Stereo Calculation of significant points using a FPGA

A. Gil, R. Gutiérrez, J.L. Alonso, S. Fernández de Ávila

Departamento de Física y Arquitectura de Computadores
Universidad Miguel Hernández
Avda. de la Universidad s/n, Ed. La Gàl·lia
03202 Elche (Alicante)
SPAIN

Abstract: - This paper describes an application, based on a FPGA (Field Programmable Gate Array), that computes stereo correspondence on a pair of images coming from a stereo rig. In this work, stereo vision is used as a way to provide a mobile robot with depth information of its surroundings. The algorithm implemented first extracts significant points in the images. Then, a stereo correspondence algorithm is computed over the points selected in the previous step. Finally, the disparity map is transmitted to a B21r mobile robot that uses this information to build a map of its environment. By doing this, we ensure that the information computed in the disparity map is highly reliable and will serve correctly in mobile robot tasks. The results show that stereo correspondence can be implemented satisfactorily on a FPGA, outperforming the results obtained by DSP-based architectures.

Key-Words: - Stereo Vision, Stereo correspondence, feature selection, mobile robot, FPGA

1 Introduction

Autonomous Mobile Robots frequently have to navigate through unknown and changing environments. Therefore, robots need to be equipped with sensors, in order to be able to extract information from its surroundings. This information is then used by the robot to accomplish basic tasks, such as moving to a goal, avoiding obstacles, exploring a new area etc.

Depth information (i.e. the distance between the robot and a near object) can be acquired with a variety of sensors, being SONAR, and proximity Laser sensors the most frequently used. However, not SONAR, nor laser sensors on their own are able to produce highly reliable depth information ([1], [2]).

Examples of Mobile Robots based on a single sensor modality have been reported (e.g. [3]). Nevertheless, those robots cannot cope with the great variety of situations that emerge nowadays. According to [4] perceptual systems based on a single kind of sensor have a inherent weakness: they generally cannot reduce uncertainty. So that, in order to achieve more demanding tasks, a robot needs to be equipped with a variety of different sensors. Fusing the data extracted from different sensors is the only mechanism to acquire a complete view of the space that surrounds the robot.

Stereo vision has been employed in many occasions as a mechanism to supply the robot with depth information of its neighbourings (see [5], [6], [7] for example) that would not be available without any other kind of sensor modality. The system that this article refers to works on a B21r robot manufactured by iRobot Corporation (www.irobot.com, shown in Fig. 1).

Fig. 1

However, typically stereo computation is highly time-consuming, leaving little CPU time to other important tasks of the robot, such as localization or map building.

In order to release the CPU from the stereo task, a FPGA has been programmed with a stereo algorithm. The main idea is to provide directly the robot with a disparity map of its environment, thus liberating the PC from the task of stereo computation.

This article discusses the use of a FPGA to compute the disparities of a pair of stereo images. In section 2 we describe the architecture design, based on a Xilinx FPGA [8]. Section 3 describes the algorithm used to select high texture points on images, based on the one exposed in [9]. In section 4 we refer to the algorithm used to make the stereo correspondence between the left and right images of the cameras. Section 5 shows the process followed to implement the algorithm on a FPGA. Finally, in section 6 we show the results obtained.

2 System architecture

Stereo calculation usually consumes greats amounts of computation time. Some authors have designed systems, based on DSPs, that achieve a fast computation of correspondence between images. For example, in [6] and [7] a system, based on two TMS320C40 DSP is used to calculate disparities in images coming from a stereo rig. On the other hand, some authors have used FPGA's to compute the stereo correspondence. The work showed in [10] and [11] presents an architecture, based on several FPGA that computes stereo disparity rapidly.

The work showed here uses a Xilinx FPGA to compute stereo correspondence in a pair of images coming from a stereo rig. In order to acquire the images the ADV7183 chip is used (manufactured by Analog Devices). This circuit is responsible of the digital conversion of the NTSC signal provided by the two XC999 cameras. The digitized images are then stored in SDRAM memory. Finally, the images are processed using the FPGA, which will be responsible of calculating the disparities in the images an transmitting them to the robot.

A FPGA (Field Programmable Gate Array) is an electronic circuit which enables the user to program its functionality. FPGA's are integrated by three basic elements: **Logical Blocks (CLBs)**, which communicate among them through **Programmable Interconnect** and **I/O Blocks** (IOBs), which communicate the FPGA with the exterior (Fig. 2).

To implement a circuit on an FPGA, each **Logical Blocks** is programmed to perform a small part of the logic required by the circuit and each **I/O Block** is programmed to act as an input or output, as required by the circuit. The **Programmable Interconnect** is also configured to make all the necessary connections between logics blocks and from logic blocks to I/O Blocks.

The FPGA used in this work belongs to the Xilinx Virtex II family. The Virtex II solution is developed specifically to enable rapid development of digital system application: data communications and digital signal processing (DSP). High logic integration, fast and complex routing of wide busses and expensive pipeline and FIFO memory requirements characterize these systems [10].

The internal configurable logic includes important elements for the image processing [10]:

- ***Block Select RAM*** memory modules provide large 18-Kbits storage elements of True Dual-Port RAM.
- ***Multiplier blocks*** *are 18-bit x 18-bit* dedicated multipliers.
- ***DCM (Digital Clock Manager)*** blocks provide self-calibrating, fully digital solutions for clock distribution delay compensation, clock multiplication and division, coarse and fine-grained clock phase shifting.

The Virtex-II prototyping module contains a Xilinx Virtex-II FPGA in the FF1152 package [12] (Fig.3).

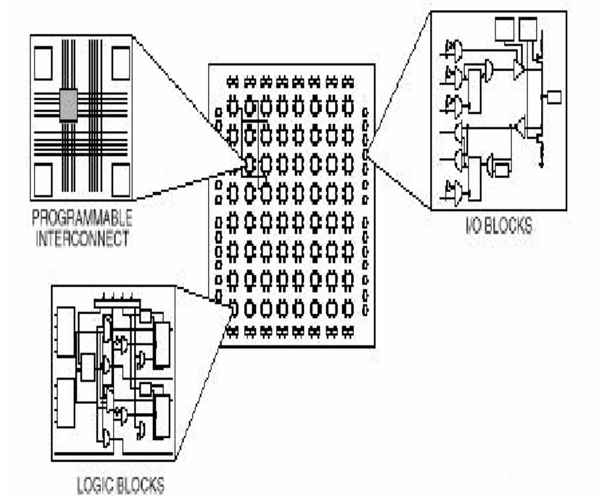


Fig. 2

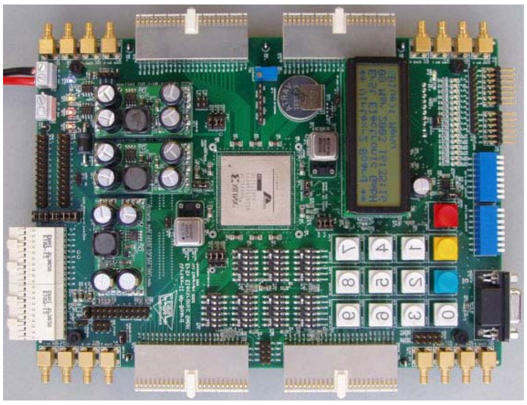


Fig. 3

3 Feature selection

Stereo vision systems determine depth from two or more images which are taken at the same time, but from different viewpoints. The most important issue in stereo vision is called the problem of stereo correspondence, which involves taking a stereo pair of images, and determining for each pixel in one image, its corresponding pixel in the other.

If stereo images are taken from two cameras oriented with parallel principal axes, perpendicular to the line that connects them, then, points that are infinitely far away will appear at the same relative position in both images, while points that are nearby will have a considerable disparity (shift in its horizontal position). If this disparity in the horizontal coordinates of two corresponding pixels can be calculated, then we will be able to calculate the position at which the point in the scene is.

Fig. 4 shows how a point *M* in space projects in the two image planes, corresponding to the left and right images. If we know the correspondent points (x_1, y_1) and (x_2, y_2), and we assume that the two optical axis are perfectly parallel, then the coordinates of *M* can be calculated as:

$$Z = \frac{fB}{d}$$

$$X = \frac{Bx_1}{d}$$

$$Y = \frac{By_1}{d}$$

The difference $d = x_1 - x_2$ is usually called disparity.

The stereo system consists of two XC999 SONY cameras forming a stereo rig (www.sony.com, shown in Fig. 5). The two principal axes of the cameras are considered to be parallel and, due to the high quality of our cameras, the lens distortion can be neglected. If we make these assumptions, each pixel in one image will have its corresponding pixel in the same horizontal scanline in the other image.

Many algorithms of stereo vision aim at producing a dense depth map taking a pair of images as input (see [6], [1], [10] and [11] for example). Those algorithms try to match every pixel in an image with its corresponding pixel in the other image. However many problems arise in practice while doing this. For example, it is possible that one pixel in one image does not appear in the other image (occlusions). Another problem are low textured areas, which typically are difficult to match.

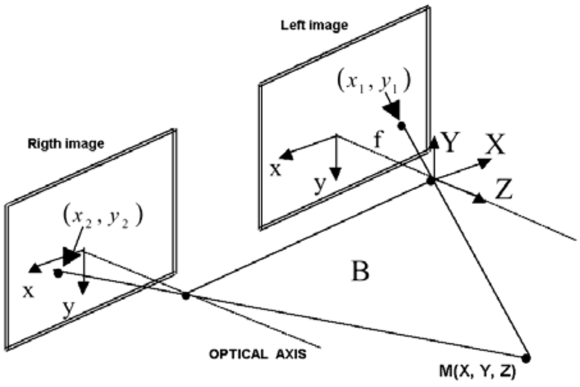


Fig. 4

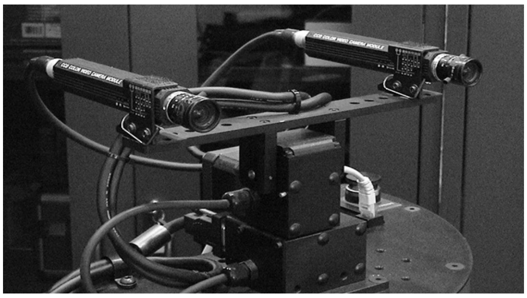


Fig. 5

In this application the stereo rig is mounted on a B21r robot that tries to navigate through an indoor environment. Walls, floor and doors do typically lack of texture, in consequence, it is not posible to find the depth of every point in the image. Thus, we have taken different approach to the problem of correspondence. We start finding good points in images just like Shi and Tomasi state in [9]. This equals to find good points in images which are placed in highly textured areas and which will be easily matched between images. The process of determining highly textured points starts by calculating the matrix *G* over a window *W* centered at pixel *p*.

$$G = \iint \begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix} dA$$

where $(g_x, g_y) = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right)$.

Then we calculate the eigenvalues of the matrix G. Pixel *p* is considered to be a good feature if:

$$\min(\lambda_1, \lambda_2) > \lambda$$

Therefore, if the minimum of both eigenvalues exceeds a predefined threshold λ, then the pixel *p* will be a good pint to match across images. By doing this, we ensure that only high quality pixels will be matched, thus producing highly reliable depth measurements.

Depth measurements made by the FPGA are used by our B21r robot to produce a map of its environment. For this application we consider that it is of great importance to find highly reliable information about certain points in the images, rather than finding a dense disparity map in which not all the points contain accurate results.

In figure 6 an image of a lamp is showed. In the image shown in figure 7, the most significant points have been marked in black. It can be observed how regions belonging to walls and window are not chosen as significant points, due to its lack of texture, while areas with noticeable changes in intensity gradient are selected by the algorithm.

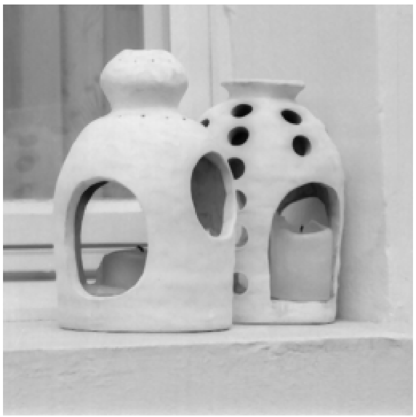


Fig. 6

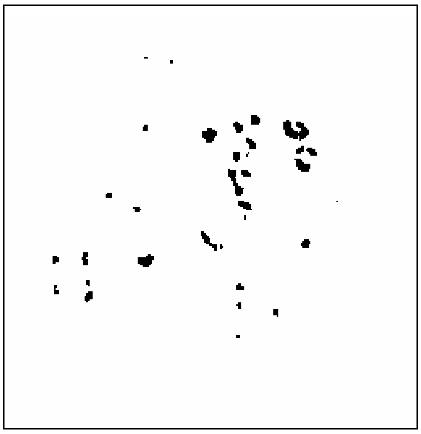


Fig. 7

4 Stereo correspondence

Once important points in both images are found, we need to find the correspondence between those points in both images. The stereo algorithm used here is based on the Census transform, described in [13] by Zabih and Woodfill. The census transform has proved to be both fast and highly reliable, providing good results in the task of stereo correspondence.

The Census transform is described as follows: Given a center pixel *p* and a window of neighbouring pixels, the transform maps each pixel and its surrounding neighbourhood into a vector of Boolean variables. Each element in the vector denotes the relation between the center pixel and a particular neighbour. Figure 8 shows how the transform works.

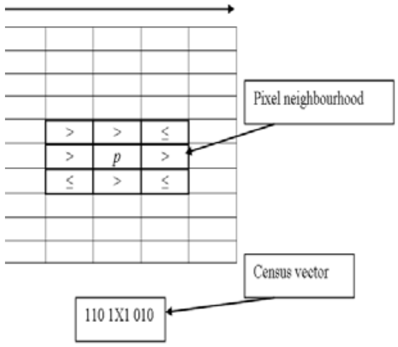


Fig. 8

If the value of the neighbours intensity is greater than the central pixel, then a value 1 is assigned, if not a 0 is assigned. By doing this, rather than parametrically comparing intensity values across images, the census algorithm performs a transform on the input images based on intra-images comparisons.

The dissimilarity between two Boolean vectors can be measured using the number of elements that differ between the two vectors (that is, the Hamming distance). Two pixel regions with nearly the same intensity structure will have nearly the same census transform, and the Hamming distance between their two representative census transformed values will be small. Thus, the problem of correspondence can be stated as finding two pixels with the minimum Hamming distance between its two Census vectors.

For each point in one of the images, calculated as described in section 3, we calculate its Census transform. The Census vector of each of those points (found in, for example, the right image) is compared with the Census vector of the pixels lying in the same scanline of the other image. Then, the correspondence across images is computed by finding the minimum Hamming distance for each pixel among the pixels that lie on the same scanline.

Once the corresponding pixels are found, the system transfers them, via USB to the robot's PC. The robot will then read the data and create a map of the space it traverses.

5 Design Methodology

5.1 Specifications

The algorithm was first programmed and tested using Matlab®. Once the algorithm proved good results in Matlab®, a block diagram of the general system was designed, including all the different functional elements (Fig 9).

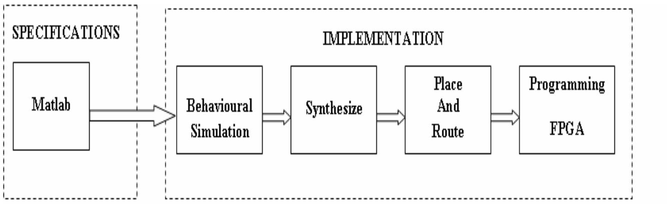


Fig. 9

5.2 Implementation

Once this algorithm was designed and tested using Matlab®, we then implemented the design in a FPGA. The stereo algorithm is described using HDLs (Hardware Description Languages). In particular, VHDL was used. In this phase we made use of the following tools:

- **FPGA Advantage [14]** : HDL Designer was used to create the project and the blocks were synthetized using LeonardoSpectrum.
- **Modelsim [15]**: It has been used for Behavioral simulation, Post-Sintesis, Post-place and Route.
- **Xilinx ISE [16]**: Was used for Place and Route and to generate the FPGA's programming file.

The system has been divided in several blocks (Fig. 10):

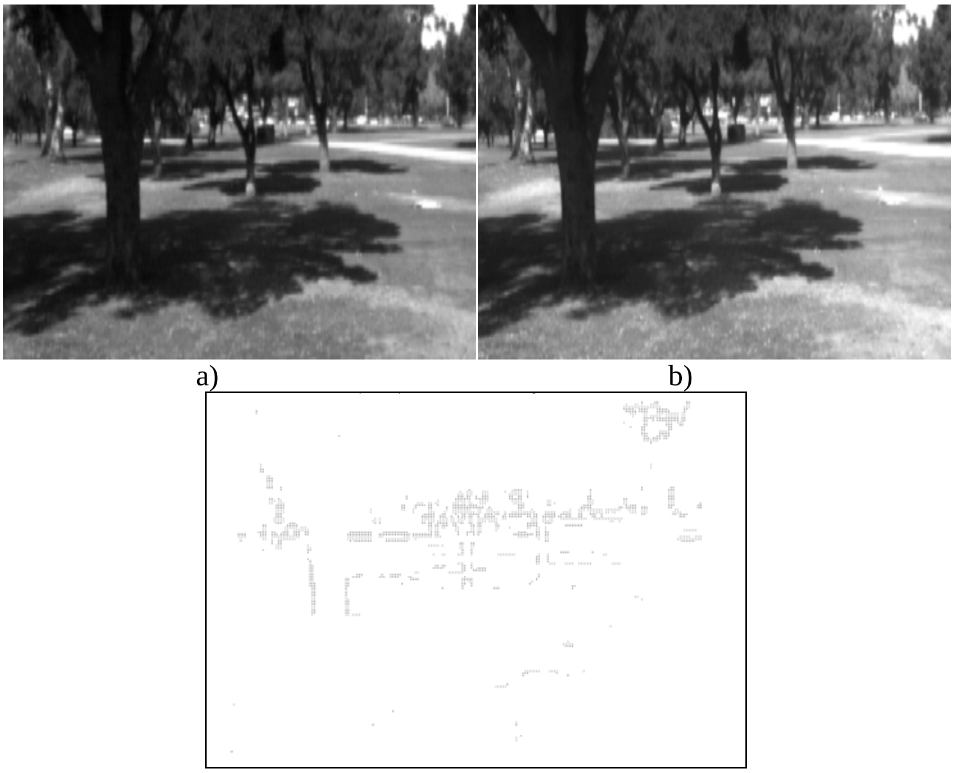
- **I2C Control:** This block is responsible of configuring the video A/D Codec [17].
- **Line Buffer:** This block is responsible of reading the luminance data from the A/D video codec and generate the image lines (see [18], [19], [20], [21]).
- **SDRAM Controller:** This block is in charge of storing the images in RAM memory [22].
- **Stereo computation:** This block accesses the RAM memory in order to calculate disparities across images. Besides, it transfers the stereo computation to the robot PC.
- **Display and PS2 controller:** It is responsible of collecting configuration commands and controlling a display, where important information about the FPGA is shown.

6 Results

The stereo images used in the tests are 720x480 pixels b/w images. Figure 11 shows the result of applying the algorithm to a pair of stereo images. Fig 11a) and b) shows a stereo pair taken respectively with the left and right cameras. On figure 11c) it is shown the disparity map referred to the the right image. We can observe how nearer points are assigned greater disparity values (lighter), while further points are assigned lower disparity values (darker points).

Thanks to the utilization of segmented architectures, the FPGA is able to perform 100 million RISC equivalent operations per second. The occupation of the FPGA is, approximately, a 85% of its full capacity.

The selection of points and the stereo correspondence is computed at a rate of 15 frames per second. This represents a clear improvement over the same algorithm implemented on a Pentium III at 500MHz, running Linux, which rated 5 frames per second.



a) b)

c)

Fig. 11

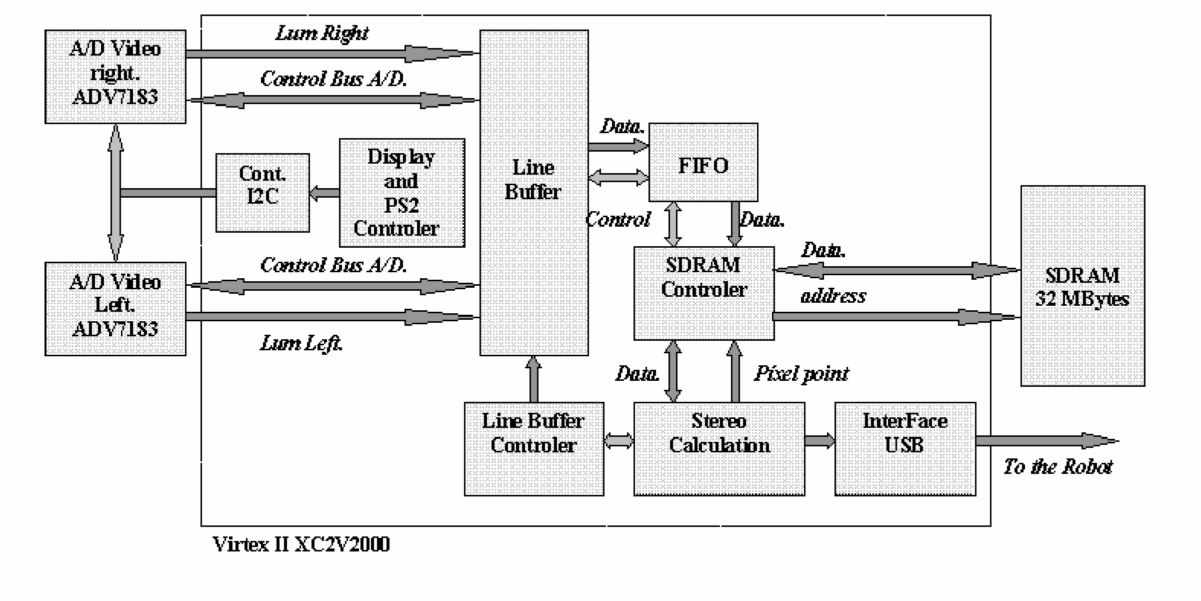


Fig. 10

7 Conclusions and future work

In this paper it has been shown how a stereo algorithm can be implemented on a FPGA. The results achieved show that the FPGA-based architecture outperforms systems based on DSP's. The solution to stereo correspondence described here extracts first significant points in images, in a manner similar to [9]. Then the points are matched across images computing a disparity map which is then used by the robot to build a map of its environment. The distances computed in this manner have showed to be very accurate and reliable.

In a future work, we plan to use the points calculated as stated in section 3 and track them in successive frames. By doing this an ego-motion of the cameras, and hence, of the robot, can be estimated (see [5] and [6], for example).

References:

[1] J. Miura, Y. Negishi, Y. Shirai, "Mobile Robot Map Generation by Integrating Omnidirectional Stereo and Laser Range Finder". *Proceedings of the International Conference on Intelligent Robots and Systems*, Lausanne, Suiza, October 2002.

[2] K. O. Arras, N. Tomatis, "Improving Robustness and Precision in Mobile Robot Localization by Using Laser Range Finding and Monocular Vision". *Proceedings of the Third European Workshop on Advanced Mobile Robots* (EUROBOT '99), Zurich, Suiza, September 1999.

[3] A. Elfes, "Using Occupancy Grids for Mobile Robot Perception and Navigation", *Computer*, pp 46-57, June 1989.

[4] R. R. Murphy, "Dempster-Shafer Theory for Sensor Fusion in Autonomous Mobile Robots". *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, April 1998.

[5] S. Se, D. Lowe, J. Little, "Global Localization using Distinctive Visual Features", *Proceedings of the 2002 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems EPFL,* Lausanne, Suiza, October 2002.

[6] D. Murray, J. Little, "Using real-time stereo vision for mobile robot navigation". *Autonomous Robots*, vol. 8, no. 2, pp. 161-171, 2000.

[7] D. Murray, C. Jennings, "Stereo Vision based mapping and navigation for mobile robots". Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'97), pp. 1694-1699, New Mexico, April 1998.

[8] Virtex™ -II Platform FPGA Handbook, December 2001.

[9] J. Shi, C. Tomasi, "Good Features to Track". *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR94) Seattle, June 1994.

[10] J. Woodfill, B.Von Herzen, "Real-Time Stereo Vision on the PARTS Reconfigurable Computer", *Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines*, Napa, CA, pp. 242-250, April 1997.

[11] A. Darabiha, J. Rose, W. J. MacLean,"Video-Rate Stereo Depth Measurement on Programmable Hardware", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR '03), volume I, Madison, Wisconsin, June 2003.

[12] Virtex-II Prototyping Board User Manual, ErSt Electronic GmbH Switzerland.

[13] R. Zabih, J. Woodfill, "Non-parametric Local Transforms for computing Visual Correspondence", *Proceedings of the 3rd European Conference on Computer Vision*, pp. 150-158, May 1994.

[14] FPGA advantage Bookcase. www.mentor.com. Mentor Graphics

[15] Modelsim SE Bookcase www.mentor.com. Mentor Graphics

[16] Xilinx ISE Software Manual www.xilinx.com. Xilinx Inc 2003

[17] Philips Semiconductor I2C Handbook. Quick Overview of general purpose I2C logic Devices.

[18] The Digital Fact Book. Edition 10 Quantel.

[19] Interfacing the ADSP-BF535 to NTSC/PAL video decoder over asynchronous port. EE-203 Analog Devices.

[20] Asynchronous FIFO in Virtex-II Devices. XAPP257. Xilinx.

[21] ADV7183 Data Sheet. Advanced Video decoder with 10-bit ADC and Componet Imput Support. Analog Devices.

[22] XAPP 134. Synthesizable High performance SDRAM Controllers. Xilinx.

[23] S. Florczyk, "New techniques for video based indoor exploration with mobile robots". *WSEAS NNA-FSFS-EC*, paper 458-162,Vouliagmeni, Athens, Greece, May 29-31, 2003.

[24] H. Lu and C. Chuang, "Navigation strategy for car-like mobile robot among irregular obstacles" *WSEAS NNA-FSFS-EC*, paper 465-148, Vouliagmeni, Athens, Greece, May 29-31, 2003.