Stereo Correspondence Algorithms for Robotic Applications Under Ideal And Non Ideal Lighting Conditions

Deepambika V A *deepambika.va@gmail.com*
Asst Professor, ECE
LBS Inst of Technology for Women
Trivandrum, Kerala, INDIA

Abdul Rahiman M *rehman_paika@yahoo.com*
Pro Vice Chancellor
Kerala Technological University
Trivandrum, Kerala, INDIA

Abstract

The use of visual information in real time applications such as in robotic pick, navigation, obstacle avoidance etc. has been widely used in many sectors for enabling them to interact with its environment. Robotics require computationally simpler and easy to implement stereo vision algorithms that will provide reliable and accurate results under real time constraint. Stereo vision is a less expensive, passive sensing technique, for inferring the three dimensional position of objects from two or more simultaneous views of a scene and there is no interference with other sensing devices if multiple robots are present in the same environment. Stereo correspondence aims at finding matching points in the stereo image pair based on Lambertian criteria to obtain disparity. The correspondence algorithm will provide high resolution disparity maps of the scene by comparing two views of the scene under the study. By using the principle of triangulation and with the help of camera parameters, depth information can be extracted from this disparity .Since the focus is on real-time application, only the local stereo correspondence algorithms are considered. A comparative study based on error and computational costs are done between two area based algorithms. Evaluation of Sum of absolute Difference algorithm, which is less computationally expensive, suitable for ideal lightening condition and a more accurate adaptive binary support window algorithm that can handle of non-ideal lighting conditions are taken for this study. To simplify the correspondence search, rectified stereo image pairs are used as inputs.

Keywords: Stereo Vision Robotic, Correspondence Algorithm, SAD.

1. INTRODUCTION

Computer vision tries to copy the way in which the human beings perceive visual information. By means of pin hole cameras acting as eyes, computers process the information in an ‘intelligent way’ as does the human brain .The three dimensional information about the environment is essential for movement and object detection in robotic applications. One method of obtaining depth information is by using binocular stereo vision, for which two pinhole cameras are sufficient. The Greek word ‘stereos’ literally means firm or rigid. Stereo vision is a passive sensing technique which provides high resolution depth map- that is objects that are seen as 3D. In stereo vision, there is no interference with other sensor devices when multiple robots are present in the same environment. Stereo vision infer depth from two 2D images called left and right images taken from different viewpoints of a scene. Figure.1 shows an overview of stereo vision system.

Stereo correspondence aims to find matching pixels (conjugate pair) of two given input images based on Lambertian surface assumption, i.e., to find out each point in left image and the corresponding best matching point in the right image. Based on accuracy and efficiency, the correspondence algorithms can be grouped two main categories-local and global[17]. Global methods are accurate but time and computational demanding due to their iterative nature. Accuracy becomes important in applications such as precise 3D surface modeling, especially

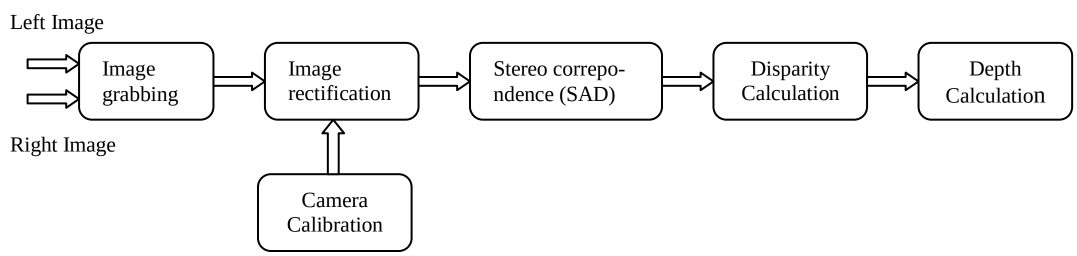


FIGURE 1: Overview of Stereo Vision System.

when dealing with object surfaces with complex reflectance behavior and poor texture. Efficiency is the main concern when the stereo system is employed in real-time applications such as robot navigation, video surveillance, and civil protection. Performance of stereo algorithms depends on the cost aggregation function used for the dissimilarity or the similarity measure. Though the feature based algorithms are faster they give only sparse disparity maps. The need of computationally less expensive stereo algorithm in combination with accuracy in real time operations are in more demanded.Also for various outdoor robotic applications, the lighting conditions are far from ideal conditions. In such conditions typical local algorithms may not give accurate results. A comparative study based on error and computational costs are done between two area based algorithms.Implementation and evaluation of Sum of absolute Difference algorithm, which is less computationally expensive ,easy to implement and suitable for ideal lightening condition and a more accurate adaptive binary support window algorithm that can handle images taken under non-ideal lighting conditions are taken for this study.

2. SUM OF ABSOLUTE DIFFERENCES (SAD) ALGORITHM

Sum of Absolute Differences (SAD) is a wide spread simple block matching algorithm. It is faster and easier for hardware implementation as it only needs a subtraction, a comparison, and a possible sign change. This area based algorithm assumes that the image window centered at a pixel have similar grey level.The correspondence problem is done by comparing blocks of pixels in each image. The dissimilarity is the sum of differences in pixel intensities, therefore the name SAD.

Computing the sum of differences between the pixel intensity values for a block around the pixel(x, y) is simple. Let f(i,j)be the intensity of the pixel at coordinates (i, j) in the reference image. For the second image ,intensity g(i, j) is used, N defines the extend of the block in either direction around the center pixel, the block has therefore width and height (2N+1). Here the rectified images are used as inputs. Therefore the 2-D stereo correspondence problem is reduced to a 1-D problem, means that the search is done along horizontal lines only as there is no vertical shift for the same pixels in the left and right images if the images are rectified[1].The variable'd' is the disparity-the displacement of corresponding matching points from one image to the other.

$$SAD(x,y,d) = \sum_{i=x-N}^{x+N} \sum_{j=y-N}^{y+N} |f_{i,j} - g_{i+d,j}| \qquad Eqn(1)$$

Each pixel in the left image is compared with every pixel on same epipolar line in right image.SAD values for all pixels are computed and the disparity is computed using winner-take-all (WTA) principle. The winner is simply the disparity associated with the smallest SAD value, in ideal case smallest SAD value will be zero.

$$disparity(x,y) = \min_{d\in D} \left(sad(x + d, y)\right) \qquad Eqn(2)$$

The Eqn (2) describes the disparity for the block around the pixel(x, y),which varies from zero to dmax, where dmax represents the highest disparity value of the stereoscopic images called disparity range. Each block from the left image is matched to a block in the right image by shifting the left block over the search area of right image [2]. Difference in index values for the SAD minimum corresponds to disparity for that area. This is continued for each block until the disparity map is completely filled and the resulting image is known as Disparity Space Image (DSI).

2.1 Performance Improvements

By applying a simple sliding window scheme, the algorithm will speed up. This works in the horizontal as well as in the vertical direction.

Horizontal sliding window:As shown in Fig.2 a window is moved from one column to next column ie., from left to the right across a horizontal scan line. The search rows are still the same, except for the pixel in the left and left and right.

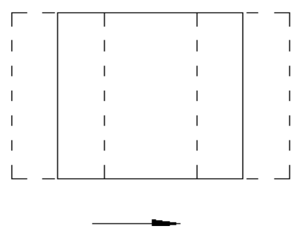


FIGURE 2: Horizontal Sliding Window: window slides across horizontal scan line.

Vertical sliding window: The sliding window scheme is also being applied when moving to the next horizontal line as shown in Fig.3 After completing a pass on one horizontal line, the window moves to the line below. The search columns are still the same, except for the pixel above and below.
The DSI obtained will be dense disparity map.

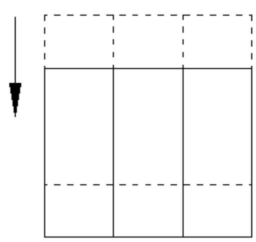


FIGURE 3: Vertical Sliding Window- window slides across vertical scan line.

2.2 Depth Computation

From the disparity values obtained, depth (Z) can be estimated by using triangulation and camera parameters [4]. The depth Z is given by Eqn.[3]

$$Z = \frac{f.b}{x_l - x_r} = \frac{f.b}{d} \qquad Eqn(3)$$

Where d is the disparity, f the focal length of the camera and b is the baseline distance between the centres of cameras.

2.3 Steps of SAD Algorithm

- Read the rectified input images.
- Select a square window from left image
- Slide this window over right image and compute SAD minimum
- Obtain the difference in index values corresponding to the SAD minimum
- Aggregate this index values (disparity) for the entire image
- Obtain depth value using triangulation.

3. ADAPTIVE BINARY SUPPORT WINDOW APPROACH

For real-time applications local and window based algorithms are suitable. They will produce decent depth maps. The size of the matching window strongly influences the performance of area-based algorithms. For high textured regions and object boundaries, small window size is good, but it fails in low textured image regions. Similarly, a large window size gives good results at low textured regions but it blurs the disparity edges. Therefore the selection of the ideal window size is important. The window size should be large enough to contain distinguished features, but small enough to keep depth discontinuities. When the local structures of image pixels are similar, it will be difficult to find their correspondences in other images without global matching. But global correspondence methods are computationally expensive and are not suitable for real-time applications. When fixed window size is used for stereo correspondence it is implicitly assumed that all the pixels with in this window are of same depth. But this is not true for object edges where depth discontinuity occurs.

Window-based stereo correspondence algorithms often exhibit a ‘foreground fattening effect’ near depth discontinuities between two objects. When this happens, samples from the far object are mistakenly measured as having the same disparity as samples on the near object. By using the local support windows, the depth ambiguity can be reduced efficiently and the discriminative power of the similarity measure is increased.

Kanade and Okutomi [5] proposed an iterative stereo matching algorithm which selects window adaptively to minimise the uncertainty in the disparity estimation. This method starts with an initial estimation of the disparity map and updates it iteratively. Hence the method is sensitive to the initial disparity estimation. Also the shape of support window is limited to a rectangle. Though the window can be expanded in chosen directions it is not suitable for pixels near arbitrarily shaped objects at depth discontinuities. This method is computationally expensive and not suitable for real time applications.

Fusiello et al. [6] presented normalised SSD (Sum of Squared Differences) algorithm with multiple window scheme using left-right consistency to compute disparity. The correlation is done with nine different windows and the disparity profile selects proper window. But this limited number of windows is not enough to cover entire range of windows of required size and shape.

Veksler [7] optimized window size by using minimum ratio cycle algorithm. This approach is the first window based algorithm using non rectangular window. Even though this correspondence algorithm performs well, it uses many user specified parameters for cost computation and is not suitable for real time systems.

Correspondence methods [8], [9] used appropriate support-weights to the pixels in a support window. But the shape and size of a local support window are fixed. Also they do not have general criteria for choosing the shape of the support window for these approaches.

The adaptive-weight algorithm proposed by Yoon [10] computes the support weight of each pixel inside a fixed sized square window. The support weights of pixels in a window are based on color similarity and geometric proximity to reference pixel. Their experiment indicates that a local based stereo matching algorithm can produce good depth maps similar to global algorithms. Though this method gives accurate results at depth discontinuities and in homogeneous regions, it is computationally expensive and is also prone to image noise.

Tombari [11] extends the adaptive-weight algorithm of Yoon's concept of colour proximity as well as information from segmentation .Though it is an accurate local stereo correspondence the computational complexity remains the same.

Leonardo De-Maeztu[12] presents a stereo correspondence using gradient similarity and locally adaptive support-weight. Here the matching measure relies on the gradient fields of neighbourhood of a pixel than the intensity of the pixel. Even though this approach improves on the accuracy of previous adaptive support-weight algorithms it needs extra computational stages by the addition of the gradient computation.

Lazaros Nalpantidis[13] proposes an algorithm for robotic application incorporates biologically and psychologically inspired features to an adaptive weighted sum of absolute differences (SAD) for stereo matching to determine the accurate depth of a scene. Though the algorithm exhibits good behaviour, its needs more computational time and it uses two user defined parameters.

Lazaros Nalpantidis & Antonios Gasteratos [14] proposes a luminosity-compensated dissimilarity measure (LCDM). Though this method process contrast differentiations it does not exhibit the same behaviour for luminosity differentiations. This stereo algorithm which is based on the proposed LCDM measure is not fast enough to achieve the real- or near real-time frame rates demanded by robotic applications.

This work presents simple stereo correspondence algorithm using binary adaptable window as support window which will meet the real time requirements. For each fixed size window a binary mask window is generated that can take any shape within the fixed window. The mask window selects the supporting pixels in the cost aggregation phase of the SAD algorithm. This selection is performed using color similarity and spatial distance metrics. The proposed method is composed of three parts: adaptive support window computation, dissimilarity computation based on the support-window and disparity selection.

3.1 Support Weight Aggregation In The Human Visual System

When aggregating support to measure the similarity between image pixels, the support from a neighboring pixel is valid only when the neighboring pixel is from the same depth. Visual grouping is very important to form a support window. There are many visual cues used for perceptual grouping. Among them, similarity and proximity are the two main grouping concepts in classic gestalt theory. The gestalt rule of organization based on similarity (or smoothness) and proximity is one of the most important ones and has been widely used in vision research. Gestalt is a German word which can be translated as 'form, shape, or pattern' in English. The similarity principle states that people tries to group similar elements together based on features such as colour or shape. The principle of proximity states that elements that are located close to each other will be grouped together [15]. The difference between pixel colors is measured in the CIE Lab color space because it provides three-dimensional representation for the perception of color stimuli.

3.2 CIELAB

The most complete color space specified by the International Commission on Illumination is CIE L*a*b* (CIELAB). It describes all the visible colors of the human eye and was created as a device-independent model for reference. The three coordinates of CIELAB represent the lightness of the colour (L* = 0 indicates black and L* = 100 indicates white), its position between red/magenta and green .The negative values a* indicate green while positive values indicate magenta. Its position is between yellow and blue. Negative values, b* indicate blue and positive values indicate yellow. The asterisk (*) after L, a and b are part of the full name, since they represent L*, a* and b*, to distinguish them from Hunter's L, a, and b. The L*a*b* model is a three-dimensional model, and hence it can be represented properly in a three-dimensional space only.

The nonlinear relations for L*, a*, and b* are intended to imitate the nonlinear response of the eye. The uniform changes of components in the L*a*b* color space aim to correspond to uniform changes in perceived color. The relative perceptual differences between any two colors in L*a*b* can be approximated by treating each color as a point in a three-dimensional space (with three components: L*, a*, b*) and taking the Euclidean distance between them.

3.3 Adaptive Binary Support Window

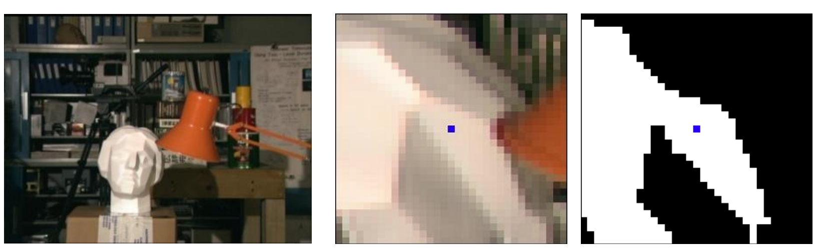
To construct the support window, with in a fixed size window we took the color similarity of the pixels in a small region of the reference image. The pixels having small Euclidean distance are grouped together to form the windows of different sizes and shapes.
The distance function can be expressed as:

$$d_{pq} = \sqrt{(Lp - Lq)^2 + \left(a_p - a_q\right)^2 + \left(b_p - b_q\right)^2} \qquad \text{Eqn}(3)$$

The values of L,a,b are calculated from RGB values in the CIE Lab color space. Matching window for a pixel is obtained by placing the centre of support window on that pixel and the distance d_{pq} between the center pixel and other pixels in support window is computed. Pixels having smaller distance, which is limited to a threshold 'T' are taken for the matching process. The binary mask window is generated as follows. The support weights are chosen binary, where '0' means that this pixel will give no support to the matching window and '1'means that this pixel gives support to the matching window. There is no gradient between these two extremes.
The intensity of binary mask,

$$W = \begin{cases} 1 & \text{for } d_{pq} < T \\ 0 & \text{otherwise} \end{cases} \qquad Eqn(4)$$

where T is a threshold value. This is taken as a masking window that selects the pixels within the fixed window that will be used in the cost aggregation phase of the SAD algorithm. Only the pixel positions where the masking window is '1' (white) will be taken into account. These are the active matching regions of support window. Figure 4(b) shows such binary mask window that has been computed for a pixel of the Tsukuba image shown in Fig. 4(a). The center of the window has been highlighted using the blue color. Thus we can see that, this method constructs adaptive windows of different shapes and sizes for each pixel based on their local information.



(a) (b) (c)

FIGURE 4: An example of adaptive binary support window computation and matching. (a) Tsukuba [16]. (b) Shows the support region of the centre pixel in left image for which the adaptive binary window is to be computed. (c) Shows an adaptive binary window calculated for the centre pixel of (b).

The cost computation for dissimilarity measure and disparity aggregation are same as that of SAD algorithm as described above.

3.4 Steps of Adaptive binary Support window algorithm.

1. Read the input images.
2. Convert the colour space in RGB to CIE Lab.
3. Compute Euclidian distance from this CIE Lab.
4. Limit the Euclidian distance to a threshold value
5. Compute binary mask window based on threshold value.
6. Select a square window from left image.
7. Multiply the two windows.
8. Slide this window over right image and compute SAD minimum.
9. Obtain the difference in index values corresponding to the minimum value.
10. Aggregate this index values (disparity) for the entire image
11. Obtain depth value using triangulation.

4. RESULTS AND ANALYSIS

The two algorithms are verified using the rectified stereo image pairs from Middlebury stereo data set [16]. Each dataset of the database consists of a pair of left and right stereo images and the corresponding ground truth disparity map. The experiments are done on Matlab Platform.

4.1 Results of SAD Algorithm

For the study and evaluation of our SAD algorithms Aloe, Baby and Bowl image pairs and their ground truth data are used.Fig.5 (a) shows left view of input images ,Fig.5(b)its ground truth disparity. Figure 5(c) & (d) shows the disparity map and the corresponding depth maps obtained using fixed window size of 9x9 pixels in SAD algorithm. Here the disparity values of SAD algorithm are mapped as disparity map and the corresponding depth values are mapped as depth map.

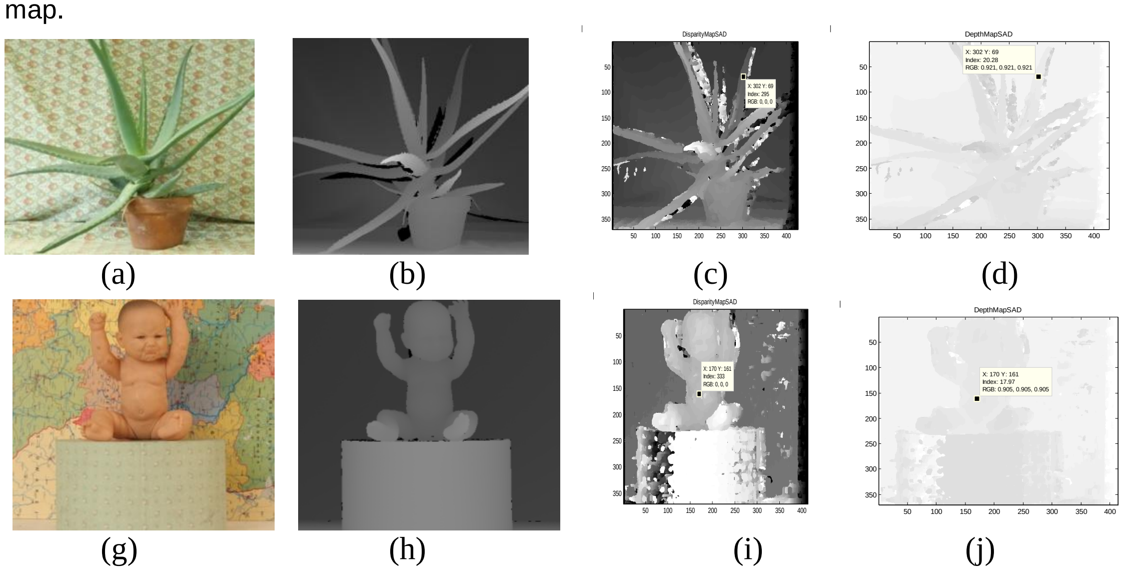


FIGURE 5: Dense disparity map &depth maps for Middlebury images and their corresponding Ground truth disparity: (a)Input image :Aloe-Left view [16] (g)Input image :Baby-Left view[16] (b) &(h) Ground truth disparities[16]. (c)& (i) Disparity map from SAD (d)&(j) corresponding depth map obtained .

4.2 Results of Adaptive Binary Support Window Algorithm

The Figure 6 (a) shows input images Tsukuba(384 x 288 image with disparity range from 0 to 15) and sawtooth (434 x 380 with disparity range from 0 to 19), These are PPM (Portable Pixel Map) images taken under bad illumination condition-texture less and scaled to a factor of 8.Fig.6 (b)&(f) are the Disparity map obtained from this algorithm. Fig.6 (c)&(g) are their corresponding depth map. The performance of this algorithm is much better since it can preserve arbitrarily shaped depth discontinuities and gives accurate results in homogeneous regions.

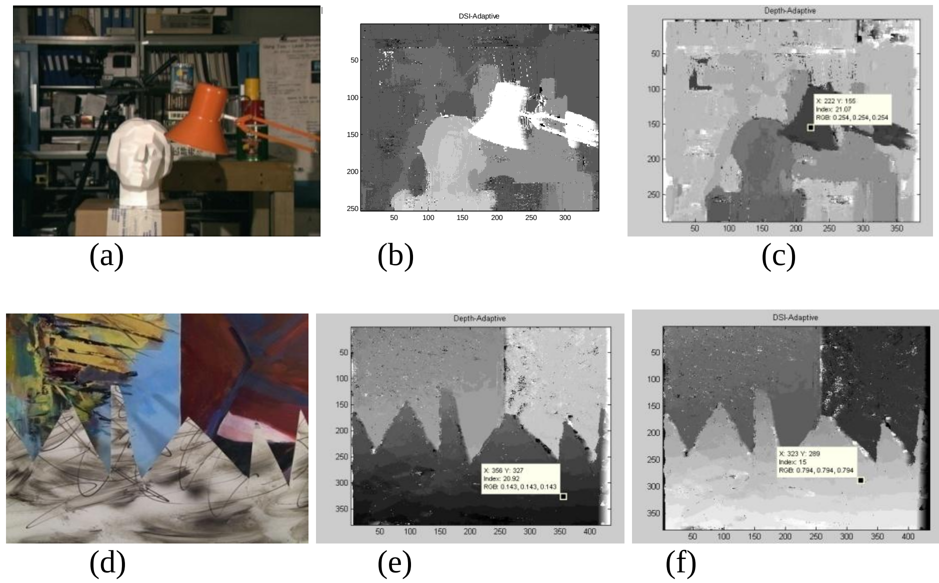


FIGURE 6: Results of Adaptive binary Support window algorithm. (a) & (d) input images- Tsukuba & Sawtooth, [16] (b)&(e)disparity map obtained and (c)&(f) Corresponding depth map.

4.3 Evaluation of Results

To evaluate the performance of a stereo algorithm we need a quantitative way to estimate the quality of the computed correspondences [17]. The error statistics is computed with respect to ground truth data.

1. RMS (root-mean-squared) error (measured in disparity units) between the computed depth map $d_C(x, y)$ and the ground truth map $d_T(x, y)$, i.e,

$$E = \left(\frac{1}{N} \sum_{(x,y)} |d_C(x,y) - d_T(x,y)|^2 \right)^{\frac{1}{2}} \qquad Eqn(5)$$

where N is the total number of pixels.

4.4 Evaluation of SAD Fixed Window

The algorithm shows an error of 1.4% for image Baby with window size 13x13. Fig. [7] shows Window size vs % Error for PNG images, Aloe, Baby and Bowl.

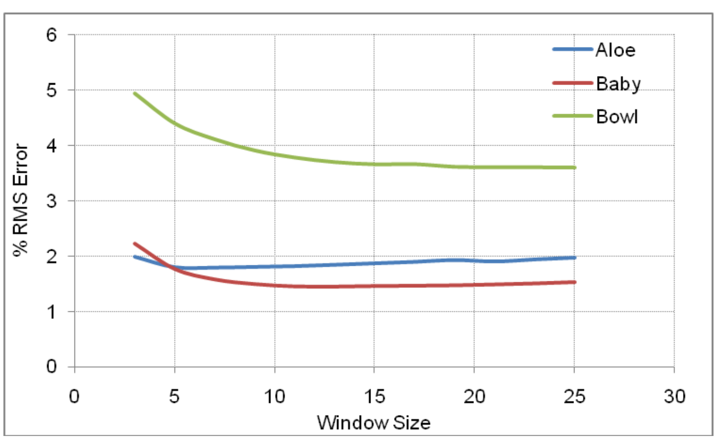


FIGURE 7: Evaluation of SAD Algorithm: Window size Vs %RMS Error for PNG images.

As the window size increases further, error increases due to edge fattening effect at depth discontinuities. For this case we go to adaptive binary approach.

4.5 Evaluation of Adaptive Binary Support Window Algorithm

The Adaptive binary support window algorithm is evaluated using the PPM images Tsukuba, Venus,and Sawtooth. Figure 8 shows the graphical representation of Window size vs % Error of our approach.

This algorithm shows an error of 0.7382% only for Tsukuba image (taken under extreme bad illumination condition) for window size 23x23.For PNG image inputs algorithm shows an error of 1.29% for Image Baby with window size 9x9.

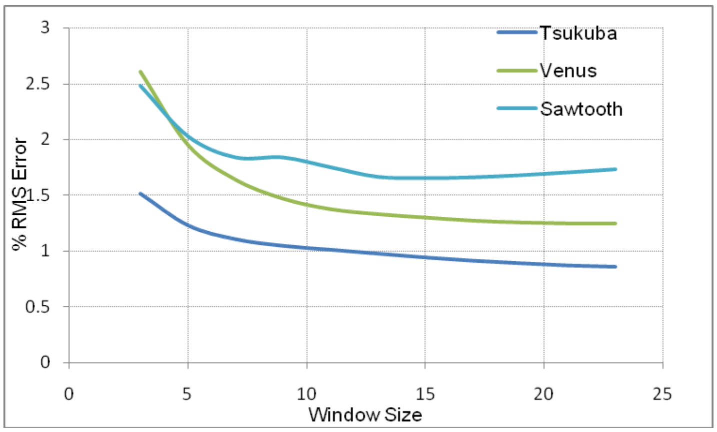


FIGURE 8: Adaptive binary support window algorithm Error evaluation. Window size vs %RMS Error for PPM images.

5. COMPARISON BETWEEN SADFIXED WINDOW AND ADAPTIVE BINARY SUPPORT WINDOW ALGORITHM

The implemented stereo algorithms focused on its real time application aspect, means that the algorithms are aimed in accuracy and computational simplicity. These two algorithms are simple and can be easily realized, if we go for a hardware implementation.

5.1 Observations in SAD Fixed Window

The algorithm is evaluated using PNG (taken under good illumination condition- good texture) and PPM (taken under bad illumination-texture less) images. Fig [9] shows the graphical representation of Window size vs % Error for PNG and PPM images. The algorithm shows an error of 1.44% for image Baby with window size 13x13.But with input image Tsukuba, error become12.7%,since this image is taken under extreme bad condition .

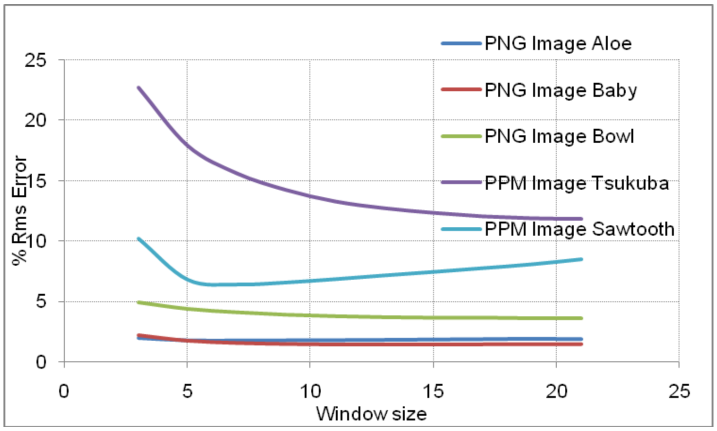
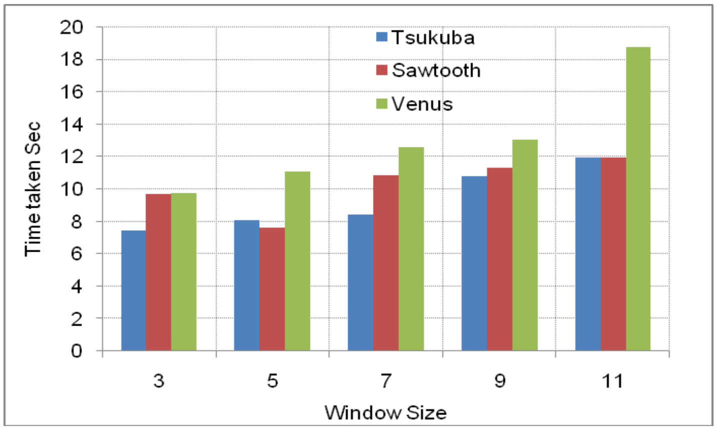
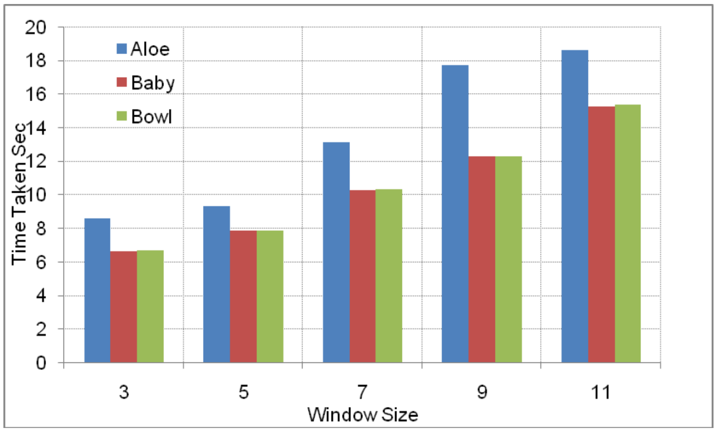


FIGURE 9: Evaluation of SAD Algorithm: Window size vs %RMS Error for PNG and PPM images.

The algorithm gives fast results, it takes around 7.4 sec for Tsukuba and 6.62 sec for Baby with smaller window sizes. The algorithm shows more error for texture less images compared to images with good texture which means that this algorithm is suitable for images having good texture. In general the algorithm gives a better performance –faster, shows less error and less computational cost for images taken under good illumination condition



(a)



(b)

FIGURE 10: Evaluation of SAD Algorithm: Window size vs. Time Taken for (a) PPM&(b) PNG images.

5.2 Observations in Adaptive Binary Support Window Algorithm

The adaptive algorithm avoids edge fattening and shows an error of 0.7382% for Tsukuba image, with window size of 23x23. If we further increase the window size this algorithm will work with a much reduced error. But with PNG images the algorithm shows more error.

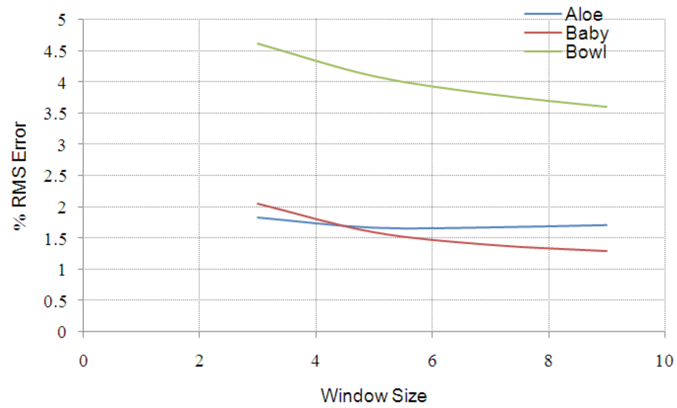


FIGURE 11: Adaptive binary support window algorithm error evaluation. Window size vs %RMS Error PNG images.

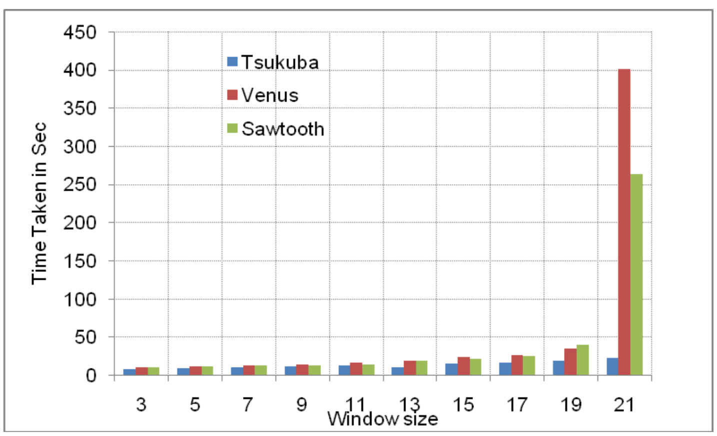


FIGURE 12: Evaluation of Adaptive binary support window Algorithm: Window size vs. Time Taken for PPM images.

Time taken for Adaptive binary support window Algorithm is 22.7 seconds for Tsukuba image with 21x21 window. The computational cost is slightly more compared to fixed window. This algorithm gives more accurate result for textureless images and make it good candidate for real-time application under bad illumination environments.

6. CONCLUSION AND FUTURE WORK

The implemented algorithms can be used for generating accurate results in different environments, suitable tor autonomous robots. The comparative study of the implemented algorithms infer that the SAD fixed window generates finer details of object's depth with moderate speed, for input images with good texture ie ,this algorithm is suitable for ideal lightning conditions .The adaptive binary window algorithm shows good accuracy for images taken under extreme bad illumination condition.

Even though this work meets its initial objective, it points towards various interdisciplinary research directions. The extension of work looks forward to an algorithm with calibration and rectification of images, which can generate more accurate results for textured and texture less images in very less time, based on FPGA implementation. From the knowledge obtained through this work more efficient stereo algorithms for various radiometric conditions, suitable to real-time applications can be developed .The above developed algorithms can be used as basis for achieving the autonomous capabilities of robotic assistance in various areas such as defense ,civil protection and interplanetary applications which makes them really interesting applied research field.

7. REFERENCES

[1] Fusiello, A., Trucco, E., & Verri, A. (2000). A compact algorithm for rectification of stereo pairs. Machine Vision and Applications, 12(1), 16-22.

[2] Hamzah, R. A., Rahim, R. A., & Noh, Z. M. (2010, July). Sum of Absolute Differences algorithm in stereo correspondence problem for stereo matching in computer vision application. In Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on (Vol. 1, pp. 652-657). IEEE.

[3] Konolige, K. (1999). Stereo geometry. SRI International.

[4] Tsai, R. Y. (1986). An efficient and accurate camera calibration technique for 3D machine vision. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 1986.

[5] Kanade, T., & 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.

[6] Fusiello, A., Roberto, V., & Trucco, E. (1997, June). Efficient stereo with multiple windowing. In 2013 IEEE Conference on Computer Vision and Pattern Recognition (pp. 858-858). IEEE Computer Society.

[7] Veksler, O. (2002). Stereo correspondence with compact windows via minimum ratio cycle. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 24(12), 1654-1660.

[8] Prazdny, K. (1985). Detection of binocular disparities. Biological Cybernetics, 52(2), 93-99.

[9] Xu, Y., Wang, D., Feng, T., & Shum, H. Y. (2002). Stereo computation using radial adaptive windows. In Pattern Recognition, 2002. Proceedings. 16th International Conference on (Vol. 3, pp. 595-598). IEEE.

[10] Yoon, K. J., & Kweon, I. S. (2006). Adaptive support-weight approach for correspondence search. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(4), 650-656.

[11] Tombari, F., Mattoccia, S., & Di Stefano, L. (2007). Segmentation-based adaptive support for accurate stereo correspondence. In Advances in Image and Video Technology (pp. 427-438). Springer Berlin Heidelberg.

[12] De-Maeztu, L., Villanueva, A., & Cabeza, R. (2011). Stereo matching using gradient similarity and locally adaptive support-weight. Pattern Recognition Letters, 32(13), 1643-1651.

[13] Nalpantidis, L., & Gasteratos, A. Biologically and psychophysically inspired adaptive support weights algorithm for stereo correspondence. Robotics and Autonomous Systems, 58(5), 457-464.

[14] Nalpantidis, L., & Gasteratos, A. (2010). Stereo vision for robotic applications in the presence of non-ideal lighting conditions. Image and Vision Computing, 28(6), 940-951.

[15] Wertheimer, M. (1938). Laws of organization in perceptual forms.

[16] http://vision.middlebury.edu/stereo/data/

[17] Scharstein, D., & Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International journal of computer vision, 47(1-3), 7-42.

[18] Wang, L., Gong, M., Gong, M., & Yang, R. (2006, June). How far can we go with local optimization in real-time stereo matching. In 3D Data Processing, Visualization, and Transmission, Third International Symposium on (pp. 129-136). IEEE.