# VHDL Description of a Synthetizable and Reconfigurable Real-Time Stereo Vision Processor

CARLOS CUADRADO AITZOL ZULOAGA JOSÉ L. MARTÍN JESÚS LÁZARO
JAIME JIMÉNEZ
Dept. of Electronics and Telecommunications
University of the Basque Country
Alda. Urquijo S/N, 48013 Bilbao
SPAIN

*Abstract:* - This paper describes a reconfigurable digital architecture to compute dense disparity maps at video-rate for stereo vision. The processor architecture is described in synthetizable VHDL and, by means of the reconfigurability, the hardware requirements are optimized for different image resolutions and matching scenarios. To have a configurable description of a stereo processor provides the entity to design hardware stereo matching systems, implementing by incremental design disparity consistence algorithms, multi-stereo correlations or multi-scale algorithms. The results of the hardware synthesis of this code have being implemented in several reconfigurable devices. We show the results of the synthesis and its implementation cost in logic elements and delays.

*Key-Words:* - Stereo Vision, Real-Time, VHDL, Reconfigurable Logic, FPGA, SoC

## 1 Introduction

Computational stereo vision for extraction of three-dimensional scene structure has been an intense area of research in the last decade [1]. In this time, significant improvements have been carried out in the development of the stereo matching algorithms and their digital implementations [2, 3, 4].

Although the digital stereo vision is a mature discipline, the matching problem is still a very expensive computational task. In addition, no general solution exits to the matching problem and several approach have being used in order to optimize the unreliable matches due to occlusions, photometric distortions or camera noise [5, 6].

In many applications it needs recovery the three-dimensional structure of the environment at video-rate. The digital stereo vision is a useful solution for these applications when it is not possible to use active light systems. The researchers have applied the stereo vision to robotics autonomous navigation, people and object tracking, gaming and telepresence applications [7], and others. Actually a main target of the digital stereo vision is to reduce the size, the cost and the power requirements to employ this technique in small and autonomous applications.

For real-time stereo vision applications it needs to appeal to specific and extensive parallel hardware. The real-time implementations use of special purpose hardware, like arrays of Digital Signal Processors (DSP) [8] or several Field Programmable Gate Arrays (FPGA) [9]. In the last years, some implementations of the real-time stereo vision algorithms have used general-purpose microprocessors [4] with a relative success.

The major advantages of the design on FPGA are the low cost of prototype and the fast design cycle. FPGA implementations allow to exploit the parallelism and the pipeline usual in vision algorithms. In addition, we can generate high specific and parameterizable hardware.

The growing of the logic elements and capabilities embedded in the programmable logic devices make possible to implement a complex video-rate stereo vision system. The reconfigurable logic also allows to use the same hardware device for vision and non-vision systems, modulating, through a parameterizable description, the hardware resources dedicate to the vision algorithms.

In this paper we propose the implementation of a Real-Time Stereo Vision Processor (RTSVP) on a reconfigurable logic device using VHDL hardware description language. The VHDL description provides the capacity of generate high specific and optimize hardware. Every logic blocks of the RTSVP are tunable in order to adjust the hardware used by the stereo vision algorithm and the necessities of a specific practical application. In our architecture, the VHDL description of the hardware is tunable with the major set of usual parameters involve in area correlation stereo vision like size of the image pairs, size of the correlation window, disparity limit and image intensity resolution and correlation resolution. In the RTSVP the most of these parameters are only limited by the amount of dedicated hardware, maintaining in every configurations a high pixel clock.

The FPGA devices and the VHDL also make possible an quickly block incremental design. In a multi-scale matching algorithm, we use a RTSVP per scale. In this configuration, each one of the RTSVP computes a different size of the correlation window. In multi-baseline stereo matching, we can dedicate one RTSVP per pair of cameras and reducing the disparity limit in function of the distance between the cameras.

In the design of the RTSVP we assume that the stereo pair is in ideal form, i.e. no exist optics deformations and epipolar lines lying on image horizontal lines.

The reminder of this paper is structured as follows: Section 2 discusses the previous works and the real-time implementations requirements. Section 3 discusses the area-correlation based algorithm. Section 4 describes the digital design of the RTSVP and its implementations cost. Section 5 discusses the building of the stereo vision system by aggregation of RTSVPs to process multi-scale and multi-resolution algorithms. We conclude in section 6 and offer our impressions of the current and future trends in stereo vision hardware architectures.

## 2 Related Real-Time Stereo Vision Implementations

In parallel with the development of the stereo vision algorithms the researchers dedicated considerable efforts to the real-time implementations. In 1993, Faugeras et al. presented a real-time implementation based on an array of DSPs and FPGAs [2]. They implemented a normalized area correlation algorithm and left/right matching consistency, processing up to 256 x 256 pixel images at 3.6 fps. In the same year, Webb implemented a fast multi-baseline stereo algorithm on the CMU Warp machine [10]. In this prototype were used 64 iWarp processors to achieve 15 fps with 256 x 240 pixel images [8]. Also at CMU, Kanade et al. presented the first 30 fps stereo architecture. Like the iWarp implementation a SSAD algorithm was used. The Kanade's CMU Stereo Machine was made up of custom hardware and an array of eight C40 DSPs [3].

In 1997, Konolige reported a real-time stereo system at SRI International called Small Vision System (SVS) [4]. The SVS was the first stereo low power system, being capable of processing 320 x 240 pixel images at 12 fps on a 233 MHz Pentium II with a SAD algorithm. In 2001, SRI's SVS runs at 30 fps on a 700 MHz Pentium III.

**Table 1**: Real-Time Stereo Implementations.

| Real-Time System | Image Size | Frame Rate | Algorithm |
|---|---|---|---|
| (1993) INRIA | 256 x 256 | 3.6 fps | Normailized Correlation |
| (1993) CMU iWarp | 256 x 240 | 15 fps | SSAD |
| (1996) CMU Stereo Machine | 256 x 240 | 30 fps | SSAD |
| (1997) SRI SVS | 320 x 240 | 30 fps | SAD |
| (1997) PARTS | 320 x 240 | 42 fps | Census |
| (1999) SAZAN | 320 x 240 | 20 fps | SSAD |
| (2001) SRI SVS | 320 x 240 | 30 fps | SAD |

A new type of implementation was developed by Woodfill and Von Herzen [9]. They implemented a census matching algorithm on a custom hardware called PARTS engine. The PARTS engine made up of 16 XC4025 FPGAs and PCI card to process 320 x 240 pixel images at 42 fps.

In 1999, Kimura et al. reported a stereo machine called SAZAN [7]. Like CMU stereo machine, they implemented a multibaseline stereo algorithm increasing the number of cameras up to nine. With the extensive use LSI filters implemented on FPGAs was capable of processing 320 x 240 images at 20 fps. In the table 1 we present a resume of these significant works.

In that last decade, the researchers have selected mainly Sum of Absolute Differences (SAD) matching algorithm and census matching algorithm for their real-time stereo vision applications. Although, the census matching algorithm consumes less resources, the behavior of the SAD algorithm is more robust, overcoat in a SAD multi-scale algorithm [11].

Seeing the advances of the real-time implementations, the objectives of the present work have been: 1) Generate high resolution - high frame rate stereo vision matching architecture; 2) Design a low cost and low power system, to include the stereo vision in small industrial applications and autonomous navigation; 3) Generate a flexible digital architecture, optimized and standard as be possible; and 4) Easy incremental block design of complex stereo vision systems.

## 3 The Matching Algorithm

We propose an effective and fast digital architecture based on area correlation algorithms [2, 3]. The algorithm evaluates the correlation between two windows, as the sum of the absolute intensity differences for each pixel in the correlation window. Denoting by $I_L(x,y)$ and $I_R(x,y)$ the intensity values at the pixel $(x,y)$ in the left image and the right image respectively, the value of the correlation per pixel would be:

$$c(x,y,d) = |I_L(x,y) - I_R(x+d,y)| \tag{1}$$

If the correlation window has dimensions $(2n+1) \times (2m+1)$. The measure of the correlation associate to the central pixel corresponds to the equation (2). The indexes, which appear in the equation, vary between $-n$ to $+n$ for the $i$-index and between $-m$ to $+m$ for the $j$-index.

$$C(x,y,d) = \sum_{i,j} c(x+i,y+j,d) \tag{2}$$

To calculate the correlation value, it separates the equation (2) in a sum of columns $VC$ as follows:

$$C(x,y,d) = \sum_{i} VC(x+i,y,d) \tag{3}$$

And each sum in a column, $VC(x,y,d)$, can be described by the sum of the every absolute differences of the pixels in the right and left images that belong to a specific column, as is expressed by the equation (4).

$$VC(x+i,y,d) = \sum_{j} c(x+i,y+j,d) \tag{4}$$

Expanding the equation (2) is possible define a iterative form of this equation, that is, the equation (6). The iterative form of $VC$ function is the equation (5).

$$\begin{aligned} VC(x,y,d) = VC(x,y-1,d) + \\ c(x,y+m,d) - c(x,y-m-1,d) \end{aligned} \tag{5}$$

$$\begin{aligned} C(x,y,d) = C(x-1,y,d) + \\ VC(x+n,y,d) - VC(x-n-1,y,d) \end{aligned} \tag{6}$$

For one disparity displacement is possible to know iteratively the correlation in a window adding the head column sum and subtracting the last column sum. Therefore, we can obtain iteratively the window correlation value of two pair of images for a certain disparity displacement. The disparity map for a pair of stereo images is obtained by the comparison of the correlation values in every disparity displacements, that is:

$$\delta(x,y) = d \rightarrow \min\{C(x,y,d)\} \tag{7}$$

In the next section we discuss the implementation of the purposed algorithm in a fast parallel architecture. In order to reduce the digital architecture in the calculus of the algorithm equations, we use integer arithmetic and only in the last stages of the process the result is rounded to accommodate to specific bus width.

## 4 The RTSVP Architecture

The RTSVP receives a pair of digital image sequence. Both signals have the same pixel-clock and synchronism. The input video-rate images have $N \times M$ pixels with a intensity level of $I_B$ bits. The correlation window size is $(2n+1) \times (2m+1)$.

The disparity limit $D_L$, is measured in pixels which imposes a limit in the correct perception of depth by the correlation process. The RTSVP is divided in three main blocks. The first block calls the Correlation Window Delayers (CWD). This block is composed by a series of registers which produces the inputs to the Stereo Disparity Correlators (SDC). Those are the second block and the core of the matching calculus. The last block, called Disparity Comparator (DC) produces the disparity measure associated to the minimum difference value. A schematic view of the RTSVP architecture is presented in the figure 1.

### 4.1 The Correlation Window Delayers

The CWD block consists in FIFO memories and registers and its function is to order the pixel intensity values for the correct evaluation of the correlation measure. The relative position between the correlation windows in the left and right images depends of if the stereo vision system is parallel or convergent, and if the reference image is the right or the left image. The CWD receives the stereo video sequence in $I_B$ bits and produces $2D_L + 2$ lines of $I_B$ bits, corresponding $2D_L$ lines to the successive displacements of the correlation window, i.e. two lines per disparity unit. The another two output lines of CWD block leave from the reference image. The configurable parameters in this block are $N$, $m$ and $D_L$. A schematic of the CWD block is represented in the figure 2.

### 4.2 The Stereo Disparity Correlators (SDC)

The second block are the Stereo Disparity Correlators (SDC). The SDC is an architecture based on the iterative equations (5) and (6). The SDC takes four intensity

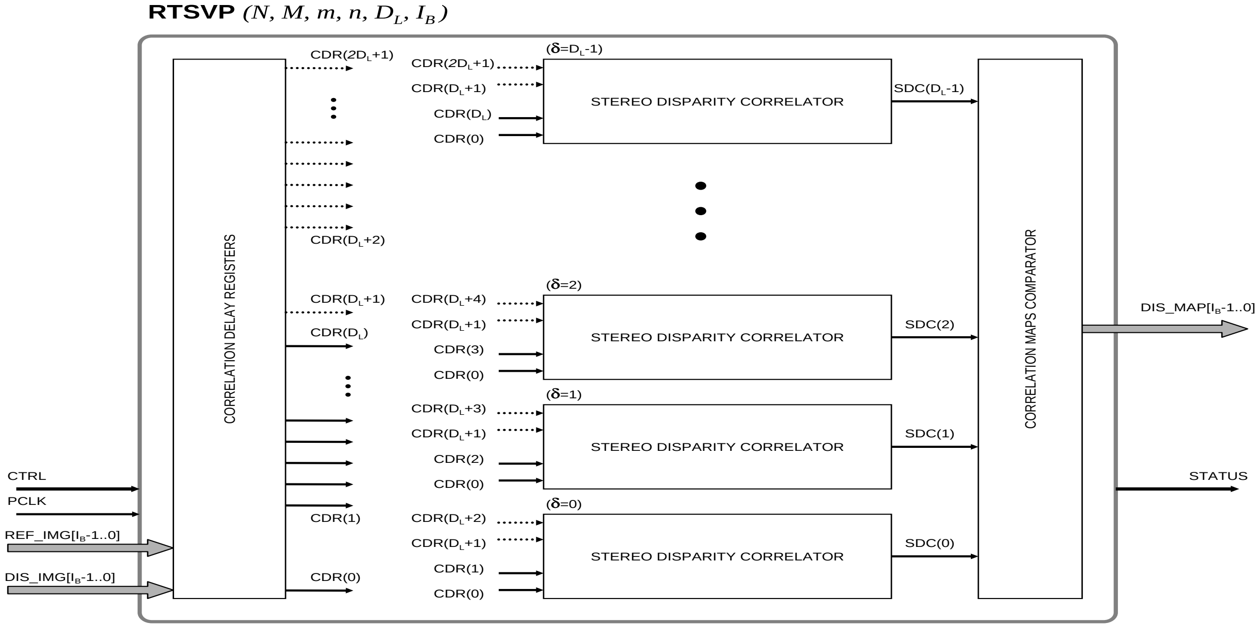

**Fig. 1**: Architecture of the RTSVP.

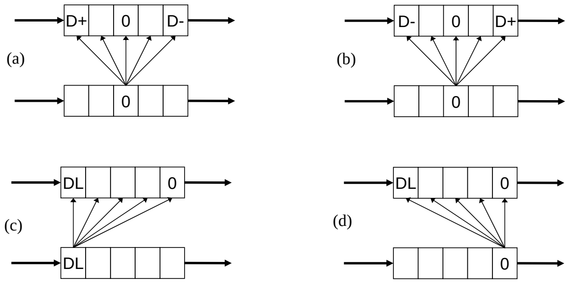

**Fig. 2**: Relation of the outputs from the Disparity Delay Registers and the Stereo-Vision System. (a) Disposition in convergent stereovision, image right over image left. (b) image right over image left. Disposition in parallel stereovision, image right over image left. (c) Disposition in parallel stereovision, image right over image left.

values, two in reference image and two in the image crossed in disparity. Each SDC computes the correlation value for a given window size $n \times m$ and for a given disparity. This means that it needs a SDC per disparity value. Thus, each SDC generates a correlation value per pixel clock.

As the figure 3 shows, the SDC block is composed by six main sub-blocks. The AD block executes the absolute difference between the selected pixels. The VC block produces the sum of the absolute differences in a column of the correlation window for every pixel clock transition, being the digital block that executes the equation (5).

The VC block output value is stored in two FIFOs. Both FIFO memories store the $N$ column sum of the matching algorithm. The VC block receives the feedback of the column sum generated in the previous computed line. The FIFO data bus has a width of $\log_2(2m+1)+I_B$ bits. This is the minimum number of bits to store any value produces by the VC block.

The C block generate the total absolute value of a correlation window per pixel. This receives the last total correlation window value, the new VC value and the $m+1$ separated column sum. Its architecture is similar to the VC block but operates with $\log_2\{(2n+1)(2m+1)\}+I_B$ bits.

The calculus of C block is stored in one register. After registering it needs to reduce the number of bits of C block output. In other manner, the number of lines that arrives to disparity comparator would be too high, although this feature is also configurable. In this last stage we reduce the length of bits of C to $I_B$ bits displacing its value in $\log_2((2n+1)(2m+1))$ bits. Thus, the total number of input lines to comparator block is $D_L \cdot I_B$ bits. The $D_L$ buses from the bank of SDCs can be interpreted us a collection of $D_L$ correlation video images.

### 4.3 The Disparity Comparator Block

The comparator block compares the correlation images that receives from the SDC blocks and produces the computed disparity value. The DC block is composed of several length FIFO and some comparators and multiplexers. In this case, we have implemented a simple comparison function based on the minimum of the cor-

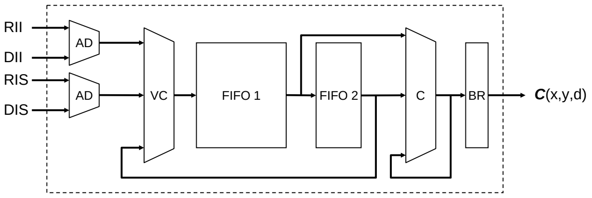

**Fig. 3**: Block diagram of a SDC block.

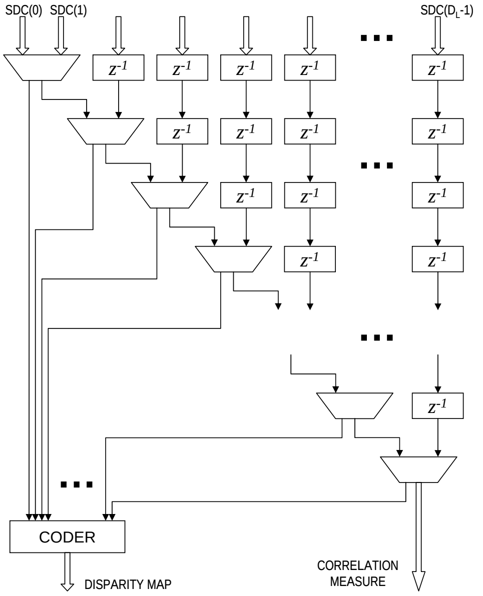

**Fig. 4**: Block diagram of DC block.

relation window. The figure 4 illustrates the internal architecture of the DC block. This architecture is fully extensible with the number of input buses and their width.

### 4.4 Results of the synthesis of the RTSVP

In the design with reconfigurable logic a main issue is the amount of hardware available for the processing. This hardware resources include logic and memory on-FPGA, off-FPGA available memory and memory access bandwidth. The table 2 shows the hardware resources embedded in several FPGA in Logic Elements (LE), general purpose I/O pins and on-FPGA available memory (ESB).

**Table 2**: Embedded Resources in Altera FPGAs

| Name | LE | I/O | ESB |
|---|---|---|---|
| 20KE300EQC | 11520 | 144 | 147456 |
| EP2C50F484C6 | 51785 | 288 | 594432 |
| EPS260F672C5 | 48352 | 493 | 2544192 |

**Table 3**: Results of the Synthesis for the Main RTSVP Parameters.

| Name | $I_B$ | $D_L$ | $M,N$ | $m,n$ | LE | ESB | fmax |
|---|---|---|---|---|---|---|---|
| 20K300EQC | 6 | 16 | 128 | 3 | 1323 | 32796 | 82.4 |
| | 8 | 16 | 128 | 5 | 2974 | 61648 | 81.5 |
| EP2C50F484C6 | 6 | 32 | 256 | 3 | 5016 | 104508 | 78.7 |
| | 6 | 32 | 256 | 5 | 5278 | 140092 | 77.9 |
| | 8 | 32 | 256 | 5 | 6667 | 175184 | 75.9 |
| EPS260F672C5 | 8 | 64 | 512 | 5 | 10540 | 557904 | 74.3 |
| | 8 | 64 | 1024 | 5 | 10802 | 1082192 | 73.3 |

The table 3 presents the hardware resources use up by the different configurations of the stereo vision processor. The maximum pixel clock frequency is measured in megahertz. A main issue of the synthesis of the RTSVP is the on-FPGA memory used. That could be a problem when the reconfigurable device is shared by other system which needs a appreciable amount of memory resources. In the RTSVP, the CWR block use about the 25% and 50% of the indicated memory. The CWR block memory is easily implemented in off-FPGA memory. The high pixel clock of the RTSVP can be used in low image resolutions to increase the frame rate. Thus, with $128 \times 128$ pixel images, we can achieve up to 150 frames per second. The figure 5 show the performance of the RTSVP with three types of image pairs.

## 5 Designing Complex Stereo Vision Systems with the RTSVP

The parameterizable architecture of the RTSVP provides the basic components to make complex stereo vision systems. Within the complex stereo architectures it have the multi-baseline architectures, multi-scale architectures or color stereo vision. To avoid erroneous matches is habitual to include in the matching algorithms the called left-right check. This procedure check that the compute disparity is the same when the image reference is the left or when the image reference is right. We can include this feature in our design adding a second RTSVP that computes the reverse disparity, with no cost in the CWR block.

Also, in order to avoid ambiguous matches, the researchers have employed course-to-fine matching strategies. By means of the presented design, we can generate a multi-scale algorithm adding RTSVPs with a different correlation window size. In tracking applications, where the objects to track remain in a reduced disparity range, we can implement a optimized architecture based on several RTSVP with no large number

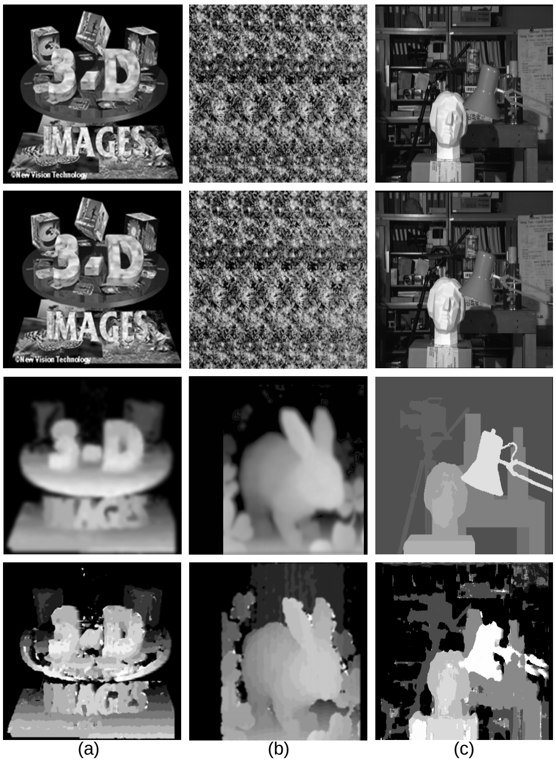

**Fig. 5**: Performance of the RTSVP using a 11 x 11 correlation window in, (a) synthetic image; (b) rabbit autostereogram; (c) Tsukuba image. From top to botton, image right, image left, truth disparity and calculated disparity

of SDC blocks. In this way, we can design a multi-baseline architecture dedicating a RTSVP to a pair of cameras. In this kind of matching algorithms with a large number of processors, it is possible to reduce the hardware resources cutting the disparity limit or other parameters.

In some applications, we also use the color to increase the efficiency of the stereo matching algorithm. In the color stereo vision, a RTSVP computes each one of the color channels. The implementation cost of a color stereo vision processor based on RTSVP (pair=256 × 256 pixels; $N$,$M$=5; $D_L$=16 and $I_B$=8) is about 10000 logic functions, 180000 bits on-FPGA memory and 180000 bits in a extern memory.

## 6 Conclusions

In this paper we have shown the feasibility of implementing vision applications on FPGA devices in order to achieve real-time performance. Also, we have shown the results of the stereo vision processor synthesis and proved that the architecture is flexible and preserve a high pixel clock and low number of logic elements and memory. Compared with the previous implementations, the RTSVP can be synthesized in only one device, reducing the board prototyping cost and the power consumption. We have shown the experimental results obtained with the proposed algorithm on some image pairs and compared these results with ground-truth maps, demonstrating the effectiveness of the proposed architecture. Our future trend will be to explore other complex stereo vision architectures and to study the implementation of the algorithms that depend of the time.

*References:*

[1] Brown M.Z., Burschka D., and Hager G.D. Advances in Computational Stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25(8), 2003, pp. 993–1008.

[2] Faugeras O., Hotz B., Matthieu H., Viéville T., Zhang Z., Fua P., Théron E., Moll L., Berry G., Vuillemin J., Bertin P., and Proy C. Real Time Correlation-Based Stereo: Algorithm, Implementations and Applications. *Tech. Rep. 2013*, INRIA, 1993.

[3] Kanade T., Yoshida A., Oda K., Kano H., and Tanaka M. A Stereo Machine for Video-Rate Dense Depth Mapping and Its New Applications. In *Proc. of the Computer Vision and Pattern Recognition Conference*, 1996.

[4] Konolige K. Small Vision Systems: Hardware and Implementation. In *Proc. 8th International Symp. Robotics Research*, 1997.

[5] Alvarez L., Deriche R., Sánchez J., and Weickert J. Dense Disparity Map Estimation Respecting Image Discontinuities: A PDE and Scale-Space Based Approach. *Tech. Rep. 3874*, INRIA, 2000.

[6] Scharstein D. and Szeliski R. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *Tech. Rep. MSR-TR-2001-81*, Microsoft Research, 2001.

[7] Kimura S., Shinbo T., Yamaguchi H., Kawamura E., and Naka K. A Convolver-Based Real-Time Stereo Machine (SAZAN). In *Proc. Computer Vision and Pattern Recognition*, vol. 1, 1999.

[8] Crisman J.D. and Webb J.A. The Warp Machine on Navlab. *IEEE Trans. Patter Recognition and Machine Intelligence*, vol. 13(5), 1991, pp. 451–465.

[9] Woodfill J. and Von Herzen B. Real-Time Stereo Vision on the PARTS reconfigurable Computer. In *Proc. IEEE Workshop FPGAs for Custom Computing Machines*, 1997.

[10] Webb J.A. Implementation and Perfomance of Fast Parallel Multi-Baseline Stereo Vision. In *Proc. ARPA Image Understanding Workshop*, 1993.

[11] Hirschmuller H., Innocent R.P., and Garibaldi J. Real-Time Correlation-Based Stereo Vision Reduced Border Errors. *International Journal of Computer Vision*, vol. 47(1-3), 2002, pp. 229–236.