

Original Research Paper

Optimised Implementation of Dense Stereo Correspondence for Resource Limited Hardware

^{1,2}Deepambika Vadakkettathu Anirudhan and ^{3,4}Malangai Abdul Rahiman

¹Department of ECE, Karpagam Academy of Higher Education, Coimbatore, Tamilnadu, India

²Department of ECE, LBS Institute of Technology for Women, Trivandrum, Kerala, India

³Karpagam Academy of Higher Education, Coimbatore, Tamilnadu, India

⁴Pro-Vice Chancellor, APJ Abdul Kalam Technological University, Trivandrum, Kerala, India

Article history

Received: 21-06-2018

Revised: 16-07-2018

Accepted: 17-10-2018

Corresponding Author:

Deepambika V.A.

Department of ECE, LBS

Institute of Technology for

Women, Trivandrum, Kerala,

India

Email: deepambika.va@gmail.com

Abstract: Computer stereo vision is a passive sensing technique that helps to recover 3D information of an environment from 2D images. The stereo correspondence is a challenging task that finds out matching pixels between the stereo image pair based on Lambertian criteria and its result is a disparity space image. The depth of the objects from the camera can be calculated from this disparity value by using the principle of triangulation. For the vision guided robot navigation, the requirement of stereo matching algorithms on low power dedicated hardware that can achieve a high frame rate is unambiguous. A new, highly optimized implementation of correlation based, Sum of Absolute Differences correspondences algorithm on a low cost resource limited FPGA is presented here. This System-on-Programmable-Chip architecture based system achieved a higher frame rate of 50 fps with 64 disparity levels without using a microprocessor. On performance evaluation, the disparity map shows a maximum error of 0.308% only. This reconfigurable parallel processing, high speed architecture of the algorithm implementation takes only 43% of available resources of low density Altera Cyclone II. This hardware implementation of stereo vision system outperforms in terms of accuracy, speed and resource utilization of all the other existing stereo systems of its similar kind. Also, it offers a better trade-off between run-time speed and accuracy and is found suitable for most of the range finding real-time applications.

Keywords: Stereo Vision, Stereo Correspondence, SAD, FPGA, SoPC

Introduction

The primary intention of robotic vision is to enable robots to cope with its surroundings in order to perform various tasks like navigation in unknown cluttered environments, moving object tracking and pick and place applications Ude (2010). Among the various sensory modalities, the most dominant one is vision. But even the latest intelligent vision system that exists in computer vision is still far away from cognitive capabilities of the human visual system and the computer stereo vision tries to mimic human vision.

The stereo vision system is an inexpensive passive sensing technique that allows to obtain precise 3D information of the surroundings from 2D images and there is no interference with other sensing devices if multiple sensors are working in the same environment.

Based on the Lambertian criteria, the stereo correspondence algorithm detects conjugate pair of pixels between the input stereo images and the result is a disparity space image. The depth of the objects from the camera is inversely proportional to this disparity values. The main categories of stereo correspondence algorithms are local and global methods Scharstein *et al.* (2001). The correspondence steps start with the matching cost computation, cost aggregation, disparity computation and finally the disparity refinement. Local stereo matching can be of feature based and area based methods that will result in sparse and dense disparity maps respectively. In the area based, also known as correlation based or window based local approach, a window of required size centered at a pixel of interest from one input image is used to search in the other image for similar intensity levels. This similarity check is done

based on a cost function and they perform the first three steps. The disparity refinement is the post processing step during which the mismatches are removed from the generated disparity maps or the subpixel disparities are estimated. In the area based correspondence, the matching costs that are aggregated within the correlation window will be assigned with same disparity Ding *et al.* (2011). These kinds of window based approach leads to reduced computational task with decent dense disparity results and thus having dominant applications in most of the real time systems. The size of the window selected should be large enough to incorporate the intensity variations, but it is to be small enough to avoid 'edge fattening' so as to preserve sharp depth discontinuities at the object boundaries Yoon and Kweon (2006). These window based approaches uses Winner-Take-All (WTA) optimization. Most of the local stereo correspondence gives good results if the input image pair is of same radiometric condition. But in real world environments, there will be radiometric variations in the input images, due to illumination variations and the camera exposure time variations. Various non-parametric approaches can be used to handle these changes Sarika *et al.* (2015).

Global methods make explicit smoothness assumptions and it uses a pixel-based matching cost. They often skip the cost aggregation step and will do some iteration steps or optimization methods after the disparity calculation. Birchfield and Tomasi (1999) Global methods give accurate results, but are computationally expensive due to their iterative nature and are not suitable for real-time purpose. They are preferred when the accuracy of the algorithm is the main criteria in applications such as precise 3D scene reconstruction.

The features of global and local stereo matching methods are combined with the semi-global matching algorithm Hirschmuller (2005; 2008) which offers the pixel wise matching that can maintain depth discontinuities as in global methods and can provide faster results as in local methods. Even though they are able to meet the real time requirements of most of the applications, they are not suitable for hardware implementation, they require more memory because of their iterative nature. They can be implemented only on the more costly high density hardware resources.

In most of the real time applications such as autonomous robot navigation, civil protection, etc., the target is the efficiency of the algorithm. Often the demand of computationally simpler algorithm in combination with good accuracy is inevitable. The local window based methods can provide decent dense disparity maps with real-time or near real-time performance.

For the vision guided autonomous robot navigation, requirement of vision algorithms implemented on a low power dedicated hardware that can achieve a high frame rate is unambiguous. Often the use of a personal computer is not an apt solution for most of the applications such as

space robots and rescue robots Lazaros *et al.* (2008a). To meet the computational demand as well as the real-time requirements of stereo algorithms, they can be implemented by using various hardware systems, such as FPGA, GPU and ASIC Jin *et al.* (2010). Even though GPUs are attractive platforms for PC based applications, its higher power requirement limits their use. ASIC implementations are process dependent, needs more prototyping time and are highly expensive except for bulk production. An FPGA consist of programmable logic blocks that can perform parallel processing. FPGAs are apt choice for the hardware accelerations of stereo algorithms since they are less expensive, possess shorter design cycle, less power consumption and also they have great flexibility for adapting to new specifications. The high computational demand of global correspondence limits their hardware implementation possibility, but the local methods are computationally simpler and can be easily implemented on hardware by incorporating suitable optimisation methods. Also, the local methods, such as Normalized Cross-Correlation (NCC), Sum of Squared Differences (SSD) etc. consist of multiplication and division. The number of logic elements required for such operations are more. Therefore, the implementation of these algorithms on FPGA leads to resource saturation. Sum of Absolute Differences algorithm (SAD) is a suitable choice to deal with these challenges as it involves only simple subtraction, addition and a probable sign change. This article discusses a new simple architecture for the hardware implementation of area based SAD correspondence. This algorithm implementation is highly optimised for resource limited hardware and is benefited by its computational simplicity and thus well suited for real time applications.

The early work in stereo vision on FPGA was done by Woodfill and Von Herzen (1997). They implemented 32-bit census transform on PARTS engine which was made up of 16 Xilinx and 16 numbers of 1MB SRAMs with 24-disparity levels. Miyajima and Maruyama (2003) implemented a stereo vision system on single FPGA that uses SAD algorithm for stereo matching with a speed of 20 fps. Perri *et al.* (2006) presented a stereo correspondence algorithm with SAD as the matching cost on Xilinx Virtex4 FPGA. Mingxiang and Yunde (2006) presented a trinocular stereo vision on Xilinx XC2VP40 FPGA. Lazaros *et al.* (2008b) proposed a stereo correspondence algorithm using absolute differences for hardware implementation with an enhanced disparity estimation. Hadjitheophanous *et al.* (2010) discussed about the stereo algorithms that are suitable for resource limited hardware.

Zhang and Chen (2013) implemented a stereo matching system on a single FPGA using 32-bit Nios II processor, which attains 23 fps using SAD stereo matching with 5×5 window size. In all of these real-time

stereo vision systems, there exists a trade-off between resource cost and the requirement of real time frame rate. Also, these existing approaches are done with highest resource utilization and their disparity results are comparatively less accurate. This work presents a new highly optimised System on Programmable Chip (SoPC) architecture on FPGA for doing stereo matching to achieve accurate disparity map with minimum resource utilization and higher processing speed.

Features and Contributions

Our work introduces a new highly optimised parallel processing architecture for stereo correspondence, which is devoid of any microprocessor usage. This architecture is suitable for the implementation of dense stereo matching with the real time frame rate on a resource limited hardware. The stereo matching is performed on a single Cyclone IIEP2C20F484C7 FPGA chip using Verilog and SoPC architecture. The analysis and synthesis of SAD stereo matching are done using Altera Quartus II software. This system can handle image size of 320×240 and it achieved a speed of 50 f/sec with 64 disparity levels. The obtained disparity map shows a maximum error of 0.308% only. The hardware implementation of this reconfigurable parallel processing high speed architecture of the algorithm takes only 43% of available resources of low density Altera Cyclone II. This implementation offers a better trade-off between run-time speed and accuracy and is well suitable for many real-time applications.

Sum of Absolute Differences Algorithm

SAD is a simple area based correspondence algorithm. This block matching algorithm assumes that the intensity of pixels within a window, centered at a pixel has similar intensity levels. The correspondence search is done by correlating the intensity of pixels within a reference window, to its similar one in the other image, based on the cost function SAD Hamzah *et al.* (2010). The dissimilarity is the sum of absolute differences between the pixel intensities, therefore it is named as sum of absolute differences algorithm. The 2-D correspondence search (along the x and y direction) can be limited to 1-D correspondence search (along the x direction only) if the input images are rectified. In rectified images, a point in left view and its conjugate pair in the right view will be on the same epipolar line Fusiello *et al.* (2000). Here all the epipolar lines in two views are in parallel with the base-line and the y coordinates of corresponding pixels will be same.

Calculating the sum of differences among the absolute intensity values for a block around a center pixel is simple. Let $I_L(x,y)$ be the intensity of the pixel

at coordinates (x, y) in the left image and $I_R(x,y)$ be the intensity of pixels in the right image, N defines the total number of pixels in either direction from the center pixel in the window. Therefore, the size of the window will be $(2N + 1)$.

Disparity d is the shift in x coordinate of the matching pixels in the image pair and the cost function SAD used for similarity measure can be computed using the Equation 1:

$$SAD(x, y, d) = \sum_{x=i}^{i+N} \sum_{y=j}^{j+N} (I_L(x, y) - I_R(x + d, r)) \quad (1)$$

Each pixel in the reference image window is compared with the pixels on the same epipolar line in the target image. SAD values are computed for all the pixels within a window and the winner is the disparity corresponding to SAD minimum. The ideal SAD value of the matching pixels (ideal SAD minimum) is zero. The disparity values can be calculated using Equation 2:

$$Disparity(x, y) = \min_{d \in D} (SAD(x + d), y) \quad (2)$$

The disparity is the shift in the x coordinates between the two similar pixels in either view corresponding to the SAD minimum. The disparity values of the pixels within a window can change from zero to a maximum disparity value d_{max} . This process repeats for each block until the disparity value is computed for the entire image and the resultant image is called Disparity Space Image (DSI). This window based method considers that the pixels within a window will have same disparity levels and the disparity value is computed for best matching pixels only.

Architecture of Stereo Correspondence

The proposed SoPC architecture for the disparity computation composed of three modules which are shown in Fig. 1.

Disparity Computation Unit

The disparity values are computed based on the cost function SAD. The disparity computation is done only for SAD minimum based on the *winner-take-all* optimisation. Here the one input image pair is divided into 64 sub images. Then the disparity values are computed using 64 SAD modules for one sub image frame.

Stereo Matching Controller (SMC)

SMC controls the entire input data read/write operations for the cost computation and the resulting disparity read/write operation.

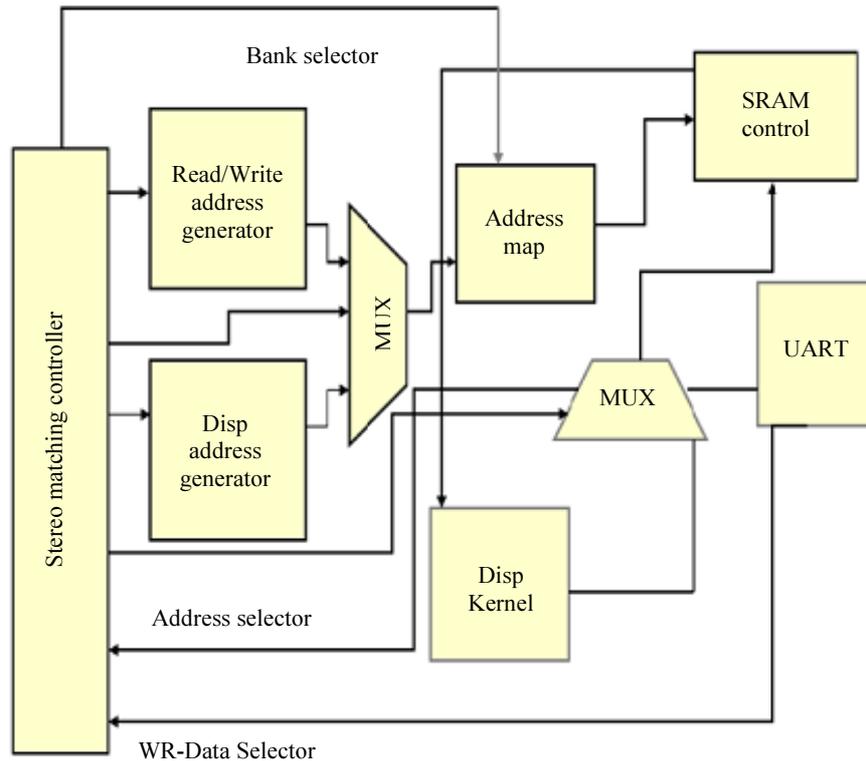


Fig. 1: SoPC architecture of disparity computation

UART

The serial interface RS-232 is used to connect FPGA to a PC.

Design of Disparity Computation Unit

Here the window size of 5×5 is used for the similarity check. The minimum and maximum disparity values are taken as zero and 63 respectively, and hence the maximum disparity levels will be 64. The SAD minimum and the corresponding disparity values are computed using the Equation 1 and 2. The input image size of 320×240 is taken for the computation of SAD minimum and is divided into 64 sub frame of size 5×240 . The disparity calculation of one sub frame is completed using 64 SAD modules within 64 clock cycles of 50 MHz frequency. 5×5 window size is chosen for the correspondence search and the SAD minimum is computed using 25 parallel adders. Figure 2 shows the block level representation of the disparity computation of one sub frame using SAD. One byte of image pixel is taken as one shift tap. During the disparity computation process 25 and 1200 bytes of data are taken from the left and right images respectively. Figure 3 shows the parallel processing frame work of SAD calculator.

The Disparity Segregator (DS) unit computes the SAD minimum using parallel comparators and the disparity is the change in the index value of the x

coordinates. The comparator module in the DS unit compares the adjacent SAD values and gives a bit level output. This bit level determines the state of selector line of the multiplexer for getting SAD minimum. This SAD minimum is used for the comparison with the neighborhood SAD value in the succeeding step.

The multiplexer prior to every absolute difference calculator (AD), receives the input pairs for absolute difference calculation. The output from each absolute difference calculator gets segregated in the parallel adder. From this the disparity segregator selects SAD minimum and outputs the disparity corresponding to every SAD module. This process will be completed for the entire sub frames and the Disparity Space Image (DSI) is computed for the complete input image.

Design of Stereo Correspondence Controller

The controller handles the entire stereo correspondence. The three main functions associated with the correspondence controller are memory management, disparity write process and stereo correspondence control.

Memory Management

Cyclone II EP2C20F484C7 possesses two Dual-port RAMs, each of them occupies a memory space of 16,384 bytes ($1,024 \times 16$) and they act as memory (line

buffers) for the stereo inputs. The stereo correspondence controller uses these buffers to initialize and update the data taken from the buffer of the Synchronous SRAM. The buffers continuously store 16 lines of data. The maximum allowable pixels per line is 1,024. The line buffer follows the direct address mapping. The formula for the address mapping is given by Equation 3:

$$A = (J \bmod 16) * N_{pixel} + I \quad (3)$$

where, A is the pixel address, I and J are the coordinates of the input pixel and N_{pixel} is the total horizontal pixels. The Finite State Machine (FSM) of the stereo correspondence process is shown in Fig. 4.

After the line buffer initialization, the FSM goes to the state of IDLE. On receiving the right and left images, the FSM enters into the SAD computation process and then the disparity is computed. Figure 5 shows the FSM of the memory management process. In this FSM, there are nine states and each state is enabled by the start and update signal from the Stereo Correspondence Controller. The Table 1 shows the state table for the disparity computation. When the signal $Start = 1$ the initialization of the two line buffers will occur and the FSM reads 16 pixel lines from the stereo pair. After this, the FSM enters into an IDLE state. When the signal $state\ update = 1$, current state of the buffer gets updated. The FSM reads 4 bytes of input image data from the SRAM and overwrites the data in the line buffers and again its state will be IDLE.

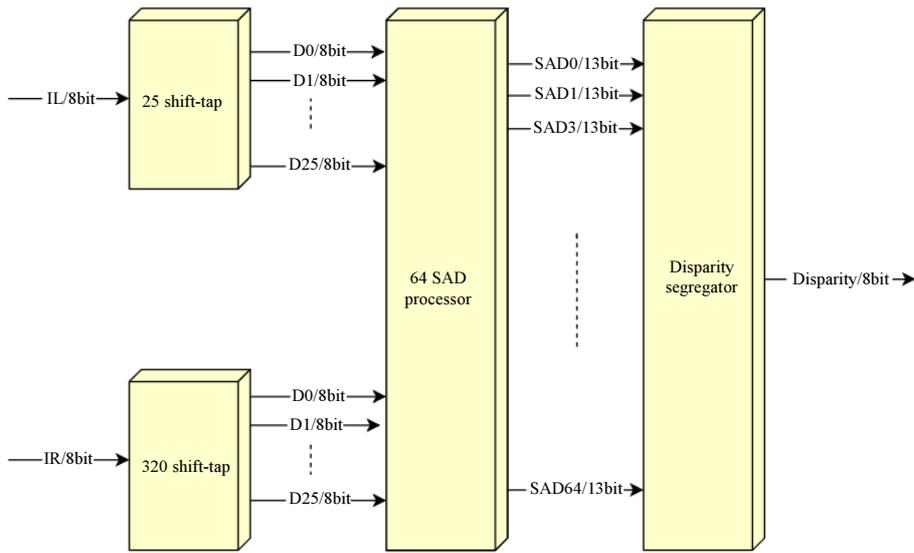


Fig. 2: Block schematic of disparity computation for one sub frame

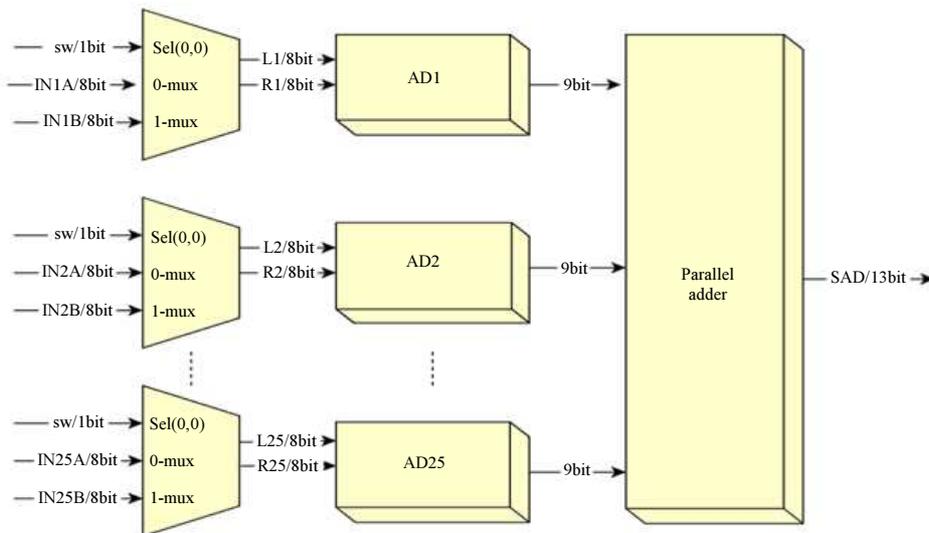


Fig. 3: Parallel Processing Frame Work of SAD Calculator

Disparity Write

After the estimation of disparity, it is to be written to the disparity table with the help of direct memory access controller. The address for the disparity write operation is given by the Equation 4:

$$A_{write} = A_{DRAM} + 2 * (j * H_{pixel} + 1) \tag{4}$$

Here A_{write} represents the address for disparity write, A_{DRAM} represents the initial address of the RAM. I and J are the coordinates of the pixel. The result, Disparity Space Image (DSI) will be stored in the on-chip FPGA memory.

Stereo Correspondence Control

The entire stereo matching process is under the control of this module. Figure 6 shows the Finite State Machine (FSM) of the stereo matching controller. The FSM of stereo correspondence control has six conditional states:

1. When the system resets, the FSM enters into an IDLE state and the initialization of all variables is done.
2. Followed by the initialization of line buffers corresponding to image pairs, the FSM enters into

3. In the *DISP* state, the system uses one clock cycle for setting and another clock cycle is used for the activating the data store module. The data store module stores the 64 SAD minimum values and computes the corresponding disparity. The disparity value is then written to DPRAM
4. After this the FSM goes for updating the pixel coordinates that involve three conditions. Let X and Y , are the current pixel coordinates that are to be updated in the *UPDATE* state, then the three conditions of the transition state are:
 - If the value of x coordinate is smaller than the total horizontal pixels, the FSM enters into the state *Feeding_Shifttap*
 - If the x coordinate value and total horizontal pixels are equal and the y coordinate value is lesser than the total vertical pixels, then the state goes to *Init_Shifttap*
 - If x and y are equal to the total horizontal and vertical pixels respectively, the state will be *DONE*

Table 1: State table

States	Comment	Condition
IDLE	IDLE state	Reset = 1
A	Initialize the memory with a line of 16 pixels corresponding to left image.	Start = 1, State = IDLE
B	Initialize the memory with a line of 16 pixels corresponding to right image.	Init-done = 1, State=WAIT-A
C	Updates the memory corresponding to left input with one pixel data.	Update = 1, State = IDLE
WAIT-A	Waits for the memory Initialization corresponding to the left input.	Start-read = 1, State = A
WAIT-B	Waits for the memory Initialization corresponding to the right input.	Start-read = 1, State=B
WAIT-C	Waits for the update of memory corresponding to the left input.	Start-read = 1, State = C
WAIT-D	Waits for the update of memory corresponding to the right input.	Start-read =1, State = D

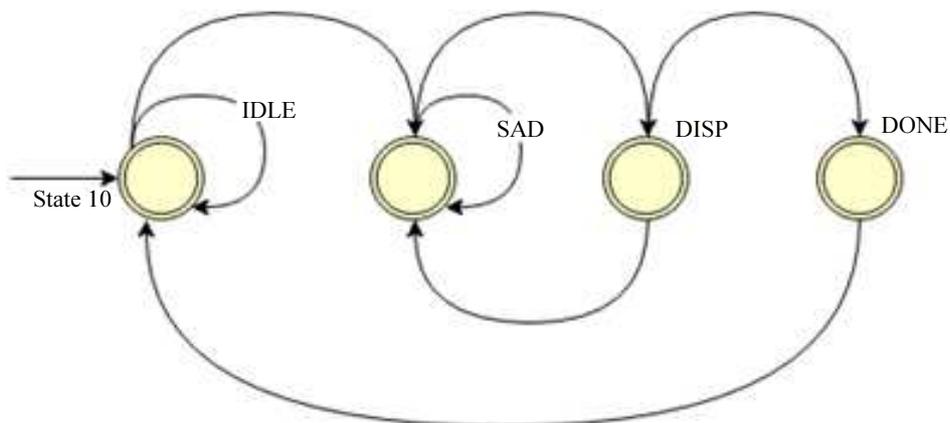


Fig. 4: FSM of Stereo matching operation

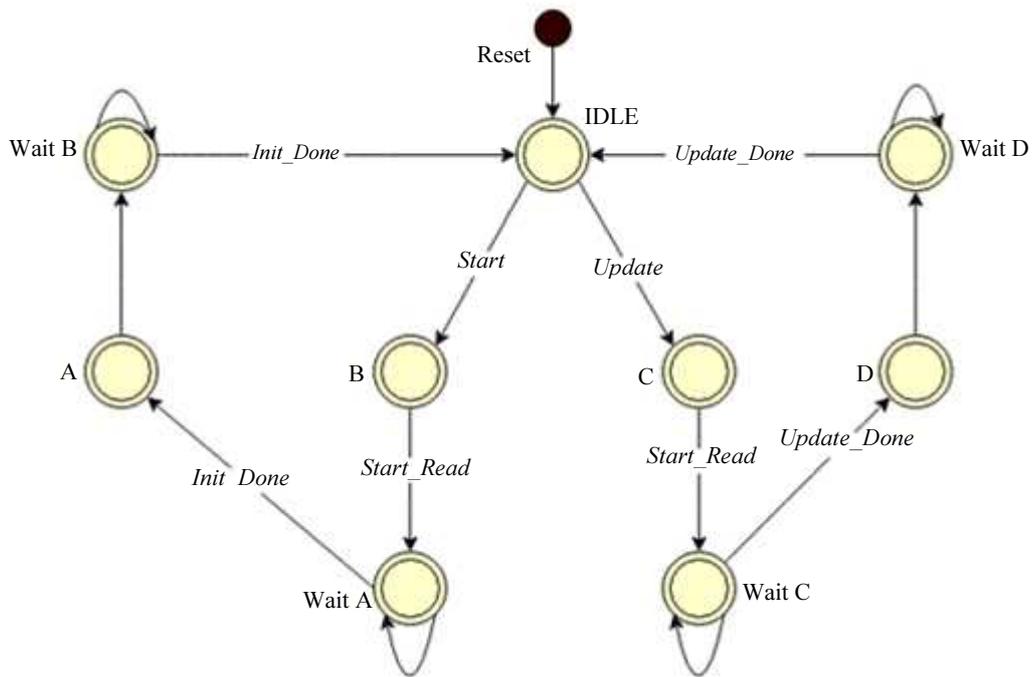


Fig. 5: FSM of the Memory Management

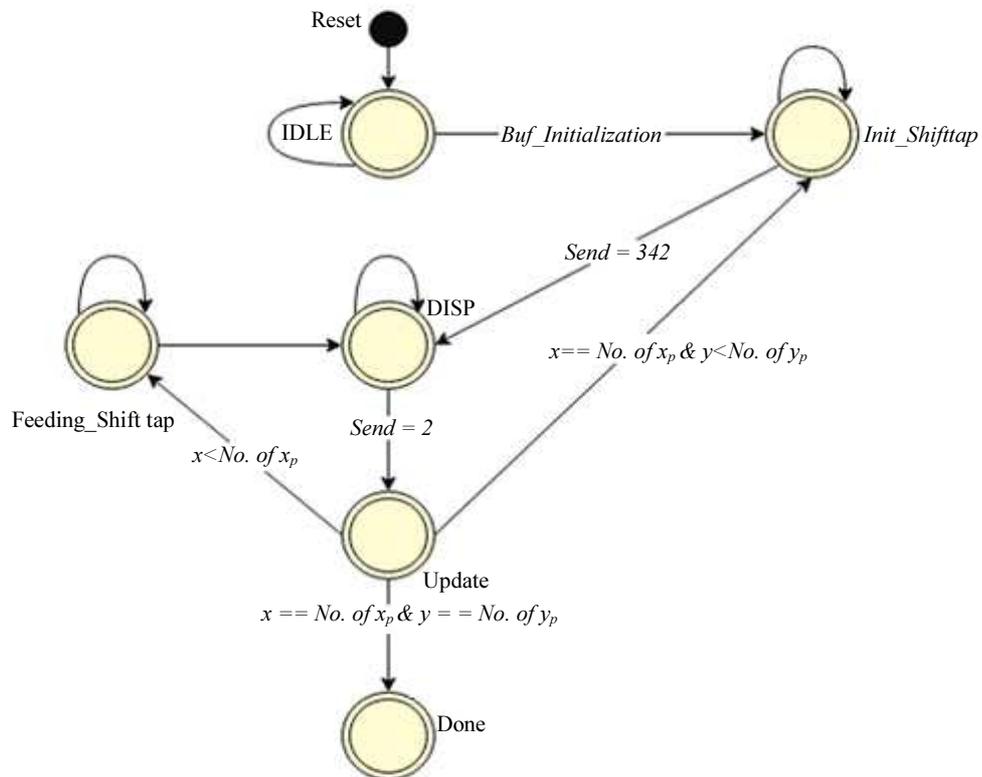


Fig. 6: FSM of stereo matching controller

Results and Analysis

The proposed SAD stereo matching system has been built on Altera Cyclone II EP2C70 FPGA board using Quartus II software. The algorithm is tested using rectified stereo images provided by Middlebury Stereo Datasets (<http://vision.middlebury.edu/stereo/data/>). All the data set consists of both left and right views and the corresponding

ground truth disparity maps Scharstein and Szeliski (2003). Window size of 5×5 has been used for the stereo matching process with a maximum disparity value of 63. Figure 7a to 7c shows the left view of input image, ground truth disparity map and the obtained disparity maps for the input images Aloe [427 \times 370], Baby [413 \times 370], Cones [450 \times 375], Reindeer [447 \times 370] and Art [463 \times 370].



Fig. 7: Inputs and disparity map results (a) input-left view (b) ground truth disparity (c) disparity map result of FPGA-SAD

Performance Evaluation

The performance evaluation has been done on the basis of RMS error and the number of bad matching pixels in the obtained disparity map Scharstein and Szeliski (2003). Root Mean Squared Error (RMSE) between the obtained disparity $d_c(i, j)$ and the ground truth disparity map $d_r(i, j)$, can be computed using the Equation 5:

$$E = \left(\frac{1}{N} \sum (i, j) | (d_c(i, j) - d_r(i, j)) |^2 \right)^{\frac{1}{2}} \tag{5}$$

N – The total pixels in the disparity map. The amount of bad matching pixels in the obtained disparity can be calculated using the Equation 6:

$$B = \left(\frac{1}{N} \sum (i, j) | (d_c(i, j) - d_r(i, j)) | > \delta_d \right) \tag{6}$$

where, δ_d is the threshold level of bad matching pixels. Here it is taken as one. The proposed stereo matching module can process inputs of size 320x240 with a speed of 50 fps. The error evaluation is done using Matlab 2014b on Intel Core i3 processor with 2.10GHz clock frequency and 4GB of RAM. The algorithm is tested and evaluated using different input image pairs taken under various illumination conditions with different camera exposure time. Figure 8-10 shows the percentage RMS error in the obtained disparity map for various image inputs taken under illumination I, II and III. A The

algorithm implementation shows minimum error of 0.1811% in the image Baby and a maximum error of 0.3083% for the image Reindeer. Figure 11-13 shows the percentage of bad matching pixels for different input images taken under various illumination condition.

The algorithm shows almost same accuracy for different input images taken under different illumination conditions with varying exposure time. The analysis shows the stability in the accuracy of results even if there are variations in illumination condition and in the camera exposure time.

Table 2 shows the performance evaluation based on bad matching pixels in the obtained disparity map for various input images. The analysis shows a maximum of 14.516% bad matching pixels for occluded and non-occluded regions together. Table 3 shows the performance evaluation based on percentage RMS Error in the obtained disparity map for various input images.

This hardware-based stereo vision system primarily targets for the use in autonomous robot navigation, which needs a system that gives precise information about the surroundings within the real-time constraints. Our stereo matching module is evaluated based on speed, RMS error and the number of bad matching pixels. To achieve the targeted performance and flexibility, this work intensively used the parallel processing architecture and optimised design. The proposed architecture is highly optimised so that only the 43% of the logic elements, 41% of total combinational functions and 18% of dedicated logic registers are used. Table 4 shows the resource utilization for the implementation.

Table 2: Percentage of bad matching pixels under different inputs under illumination conditions

Input image	Illumination I Exposure time (ms)			Illumination II Exposure time (ms)			Illumination III Exposure time (ms)		
	200	800	3200	125	500	2000	125	500	2000
	Aloe	10.5470	10432.0000	10.2250	11.5161	10.5470	12.0987	12.0487	11.5641
Cones	9.3320	9.3100	9.2250	10.5161	9.5440	9.4310	11.8163	12.5440	12.4310
Reindeer	12.6330	12.5340	12.4310	13.5160	12.5340	12.6330	14.5160	13.5340	12.4340
Baby	9.4378	9.4376	9.4456	9.4391	9.4376	9.4386	9.4410	9.4398	9.4378
Art	12.3450	12.3420	12.5145	12.5253	12.5155	12.5251	12.5358	12.5355	12.5155

Table 3: Percentage of RMS error under different inputs under illumination conditions

Input image	Illumination I Exposure time (ms)			Illumination II Exposure time (ms)			Illumination III Exposure time (ms)		
	200	800	3200	125	500	2000	125	500	2000
	Aloe	0.2190	0.2180	0.2170	0.2466	0.2191	0.2525	0.2524	0.2459
Cones	0.1980	0.1970	0.1950	0.2182	0.1960	0.1950	0.2294	0.2396	0.2495
Reindeer	0.2622	0.2520	0.2495	0.2683	0.2520	0.2622	0.3083	0.2822	0.2543
Baby	0.1825	0.1823	0.1811	0.1838	0.1823	0.1832	0.1857	0.1845	0.1825
Art	0.2331	0.2328	0.2325	0.2433	0.2335	0.2431	0.2538	0.2535	0.2335

Table 4: Resource utilisation

Quartus II 32-bit Version	
Revision Name	Stereo vision core v2
Top-level Entity Name	Stereo vision core v2
Family	Cyclone II
Device	EP2C20F484C7
Timing models	Final
Total logic elements	7,975/18,752 (43%)
Total combinational functions	7,598/18,752 (41%)
Dedicated logic registers	3,462/18,752 (18%)
Total registers	3462
Total pins	59/315 (19%)
Total virtual pins	0
Total memory bits	0/239,616 (0%)
Embedded Multiplier	9-bit elements 0/52 (0%)
Total PLLs	0/4 (0%)

Table 5: Comparison with Existing Methods

Authors	Frame rate	Size of image	Window size	Disparity Range	Algorithm	Platform
Miyajima and Maruyama (2003)	18.9 fps	640×480	7×7	27	SAD	FPGA ADM-XRC-II with one additional SSRAM board and PC
Zhang and Chen (2013)	23 fps	640×480	5×5	64	SAD	1 FPGA - Altera Cyclone II using Nios II processor
Perri <i>et al.</i> (2006)	25.6 fps	512×512	5×5	256	SAD	FPGA -VIRTEX4 XC4VLX15
Niitsuma and Maruyama (2004)	30 fps	640×480	7×7	27	SAD	FPGA
Darabiha <i>et al.</i> (2006)	30 fps	256×360	N/A	20	Local weighted phase correlation	4 FPGAs on TM-3A board
Ttofis <i>et al.</i> (2010)	30 fps	640×480	11×11	64	Segmentation-based ADSW (SAD)	Virtex-5 FPGA
Jin and Maruyama (2012)	30 fps	640×480	11×11	64	Census transformation	Virtex-4 XC4VLX200-10 FPGA
Proposed method	50 fps	320×240	5×5	64	SAD	FPGA Altera Cyclone II EP2C70 without using Nios II processor

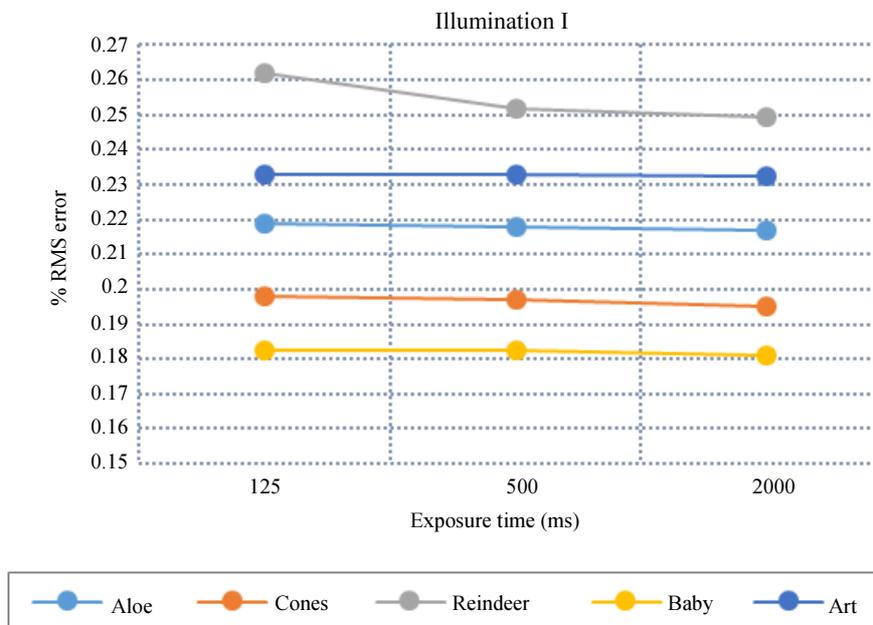


Fig. 8: RMS error evaluation using different inputs taken under Illumination I

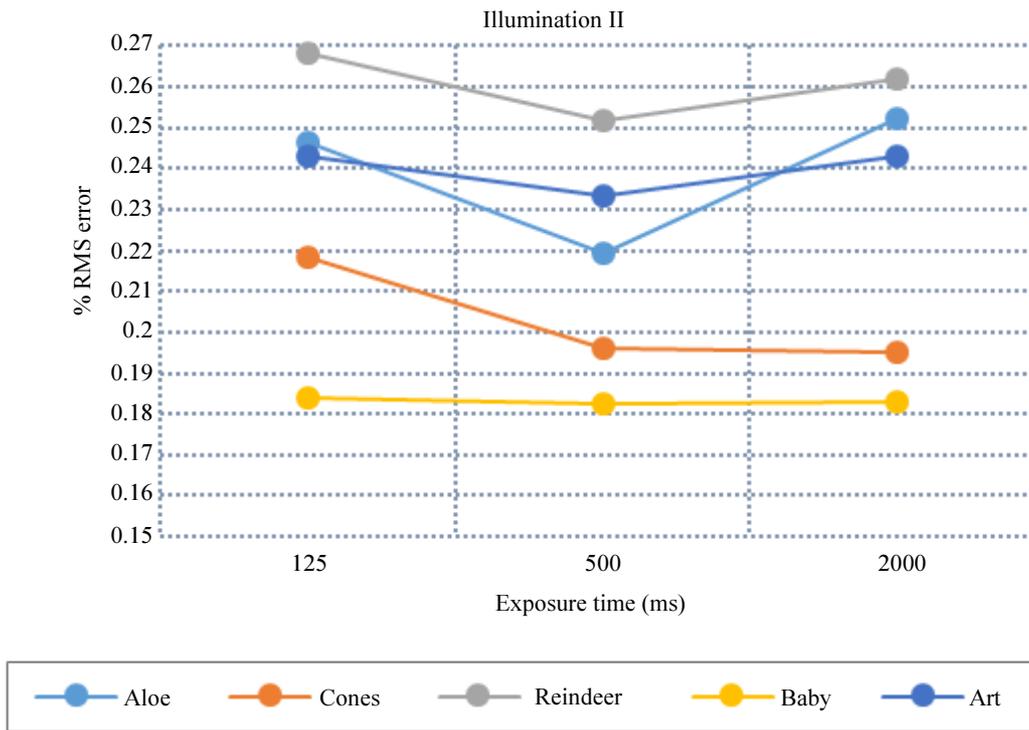


Fig. 9: RMS error evaluation using different inputs taken under Illumination II

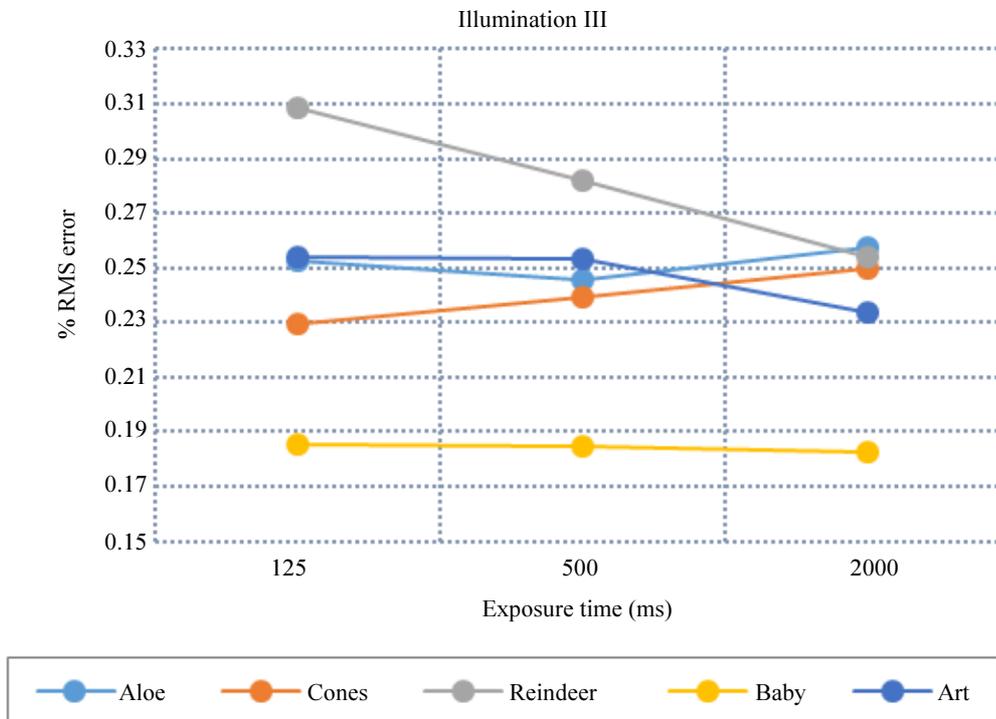


Fig. 10: RMS error evaluation using different inputs taken under Illumination III

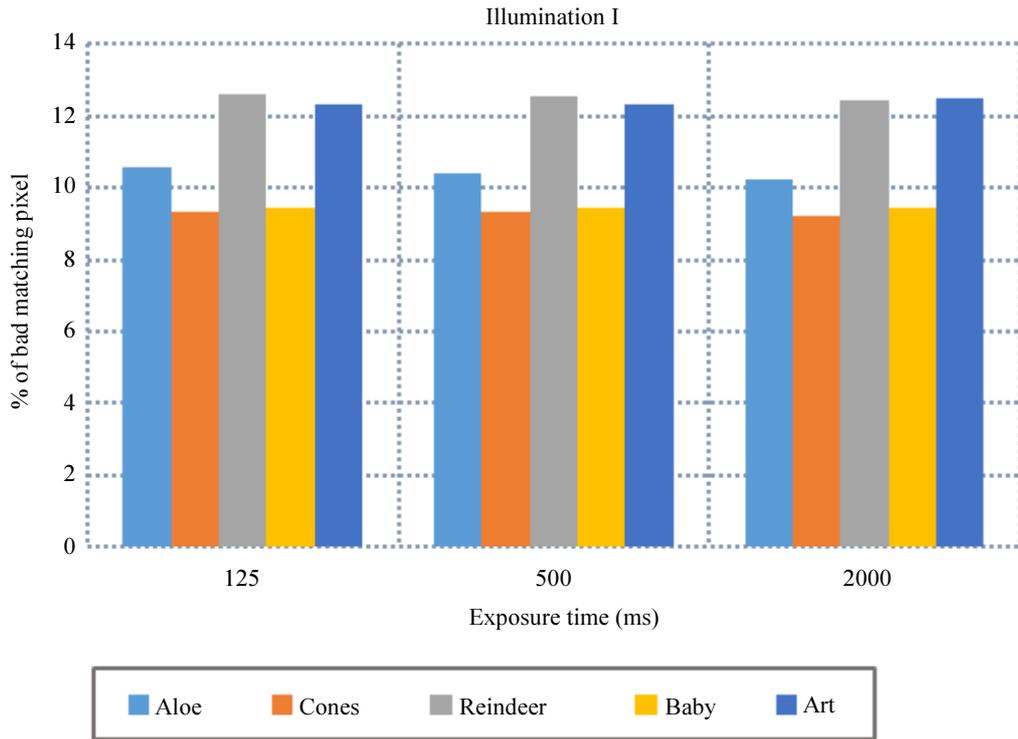


Fig. 11: Percentage of bad matching pixels using different inputs taken under Illumination I

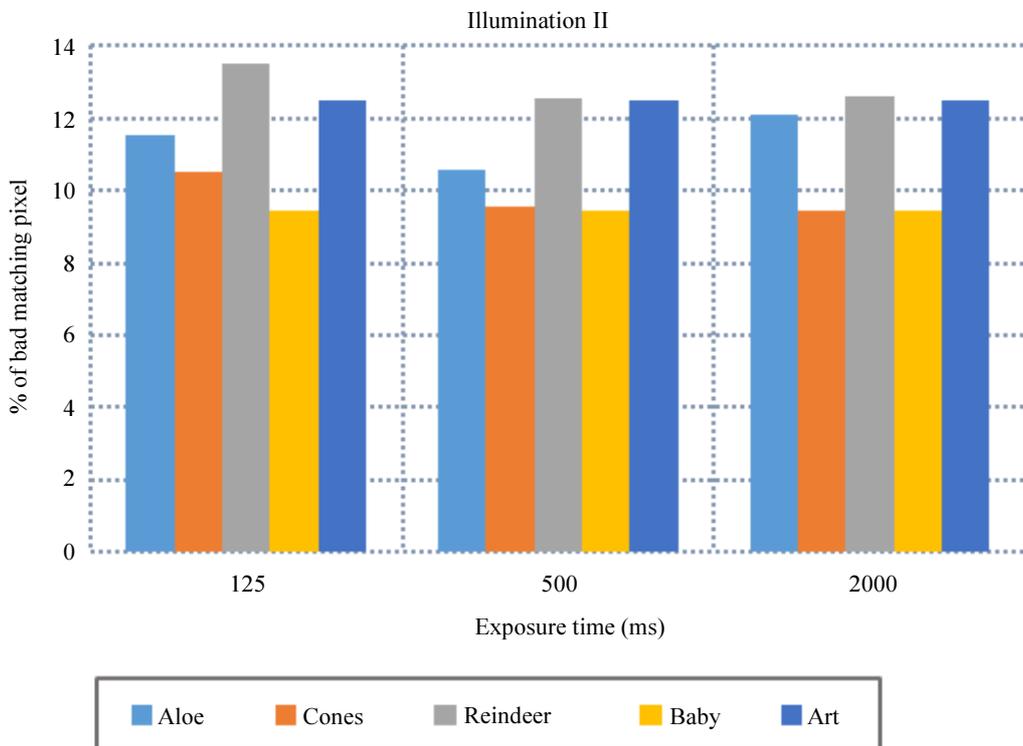


Fig. 12: Percentage of bad matching pixels using different inputs taken under Illumination II

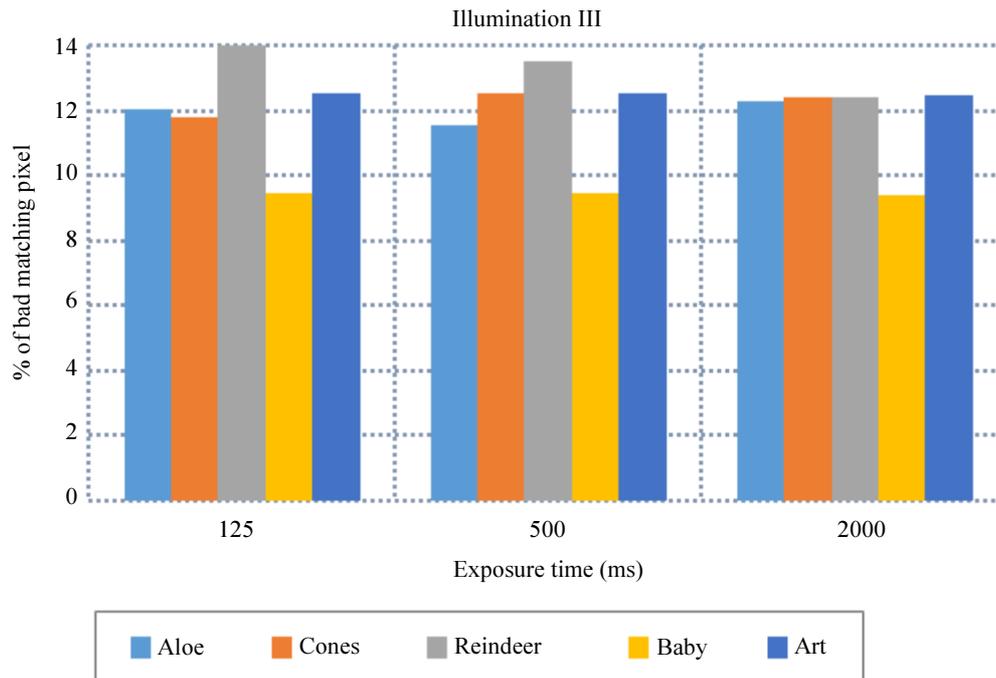


Fig. 13: Percentage of bad matching pixels using different inputs under Illumination III

Comparison with the Existing Methods

Table 5 lists the comparison of our method with the existing methods in the hardware implementation of stereo vision system based on the speed, image size, window size and the platform used for the implementation. Our method can process an image size of 320×240 and it achieved a speed of 50 frames per second with 64 disparity levels. This highly optimised parallel processing high speed architecture takes only 43% of available logic elements 41% of combinational functions and 18% of registers and is well suitable for the implementation on resource limited hardware. Miyajima and Maruyama (2003) implemented a stereo vision system on FPGA ADM-XRC-II with one additional SSRAM board and PC, This system attained a speed of 18.9 frames per second for an image size of 640×480 with a disparity range of 27. Zhang and Chen (2013) implemented stereo matching SAD block matching on Altera Cyclone II using the Nios II processor. They achieved a speed of 23 fps with a disparity range of 64. Perri *et al.* (2006) and Niitsuma and Maruyama (2004) attained similar performance using SAD algorithm. But in the above implementations, the accuracy of the disparity map obtained is not discussed. Jin *et al.* (2010) proposed a stereo matching system on high density Virtex-4 XC4VLX200-10 FPGA with a frame rate of 30fps using census transformation. Our optimised design achieved high speed with more accurate disparity result using the low density, low cost FPGA. This system can

be used to meet the demands of intelligent vision applications such as autonomous navigation of robots, SLAM and self driving vehicles.

Conclusion and Future Scope

For achieving the targeted performance and flexibility, this system uses parallel processing architecture and optimised design. The algorithm implementation shows a maximum error of 0.3083% only. Our stereo matching module can process inputs with a size of 320×240 with a speed of 50 fps and maximum disparity of 64. This work primarily aims to use this vision system in space exploration robots. For the vision guided autonomous robot navigation, the requirement of algorithms on a low power dedicated hardware that can achieve a high frame rate is unambiguous. Often the use of a personal computer is not a convenient solution for many applications such as space robots and rescue robots. In such situations the implementation our stereo vision system on low power FPGA is an apt choice. This stereo vision system can be used to find the position of objects in the path of a robot. Since the stereo vision system mimics human vision, the FPGA implementation of such a system can be used as the sensor system for the navigation of autonomous robots in unknown outdoor environments. Stereo vision is an inexpensive passive range finder and there will not be any interference if multiple robots are working in one environment.

Even though thoroughly investigated for decades by many researchers, the stereo vision still remains an active research area. The large gap between the cognitive capabilities of human visual perception and the achieved capabilities of computer vision yet to be filled. Often many of the problems in this area are ill-posed and the solutions are not unique and have not expected to reach in a state of saturation. Most of the real time applications in computer vision demand on accuracy and speed. As the role of robots has increased in many aspects of our lives, incorporation of the latest outbreaks in neuroscience findings in computer vision will help in solving the open issues in intelligent robot vision. Future scope of this work targets on the incorporation of Convolutional Neural Networks (CNN) in our system for getting anthropomorphic capabilities to deal with the occluded areas and the depth discontinuities that occurs at the edge of objects. Such an intelligent vision system will be of great use in the indoor and outdoor applications such as in rescue robots and industrial robots.

Acknowledgment

We thank Dr. S. Bhavani, Department of ECE, Karpagam University, Coimbatore for sharing her valuable knowledge that significantly improved this article on its earlier version. We would also like to show our gratitude to the Dr. S. Suresh Babu, Sreebudha College of Engineering, Kerala for sharing his sagacity with us during the period of this work even though any errors if any are of our own and should not defame these esteemed persons.

Author Contributions

Deepambika Vadakkettathu Anirudhan: Contributed in formulating the concept design, implementation, experimental analysis and in manuscript preparation.

Malangai Abdul Rahiman: Took part in the development of proposed methodology, experimental analysis and article revision.

Conflict of Interest

The authors declare no conflict of interest.

References

- Birchfield, S. and C. Tomasi, 1999. Depth discontinuities by pixel-to-pixel stereo. *Int. J. Comput. Vis.*, 35: 269-293. DOI: 10.1109/ICCV.1998.710850
- Darabiha, A.W., J.M. Lean and J. Rose, 2006. Reconfigurable hardware implementation of a phase-correlation stereo algorithm. *Mach. Vis. Applic.*, 17: 116-132. DOI: 10.1007/s00138-006-0018-2
- Ding, J., J. Liu, W. Zhou, H. Yu and Y. Wang *et al.*, 2011. Real-time stereo vision system using adaptive weight cost aggregation approach. *EURASIP J. Image Video Process.*, 2011: 20-20. DOI: 10.1186/1687-5281-2011-20
- Fusiello, A., E. Trucco and A. Verri, 2000. A compact algorithm for rectification of stereo pairs. *Mach. Vis. Applic.*, 12: 16-22. DOI: 10.1007/s001380050120
- Hadjitheophanous, S., C. Ttofis, A.S. Georghiades and T. Theocharides, 2010. Towards hardware stereoscopic 3D reconstruction a real-time FPGA computation of the disparity map. *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, Mar. 8-12, IEEE Xplore Press, Dresden, Germany*, pp: 1743-1748. DOI: 10.1109/DATE.2010.5457096
- Hamzah, R.A., R.A. Rahim and Z.M. Noh, 2010. Sum of absolute differences algorithm in stereo correspondence problem for stereo matching in computer vision application. *Proceedings of the 3rd International Conference on Computer Science and Information Technology, Jul. 9-11, IEEE Xplore Press, Chengdu, China*, pp: 652-657. DOI: 10.1109/ICCSIT.2010.5565062
- Hirschmuller, H., 2005. Accurate and efficient stereo processing by semi-global matching and mutual information. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Jun. 20-25, IEEE Xplore Press, San Diego, CA, USA*, pp: 807-814. DOI: 10.1109/CVPR.2005.56
- Hirschmuller, H., 2008. Stereo processing by semiglobal matching and mutual information. *IEEE Trans. Patt. Anal. Mach. Intell.*, 30: 328-341. DOI: 10.1109/TPAMI.2007
- Jin, M. and T. Maruyama, 2012. A real-time stereo vision system using a tree-structured dynamic programming on FPGA. *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays, Feb. 22-24, ACM, Monterey, California, USA*, pp: 21-24. DOI: 10.1145/2145694.2145698
- Jin, S., J. Cho, X.D. Pham, K.M. Lee and S.K. Park *et al.*, 2010. FPGA design and implementation of a real-time stereo vision system. *IEEE Trans. Circuits Syst. Video Technol.*, 20: 15-26. DOI: 10.1109/TCSVT.2009.2026831
- Lazaros, N., G.C. Sirakoulis and A. Gasteratos, 2008a. A dense stereo correspondence algorithm for hardware implementation with enhanced disparity selection. *Proceedings of the 5th Hellenic conference on Artificial Intelligence: Theories, Models and Applications, Oct. 02-04, Springer, Syros, Greece*, pp: 365-370. DOI: 10.1007/978-3-540-87881-0_34

- Lazaros, N., G.C. Sirakoulis and A. Gasteratos, 2008b. Review of stereo vision algorithms: from software to hardware. *Int. J. Optomechatron.*, 2: 435-462. DOI: 10.1080/15599610802438680
- Middlebury Stereo Dataset. <http://vision.middlebury.edu/stereo/data/>
- Mingxiang, L. and J. Yunde, 2006. Stereo Vision System on programmable Chip (SVSoC) for small robot navigation. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 9-15, IEEE Xplore Press, Beijing, China, pp: 1359-1365. DOI: 10.1109/IROS.2006.281923
- Miyajima, Y. and T. Maruyama, 2003. A real-time stereo vision system with FPGA. *Proceedings of the 13th International Conference on Field Programmable Logic and Application*, Sept. 1-3, Springer, Lisbon, Portugal, pp: 448-457. DOI: 10.1007/978-3-540-45234-8_44
- Niitsuma, H. and T. Maruyama, 2004. Real-time detection of moving objects. *Proceedings of the 10th International Conference on Field Programmable Logic and Applications*, (PLA' 04), Springer, Berlin, Heidelberg, pp: 1155-1157. DOI: 10.1007/978-3-540-30117-2_154
- Perri, S., D. Colonna, P. Zicari and P. Corsonello, 2006. SAD-based stereo matching circuit for FPGAs. *Proceedings of the 13th IEEE International Conference on Electronics, Circuits and Systems*, Dec. 10-13, IEEE Xplore Press, Nice, France, pp: 846-849. DOI:10.1109/ICECS.2006.379921
- Sarika, S., V.A. Deepambika and M.A. Rahman, 2015. Census filtering based stereo matching under varying radiometric conditions. *Proc. Comput. Sci.*, 58: 315-320. DOI:10.1016/j.procs.2015.08.026
- Scharstein, D. and R. Szeliski, 2003. High-accuracy stereo depth maps using structured light. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 18-20, IEEE Xplore Press, Madison, WI, USA, pp: I-195-I-202. DOI: 10.1109/CVPR.2003.1211354
- Scharstein, D., R. Szeliski and R. Zabih, 2001. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Proceedings of the IEEE Workshop on Stereo and Multi-Baseline Vision*, Dec. 9-10, IEEE Xplore Press, Kauai, HI, USA, pp: 131-140. DOI: 10.1109/SMBV.2001.988771
- Ttofis, C., S. Hadjitheophanous, A.S. Georghiadis and T. Theocharides, 2010. Towards hardware stereoscopic 3D reconstruction: A real-time FPGA computation of the disparity map. *Proceedings of the Conference on Design, Automation and Test in Europe*, (ATE' 10), pp: 1743-1748.
- Ude, A., 2010. *Interactive object learning and recognition with multiclass support vector machines*. INTECH Open Access Publisher.
- Woodfill, J. and B. von Herzen, 1997 Real-time stereo vision on the PARTS reconfigurable computer. *Proceedings of the 5th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, Apr. 16-18, IEEE Xplore Press, Napa Valley, CA, USA, pp: 201-210. DOI: 10.1109/FPGA.1997.624620
- Yoon, K.J. and I.S. Kweon, 2006. Adaptive support-weight approach for correspondence search. *IEEE Trans. Patt. Anal. Mach. Intell.* 28: 650-656. DOI: 10.1109/TPAMI.2006.70
- Zhang, X. and Z. Chen, 2013. SAD-based stereo vision machine on a System-on-Programmable-Chip (SoPC). *Sensors*, 13: 3014-3027. DOI: 10.3390/s130303014