

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/331960044>

"Stereo Vision for 3D Measurement in Robot Systems"

Article · March 2019

CITATION

1

READS

173

6 authors, including:



Dhafer Zaghar

Al-Mustansiriya University

30 PUBLICATIONS 22 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



invariant gait recognition [View project](#)



Neural network [View project](#)

"Stereo Vision for 3D Measurement in Robot Systems"

"الرؤية المجسمة لمقياس ثلاثي الأبعاد في أنظمة الإنسان الآلي"

Muhyi M. AL-Azawi,
Asst. Prof.,
Manchester University.

Dhafer R. AL-Rawy,
Asst. Prof.,
Baghdad University.

Saba A. AL-Jobory,
MA. Student,
AL-Mustansiriya University.

Abstract

The major obstacle in the application of stereo vision to extract 3D information is the error in disparity map and highly computational cost at conventional computers. This paper is concerned with finding software solution to obtain tradeoff between disparity map accuracy and reductions in execution time through developing the stereo vision algorithm. This algorithm is based on block matching technique, in which an image is partitioning into blocks. An analysis of the essential parameter of this technique (the size of block) is performed to obtain the optimal solution. Adaptive block size with vertical edge detection has been adopted to implement the proposed algorithm. The significance of this work is reduction of the execution time of the stereo vision algorithm, where the execution time of the proposed algorithm is about (2%) of the required time to execute standard vision algorithm, when running at conventional computers.

Key words: Stereo vision, disparity mapping, block matching, 3D vision, adaptive block.

الكلمات المفتاحية: الرؤية المجسمة, خارطة التباين, تطابق المقاطع, رؤية ثلاثية الأبعاد, مقطع تكيفي.

المستخلص

ان العقبة الرئيسية في تطبيق الرؤية المجسمة (Stereo Vision) لإستخلاص البعد الثالث هي الخطأ في مخطط التباين والتكلفة الحسابية العالية في الحواسيب التقليدية, لذا يتناول هذا البحث إيجاد حلول بَرمجية للحصول على موازنة بين دقة مخطط التباين (Disparity Map) وتقليل وقت تنفيذ الخوارزمية من خلال تطوير خوارزمية الرؤية المجسمة. تستند هذه الخوارزمية على تقنية تطابق اجزاء الصور, حيث يتم فيها تجزئة الصور الى مقاطع (Blocks). وأجري تحليل للمعيار الأساسي لهذه التقنية (والذي يمثل حجم المقاطع block size) للحصول على نتائج مثالية.

تم إستخدام حجم مقطع تكيفي (مُنغَير) (Adaptive block size) مع خاصية إظهار الحافات العمودية في الصور (Sobel Filter) لتنفيذ الخوارزمية. تتمثل اهمية هذا العمل في تقليل وقت تنفيذ خوارزمية الرؤية المجسمة, حيث استغرق وقت تنفيذ الخوارزمية المقترحة حوالي (2%) من الوقت اللازم لتنفيذ الخوارزمية القياسية عند تنفيذها في الحواسيب التقليدية.

1. Introduction

For safe operation in uncontrolled environments of robots, dependable 3D perception modules are required. Commonly used 3D sensors for mobile robots are laser range finder and time-of-flight cameras. Although these types have advantage of delivering accurate 3D data, but suffer from low resolution, and cannot provide enough information about the objects [1]. A promising alternative for robot navigation and mapping is vision sensors/cameras.

Which are low-cost, light and compact, easily available, have low power consumption, and provide rich information about the environment ^[2]. The stereoscopic camera (*Binocular System*) can provide the third dimension by grabbing left and right images simultaneously, like as human's binocular vision ^[3]. In this paper *binocular vision* hardware module has been implemented using two cameras to *imitate human vision*. The proposed vision algorithm has been implemented and tested with real stereo images to illustrate its performance, and a comparison with classical block matching algorithm will be introduced.

2. Stereo Vision Principle

In stereo vision, the same seen is captured using two cameras, displaced horizontally from one another, to obtain two different views on a scene from two different angles. The two captured images have a lot of similarities and smaller number of differences. In human sensitivity, the brain combines the two captured images together to get a 3D model for the seen objects. In machine vision, the 3D model for the captured objects is obtained by finding the similarities between the stereo images and using projective geometry to process these matches. The difficulties of reconstruction using stereo is finding matching correspondences between the stereo pair.

The computer compares the images while shifting the two images together over top of each other to find the parts that match. The amount of the *shift* between two matched pixels is called “*disparity*”, which relates to the object distance. The set of displacements between matched pixels is usually indicated as “*disparity map*” ^[4]. The higher disparity of object pixel means that the object is closer to the cameras and appears *brighter* in disparity map, while the less disparity means the object is far from the cameras appears *darker*. In addition, if the object is very far away, the disparity is zero that’s means the object on the left images is the same pixel location on the right image ^[3]. Figure (1) depicts the geometrical basis for stereoscopic images by using two identical cameras.

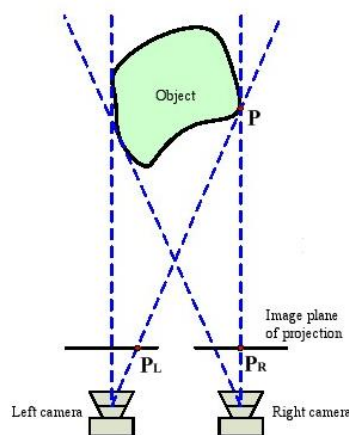


Figure (1): The position of two cameras and their image Planes of projection.

These cameras are set up on the same plane and turned in the same direction. The image planes are presented in front of the cameras for ease to model the projection (like as the model of pinhole camera) ^[3].

2.1. Epipolar Geometry

Epipolar geometry is a simplest stereo camera geometry that has optical axes that are *parallel* and *normal* to the *baseline*, the line connecting the lens centers of the cameras. This geometry depicted in figure (2). This model shows two different perspective view of an object point (P) from two identical cameras center (O_L & O_R), which separate only in x direction by a *baseline* distance (b). The points P_L and P_R in the image plane are the perspective projections of (P) in left and right view, which are called a *conjugate pair*. The plane passing through the camera centers and the object point (p) in the scene is called the *epipolar plane*. The intersection of the epipolar plane with the image plane is called *epipolar line*. By referring the epipolar geometry, correspondences at points P_L and P_R must lie on the epipolar line. In this case, corresponding epipolar lines are horizontal and have the same y -coordinate. This implies that a one-dimensional search is sufficient to find the correspondences.

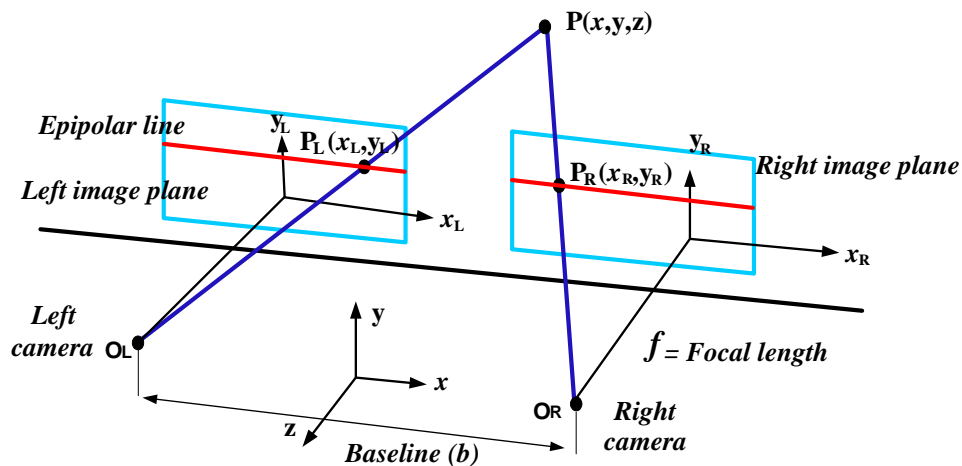


Figure (2): The epipolar geometry of stereoscopic vision.

The third dimension or the depth (Z) can be determined by comparing the triangles which are similar:

$$\frac{x_L}{f} = \frac{x + \frac{b}{2}}{z} \quad (1)$$

$$\frac{x_R}{f} = \frac{x - \frac{b}{2}}{z} \quad (2)$$

$$z = \frac{b \cdot f}{x_L - x_R} = \frac{b \cdot f}{d} \quad (3)$$

Where: the disparity is $d = x_L - x_R$, and f is the *focal length* that represents the distance between the *image plane* and *camera focal plane*.

Measurement of depth from equation (3) requires knowledge about the coordinates of corresponding points in the images ^[5].

2.2. Block Matching Correspondence

Extracting depth information from two or more cameras requires solving the correspondence problem. A block matching based on local (area-based) method is used, since it produced a dense disparity maps and can be implemented effectively. Indeed analyzing only pixels is not sufficient, because there may be several candidates that have the same gray value. Block matching method uses the intensity values of pixels within a neighborhood called block or (window) to find matching pixels in two images. For example, this happens with blocks of sizes $[5 \times 5]$ or $[7 \times 7]$. The blocks are usually defined on epipolar line for matching ease. Each block from the left (reference) image is matched to a block over the searching area of pixels in right image as shown in figure (3). The *shift* that gives a best result of the matching criteria is considered as the best match or correspondence ^{[6][7]}.

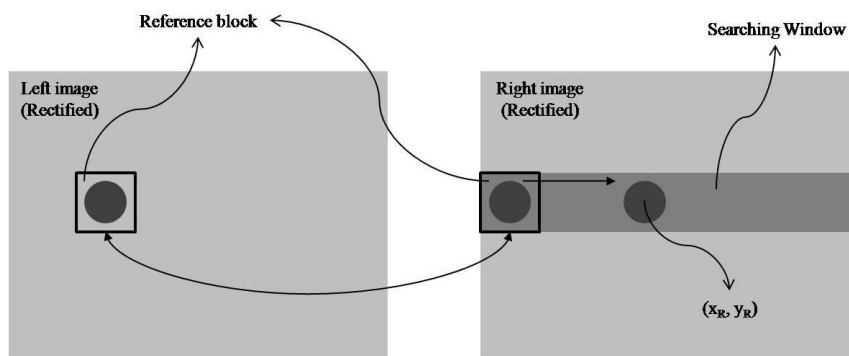


Figure (3): The block matching technique.

3. Criteria of Algorithm Performance

Matching algorithm is based on two main criteria, which would be used to support the analysis of matching algorithm results:

1. **Execution time:** represent the execution time of algorithm, and it is different from image to another.
2. **Reliability:** the information about objects should be accurate enough to recognize, and avoid obstacles.

In order to demonstrate the effectiveness of the implemented algorithms, it have been tested with many real stereo image pairs, which provided on ^[8] with ground-truth (true disparity map) that required to calculate the error percentage. The general approach used to compute error statistics with respect to ground truth data is:

- The percentage of bad matching pixels;

Is the 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.

$$p = \frac{1}{N} \sum_{(x,y)} |d_C(x,y) - d_T(x,y)| \quad (4)$$

N is the total number of pixels ^[9].

4. Standard Matching Algorithm (SMA)

This algorithm represents the standard algorithm in stereo vision, which is based on block matching ^[7]. Using this algorithm as a first pass for determining a disparity map and to illustrate its result. Many statistical measurements for the intensity values are used to check the correspondence between two blocks such as Sum of Absolute Differences (SAD), Sum of Square Differences (SSD) and Normalized Cross Correlation (NCC). These functions give equivalent results in terms of accuracy, but (SAD) requires less computational time compared with others, therefore it will be adopted in this paper. Extraction disparity map is achieved by moving the reference window in left image *pixel-by-pixel* (full search) over the target windows in right image within a specific search range called *disparity range* and compute (SAD) at each moving step, the minimum SAD calculated between two windows representing that these windows are *matched*. The amount of the *shift* between matched windows represents the *disparity value* (d) of the *central pixel* of the reference image. This procedure is repeated for whole pixels in the reference image to compute their disparity values. These values are sorted to produce *disparity map*. The SAD is calculated according to the equation:

$$SAD = \sum_{(i,j) \in W(x,y)} |I_L(i,j) - I_R(i,j+d)| \quad (5)$$

Where: $W(x,y)$ is a window surrounding the position (x,y) , d is a disparity value, I_L and I_R are the intensity values in the left and right images respectively. This algorithm has been implemented with MATLAB program and executed with different stereo image pairs at different window sizes to illustrate its results. One of the stereo pairs called (Tsukaba) with size of 384×288 shown in figure (4) would be chosen as an example to compare the results of the algorithms.

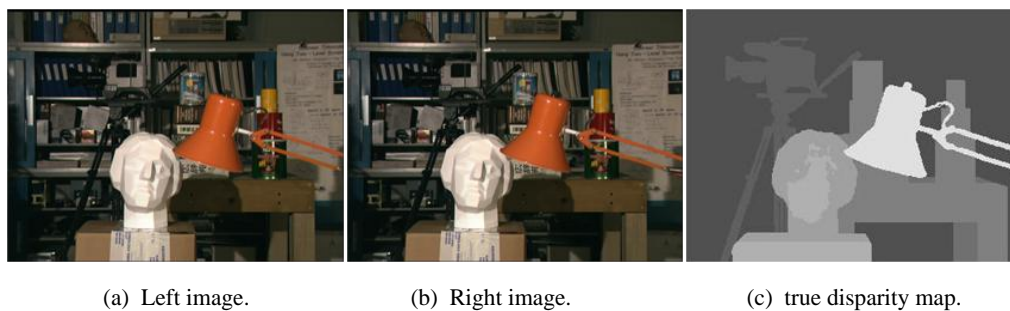


Figure (4): Tsukaba stereo image pair with true disparity map.

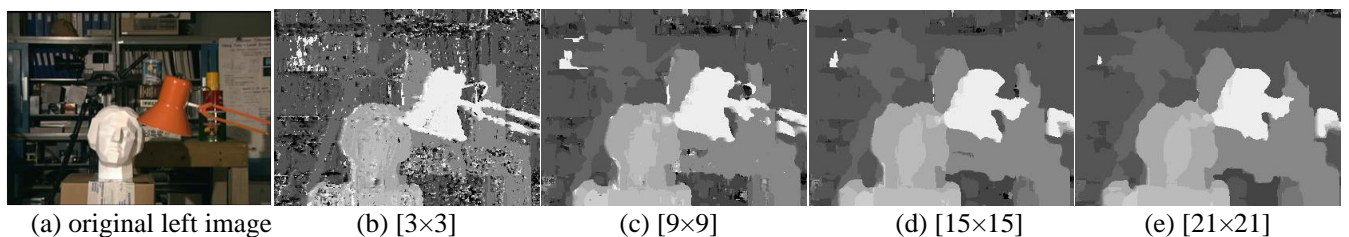


Figure (5): Shows (SMA) results, (a) represent Tsukaba left image, (b), (c), (d), and (e) are the disparity maps at different window sizes.

The result of matching algorithm is a disparity map that can be showed as image, which represents the 3D of the 2D images (left and right) as shown in figure (5). The disparity is inversely proportional to depth, it can be seen that the closes object to the stereo cameras has large disparity value and appears in *bright gray color*. On the other hand, the farthest object has less disparity value and appears in *dark gray color*.

The results of the (SMA) in figure (5) show that, at small window size the map seems noisy due to multi-matching problem, in which for each reference window in the left image there is *many matching windows* in the right image, because using too small window which does not cover enough intensity variation gives poor disparity estimation especially at *texture-less region*. Additionally at too large window size high error occurs due to ambiguity problem, this ambiguity arises from covering multi-objects by the window. Consequently, the matching result caused blurring at objects borders as shown in figure (5 (e)).

In general, this algorithm gives good disparity maps with less error as shown in (table1 column3), because it is based on moving the reference window *pixel-by-pixel* over the target image (full search technique), and calculate the *disparity* for each pixel. But the drawback of this method represented in highly processing time. This time increases as the window size increases, due to increase the number of pixels within the window, thus more computational operation is required as shown in (table1 column5). This reason makes this algorithm unsuitable for robot application that required fast performance. Therefore, another algorithm has been proposed to overcome these drawbacks. Variable window size (adaptive window) is adopted to solve these problems with vertical edge detection to reduce the computation that results in decreasing the time.

5. The Proposed Matching Algorithm (SABMA)

In this method the search for the matched-blocks is achieved by moving the reference window *Block-by-Block* over the target windows. This method achieves a *large reduction* in execution time. In which the same disparity (depth) is given to all objects that exist in the same window, but this caused blurring at object borders when the window covers Multi-objects as shown in figure (6).

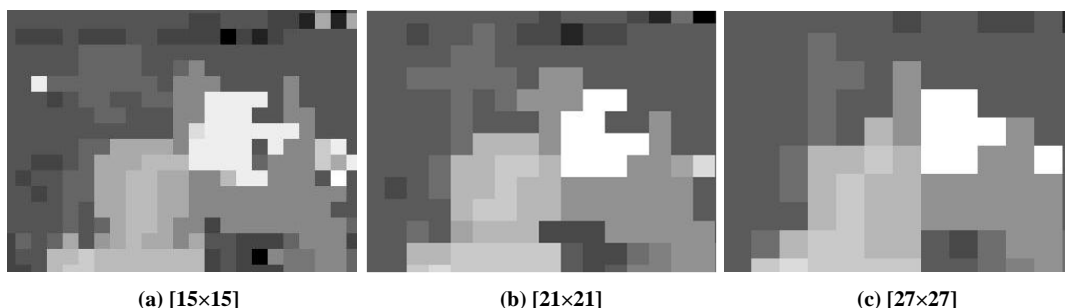


Figure (6): The blurring effect when applying Block-by-Block searching technique.

Therefore, to avoid this problem and the problems mentioned previously in SMA at objects borders, the selection of window size must be large enough to avoid the unwanted

noise, but small enough to avoid ambiguity and blurring. Therefore, *adaptive window size* is adopted, in which the window size changes adaptively according to local disparity characteristics.

The proposed algorithm is suggested to use *adaptive window* with *edge filter detection* to discriminate the borders of the objects in the scene in order to calculate the true disparity value for each object as shown in figure (7). The proposed algorithm will be called Sobel Adaptive Block Matching Algorithm (SABMA). The following three consecutive steps explain this method:

Step1. Objects Boundary Detection

Vertical edge detection is performed for both stereo images by using *Sobel filter* with suitable threshold value. Where this value determines the amount of the detected edges.

Step2. Select Window Size Adaptively

Two windows with different sizes are used:

1. Large window $[a \times a]$ to solve the problems of texture-less, and multi-texture regions, and to accelerate the matching process.
2. Small sub-window $[b \times b]$ to solve the problem of ambiguity at objects boundaries.
Where $a \geq 3b$.

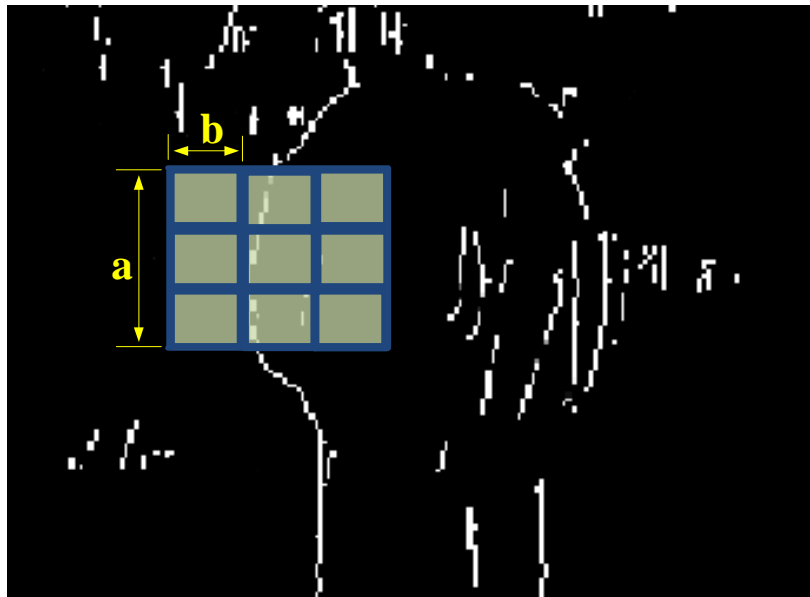


Figure (7): The division of large window $[a \times a]$ into many Sub-Windows $[b \times b]$ at object border under Sobel filter.

Step3. Perform Matching Process

The search for the matching blocks is achieved by moving the reference window *Block-by-Block* over the target windows in order to speed up the matching performance.

The division of window size is performed depending on the filtered image (*Sobel vertical edge detection*). Firstly large window size is used, and a conditional step is done on filtered image to check the presence of edges. If the window contained edges that means that there are many objects within the window that required dividing the window to obtain *accurate* depth result. Else if the window doesn't contained edges that means that the pixels within the window belong to same object, thus all these pixels have the same depth. Therefore, there is no need to divide the window. The effectiveness of this method can be seen when comparing figure (6) that represent block-by-block searching technique without adaptive window, and figure (9) which represent using the same searching technique with adaptive window size. Therefore, SABMA accomplishes compromise between the accuracy that achieved by adaptive window size and reduction in execution time that achieved by block-by-block searching technique.

To illustrate the effectiveness of (SABMA) it has been executed with different images at different window sizes as shown in figures (8), and (9):

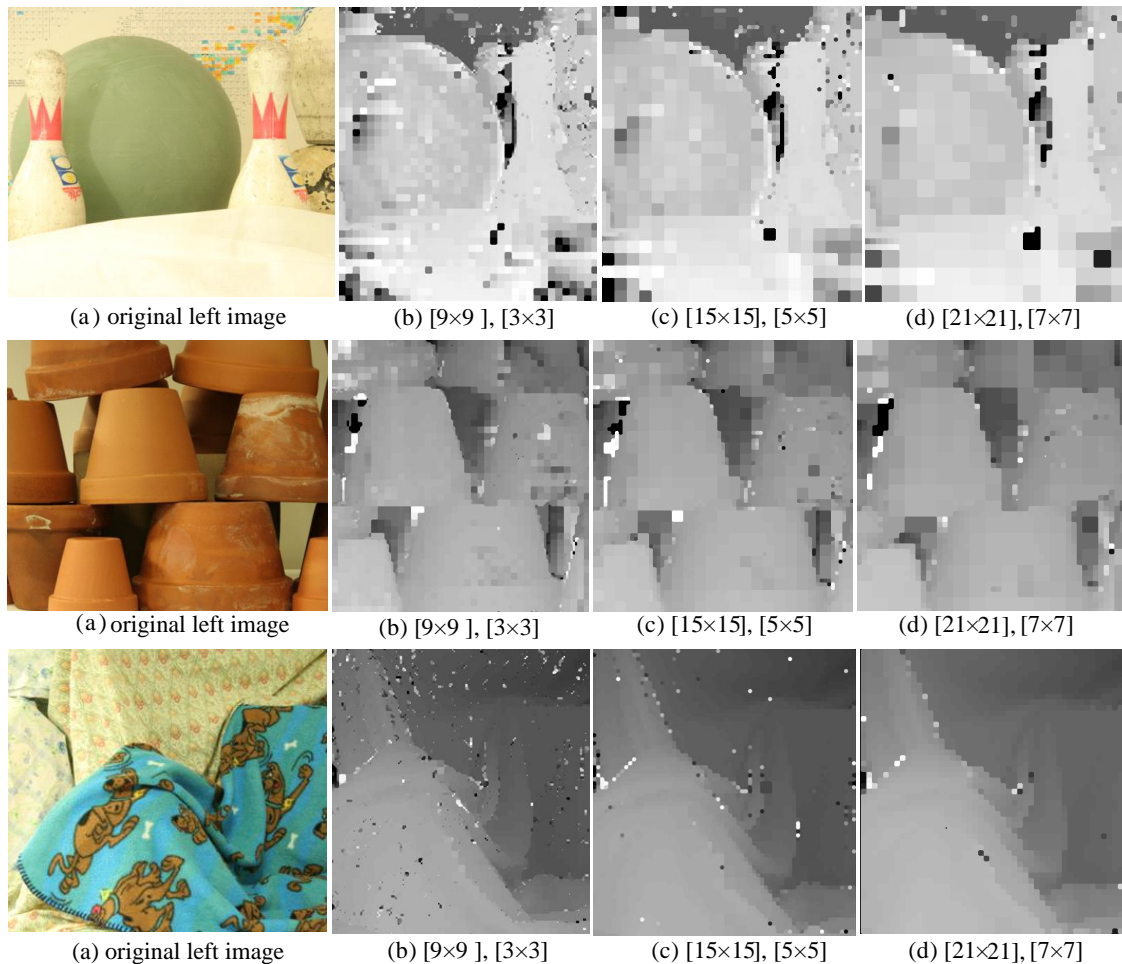


Figure (8): Disparity maps of (SABMA) algorithm at different images and different window sizes.

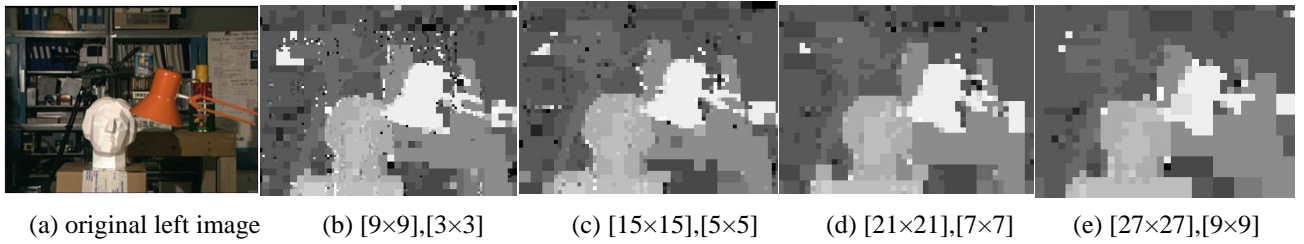


Figure (9): Tsukaba left image with disparity maps at different window sizes.

A comparison has been performed between SMA and SABMA in error percentage, and execution time for Tsukaba image pair as shown in table (1):

Table (1): The execution time and error calculation comparison of Tsukaba image pair between (SMA) and (SABMA) algorithms.

Large window size	Sub-window size	Error percentage(%) of SMA	Error percentage(%) of SABMA	Execution time (Sec.) of SMA	Execution time (Sec.) of SABMA	Execution time Percentage of SABMA with respect to SMA
[9×9]	[3×3]	11.65 %	14.25 %	8.18	0.3	3.66 %
[15×15]	[5×5]	9.93 %	11.94 %	17.04	0.28	1.64 %
[21×21]	[7×7]	11.89 %	10.81 %	28.55	0.26	0.91 %
[27×27]	[9×9]	13.82 %	13.44 %	42.50	0.25	0.58 %

Table (1) shows the reduction in execution time of SABMA as compared with SMA and the same amount of reduction has been achieved with different image pairs. This reduction is attributing to the searching for the matched-blocks technique which based on moving the reference window **Block-by-Block** over the target windows instead of **Pixel-by-Pixel** searching technique that used in SMA. Thus a large amount of computational operation required to calculate SAD is reduced as shown:-

- **For Pixel-by-pixel searching technique**

Number of SAD computation required to extract disparity map for $M \times N$ image size = $M \times N \times \text{disparity range}$.

- **For Block-by-Block searching technique**

Number of SAD computation required to extract disparity map for $M \times N$ image size = $\frac{M \times N \times \text{disparity range}}{a^2}$.

Figure (10) shows a comparison in execution time of the two algorithms.

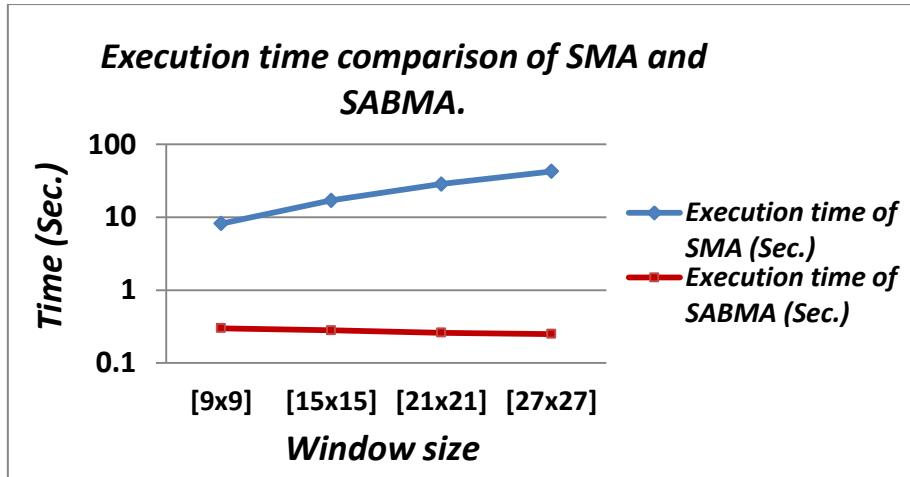


Figure (10): Comparison in Execution time between two algorithms.

Also table (1) shows that at each point the expected value of the error by the SABMA is approximately equal to that produced when any fixed-size window is used (at SMA). This attribute to use adaptive window size that achieves an enhancement in extracting disparity values at object borders, as show in figure (11).

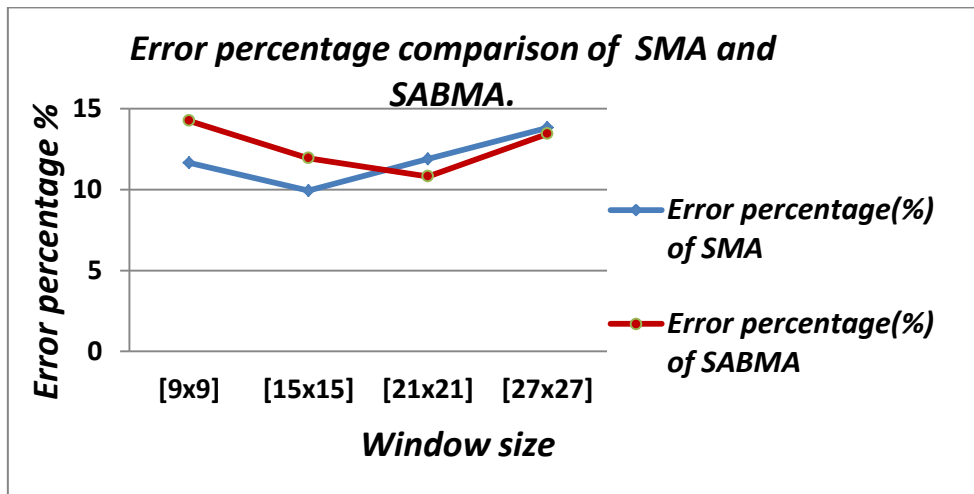


Figure (11): Comparison in Error percentage between two algorithms.

6. Implementation and Experimental Results

The stereo vision hardware used in this research is implemented using two identical models USB web cameras connected to PC (P4, core i5, 2.4 GHz processor and 6GB RAM) with USB (Universal Serial Bus) cable. As the baseline increases, there is a possibility to detect a disparity in object that farther away, since the baseline b is proportional to the range (depth) Z according to equation (3). Therefore, in this research the baseline is **16cm** selected to compromise the increases of range detection and reduces non-overlap between two images as shown in figure (12).

In order to calculate the accuracy of the estimated depth Z , a comparison has been made between the estimated depth values and the real depths of some selected points. Figure (13)

shows stereo images (left and right) with size of 320×240 captured from the implemented stereo cameras, and the disparity map which calculated with respect to left image.

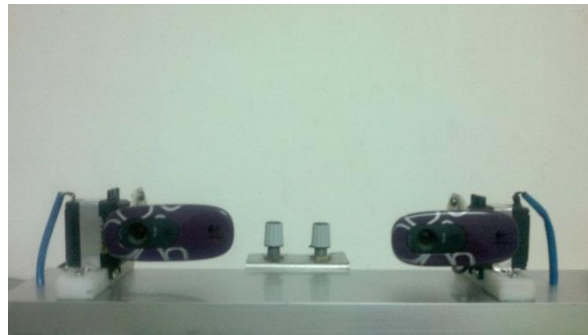
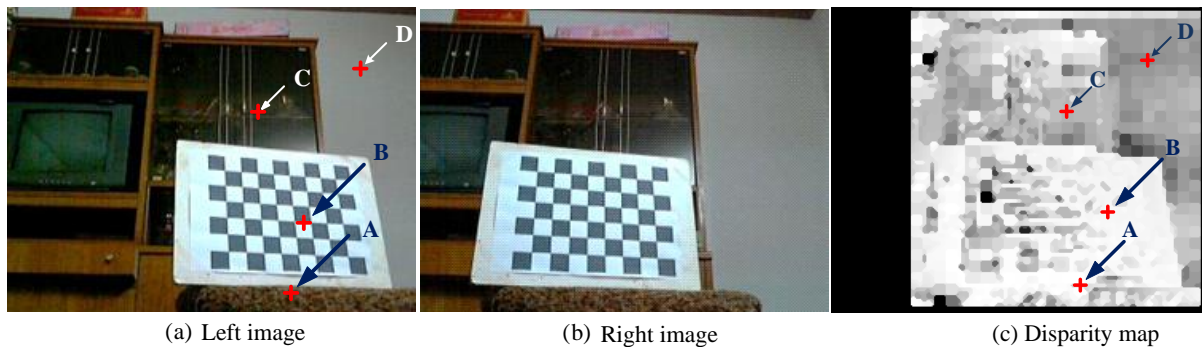


Figure (12): Stereo camera placement.



(a) Left image

(b) Right image

(c) Disparity map

Figure (13): Selecting different points at different depths.

Table (2) shows the disparity values and depths for the selected points:

Table (2): Results of depth measurement and error percentage of the estimated depth.

Tested points	disparity value (pixels)	Real depth (cm)	Estimated depth (cm)	Error percentage (%)
A	102	68	67.81	0.2 %
B	83	83	83.34	0.41 %
C	28	245	247.05	0.83 %
D	24	290	288.22	0.61 %

Table (2) shows that, the disparity value decreases for the objects which are far from the camera and it appears darker in disparity map and vice versa. Also the estimated depths that extracted from disparity map are approximated to the real objects depth with less error.

7. Conclusions

The following summarize the main important points that are concluded from this work:-

- ✚ The execution time of the SABMA reaches about **2%** of the required time to execute SMA at conventional computers, which attribute to the following proposed methods:

1. Sobel filter has a noticeable effect in reducing the time. It is found that the execution time of SMA with Sobel filter is about **72.2%** of the time required to execute this algorithm without Sobel filter.
2. **Block-by-Block** searching technique has a great influence on reducing the computational operation required to calculate SAD for disparity extraction.

✚ The adaptive window size together with Sobel filter help in enhancing the reliability of the disparity map by reducing the ambiguity at objects borders.

✚ When stereoscopic vision hardware has been implemented using two cameras, the experiment to estimate the depth of objects in real stereo images is performed. The results show good outputs with less error as compared with the real objects depths.

References:

- [1] Ambrosch Kristian and Humenberger Martin, "*Parameter Optimization of the SAD-IGMCT for Stereo Vision in RGB and HSV Color Spaces*", Article, Vienna, Austria, 2010.
- [2] Barrera Alejandra, "*Mobile Robots Navigation*", India: In-Teh, pp. (45-52), 2010.
- [3] Chinapirom Teerapat, et al, "*Stereoscopic Camera for Autonomous Mini-Robots Applied in KheperaSot League*", paper, Heinz Nixdorf Institute, University of Paderborn, Germany, 2007.
- [4] Hamzah R A, et al, "*An Aligned Epipolar Line for Stereo Images with Multiple Size ROI in Depth Maps for Computer Vision Application*", International Journal of Information and Education Technology, Vol. 1, No. 1, April 2011.
- [5] Goshtasby A. Ardeshir, "*2-D and 3-D Image Registration*", USA: A John Wiley & Sons, Inc., Publication, pp. (197-210), 2005.
- [6] Khaleghi Bahsdor, et al, "*A New Miniaturized Embedded Stereo-Vision System (MESVS-I)*", Intelligent Sensing System Laboratory, E.C.E. Department, University of Windsor, 2008.
- [7] Florczyk Stefan, "*Robot Vision-Video-based Indoor Exploration with Autonomous and Mobile Robots*", Germany: WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim, 2005.
- [8] www.middlebury.edu/stereo.
- [9] Scharstein Daniel, "*A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms*", International Journal of Computer Vision, USA: Kluwer Academic Publishers, 2002.