

Disparity of Stereo Images by Self-Adaptive Algorithm

Md. Abdul Mannan Mondal¹, Mohammad Haider Ali²

Dept. of Computer Science and Engineering
University of Dhaka
Dhaka, Bangladesh

Abstract—This paper introduces a new searching method named “Self Adaptive Algorithm (SAA)” for computing stereo correspondence or disparity of stereo image. The key idea of this method relies on the previous search result which increases searching speed by reducing the search zone and by avoiding false matching. According to the proposed method, stereo matching search range can be selected dynamically until finding the best match. The searching range $-d_{max}$ to $+d_{max}$ is divided into two searching regions. First one is $-d_{max}$ to 0 and second one is 0 to $+d_{max}$. To determine the correspondence of a pixel of the reference image (left image), the window costs of the right image are computed either for $-d_{max}$ to 0 region or for 0 to $+d_{max}$ region depending only on the matching pixel position. The region where the window costs will be computed will be automatically selected by the proposed algorithm based on previous matching record. Thus the searching range is reduced to 50% within every iteration. The algorithm is able to infer the upcoming candidate’s pixel position depending on the intensity value of reference pixel. So the proposed approach improves window costs calculation by avoiding false matching in the right image and reduces the search range as well. The proposed method has been compared with the state-of-the-art methods which were evaluated on Middlebury standard stereo data set and our SAA outperforms the latest methods both in terms of speed and gain enhancement with no degradation of accuracy.

Keywords—Stereo correspondence; stereo matching; window cost; adaptive search; disparity; sum of absolute differences

I. INTRODUCTION

Stereo correspondence is required in applications such as autonomous vehicle and robot navigation, virtual reality, stereo image coding in 3D-TV and pedestrian detection. The difference between the coordinates of left image and right image of the corresponding pixels is known as stereo correspondence or disparity, which is inversely proportional to the distance of the object from the camera. Stereo correspondence is a common tool in computer or robot vision, essential for determining three-dimensional depth information of an object using a pair of left and right images from a stereo camera system. For a pixel in the left image, its correspondence has to be searched in the right image based on epipolar line and maximum disparity. Stereo correspondence is conventionally determined by matching windows of pixels using Sum of Square Differences (SSD), Sum of Absolute Differences (SAD), or Normalized Correlation Techniques (COR).

Window-based stereo correspondence estimation technique is widely used due to its efficiency and ease of implementation. However, there is a well-known problem in the selection of an appropriate size and shape of window [1- 2]. If the window is small and does not cover enough intensity variation, it gives erroneous result due to the low signal to noise ratio. If the window is large, it includes a bigger region where the disparity varies or discontinuity of disparity happens and the result becomes erroneous due to different projective distortions in the left and right images. To overcome this problem, many researchers proposed adaptive window techniques using windows of different shapes and sizes [3-7]. In the existing adaptive window technique involves comparing the window costs for different window sizes and shapes, so the computation time is relatively higher than that of fixed window technique. In case of gray scale stereo images, the use of color stereo images brings a substantial gain in accuracy with the expense of computation time [8]. The Virtual Masking System has been employed in [9]. The Virtual Masking System is based on the stereo matching constraint states the corresponding pixels should be similar in color or intensity [10]. The Virtual Masking System technique improves the computational efficiency by excluding unlikely correspondences. Two-stage Approximation Algorithm, Arbitrary Window Pixel, Diagonal Mask Searching were also reported in [11-12] and survived the minimum values among the three searches to overcome the window-based problems.

The main achievement of the proposed algorithm is to improve the huge incremental computational schemes by dynamically readjusting the search range and eliminating redundant computational calculations. The proposed adaptive search method (SAA) is able to avoid false matching in the right image pixels that causes to increase the visual quality of the estimated stereo correspondence. SAA performs according to needful in the two new searching zones of the right image. Thus reduces the computational time that ensures the better speed of experimental dense disparity images as shown in Fig. 15 and Fig. 16. In fact, the proposed SAA approach optimizes the computational time and speed. The main contribution of this paper is fully self-adaption based adaptive function with the following advantages.

- Experimental results show that the 3D reconstruction of output disparity map is very similar to ground truth dense disparity.

- We observe 535 *fps* for input images with 384×288 (Tsukuba head pair) pixel resolution and 377 *fps* for input images of Venus stereo pair with 434×383 resolution.
- The proposed method is applied on the Middlebury standard stereo images and the results show that it reduces 70.53% computational time for Venus stereo pair and 99.93% computational time for Tsukuba stereo pair with no degradation of accuracy.
- Numerical evaluations confirm that the bad pixel is only 6.2% with error threshold 1 for Tsukuba head.

II. RELATED WORK

Many researchers do research on a dense two-frame stereo in many ways. They try to optimize the dense disparity in locally or globally on a stereo pair. So the dense matching algorithms are divided into local and global ones [36]. The best classifications have presented by Scharstein and Szeliski [13] and many new methods have been proposed here. Local methods are also known as area based stereo matching that can perform better speed compare to global methods. According to this, disparity is being calculated at a point in a fixed window. Global methods are also known as intensity or energy based stereo matching that can perform better accuracy compare to local methods. In this method, the global cost function is reduced to minimum as possible. This cost function synthesizes image data and smoothness terms. Neural adaptive stereo matching [15] are done by trained neural networks based on window size and shape. The effects of shape of the stereo image pair presented in [16]. Yoon et al. [17] used SAD method and a left-right consistency check that performed like real time system. Yoon's method can process 7 *fps* for 320×240 pixels images with 32 disparity levels. The experiment has been implemented by using an Intel Pentium 4 at 2.66GHz Processor. The uses of Cellular Automata (CA) are presented in [18]. Real-time extraction of disparity maps was demonstrated by this method. It can process the input image with a speed of 40 *fps*. This method also based on SAD matching cost. A window-based method uses different support-weights in [19]. The support-weights of the pixels in a given support window are adjusted based on geometric proximity and color similarity to reduce the image ambiguity in [20]. The running time for the Tsukuba image pair with a 35×35 pixels support window is about 0.016 *fps* on an AMD 2700+ processor. The research work presented in [20] based on unified framework that supports the fusion of any partial knowledge such as matching features and surfaces about disparities. Physical processes and systems are designated by a cell array which expressed the new state of a cell depending on the states of its neighbors [21].

Bayesian estimation theory with a prior Markov Random Fields model for the assigned disparities is described in [22]. According to this method, the continuity, coherence, occlusion constraints and the adjacency principle are taken into considerations. Image color segmentation is reported in [23]. By this method, disparity map is estimated using an adapting window based technique. The segments are combined in larger layers iteratively. Bidirectional matching is used in [24] to

discard the false matching and the authors employed the technique to calculate the disparities of stereo images without known epipolar geometry. In [25] pruning technique is used to estimate the disparities.

The recent works of related problem to the matching costs are stated in [26] and [27]. The authors used bilateral filter to determine the cost aggregation and in order to reduce the computational cost they also limit the label space. The work in [28] can be considered as a cost aggregation method by guided image filter. The average runtime [28] of the four standard Middlebury data sets (including Tsukuba, Venus, Teddy and Cones data sets) is 960 milliseconds reported in [31]. So the run time of single image pair like Tsukuba or Venus is about 240 milliseconds. Disparity space image (DSI) structure and gradient information has been combined as a new technique is first time introduced by Nadia Baha and Slimane Larabi [29]. They used DSI technique with adaptive window – support. Another approach is introduced by themselves as DSI and refinement. The experimental results take time 0.2 *second* and 0.39 *second* respectively for processing Tsukuba head image pair. A new geodesic $o(1)$ filter is employed in [30] for the reliable disparity propagation. Such type of filter is very effective for the cost matching. As it is state-of-the-art method and the speed of this method has been justified on the Middlebury standard data set, so we can compare this paper to our proposed SAA method. Xun Sun et al. [30] perform the experiment on PC equipped with a 3.0 GHz Intel i5 CPU, 8 GB of memory and a Geforce GTX 580 graphics card. The processing time on Middlebury standard data set is only 9 milliseconds. A cost aggression has been adaptively estimated on a tree structure derived from the stereo image pair to preserve depth edges [31]. This latest idea is launched by Q. Yang [31] in which shortest distances measure the similarity between two pixels on the tree. The average runtime of the four standard Middlebury data sets (including Tsukuba, Venus, Teddy and Cones data sets) is about 90 milliseconds using the tree filtering method. But He et al. [31] mentioned in same section that the runtime is about 7 milliseconds on average on the Middlebury data sets. For identical comparison to our proposed method we consider his second result that takes 7 milliseconds on average on the Middlebury data sets. Q. Yang tested his experiment on a MacBook Air laptop with a 1.8 GHz Intel Core i7 CPU and 4 GB memory. Another recent method achieves state-of-the-art result on Middlebury stereo data sets that performs stereo matching as a two steps energy-minimization algorithm [32]. The running time of this method is 3 seconds only for Tsukuba data set and 20 seconds for Teddy data set on a computer containing an Intel Core i5-4300U 1.9-GHz CPU and a 6-GB RAM. Semi-global matching and cost is refined by cross-based aggression [33] has been introduced by J. Zbontar and Y. LeCun. Y. LeCun et al. [33] also uses left-right consistency check to eliminate the errors. The experiment performs on KITTI stereo data set. Depth of information is estimated by learning a similar measure on image patches [34]. At very recent, Fusing Adaptive Support Weights [38] has been launched by Wenhuan Wu and others. Local and global support windows used for each pixel in [38]. Self-guided cost aggression [39] has been determined by deep learning method that depends on two sub-networks. A pyramid stereo matching network [40] also consists of two modules

based on pyramid and 3D CNN that have been tested on KITTI 2012 and 2015 data set. Adaptive Weighted Bilateral Filter [41] is used as main filter at cost aggregation step for edge preserve factor.

With the above reviews we found that some researchers employed adaptive window-based techniques to calculate the matching costs. But in our proposed method we employ *self-adaptive function* to calculate the matching costs dynamically. This is one of the main distinguishable points between the proposed SAA method and existing state-of-the-art methods. The mentioned recent methods are very similar to our proposed method but differing mostly in optimal searching technique. Besides these analysis, the work in [30] requires preprocess and the works in [29], [31], [33] and [35] needed post processing steps like refinement, filtering and histogram equalization. The proposed SAA method also runs without preprocessing and post processing. The experimental disparity maps are directly eligible to compare with ground truth dense disparity. So considering the adaptive similarity, identical stereo data set (Middlebury Standard data set) and hardware platform we can consider the papers of [14], [24], [25], [29], [30], [31], [32], [38], [39] and [40] to compare between the state-of-the-art method and our proposed SAA method.

III. COMPARISON WITH EXISTING MATCHING ALGORITHMS

The proposed matching mechanism is compared with the early established four stereo matching methods: (1) A Fast Area based Algorithm (2) Bidirectional matching or left-right checking (3) Hierarchical disparity estimation and (4) Window-based fast algorithm. First one approach is kept trace previously matched points, while the second one calculates every possible combination of matches from left to right and right to left. So it is also called left-right consistency checking. The hierarchical disparity method calculates disparities either for rectified stereo images or uncalibrated pairs of stereo images without known epipolar geometry [24]. Additionally this method also uses bidirectional matching to remove false matches. Window-based fast algorithm uses the same idea of first method but it additionally uses the threshold technique. The coloring area defines the probable matching points of right image. According to bidirectional matching, first scan takes place in the direct phase i.e. from left to right and find the all scores associated with the reference image. Suppose the matching pixels are marked by white circle along the horizontal color path in Fig. 1. Secondly in reverse phase it is searched to find the best match again and a match is accepted if it is the only one the matching pixels of first search i.e. left to right. It is noted that during the reverse phase bidirectional searching, checks all the potential matches are along the middle line of horizontal path but only those are allowed marked by black circle with white center and those fall in the white circle area of the middle horizontal line [14] for $R(x,y)$. The comparisons of the proposed method and recent works [26-40] are explained briefly in the last portion of section-II.

On the contrary in our proposed adaptive matching method, the best match occurs by minimum computational cost with dynamically readjusted the searching starting and ending pixels. According to the main concepts matching pixels are

those which are the nearest neighbors of $R(x,y)$. The searching points are dynamically readjusted based on the said main concepts of underlying algorithm. Accordingly, 1st search occurs on all the lying pixel ranges from $R\{x +(- d_{max}), y\}$ to $R\{x +(+d_{max}), y\}$.

First search for first reference pixel is to be accomplished through the whole area of searching regions. Second, third and consecutive searches are to be adapted. By the completion of first search the proposed algorithm keeps track the position of matching pixel also. So the second matching may occur surrounding the first matching pixel, as we use the concepts that neighbor pixels have the same photometric properties. So depending on the position of matching pixel, the successive search areas are reselected. It will be either $- d_{max}$ to 0 or 0 to $+ d_{max}$ region. This adaptive matching procedure will be continued for every candidate's pixel within the search range. The main advantages of proposed method (SAA) compared to the above mentioned existing established methods relies on its lowest computational costs. The *left-right check* has proven to be particularly effective in detecting and discarding the erroneous matches but it requires two matching phases (direct and reverse). This implies doubling the computational complexity of the matching process. The fast area based algorithm is based on a matching core that it does not require a reverse matching phase but some details such as for example the lamp's wire (Tsukuba pair), the lamp's switch and the two roads that sustain the lamp, have been vanished. Moreover, the disparity map shows the *border-localization* problem, i.e. the objects' borders are not perfectly localized with respect to their original position. This algorithm requires only direct matching phase. The hierarchical method executed on DirectX 8.1 class 3D hardware (ATI Radeon 9000 Mobility). The disparity map is verified by bidirectional procedure. Window-based fast algorithm [25] uses the different threshold values like 10,20,30,40 and 50. From the experiment we explore that as soon as the threshold value increased the searching range also increased causes to high computational costs. Conversely, the proposed SAA method shows lowest computational cost because it involves only minimum matching spaces. Suppose there are n candidate pixels appear on the searching range. According to the above analysis, the fast area based algorithm requires n^2 numbers of matching iterations and the bidirectional or hierarchical search requires $2n^2$ matching iterations. The proposed method requires only $\{(n+1) + (n/2) \times (n-1)\}$ matching iterations.

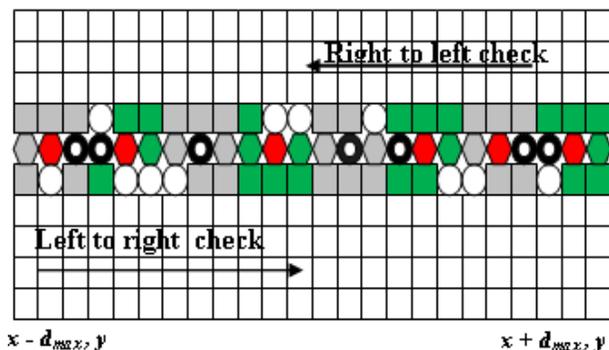


Fig. 1. Computational Path of Bidirectional and Unidirectional Matching from the Computational Point of View.

IV. PROPOSED SELF-ADAPTIVE SEARCH METHOD

A new stereo imaging search technique has been introduced in this section. In traditional window based searching algorithm, a particular pixel $L(x,y)$ is selected by search method along the corresponding epipolar line in the right image within a search range from $-d_{max}$ to $+d_{max}$. Assuming the left image as the reference image, for most pixels in the left image, there is a corresponding pixel in the right image within a search range from $-d_{max}$ to $+d_{max}$.

Accordingly, the first search of first reference pixel of $L(x-d_{max})$ is searched to the right image from $R(x+(-d_{max}))$, $R[x+(-d_{max}+1)]$, $R[x+(-d_{max}+2)]$... $R[x+0]$... $R[x+(d_{max}-1)]$ to $R(x+d_{max})$. During the matching process the algorithm finds the best candidate pixel by evaluating its window costs function within the interval $[R\{x+(-d_{max}), y\} \dots R\{x+(+d_{max}), y\}]$. The method is visually explained in Fig. 2 and Fig. 3 by mortars its coordinate's pixel. Suppose the window cost function $f(wc) = \{wc_1, wc_2, wc_3 \dots wc_n\}$. Let $wc_2 < wc_1$ so the best match occurs for the cost function $f(wc_2)$ and the function associated with the corresponding pixel of right image say $R(x_2, y)$ to indicate that this match from left to right has been established. Assume another pixel $R(x_4, y)$ is associated with the cost function $f(wc_4)$. If $f(wc_4)$ has better score than the previous $f(wc_2)$.i.e., $f(wc_4) < f(wc_2)$, this algorithm will reject the score of wc_2 and will accept wc_4 . Therefore, the function $f(wc_4)$ associated with the corresponding pixel of right image say $R(x_4, y)$ indicates the new matching establishment. Thus the coordinate distance from $R(x,y)$ to $R(x_4,y)$ is the final disparity of reference pixel of $L(x-d_{max})$. The process is then repeated for the successive pixels of reference image along with the scan line from left to right of the whole image.

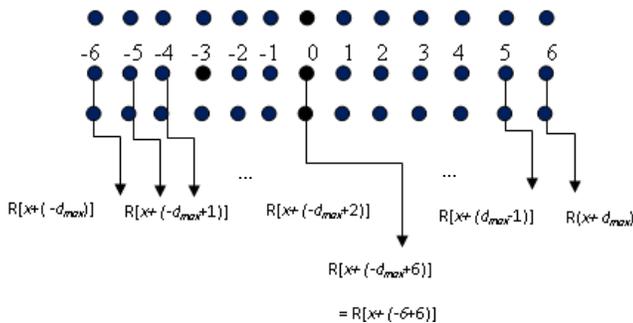


Fig. 2. The Total Search Regions of Right Image for the Particular Pixel of $L(x,y)$.

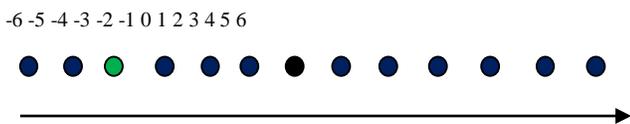


Fig. 3. First Search Interval $\{R(x+(-d_{max})) \dots R(x+(+d_{max}))\}$.

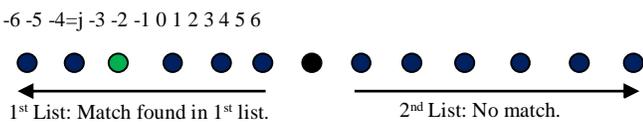


Fig. 4. Search Range Separated by 1st List and 2nd List with their Candidate's Pixels.

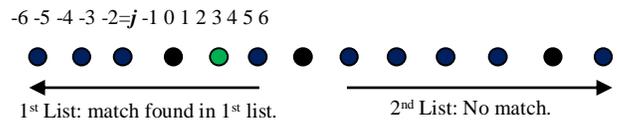


Fig. 5. Second Search Occurs in 1st List too, and the Matching Pixel is Indicated by Green Color also.

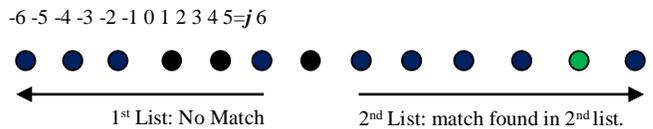


Fig. 6. 4th Search Occurs in 2ndList, Match Indicated by Green Color.

First search matching occurs into in 1st list and the matching pixel indicated by the green color at position $j=-2$.

According to SAA method, if the matching pixel point $R(x_4,y)$ of first search is “ j ” indicates the x -position of matching point in the 1st list. This procedure is illustrated in Fig. 4. In the proposed matching approach we use prior knowledge-based adaptive search technique for 2nd reference pixel matching. As per sequence of proposed method, 2nd reference pixel obviously resides surrounding to the j^{th} position i.e., within the 1st list, as the neighbor pixel's photometric properties are approximately same as $L(x-d_{max})$. So the second search for 2nd reference pixel performs in the 1st list too. This search matching procedure is depicted in Fig. 5. In case of critical situation occurs in second search; suppose the 2nd reference pixel does not match with the candidate's pixels of 1st list. Then the SAA search sequence goes to new searching zone i.e. in the 2nd list with readjusting the search interval. Accordingly, the SAA algorithm calculates the window costs of 2nd reference pixel in 2nd list. This procedure is outlined in Fig. 6. In this case, the candidates' pixels of 1st list are not taken into consideration for matching process. Next search for 3rd reference pixel occurs again in 2nd list too (which is not shown here). In case of no match in 2nd list for 3rd reference pixel, the program search sequence goes to 1st list by resetting the new starting and ending points of 1st list. The searching procedure of 3rd reference pixel will be the same as illustrated in Fig. 5. Next search for 4th reference pixel occurs in 1st list also as per base criteria of the proposed approach. So the matching procedure occurs either in 1st list or in 2nd list. In essence, if no match occurs in 1st list, the program sequence goes to 2nd list and if no match occurs in 2nd list, the program sequence goes to 1st list and vice-versa. The above stated SAA approach is repeated for the successive pixels of reference image along with the scan line from left to right on the whole image. The algorithm divides automatically the search interval $[R\{x+(-d_{max}), y\} \dots R\{x+(+d_{max}), y\}]$ into two regions; 1st list and 2nd list. 1st list ranges from $-d_{max}$ to 0 while 2nd list dominates from 0 to $+d_{max}$. The capability of proposed SAA algorithm is to adapt itself the search range automatically. This process reduces the iterations and tracks the exact position of next matching pixel. The proposed approach relies on j^{th} position of x axis. According to the proposed approach the algorithm explores the new searching starting and ending points of first and second list. This procedure is iterated for successive pixels in interval $[R\{x+(-d_{max}), y\} \dots R\{x+(+d_{max})$

, y] along with the scan line for the whole image. In this paper $(2n+1) \times (2n+1)$ mask size is used to estimate the window cost over the every scan line; where $n=1,2,3...k$. The right image is scanned by this mask from left to right and from top to bottom during the matching process.

Suppose, if $d_{max}=6$ and $j=0$, the algorithm resets the starting point at $d_{max1}=-d_{max}$ and ending point $d_{max2}=+d_{max}$. In this case matching process occurs on the full scan line. Actually it is treated as 1st search as shown in Fig. 7 in which window costs calculation are shown here. There are 13 candidates' pixels in right image along with the scan line. For each pixel, window cost is calculated according to SSD method. So there are thirteen window costs that have been extracted at 1st search, although five window costs are shown here for simplicity. The proposed algorithm arranges the window costs function in ascending order. Suppose the order for 1st search is like as $f(wc) = \{wc_2 < wc_1 < wc_3 < \dots < wc_{13}\}$. So the proposed method rejects all other window cost except wc_2 . Best match occurs for 2nd window due to its minimum window cost. Thus the reference pixel $L(x-d_{max})$ matches with the candidate's pixel (center pixel) of the second window of right image.

As the first matching pixels position is $j=-2$ (since $j < 0$) illustrated in Fig. 4 and Fig. 8, the algorithm resets the search interval $d_{max1}=-d_{max}$ to $d_{max2}=0$. In this region only six pixels are participants, rest of the pixels of 2nd list are ignored due to $j < 0$. This idea is exactly illustrated in Fig. 8.

So there are only six window costs has been extracted from 2nd search, though there three window costs are shown in Fig. 8 for simplicity. The proposed algorithm arranges the window costs function in ascending order. Suppose the order for 2nd search is like as $f(wc) = \{wc_1 < wc_4 < wc_3 < wc_2 < wc_1 < wc_6\}$. Hence the proposed method rejects all other window costs except wc_1 . Best match occurs for 1st window cost in first list due to its minimum window cost. Thus the reference pixel $L(x-d_{max}+1)$ matches with the candidates pixel (center pixel) of the first window of right image. Since the reference pixel $L(x-d_{max}+1)$ matching position is $j=-5(j < 0)$ shown in Fig. 8, so third search (not shown here) will be continued into 1st list also. If it falls into the critical situation in 4th search, i.e. pixel $L(x-d_{max}+1)$ does not match with 1st list, the algorithm readjusts the new searching zone's interval $d_{max1}=0$ to $d_{max2}=+d_{max}$ and the searching sequence goes to 2nd list. The proposed approach calculates the window costs function following the same procedures applied in 2nd search. The calculations of window costs for 4th search are outlined in Fig. 9 in which the starting and ending points are quite different from 2nd and 3rd search and three window costs are shown for simplicity. Actually there are six window costs has been estimated. So the proposed SAA method achieves the prior knowledge of matching pixel position in x axis and accordingly it redirects the program sequence in the 1st list or the 2nd list. As the matching procedure occurs either in 1st list that contains half of candidates pixels or in 2nd list that also contains half of candidate's pixels, so it reduces 50% or above the iterations process. In next section, the iterative reductions in percentage are figured out.

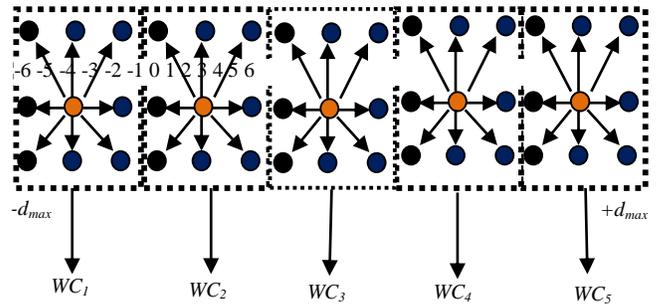


Fig. 7. Window Cost Estimation Process Over the Scan Line (1st Search).

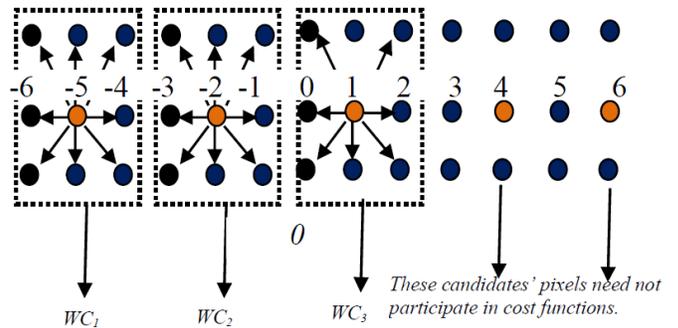


Fig. 8. Window Cost Calculation Process for 1st List Only (2nd Search).

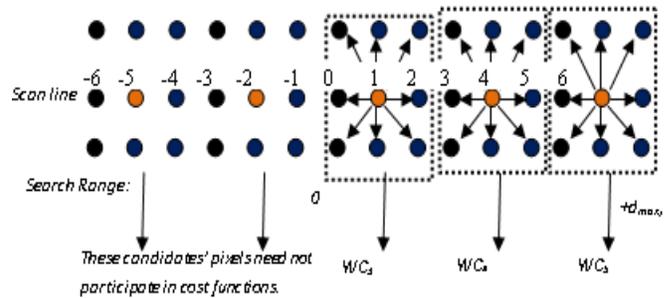


Fig. 9. Window Cost Calculation Process for 2nd List Only (4th Search).

A. Disparity estimation Algorithm of SAA method for a pixel (x,y) .

1. Start.
2. Repeat step 2 to 10 for each pixel (x,y)
3. If $flag=0$,
4. Set: $d_{max1}=-d_{max}, d_{max2}=d_{max}$;
5. else if $(flag < 0)$
6. Set: $d_{max1}=-d_{max}, d_{max2}=0$;
7. else
8. Set: $d_{max1}=0, d_{max2}=d_{max}$;
- [end of if]
9. for $d' = d_{max1}$ to $+d_{max2}$ do
Calculate $W_c((x,y,d'))$
10. Find best $W_c(x,y,d) \in W_c(x,y,d')$
11. [End of the step 2]
12. Disparity of $(x,y) = d$;
13. if $(j < 10)$
14. Set: $flag = -1$, goto step 6;
- else
15. Set: $flag=1$, goto step 8;
16. Stop.

The above procedure shows the calculation of window cost for one pixel only. The next matching position range λ' is being selected by setting the flag pointer either -1 or +1 with the following statements:

```
if(j<10)
flag = -1;
else
flag = 1;
```

Flag pointer controls the reduced matching position range λ' and thus both the computational costs and number of iterations have to reduce always.

V. OPTIMIZATION OF SELF-ADAPTIVE SEARCH

The main highlight of SAA approach is optimization technique. Two types of optimizations are accomplished here simultaneously i) iteration and ii) computation. We have already mentioned 50% or above iterations reduced at every reference pixel of left image except first pixel $L(x-d_{max})$. First pixel (black color of left image) traverses from $R(x-d_{max})$, $R(x-d_{max}+1)$, $R(x-d_{max}+2)$... $R(x+d_{max})$ along the scan line. So the first search occurs from $-d_{max}$ to $+d_{max}$ over the scan line. In this case, for first search window cost has been calculated for every candidate's pixels of right image for every window using the mask of size 3x3. Suppose the first reference pixel matches with the third pixel($x-d_{max}+2$) of right image as shown in Fig. 10. After tracking the first matching, the area only. This process will be continued for successive pixels along the scan line.

Let $w_1 \times w_2$ is the mask size and matching range is λ . So the first searching computational cost $F_1 = \{(w_1 \times w_2) - 1\} \times (\lambda + 1)$.

Second optimization cost $F_2 = \{[(w_1 \times w_2) - 1] \times (\lambda + 1)\} / 2$. Third, fourth ... λ^{th} searching computational costs be the same i.e. $F_{\lambda-1} = \{[(w_1 \times w_2) - 1] \times (\lambda + 1)\} / 2$. Fast area based algorithm [14] requires for every reference pixel computational cost, $F_{AB} = \{(w_1 \times w_2) - 1\} \times (\lambda + 1)$.

But in hierarchical disparity [24] or left -right checking algorithm requires for every reference pixel computational cost, $F_{HD} = 2 \times \{(w_1 \times w_2) - 1\} \times (\lambda + 1)$.

For the image size is $M \times N$, the total computational costs of Fast area based algorithm [14] is $F_{AB} = (M \times N) \times \{(w_1 \times w_2) - 1\} \times$

$(\lambda + 1)$. Total computational costs of hierarchical disparity method's [24] is $F_{HD} = 2 \times [(M \times N) \times \{(w_1 \times w_2) - 1\} \times (\lambda + 1)]$.

The computational costs of proposed SAA method, $F_{SAA} = [(M \times N - 1) \times \{(w_1 \times w_2) - 1\} \times (\lambda + 1)] / 2 + \{(w_1 \times w_2) - 1\} \times (\lambda + 1)$.

Suppose there are n candidates' pixels within the search range λ .

1) So according to the proposed SAA method the 1st search comparison occurs only in $n+1$ pixels. For $(-n+1)^{th}$ pixel matching i.e. for 2nd search the SAA method sets the reduced match range λ' . proposed method divides the searching space into two regions indicated by 2nd and 3rd arrow in right image.

Suppose $flag$ is a tracking pointer, if $flag < 0$ then SAA algorithm sets $d_{max1} = -d_{max}$; and $d_{max2} = 0$; else $d_{max1} = 0$, $d_{max2} = +d_{max}$. So the upper bound and lower bound of matching regions are set according to above criterion and falls into two regions. As per theme of section -IV the second pixel (green color of left image) of reference image searched only 2nd search area. Accordingly, the third pixel (violet color of left image) of reference image is matched only 3rd search area indicated by the same color in right image. So the subsequent reference pixel matches either 2nd search area or 3rd search.

This match occurs in $n/2$ pixels and all forthcoming matching will be occurred within $n/2$ pixels. So excluding first pixel there are $(n-1)$ rest of pixels within the search range. Hence the total iterations are for proposed SAA method = $(n+1) + (n/2) \times (n-1)$. But for Fast area based algorithm the total iterations are = n^2 . Left -right checking or bidirectional requires the total iterations = $2 \times n^2$.

2) In a stereo matching n varies from -10 to $+10$. So let $n=10$ means $n/2=5$. Thus the total iterations are for proposed SAA method = $(n+1) + (n/2) \times (n-1) = (10+1) + (11/2) \times (11-1) = 66$.

3) Fast Area Based algorithm requires the total iterations are = $n^2 = 11 \times 11 = 121$.

4) Hierarchical disparity method or bidirectional requires the total iterations = $2 \times n^2 = 2 \times 11^2 = 242$.

5) So iteration reduction compared to the Fast Area Based algorithm = $(55/121) \times 100\% = 45.45\%$.

6) Iteration reduction compared to Hierarchical disparity method or bidirectional method = $(176/242) \times 100\% = 72.72\%$.

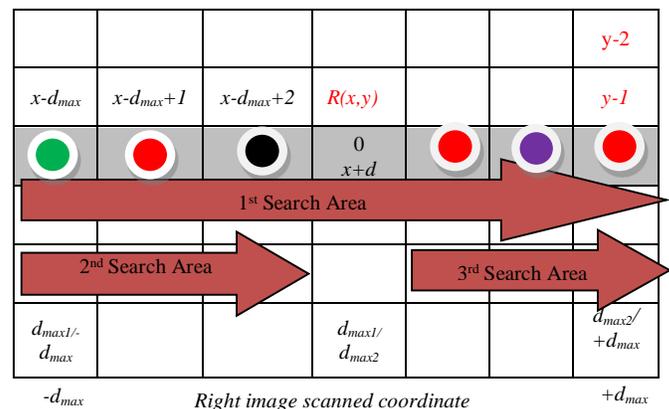
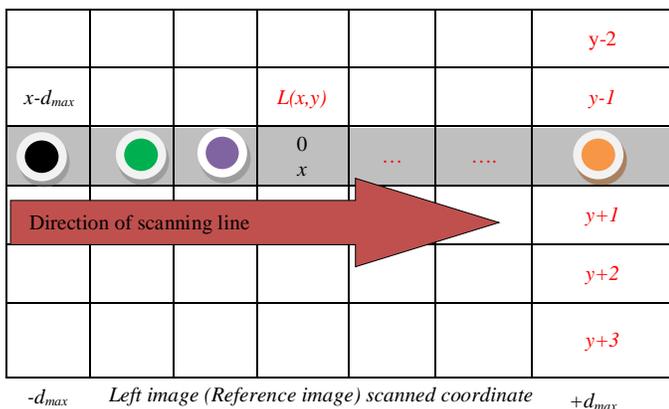


Fig. 10. Scanning Process from $-d_{max}$ to $+d_{max}$ on Right Image Mentioning the Co-Ordinate System.

Flag pointer always controls match range λ' by changing the parameter d_{max1} and d_{max2} that depends upon the previous matching position of x . From d_{max1} to d_{max2} for every window cost is determined by summing up the product of difference between two pixels by the following pseudo statements.

sum = sum + (v_left - v_right)*(v_left - v_right); the parameters v_left and v_right of (i,j)th set up by

v_left = image_left.pixel[m+i][n+j];

v_right = image_right.pixel[m+i+d][n+j];

where i and j varies from -1 to +1 for of a mask size 3×3 . The window costs are stored in an array Mtemp[$d + d_{max}$].pixel[m][n]. These cost values are gone to minimum function as arguments. The minimum function implements the key idea of the proposed method by assessment of its comparisons among the window costs. The function performs the dual tasks i) it can be able to determine the matching x axis pixel position " j " by setting $j=i$ and find the desired disparity with the following pseudo statement-

1) if (temp[i].pixel[x][y] < temp[j].pixel[x][y])

j = i;

and

2) The SAA algorithm selects the upcoming matching position range λ' by setting the flag pointer either -1 or +1 with the following statement.

if(j<10)

flag = -1;

else

flag = 1;

Flag pointer controls the reduced matching position range λ' and thus both the computational costs and number of iterations has been reduced always.

VI. EXPERIMENTAL RESULTS

The experiments have been performed on two Middlebury standard stereo images: i) Tsukuba stereo pair, ii) Venus stereo pair as shown in Fig. 11. The computational time, speed, accuracy and gain performances of the proposed algorithm have been justified over the said stereo images (Tsukuba Head and Venus stereo pair). Experiments are performed on Intel Core i3, 2.3 GHz processor with 4 GB DDR3 RAM. The algorithms have been implemented using Visual C++ programming language. The size of the left and right images of Tsukuba head is (width×height) = (384×288) pixels and, the ground truth image size is (width×height) = (348×252) pixels. The size of the left and right images of Venus stereo is (width×height) = (434×383) pixels and the ground truth image size of Venus is (width×height) = (348×252) pixels. Mask size of 11×11 is used for every operation in this research.

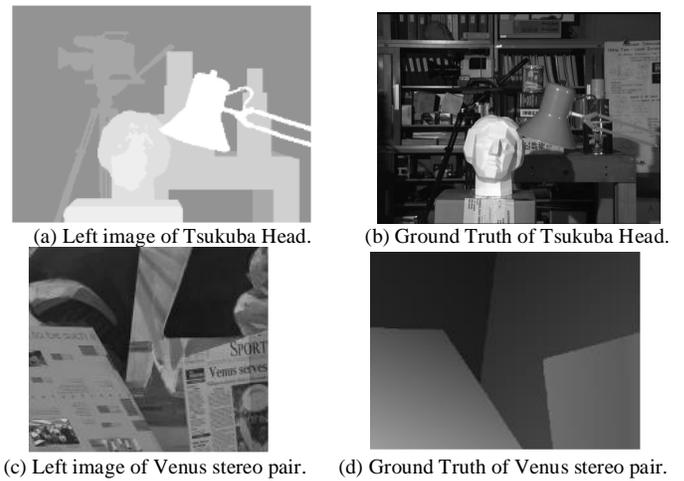


Fig. 11. Standard Stereo Image (Reference Image) and their Ground truth Image.

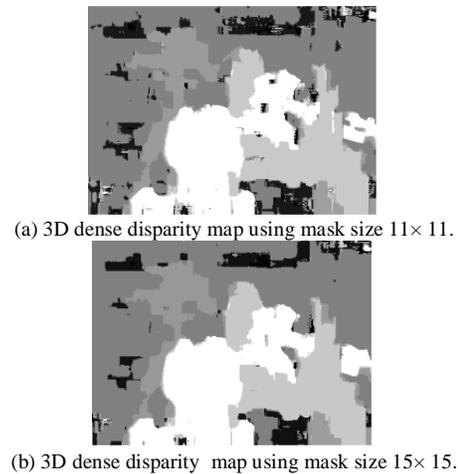


Fig. 12. Estimated 3D Dense Disparity Map of Tsukuba Head using SAA Method.

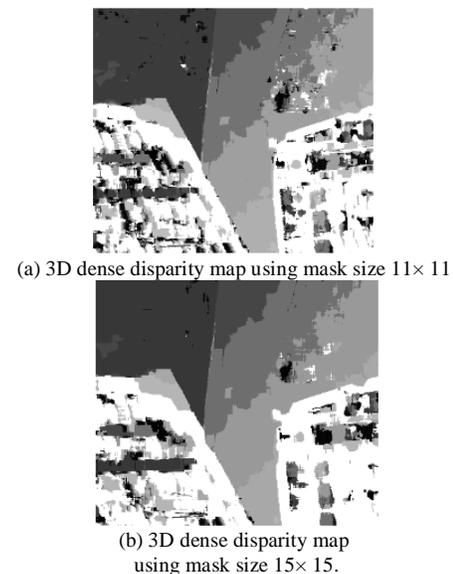


Fig. 13. Estimated 3D Dense Disparity Map of Venus Stereo using SAA Method.

The experimental results state that the SAA algorithm is currently the best window cost method among all existing state-of-the-arts methods. The top performer algorithms are reported in [28], [29], [30], [31] and [32]. All are ranked by Middlebury benchmark [37]. So we have to prove the claim by comparing the time and speed with the top performer algorithms which is demonstrated in Table I and Table III. The disparity maps of the Middlebury data sets for Tsukuba head and Venus stereo pair are estimated by proposed SAA method illustrated in Fig. 12 and Fig. 13. The corresponding numerical time reductions are depicted in Table II and Table IV. From the tables it is evident that in both cases the proposed SAA algorithm outperforms the current and earlier top performer algorithms. Moreover, the proposed method is faster than all others top performer algorithms. The accuracy of the SAA algorithm for Tsukuba head is 93.8% i.e. the bad pixel in percentage with the error threshold is only 6.2% which is almost the same of the top algorithms. Little variation of accuracy occurs due to orientations of pixel redundancy. The performance enhancements by the said SAA method are discussed in Section VII. Table V and VI show the noteworthy gain enhancement of proposed SAA method. The experimental results analyzed in four phases are stated.

A. Experiment 1: Observation of 3D Structures of Experimental Output

Both the Tsukuba and Venus stereo pair of input images contain different objects at variable depth of positions. Contextual and forefront objects are positioned at different depth. Four objects are placed at different depth of Venus stereo of input image. Stereo pair also encloses some distinct areas like head of the statue, table lamp, video camera, Venus sport paper, and another paper and background paper wall. Such types of areas are quite challenging to isolate from other objects by stereo matching. So the first work is to differentiate the variable depth of objects by assigning the altered gray level value of output image. Nearby object is presented by deep white color and outermost object is presented by dark or deep black color. It is detecting that the experimental 3D construction of output image is recreated evidently in Fig. 12 and Fig. 13 where the face of the statue, table lamp, video camera, Venus sport paper, another paper as well as remarkable objects are known easily. Comparing the output images of Fig. 12(a) and Fig. 13(a) with ground truth image of Fig.11(b) and Fig.11(d), respectively, the camera and its background objects have been recovered almost correctly and the object closer to the stereo acquisition system. The objects depths are noting that nearby objects are realized by additional white color and outermost objects are realized by dark grey level value or black as shown in Fig. 12 and Fig. 13. Thus, the camera and nearby objects such as face of Tsukuba, table lamp and sports paper, 2nd paper (to be such: left side) of Venus stereo are visualized by more white color. On the other hand, the camera and outermost objects such as video camera, book shelf, background wall of Tsukuba stereo pair and background paper wall of Venus stereo are recreated with dark grey or black color. Object borders are evidently detecting in computed dense disparity image, i.e. border localization problem of article [14] are resolved by the SAA method. The experimentally estimated images are tested again by the edge

detection algorithm and the output object's borders are acknowledged which are illustrated in Fig. 14.

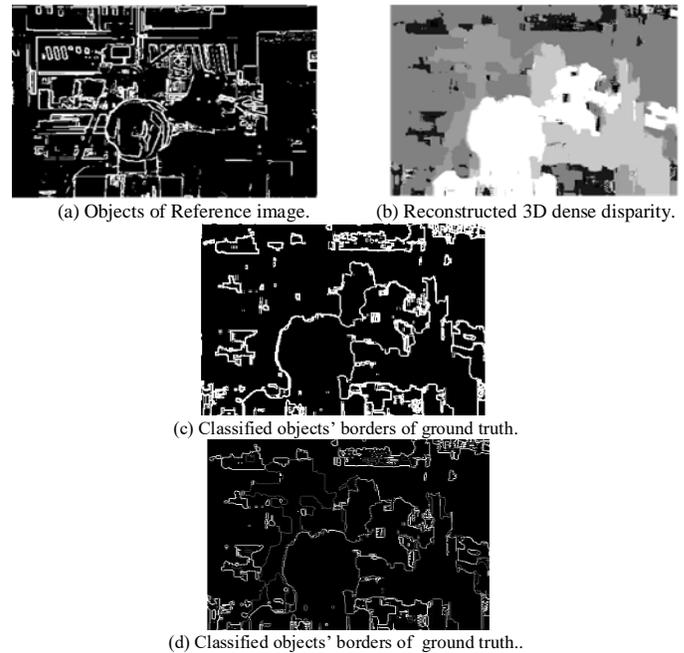


Fig. 14. Localized Objects' Borders.

The estimated dense disparity's 3D structure is recovered and its objects border are correctly identified which are outlined in Fig. 14(c) and (d). So the result ensures that the similar depths are found in estimated dense disparity.

B. Experiment 2: Computational Cost Calculation and Comparison with Existing State-of-the-Arts Methods for Middlebury Standard Tsukuba Head Image Pair

Disparities of reference image are estimated by SSD technique using adaptive search algorithm without any pruning. The disparities are estimated with the search range from -10 to +10. The effects of self-adaptive search are investigated with respect to computational costs, speed (in *fps*), gain and accuracy. The computational costs, speed and gain performance results of adaptive method has been compared with previous fastest literatures [14, 24, 25] and the current state-of-the-arts methods [29-32]. Fast Area Based algorithm [14] reports 3229 μ s required by Intel Core i3, 2.3 GHz processor with 4 GB DDR3 RAM for 384 \times 288 image resolution of Tsukuba head. This experiment also results 310 *fps* and 16 disparity levels. Hierarchical Disparity algorithm [24] reports 4243 μ s required by Intel Core i3, 2.3 GHz processor with 4 GB DDR3 RAM for 384 \times 288 image resolution of Tsukuba head. This experiment also results 235 *fps* and 88 disparity levels. Fast algorithm [25] reports 4617 μ s required by Intel Core i3, 2.3 GHz processor with 4 GB DDR3 RAM for 384 \times 288 image resolution of Tsukuba head. This experiment also results 216 *fps* and 11 disparity levels. But the proposed SAA algorithm requires only 1872 μ s on the same hardware with minimum 66 iterations instead of 121 and 242 iterations as mentioned in Section-V. The proposed method also performs the better speed compared to previous popular methods [14, 24, 25]. It shows the highest speed 535 *fps* while

reporting 11 distinct disparity levels. It is mentioned that the ground truth image of Tsukuba pair contains only 7 disparity levels. Table I illustrates the summary of comparisons among the proposed method and existing state-of-the-arts methods.

The SAA's experimental results have been also compared with the result of methods those are tested on Middlebury standard data set. The ranking results in Table I indicate that the proposed SAA method is ranked 1st out of existing top state-of-the-arts methods [29-40]. It shows the highest speed 535 *fps* and lowest computational time 1872 microseconds among the latest methods with lower configuration of machine. So the proposed method outperforms all the state-of-the-arts methods in speed and computational time on Middlebury standard Tsukuba head image pair.

So it is claimed that the proposed method is currently the state-of-the-arts method for Middlebury standard Tsukuba head image pair with 1.7X, 2.2X, 2.4X, 106X, 4.8X, 3.7X, 1602X,

1527X, 293X faster than the methods of [14], [24], [25], [29], [30], [31], [32], [39] and [40], respectively. The proposed method performs 59.45% time reduction compare to window-based method by a fast algorithm [25]. From Table II, we find that a reduction of 55.88% computation time comparing with the proposed SAA method and Hierarchical Disparity Estimation [24]. The SAA method also achieves the reduction of 42.02% computation time comparing with fast Area Based Stereo Matching Algorithm [14].

Finally the SAA method achieves 99.06%, 79.20%, 73.25% and 99.93% computational time reduction against the recent state-of-the-arts methods of [29], [30], [31], [32] and [39], respectively using the lower configurations of hardware. The graphical comparison of computational costs and speed (in *fps*) are illustrated in Fig. 15. So from the Table I, Table II and graph of Fig. 15, the proposed SAA algorithm is the better choice on the basis of trade-off between computational time and speed.

TABLE I. DISPARITY ESTIMATION COMPUTATIONAL TIME (IN μ S) AND SPEED (IN FPS) FOR THE MIDDLEBURY STANDARD DATA OF TSUKUBA HEAD IMAGE USING SELF-ADAPTIVE SEARCH METHOD

Methods' Name	Computational Time(μ s)	Speed (<i>fps</i>)	Computational Machine	Input image & Resolution	Rank
Self-Adaptive Algorithm [Proposed]	1872	535	Intel Core i3 Speed: 2.3 GHz. RAM: 4GB	Middlebury Standard Tsukuba Head 384x288	1
Fast Area Based [14]	3229	310	Intel Core i3 Speed: 2.3 GHz. RAM: 4GB		2
Hierarchical Disparity [24]	4243	235	Intel Core i3 Speed: 2.3 GHz. RAM: 4GB		3
Fast Algorithm [25]	4617	216	Intel Core i3 Speed: 2.3 GHz. RAM: 4GB		4
Tree filtering [31]	7000 (7ms)	143	Intel Core-i7 Speed: 1.8 GHz. RAM: 4 GB		5
Edge-aware & Geodesic filter[30]	9000 (9ms)	111	Intel Core-i5+Geforce GTX card Speed: 3.0GHz. RAM: 8GB		6
DSI & Adaptive Support[29]	200000 (0.2s)	5	Core Duo Speed: 2.2GHz. RAM:NA		7
Pyramid stereo matching [40]	550000(0.55s)	2	Nvidia GeForce GTX 1080 Ti/PCIe/SSE2	KITTI [2012] Data set.	8
Deep self-guided[39]	2860000(2.86s)	0.35	Intel Core- i7-4770 Speed:3.4GHz. RAM.16GB RAM: 16 GB	Middlebury training data set and KITTI [2015]	9
Energy Minimization[32]	3000000 (3s)	0.33	Intel Core- i5 Speed: 1.9 GHz. RAM:6GB	Middlebury Standard data set	10
Fusing Adaptive Support[38]	40500000(40.5s)	0.025	Intel Core- i5-6500 Speed: 3.2 GHz. RAM: 8GB	Tsukuba, Venus, Teddy, Cones [Middlebury Benchmark]	11

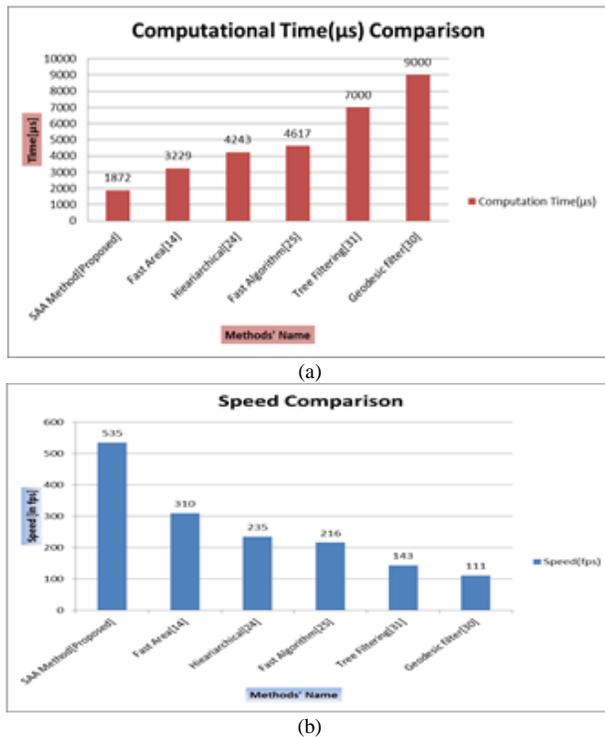


Fig. 15. Upper Side Graph(a) shows the Comparison of Computational Costs and Lower Side Graph(b) shows the Comparison of Speed (in fps) among Proposed and existing State-of-the-Art Methods for Tsukuba Head Image.

C. Experiment 3: Computational Cost Calculation and Comparison with Existing State-of-the-Arts Methods for Middlebury Standard Venus Stereo Image

The computational time and speed performances have been numerically evaluated in Table III for Middlebury standard data of Venus stereo pair. The proposed SAA algorithm also outperforms all other algorithms summarized in Table III. Table III illustrates the comparison of computational costs, speed (fps) of proposed method and other recent state-of-the-arts methods with hardware specifications. The proposed SAA algorithm also obtains the better performance in computational time and speed. It requires only 2652 μs instead of 6318 μs , 6724 μs , 7000 μs , 7473 μs and 9000 μs respectively but with a higher potential speed of 377 fps. The graphical comparisons of time and speed are illustrated in Fig. 16 which depicts the

proposed SAA method runs in lowest time and achieves the highest speed among the current state-of-the-arts methods. Another numerical evaluation and comparisons are depicted in Table IV in which proposed method performs 64.51% time reduction compared to window-based method by a fast algorithm [25] for Venus stereo pair image. From Table IV 60.55% computational time is reduced by proposed SAA method compared to the Hierarchical Disparity [24]. The algorithm also performs the reduction of 58.02% computational time comparing with the Fast Area based method [14].

So it is evident that the proposed method is currently the best state-of-the-arts method for Middlebury standard Venus stereo pair with 2.3X, 2.5X, 2.8X, 3.3X, 2.6X faster than the top five methods of [14], [24], [25], [30] and [31], respectively.

Finally the SAA method achieves 70.53% and 62.11% computational time reduction against the recent state-of-the-arts methods of [30] and [31] respectively using the lower configurations of hardware. So from the Table IV and graph of Fig. 16, the proposed SAA algorithm again proves that it is the best choice on the basis of trade-off between computational time and speed. In both cases (Tsukuba and Venus stereo pair) the SAA algorithm performs the lowest computational costs and highest speed.

D. Experiment 4: Accuracy Measurement and Comparisons with Three Established Methods

The accuracy of this algorithm has been justified over standard stereo images of Tsukuba head. Experiments are performed on Intel Core i3, 2.3 GHz processor with 4 GB DDR3 RAM. The algorithm has been implemented using Visual C++ programming language. To determine the correspondence of a pixel of reference image, the window costs are estimated for the candidates' pixels of right image within the search range -10 to +10 pixels. The proposed SAA method estimates the accuracy in percentage with the error threshold 1. The accuracy of SAA method is 93.8%. The numerical evaluations confirm that the bad pixel in percentage is only 6.2% with error threshold 1 for proposed method. But using the same resolution of image, bad pixels in percentage were 6.33%, 7.88% , and 7.18% reported in [30], [31] and [32] respectively for Tsukuba head with the experiments of Middlebury stereo data set.

TABLE II. COMPUTATIONAL TIME REDUCTION (IN %) OF PROPOSED METHOD FOR TSUKUBA HEAD

Computational Time(μs) for Self-Adaptive Algorithm [Proposed]	Existing state-of-the-art methods		Computational Time Reduction (%) by SAA method compared to the methods of 2 nd column
	Methods' Name	Computational Time(μs)	
1872	Fast Area Based [14]	3229	42.02
	Hierarchical Disparity [24]	4243	55.88
	Fast Algorithm[25]	4617	59.45
	Tree filtering [31]	7000 (7ms)	73.25
	Edge-aware & Geodesic filter[30]	9000 (9ms)	79.20
	DSI & Adaptive Support[29]	200000 (0.2s)	99.06
	Energy Minimization[32]	3000000 (3s)	99.93
	Pyramid stereo matching[40]	550000(0.55s)	99.65
	Deep self-guided[39]	2860000(2.86s)	99.93

TABLE III. DISPARITY ESTIMATION COMPUTATIONAL TIME (IN μ S) AND SPEED (IN FPS) FOR THE VENUS STEREO IMAGE USING SELF -ADAPTIVE SEARCH METHOD

Methods' Name	Computing Time(μ s)	Speed (fps)	Computing Machine	Input image & Resolution	Rank
Self-Adaptive Algorithm [Proposed]	2652	377	Intel Core i3 Speed: 2.3 GHz. RAM: 4GB	Middle -bury Standard Venus Stereo data set 434x383	1
Fast Area Based [14]	6318	158	Intel Core i3 Speed: 2.3 GHz. RAM: 4GB		2
Hierarchical Disparity [24]	6724	148	Intel Core i3 Speed: 2.3 GHz. RAM: 4GB		3
Tree filtering [31]	7000 (7ms)	143	Intel Core-i7 Speed: 1.8 GHz. RAM:4GB		4
Fast Algorithm [25]	7473	133	Intel Core i3 Speed: 2.3 GHz. RAM: 4GB		5
Edge-aware & Geodesic filter[30]	9000 (9ms)	111	Intel Core-i5+Geforce GTX card Speed: 3.0Ghz RAM: 8GB		6

TABLE IV. COMPUTATIONAL TIME REDUCTION (IN %) OF PROPOSED METHOD FOR VENUS STEREO PAIR

Computational Time(μ s) for Self-Adaptive Algorithm [Proposed]	Existing state-of-the-art methods		Computational Time Reduction (%) by SAA method compared to the methods of 2 nd column
	Methods' Name	Computational Time(μ s)	
2652	Fast Area Based [14]	6318	58.02
	Hierarchical Disparity [24]	6724	60.55
	Tree filtering [31]	7000 (7ms)	62.11
	Fast Algorithm[25]	7473	64.51
	Edge-aware & Geodesic filter[30]	9000 (9ms)	70.53

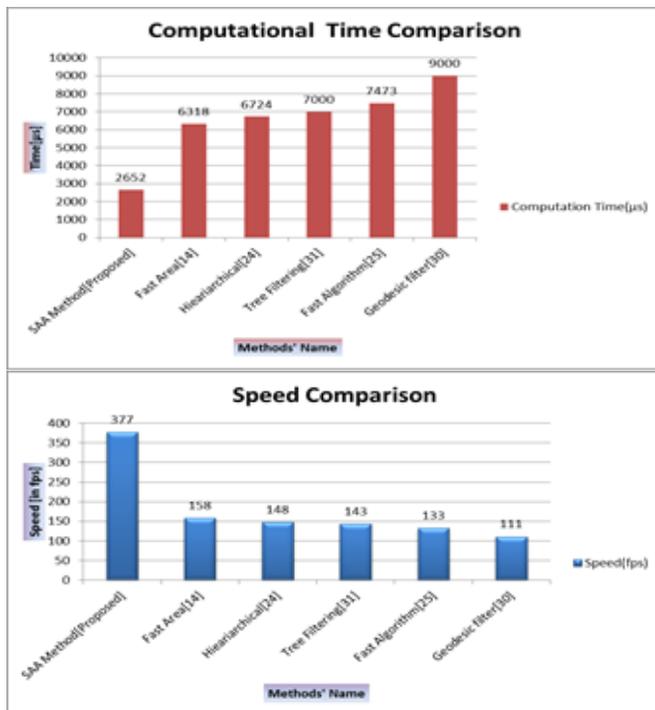


Fig. 16. Upper Graph shows the Comparison of Computational Costs and Lower Graph shows the Comparison of Speed (in fps) among Proposed and Existing State-of-the-Arts Methods for Venus Stereo Images.

VII. PERFORMANCE ENHANCEMENT ANALYSIS

The performance of the proposed SAA approach has been compared to the state-of-the-arts methods. The comparison tools are computational time, speed and gain. Our target was to speed up computational costs with no degradation of accuracy. Since the state-of-the-arts algorithms yield very similar accuracies and 3D dense disparity maps confirm the effectiveness of the proposed matching algorithm. Table V illustrates the performance enhancement of SAA method compared to the established state-of-the-art methods. Proposed SAA method shows 72% gain enhancement compared to Fast Area Based Method. Enhanced gain is calculated as follows.

Gain Enhancement by proposed SAA method against Fast Area Based Method [14]:

$$= \{(\text{Computational time of Fast Area Based Method} \div \text{Computational time of SAA Method}) \times 100 - 100\} \%$$

$$= \{(3229/1872) \times 100 - 100\} \%$$

$$= 72\%$$

Similarly the gain enhanced by proposed SAA method against the state-of-the-art methods in [24], [25], [30] and [31] is 126%, 146%, 380% and 273% respectively which is illustrated in Fig. 17. Fig. 17 shows the performance enhanced graph of proposed SAA method. Since the speed of proposed method is very high, the estimated gains of this algorithm are automatically high. The numerical measurements of this claim is

strongly supported by experimental data of Table V in which the SAA algorithm is compared with the top five (5) algorithms. The bar diagrams illustrate the increased gain (in %) of SAA method compared to the current methods. The proposed method outperforms the existing state-of-the-art methods with respect to the performance enhancements.

Accordingly, the performance of the SAA method has also been tested on Middlebury standard Venus stereo data set and the estimated experimental data are summarized in Table VI. From this table the proposed SAA method also performs better than the existing top five algorithms for Venus stereo pair data set.

The gains have been estimated following the same procedures as mentioned earlier and the SAA method enhances 138%, 153%, 181%, 239%, 163% gains compared to established state-of-the-art methods of [14], [24], [25], [30] and [31], respectively.

The SAA method enhances 239% gain over Geodesic filter method [30], but it actually increases $(239-100)= 139\%$ gain against Geodesic filter method [30]. Similarly, the proposed

method enhances 163% gain over Tree Filtering [31] method and it actually increases $(163-100) = 63\%$ gain against Tree Filtering [31] method. So in the case of Venus data, the proposed SAA algorithm also outperforms the existing top five algorithms. In terms of computational time, speed and gain achievement, the proposed matching approach is faster and better than the existing state-of-the-art methods.

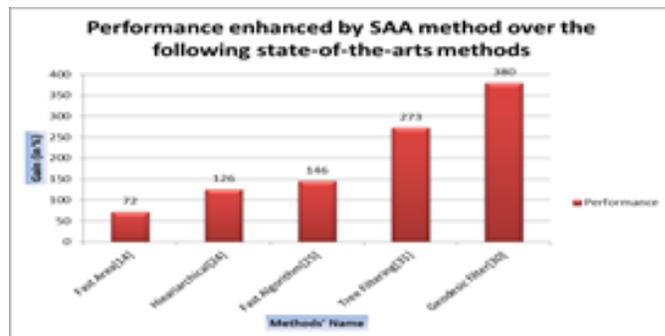


Fig. 17. Graph shows the Performance Enhancement of SAA Method Comparing to Existing State-of-the-Art Methods.

TABLE V. QUANTITATIVE EVALUATION OF PERFORMANCE OF PROPOSED SAA METHOD WITH TOP FIVE (5) ALGORITHMS

Methods' Name	Computational Time(μ s)	Speed (fps)	Performance Enhanced (%) by SAA method compared to the methods of first column	Input image & Resolution
Self-Adaptive Algorithm [Proposed]	1872	535	x	Middle-bury Standard Tsukuba Head 384x288
Fast Area Based [14]	3229	310	72	
Hierarchical Disparity [24]	4243	235	126	
Fast Algorithm[25]	4617	216	146	
Tree filtering [31]	7000 (7ms)	143	273	
Edge-aware & Geodesic filter[30]	9000 (9ms)	111	380	

TABLE VI. QUANTITATIVE EVALUATION OF PERFORMANCE OF PROPOSED SAA METHOD WITH TOP FIVE (5) ALGORITHMS FOR VENUS STEREO PAIR

Methods' Name	Computational Time(μ s)	Speed (fps)	Performance enhanced (in %) by SAA method compare to the methods of first column	Input image & Resolution
Self-Adaptive Algorithm [Proposed]	2652	377	x	Middlebury Standard Venus Stereo data set. 434x383
Fast Area Based [14]	6318	158	138	
Hierarchical Disparity [24]	6724	148	153	
Tree filtering [31]	7000 (7ms)	143	163	
Fast Algorithm[25]	7473	133	181	
Edge-aware & Geodesic filter[30]	9000 (9ms)	111	239	

VIII. CONCLUSION

The main objective of this paper is to speed up the computational time. We have done this by a new technique called Self-Adaptive Algorithm that infers the upcoming matching pixel's position. This algorithm itself readjusts as well as reduces the search range of candidate's pixels based on the previous matched pixel's position. The speed of our algorithm is 535 fps for input images of Tsukuba head pair and 377 fps for input images of Venus stereo pair. Thus it can scan, calculate, process and display output 535 frame/second for the case of standard Tsukuba head image pair and 377 frame/second for the standard Venus stereo pair. The estimated gains of proposed SAA method are 380 and 239 for Tsukuba head and Venus stereo respectively whereas the gain of existing state-of-the-arts- methods is 100. The estimated accuracy is almost same as the accuracies of previous methods. Since the accuracy and 3D dense disparity maps of proposed method are very similar with the existing state-of-the-art algorithms, it confirms the effectiveness of the proposed matching algorithm. Moreover, the SAA algorithm does not require additional programmable 3D hardware like 3D Graphics Processing Unit (GPU). The proposed SAA method demonstrates the state-of-the-arts results and outdoes the existing top methods. In future we will consider fuzzy based technique to extend the research works for better realization of its behaviors.

REFERENCES

- [1] S. T. Barnard and M. A. Fischler, "Stereo vision," in Encyclopedia of Artificial Intelligence, pp. 1083-1090. John Wiley, New York, 1987.
- [2] W. Hoff and N. Ahuja, "Surfaces from stereo: Integrating feature matching, stereo correspondence estimation and contour detection," IEEE Trans. Pattern Anal. Machine Intell., vol. 11, no. 2, pp.121-136,1989.
- [3] T. Kanade and M. Okutomi, "A stereo matching algorithm with an adaptive window: Theory and experiment," IEEE Trans. Pattern Anal. Machine Intell., vol. 16, no. 9, 1994.
- [4] Olga Veksler, "Stereo matching by compact windows via minimum ratio cycle," Proceedings of the IEEE International Conference on Computer Vision (ICCV),2001, pp. 540-547.
- [5] Md.Abdul Mannan Mondal, Md.Al-Amin Bhuiyan "Disparity Estimation By A Two-Stage Approximation Real Time Algorithm" Proceedings of the International Management and Technology Conference (IMT), 2004, pp. 12-17.
- [6] S. S. Intille and A. F. Bobick, "Stereo correspondence-space images and large occlusion stereo," Proceedings of the European Conference on Computer Vision (ECCV), 1994, pp. 179-186.
- [7] A. Fusiello and V. Roberto, "Efficient stereo with multiple windowing," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (ICVPR),1997, pp. 858-863.
- [8] K. Muhlmann, D. Maier, J. Hesser, R. Manner, "Calculating dense stereo correspondence maps from color stereo images, an efficient implementation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (ICVPR), 2001, pp. 30-36.
- [9] Md.Abdul Mannan Mondal, Md. Mustafa Kamal, Md. Al- Amin Bhuiyan,"Stereo correspondence Estimation By Real Time Approximation Algorithm Using Virtual Masking System.," Proceeding of 8th International Conference on Pattern Recognition and Information Processing (PRIP'05),2005.
- [10] Md. Mozammel Hoque Chowdhury, Md.Abdul Mannan Mondal, Md.Al-Amin Bhuiyan, "3D Imaging with stereo vision", Proceeding of 7th International Conference on Computer & Information Technology(ICCIT), 2004, pp. 307-312.
- [11] Md.Abdul Mannan Mondal, Md. Mustafa Kamal, Md.Ariful Hyder Md.Al-Amin Bhuiyan, "Stereo Correspondence Estimation by Diagonal Mask Searching ", Proceeding of 8th International Conference on Computer & Information Technology, (ICCIT), 2005, pp. 379-383.
- [12] Md.Abdul Mannan Mondal, Md. Mustafa Kamal, , Md. Mozammel Hoque Chowdhury , Md.Ariful Hyder ,A.K.M Zaidi Satter ,Md.Al-Amin Bhuiyan, "Stereo Correspondence Estimation: A Two-Stage Approximation Algorithm Using Arbitrary Window Pixel ", Proceeding of 8th International Conference on Computer & Information Technology, (ICCIT), 2005, pp. 384-387.
- [13] D.Schartein ,R.Szeliski, " A taxonomy and evaluation of dense two frame stereo correspondence algorithms", Int. J. of Comp. Vission,Vol. 47, no.1-3,pp.7-42,2002.
- [14] Lugi Di Stefano,Massimiliano Marchionni, Stefano Mattocchia, "A fast Area Based Stereo Matching Algorithm" , Image and vision Computing 22,pp.983-1005,2004.
- [15] Elisabetta Binaghi, Ignazio Gallo, Giuseppe Marino, Mario Raspanti, "Neural adaptive stereo matching", Pattern Recognition Letters, Vol.25,pp. 1743-1758,2004.
- [16] Abijit S.Ogale and Yiannis Aloimonos , "Shape and the Stereo Correspondence Problem", Int. J. of Comp.Vission, Vol.65,no.3,pp.147-167,2005.
- [17] Sukjune Yoon,Sung-Kee Park, Sungehul Kang, Yoon Keun Kwak, "Fast correlation-based stereo matching with the reduction of systematic errors", Pattern Recognition Letters,Vol. 26, pp. 2221- 2231,2005.
- [18] L. Kotoulas, A. Gasteratos,G.Ch.Sirakoulis , C. Georoulas, and I. Andreadis, "Enhancement of Fast Acquired Disparity Maps using a 1-D Cellular Automation Filter", proceedings of the fifth IASTED International Conference on Visualization, Imaging and Image Processing, 2005, pp.7-9.
- [19] Kuk-Jin Yoon and In So Kweon, "Adaptive Support -Weight Approach for Correspondence Search", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.28,no. 4, pp.650-656, April 2006.
- [20] P.H.S. Torra, A. Criminisi, "Dense stereo using pivoted dynamic programming", Image and Vision Computing Vol.22, pp. 795-806, 2004.
- [21] Lazaros Nalpantidis,Georgios Ch. Sirakoulis and Antonios Gasteratos, "Rievview of stereo matching algorithms for 3D vision", 16th International Symposium on Measurement and Control in Robotics ISMCR, 2007, pp.116-124.
- [22] Salvador Gutierrez,Jose Luis Marroquin, "Robust approach for disparity estimation in stereo vision", Image and Vision Computing, Vol.22, pp. 183-195, 2004.
- [23] Michael Bleyer, Margrit Gelautz, " A layered stereo matching algorithm using image segmentation and global visibility constraints", ISPRS Journal of Photogrammetry and Remote Sensing Vol.59,pp.128-150,2005.
- [24] Christopher Zach,Konrad Karner,Horst Bischof,"Hierarchical Disparity Estimation with Programmable 3D Hardware " Proc. of Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision, 2004, pp.275-282.
- [25] Mohammad Shorif Uddin "Stereo correspondence estimation using window-based methods by a fast algorithm", Journal of Electronics and Computer Science,vol.4,pp.5-11,2003.
- [26] D. Min, J. Lu, and M. N. Do, "A revisit to cost aggregation in stereo matching: How far can we reduce its computational redundancy?" Proc.of IEEE ICCV, Nov. 2011. pp. 1567-1574.
- [27] D.Min, J. Lu, and M. N. Do, "Joint histogram-based cost aggregation for stereo matching," IEEE Trans. Pattern Anal. Mach. Intell., vol.35, no. 10, pp. 2539-2545, Oct. 2013.
- [28] A. Hosni, C.Rhemann, M. Bleyer, C. Rother, and M. Gelautz. Fast cost volume filtering for visual correspondence and beyond . PAMI,35,pp.504-511, 2013.
- [29] Nadia Baha & Slimane Larabi "Accurate real-time disparity map computation based on variable support window " International Journal of Artificial Intelligence & Applications (IJAIA), Vol.2, No.3, pp. 22-33, July 2011.

- [30] Xun Sun , Xing Mei , Shaohui Jiao , Mingcai Zhou , Zhihua Liu , Haitao Wang “Real-time local stereo via edge-aware disparity propagation” *Pattern Recognition Letters* , vol. 49,pp. 201-206, 2014.
- [31] Q. Yang, “Stereo matching using tree filtering” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 4,pp. 834-846, 2015.
- [32] Mikhail G. Mozerov & van de Weijer, “Accurate stereo matching by two-step energy minimization”, *IEEE Transactions on Image Processing*,vol. 24,no. 3,pp. 1153-1163, March 2015.
- [33] Zbontar, J., & LeCun, Y. “ Computing the stereo matching cost with a convolutional neural network”, *Proceedings of the IEEE conference on computer vision and pattern recognition*, October 2015, pp.1592-1599.
- [34] Zbontar, J., & LeCun, Y. “Stereo matching by training a convolutional neural network to compare image patches”. *Journal of Machine Learning Research*, vol.17, no.2, pp. 1-32, April 2016.
- [35] Md. Abdul Mannan Mondal & Md. Haider Ali, “Disparity Estimation by a Real Time Approximation Algorithm” *International Journal of Image Processing (IJIP)*, Vol. 10 : Issue -3, pp.126-134, 2016.
- [36] Md. Abdul Mannan Mondal & Md. Haider Ali, “Performance Review of the Stereo Matching Algorithms” *American Journal of Computer Science and Information Engineering*,vol. 4,no.1,pp- 7-15, 2017.
- [37] D. Scharstein, R. Szeliski, *Middlebury stereo evaluation-version 2*, 2010, <http://vision.middlebury.edu/stereo/eval/>.
- [38] Wenhuan Wu,Hong Zhu,Shunyuan Yu and Jing Shi “ Stereo Matching with Fusing Adaptive Support Weights” *IEEE Access*,vol.7 ,pp. 61960-61974, May 2019.
- [39] Williem , In Kyu Park “Deep self-guided cost aggregation for stereo matching” *Pattern Recognition Letters*,vol. 112,pp. 168-175,September 2018.
- [40] Jia-Ren Chang and Yong-Sheng Chen “Pyramid Stereo Matching Network” *Proceeding of IEEE conference Computer vision and pattern recognition*, June 2018, pp. 5410-5418.
- [41] Siti Safwana Abd Razak, Mohd Azlishah Othman, Ahmad Fauzan Kadmin “The effect of Adaptive Weighted Bilateral Filter on Stereo Matching Algorithm” *International Journal of Engineering and Advanced Technology*, vol. 8, Issues 3, pp.284-287, February 2019.