SOC-BASED REAL-TIME PASSIVE STEREO IMAGE PROCESSING IMPLEMENTIATION AND OPTIMIZATION

R.Fütterer, M.Schellhorn, M.Hänsel

Technische Universität Ilmenau

ABSTRACT

Stereo Image Processing as a part of three dimensional image processing become more and more important for industrial measuring, quality assurance and industrial automation. While classical image processing get it features from an image plane, additional information is obtained in direction of the optical axis. In comparison to active stereo methods, which need a projector or laser source and scanning device, passive stereo need at minimum two images from different perspectives. The paper starts with the basics of passive stereo, required optical setup and electronics. Some Information about the implementation of a stereo IP core in the used Xilinx SoC FPGA embedded system given. The program flow in ARM core and FPGA is illustrated. To get a high performance image processing system, the optimization of the parameters and the implementation settings on the used FPGA is very important. A comparison of several core parameter setups is done. Finally, some ways for further optimization with new hardware technologies are given.

Index Terms – Three dimensional image processing, stereo imaging, passive stereo, embedded image processing with FPGA SoC

1. INTRODUCTION

Three-dimensional image processing can be applied for many novel use cases compared to classical two-dimension image processing. When classical image processing is getting to its borders, an additional dimension can bring a benefit. Surfaces, three dimensional structures and objects could measure in three dimensions or checked in an industrial application. Beside classical measuring or feature detection of an object in image plane, for each pixel a height information is given. In example height gradients, absolute height of object features or the amount of objects in a stack could calculated.

Three-dimensional imaging can roughly be divided into two categories: active methods, which require specific structural illumination, and passive methods. Out of the passive methods stereo imaging is the most prominent. An advantage of this method is, that in many cases ambient light is sufficient for illumination. Passive stereo is very similar to the human visual perception and requires (at minimum) two image sensors.

Providing real-time disparity images is a difficult task and ambitious goal because it necessitates a lot of computing power. For classical image processing a PC or an industrial computer is used. There are some disadvantages, which limit the area of operations. An embedded image processing system is mostly faster than a classical computer, for example due to the utilization of parallel computing architectures, special hardware blocks for fast calculation and heterogeneous hardware structures. Furthermore, a big advantage is the usually much lower energy consumption and smaller form factor. Depending the field of

application and cost volume, in embedded systems often DSPs (Digital Signal Processors) or FPGAs (Field-Programmable Gate Arrays) are used.

System on a Chip (SoC) technology as part of heterogeneous hardware offer additional benefits. The combination of FPGA and processing power of ARM technology in the same package eliminates common bottlenecks for external busses and replaces them with wide internal bus systems. ARM and FPGA share the same system memory and work parallel and independently. SoC technology gave rise to a new generation of image processing systems. For example Xilinx, Intel and Microsemi provide heterogeneous one-chip solutions. This paper describes the implementation and optimization of a passive stereo system based on the Xilinx Zynq 7000 architecture. Zynq technology offers a wide spectrum of FPGA power. The differences are mainly in FPGA hardware resource count and maximum clock speed. The subsequent described investigations are done on Zynq Z-7020 including a Artix-7 FPGA and Zynq Z-7030 including a Kintex-7 FPGA.

2. BASICS OF STEREO IMAGE PROCESSING

The first step to get three-dimensional image points is the acquisition of a stereo image pair. In this case, two images from a scene are captured and transmitted to the Zynq FPGA.

The image acquisition system consists of two parallel aligned and synchronised image sensors with c-mount lens. The sensor's LVDS video-interface is directly connected to the Zynq FPGA part. The synchronously acquired images are stored in external DDR3 random-access memory (RAM) by Xilinx Video-Direct-Memory-Access (VDMA) IP Core.

After image acquisition, image pre-processing is performed. This compensates uncontrolled lighting conditions and aperture settings. Next step is the undistortion of the two images. The optical deviation from the rectilinear projection of images is called distortion. It is necessary to undistort the images to find later corresponding image points in a line. If there is a deviation from pixel in a line, it would be confusing the issue.

Another important processing step after image acquisition is the image rectification. The rectification transforms the two images onto a common image plane. In a standard epipolar geometry object point p has different horizontal coordinates in both images (as illustrated in Figure 1). The difference between the coordinates p_L and p_R is referred to as disparity. The depth information is inversely proportional to the disparity. The distance between both cameras (lenses) O_L and O_R is called baseline. Increasing the baseline increases the precision of depth information. Thereby the overlap of both images is reduced. In the case of parallel cameras, the baseline distance, sensor and pixel size and focal length define the amount of image overlap. Only in the overlapping area valid object points of both images are present.
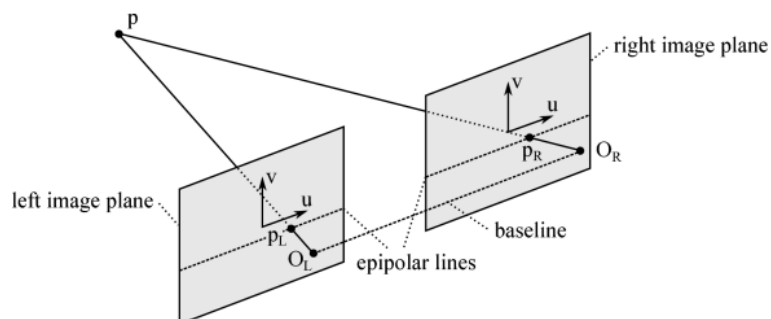

Figure 1: Standard epipolar geometry :

The disparity image or disparity map contains the displacement of the image pixels of the same object point in the left and right image. The correction of distortion and image rectification is done in one step.

After the mechanical setup, which contains the adjustment of camera sensors (parallelise, flat arrangement) and lenses (focal length, aperture, object distance), a calibrations process is necessary. In this process a set of images pairs in different distances to the sensors is acquired and saved. A calibration pattern is used to calculate the information of lens distortion and camera misalignment. By using lenses with a small distortion and a good mechanical camera attachment the amount of corrections could be minimized. After calibration of the system, a rectification map is calculated. This map contains a subpixel accurate offset for each pixel in both images. The rectification map is stored permanently in the embedded FPGA system (i.e. SD-Card, QSPI-flash) and is loaded to RAM after system start.

Searching corresponding points with pixel wise matching of mutual information requires a similar optical grey value gradient in both images. Rough-textured inhomogeneous surfaces are particularly well suited and can help to improve the accuracy of the matching point search. When a pair of matching points is found, the corresponding disparity value is derived from the offset between their pixel positions in direction of u. The calculation is performed using a variant of the semi-global matching (SGM) algorithm. The SGM algorithm works line by line, pixel by pixel through the whole image with the goal, to label each image point of the left image with a corresponding point of the right image. The original SGM (as introduced by Hirschmüller (2005)) algorithm computes along several paths in two dimensions through the image, symmetrically from all directions up to a certain maximum disparity. The cost trough each path is stored and compounded by image similarity and smoothness (penalizing small disparity steps). The disparity with the lowest cost is chosen. The variant of SGM used here differs from original SGM in the number of paths. Here only paths in horizontal and vertical downwards directions are calculated.

Further post-processing steps are executed to fill gaps, reduce speckles and noise, and filter patches with no or little texture.

3. IMPLEMENTATION OF THE STEREO CORE AND ZYNQ SOC DESIGN

The implementation of the disparity calculation on the FPGA System provides a lot of options for calculation time optimisation, possible maximum disparity and FPGA resource usage. Rectification and disparity map calculation are done in one IP-Core. A second IP-Core provides the data transfer between Stereo-Core and RAM via direct memory access.

The whole FPGA design is made in Xilinx Vivado Design Suite with the Block Design Editor.

The whole control of image acquisition, disparity map calculation and image output is done by the ARM unit of the Zynq SoC. All timing critical functions are handled by interrupts. The workflow is shown in Figure 2.

There are several steps between image acquisition and disparity image output:
- Trigger both image sensors via SPI software trigger
- Wait for Sensor Strobe signal, image acquisition and transfer is done
- Configure the stereo core, begin disparity image calculation
- Wait for return signal
- Transform image from 8-bit grayscale to 24-bit RGB, optional calculate pseudo-colour image

- Image output is done in background the whole time with 60 frame per second via HDMI
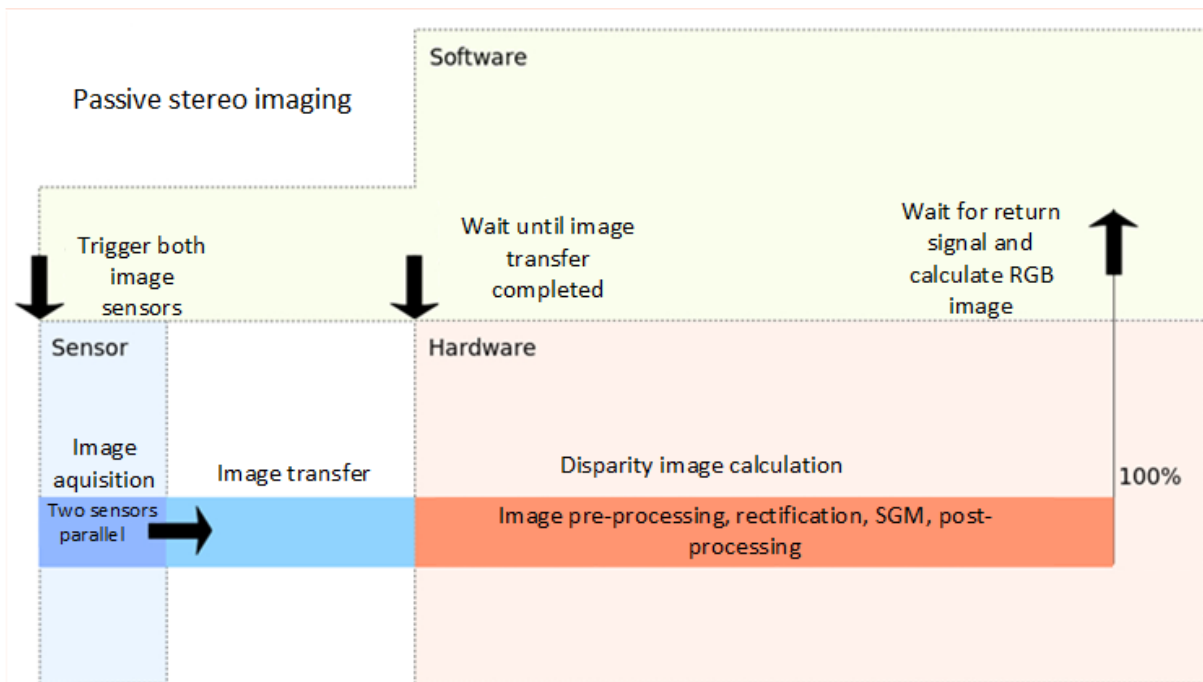
Figure 2:disparity image calculation - workflow

The performance of the FPGA is significantly depending on the quantity of logic cells and block RAM as well as the FPGA speed grade. A higher speed grad allows a higher clock rate. More block RAM and logic cells permit more parallel operations.

The SGM algorithm allows the parallelization of calculations. Therefore, the calculation time depends greatly on the number of matching point search tasks that are executed in parallel in the FPGA. This number has to be a power of two and a higher grade of parallelisation results in decreased time required for the calculations. When keeping the maximum disparity constant, the doubling of parallelization grade allows the bisection of the required calculation iterations in the core and an increase of calculation speed.

A very high parallelization in combination with a very high computing clock will generate FPGA timing problems, so a good balance is necessary to get a minimum of calculation time. The image processing also requires external RAM for the caching of intermediate data. The external RAM and the memory controller confine the maximum throughput of intermediate data. There is a very big throughput of incurred intermediate data of the core during the disparity calculation. So the possible system throughput extensively affects the core calculation time.

Among the hardware limitation, the calculation time depends on the image resolution and the maximum possible disparity.

Currently the acquired images have VGA (640 x 480) image resolution. Image acquisition, image transfer from the image sensor to RAM and the transfer of the disparity calculation result are a systematic offset and not included in the disparity calculation time. So the total frame time is the sum of aforementioned offset and the disparity calculation time.

To minimize the core calculation time, the following parameters were optimized in their respective boundaries:

- Grade of parallelization: 8 or 16 or 32 parallel tasks
- Maximum disparity: from 79 to 213
- Calculation clock: from 133MHz to 177 MHz
- Base clock from 100 MHz to 160 MHz

Calculation Time on Zynq 7020 SoC for various parameter settings:

Table 1: Zynq 7020 Implementation

Grade of parallelization	Iterations	Base clock [MHz]	Calculation clock [MHz]	Time [ms]
8	12	100	145	31,1
8	14	100	145	36,82 *
16	7	100	133	34,25 *
16	8	100	133	39,6
16	10	100	133	50,4
16	12	100	133	61,7
16	14	100	133	72,6
16	5	100	133	25,0

Image acquisition, disparity calculation and image output require a lot of memory bandwidth. There are four memory ports (AXI High Performance) on Zynq 7020 and 7030 SoCs. The distribution of these ports can be defined by the user and depends amongst others from the used clock. Changing the connection of the calculation buffer port produces a better timing:

Table 2: Zynq 7020 Implementation with changed Memory Ports

Grade of parallelization	Iterations	Base clock [MHz]	Calculation clock [MHz]	Time [ms]
16	7	100	133	32,24 *
16	7	100	145	31,9

By using the Zynq accelerator coherency port (ACP) as memory interface for intermediate data, the overall calculation time on Zynq 7020 SoC for a disparity image was decreased from 34.4 ms to 21.72 ms.

Table 3: Zynq 7020 Implementation with usage of ACP Port

Grade of parallelization	Iterations	Base clock [MHz]	Calculation clock [MHz]	Time [ms]
16	7	100	133	22,55 *
16	7	106	133	21,72

Changing the hardware environment provides further calculation time advantages. For example, a higher FPGA speed grade allows a higher calculation clock. A FPGA with more hardware resources, e.g. a Zynq 7030 SoC, allows higher parallelisation - from 16 up to 32 simultaneous tasks.

Table 4: Zynq 7030 Implementation

Grade of parallelization	Iterations	Base clock	Calculation clock	Time [ms]
16	7	100	133	20,84 *
16	7	100	160	17,98
16	7	133	133	17,23
16	7	152	160	17,02
16	7	152	177	15,48
16	8	100	133	23,95
16	8	133	133	23,42
16	8	133	160	19,81
16	8	152	160	19,0
16	8	152	177	17,81
32	4	100	133	13,77

32	4	100	145	13,71
32	4	100	160	13,68
32	4	152	166	10,68
32	4	152	177	10,21
32	4	152	190	9,78

The lines indexed by "*" after Time are measured values, which are comparable in the different tables because of a similar amount of executed calculations (comparable parameters) and a maximum possible disparity of 111 pixels. In Figure 3 the dependency of calculation time and iterations is shown.
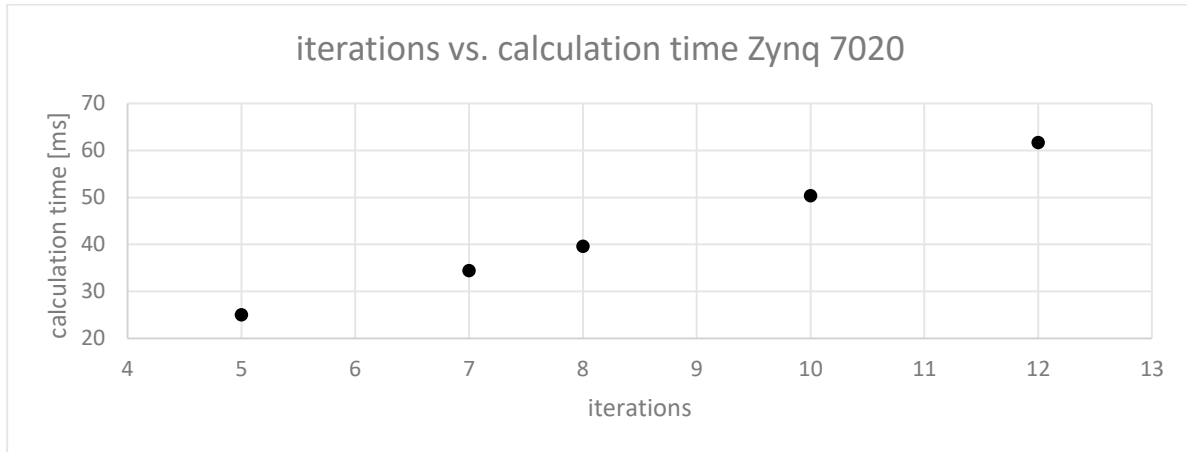

Figure 3: iterations vs. calculation time

Visible is a nearly linear relationship when more iterations are done and all other parameters are constant. When the maximum required disparity is known, the number of iterations can be set on the next even number. The maximum disparity d_{max} results from the product of the iteration count i and the grade of parallelisation p minus 1.

$$d_{max} = i \times p - 1$$

Another way to speed up the calculation time is to limit the minimum disparity and with that the maximum distance between the cameras and the objects. A disparity of zero represents an infinite distant object. In image processing applications, mostly the range where objects are present isn't extensive like a distance from nearly zero to infinity. With an offset o with a disparity range from zero to o a field of depth could defined where no matching points are calculated.

The new disparity d_{max} is calculated from the equation:

$$d_{max} = o + i \times p - 1$$

Investigations with a focal length of 6mm and a maximum distance of 40cm provide an offset value of 40 pixels. In this example, the iteration count could be decreased from 7 to 5 with a parallelization grade of 16 in Zynq 7020 technology. The processing time would sink to about 70 percent of the value without offset consideration.

Another way to speed up the system and reduce memory throughput requirements is the reduction of the output image size for the external display from currently 1920 x 1080 pixels to 1280 x 720 or 800 x 600 pixels.

If live output is not necessary it could even be switched off, so an additional bandwidth of 2.985 Mbit is available.

4. OUTLOOK

In future, a FPGA with more hardware resources is favoured to achieve a parallelisation of 64 concurrent processes. Furthermore, an external memory controller could be added to the PCB

setup to store parts of the intermediate data required for the calculation. This would lessen the limitations imposed by the memory bus and therefore accelerate the calculations. In late 2016 Xilinx introduced the Zynq Ultrascale+ technology. This technology offers faster ARM and FPGA units and allows higher memory throughput with DDR4 technology. The fastest way to calculate the disparity image is an in-line-calculation with the image input data. That means, the calculation must be clocked with n-times pixel clock. The clock factor n is given by the maximum disparity and grade of parallelisation and is equal to the iteration count i of the IP core. This provides extra memory bandwidth which could be used for caching intermediate data.

5. ILLUSTRATIONS, GRAPHS

ACKNOWLEGDEMENTS

REFERENCES

[1]   K. Schauwecker, Rechner für Objektiv Brennweite und Stereo-Basisbreite, Website 25.7.2017 https://nerian.de/support/resources/calculator/

[2]   Hirschmüller, H. (2005): "Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information", IEEE Conference on Computer Vision and Pattern Recognition, June 2005, San Diego, CA, USA

CONTACTS

R. Fütterer   richard.fuetterer@tu-ilmenau.de
M. Schellhorn   mathias.schellhorn@tu-ilmenau.de
M. Hänsel   michael.haensel@tu-ilmenau.de