Design Considerations for Embedded Stereo Vision Systems

AMBROSCH, Kristian; HUMENBERGER, Martin & KUBINGER, Wilfried

Abstract: This Paper describes our current work on a novel hardware architecture for an embedded stereo vision system that is suitable for robotics applications. The architecture is based on two Digital Signal Processors (DSPs) for the pre- and post-processing as well as a Field Programmable Gate Array (FPGA) for stereo matching.

Keywords: Stereo Vision, Embedded Systems, DSP, FPGA, Robotics

1. Introduction

The focus on research in autonomous systems is steadily growing in the last years. Even if there are still only very few resulting products on the market, which are usually limited to industrial applications, there already exist various prototypes of autonomous systems like self navigating wheel chairs, autonomously driven cars or household robots. Thus it can be assumed that this field of robotic applications has a very high potential. To enable an autonomous navigation it is necessary to have detailed information about the three dimensional (3D) surrounding of the system. Therefore, a priori knowledge can be very useful, but as soon as the system is not in a known and protected area, it must be able to detect and classify obstacles. Thus there is a high demand for sensor systems that can deliver this kind of information.

Today, laser range finders, which have a limited operating range at high frame rates, are mostly used for this kind of application. An alternative method for 3D information calculation is stereo vision. Stereo vision algorithms compare the images of two cameras and extract the displacement of the objects in those images using area or feature based matching. The displacement of the objects is also called disparity and measured in pixels. Using the disparity and the a priori knowledge of the distance between the cameras, the 3D depth map can be computed using triangulation.

Stereo vision algorithms are computationally extremely expensive. As a consequence of the resulting high system costs, they are currently not widely used for robotic applications. Therefore, we propose a novel hardware architecture that is based on the use of a Field Programmable Gate Array (FPGA) for stereo matching. This enables the design of an embedded stereo vision system that can be produced with minimal costs in series production, when taking into account that the FPGA design can be used for the production of an Application Specific Integrated Circuit (ASIC) which has low costs per unit in mass production.

2. Related Work

Various examples of stereo vision algorithms using FPGAs exist in the literature.
Some of these implementations use more than one FPGA like (Corke et al., 1997), (Miyajima et al., 2003) and (Niitsuma et al., 2005). Since the assembly costs rise with the use of multiple chips, an architecture using more than one FPGA would be far too expensive.
(Woodfill et al., 2006) have developed an embedded stereo vision sensor called G2 Vision System that is based on an ASIC for stereo matching, an FPGA for pre-processing as well as an Analog Devices Blackfin Digital Signal Processor and a Power-PC. The system is capable to calculate a depth map from two 512x480 images at 60fps but only with a maximum disparity of 52 pixels. Thus the systems suitability for robotic applications is very limited. Here objects with distances reaching from a few centimetres up to several meters distance have to be detected. This enforces much higher disparities.
Other works that are using only one FPGA, but also have a too small disparity are (Yi et al., 2004), (Niitsuma et al., 2004) and (Lee et al., 2005).

3. Stereo Vision

Before a stereo vision algorithm can start processing the images taken from both cameras, it is necessary that the images fulfil the epipolar geometry (Zhang, 1998). This means that for each point in the primary image the corresponding point in the other image lies on a line specific for it, called *epipolar line*. For an easier implementation of the stereo vision algorithm it is obvious to use image rows for the epipolar line.
The first step to get the epipolar geometry is to get the lens distortion out of the image. Afterwards the image needs to be rectified. Therefore it has to be rotated around the x, y and z axes of the camera (Fusiello et al., 2000).
Stereo vision algorithms can be divided into feature based and area based algorithms. Feature based algorithms search for characteristics in the primary image and try to find the corresponding characteristics in the other one. The performance of these algorithms mainly depends on the density and uniqueness of these characteristics. The search for features enforces mainly serial processing and has only a limited potential for optimization by parallelization.
Area based algorithms use blocks of pixels in the images and compare the matching costs of these blocks to solve the correspondence problem. The performance of area based algorithms is directly correlated to the density of the textures in the images. The calculation of matching costs is a calculation that can be performed independently for each block of pixels. Thus these algorithms have a very high potential for optimization by parallelization.

3. Architecture

3.1 Requirements

The requirements of an embedded stereo vision system are not only limited to the performance of the stereo vision algorithm. They also contain the reliable real time behaviour of the whole system and its robustness to the environmental influences.

Since the embedded stereo vision system can be maintained in indoor or outdoor applications, its algorithm must be able to cope with both situations. While outdoor situations are primarily texture rich, indoor environments can have poor texture e. g. when the environment is not a living room but an office building. The application in autonomous vehicles results in a very challenging temperature range. Even at most protected places within a car, the temperature ranges from -40°C to +85°C. To ensure a high reliability, the number of moving parts of the system has to be at a minimum, which means that an active cooling has to be avoided. Thus it has to be considered, that the heat emission of each single chip must be minimized. This enforces the deployment of multiple chips with small heat emission rather than a single one with large heat emission. Even if the total heat emission stays the same, the total cooling surface of the chips is increased.

3.2 Cameras

A very important decision is whether to take analogue or digital cameras. Analogue cameras have the advantage of a very robust data transmission over long distances. Also a lot of multimedia Digital Signal Processors (DSPs) already have analogue video capturing implemented on the silicon.

Analogue cameras usually provide interlaced pictures, which not only increases the latency of the system (it has to wait for the second frame to create a non interlaced picture) but also creates image blur when objects are moving between the interlaced frames. Of course also non interlaced cameras exist, but digital cameras are about to replace the analogue ones. Thus only the most common types of analogue cameras will be available in the future, which will definitely be the interlaced ones. For digital cameras three very common interfaces exist. The first one is USB (www.usb.org), which takes a lot of computational resources for the handling of the communication. Therefore it is not advisable to use it in a system with limited resources. The other ones are IEEE1394 (www.1394ta.org) and CameraLink (www.machinevisiononline.org). Both are suitable for our application and we recommend using a system that can switch between both standards (e.g. by having a daughter interface board).

3.3 Pre-Processing

The task of the pre-processing stage is to compute undistorted images that are rectified to fulfil the epipolar geometry (Zhang, 1998).

The removal of the lens distortion can be retracted using equations (1), (2) and (3) according to (Azad et al., 2007), where x_d, y_d are the distorted image coordinates and x_n, y_n are the undistorted image parameters. The lens parameters are given by d_1, d_2, d_3 and d_4.

$$x_d=x_n(1+d_1r^2+d_2r^4)+2d_3x_ny_n+d_4(r^2+2x_n^2) \tag{1}$$

$$y_d=y_n(1+d_1r^2+d_2r^4)+d_3(r^2+2y_n)+2d_4x_ny_n \tag{2}$$

$$r^2=x_n^2+y_n^2 \tag{3}$$

This calculation needs to be performed backwards, using the current positions in the undistorted images to calculate the pixel positions in the distorted image which does not accommodate the data flow within an FPGA. It also contains a high number of multiplications which requires a lot of resources as well. Thus it is more suitable for a calculation on a DSP rather than an FPGA.

The rectification of the images is calculated using the perspective projection (Azad et al., 2007) where **x** is the vector of the image coordinates in the unrectified image, **x**' the vector of the rectified image coordinates and **A** is the 3×3 transformation matrix that is pre-calculated according to the geometry of the stereo head (Fusiello et al., 2000):

$$\mathbf{x}'=\mathbf{A}\mathbf{x} \tag{4}$$

This is also a very multiplication intensive calculation and again is more appropriate to perform on a DSP rather than an FPGA. Another issue is the need for reconfiguration of the pre-processing as the camera and camera head parameters change, which might lead to a re-synthesization of the FPGA design on an optimized implementation, while a DSP software could be reconfigured any time.

Taking together all these concerns it is advisable to perform the pre-processing on a DSP.

3.4 Stereo Matching

To evaluate the best way to implement the stereo matching, we implemented a software as well as a hardware solution for a small stereo matching algorithm.

The algorithm was implemented in VHDL and synthesized for an Altera EP2S60 using Quartus II. The implemented algorithm was an Sum of Absolute Differences (SAD) (Banks et al., 1997) with a block size of 3x3 pixels using 320x240 input images in 8 bit greyscale with a maximum disparity of 100 pixels on the architecture shown in Fig. 1. The design used 19520 Logic Elements (LEs) which is about 30% of the chip surface and 884739 bits of memory. The resulting computing time per image pair is 2.3ms.

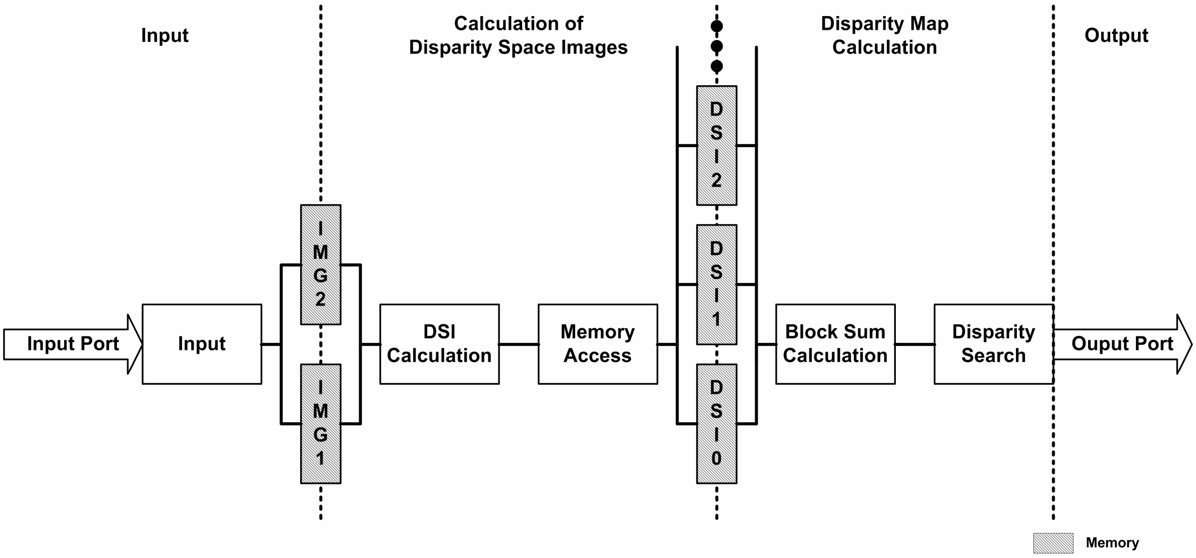

Fig. 1: Architecture of the FPGA implementation (Ambrosch et al., 2007)

The software implementation was executed on an Intel Pentium 4 with 3 GHz clock frequency and 1 GB memory. For the optimization of the software we used Intel's Open Source Computer Vision Library (Intel, 2007). In this case the computation time was 391ms, which is 166 times slower than the FPGA implementation. This example shows that an FPGA solution is much more suitable for stereo matching than a purely processor based system. Fig. 2 shows the result of the algorithm on resized images taken from the Middlebury Dataset (Scharstein et al., 2003). Since the matching of the last 100 pixels requires information that is beyond the right border of the left image, there are no values in the disparity map for this region.

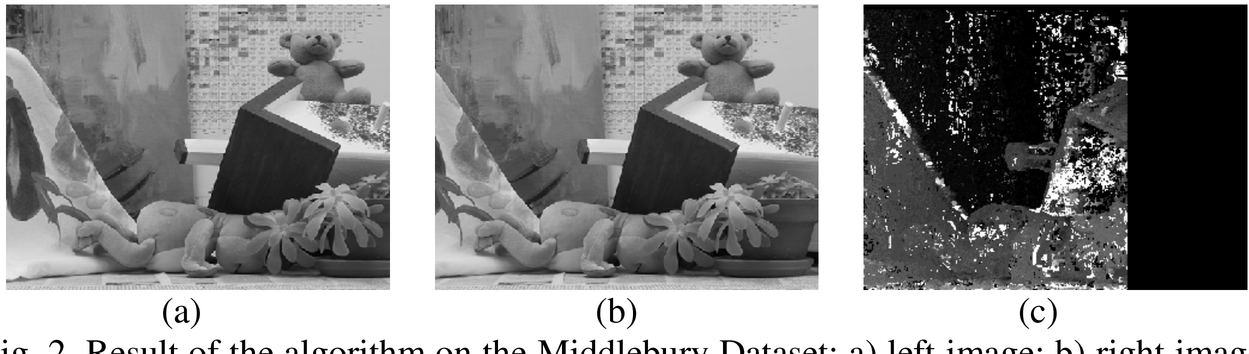

Fig. 2. Result of the algorithm on the Middlebury Dataset: a) left image; b) right image; c) disparity map.

3.5 Post-Processing

The task of the post-processing stage is to refine the output of the stereo matching. Since the post-processing is very frame dependent it cannot easily be implemented in hardware. This is the reason why we did not evaluate the use of an FPGA for this task and took a DSP.

3.6 Communication

For the communication between DSPs and the FPGA, we evaluated five different interfaces.

DSPs from Texas Instruments (TI) support serial RapidIO (www.rapidio.org). This communication interface offers speeds up to 1 GBit/s. It gets along without a separate clock line, because the interfaces synchronize via the communication line. The main problem is that an Intellectual Property (IP) core from Altera is extremely expensive, which made it less attractive for our approach. HyperTransport (www.hypertransport.org) is a parallel high speed interface which is easily implementable in hardware. Furthermore there are public licence IP cores available from OpenCores (www.opencores.org). But TI does not support HyperTransport and therefore it is not suitable for our purpose. PCI is a standard interface for connecting hardware. It offers a data transfer rate of 1 Gbit/s while operating at 33 MHz and there are public licence IP cores available at OpenCores. But it needs 47 bus lines and is only rarely supported by DSPs. Thus we decided not to take PCI. Another method for inter processor communication is true dual ported SRAM. Here chips can exchange data without the need for synchronization, but SRAM is very expensive and the power consumption is definitely too high for our purpose.

DSPs from TI have a memory interface called EMIF. This memory interface can be used to access registers on the FPGA, offering another method of transferring data between DSPs and FPGAs. Using the Direct Memory Access (DMA) controller to transfer the data, it is possible to transfer the data from the FPGA to the DSPs main memory without interrupting the DSP core. This is the main reason why we decided to take this interface for our architecture.

3.7 Results

Fig. 3 shows our hardware concept. It contains two digital video cameras connected to a fixed point DSP from TI via IEEE1394a or CameraLink. This DSP is designated to pre-process the image data. Then the data is fed into a Stratix II EP2S90 FPGA using the DSPs EMIF port. In the EP2S90 the stereo matching is performed. Afterwards the disparity map is read by the post-processing DSP using its EMIF port and transferred into its main memory. When the post-processing is finished, the final depth map is sent to the receiver via the IEEE1394a port. Furthermore, the post-processing DSP can detect deviations in the vertical alignment of the camera system and send new calibration data to the pre-processing DSP using their RapidIO connection. To store the calibration data, the pre-processing DSP is connected to an EEPROM.

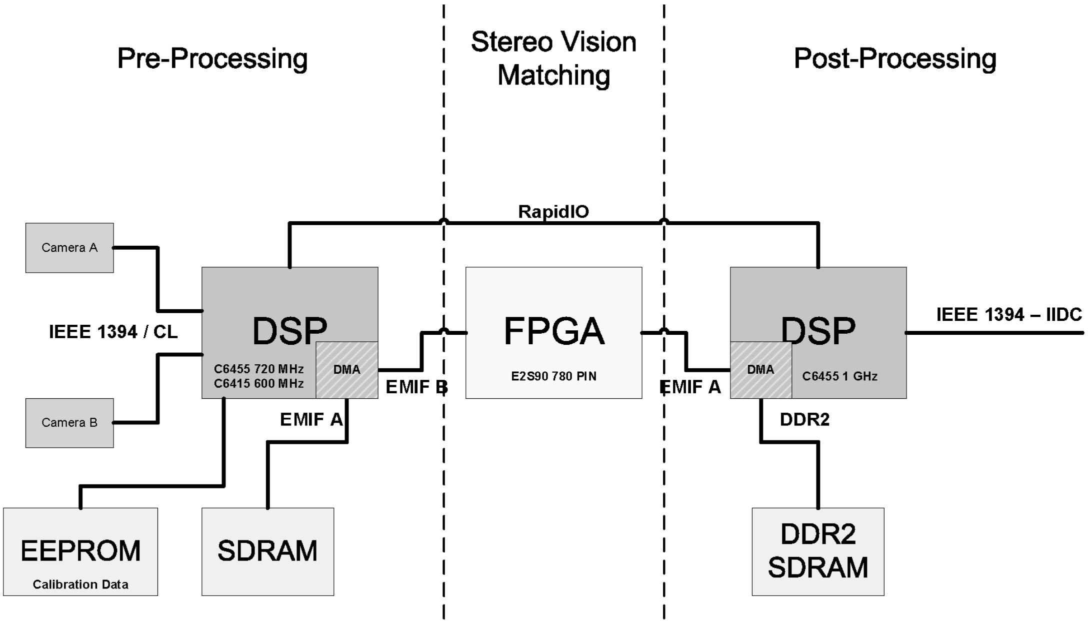

Fig. 3. Proposed Hardware Architecture

4. Conclusion

We proposed a novel hardware architecture that is based on two DSPs for pre- and post-processing as well as on an FPGA for stereo matching.
Furthermore, we compared the implementations of a stereo matching algorithm on an FPGA and a PC with the result that the FPGA is 166 times faster. This shows that FPGAs are an excellent choice to guarantee real-time behaviour on an embedded stereo vision system, when combined with DSPs for the pre- and post-processing.

5. Acknowledgements

This research has been supported by the European Union project ROBOTS@HOME under grant FP6-2006-IST-6-045350.

6. References

Ambrosch, K.; Humenberger M.; Kubinger W. & Steininger A. (2007). Hardware implementation of an SAD based stereo vision algorithm, *Proceedings of the CVPR 2007 - Workshop on Embedded Computer Vision.*

Azad, P.; Gockel, T. & Dillmann, R. (2007). *Computer Vision,* Elektor-Verlag, Aachen.

Corce, B. & Dunn, P. (1997). Real-Time Stereopsis Using FPGAs, *Proceedings of the IEEE Conference on Speech and Image Technologies for Computing and Telecommunications.*

Banks, J.; Bennamoun, M. & Corke, P. (1997). Non-parametric techniques for fast and robust stereo matching, *Proceedings of IEEE Conference on Speech and Image Technologies for Computing and Telecommunications.*

Lee, S.; Yi, J. & Kim, J. (2005). Real-Time Stereo Vision on a Reconfigurable System, *Lecture Notes in Computer Science*, Vol. 3553, pp 299-307.

Fusiello, A.; Trucco, E. & Verri, A. (2000). A compact algorithm for rectification of stereo pairs, Machine Vision and Applications, Vol. 12, No. 1, pp 16-22.

Intel (2007). Intel Open Source Computer Vision Library, *Available from:* www.intel.com/technology/computing/opencv/, Accessed: 2007-11-29.

Miyajima, Y. & Maruyama, T. (2003). A Real-Time Stereo Vision System with FPGA, *Lecture Notes in Computer Science*, Vol. 2778, pp 448-457.

Niitsuma, H. and Maruyama, T. (2004). Real-Time Detection of Moving Objects, *Lecture Notes in Computer Science*, Vol. 3203, pp 1155-1157.

Niitsuma, H. and Maruyama, T. (2005). High Speed Computation of the Optical Flow, *Lecture Notes in Computer Science*, Vol. 3617, pp 287-295.

Scharstein, D. & Szeliski R. (2003). High-Accuracy Stereo Depth Maps Using Structured Light, *Proceedings of the 2003 Conference on Computer Vision and Pattern Recognition.*

Woodfill, J.; Gordon, G.; Jurasek, D.; Brown, Te. & Buck, R. (2006). The Tyzx DeepSea G2 Vision System, A Taskable, Embedded Stereo Camera, *Proceedings of the CVPR 2006 - Workshop on Embedded Computer Vision.*

Yi, J.; Kim, J.; Li, L.; Morris, J.; Lee, G. & Leclercq, P. (2004). Real-Time Three Dimensional Vision, *Lecture Notes in Computer Science*, Vol. 3189, pp 309-320.

Zhang Z. (1998). Determining the Epipolar Geometry and its Uncertainty: A Review, *International Journal of Computer Vision*, Vol. 27, No. 2, pp 161-195.