UTILIZATION AND PERFORMANCE CONSIDERATIONS IN RESOURCE OPTIMIZED STEREO MATCHING FOR REAL-TIME RECONFIGURABLE HARDWARE

Fredrik Ekstrand, Carl Ahlberg, Mikael Ekström, Lars Asplund and Giacomo Spampinato
School of Innovation, Design and Technology, Mälardalen University, Sweden
fredrik.ekstrand,carl.ahlberg,mikael.ekstrom,lars.asplund,giacomo.spampinato@mdh.se

Keywords: FPGA, stereo-vision, resource utilization, real-time

Abstract: This paper presents a quantitative evaluation of a set of approaches for increasing the accuracy of an area-based stereo matching method. It is targeting real-time FPGA systems focused on low resource usage and maximized improvement per cost unit to enable concurrent processing. The approaches are applied to a resource optimized correspondence implementation and the individual and cumulative costs and improvements are assessed. A combination of the implemented approaches perform close to other area-matching implementations, but at substantially lower resource usage. Additionally, the limitation in image size associated with standard methods is removed. As fully piped complete on-chip solutions, all improvements are highly suitable for real-time stereo-vision systems.

1 INTRODUCTION

The extraction of depth data through the localization of the same point in two images is not trivial. Stereo matching of an entire scene 30 times per second (real-time) is computationally demanding, and require high-performing hardware. Hardware implementations range from regular computers, to specialized hardware such as GPUs and FPGAs. Lazaros et al. (Lazaros et al., 2008) make a thorough presentation of various implementations.

FPGAs, often referred to as reconfigurable parallel hardware, are utilized in mobile applications using vision, as they outperform other approaches in terms of speed, size, and power requirements. The major obstacle is the limited resources, which restricts which algorithms are possible to implement. In general, approaches for stereo matching are divided into global and local methods, with the latter being the preferred real-time stereo matching approach for a long time due to ease of implementation and speed (Lazaros et al., 2008).

A complete vision system residing in an FPGA requires several processing components just for preprocessing the image, such as, image rectification, motion compensation, and depth estimation. Additionally, higher-level tasks, such as tracking, object recognition, or navigation, should also be encompassed in the FPGA.

In this paper we examine the impact of heavily reducing the resource usage of a stereo matching approach. The goal is to achieve high throughput at minimal system cost. This work is part of the Two Camera-project at Mälardalen University. The aim is to construct a compact, vision-based autonomous system encompassing both sequential and parallel processing units (Ahlberg et al., 2011). Previous work in the project include the construction of the FPGA-based stereo platform (Lidholm et al., 2008), and an implemented resource optimized basic stereo matching algorithm (Ekstrand et al., 2011). The code composing the components in this project will be made available as open source to promote FPGA-based image processing on our publicly available vision system.

Matching is evaluated using stereo images with ground truth, as shown in figure 1, and the online tool provided by the vision department at Middlebury University (Evaluation, 2011).

2 BACKGROUND

For all approaches, we assume rectified and parallel images with a unified baseline, in order to reduce the correspondence problem to a 1-dimensional

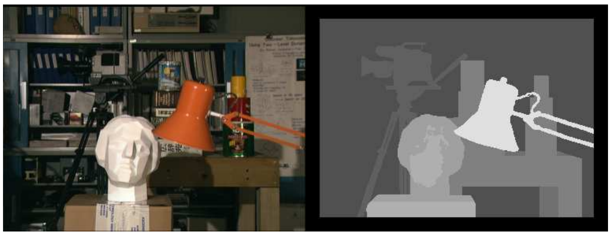

Figure 1: The Middlebury test image Tsukuba with the associated ground truth.

search (Scharstein and Szeliski, 2002).

In (Ekstrand et al., 2011), we use SAD (*Sum of Absolute Differences*) for a resource optimized correspondence implementation for real-time systems. The support window is reduced to a single row, thus producing a disparity map with preserved salient details but with increased noise, as can be expected when compared to the standard 2D implementation. The noise is primarily located in low-texture areas with low signal-to-noise ratio. In fact, the approach outperforms the 2D version around discontinuities due to reduced foreground fattening. The major advantages of the 1D approach are the substantial reduction in resource usage and the removal of the need for complete scan-line retention. The question is by how much can the matching quality can be improved, and at what cost?

3 RELATED WORK

An FPGA consists of different elements that can be configured in a multitude of ways. Resource utilization is normally expressed in slices and LUTs (LookUp-Tables which realize boolean operations). The 1D implementation from (Ekstrand et al., 2011) produced the disparity maps in this paper from 1.2K slices when implemented in a Spartan-3 FPGA. This is just above 4% of the available slices in the chip.

Several other stereo matching approaches with low resource usage exists, such as the one proposed by Arias-Estrada et al. (Arias-Estrada and Xicotencatl, 2001). The utilization is only 4.2K slices on a Virtex-II, but the disparity map is only fair. The implementation is capable of 71 fps with images of 320x240 pixels. Lee et al. (SUNGHWAN et al., 2005) present an implementation below 10K slices in resource usage. The resulting disparity map is moderate with extensive blurring of edges and noise.

4 IMPROVEMENTS

Noise in stereo matching is evident as false matches. False matches occur from the fundamentals of matching two images from different viewpoints because of projective distortion, Kanade and Okutomi (Kanade and Okutomi, 1994). Fusing two views together will leave areas where depth estimation is impossible as they are occluded in one of the views. This affects all area-based approaches, but is even more evident for smaller support windows as they have lower signal-to-noise ratio. Post-processing of the estimated disparity map is usually adopted to remove false matches, and established methods include left-right consistency check (LRC) (Fua, 2004), propagation, and median filtering.

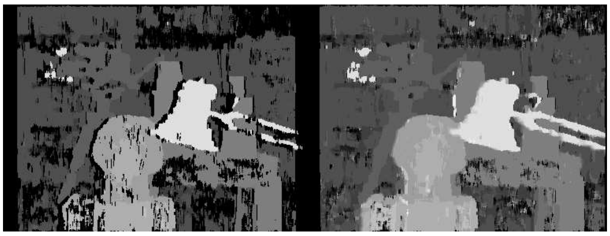

Figure 2: Impact of applying LRC (left), or not (right). Both images have been median filtered as the last stage.

4.1 Consistency Check

The left-right consistency check verifies that only disparity values with mutual correspondence are accepted as matches, as detailed by Fusiello et al. (Fusiello et al., 1997). Our implementation practically doubles the resource usage for the matching process (1.2K vs 2.4K slices), but no external memory nor any reduction in system performance need to result from it.

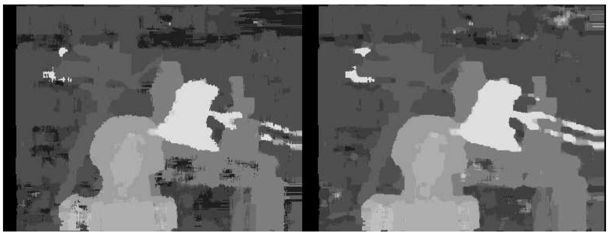

Figure 3: The median filter preceding the propagation (left) does not perform as well as the reverse (right). LRC-check was performed initially on both images.

The effect of the consistency check can be observed in figure 2. The images are with and without consistency check, but both have median filtering performed subsequently, to minimize the empty regions.

Table 1: Performance and cost for median filtering. The % is the errors in the image compared to the ground truth for Non-occluded, All, and regions near Discontinuities.

Approach	Non-Occ %	All %	Disc %
SAD 7x1	29.1	30.7	27.4
SAD-MED 7x1	22.2	23.9	24.3
SAD 7x7	22.7	24.4	28.6
SAD-MED 7x7	21.9	23.6	28.1

The consistency check identifies almost all of the occluded areas. However, it also removes pixels that are not occluded but still differ due to poor correlation data. Noteworthy is the deterioration of the lamp arm, partly due to the check but also due to the filter. The removal of data in the disparity map reduces the quality, and it is evident that the median filter (here a 7x1) is not filling the empty areas. For this to happen we need to propagate.

4.2 Propagation

With propagation, the underlying data is important. A logic assumption is that it is important to remove as much noise as possible before performing propagation, to avoid propagating false matches. As can be seen in figure 3, there is a difference between performing median filtration before or after the propagation. Propagation directly after the consistency check followed by a median filter produces a disparity map of the highest accuracy. However, some areas deteriorate, such as the lamp arm, when compared to a non-consistency checked image.

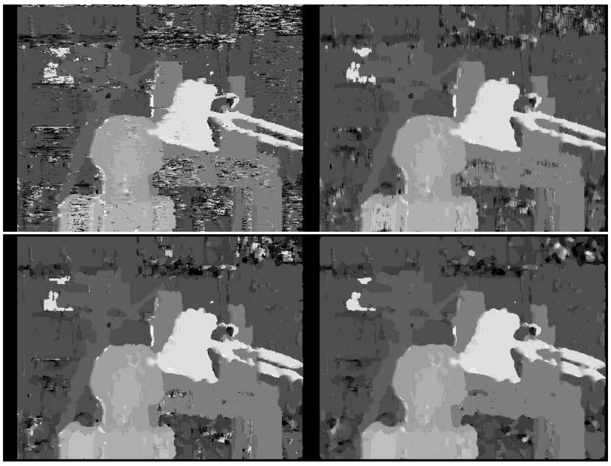

Figure 4: The median filter (right column) significantly reduces the noise for the 1D (top) but not the 2D (bottom) approach.

4.3 Filtering

Median filtering is a well-known approach to remove sporadic noise and is frequently used in post-processing to improve disparity maps (Muhlmann et al., 2002). Realization of a median filter is a search and rank problem with large filters being difficult to implement for real-time (Vega-Rodrguez et al., 2007). We have implemented the median filter as a classic systolic array, according to (Vega-Rodrguez et al., 2007), for sorting 9 elements. This translates to a 9x1 median filter for 1D and a 3x3 filter for 2D.

The improvement with a median filter are quite significant for the 1D approach, but not so much for the 2D, as can be seen in figure 4 and in table 1. The filter removes noise and the 2D implementation is already noise reduced by design. It is obvious that the noise in the 1D approach fits the characteristics of a median filter. Noteworthy is the fact that the 1D approach outperforms the standard 2D in regions of discontinuities, due to the lack of vertical summing. This is the case already with the basic 1D, but is even more improved with the added filter. The cost of the filter is very low, only 247 slices, an increase of 20%. As a conclusion, median filtering closes the gap between 1D and 2D implementations.

5 RESULT SUMMARY

Table 2 shows the improvement for the stereo matching component with the implemented approaches, both individually and combined. The improvements are evaluated with the Middlebury stereo evaluation tool (Evaluation, 2011) which show the error percentage in the disparity image. Three different parameters are presented: Non-occluded pixels which are visible in both images; Pixel at or around discontinuities in the image; All image pixels. From observing the matching scores in table 2, it can be noted that the individual order of tools is important when combining for improvement.

6 CONCLUSIONS

Utilizing an inexpensive median filter effectively closes the gap between the 1D and the 2D approaches. From a cost/performance perspective, only using a median filter with the 1D is the best approach. However, there is only so much a 1D median filter can do with noisy data. For further improvement, noise reduction is a must. A function removing, or never allowing, false matches in the initial disparity map,

Table 2: Impact of improvements; individually and combined. All values are for a 7x1 implementation.

Approach	**Non-Occ %**	**All %**	**Disc %**	**Slices**	**LUTs**
SAD	29.1	30.7	27.4	1,221	6,086
LRC	40.5	41.9	40.1	2,399	7,689
Median	22.2	23.9	24.3	1,468	6,371
LRC-Med	38.6	39.9	39.5	3,135	8,204
LRC-Prop	27.2	28.4	24.9	3,174	8,237
LRC-Med-Prop	31.1	32.0	28.8	3,986	8,844
LRC-Prop-Med	20.4	21.8	21.7	3,986	8,844
Available				**33,280**	**33,280**

through confidence assessment, could render a substantial improvement together with a competent propagation method. Implementing a small confidence measurement would be a good continuance of this work.

It is further evident that it is possible to achieve acceptable disparity maps without extensive memory usage and without a limitation on image size. Megapixel images will not affect the throughput or the resource utilization of the suggested approach as image data is only stored in a shift register approach without the need for multi-scanline retention. Furthermore, the 1D implementation is resource reduced, and can be fitted to practically any FPGA. It has been implemented with a maximum disparity range of 64 for images of 1024x1024 pixels.

The implementations run at 125 MHz, the system clock of the FPGA-board (Lidholm et al., 2008). As the implementations are fully piped, the frame rate is dependent on the speed of the cameras and the size of the frame. Theoretically, it is capable of processing over 100 frames per second for Megapixel images.

ACKNOWLEDGEMENTS

The authors would like to acknowledge *Xilinx* for their kind donation of our FPGA's and design software tools, *Hectronic* for the design and manufacturing of our FPGA boards, and *The Knowledge Foundation* for providing funding for the project.

REFERENCES

Ahlberg, C., Spampinato, G., Lidholm, J., Ekstrand, F., Ekström, M., and Asplund, L. (2011). Gimme - a general image multiview manipulation engine. Technical paper, School of Innovation Design and Engineering, Mälardalen University.

Arias-Estrada, M. and Xicotencatl, J. (2001). Multiple stereo matching using an extended architecture. In Brebner, G. and Woods, R., editors, *Field-Programmable Logic and Applications*, volume 2147 of *Lecture Notes in Computer Science*, pages 203–212. Springer Berlin / Heidelberg.

Ekstrand, F., Ahlberg, C., Ekström, M., Asplund, L., and Spampinato, G. (2011). Resource limited hardware-based stereo matching for high-speed vision system. In *5th International Conference on Automation Robotics and Applications, 2011. (ICARA 2011).*

Evaluation, M. U. O. S. (2011). http://vision.middlebury.edu/stereo.

Fua, P. (2004). A parallel stereo algorithm that produces dense depth maps and preserves image features * ii id ii ii / i i 111 ? 1 1 1 1. *Machine Vision and Applications*, 6(1993):35–49.

Fusiello, A., Roberto, V., and Trucco, E. (1997). Experiments with a new area-based stereo algorithm.

Kanade, T. and Okutomi, M. (1994). A stereo matching algorithm with an adaptive window: theory and experiment. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(9):920 –932.

Lazaros, N., Sirakoulis, G. C., and Gasteratos, A. (2008). Review of stereo vision algorithms: From software to hardware. *International Journal of Optomechatronics*, 2(4):435–462.

Lidholm, J., Ekstrand, F., and Asplund, L. (2008). Two camera system for robot applications; navigation. In *13th IEEE International Conference on Emerging Technologies and Factory Automation, 2008. (ETFA 2008)*, pages 345–352. IEEE.

Muhlmann, K., Maier, D., Hesser, J., and Manner, R. (2002). Calculating dense disparity maps from color stereo images, an efficient implementation. *International Journal of Computer Vision*, 47(1):30–36.

Scharstein, D. and Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47:7–42.

SUNGHWAN, L., JONGSU, Y., and JUNSEONG, K. (2005). Real-time stereo vision on a reconfigurable system. *Lecture Notes in Computer Science*, 3553:299–307.

Vega-Rodrguez, M. A., Snchez-Prez, J. M., and Gmez-Pulido, J. A. (2007). *An FPGA-based implementation for median filter meeting the real-time requirements of automated visual inspection systems*. Citeseer.