Real Time Stereo Vision with a modified Census transform and fast tracking in FPGA

Juan Manuel Xicoténcatl-Pérez, Arturo Lezama-León, José Miguel Liceaga-Ortiz-De-La-Peña,Rubén O. Hernández-Terrazas

Polytechnique University of Pachuca
Km. 20 Carretera Pachuca Sahagún Ex Hacienda de Sta. Bárbara , Zempoala, Hidalgo, México
jmxico@upp.edu.mx

Abstract. In this work, architecture for real stereo processing is implemented using a modified census transform based in a non central pixel census technique. The non central census technique allows a compact architecture. Proposed architecture is segmented in image rectification module to avoid lens distortion and to align epipolar lines, stereo processing module with modified census transform and finally, a post processing module with a propagation algorithm to correct false disparity values. Proposed architecture uses low hardware resources and memory requirement in a way that Spartan Low Cost FPGA can be used for implementation. Additionally, a correlation tracking module is incorporated to one camera with neglected hardware cost in comparison with other architectures for security applications.

Keywords: real time, stereo vision, fast tracking

1 Introduction

Stereo vision, compared with other range sensors such as laser scanner or time-of-flight, is a technology that can deliver the sufficient description of the surrounding environment. It is purely passive technology and thus offers low cost and has potential uses in many visual domains such as autonomous navigation in which accurate 3D information about the road is crucial, object 3D reconstruction, object segmentation, and surveillance systems [10]. However, calculation of three dimensional depth maps on signal processors that meets these requirements is very time consuming. In this way, real-time dense stereo is difficult to be achieved with general purpose processors even CUDA. For real-time requirements of most applications, the specific algorithms were often implemented using dedicated hardware and it is possible because of many stereo vision algorithms do not enforce a purely sequential implementation and are therefore apply to parallelized solutions. Additionally, stereo applications are used for security applications or video games in parallel with a tracking algorithm or system. For this reason, a tracking architecture is designed using the correlation structure from the stereo module.

Advances in Computer Science and Engineering
Research in Computing Science 58, 2012 pp. 297–310
Paper Recived 20-08-2012 and Acepted 07-11-2012

In the last few years, the GPUs have become more and more popular. Using GPUs for stereo acceleration can directly be a solution for PC-oriented applications. But, the high power consumption limits their applications. FPGAs have already shown their high performance capacity for image processing tasks in parallel especially for embedded systems. In this paper, it is simulated and synthesized a stereo vision core algorithm implemented in VHDL for the Spartan XC3S1000 from XILINX, an FPGA that is suitable for this kind of application. The algorithm is based on a Census transform modified algorithm with produces a small hardware implementation. It is small enough to enable the pre- and post-processing of the images on the same FPGA, but with a maximum disparity of 50 pixels all in real time for a compact system.

2 Background

The task of a stereo vision algorithm is to analyze the images taken by a pair of cameras and to extract the displacement of the objects in both images. This displacement is counted in pixels and called disparity. All these disparities form the disparity map, which is the output of a stereo vision algorithm and enables the calculation of distances to objects using triangulation.

Detecting conjugate objects in stereo images to obtain dense disparity maps is a challenging research problem known as the correspondence problem, i.e. to find for each point in the left image, the corresponding point in the right one. To determine a conjugate pair, it is necessary to measure the similarity of the points. The point to be matched should be distinctly different from its surrounding pixels. In order to minimize the number of false correspondences and try to diminish time processing in the image pair, several constraints have been imposed. The uniqueness constraint requires that a given pixel from one image cannot correspond to more than one pixel on the other image. In the presence of occluded regions within the scene, it may be impossible at all to find a corresponding point. The ordering constraint [6] requires that if a pixel is located to the left of another pixel in image, i.e. left image, the corresponding pixels in right image must be ordered in the same manner, and vice versa, i.e. ordering of pixels is preserved across the images. The ordering constraint may be violated if an object in the scene is located much closer to the camera than the background, and one pixel corresponds to a point on the object while the other pixel corresponds to a point in the background. Finally, the continuity constraint [4], which is valid only for scenarios in which smooth surfaces are reconstructed, requires that the disparity map should vary smoothly almost everywhere in the image. This constraint may be violated at depth discontinuities in the scene.

According to a recent taxonomy [17], stereo algorithms that generate dense depth measurements can be divided into two classes, global and local algorithms. Global algorithms, e.g. [14], rely on iterative schemes that carry out disparity assignments on the basis of the minimization of a global cost function. These algorithms yield accurate and dense disparity measurements but exhibit a very high computational cost that renders them unsuited to real-time applications. Local algorithms [6, 8, 13] can be based in different concepts to establish a correspondence between images, so it is

possible to find area-based and feature based. Area-based stereo algorithms are approaches that propose a dense solution for calculating high-density disparity maps. Additionally, these approaches have a regular algorithmic structure which is suitable for convenient hardware architecture. A simple method would be to calculate the absolute difference between two pixels; this method is extremely cheap but not robust. The second possible method seeks to improve upon the previous one by considering a small window around the pixels in the left and right image and then using the sum of the differences. Small windows give support to central pixel to avoid false matching during the stereo process. In this paper, it is used a modified Census transform to achieve a stereo correlation. Modified census transform uses a non central pixel reference, i.e., it uses the most right pixel in the central row from a given window. This change just affects the generated census vector but results and advantages are same like other census implementations. Census transform is a non-parametric measure used during the matching process for measuring similarities and obtaining the correspondence between the points into the left and right images. Advantages with census transform are: a windows support, tolerance to illumination changes and possibility to hardware parallel implementation. On the other hand, feature-based algorithms rely on certain points of interest. These points are selected according to appropriate feature detectors. The major limitation of all feature-based techniques is that they cannot generate dense disparity maps, and hence they often need to be used in conjunction with other techniques. Because of the sparse and irregularly distributed nature of the features, the matching results should be augmented by an interpolation step if a dense disparity map of the scene is desired. Additionally, an extra stage for extensive feature detection in the two images is needed, which will increase the computational cost. Thus feature-based methods are not suitable for real-time applications.

3 Hardware implementation

Stereo hardware implementation can be segmented as next figure indicates

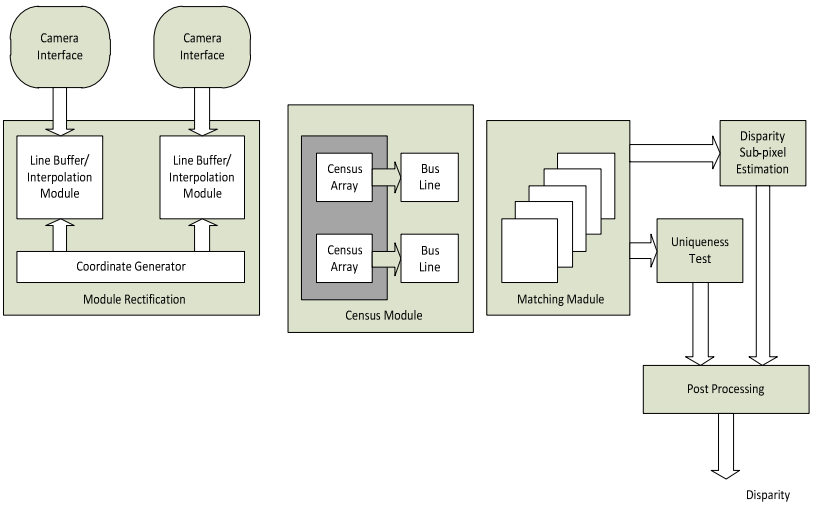


Fig. 1. Stereo diagram block. Camera module is an interface with stereo camera board. Module rectification uses a predefined algorithm for image coordinate transformation. After that, census module obtain a census vector for stereo matching and finally, postprocessing and subpixel calculus is achieved.

Hardware modules in figure 1 show stages for rectification, modified census transform and census matching. Additional hardware is used to check the uniqueness restriction and sub pixel estimation.

3.1 Rectification process

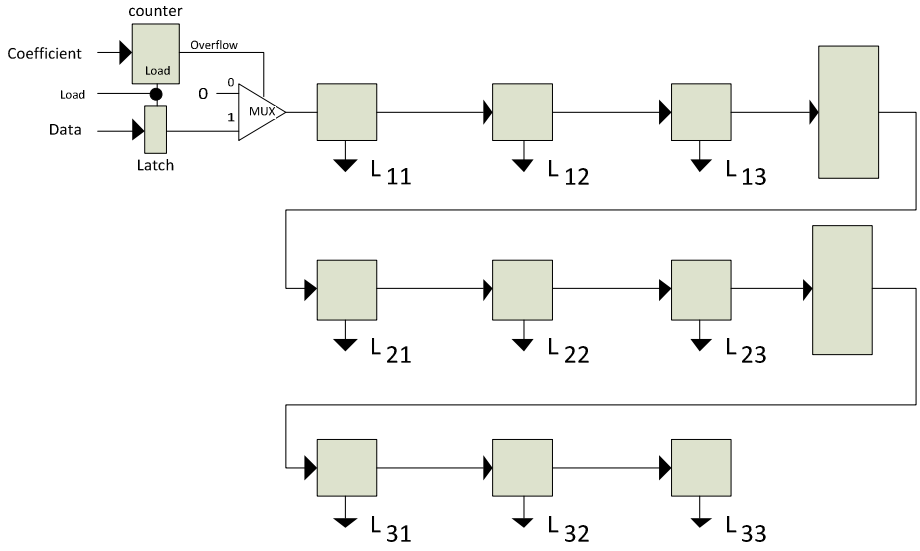
Rectification involves lens distortion correction, alignment rows in stereo images, back or reverse projection, and bilinear interpolation. In first place, due to manufacturing errors there will always appear distortions caused by the lens and a misalignment between lens and camera chip. These distortions happen before the light ray hits the image chip and consequently affect coordinates. In order to obtain a pair of rectified images from the original images after processing lens distortion and to align scanlines a homography is applied. In the equation (1) *(x, y)* and *(x', y')* are coordinates of a pixel in the original images and the rectified images, respectively.

$$\begin{bmatrix} x'_{lr} \\ y'_{lr} \\ z'_{lr} \end{bmatrix} = H_{LR}^{-1} \begin{bmatrix} x''_{lr} \\ y''_{lr} \\ 1 \end{bmatrix} \qquad \begin{bmatrix} x_{lr} \\ y_{lr} \end{bmatrix} = \begin{bmatrix} \frac{x'_{lr}}{z'_{lr}} \\ \frac{y'_{lr}}{z'_{lr}} \end{bmatrix} \tag{1}$$

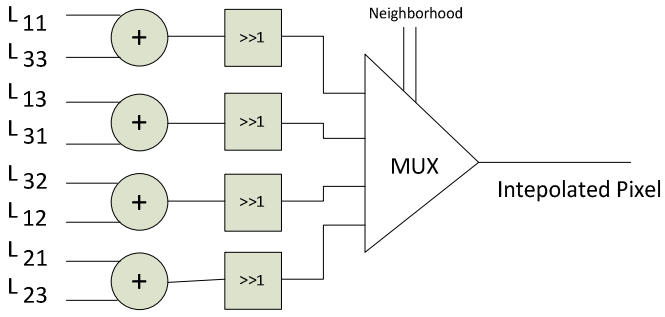
To avoid problems such as reference or coordinate duplication, reverse mapping is used with interpolation. Once the image pair is rectified, 1-D searching with the corresponding line is sufficient to evaluate the disparity.

A common hardware implementation of the rectification process is through Look-Up Tables. Only at the start, an offline calibration using MatLab is done and from calibration stage two LUTs -one for each camera – results [1]. Generated data includes: new pixel positions and best pixels for interpolation (Figure 2). Although, using LUTs directly in FPGA is possible, it is more suitable to have the LUT in external memory to extent capacity for future architectural expansions.

In hardware, rectification is applied as in [1][2] with some important differences: in proposed architecture new pixels positions are previously computed and a defined neighborhood (from calibration stage) is used to interpolate absent pixels. In this way, it is no necessary to process coordinates using a homography in the architecture. Finally, rectification module is shown in figure 2.



a)



b)

Fig. 2. Rectification module, a)Hw for the interpolation table, it detects absent pixels with coordinates from LUTs , b)Bilineal Interpolation module uses neighborhood for establish pixels values in real time.

In figure 2a, new pixel coordinates generates black pixels in the rectified image, so these black pixels are filled with interpolated pixels from specified neighborhood obtained in the calibration process. Architecture is based in row buffers to avoid store images. In figure 2b, a module for bilinear interpolation is shown. In both cases 3x3 windows are used for explication purpose, real architecture uses 11x11 windows.

3.2 Stereo Processing

Stereo matching is divided into two stages, the census transform stage and the correlation stage. In the census transform stage, the left and right images are transformed into images with census vector pixel values instead gray-level intensity. This transform is a non-parametric measure for window based processing used during the matching process for measuring similarities and obtaining the correspondence between the points into the left and right images. In a classic hardware implementation for census transform, pixel neighborhood associated with central pixel needed to be accessed simultaneously to calculate in a single instant census vector. Last technique increases HW requirement with consequent frequency decrement and complexity [2]. However, in the presented architecture, a modified census transform is used. Proposed modification just takes, instead central pixel for census, first right pixel in central row. In test changes with proposed modification were not appreciable. With this in mind, necessary hardware to calculate census transform decrease significantly. But pixels in the same column need to be present at same time. To achieve this, architecture uses a scan-line buffer and windows buffer. Scan-line buffer is memory which is able to contain a row from input image in order to synchronize data and avoid additional access to external memory and a window buffer, which is a set of shift registers with the pixels belong to the window. Such windows buffers in other architectures [5][15] consists of 8 bit registers, but in this case registers contains only one bit. The scan-line buffer used in the proposed system consists of 10 dual-port memories, and each memory can store one scan-line of an input image. Assuming that the coordinates of

the current input pixel are *(x, y)* and the intensity value of the pixel is *I(x, y)*, the connections between the memory are shown in Fig. 3.

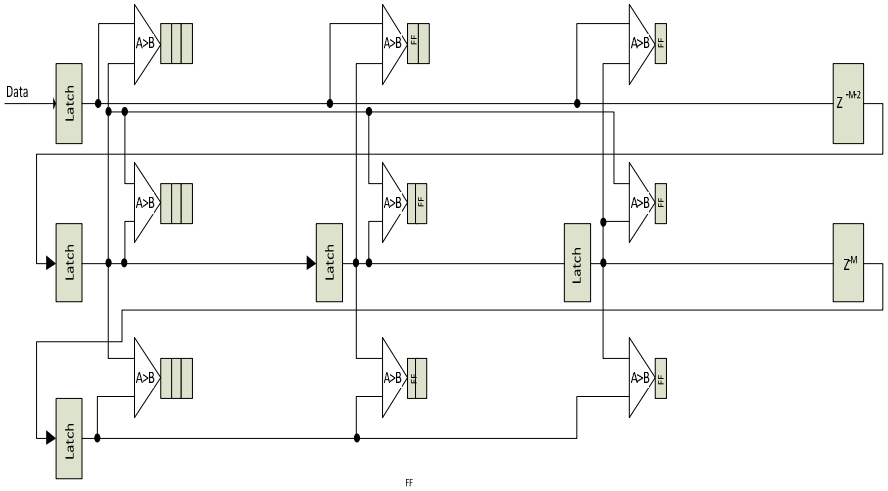


Fig. 3. 3x3 Architecture example. Implemented architecture is 11x11. Blocks in dark gray are 1 bit registers. In this case, after 3 latency clocks, registers contains a valid census transform. There are in the architecture two modules one for each channel.

Figure 3 shows a scan-line buffer converting a single row pixel input into a column pixel vector output. A window buffer is a set of 1 bit shift registers, but central line can store one 8 bit pixel from input image. One bit registers with comparison modules store values and delay results until all columns have been processing. Intensity values in central row register are shifted from left to right at each pulse clock to build the census transform. In comparison, with the classical technique where all 11×11 pixel registers are implemented, it is obtained a windows buffer with 121×8=968 bits, but in proposed architecture just 21 registers (11 left column register + 10 central row registers) ×8+10×10=261 bits are necessary to implement census transform.

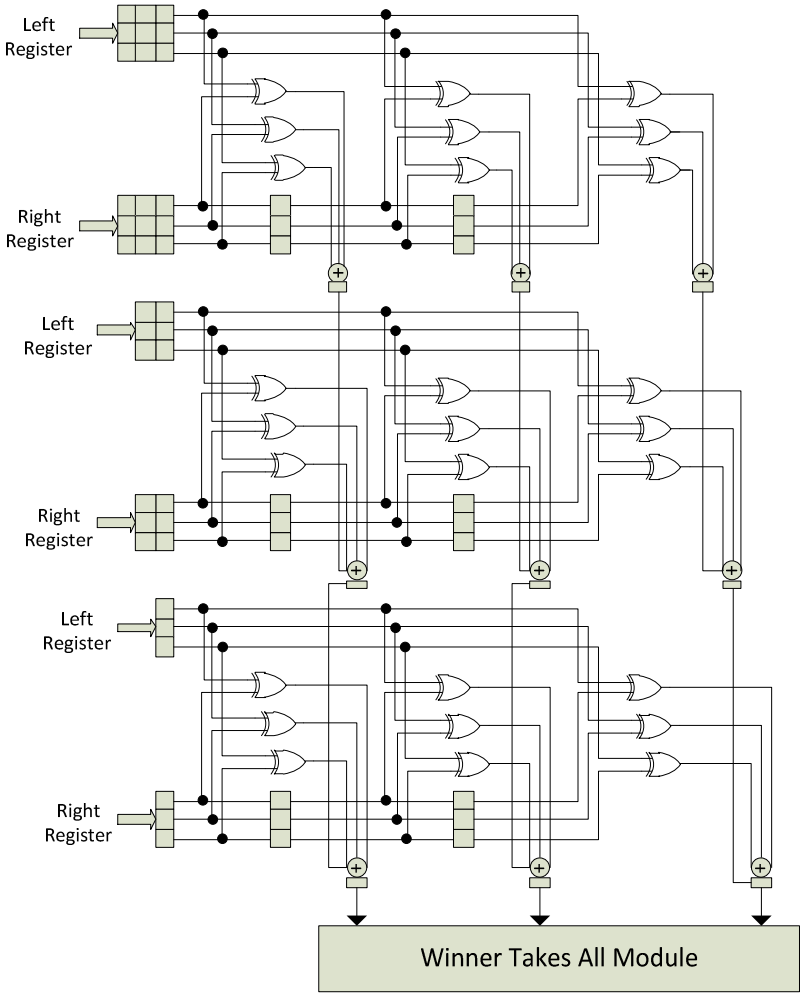


Fig. 4. Correlation module uses Hamming distance to establish similarity. Here is shown a 3×3 windows module with a 3 level disparity as example. Proposed architecture uses 50 disparity levels.

Correlation stage evaluates the correlation between the census vectors generated by the left and right census transform. Stereo procedure in the correlation stage is: *N* hamming distances are evaluated using a template window for a pixel in the left image and the corresponding *N* correlation windows for pixels in the right image. After the comparison, the two pairs with the shortest hamming distances are used to define the resulting disparity. Since the windows being compared can be regarded as bit vectors, it is possible to obtain the hamming distance by counting '1' in the vector after applying an XOR operation [5]. Here, proposed architecture introduces another hardware artifice to decrease hardware: proposed architecture uses pipeline from fig-

ure 3 to calculate Hamming distance from left and right images without using a sum combinatorial tree (figure 4).

In order to decide upon the disparity result, the template window in the left image is compared with all *N* candidate windows from the right image. First, the census vector from the census transform module is delayed for *N* pixel clocks. Next, the distance between any two census vectors is calculated. Tournament selection method is used to find the shortest distance among these *N* hamming distances and winner takes all. The candidate window, which has the shortest distance from the template window, is selected as the closest match, and the coordinate difference of the selected windows along with the x-axis is extracted as the disparity result. In proposed system there are not sum combinational trees but RTL structures which cut combinational paths and increase frequency. Most left blocks are one bit registers en figure 3.

3.3 Tracking module

Tracking objects by correlation is a basic technique in tracking systems. In this case, add a tracking module in a real time architecture which does correlations with windows it is not a problem, although this tracking module just is added to one camera because of hardware circuitry complexity. Figure 5 shows the tracking module

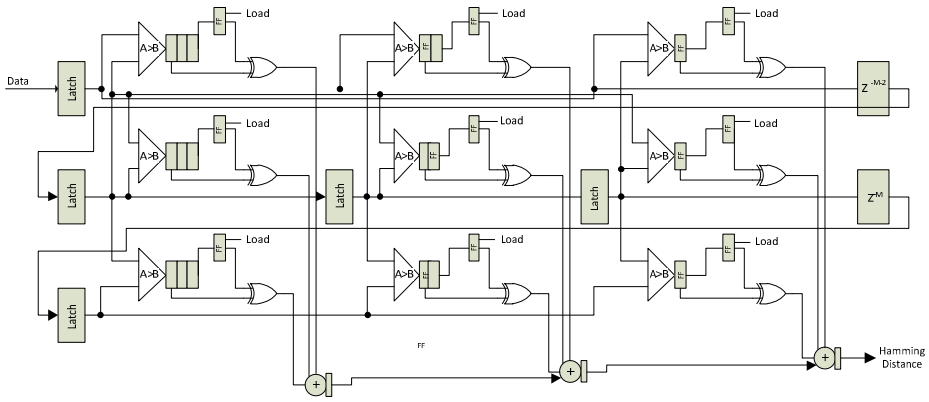


Fig. 5. 3x3 module for object tracking.

From figure 5, module is exactly same that census module except for some details: additional flip flop in the output from shift registers, XOR gates to evaluate hamming distance and sum sequential tree. Output from sequential tree is input to a coordinate module that gives to the architecture a coordinate range where objective tracking is calculated.

3.4 Post Processing

Post processing uses online subpixel disparity, i.e., no winning disparity is stored with near pixels to after calculate a subpixel estimation. Additionally, in order to avoid false correspondences, classical left-right check is substituted by uniqueness restriction. Briefly, assume that the left image is chosen as reference and the disparity candidates range build a disparity array *[0, ..., d_{max}]*. *L(x,y)* is one point of left image, the algorithm searches for the best candidate by minimizing, in this case, a matching cost C. Suppose now the best match found for *L(x-a+ d_{max}, y)* is *R(x, y)*, with matching cost *C(x-a+d_{max}, x, y)*. And another point of the left image *L(x-b+d_{max},y)* has previously been matched with *R(x, y)* with cost *C(x-b+d_{max}, x, y)*. And another point of the left image *L(x-b+d_{max},y)* has previously been matched with *R(x, y)* with cost *C(x-b+d_{max}, x, y)*. Based on the uniqueness constraint, we conclude that at least one of the two matches is incorrect and only the match with minimum cost is retained. This implies that the proposed approach allows for recovering from previous errors as long as better matches are found during the search. During the implementation, it only needs to set up d_{max} registers to keep track of the best match and corresponding matching cost for right image points in the range of interval. The match newly created for *R(x,y)* is compared with previous match, and the one be replaced will be labeled “incorrect”.

4 Results

The proposed real-time stereo vision system is designed and coded using VHDL and implemented using a Spartan XC3S1000 FPGA from Xilinx.. The implemented system interfaces two MT9M112 CMOS sensors from Micron as a stereo camera pair. Table I summarizes the device utilization reports from the Xilinx synthesis tool in ISE release 13i, Used FPGA resources for architecture are indicated in table 1:

Table 1. Device utilization summary

	Used	Available	Utilization
Occupied Slices	7554	7680	98%
Rectification	2311		30%
Census transform	1034		13%
Hamming Distance	1566		20%
Correlation	1867		24%
Post processing	776		10%

Since the hardware was built for real-time processing of an incoming image, the disparity results of the proposed design were generated through HDL functional simulation, i. e., a test bench was generated to probe architecture.

Table 2 compares different architectures with the proposed in this paper.

Table 2. Real time performance of reported stereo vision systems based on FPGA

Implemented system	Image Size	Matching method	Disparity	Fps
MSVM-III	640x480	Census	64	30
Kunh et al.	256x192	Ssd/census	25	50
Proposed Architecture	640x480	Census	50	52

Table 2 shows some systems found in literature the proposed architecture is based in a cheap FPGA and it is comparable with bigger and elaborate systems. Proposed architecture uses less hardware and is more suitable for mobile applications like robotic platforms. DeepSea founded in the literature is a multi board platform so it can be used for comparison.

Finally, Fig. 4 is the resultant disparity image from test images captured in different environments. The images were processed and obtained from the implemented system at different post-processing levels.

Fig. 6. Test stereo images and disparity maps from the proposed architecture

5 Conclusions

In this article, we have proposed a high performance FPGA-based stereo vision system with minimum cost using modified census transform, which can provide dense disparity information with additional sub-pixel accuracy in real time. The proposed system was implemented within a single FPGA including all the pre and post-processing functions such as rectification and uniqueness test. To achieve the targeted performance and flexibility, architecture was focused on the intensive use of pipelining and modularization. The proposed system can be used for higher level vision applications such as intelligent robots, surveillances, automotives, and navigation. Additional application areas in which the proposed stereo vision system can be used will continue to be evaluated and explored.

6 References

1. Vancea, C. and S. Nedevschi: LUT-based Image Rectication Module Implemented in FPGA. Intelligent Computer Communication and Processing, 2007 IEEE International Conference on, pages 147{154, Sept. 2007.
2. S. Birchfield and C. Tomasi. Depth discontinuities by pixelto-pixel stereo. Int. J. Comput. Vision, 35(3):269–293, 1999.
3. M. Bleyer,M. Gelautz, C. Rother, and C. Rhemann. A stereo approach that handles the matting problem via image warping. In CVPR09, pages 501–508, 2009.
4. D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. IEEE Trans. PAMI, 24:603– 619, 2002.
5. Y. Deng and X. Lin. A fast line segment based dense stereo algorithm using tree dynamic programming. In Proc. European Conf. on Computer Vision (ECCV 2006), volume 3, pages 201–212, 2006.
6. P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. Int. J. Comput. Vision, 70(1):41–54, 2006.
7. S. Gehrig, F. Eberli, and T. Meyer. A real-time low-power stereo vision engine using semi-global matching. In CVS09, pages 134–143, 2009.
8. M. Gong, R. Yang, W. Liang, and M. Gong. A performance study on different cost aggregation approaches used in real-time stereo matching. Int. Journal Computer Vision, 75(2):283–296, 2007.
9. H. Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In Proc. Conf. on Computer Vision and Pattern recognition (CVPR 2005), volume 2, pages 807–814, 2005.
10. H. Hirschmuller. Stereo processing by semi-global matching and mutual information. IEEE Trans. on PAMI, 2(30):328–341, 2008.
11. H. Hirschmuller and D. Scharstein. Evaluation of stereo matching costs on images with radiometric differences. IEEE Trans. Pattern Anal. Mach. Intell., 31(9):1582–1599, 2009.
12. A. Hosni, M. Bleyer, M. Gelautz, and C. Rhemann. Local stereo matching using geodesic support weights. In ICIP, 2009.
13. A. Klaus, M. Sormann, and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In ICPR ’06, pages 15–18, 2006.

14. C. Lei, J. Selzer, and Y. Yang. Region-tree based stereo using dynamic programming optimization. In CVPR06, pages II: 2378–2385, 2006.
15. S. Mattoccia. A locally global approach to stereo correspondence. In 3DIM2009, pages 1763–1770, Kyoto, Japan, 2009.
16. S. Mattoccia, S. Giardino, and A. Gambini. Accurate and efficient cost aggregation strategy for stereo correspondence based on approximated joint bilateral filtering. In Proc. Of ACCV2009, 2009.
17. D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int. Jour. Computer Vision, 47(1/2/3):7–42, 2002. 1, 2, 6