An Efficient Adaptive Window Based Disparity Map Computation Algorithm by Dense Two Frame Stereo Correspondence

Narendra Kumar Shukla, Vivek Rathi, and Vijaykumar Chakka

Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, India

Abstract. This paper presents an efficient algorithm for disparity map computation with an adaptive window by establishing two frame stereo correspondence. Adaptive window based approach has a clear advantage of producing dense depth maps from stereo images. In recent years there has not been much research on adaptive window based approach due its high complexity and large computation time. Adaptive window based method selects an appropriate rectangular window by evaluating the local variation of the intensity and the disparity. Ideally the window need not be rectangular but to reduce algorithmic complexity and hence computation time, rectangular window is taken. There is a need for correction of errors introduced due to the rectangular window which is not dealt by the existing algorithm. To reduce this error, a method has been proposed which not only improves the disparity maps but also has a lesser computational complexity. To demonstrate the effectiveness of the algorithm the experimental results from synthetic and real image pairs (provided by middlebury research group) including ones with ground-truth values for quantitative comparison with the other methods are presented. The proposed algorithm outperforms most of the existing algorithms evaluated in the taxonomy of dense two frame stereo algorithms. The implementation has been done in C++. The algorithm has been tested with the standard stereo pairs which are used as benchmark for comparison of algorithms in the taxonomy implementation.

1 Introduction

Stereo correspondence for obtaining dense disparity map in two frame stereo is a classical problem in computer vision. Various algorithms have been proposed for the disparity map computation in past whose taxonomy was very well presented by Scharstein and Szeliski [3]. Most of these techniques utilize intensity variation to compute disparity map. The most common amongst them are SSD (Sum of Squared intensity Difference) based, which compute the window with minimum SSD to estimate the disparity. One Common problem these algorithms face is the computation of support region or window size. Each pixel has neighborhood (support region/window) with different intensity and disparity variations. So selecting an efficient window becomes a difficult task. The region enclosed by

P. Kalra and S. Peleg (Eds.): ICVGIP 2006, LNCS 4338, pp. 894–905, 2006.

window must be large enough to include enough intensity variations and small enough to avoid the effect of projective distortion. We need to adaptively compute the window dimensions for each pixel based on the intensity and disparity variations around the pixel.

The algorithms gaining popularity now-a-days are Graph cut based [4]. These are global algorithms which make explicit smoothness assumptions.

2 Adaptive Window Algorithm

M. Okutomi and T. Kanade [1] proposed a method to compute adaptive window for each pixel which iteratively updates window size and disparity estimate in each run. Adaptive Windows can be considered a form of local segmentation, as they divide the image into logical units to be considered separately. Here the logical unit is an image area with enough visual interest for a good match but not too much depth variation. The major problem in computing a locally

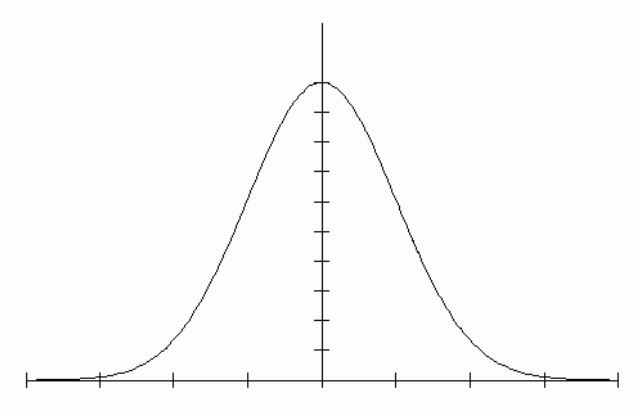


Fig. 1. Gaussian Distribution of Certainty in Disparity Estimation

adaptive window is in computing and using disparity variances. All we can measure directly is intensity variation. Two major algorithms proposed so far are based on rectangular window [1] and arbitrary shaped window [2]. The algorithm with arbitrary shaped window requires a higher computation time in comparison to the former. [1] employed a statistical model of the disparity distribution within the window with the assumption that difference of disparity at a point in the window from the center point $(0, 0)$ has a zero-mean Gaussian distribution with variance proportional to the distance between these points as shown in Figure (1). As we move farther from the center pixel, the uncertainty of disparity estimate as compared to that of center pixel increases. The Disparity estimate and its uncertainty for a given window W can be calculated by:

$$\hat{\Delta d} = \frac{\sum_{i,j \in W} \frac{(f_1(\xi_i,\eta_j) - f_2(\xi_i + d_0(0,0),\eta_j)) \frac{\partial}{\partial \xi} f_2(\xi_i + d_0(0,0),\eta_j)}{2\sigma_n^2 + \alpha_f \alpha_d \sqrt{\xi_i^2 + \eta_j^2}}}{\sum_{i,j \in W} \frac{(\frac{\partial}{\partial \xi} f_2(\xi_i + d_0(0,0),\eta_j))^2}{2\sigma_n^2 + \alpha_f \alpha_d \sqrt{\xi_i^2 + \eta_j^2}}} \tag{1}$$

$$\sigma_{\Delta d}^2 = \frac{1}{\sum_{i,j \in W} \frac{(\frac{\partial}{\partial \xi} f_2(\xi_i + d_0(0,0),\eta_j))^2}{2\sigma_n^2 + \alpha_f \alpha_d \sqrt{\xi_i^2 + \eta_j^2}}} \tag{2}$$

where, $f_1(x, y)$ and $f_2(x, y)$ are the intensity functions of reference and matching image respectively, $d_0(x, y)$ is the initial disparity estimate, σ_n^2 is the power of noise of error per image. The parameters α_f and α_d represents the disparity and intensity fluctuation respectively. We can compute the values of α_f and α_d within the window as:

$$\alpha_d = \frac{1}{N_w} \sum_{i,j \in W} \frac{(d_0(\xi_i, \eta_j) - d_0(0,0))^2}{\sqrt{\xi_i^2 + \eta_j^2}} \tag{3}$$

$$\alpha_f = \frac{1}{N_w} \sum_{i,j \in W} (\frac{\partial}{\partial \xi} f_2(\xi_i + d_0(0,0), \eta_j))^2 \tag{4}$$

where N_w is the number of samples within the window. Therefore given all the required parameters, equations 1 - 4 will enable to calculate a better estimate of disparity at the center of the window as $d_0(x, y) + \Delta d$ with the minimum uncertainty. So we can improve the disparity estimate given initial estimate by minimizing its uncertainty and simultaneously replacing the new disparity estimate by incrementing the current disparity estimate by $\hat{\Delta d}$.

2.1 The Algorithm

Let us go through the algorithmic approach given by [1] for computing better disparity estimates, given the input stereo pair and their initial disparity estimate:

1. Start with an initial disparity estimate $d_0(x, y)$.
2. For each pixel (x, y) in f_1,
   (a) Place a 3 x 3 Window centered at x, y and compute uncertainty by using equation 2.
   (b) Expand the window by one pixel in one direction, e.g., to the right $x+$, for trial, and compute the uncertainty for the expanded window. If the expansion increases the uncertainty, the direction is prohibited from further expansions. Repeat the same process for each of the other three directions $x-, y+$, and $y-$ (excluding the already prohibited ones).
   (c) Compare the uncertainties for all the directions tried and choose the direction which produces the minimum uncertainty.
   (d) Expand the window by one pixel in the chosen direction.

(e) Iterate steps (b) to (d) until all directions become prohibited from expansion or until the window size reaches a limit that is previously set.
(f) Update the disparity $d_0(x, y)$ by adding Δd computed by equation 1 for the modified window.

3. Iterate the above process until the disparity estimate $d(x, y)$ converges, or up to a certain maximum number of iterations.

The algorithm truly justifies its approach. Flat surfaces have very less disparity variation but taking large window may blur the edges. In contrast, a smaller window gives sharper disparity edges at the cost of noisy surfaces. The adaptive window algorithm takes care of flat surfaces as well as sharp edges, however, there are two major problems with the above algorithm:

1. When window size is increased by a row or a column, the new row or column might have some pixels, although in lesser quantity, which increase the uncertainty of the disparity estimate. This causes errors in disparity estimation.
2. The above algorithm requires a lot of computation especially when there is a large area of flatness in the image.

Although researchers have tried to solve the first problem by taking arbitrary shaped (non-rectangular) window, but computation of such windows increases the computation time even more and hence makes the second problem even worse. Also applying the above algorithm till convergence of disparity makes it unsuitable for real-time applications.The next section solves these problems of the adaptive window.

3 The Proposed Efficient Adaptive Window Algorithm

On the basis of major pitfalls identified for the algorithm described above, the new approach is categorized into two parts viz. Reducing the errors in disparity estimate and Reducing the computation time, although the solution to the first one also reduces the computation time. Then the approach is compiled into an algorithm succeeding the two sections. Thereafter the results of the improved algorithm are compared and analyzed with the existing algorithms.

3.1 Reducing the Errors in Disparity Estimate

The errors introduced by rectangular window tend to be large specially when image contains less flat surfaces or more curved surfaces. This is because, when the optimum rectangular window is computed, it may consist of pixels which have less intensity correspondence but other pixels in the row or column nullify its effect leading to errors in disparity estimation. This my introduce error in prediction up to three-four pixels. Therefore, there is a need for correction of this error. A new method to correct the error is proposed.

For each pixel take the optimized adaptive window as computed by the adaptive window algorithm. Now, for this window compute SSD by shifting the

window in the range $-d$ to $+d$ around the disparity predicted by the above algorithm.d may be 0, 1, 2, 3, 4... . Let d_i be the value between $-d$ and $+d$ at which SSD comes out to be minimum. Now update the new value of disparity as:

$$d_r(x, y) = d_r(x, y) + d_i$$

This value is the new value of the disparity of pixel of interest. By shifting the window over the range $-d$ to $+d$ we compute the disparity which further reduces the error as by doing the above process we compute the best possible match of the pixel in the reference image with the matching image within the neighborhood of the disparity estimate. Thus the disparity estimate computation by this algorithm will converge faster as compared to adaptive window algorithm of [1]. Hence it also reduces the computation time and increases convergence rate. The next section deals with reducing the computation time of the optimum window for the pixels in image.

3.2 Reducing the Computation Time

In the adaptive algorithm of [1] we take the initial window of size 3 x 3 and increase the size of the window in the direction of minimum uncertainty. And then for the next pixel we again start from 3 x 3. By the property of disparity smoothness for most of the region in the image, we can start with the window size averaged over the surrounding left, upper-left and diagonal-left, diagonal-right, i.e., the window size of surrounding pixels which are already computed.

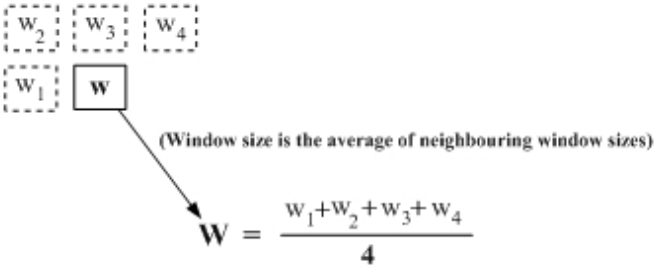


Fig. 2. Initial window estimate: Direction-wise Average of already computed neighboring window sizes

Now taking this window size contract the window by one pixel in each direction and check if the uncertainty decreases for any direction. If for a direction the uncertainty $\sigma^2_{\Delta d}$ decreases then from next iteration start contracting in that direction(i.e. reduce the window size in that direction) otherwise start expanding in that direction. This approach applies to all the directions. After this, if at a particular iteration, the uncertainty does not decrease, then that direction is prohibited from further expansion/contraction.

3.3 The Proposed Algorithm

Given below is the complete algorithm for the improved adaptive window approach described in the previous two sections:

1. Start with an initial disparity estimate $d_0(x, y)$.
2. Make label for all the pixel as 'NW'.
3. For each pixel (x, y) in f_1,
   (a) Place a Window centered at x, y of dimensions average of windows of all the neighboring pixels having label 'W' and compute uncertainty by using equation 2. If all the neighboring pixels have label NW then take the initial window size as 3 x 3, label all the edges of this window as 'E' and skip the next step.
   (b) Contract the one of window edge by one pixel and compute the uncertainty. If it decreases the uncertainty then label the edge as 'C' else label the edge as 'E'. Repeat the process for all the edges.
   (c) Expand/Contract the edge by one pixel (Depending on whether label associated with it is 'E' or 'C') in one direction, e.g., to the right $x+$, for trial, and compute the uncertainty for the expanded/contracted window. If the expansion/contraction increases the uncertainty, the direction is prohibited from further expansions and label the edge as 'P'. Repeat the same process for each of the other three directions $x-, y+$, and $y-$ (excluding the edges with 'P' label).
   (d) Compare the uncertainties for all the directions tried and choose the direction which produces the minimum uncertainty.
   (e) Expand/Contract the window by one pixel in the chosen direction.
   (f) Iterate steps (c) to (e) until all the edges become prohibited with label 'P' or until the window size reaches a limit that is previously set.
   (g) Store the final window size and label the pixel as 'W'.
   (h) Update the disparity $d_0(x, y)$ by adding Δd computed by equation 1 for the modified window.
   (i) Compute the SSD for the disparity set $\{d_0(x, y) - d, d_0(x, y) + d\}$ (d is the maximum disparity error to be rectified) and update $d_0(x, y)$ with the disparity with minimum SSD.
4. Iterate the above process until the disparity estimate $d(x, y)$ converges, or up to a certain maximum number of iterations.

The above algorithm computes the new disparity map in much less time. This is because, in general, the window sizes comes out to be nearly same to the neighboring window sizes. Also as in each step, the disparity is also improved by computing disparity corresponding to minimum SSD, the algorithm converges in less number of iterations. The results which top each category are shown in bold face. This algorithm converges faster for the images with large planar or textured surfaces. The results of the proposed algorithm are examined in the next section.

4 Experiment and Results

The evaluation criteria used for the experiments is based on the propositions given in [3]. B_O stands for occluded pixels, $B_{\bar{O}}$ stands for non occluded pixels, B_T stands for textured pixels, $B_{\bar{T}}$ stands for non-textured pixels and B_D stands for pixels at discontinuity. For complete description of evaluation criteria, refer [3]. In all of the experiments, the window size threshold has been kept as 16, due to increased time complexity. The value of d (Range of disparity for correction) is taken as 6 for all the results. For a complete set of input images, please refer [5]. The section proceeds with the results of the algorithms on images followed by the comparison with other existing algorithms.

4.1 Improvement in Results

Results with Tsukuba Image: Tsukuba image contains non-planar surfaces with occlusions. So the window sizes computed comes out to be lower. Figure 3 show the ground truth image, disparity image computed with SSD 9x9 window, disparity image of adaptive window algorithm using initial disparity estimate of the ground truth image, and disparity image of proposed algorithm. Clearly the proposed algorithm outperforms the other two algorithms as clear with the data shown in the Table 1 and 2.

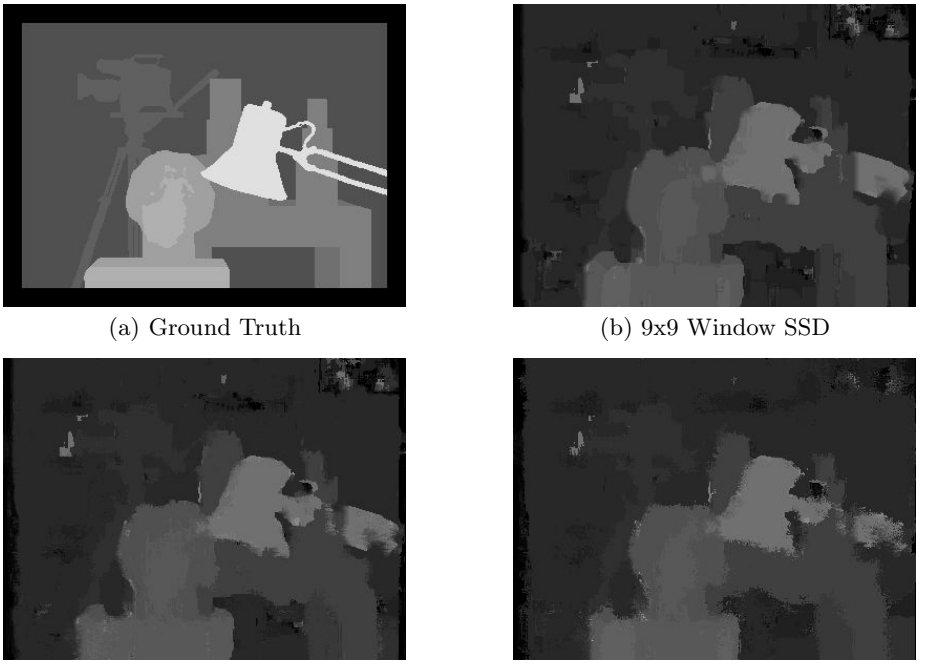
The result shown in Table 1 and 2 are tested with value of d = 3 and only two iterations are performed. Obviously, the proposed improved adaptive algorithm will converge faster than the existing adaptive window algorithm. Clearly, the edges of the objects have sharpened and errors textured and non-textured have

Table 1. Root mean square error comparison of simple SSD 9x9 window algorithm, adaptive window algorithm and the Proposed Adaptive Window Algorithm of tsukuba image based on several parameters.The results which top each category are shown in bold face.

Algorithm	Iterations	B_O	$B_{\bar{O}}$	B_T	$B_{\bar{T}}$	B_D	All
SSD 9x9 Window	-	5.06	1.67	1.72	1.60	3.27	1.84
Adaptive Window	2	5.06	1.60	1.69	1.46	3.18	1.77
Proposed Adaptive Window	**2**	**4.96**	**1.43**	**1.57**	**1.21**	**2.96**	**1.62**

Table 2. Bad pixel percentage (with disparity error greater than 1 pixel) comparison of simple SSD 9x9 window algorithm, adaptive window algorithm and the Proposed Adaptive Window Algorithm of tsukuba image based on several parameters. The results which top each category are shown in bold face.

Algorithm	Iterations	B_O	$B_{\bar{O}}$	B_T	$B_{\bar{T}}$	B_D	All
SSD 9x9 Window	-	87.95%	9.88%	9.55%	10.33%	37.25%	11.89%
Adaptive Window	2	88.49%	9.12%	8.59%	9.85%	34.56%	11.17%
Proposed Algorithm	**2**	**83.53%**	**7.25%**	**7.15%**	**7.38%**	**30.84%**	**9.21%**



(a) Ground Truth (b) 9x9 Window SSD

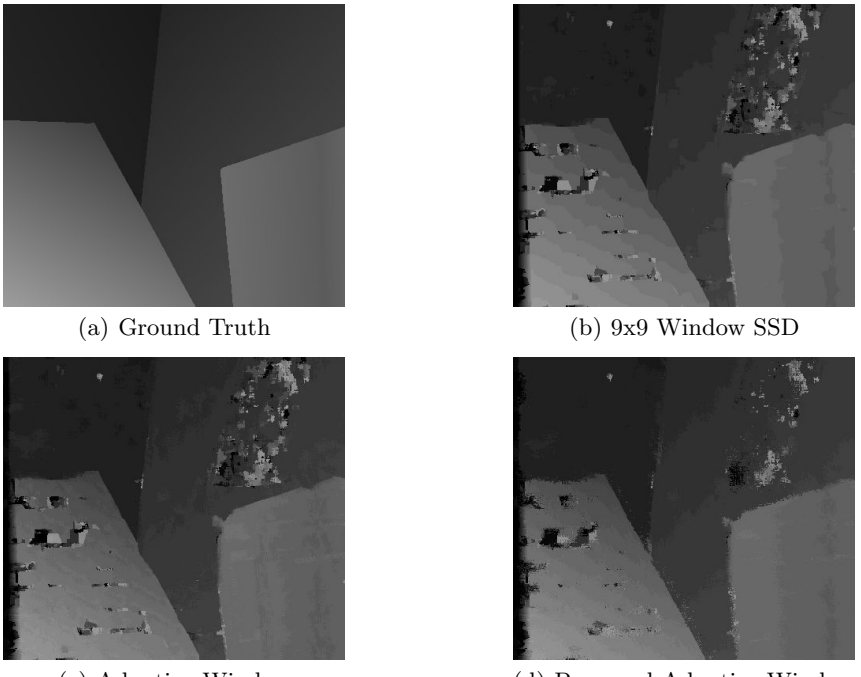
(c) Adaptive Window (d) Proposed Adaptive Window

Fig. 3. Result of image tsukuba with Proposed Adaptive window Algorithm (2 Iterations)

improved a lot. This is because the errors are expected to be in the vicinity of the current estimation. So even if correct result is not predicted in the first iteration, it is likely to converge in the second iteration. Further the bad pixel percentage have decreased in all the areas. If the initial estimate was better, the new estimate could have been much better.

Results with Venus Image: Venus image contains planar surfaces with occlusions. It has 5 planes, some slant, untextured regions and one crease. Figure 4 show the ground truth image, disparity image computed with SSD 9 x 9 window, disparity image of adaptive window algorithm using initial disparity estimate of the ground truth image, and disparity image of the proposed algorithm.

The result shown in Table 3 and 4 are tested with value of d = 3 and only two iterations are performed. Although, with this image, the proposed algorithm does not perform so well in the occluded regions, but it performs considerably well in textured as well as non textured region. As we can see from Figure 4(a), the algorithm has improve the results considerably in the top right region of the image, which is textured region. Obviously the proposed algorithm outperforms the other two algorithms, as it is clear with the data shown in the succeeding tables.



(a) Ground Truth (b) 9x9 Window SSD

(c) Adaptive Window (d) Proposed Adaptive Window

Fig. 4. Result of image venus with Proposed Adaptive window Algorithm (1 Iteration)

Table 3. Root mean square error comparison of simple SSD 9x9 window algorithm, adaptive window algorithm and the Proposed Adaptive Window Algorithm of Venus image based on several parameters.The results which top each category are shown in bold face.

Algorithm	Iterations	B_O	$B_{\bar{O}}$	B_T	$B_{\bar{T}}$	B_D	All
SSD 9x9 Window	-	6.81	2.31	1.65	3.27	1.70	2.47
Adaptive Window	1	7.73	2.01	1.43	2.84	3.01	2.25
Proposed Adaptive Window	**1**	**8.18**	**1.71**	**1.27**	**2.36**	**2.21**	**2.02**

4.2 Time Complexity Analysis

As proposed in section 3.2, the time analysis has been done for the four image sets viz. Map, Sawtooth, Tsukuba, Venus. Table 5 shows the computation time taken for the four images including the initial disparity estimation time. The experiments have been performed on 2.6 GHz, Pentium 4 computer running windows XP operating system.

Table 5 clearly shows the large reduction in computation time. Note that it does not include SSD optimization as proposed in section 3.1, because at each iteration SSD has its own computation time at the same time helps the results

Table 4. Bad pixel percentage (with disparity error greater than 1 pixel) comparison of simple SSD 9x9 window algorithm, adaptive window algorithm and the Proposed Adaptive Window Algorithm of Venus image based on several parameters. The results which top each category are shown in bold face.

Algorithm	Iterations	B_O	$B_{\bar{O}}$	B_T	$B_{\bar{T}}$	B_D	All
SSD 9x9 Window	-	85.95%	14.11%	6.96%	28.70%	10.24%	15.43%
Adaptive Window	1	93.75%	11.13%	6.35%	20.88%	40.68%	12.65%
Proposed Algorithm	**1**	**92.42%**	**7.36%**	**4.54%**	**13.12%**	**24.44%**	**8.93%**

Table 5. Time Taken to compute disparity estimate without ssd optimization including initial disparity estimation time

Image	Computation Time of existing algorithm (in secs.)	Computation Time of Proposed algorithm (in secs.)
Map	86.387	13.343
Sawtooth	338.25	51.248
Tsukuba	229.474	21.234
Venus	345.312	52.232

converge faster then existing adaptive window algorithm. So SSD reduces overall time but increases the iteration time. As iteration time and overall time are not related in the direct way, therefore analysis has been done without SSD optimization. Also note that the window size has been limited to 16 x 16 so the optimization gives errors at some point. It is intended to improve this in the future. Although the time has considerably reduced with the proposed approach, further optimizations are required to make the algorithm work for real-time applications.

4.3 Comparison with Other Algorithms

An important feature of adaptive window algorithm proposed by [1] is that it is completely local and does not include any global optimization. Also, the algorithm does not use any post-processing smoothing, but smooth surfaces are recovered as smooth while sharp disparity edges are retained. Therefore, it performs better in most of the region than existing algorithms. A comparison of proposed algorithm has been done with the other algorithms based on the results given in [3]. The parameters used are $B_{\bar{O}}$ (RMS Error in non occlude pixels), $B_{\bar{T}}$ (RMS Error for non textured pixels) and B_D (RMS error at discontinuity).The proposed algorithm with reduced computation time has not been compared with existing algorithm due to unavailability of computation results of other algorithms. Figure 5 show the comparison of proposed algorithm with other algorithms. The results of the proposed algorithm are underlined. Clearly the algorithm outperforms most of the algorithms, given that a good initial disparity estimate is taken. The correctness of this comparison is subject to the

	Tsukuba			Sawtooth			Venus			Map	
	$B_{\overline{\mathcal{O}}}$	$B_{\overline{\mathcal{T}}}$	$B_{\mathcal{D}}$	$B_{\overline{\mathcal{O}}}$	$B_{\overline{\mathcal{T}}}$	$B_{\mathcal{D}}$	$B_{\overline{\mathcal{O}}}$	$B_{\overline{\mathcal{T}}}$	$B_{\mathcal{D}}$	$B_{\overline{\mathcal{O}}}$	$B_{\mathcal{D}}$
SSD Window	*1.67*	*1.67*	*3.27*	*1.47*	*1.32*	*3.53*	*2.31*	*3.27*	*1.71*	*2.84*	*9.36*
Adaptive Window	*1.60*	*1.46*	*3.18*	*1.42*	*1.27*	*3.43*	*1.99*	*2.84*	*3.01*	*2.82*	*9.30*
New. Adaptive W.	*1.43*	*1.21*	*2.96*	*1.17*	*0.90*	*2.59*	*1.81*	*2.46*	*2.61*	*2.81*	*9.29*
20 Layered	1.58 *3*	1.06 *4*	8.82 *3*	**0.34** *1*	**0.00** *1*	**3.35** *1*	1.52 *3*	2.96 *10*	2.62 *2*	0.37 *6*	5.24 *6*
*4 Graph cuts	1.94 *5*	1.09 *5*	9.49 *5*	1.30 *6*	0.06 *3*	6.34 *6*	1.79 *7*	2.61 *8*	6.91 *4*	0.31 *4*	3.88 *4*
19 Belief prop.	**1.15** *1*	**0.42** *1*	**6.31** *1*	0.98 *5*	0.30 *5*	4.83 *5*	1.00 *2*	0.76 *2*	9.13 *6*	0.84 *10*	5.27 *7*
11 GC+occl.	1.27 *2*	0.43 *2*	6.90 *2*	0.36 *2*	**0.00** *1*	3.65 *2*	2.79 *12*	5.39 *13*	**2.54** *1*	1.79 *13*	10.08 *12*
10 Graph cuts	1.86 *4*	1.00 *3*	9.35 *4*	0.42 *3*	0.14 *4*	3.76 *3*	1.69 *6*	2.30 *6*	5.40 *3*	2.39 *16*	9.35 *10*
8 Multiw. cut	8.08 *17*	6.53 *14*	25.33 *18*	0.61 *4*	0.46 *8*	4.60 *4*	**0.53** *1*	**0.31** *1*	8.06 *5*	0.26 *3*	3.27 *3*
12 Compact win.	3.36 *8*	3.54 *8*	12.91 *9*	1.61 *9*	0.45 *7*	7.87 *7*	1.67 *5*	2.18 *4*	13.24 *9*	0.33 *5*	3.94 *5*
14 Realtime	4.25 *12*	4.47 *12*	15.05 *13*	1.32 *7*	0.35 *6*	9.21 *8*	1.53 *4*	1.80 *3*	12.33 *7*	0.81 *9*	11.35 *15*
*5 Bay. diff.	6.49 *16*	11.62 *19*	12.29 *7*	1.45 *8*	0.72 *9*	9.29 *9*	4.00 *14*	7.21 *16*	18.39 *13*	**0.20** *1*	2.49 *2*
9 Cooperative	3.49 *9*	3.65 *9*	14.77 *11*	2.03 *10*	2.29 *14*	13.41 *13*	2.57 *11*	3.52 *11*	26.38 *17*	0.22 *2*	**2.37** *1*
*1 SSD+MF	5.23 *15*	3.80 *10*	24.66 *17*	2.21 *11*	0.72 *10*	13.97 *15*	3.74 *13*	6.82 *15*	12.94 *8*	0.66 *8*	9.35 *10*
15 Stoch. diff.	3.95 *10*	4.08 *11*	15.49 *15*	2.45 *14*	0.90 *11*	10.58 *10*	2.45 *9*	2.41 *7*	21.84 *15*	1.31 *12*	7.79 *9*
13 Genetic	2.96 *6*	2.66 *7*	14.97 *12*	2.21 *12*	2.76 *16*	13.96 *14*	2.49 *10*	2.89 *9*	23.04 *16*	1.04 *11*	10.91 *14*
7 Pix-to-pix	5.12 *14*	7.06 *17*	14.62 *10*	2.31 *13*	1.79 *12*	14.93 *17*	6.30 *17*	11.37 *18*	14.57 *10*	0.50 *7*	6.83 *8*
6 Max flow	2.98 *7*	2.00 *6*	15.10 *14*	3.47 *15*	3.00 *17*	14.19 *16*	2.16 *8*	2.24 *5*	21.73 *14*	3.13 *17*	15.98 *18*
*3 Scanl. opt.	5.08 *13*	6.78 *15*	11.94 *6*	4.06 *16*	2.64 *15*	11.90 *11*	9.44 *19*	14.59 *19*	18.20 *12*	1.84 *14*	10.22 *13*
*2 Dyn. prog.	4.12 *11*	4.63 *13*	12.34 *8*	4.84 *19*	3.71 *19*	13.26 *12*	10.10 *20*	15.01 *20*	17.12 *11*	3.33 *18*	14.04 *17*
17 Shao	9.67 *18*	7.04 *16*	35.63 *19*	4.25 *17*	3.19 *18*	30.14 *20*	6.01 *16*	6.70 *14*	43.91 *20*	2.36 *15*	33.01 *20*
16 Fast Correl.	9.76 *19*	13.85 *20*	24.39 *16*	4.76 *18*	1.87 *13*	22.49 *18*	6.48 *18*	10.36 *17*	31.29 *18*	8.42 *20*	12.68 *16*
18 Max surf.	11.10 *20*	10.70 *18*	41.99 *20*	5.51 *20*	5.56 *20*	27.39 *19*	4.36 *15*	4.78 *12*	41.13 *19*	4.17 *19*	27.88 *19*

Fig. 5. Initial window estimate: Direction-wise Average of already computed neighboring window sizes

data of other algorithms given in [3]. The algorithm is performing a bit worse at discontinuity in Venus image due to slanted regions, but overall it is better. After a set of experiments most of the stereo pairs gave best results with minimum limiting window size of 16 x 16 and value of d as 6, however, the value of d must be decreased with the number of iterations, as the disparity error reduce in each iteration.

5 Conclusion

We have presented an improved and efficient iterative stereo matching algorithm using adaptive window in this paper. The algorithm selects a window adaptively computed by the algorithm proposed by [1] and performs WTA for SSD around each pixel to reduce disparity errors introduced by usage of rectangular window and floating point disparity errors. The proposed algorithm helps improving the disparity estimate at each iteration over the image which in turn helps the disparity estimate to converge faster. The adaptive window algorithm proposed by [1] has been taken because, it is completely local and does not include any global optimization. Also, the algorithm does not use any post-processing smoothing, but smooth surfaces are recovered as smooth while sharp disparity edges are retained. Given a good initial disparity estimate, the proposed algorithm will reduce the disparity errors. However, the number of iterations and computation

time depends on the algorithmic parameters. The experimental results demonstrate a clear advantage of the proposed algorithm over the algorithms with a fixed window size and existing adaptive window based algorithm for standard stereo pairs. Also the proposed algorithm improves the computational complexity by a large factor. We are working towards further reducing the computation for the real-time applications.

References

1. Takeo Kanade and Masatoshi Okutomi, A stereo matching algorithm with an adaptive window: Theory and Experiment, IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(9):920 932, September 1994.
2. O Veksler, Stereo correspondence with compact windows via minimum ratio cycle, IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(12):1654-1660, December 2002.
3. Daniel Scharstein and Richard Szeliski, A Taxonomy and Evaluaion of Dense Two-Frame Stereo Correspondence Algorithms, International Journal of Computer Vision, Vol. 47, Numbers 1-3, pages 7-42, April 2002.
4. V. Kolmogorov, Graph Based Algorithms for Scene Reconstruction from Two or More Views. PhD thesis, Cornell University, 2004.
5. http://cat.middlebury.edu/stereo/, Middlebury Stereo data and basic implementation.