AN EFFICIENT RECONFIGURABLE ARCHITECTURE TO IMPLEMENT DENSE STEREO VISION ALGORITHM USING HIGH-LEVEL SYNTHESIS.

Mario Alberto Ibarra-Manzano *, *Michel Devy, Jean-Louis Boizard, Pierre Lacroix and Jean-Yves Fourniols*

CNRS; LAAS; 7 avenue du Colonel Roche, F-31077 Toulouse, France
Université de Toulouse; UPS, INSA, INP, ISAE; LAAS-CNRS : F-31077 Toulouse, France

email: {maibarra, michel, boizard, fourniol}@laas.fr

ABSTRACT

This article presents a reconfigurable architecture to calculate a dense disparity map of two stereo images based on census transform. This architecture is simplified and efficient as a result of binary operations and integer arithmetic used by census transform. Our architecture was prototyped using GAUT which is a practical tool to develop high-level synthesis. We take advantage of GAUT rapid prototyping to implement different architectures and to make a general comparison among them, that lets us to optimize the architecture, and at the same time, to improve the system's performance. The optimization and the resource saving rend our architecture an interesting option to solve the problem of stereo vision in real time, quite used in autonomous navigation.

1. INTRODUCTION

The stereo vision process intends to reconstruct the 3D information of a scene from two different images. These images are captured from two cameras that are separated by a previously established distance called baseline. However, in some cases, depth information is unreliable or insufficient, e.g. in mobile robotics or surface recovering for automatic 3D model acquisition. Furthermore, applications such as robotics platforms and autoguided vehicles require real-time performances that can not be accomplished by conventional computers.

Different real-time stereo systems have been implemented by custom hardware recently. The parallelism provided by this hardware allows users to meet the performance constraints as well as minimising either the power or cost per unit of the system. Nevertheless, the major problem with hardware implementations is the time and cost required by the design and fabrication stages and the non recoverable costs of production. Thus, hardware-based stereovision is more appropriate for systems that are either cost-insensitive, where the high development cost is tolerated, or systems that are produced in massive volumes, where the cost is absorbed by the lower cost per unit of the hardware implementation.

Field-Programmable Gate Array (FPGA) has enabled the creation of hardware designs in standard, high-volumen parts, thereby amortizing the cost of mask sets and significantly reducing time-to-market for hardware solutions. However, engineering cost and design time for FPGA-based solutions still remain significantly higher than software-based solutions. Designers must frequently iterate the design process in order to meet system performance requirements while simultaneously minimizing the required size of the FPGA. Each iteration of this process takes hours or days to complete.

Traditionally, FPGA systems design was developed by hardware engineers using a Hardware Design Language. Although many proprietary HDLs have been developed, Verilog and VHDL are the most used languages in FPGA systems design. These languages require wide knowledge and specialized techniques design to obtain system desired performance, so that, these languages are not easy to manipulate. Furthermore, if the algorithm is complex and/or system constraints are hard, system development could spend several years. So that, over the last few years there has been several approaches at translating algorithmic developed in languages like C or SystemC into hardware descriptions, this approach is called high-level synthesis. Many high-level synthesis tools can be used: Catapult-C, SystemC, HandelC, Impulse C, GAUT, SPARK, among others.

In this paper, we present the analysis of different architectures to solve the problem of stereo vision systems in real-time. We are interested in stereo correlation approaches with a special stress in the development of reliable algorithms, and also that they be suitable for hardware implementations. Next section describes passive stereo vision algorithm, used

*This work is supported by the Mexican National Council for Science and Technology (CONACyT), scholarship number 183807, the Mexican Ministry of Education (SEP) and the Mexican Government.

978-1-4244-3892-1/09/$25.00 ©2009 IEEE

Authorized licensed use limited to: UNIVERSITATSBIBLIOTHEK PADERBORN. Downloaded on March 29,2010 at 08:24:57 EDT from IEEE Xplore. Restrictions apply.

to obtain depth information, which is based on the census transform. In the section 3, we introduce GAUT, the high-level synthesis tool used to implement the system. Several comparative tables showing the results of performance tests to proposed architectures are presented at the end of this document.

2. OVERVIEW OF PASSIVE STEREO VISION

In computer vision, stereo vision intends to recover depth information by two images of the same scene, a pixel in one image corresponds to a pixel in the other, if both pixels are projected along of sight of the same physical scene element. Also, if the two images are spatially separated but simultaneous, then computing correspondence determines stereo depth. There are two main approaches to process the stereo correlation: feature-based and area-based approaches. In this work, we are rather interested in area-based approaches, because it proposes a dense solution that produces high-density disparity maps. Furthermore, it has extremely regular algorithmic structure and it can be used to propose a convenient hardware architecture. The census transform is the area-based approach chosen to solve the stereo vision problem [1]. The algorithm of dense stereo vision based on Census transform process left and right images independently, the global algorithm is illustrated in Fig. 1. In order to decrease the hardware complexity and the number of operations during image processing, the epipolar restriction is used. Due to this restriction, in a two cameras arrangement with parallel principal axes, the epipolar lines are constrained to lie along scan lines or columns. So the difference in cameras position is purely reduced to horizontal or vertical translation. Any pixel visible in an epipolar line of one image will appear at a shifted position in the corresponding epipolar line of the other image, assuming that the pixel is not hidden in the second image.

3. HIGH-LEVEL SYNTHESIS

Lately, common sequential algorithm synthesis to FPGA circuits has achieved big popularity thanks to their speedup for numerous applications. Nevertheless, digital circuits design needs both advanced knowledge and high development time in order to achieve required performance and efficiency. Also, rapid prototyping is considered as a key to speed up the system design [2]. In this context, high-level synthesis is a complementary approach that trades off hardware performance for reduced design time [3]. High-level synthesis is analogous to software compilation transposed to the hardware domain. In the next subsection, one tool used to high-level synthesis are described.

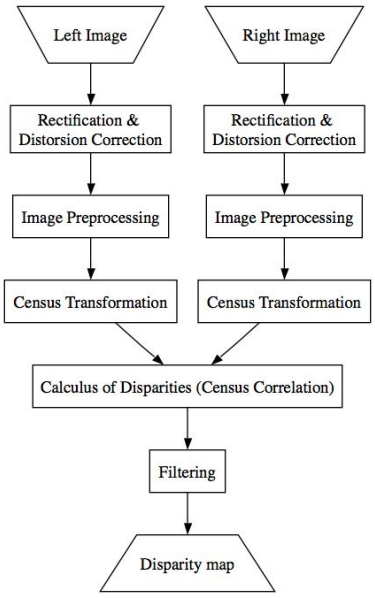


Fig. 1. Stereovision algorithm.

3.1. GAUT

Automatic Generator of Processing Units (GAUT) is an academic open-source high-level synthesis tool dedicated to signal and image processing applications under real time execution constraints [4]. Starting from an algorithmic bit-accurate specification written in C/C++, GAUT extracts the potential parallelism before processing the allocation, the scheduling and the binding tasks. Mandatory synthesis constraints are throughput and the clock period while the memory mapping and the I/O timing diagram are optional. GAUT next generates a potentially pipelined architecture composed of a processing unit, a memory unit and a communication with a GALS/LIS interface [4].

4. RESULTS AND CONCLUSIONS

In this section, we present several performance tests of our design. First of all, our architecture was prototyped using GAUT, Quartus II and ModelSim. After, it was synthesized for a EP2C35F672C6 Cyclone II family device from Altera. Each module architecture implemented in the FPGA was tested to several cadency time.

In Tables 1 to 3 we present, that we consider, the three most representatives cadency values, labeled as design 1, 2 and 3, for each module and their most important selecting characteristics. To better interpret the information presented in these tables and, so that, to know how we have selected our final architecture, first of all we will describe the relation among the characteristics in order to get the best compromise between area and performance. If performance

Authorized licensed use limited to: UNIVERSITATSBIBLIOTHEK PADERBORN. Downloaded on March 29,2010 at 08:24:57 EDT from IEEE Xplore. Restrictions apply.

Table 1. Comparative table from the arithmetic mean filter.

Characteristics	Design 1	Design 2	Design 3
Cadency (ns)	20	30	40
Performance (fps)	160	100	80
Logic elements	118	120	73
Comb, functions	86	72	73
Ded. log. registers	115	116	69
# Stages in pipeline	3	2	2
# Operators	2	2	1
Latency (μs)	25.69	38.52	51.35

Table 2. Comparative table from the census transform.

Characteristics	Design 1	Design 2	Design 3
Cadency (ns)	40	80	200
Performance (fps)	80	40	15
Logic elements	2,623	1,532	1,540
Comb. functions	2,321	837	864
Ded. log. registers	2,343	1,279	1,380
# Stages in pipeline	48	24	10
# Operators	155	79	34
Latency (μs)	154.36	308.00	769.50

Table 3. Comparative table from the census correlation.

Characteristics	Design 1	Design 2	Design 3
Cadency (ns)	20	40	80
Performance (fps)	160	80	40
Logic elements	1,693	2,079	2,644
Comb. functions	1,661	1,972	2,553
Ded. log. registers	1,369	1,451	1,866
# Stages in pipeline	27	12	8
# Operators	140	76	46
Latency (ns)	290	160	100

Table 4. Comparative table from different architectures.

	Design 1	Design 2	Design 3
Measure	Census	Census	Census
Image size	640×480	640×480	320×240
Window size	17×17	17×17	13×13
Disparity max	64	64	20
Performance	40	130	40
Latency (μs)	206	274	–
Logic elements	$6,977$	$11,100$	$26,265$
Memory size	109 Kb	174 Kb	375 Kb

decrease then cadency increase, as consequence, both operators and stages in pipeline decrease. The rest of characteristics follow a less evident relationships. That is, the number of logic elements are a direct consequence of both combinational function and dedicated logic register number. At the same time, combinational functions depend strongly on number of operators and weakly on the number of states in the state machine that consequently depends on cadency time. Contrary, dedicated logic registers depend strongly on the number of states in the state machine and weakly on the number of operators. Finally, latency depends on the number of operators, number of states in pipeline and in a stronger way on the cadency time due to design architecture selected in this project. The results showed in the tables was performed by using an image size of 640×480 pixels, with a 3×3 and 7×7 pixels for mean and census transform window size respectively, and a maximal disparity measure of 64 pixels, for a 100 Mhz frequency clock in the FPGA.

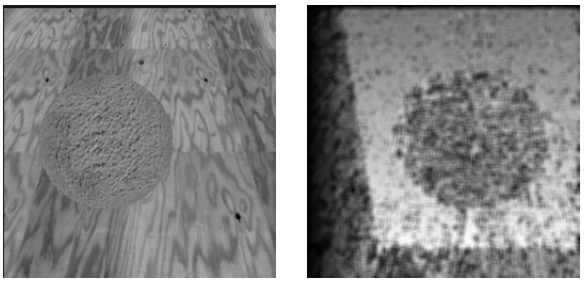
Taking into account all described relationships, we choose design 3 to develop the mean filter module, since it has the best compromise between area-performance. In the same way, design 2 was chosen to develop census transform. Design 3 was chosen to develop the census correlation since it was the best option to link this module to last two. Some synthesis results of FPGA device are summarized as follows: the architecture implemented required 6,977 logic elements, and required memory is 112,025 bits. The quantity of logic elements represents only the 21% of total capacity in the device while memory size represents the 23%.

We present FPGA implementation to compare three architecture designs. Their most important characteristics are presented in the first column of the Table 4. Second column shows the performance results of our design, labeled as Design 1. The rest of the columns show the results of two designs that we are labeled as Design 2 and 3 implemented by different authors (refers to [5], and [6] for technical details of implementation respectively). Obviously, all of them are FPGA implementations which calculate the disparity map of two stereo images. Our design can be directly compared with Design 2 and 3 because they use the census transform algorithm to calculate the disparity map. We obtain three important improves with respect to Design 2, these are less latency time, number of logic elements and memory size. The cost of these improves is reflected in the performance and in our case is less. Although Design 3 uses a smaller image and window size and a lower disparity max, the performance is the same than our design, also, the logic elements and memory size that it uses are the highest of all designs.

The stereo vision architecture based on census trans-

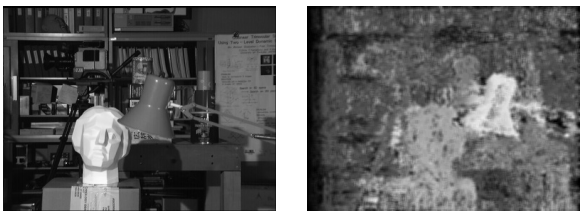
Authorized licensed use limited to: UNIVERSITATSBIBLIOTHEK PADERBORN. Downloaded on March 29,2010 at 08:24:57 EDT from IEEE Xplore. Restrictions apply.

form, resulted of that choices, was tested on three pairs of stereo images. Fig. 2-a shows one synthetic image of the first pair. Fig. 2-b shows its corresponding result from FPGA implementation. This shows in the same plane most of the points that forms the circle (similar gray level in the circle) and in different gray levels, the background plane. In order to explain background results, we want to remind that correlation process is carried out from left to right image, thereby the positions on resulted image are more similar than positions on the left one. Furthermore, dark pixels on the left of the Fig. 2-b correspond to not found points from the left image to the right one.



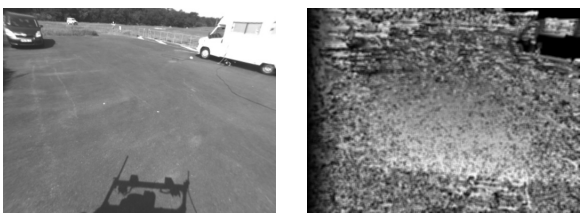
(a) (b)

Fig. 2. Stereo image and result, a) Original right image, b) Simulation from the FPGA implementation



(a) (b)

Fig. 3. Stereo image and result, a) Original right image, b) Simulation from the FPGA implementation



(a) (b)

Fig. 4. Stereo image and result, a) Original right image, b) Resulted image from the FPGA implementation

Fig. 3-a shows the well-known image from the University of Tsukuba which we also use to test our architecture. These stereo image shows several planes of depth, at the same time, output disparity image shows them in the Fig. 3-b, as in the later case, dark pixels on the left correspond to not found pixels. Despite of luminosity controlled conditions, this image gives our architecture an important evaluation of its performance. In last test we use outdoor images acquired from a mobil robot (Fig. 4-a). The image in the Fig. 4-b shows in light levels of gray, a part of the road as the nearest zone of the scene. As gray levels became darker, it represents more distant zones. Also, we note the white vehicle on the original images depicted in black level on the image result which deserves a more detailed explanation. Rather than it means a so far position in the scene, it demonstrates that no textured objects could not be easily detected in the stereo vision algorithms. So black pixels represent the unidentified pixels and some light gray levels depicted inside of the vehicle corresponds to the lateral windows which they show indeed a texture. A possible solution of this detection failed would be to use a bigger window searching used by census transform. Although, as we said before, that means to trade off between consumed resources and latency time. Even though, this is an interesting result which can be exploited in the case of mobile robotic applications.

5. REFERENCES

[1] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *ECCV '94: Proceedings of the Third European Conference on Computer Vision*, vol. II. Secaucus, NJ, USA: Springer-Verlag New York, Inc., June 1994, pp. 151–158.

[2] E. Casseau, B. Le-Gal, P. Bomel, C. Jego, S. Huet, and E. Martin, "C-based rapid prototyping for digital signal processing," in *13th European Signal Processing Conference*, S. O. Ltd., Ed., Antalya, Turkey, September 4-8 2005.

[3] G. Stitt, F. Vahid, and W. Najjar, "A code refinement methology for performance-improved synthesis from c," in *ICCAD '06: Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design*. New York, NY, USA: ACM New York, NY, USA, November 5-9 2006, pp. 716–723.

[4] E. Martin, O. Sentieys, H. Dubois, and J. L. Philippe, "Gaut: An architectural synthesis tool for dedicated signal processors," in *Proceedings of the IEEE European Design Automation Conference*. IEEE, September 20-24 1993, pp. 14–19.

[5] A. Naoulou, J.-L. Boizard, J. Y. Fourniols, and M. Devy, "A 3d real-time vision sytem based on passive stereovision algorithms: Application to laparoscopic surgical manipulations," in *Proceedings of the 2nd Information and Communication Technologies, 2006 (ICTTA)*, vol. 1. IEEE, April 24-28 2006, pp. 1068–1073.

[6] C. Murphy, D. Lindquist, A. M. Rynning, T. Cecil, S. Leavitt, and M. L. Chang, "Low-cost stereo vision on an fpga," in *FCCM '07: Proceeding of the 15th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 333–334.

Authorized licensed use limited to: UNIVERSITATSBIBLIOTHEK PADERBORN. Downloaded on March 29,2010 at 08:24:57 EDT from IEEE Xplore. Restrictions apply.