, 2001.

[7] I. Pitas and A. N. Venetsanopoulos. A New Filter Structure for the Implementation of Certain Classes of Image Processing Operations. *IEEE Transactions on Circuits and Systems*, 35(6):636–647, June 1988.

[8] B. Ross. A Practical Stereo Vision System. In *Computer Vision and Pattern Recognition (CVPR). Proceedings. 1993 IEEE Computer Society Conference on*, pages 148–153. IEEE, 1993.

[9] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal on Computer Vision*, 47(1/2/3):7–42, April-June 2002.

[10] C. Torres-Huitzil, S. E. Maya-Rueda, and M. Arias-Estrada. A Reconfigurable Vision System for Real-time Applications. In *Field-Programmable Technology, 2002. (FPT). Proceedings. 2002 IEEE International Conference on*, pages 286–289. IEEE, 2002.

[11] J. Woodfill, G. Gordon, and R. Buck. Tyzx DeepSea High Speed Stereo Vision System. In *Real Time 3-D Sensors and Their Use, 2004. IEEE Computer Society Workshop on*, pages 1–5, 2004.

[12] J. Woodfill and B. V. Herzen. Real-Time Stereo Vision on the PARTS Reconfigurable Computer. In *Field-Programmable Custom Computing Machines, 1997. IEEE Symposium on*, pages 242–250, 1997.

[13] R. Yang and M. Pollefeys. Multi-Resolution Real-Time Stereo on Commodity Graphics Hardware. In *Computer Vision and Pattern Recognition, 2003. IEEE Computer Society Conference on*, pages 211–218, 2003.