"Stereo Vision for ٣D Measurement in Robot Systems"

Asst. Prof. Dr. Muhyi. AL-Azawi
Electrical Eng. Dept.
AL-Mustansiriyah Univ.
Baghdad /Iraq.

Asst. Prof. Dr. Dhafer R. Zaghar
Electrical Eng. Dept.
AL-Mustansiriyah Univ.
Baghdad /Iraq.

M.Sc. Student Saba Ali Sadeq
at Electrical Eng. Dept.
AL-Mustansiriyah.
Baghdad /Iraq.

Abstract:

The major obstacle in the application of stereo vision to extract ٣D 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 (٢٪) of the required time to execute standard vision algorithm, when running at conventional computers.

Key words: *Stereo vision, disparity mapping, block matching, ٣D vision, adaptive block.*

"الرؤية المجسمة لمقياس ثلاثي الأبعاد في أنظمة الانسان الآلي"

أ.م.د.محي العزاوي
قسم الهندسة الكهربائية
كلية الهندسة/الجامعة المستنصرية

أ.م.د. ظافر رافع صغير
قسم الهندسة الكهربائية
كلية الهندسة/الجامعة المستنصرية

الطالبة صبا علي صادق
قسم الهندسة الكهربائية
كلية الهندسة/الجامعة المستنصرية

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 ايجاد حلول برمجية 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 (متغير) (Adaptive block size) مع خاصية إظهار الحافات العمودية في الـــصـور (Sobel Filter) لتنفيذ الخوارزمية. تتمثل اهمية هذا العمل في تقليل وقت تنفيذ خوارزمية الرؤية المجسمة, حيث استغرق وقت تنفيذ الخوارزمية المقترحة حوالي (٢٪) من الوقت اللازم لتنفيذ الخوارزمية القياسية عند تنفيذها في الحواسيب التقليدية.

الكلمات المفتاحية: ***الرؤية المجسمة, خارطة التباين, تطابق المقاطع, رؤية ثلاثية الابعاد, مقطع تكييفي.***

١. Introduction

For safe operation in uncontrolled environments of robots, dependable ٣D perception modules are required. Commonly used ٣D sensors for mobile robots are laser range finder and time-of-flight cameras. Although these types have advantage of delivering accurate ٣D data, but suffer from low resolution, and cannot provide enough information about the objects [١]. 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 [٢]. The stereoscopic camera (*Binocular System*) can provide the third dimension by grabbing left and right images simultaneously, like as human's binocular vision [٣]. 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.

٢. 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 ٣D model for the seen objects. In machine vision, the ٣D 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***" [٤]. 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 [٣]. Figure (١) depicts the geometrical basis for stereoscopic images by using two identical cameras.

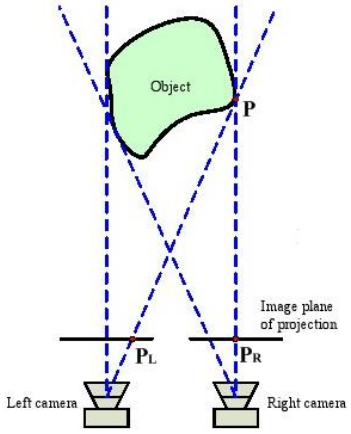


Figure (١): The position of two cameras and their image Planes of projection.

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) [٣].

٣. 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 (٢). 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.

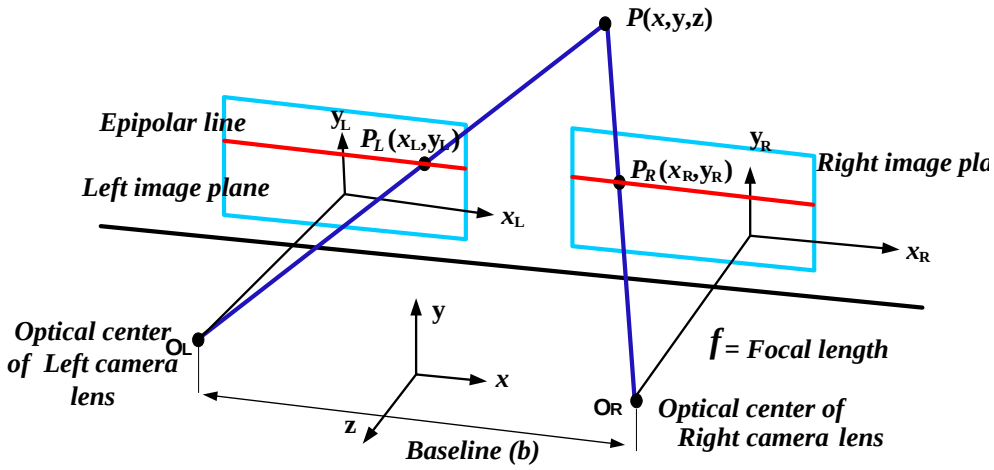


Figure (٢): The epipolar geometry of stereoscopic vision.

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

$$\frac{x_L}{f} = \frac{x + \frac{b}{2}}{z} \qquad (١)$$

$$\frac{x_R}{f} = \frac{x - \frac{b}{2}}{z} \qquad (٢)$$

$$z = \frac{b*f}{x_L - x_R} = \frac{b*f}{d} \qquad (٣)$$

Where: the disparity is $d = (x_L - x_R)$. x_L and x_R are the x-axis coordinate of point *P* in the left and right image planes respectively, and ***x*** is x-axis of stereo-camera coordinate that considered to be the midway between the left and right camera coordinate systems. f is the *focal length* that represents the distance between the *image plane* and *the optical center of camera lens*. The baseline (b) is the distance between the lens centers of the cameras.
Measurement of depth from equation (٣) requires knowledge about the coordinates of corresponding points in the images [٥].

٤. 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 [٥×٥] or [٧×٧]. The blocks are usually defined on epipolar line for matching ease. Each block from the left (reference) image is matched into a block over the searching area of pixels in right image as shown in Figure (٣). The ***shift*** that gives a best result of the matching criteria is considered as the best match or correspondence [٦] [٧].

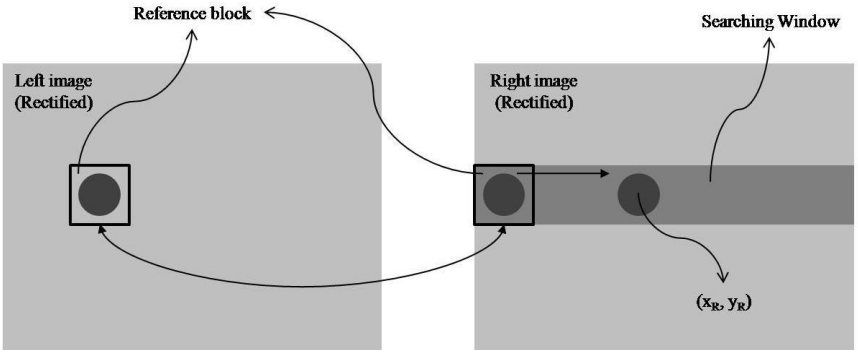


Figure (٣): The block matching technique.

٥. Criteria of Algorithm Performance

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

١. ***Execution time*:** represent the execution time of algorithm, and it is different from image to another.

٢. ***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 [٨] 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 (p);**

Is the error (measured in disparity units) between the computed depth map $d_{C(x,y)}$ and the ground truth map $d_{T(x,y)}$, as shown in Equation (٤) [٩].

$$p = \frac{1}{N} \sum_{(x,y)} | d_{C(x,y)} - d_{T(x,y)} | \qquad (٤)$$

where N is the total number of pixels.

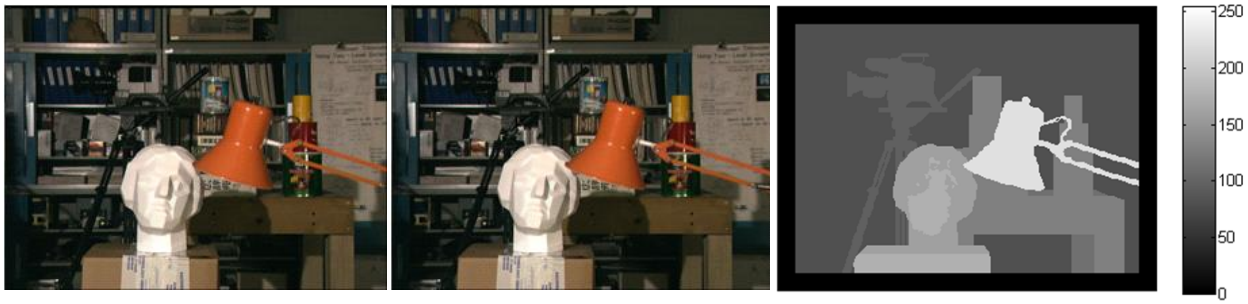
٦. Standard Matching Algorithm (SMA)

This algorithm represents the standard algorithm in stereo vision, which is based on block matching [٧]. 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 (٥) [٣١]:

$$SAD = \sum_{(i,j)\in W(x,y)} \left| I_L(i,j) - I_R(i,j+d) \right| \qquad (٥)$$

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 (٧,٦) technical programing language, 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 ٣٨٤×٢٨٨ shown in Figure (٤) would be chosen as an example to compare the results of the algorithms.



(a) Left image (b) Right image (c) true disparity map

Figure (٤): Tsukaba stereo image pair with true disparity map.

٦,١. Summary of SMA

The matching algorithm SMA which implemented by MATLAB, can be summarized as the following steps:

Step ١. Convert left and right images from (RGB scale) color image to Gray scale image, to reduce the algorithm's complexity, which is multiplied with the number of color channels used.

Step ٢. Prepare a ٣D array (x, y, d) for disparity map, where (x, y) represent the coordinates of each pixel in the reference image (left), and d is for the disparity value of these pixels.

Step ٣. Set the size of matching window, and Set initial match value (Initial SAD).

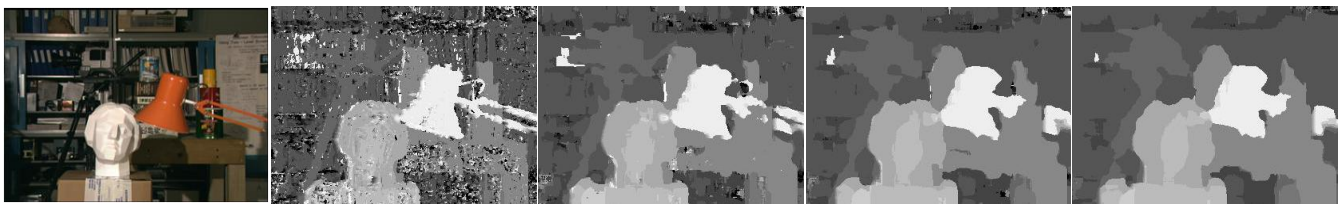
Step ٤. Set the value of "Search Range" (d = ٠,…., d_{max}.) that represents the search range within a loop. The value of (d_{max}) is empirical and represents the maximum value of the shift in pixels between the left and right images.

Step ٥. Compute SAD between reference window and candidate windows according to the Equation (٥) at each shift.

Step ٦. Iteratively update the initial SAD value with every new minimum SAD computed, until the match value converges.

Step ٧. For each pixel (x, y) of the reference image, find the disparity element (x, y, d). The set of disparity values represents the disparity map; the disparity is often treated as synonymous with inverse depth.

The result of matching algorithm is a disparity map that can be shown as image, which represents the ٣D of the ٢D images (left and right), as shown in Figure (٥). 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***.



(a) original left image (b) [3×3] (c) [9×9] (d) [15×15] (e) [21×21]

Figure (٥): Shows (SMA) results, (a) represent Tsukaba left image, (b), (c), (d), and (e) are the disparity maps at different window sizes.

The results of the (SMA) in Figure (٥) 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 (٥.e).

In general, this algorithm gives good disparity maps with less error as shown in (Table١ column٣), 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 (Table١ colunm٥). 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.

٧. The Proposed Matching Algorithm (SABMA)

In this method the search for the matched-blocks is achieved by moving the reference widow ***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 (٦).

(a) [15×15] **(b) [21×21]** **(c) [27×27]**

Figure (٦): 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 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*** (***Sobel filter***) 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 (٧). The proposed algorithm will be called Sobel Adaptive Block Matching Algorithm (SABMA). The following three consecutive steps explain this method:

Step١. Objects Boundary Detection

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

Step٢. Select Window Size Adaptively

Two windows with different sizes are used:

١. Large window [a×a] is used to solve the problems of texture-less regions, and to accelerate the matching process.

٢. Small sub-window [b×b] is used to solve the problem of ambiguity at objects boundaries. Where (b) is chosen as a part of (a) to increase the accuracy, e.g., $b \leq {}^{٣}a$.

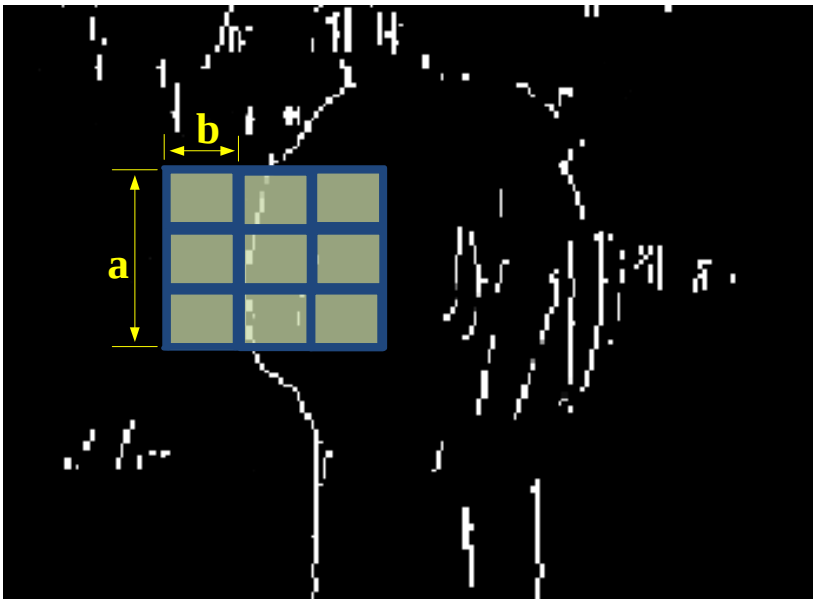


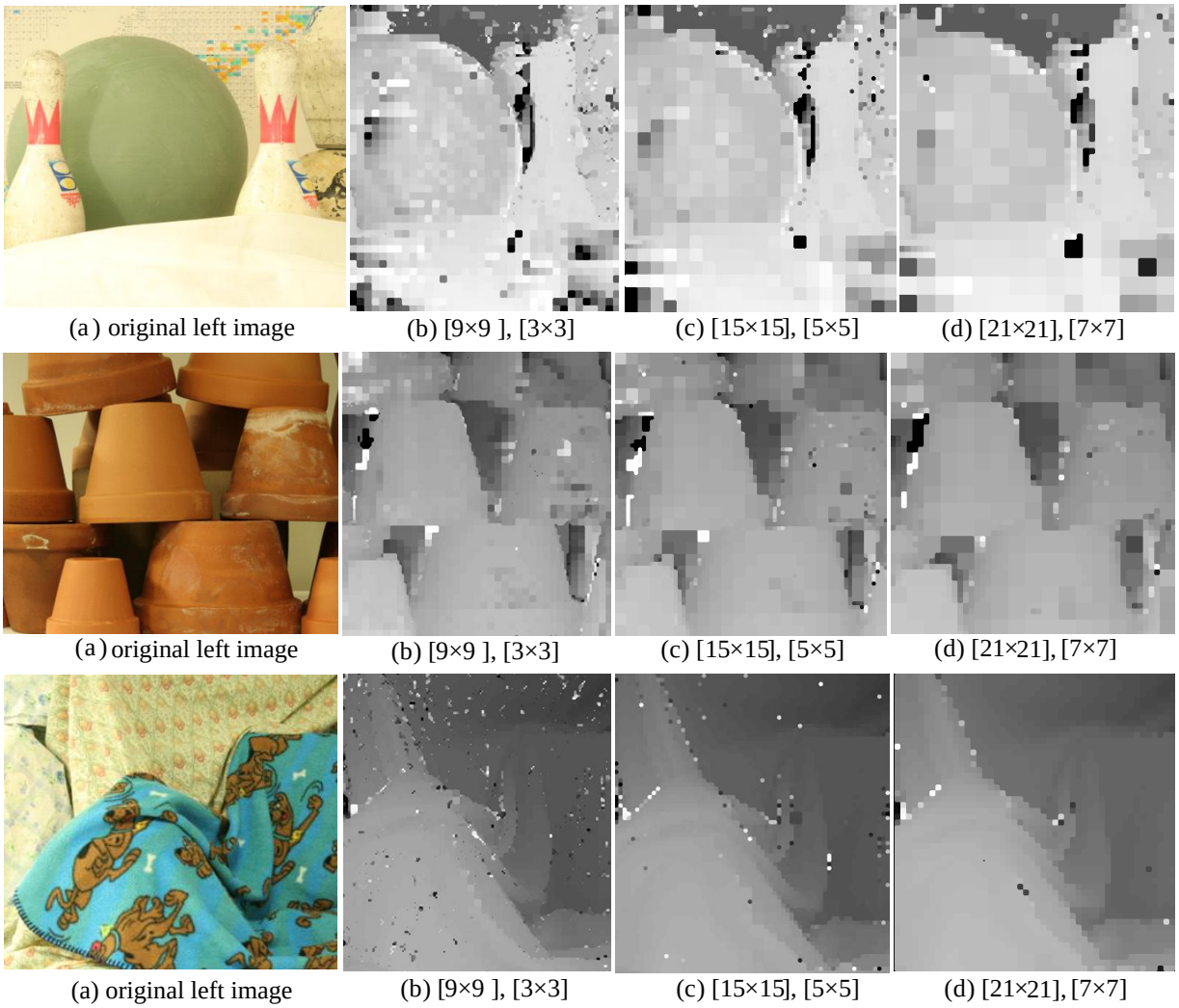
Figure (٧): The division of large window [a× a] into many Sub-Windows [b× b] at object border under Sobel filter.

Step٣. 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, 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 which 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 (٦) that represents block-by-block searching technique without adaptive window, and Figure (٩) that represents 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 (٨) and (٩):

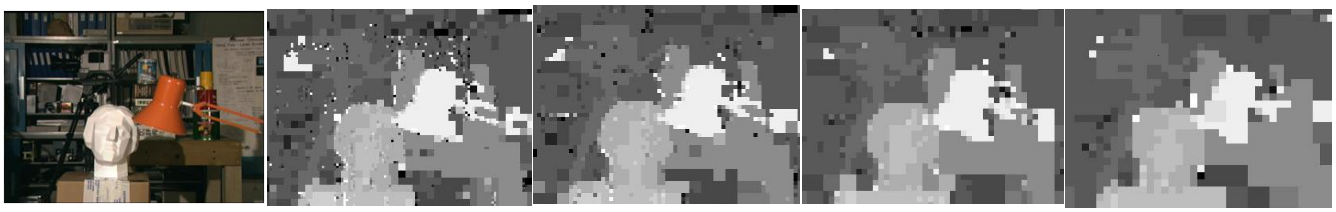


(a) original left image (b) [9×9], [3×3] (c) [15×15], [5×5] (d) [21×21], [7×7]

(a) original left image (b) [9×9], [3×3] (c) [15×15], [5×5] (d) [21×21], [7×7]

(a) original left image (b) [9×9], [3×3] (c) [15×15], [5×5] (d) [21×21], [7×7]

Figure (٨): Disparity maps of (SABMA) algorithm at different images and different window sizes.



(a) original left image (b) [9×9],[3×3] (c) [15×15],[5×5] (d) [21×21],[7×7] (e) [27×27],[9×9]

Figure (٩): 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 (١):

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

[٩×٩]	[٣×٣]	١١,٦٥ %	١٤,٢٥ %	٨,١٨	٠,٣	٣.٦٦ %
[١٥×١٥]	[٥×٥]	٩,٩٣ %	١١,٩٤ %	١٧.٠٤	٠,٢٨	١.٦٤ %
[٢١×٢١]	[٧×٧]	١١,٨٩ %	١٠,٨١ %	٢٨.٥٥	٠,٢٦	٠.٩١ %
[٢٧×٢٧]	[٩×٩]	١٣,٨٢ %	١٣,٤٤ %	٤٢.٥٠	٠,٢٥	٠.٥٨ %

Table (١): The execution time and error calculation comparison of Tsukaba image pair between (SMA) and (SABMA) algorithms.

Table (١) 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×N image size = **M× N×** ***disparity range***.

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

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

Figure (١٠), shows a comparison in execution time of the two algorithms.

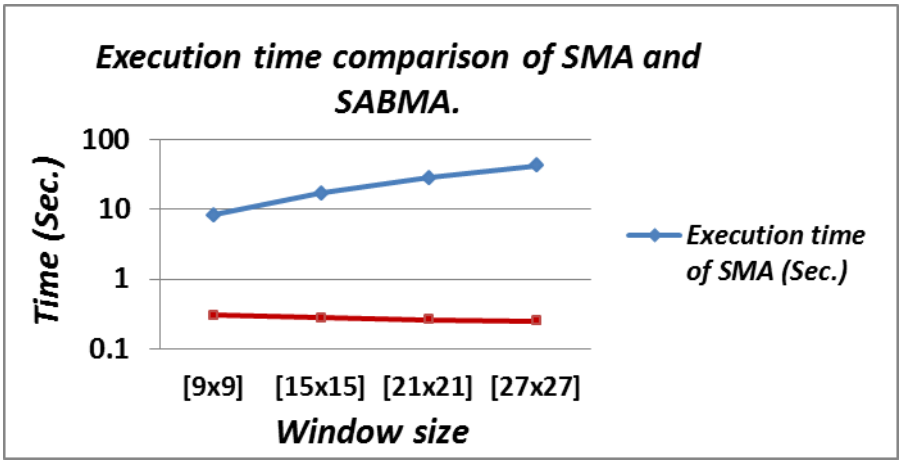


Figure (١٠): Comparison in Execution time between two algorithms.

Also Table (١) 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 (١١).

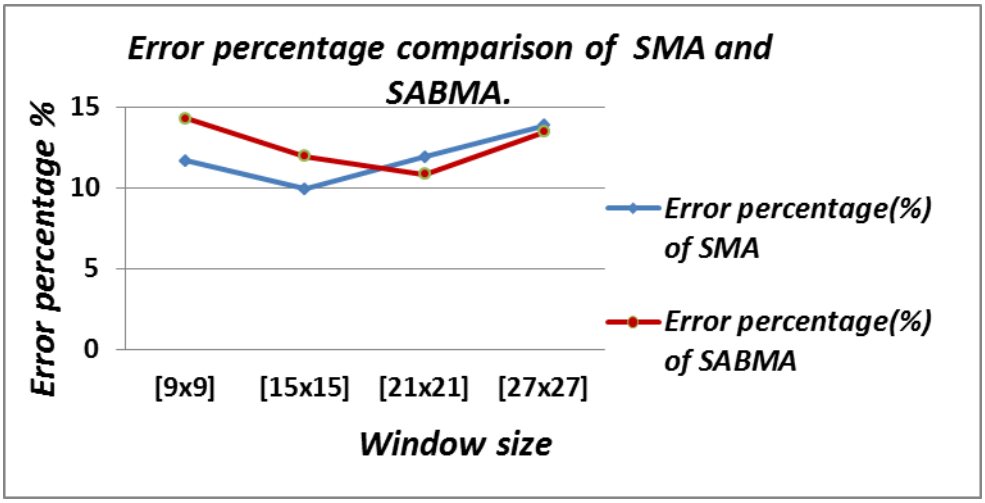


Figure (١١): Comparison in Error percentage between two algorithms.

٨. Implementation and Experimental Results

The stereo vision hardware used in this research, is implemented using two identical models USB web cameras, connected to PC (P٤, core i٥, ٢,٤ GHz processor and ٦GB 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 (٣). Therefore, in this research the baseline is **١٦cm** selected, to compromise the increases of range detection and reduces non-overlap between two images, as shown in Figure (١٢).

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 (١٣) shows stereo images (left and right) with size of ٣٢٠×٢٤٠ captured from the implemented stereo cameras, and the disparity map which calculated with respect to left image.

Figure (١٢): The implemented stereo cameras.

(a) Left image (b) Right image (c) Disparity map

Figure (١٣): Selecting different points at different depths.

Table (٢), shows the disparity values and depths for the selected points:

Table (٢): 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	١٠٢	٦٨	٦٧.٨١	٠,٢ %
B	٨٣	٨٣	٨٣,٣٤	٠,٤١ %
C	٢٨	٢٤٥	٢٤٧,٠٥	٠,٨٣ %
D	٢٤	٢٩٠	٢٨٨,٢٢	٠,٦١ %

Table (٢) 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.

٩. Conclusions

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

١. The execution time of the SABMA reaches about ٢٪ of the required time to execute SMA at conventional computers, which attribute to the following proposed methods:

- **i.** Sobel filter has a noticeable effect in reducing the time. It is found that the execution time of SMA with Sobel filter is about ٧٢,٢٪ of the time required to execute this algorithm without Sobel filter.
- **ii.** ***Block-by-Block*** searching technique has a great influence on reducing the computational operation required to calculate SAD for disparity extraction.

٢. Selection of window size does not only influence the execution time of the algorithm, but also the accuracy of disparity map. Therefore, the use of adaptive window size with Sobel filter helps in enhancing the reliability of the disparity map, by reducing the ambiguity at objects borders. This can be seen when less error is obtained at large window size, e.g., [٢١×٢١] and [٢٧×٢٧] of SABMA compared with SMA, as shown in Table (١).

٣. 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, which is attributed to good result of disparity (d) values obtained by the SABMA.

١٠. Suggestions for Future Works

The following suggestions can be suggested for future works:

١. Extending the current work for motion estimation and velocity extracting of the moving objects from many successive pairs of frames captured at different times.

٢. Extending the current work to increase the field of view of the implemented stereo vision system by using more than two cameras.

٣. It is suggested to implement the SABMA with another programing language (e.g., C++) to approach real time performance.

References:

[١] Ambrosch Kristian and Humenberger Martin, "***Parameter Optimization of the SAD-IGMCT for Stereo Vision in RGB and HSV Color Spaces***", Article, Vienna, Austria, ٢٠١٠.

[٢] Barrera Alejandra, **"*Mobile Robots Navigation*",** India: In-Teh, pp. (٤٥-٥٢), **٢٠١٠.**

[٣] Chinapirom Teerapat, et al, **"*Stereoscopic Camera for Autonomous Mini-Robots Applied in KheperaSot League*",** paper, Heinz Nixdorf Institute, University of Paderborn, Germany, **٢٠٠٧.**

[٤] 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. ١, No. ١, **April ٢٠١١.**

[٥] Goshtasby A. Ardeshir, **" *٢-D and ٣-D Image Registration*",** USA: A John Wiley & Sons, Inc., Publication, pp. (١٩٧-٢١٠), **٢٠٠٥.**

[٦] Khaleghi Bahsdor, et al, **"*A New Miniaturized Embedded Stereo-Vision System (MESVS-I)*",** Intelligent Sensing System Laboratory, E.C.E. Department, University of Windsor**,** **٢٠٠٨.**

[٧] Florczyk Stefan, **"*Robot Vision-Video-based Indoor Exploration with Autonomous and Mobile Robots*",** Germany: WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim, **٢٠٠٥.**

[٨] **www.middlebury.edu/stereo.**

[٩] Scharstein Daniel, **"*A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms*",** International Journal of Computer Vision, USA: Kluwer Academic Publishers, **٢٠٠٢.**