A Method for Learning Matching Errors in Stereo Computation

Dan Kong and Hai Tao

Department of Computer Engineering

University of California, Santa Cruz

Santa Cruz, CA 95064

(kongdan,tao)@soe.ucsc.edu

Abstract

This paper describes a novel learning-based approach for improving the performance of stereo computation. It is based on the observation that whether the image matching scores lead to true or erroneous depth values is dependent on the original stereo images and the underlying scene structure. This function is learned from training data and is integrated into a depth estimation algorithm using the MAP-MRF framework. Because the resultant likelihood function is dependent on the states of a large neighboring region around each pixel, we propose to solve the high-order MRF inference problem using the simulated annealing algorithm combined with a Metropolis-Hastings sampler. A segmentation-based approach is proposed to accelerate the computational speed and improve the performance. Preliminary experimental results show that the learning process captures common errors in SSD matching including the fattening effect, the aperture effect, and mismatches in occluded or low texture regions. It is also demonstrated that the proposed approach significantly improves the accuracy of the depth computation.

1 Introduction

The problem of recovering dense geometric information of a scene from images taken from multiple cameras has been extensively studied in the computer vision community for more than three decades. Steady progress has been made in improving the accuracy, the robustness, and the efficiency of stereo algorithms. The extraction of dense 3D structure from stereo images involves establishing correspondences between images and computing depth values using the triangulation method. Main challenges in stereo computation stem from the ambiguous or erroneous correspondences caused by the aperture effect, repetitive patterns, textureless regions, occlusion, and scene appearance changes. Methods have been proposed to solve these problems by improving the matching function or by applying sophisticated regularization schemes to suppress the matching errors. Several stereo algorithms in the first category include the adaptive windows matching [9], multi-view stereo [15], non-linear diffusion [16], and mutual information based matching [10]. Some popular regularization techniques include the cooperative-competitive algorithms [13],[21], scan line based Bayesian inference [2], coarse-to-fine stereo [6], graph cut methods [3], and surface model fitting [8]. An excellent review of early stereo algorithms can be found in [5]. Recently, Shastein and Szeliski [17] reviewed and compared more than twenty existing stereo algorithms and made the evaluation method and implementations of several popular stereo algorithms available at the web site [22].

In this paper, we propose a new approach that learns the behaviour of the sum-of-squared-differences (SSD) matching and integrates this knowledge into a probabilistic framework to improve the depth

computation. By observing that the occurrence of matching errors is determined by the image texture, the 3D scene structure, and the size of matching window, the proposed method is designed to learn the probability of matching errors as a function of these variables. The learned probabilities are used to correct the false matching of SSD and help infer the true depth.

To illustrate this idea, let us consider the fattening effect in the SSD-based matching (Figure 1). This refers to the phenomenon that foreground objects appear to be bigger in the depth maps. The explanation for this effect is that when a background pixel near a foreground object is matched using an image window, some foreground pixels are also included in the computation. Whether this type of error occurs is determined by how far a pixel is from a close-by foreground object, how large the matching window is, and how strong the background texture is. In other words, the behaviour of SSD matching at each pixel is a function of the stereo images texture, the true depth map and the matching window size. This observation is also true for other matching errors caused by low texture content, aperture effect, or occlusion (Figure 1). The proposed method learns this function from a training dataset and integrates it into an MAP-MRF framework. One of the new challenges in this approach is that the likelihood function of each site has to be evaluated based on the states of many other sites in a relative large neighbouring area. We solve this difficult inference problem in high-order MRF using the Metropolis-Hastings sampling algorithm. To improve the performance, a segmentation based method is further developed. Experimental results show that (1) the learned matching state distribution does capture various types of matching errors, and (2) integrating this distribution into the depth computation significantly improves the performance. It should be noted that although some the latest algorithms reported the best performance by using single pixel matching [22], in many real applications the image quality might not be as ideal as the test images in [22] due to image noise, illumination changes, and camera calibration errors. Window-based matching proved to be more robust under such conditions.
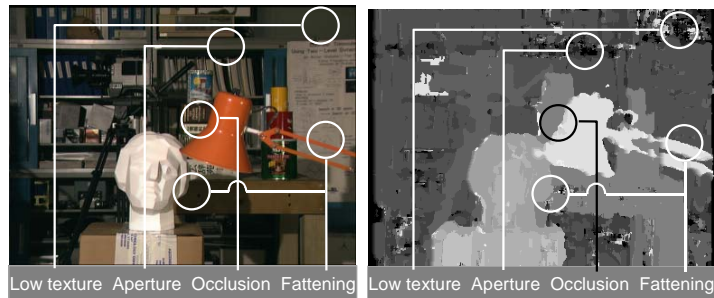

Figure 1. Several common errors in window-based stereo matching. Left: the original left image. Right: the depth map computed using SSD matching with 9×9 windows.

2 The approach

2.1 Modelling the SSD matching errors

The main questions in modelling matching errors are which types of errors need to be modelled, how to model them, and how to use the learned model to improve depth estimation. We notice that besides the fattening effect discussed in the previous section, another common problem in window-based matching is the matching ambiguities in low-texture regions, where the matching scores provide little information on the actual depth of the scene. In addition, aperture effect often occurs when the dominant orientation of the texture is horizontal. Under this situation, matching scores are similar along the epipolar lines and do not provide accurate depth information. Finally, occlusion causes matching errors that are difficult to correct due to the lack of information. In this paper, we propose to define a matching state variable l_i for

each pixel i to indicate whether the window-based matching leads to the true depth, the nearby foreground depth, or other wrong depth values, i.e. $l_i \in \{true, foreground, outlier\}$.

As discussed previously, l_i is a function of the stereo images, the underlying 3D scene structure, and the size of SSD window. This function is denoted as $P(l_i \mid X, I, A)$, where I is the reference image, X is the true depth map, and A is the matching window size. For simplicity, we will omit A sometimes when we assume a fixed window size. This distribution is learned from a training dataset. Once this distribution is available, the raw SSD matching scores can be interpreted in more accurately for estimating depth. More specifically, suppose SSD matching score is $C_{i,j}$ for each pixel i at a disparity level j. Using the matching state l_i, the combined new likelihood becomes

$$
\begin{aligned}
P(C_i \mid X, I) = P(C_i \mid X_i, l_i = True)P(l_i = True \mid X, I) + \\
P(C_i \mid fg(i, X))P(l_i = foreground \mid X, I) + \\
P(C_i \mid X_i, l_i = outlier)P(l_i = outlier \mid X, I)
\end{aligned}
\tag{1}
$$

where $P(C_i \mid X_i, l_i = true) = C_{i,X_i}$ is the likelihood of observing matching scores C_i when the matching is correct. Its value can be approximated as C_{i,X_i}. $fg(i,X)$ is a depth extrapolation function that returns the depth of the nearest foreground object. $P(C_i \mid fg(i,X))$ is the likelihood of observing matching scores C_i if given the depth $fg(i,X)$. It can be approximated as $C_{i,fg(i,X)}$. Finally, $P(C_i \mid X_i, l_i = outlier) = \alpha$ is a constant representing the likelihood of observing matching scores C_i for outlier, the probability in Eq. (1) can be used as a new matching measure or likelihood at pixel i.
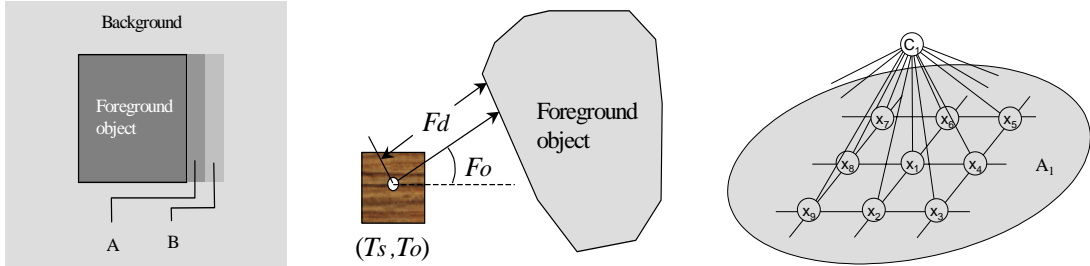

Figure 2. Left: Matching measures in region B do not support the depth map with the fattening effect. Middle: The image and depth attributes around a pixel. Right: The computation of the likelihood function depends on depth values in a neighbouring region.

To illustrate why modelling the matching state l_i improves the performance of depth estimation, Figure 2 shows an example of the fattening effect. A background region A and the neighbouring background regions B are considered for the two candidate depth maps: the true depth map and the depth map with the fattening effect where A has the foreground depth. It can be observed that both depth hypotheses have high scores in region A using our likelihood function. However, for the wrong depth map where region A has the foreground depth, it expects the fattening effect in B, which is not observed. Therefore, the foreground depth in region A must be the result of the fattening effect and the true depth has higher likelihood.

2.2 Image attributes and the matching state distribution

The conditional matching state distribution $P(l_i | X, I, A)$ is high dimensional because the matching scores and therefore the variable l_i is affected by neighbouring image pixels and 3D scene structures. Direct learning of this high dimensional distribution is difficult. However, this distribution can be approximated by extracting a small set of image and structure attributes and replacing the original image I and depth map X with these attributes. The image and structure attributes should be complete in the sense that all the factors that influence the window-based matching are included. Meanwhile, the attribute set should also be compact enough so that learning is computationally feasible. We propose to use the following four attributes to condition l_i. They are the texture strength T_s, the texture orientation T_o, and the distance F_d and the orientation F_o of the displacement vector to the nearest foreground object (see Figure 2 for an illustration). Using these attributes, the conditional distribution $P(l_i | X, I, A)$ can be approximated as

$$P(l_i | X, I, A) \approx P(l_i | T_{s,i}, T_{o,i}, F_{d,i}, F_{o,i}, A)$$ (2)

The conditional distribution $P(l_i | T_{s,i}, T_{o,i}, F_{d,i}, F_{o,i}, A)$ is represented as a histogram and is learned from a dataset with both stereo images and ground truth depth available. Staring from a SSD depth map computed using window size A, the learning procedure consists of the following steps:

Step 1: Compute image attributes T_s, T_o from the reference image I and attributes F_d, F_o from the ground truth depth map X

Step 2: For each pixel i, based on the ground truth, determine if depth resulted from the matching scores is the true depth, a nearby foreground depth, or an outlier. The result is assigned to l_i

Step 3: For each pixel, quantize the four attributes and store l_i to one of the three histograms representing

$$P(l_i = true | T_{s,i}, T_{o,i}, F_{d,i}, F_{o,i}, A),$$
$$P(l_i = foreground | T_{s,i}, T_{o,i}, F_{d,i}, F_{o,i}, A),$$
$$P(l_i = outlier | T_{s,i}, T_{o,i}, F_{d,i}, F_{o,i}, A)$$

Step 4: After the histograms are constructed based on all training images, normalize the corresponding bins in the three histograms to 1.

In our implementation, the texture strength $Ts(x)$ is quantized to 16 levels. The texture orientation T_o is quantized into 8 levels from 0 to 180 degrees. The quantization levels of F_d and F_o are dependent on the size and shape of the matching window. Because most matching methods use windows no larger than 11×11 pixels, F_d is within the range of 8 pixels, which is roughly half of the diagonal length of a 11×11 window. F_d is quantized to 8+1 levels with one special level representing pixels without foreground objects around them. F_o is divided into 8 levels from 0 to 360 degrees. Therefore, the total number of bins in a histogram is $16 \times 8 \times 9 \times 8$.

2.3 MAP-MRF depth estimation using the matching state distribution

The depth map X is modelled as a Markov random field. The potential function between the neighbouring sites is designed to enforce the smoothness constraint. Line process is also integrated into the formulation using a robust function similar to the one used in [18]. More specifically, the prior function is defined as

$$P(X) \propto \prod_i P(X_i \mid N_i(X_i)) = \prod_i \exp(-V_c(X_i))$$
$$= \prod_i \prod_{j \in N_i} \exp(-\rho(X_i, X_j)) \tag{3}$$

where N_i is first-order neighbour system including the four adjacent pixels of location i. The potential function $V_c(X_i)$ is defined using the robust function

$$\rho(X_i, X_j) = -\ln((1 - e_p) \exp(-\frac{|X_i - X_j|}{\sigma_p}) + e_p) \tag{4}$$

Given the reference image I and the matching scores C, the depth map X is estimated as

$$\max_X \arg P(X \mid C, I) \propto \max_X \arg P(X)P(C \mid X, I) =$$
$$\max_X \arg \prod_i P(X_i \mid N_i(X_i)) P(C_i \mid X, I) \tag{5}$$

where $P(C \mid X, I) = \prod_i P(C_i \mid X, I)$ is the likelihood function and $P(C_i \mid X, I)$ is defined in Eq. (1).

Eq. 5 is similar to the MRF formulation previously proposed in [3] and [18]. However, it should be noticed that the likelihood function in Eq. (1) needs to be computed using neighbouring depth values. This is equivalent to estimating the graph depicted in the right of Figure 2.

2.4 Estimating the depth

In an MRF where the likelihood of each site is independent of other sites, the graph cut [3] and the belief propagation [18] algorithms proved to be effective for stereo computation. However, it is more difficult to optimize a high-order MRF such as the one shown in Figure 2 [12],[11]. We apply the Metropolis-Hastings sampling algorithm [14],[7] to solve this problem. Using Metropolis-Hastings algorithm, a proposal move from current depth solution X to a new solution X' is accepted with the probability

$$P(X \rightarrow X') = \min\left(1, \left[\frac{P(X' \mid C, I)q(X' \rightarrow X)}{P(X \mid C, I)q(X \rightarrow X')}\right]^{1/T}\right) \tag{6}$$

where $P(X \mid C, I)$ and $P(X' \mid C, I)$ are the posterior probabilities of the two configurations and $q(X \rightarrow X')$ and $q(X' \rightarrow X)$ are the proposal probabilities. In our algorithm, for pixel i, each proposal move either keeps its current depth X_i, changes the depth to its initial depth X_i^0, or changes the depth to those of its neighbouring pixels $X_i' \in N(X_i)$. In other words, $X_i' \in \{X_i, N(X_i), X_i^0\}$. As discussed previously, by changing the depth of a single pixel, the likelihood values of neighbouring pixels are affected and need to be recomputed. This computation can be simplified because only the ratio between the likelihood values is needed in Eq. (6). To compute this ratio, the likelihood values of the nearby affected pixels are computed for configurations X and X'. More specifically, the posterior ratio is calculated as

$$\frac{P(X' \mid C, I)}{P(X \mid C, I)} = \frac{P(X')}{P(X)} \frac{P(C \mid X', I)}{P(C \mid X, I)} = e^{-(V_c(X'_i) - V_c(X_i))} \cdot \prod_{j \in A_i} \frac{P(C_j \mid X', I)P(X')}{P(C_j \mid X, I)P(X)} \tag{7}$$

where A_i defines the set of pixels whose likelihood values are affected by the depth change at pixel i. The temperature T in Eq. (6) controls the speed of the cooling process and decreases according to $T^{(t)} = \kappa T^{(t-1)}$, where κ is a constant between 0.8 and 0.99.

2.5 Segmentation-based optimization

Though pixel-based sampling converges to the global optimal solution, it is not efficient in practice. To solve this problem, we further propose a segmentation-based Metropolis-Hastings algorithm. In our implementation, we adopt a scheme that uses the joint colour and depth segmentation. The colour segmentation is computed using the mean-shift algorithm [4]. The depth segmentation is obtained using the SSD matching scores convolved with a median filter. The intersection of the two segmentation maps is obtained to get an over-segmentation of the reference image. Figure 3 shows an example of the joint segmentation result.
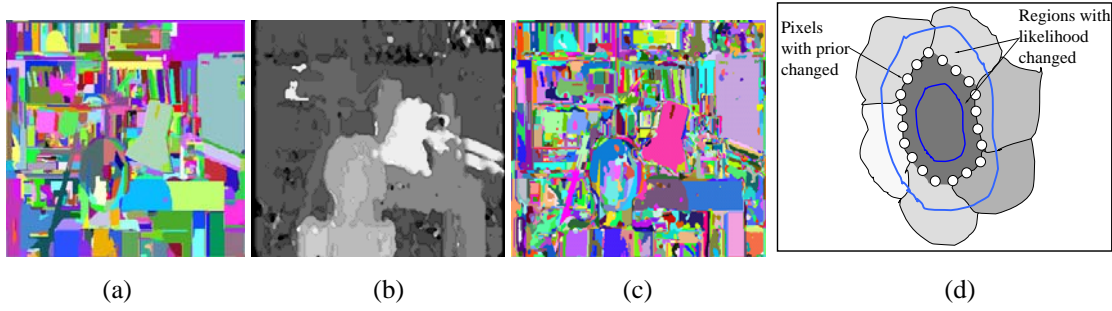

(a)   (b)   (c)   (d)

Figure 3 .(a): color segmentation result using mean-shift. (b): median filtered depth map. (c): joint color and depth segmentation result. (d): The computation of likelihood and prior change for one super-pixel in segmentation-based approach

By introducing segmentation, we can regard each segment as a super-pixel. Then the depth estimation proceeds by iteratively hypothesizing the depth of each super-pixel. The possible proposal moves from current depth solution to a new solution can still be written as $X_i' \in \{X_i, N(X_i), X_i^0\}$, but the subscript i denotes the segment index. Another difference is that $N(X_i)$ varies depending on the neighbourhood configuration. Segmentation-based optimization also makes the computation of the prior and the likelihood different from the pixel-based approach. This can be illustrated in figure 3(d). By flipping the depth of a segment to a new value, we need to compute the likelihood change in a window larger than the bounding box of the current segment. For the prior term, since each segment takes a single depth value before and after flipping, we only need to compute the prior change for the pixels around the segment boundary.

3 Implementation and experimental results

3.1 Learning the matching state distribution

The matching state distribution is learned based on stereo image pairs and the associated ground truth depth maps provided at [22]. To compute the image attributes, we used the canny edge detector to find the depth boundaries in the ground truth depth maps. Then the search of the nearest foreground object is carried out in a 9×9 window centered at each pixel. If a depth boundary is detected in the window, F_d and F_o are computed and quantized. The texture strength T_s and texture orientation T_o are computed using the gradient information in a small window around each pixel. Our experiments showed that even

for this relatively small dataset, the learned matching state distribution reveals all the types of matching errors previously discussed. The results presented in this section are based on 5×5 SSD matching.

3.1.1 Fattening effect and occlusion

The learned matching state distribution clearly shows the fattening effect. Figure 4(a) plots the conditional probabilities $P(l_i = true \mid F_d)$ and $P(l_i = foreground \mid F_d)$ with F_d as the variable. These probabilities are derived from the joint conditional probability $P(l_i = true \mid F_d, F_o, T_s, F_o)$ and $P(l_i = foreground \mid F_d, F_o, T_s, F_o)$. The horizontal axis represents the distance of a pixel to its nearest foreground depth (in pixels). It is observed that the probability of a background pixel being mistakenly computed as the nearby foreground depth decreases as F_d increases. The probability of computing true depth has a reverse trend. Figure 4(c-d) compare the pair $P(l = \{true, foreground\} \mid F_d, F_o = left)$ and the pair $P(l = \{true, foreground\} \mid F_d, F_o = up)$. They show that the fattening effect occurs mainly along the vertical depth boundaries. In Figure 4(b), the probability $P(l = true \mid F_d, F_o = left)$ is compared with $P(l = true \mid F_d, F_o = right)$. From these two curves, it is observed that $P(l = true \mid F_d, F_o)$ is not isotropic. The curve $P(l = true \mid F_d, F_o = left)$ is always below the curve $P(l = true \mid F_d, F_o = right)$. This phenomenon can be explained by occlusion. Because in the training phase, the left images are always treated as the reference images, therefore occlusion occurs on the left side of foreground objects.
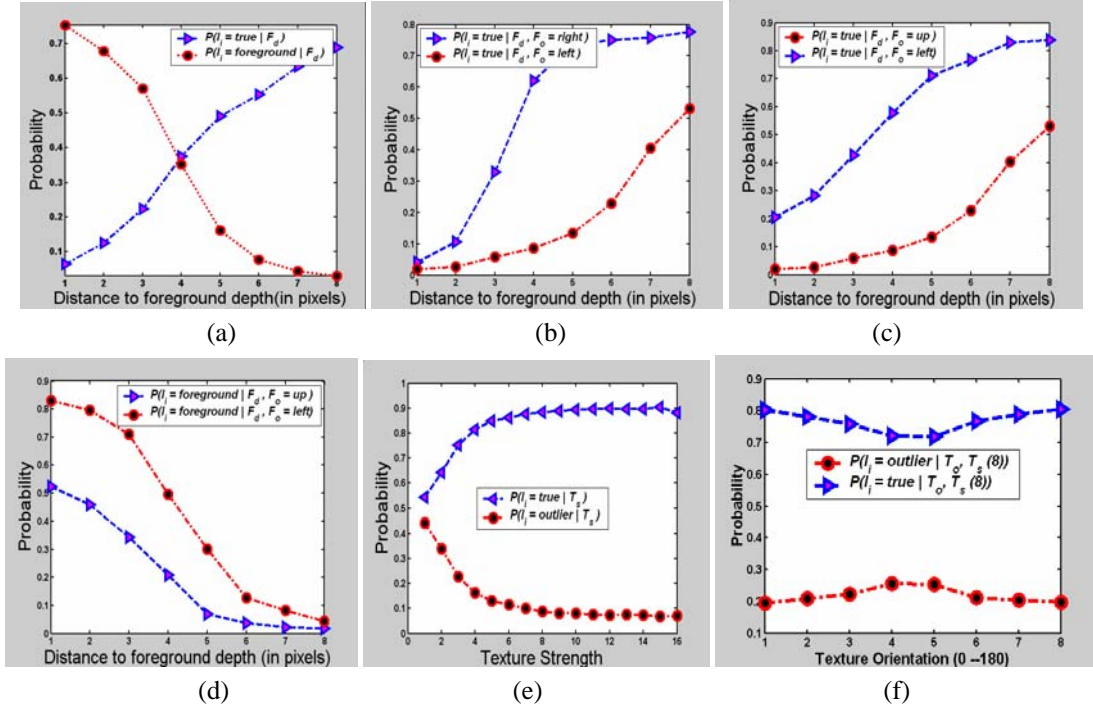

Figure 4. (a)(c)(d) Fattening effect, (b) occlusion, (e) outliers in low texture regions, (f) aperture effect.

3.1.2 Low-texture regions

The learned distribution also reveals the matching ambiguities in regions with low texture. Figure 4(e) shows the distributions $P(l = true \mid T_s)$ and $P(l = outlier \mid T_s)$, with the texture strength as the variable. Higher probability of being outliers can be observed in low-texture regions.

3.1.3 Aperture effect

Figure 4(f) shows the matching state distribution $P(l = outlier, true \mid Ts(8), To)$. These curves are obtained by fixing the texture strength at intermediate levels and using the texture orientation as the changing variable. It can be observed that the learned distribution has higher probability of being outliers when the texture orientation is horizontal, while the probability of matching the true depth is higher when the texture orientation is vertical.

3.2 Depth computation

Depth maps are estimated using the method described in Section 2.4. The initial depth values are obtained using window-based SSD matching. The computation of matching scores is available at the stereo evaluation web site [22]. They are first truncated and then converted into raw matching measures

When computing the likelihood according to Eq. (1), the outlier constant is $\alpha = 0.1$. For the prior function in Eq. (3), the parameters in the robust function are $e_p = 0.05$ and $\sigma_p = 0.6$. On a 2 GHz PC, each iteration of Metropolis-Hastings sampling takes 5 to 10 seconds. The algorithm converges within 100 to 200 iterations. With the segmentation-based optimization, the algorithm converges within 30 to 60 seconds, which is a great improvement over single-pixel based method in terms of computational speed.

We tested the proposed algorithm using several image pairs. A leave-one-out strategy is adopted in the training and testing process to avoid using the same images in training and testing. To evaluate the proposed algorithm, we compared the single-pixel based Metropolis-Hastings algorithm with the segmentation-based optimization. The MAP-MRF portion of the algorithm remains unchanged in this comparison. Table 1 shows that the segmentation-based method largely improves the depth error for the benchmark images.

Algorithm	LBS			LBS (seg)			GC			BP		
	all	untex	disc	all	untex	disc	all	untex	disc	all	untex	disc
Tsukuba	3.86	3.53	16.74	**2.03**	**0.77**	**11.75**	4.91	4.36	20.79	3.62	3.83	15.17
Sawtooth	2.19	1.00	11.3	**1.34**	**0.90**	**5.88**	3.94	1.89	19.86	4.82	9.73	12.13
Venus	4.32	7.89	14.81	**2.28**	**3.26**	**18.80**	2.79	3.00	25.69	12.72	2.60	18.5
map	**0.75**	**0.00**	9.84	0.83	1.67	9.70	1.33	2.14	12.7	0.76	2.61	**7.73**

Table 1. Comparison of the error percentage rates in stereo algorithms with the matching scores computed using 5x5 windows. LBS – the proposed algorithm. LBS (seg) – the proposed algorithm with segmentation-based optimization. GC- the graph cut algorithm. BP – the belief propagation algorithm.

The proposed algorithm was also compared with other state-of-art algorithms including the graph cut [3] method and the belief propagation method [18]. Table 1 shows that the proposed algorithm has the lowest overall error rates. One thing we need to point out is that the score we quoted here for other state-of-art algorithm differ from those listed on [22] since we run those algorithms with a window based matching cost. It is true that using single pixel based matching on the Tsukuba images and some of the other test images in [22] will result in lower error rates. The reason we investigated window based matching is that it is more robust when image noise or illumination changes are not negligible. Figure 5 shows depth maps estimated using different algorithms. It can be observed that the fattening effect around the foreground objects such as the lamp has been suppressed in our results and correct depth is computed in the textureless areas.

4 Discussions

In this paper, an algorithm is proposed to learn the SSD matching errors as a function of the stereo images, the scene structure, and the matching windows size. Preliminary results show that this approach is very promising. However, several issues need to be further studied. Among them, the first question is what are the other image and structure attributes that can be used in the learning of the matching state distribution. Several promising features include thin structures, depth gaps, and texture attributes on the foreground and the background objects along depth boundaries. Another issue is the segmentation in stereo computation. Currently, a joint colour and depth segmentation significantly improves the performance and results of the algorithm. In our experiments, we found that the final depth results are sensitive to the segmentation parameters which implied that segmentation and stereo computation should not be in separate processes. Segmentation-based depth search [19],[20] and fast algorithms such as the Swendsen-Wang cuts [1] can be explored to further improve results. Finally, it is conceivable that the proposed learning-based approach can be generalized to other stereo algorithms. How to learn the depth estimation errors in these algorithms is an interesting research problem.

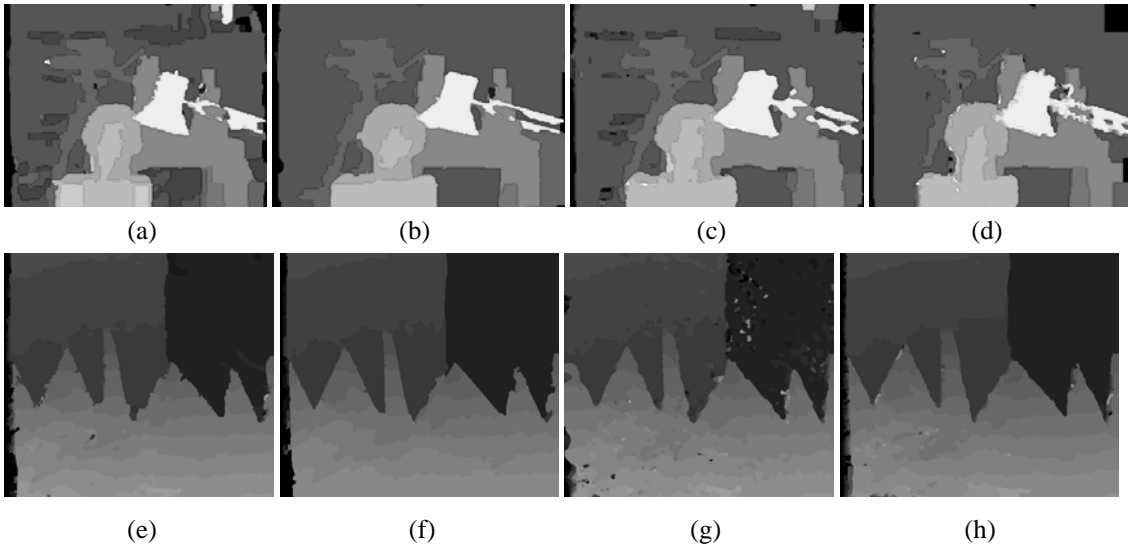| (a) | (b) | (c) | (d) |
| (e) | (f) | (g) | (h) |

Figure 5. (a) LBS (pixel-based) (b) LBS (segmentation-based) (c) BP, 5x5 SSD (d) GC, 5x5 SSD. (e) LBS (pixel-based) (f) LBS (segmentation-based) (g) BP, 5x5 SSD, (h) GC, 5x5 SSD.

References

[1] A. Barbu, S. C. Zhu, "Graph partition by Swendsen-Wang cuts," in *Proc. Intl. Conf. on Computer Vision*, pp. 320-327, 2003.

[2] P. N. Belhumeur, "A Bayesian-approach to binocular stereopsis," *Int. Journal of Computer Vision, vol. 19, no. 3*, pp. 237-260, August 1996.

[3] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," in *Proc. Int. Conf. on Computer Vision*, September 1999.

[4] D. Comaniciu and P. Meer, "Robust analysis of feature spaces: color image segmentation," in *Proc. of IEEE conference on Computer Vision and Pattern Recognition*, pp. 750-755, 1997.

[5] U. R. Dhond and J. K. Aggarwal, "Structure from stereo: a review," *IEEE Transactions on System, Man, and Cybernetics, vol. 19, no. 6*, pp. 1489-1510, 1989.

[6] K. J. Hanna and Neil E. Okamoto, "Combining stereo and motion analysis for direct estimation of scene structure," in *Proc. Intl. Conf. on Computer Vision*, pp 357-265, 1993.

[7] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, 57, pp. 97-109, 1970.

[8] W. Hoff and N. Ahuja, "Surfaces from stereo: integrating feature matching, disparity estimation, and contour detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *vol. 11, no. 2*, pp. 121-136, February 1989.

[9] T. Kanade and M. Okutomi, "A stereo matching algorithm with an adaptive window: theory and experiment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *vol. 16*, *no. 9*, pp. 920-932, September 1994.

[10] J. Kim, V. Kolmogorov, R. Zabih, "Visual Correspondence Using Energy Minimization and Mutual Information," in *Proc. Intl. Conf. on Computer Vision*, pp. 1033-1040, October 2003.

[11] V. Kolmogorov and R. Zabih, "What Energy Functions can be Minimized via Graph Cuts?" in *Proc. European Conference on Computer Vision*, pp. 65-81, 2002.

[12] S.Z. Li, Markov Random Field Modeling in Image Analysis, Springer-Verlag, 2001.

[13] D. Marr and T. Poggio, "A computational theory of human stereo vision," in *Proceedings of the Royal Society London B, 204*, pp. 301-328, 1979.

[14] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A, Teller, and E. Teller, "Equations of the state calculations by fast computing machines," *J. Chem. Physics*, 21, pp. 1087-1091, 1953.

[15] M. Okutomi and T. Kanade, "A multiple-baseline stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence vol. 15*, *no. 4*, pp. 353-363, April 1993.

[16] D. Scharstein and R. Szeliski, "Stereo matching with nonlinear diffusion," *International Journal of Computer Vision*, 28(2):155-174, July 1998.

[17] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. Journal of Computer Vision*, 2002.

[18] J. Sun, H. Y. Shum, and N. N. Zheng, "Stereo matching using belief propagation," in *Proc. European Conference on Computer Vision*, 2002.

[19] Hai Tao and Harpreet S. Sawhney, "Global matching criterion and color segmentation based stereo," in *Proc. Workshop on the Application of Computer Vision (WACV2000)*, pp. 246-253, Dec. 2000.

[20] Z. Tu and S. C. Zhu, "Image Segmentation by Data-Driven Markov Chain Monte Carlo," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(5): 657-673, 2002.

[21] C. L. Zitnick and T. Kanade, "A cooperative algorithm for stereo and occlusion detection," CMU-RI-TR-99-35, October 1999.

[22] http://cat.middlebury.edu/stereo/