International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue01,January-2015

A Survey on Hardware Implementation of Disparity Estimation Algorithms

Swathi. R
M.E. Final Year Student
SREC
Coimbatore, India

N. Kirthika M.E.,
Assistant Professor, Dept of ECE (PG-VLSI Design)
SREC
Coimbatore, India

***Abstract*—Disparity estimation is one of the most important and difficult task in computer vision. Several stereo matching algorithms have been developed, but on characterizing their performance only a little work has been done. In this paper, we present different types of real-time disparity estimating algorithms. We compare existing stereo matching algorithms and present experiments evaluating their performance and average number of bad pixels in an image These algorithms are applicable in industrial and consumer applications. In automotive environment, for robot navigation or for consumer devices like Blue-ray players or 3D television sets the resulting depth information is used.**

Keywords— FPGA; stereo matching; disparity estimation; L-HRM; SGM; Census Transform.

I. INTRODUCTION

The process of extracting 3D information from multiple 2D views of a scene is called Computer Stereo vision. The 3D information can be obtained from a pair of images, known as stereo pair. From multiple captures of the same scene it is possible to estimate the depth of the scene. Disparity in an image refers to the distance between two corresponding points in the left and right image of a stereo pair. Disparity is higher for points closer to the camera. In an image disparity of a point relates with the depth of the point but it is not the same as depth, it depends on the camera configuration. Even though stereo displays require 3D-glasses they are popular. Auto stereoscopic displays are used to avoid the need of 3D glasses. To achieve this, disparity maps with high resolution have to be generated. In this paper several disparity estimation algorithms have been presented, and their performance is compared. This paper is organized as follows: Section II presents taxonomy of stereo algorithms. Section III presents results and discussion. Section IV provides conclusion.

II. TAXONOMY OF STEREO ALGORITHMS

To support an informed comparison of stereo matching algorithms, we develop in this section, the three different types of stereo matching algorithms used to estimate the disparity. They are,

(i) L-HRM [1]

(ii) Rank-transform and Semi-Global Matching [2]

(iii) Census Transform [3]

This paper is based on the observation that all the above mentioned stereo algorithms generally perform the following steps Rectification, Disparity Estimation and Post Processing. The process of transforming stereo images, such that the corresponding points in two images have the same row coordinates is called rectification. When rectified image pairs are used, the 2-D stereo correspondence problem is reduced to a 1-D problem. Disparity estimation is to estimate the disparity map for stereo pair. Post processing is the process of utilizing some filters to refine or to smoothen the disparity map.

A. L-HRM

L-HRM algorithm [1] due to its high degree of parallelization is well suited for hardware implementation. It consists of two major processing steps: the initial disparity estimation and the post-processing.

1) Initial disparity estimation

To estimate the disparity two rectified grayscale images are required. This algorithm is a modified version of "hybrid recursive matcher (HRM) [5]**.** The HRM algorithm structure doesn't allow multiple samples to be processed in parallel. To overcome this problem, line recursion and column recursion steps are employed in L-HRM [1]. Independent lines or columns can be processed in parallel, which makes the algorithm more suitable for hardware implementation.

a) Pixel Recursion

The pixel recursion step is to introduce new disparity candidates and is similar to the pixel recursion of the HRM [4]. Following the principle of optical flow, and on the basis of spatial gradients and gradients between left and right input image a new disparity candidates is introduced by Eq.(1).

$$d(x,y) = d_i - DPD(d_i,x,y).\frac{\text{grad } I(x,y)}{||\text{grad } I(x,y)||^2} \quad (1)$$

The intensity difference of corresponding pixels gives the gradient between left and right image in Eq.(2).

$$\Delta I(x,y,d_n) = I_L(x,y) - I_R\left(x + d_{n'}y\right) \quad (2)$$

The new candidate for the recursion process is the disparity update value with the smallest absolute intensity difference between corresponding pixels in left and right image.

b) Line-wise and column-wise recursion

The line recursion considers three disparity candidates: the disparity result from the pixel recursion process, the recursion result from the previous line and the final disparity value at the same position of the previous frame. An 8x8 normalized cross correlation (NCC) is used for comparison of the candidates and for testing them, finally the winning candidate is chosen as the initial candidate for the column recursion. The direction of the line-wise recursion process switches from left-to-right and from right-to-left for each consecutive line. In column-wise recursion, only the result of line-wise recursion process and the result of the column recursion of the pixel in the same line are considered. The direction of the column-wise recursion process switches from top-to-bottom and from bottom-to-top for each consecutive line. The above mentioned steps are for left-to right disparity estimation. The column recursion is evaluated first in right-to-left disparity estimation and after that line recursion is evaluated. The below architecture represent the hardware implementation of disparity estimation block.

c) Post processing

In post-processing, first step is crosscheck (CC), which compares the disparity values at the corresponding positions is employed to locate and eliminate the false disparities. The resulting disparity is discarded and marked as unavailable if it exceeds a certain threshold. Also, to remove isolated false disparity a 3x3 speckle filter (SF) is applied in Fig. 1.

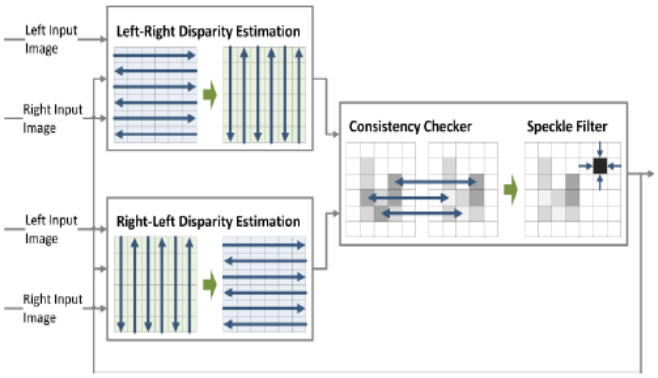

Fig. 1. Disparity estimation workflow.

To smoothen the resulting disparity a cross-bilateral median filter (CBF) is applied as the second step. For better results, up-sampling filter with a smaller 5x5 kernel is then applied. The below architecture shown in Fig. 2. represents the hardware implementation of post-processing block.

d) Implementation results

The algorithm has been implemented in hardware using VHDL and SystemVerilog and it is bit accurate. This hardware implementation creates a solution in terms of size, cost and power consumption, which cannot be achieved in software and is also suitable for consumer applications. This algorithm also provides sophisticated disparity maps for FHD in real-time. The power consumption has been estimated to be less than 5 W using the power analyzer.

Fig. 2. FPGA demonstration setup.

The final resource utilization using L-HRM algorithm is 70745 LUTs, 132893 Registers, 3MBit ROM, 6.5 MBit RAM and 625 DSP-multipliers (47%) for a 25x17 CBF,180 MHz frequency is achieved , 32-bit external memory is used and bandwidth required is 1.6 GB/s. The following table gives the Output for Middlebury Images Using L-HRM Algorithm.

TABLE I. L-HRM ALGORITHM RESULTS

Algorithm	**Results For Middlebury Images**		
	Avg. percent of bad pixels		*Max. Performance*
	Image	*%*	
L-HRM Algorithm	Cones	13.2%	1920x1080p30
	Teddy	16.3%	
	Venus	2.6%	
	Tsukuba	6.4%	

B. *Rank-Transform and Semi-Global Matching*

The rectification unit consists of a displacement vector, for which the reverse mapping is calculated on the fly from the stereo camera parameters which is calculated using the OpenCV library [4]. This Calculation requires additional hardware resources but requires no external bandwidth. The hardware architecture for rectification unit is shown in Fig. 3.

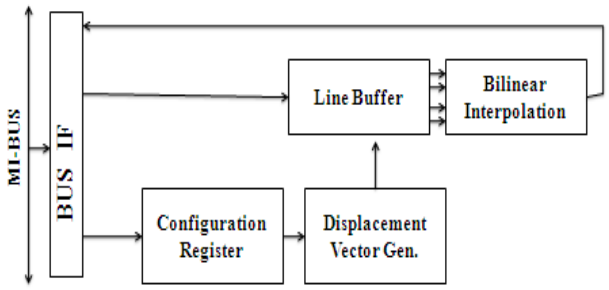

Fig. 3. Top-level block diagram of hardware architecture of the rectification unit.

1) Semi-global matching architecture

The hardware architecture for calculating the disparity maps is given in Fig. 4. For both images parallel operation is carried out for computing the rank transform and to calculate the data dependent penalty term P2. It also synchronously utilizes the same data path. For hardware-friendly, division-less calculation the penalty term P2 is adjusted to the characteristics of the rank-transform based matching cost calculation. This data is given to the systolic array by an N-row buffer, which calculates the disparities of all N rows in parallel. A median filter realizes the outlier suppression.

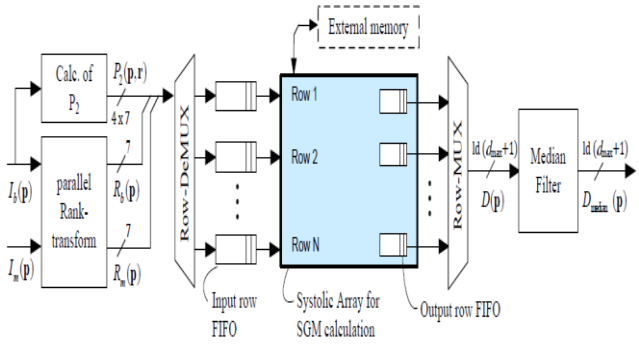

Fig. 4. Hardware architecture for calculation of disparity maps using rank-transform and semi-global matching

According to Eq. (5) processing of a pixel p is carried out sequentially and Matching costs C(p; d) is calculated by the first processing elements (C-PEs) and the path costs Lr along a path r is calculated by the PEs (L-PEs, all are identical).

$$L_r(p,d) = C(p,d) + \min\{L_r(p-r,d), L_r(p-r,d-1) + P_1, L_r(p-r,d+1) + P_1, \min_t L_r(p-r,i) + P_2\} - \min L_r(p-,i) \quad (3)$$

The results are buffered in the appropriate path cost buffers. The delays introduced by the path cost buffers define the path orientations and then the Path costs are summed to S and the disparity computation PEs (D-PEs) processes it. D-PEs locate the minimum, i.e. the correct disparity, for the disparity maps Db and Dm. the disparity map Dm is projected to the base camera by L/R-Check-PE , which executes the left/right check ,occlusion detection. A local single row buffer which functions as an output buffer is needed for projection. Boundary treatment for pixels reduces the number of entries of the cost spaces C (p; d), Lr (p; d), S (p; d), which leads to reduction in computing time. Due to the systolic array an additional latency of two pixels per parallel processed row is introduced, resulting in performance decrease of <1 % and for storing the path costs an external interim memory is required. With respect to the boundary treatment the required memory size is given in Eq.(4).

$$m = 3.e.[\log_2(L_{max} + 1)] \quad (4)$$

Where e is the number of elements per path over all pixels of a row. Let w denote the image width, then it is

$$e = w.(d_{max} + 1) - d_{max}\frac{(d_{max} + 1)}{2} \quad (5)$$

Fig. 5. Shows the hardware setup of the stereo vision system, which comprises of Xilinx Virtex-5 LX220T-1 and an Intel IXP 460 ARM X Scale RISC. The entire design is mapped on the FPGA.

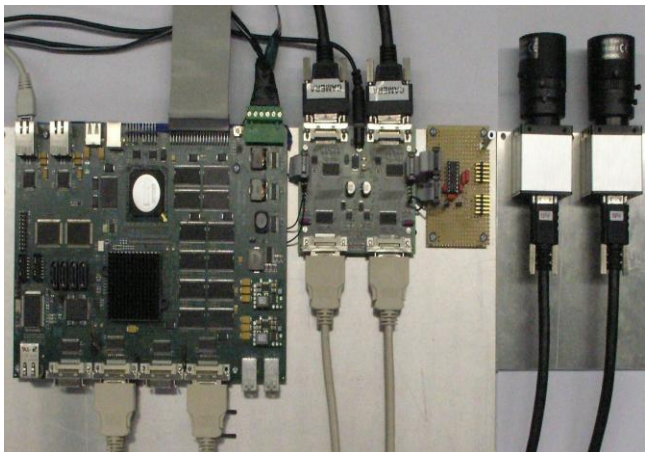

Fig. 5. Hardware setup of the stereo vision system

The RISC is running a Linux 2.6 kernel which allows debugging and controlling of the tasks executed on the FPGA. To supports Power over Camera Link (PoCL), two synchronously triggered Leutron Vision P83B-RTF cameras with 1033_779px at 30fps are connected via a converter board. A standard LCD display is used and a Channel Link interface is employed for analyzing in PC.

2) Experimental results and evaluation

Depending on the level of parallelism [2], the SGM unit achieves, frame rates from 37 up to 103 fps for VGA images with 128 px disparity range. FPGA resource usage, Increase in number of parallel processed image rows increases Resource usage for the systolic array of the SGM unit. The path cost buffers account for about 10 % of the LUTs utilized by the SGM unit. Due to the small size of required external memory, it has been emulated using Block RAMs (BRAM). The following table gives the output for Middlebury images using Rank-Transform and Semi-Global Matching algorithm.

TABLE II. RANK TRANSFORM AND SGM RESULTS

Algorithm	Results For Middlebury Images		
	Avg. percent of bad pixels		*Max. Performance*
	Image	*%*	
Rank-Transform and Semi-Global Matching	Cones	9.5%	1920x1080p30
	Teddy	13.3%	
	Venus	4.1%	
	Tsukuba	6.8%	

C. Census Transform/Correlation

The census transform [3] requires $(r + 1) \cdot m2 \cdot (n2 - 1)$ times of pixel subtraction, $r \cdot m2$ times of hamming distance measurement for $n2 - 1$ length of bit vector, an $r \cdot m2$ (times of hamming distance summation) operations to compute the disparity for an $m \times m$ and $n \times n$ windows, correlating with a searching range, r. Hardware architecture of the proposed real-time stereo vision system is shown in Fig. 6. A single data path controlled by a controller module performs all the stereo vision process. The structure of the data path is defined by three modules rectification, stereo matching, and post-processing. Intensive use of parallelism and pipelining, single pixel clock synchronization is employed to eliminate the use of an external image buffer. To achieve parallelism, each module is divided into a set of simpler functional elements and then it is replicated and executed in parallel.

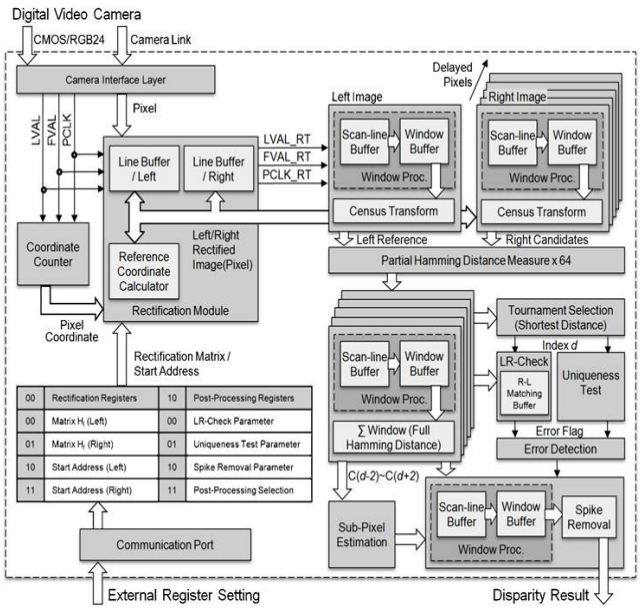

Fig. 6. Hardware architecture for stereo matching (census transform/correlation)

To achieve good performance and scalability, Synchronization of all modules with a single pixel clock is done. This will increase the maximum allowed frequency, thereby maximizing the throughput of the system.

1) Stereo matching module

After rectification [3], stereo matching is performed as shown in Fig. 6. It consists of two stages, the census transform stage and the correlation stage. In the first stage, instead of gray-level intensity pixel values the left and right images are transformed into images with census vector pixel values .A scan-line buffer and a window buffer is used to access the neighboring pixels of the processing pixel simultaneously. The pixel column vector, generated from the scan-line buffer is given to the right-most column of the window buffer, and at each pixel clock, registers intensity values are shifted from right to left. As a result, in window buffer all 11×11 pixels exist, and it can be accessed simultaneously. By comparing the intensity of the centered pixel, and its neighboring pixels the census vector is obtained. The relation between each pixel and its neighborhood is generated in the form of an image and is given to the correlation stage. The correlation stage uses a window-based method to evaluate correlation between the census vectors. After the comparison, to define the resulting disparity, the two pairs with the shortest hamming distances are used. The result is represented by 7 bits range, since the hamming distance between two pixels can be from 0 to120, and is taken as the input of the correlation window. The candidate window, which has the shortest distance from the template window, is selected as the closest match, and the coordinate difference of these selected windows is extracted as the disparity result.

2) Post-processing module

It contains four sub-modules namely LR-check, uniqueness test, spike removal, sub pixel estimation to increase the accuracy. The LR-check [3] is required to remove mismatches caused by occlusion. By comparing the two disparities, one from the vertical direction and the other from the diagonal direction, we can decide whether this disparity passes the LR-check. To validate the disparity result generated by the stereo matching module, uniqueness test is also performed can decide whether this disparity passes the LR-check. In uniqueness test, we keep a track of three lowest ranking hamming distances, represented as $C(d_i)$, $C(d_i - 1)$ and $C(d_i + 1)$, where $C(d_i)$ is the hamming distance at the selected disparity result. The sub-pixel estimation gives additional precision. Finally Spike removal is implemented. The hardware setup for this algorithm is shown in Fig. 7.

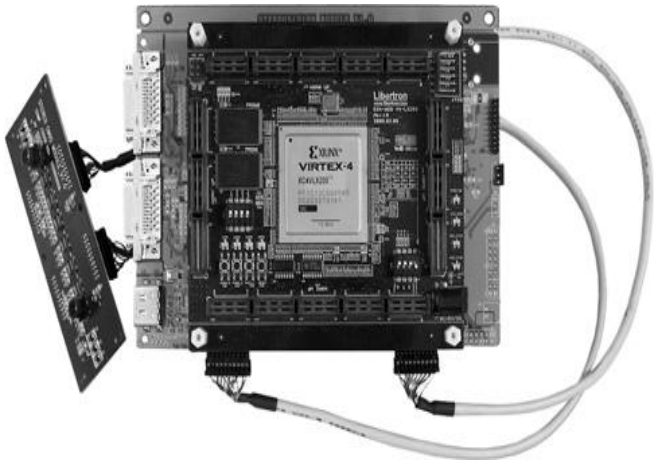

Fig. 7. Implemented real-time stereo vision system based on FPGA.

3) Experimental results

The algorithm has been designed and coded using VHDL and implemented using a Virtex-4 XC4VLX200-10 FPGA from Xilinx. The number of used slices and the maximum allowed frequency of the proposed system are 51,191 (about 57% of the device) and 93.0907 MHz the maximum performance of the proposed system to occur when using a camera with 230 f/s is used. The following table gives the output for Middlebury images using Census Transform /Correlation

TABLE III. CENSUS BASED TRANSFORM/CORRELATION

Algorithm	**Results For Middlebury Images**		
	Avg. percent of bad pixels		*Max. Performance*
	Image	*%*	
Census Transform /Correlation	Cones	17.6 %	640x480p230
	Teddy	21.5%	
	Venus	5.3%	
	Tsukuba	11.6%	

III. RESULTS AND DISCUSSIONS

The comparison of three different types of algorithms on hardware is done. All the algorithms performed the process of Rectification, Post-Processing, Hardware Implementation on Middlebury stereo images. The average percentage calculation of bad pixels has been done in all the algorithms and its maximum performance is obtained. The following table gives the comparison of three different types of algorithm on Middlebury stereo images.

TABLE IV. COMPARISON OF ALGORITHMS

Algorithm	**RESULTS FOR MIDDLEBURY IMAGES**				
	Avg. percent of bad pixels				*Max. Performance*
	cone	*teddy*	*venus*	*tsuba*	
[1]	13.2%	16.3%	2.6%	6.4%	1920x1080p30
[2]	9.5%	13.3%	4.1%	6.8%	640x480p103
[3]	17.6%	21.5%	5.3%	11.6%	640x480p230

IV. CONCLUSION

In this paper three different types of stereo matching algorithms are analyzed and performance comparison is made. It is clear that the L-HRM algorithm is the best suited for disparity estimation, its performance is higher compared with other two algorithms in terms of Resolution, Average percentage of bad pixels, Speed.

ACKNOWLEDGMENT

I wish to thank my institution, 'Sri Ramakrishna Engineering College, Coimbatore' for giving me the opportunity to write a research paper. A special thanks to my Head of the Department, Dr. M. Jagadeeswari for encouraging us, Co-coordinator Mr. C. S. Manikanda Babu and to Ms. N. Kirthika for her support and guidance throughout and without whom, this work would have not been possible. Last but not the least; I would like to thank the authors of the various research papers that I have referred to, for the completion of this work.

REFERENCES

[1] Martin Werner, Benno Stabernack, and Christian Riechert, "Hardware Implementation of a Full HD Real-time Disparity Estimation Algorithm", IEEE Transactions on Consumer Electronics, Vol. 60, No. 1, February 2014

[2] C. Banz, S. Hesselbarth, H. Flatt, H. Blume, and P. Pirsch, "Real-time stereo vision system using semi-global matching disparity estimation: Architecture and FPGA-implementation," in Proceedings of the IEEE International Conference on Embedded Computer Systems, pp. 93-101, 2010.

[3] S. Jin, J. Cho, X. D. Pham, K.-M. Lee, S.-K. Park, M. Kim, and J. Jeon,"FPGA design and implementation of a real-time stereo vision system, "IEEE Transactions on Circuits and Systems for Video Technology, vol.20, no. 1, pp. 15-26, Jan. 2010.

[4] G. Bradski and A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV library. O'Reilly Media, Inc., 2008.

[5] C. Riechert, F. Zilly, M. Muller, and P. Kauff, "Real-time disparity estimation using line-wise hybrid recursive matching and cross-bilateral median up-sampling," in Proceedings of the IAPR 21st International Conference on Pattern Recognition, pp. 3168-3171, 2012.