Efficient Edge-Awareness Propagation via Single-Map Filtering for Edge-Preserving Stereo Matching

Takuya Matsuo, Shu Fujita, Norishige Fukushima, Yutaka Ishibashi

Nagoya Institute of Technology

ABSTRACT

In this paper, we propose an efficient framework for edge-preserving stereo matching. Local methods for stereo matching are more suitable than global methods for real-time applications. Moreover, we can obtain accurate depth maps by using edge-preserving filter for the cost aggregation process in local stereo matching. The computational cost is high, since we must perform the filter for every number of disparity ranges if the order of the edge-preserving filter is constant time. Therefore, we propose an efficient iterative framework which propagates edge-awareness by using single time edge-preserving filtering. In our framework, box filtering is used for the cost aggregation, and then the edge-preserving filtering is once used for refinement of the obtained depth map from the box aggregation. After that, we iteratively estimate a new depth map by local stereo matching which utilizes the previous result of the depth map for feedback of the matching cost. Note that the kernel size of the box filter is varied as coarse-to-fine manner at each iteration. Experimental results show that small and large areas of incorrect regions are gradually corrected. Finally, the accuracy of the depth map estimated by our framework is comparable to the state-of-the-art of stereo matching methods with global optimization methods. Moreover, the computational time of our method is faster than the optimization based method.

Keywords: local stereo matching, real-time stereo matching, edge-preserving filter

1. INTRODUCTION

Recently, image processing with depth maps attracts attentions, e.g. pose estimation, object detection, point cloud processing and free viewpoint video rendering. In these applications, accurate estimated results are required if we want high-quality performance. Especially, for 3DTV and free viewpoint TV [1], the accuracy of depth maps is important for view synthesis [2, 3] and its compression [4, 5].

Depth maps are usually computed by stereo matching with stereo image pair. The depth values are computed from disparities in the correspondence between left and right images. Stereo matching consists of four steps [6]: matching cost computation, cost aggregation, depth map computation/optimization and depth map refinement.

The stereo matching can be classified into local methods and global methods. The global stereo matching, which contain an optimization process, generates accurate depth maps, but it is time-consuming. On the other hands, the local methods are less accurate but are faster than the global methods; thus the local methods are suitable for real-time applications. We focus on the local stereo matching in this paper.

The local stereo matching does not have the depth map optimization step. We can obtain more accurate depth maps if we can utilize edge-preserving filtering for the cost aggregation instead of using box filtering. The computational cost, however, is high since we should perform filtering for every disparity ranges, i.e., 256 times filtering for a byte accuracy. The problem of the computational cost can be solved if we exploit box filtering for the cost aggregation. The box filter can be efficiently performed by using integral images or summed area tables. Nevertheless, the accuracy of the estimated depth map becomes lower.

To solve the issue of the trade-off between matching accuracy and cost, therefore, we propose an efficient framework for real-time stereo matching to obtain accurate depth maps that is edge-preserved. The proposed framework iteratively performs the following steps. The method makes box filtering for the cost aggregation at first, and then performs edge-preserving filtering for the refinement process. The refined depth map is used for feedback of the matching cost computation. We call this stereo matching framework as in-loop feedback matching (IFM).

The rest of this paper is organized as follows. Section 2 describes related works. We present the proposed new stereo matching framework in Sec. 3. In Sec. 4, experimental results are shown. Section 5 concludes this paper.

2. RELATED WORKS

The global methods are based on the Markov random field, e.g. dynamic programming [7], multi-pass dynamic programming [8, 9], semi-global matching [10], belief propagation [11] and graph cuts [12]. Though the energy minimization formulation is elegant and generates accurate depth maps, these optimization processes still consume much time. Disregarding costs for estimating more accurate depth map, manual user inputs, which indicate object edges, are used with the global method [13]. The approach is take in the MPEG's reference software of depth estimation in 3DV [14].

The local method is faster than the global stereo matching. Leveraging the state-of-the-art of edge-preserving filtering, e.g. bilateral filter with efficient implementation [15, 16], guided filter [17], domain transform [18] and cross-based filtering [19], cost volume filtering for cost aggregation process in the local stereo matching become more accurate [20, 21, 22, 23, 24]. Though the edge-preserving filtering and box filtering for cost slices take same computational order that is O (1), box filtering is significantly fast. The speed thanks to a data structure of the cost volume in the loaded memory. For efficient edge-preserving filtering, memory should be continuously allocated along cost slice image. For cache efficiency, however, memory should be continuous along disparity search range. With summed area tables for box filtering can utilize the latter data structure, but its accuracy does not reach well-optimized approaches.

The performance of post filtering for the depth map refinement filters is leveraged recent edge-preserving filtering techniques. The edge-preserving filters can improve depth map accuracy with lower cost than the optimization methods and cost volume filtering method. Bilateral filter [25] is an early approach of them. The bilateral filter can remove noise while preserving edges, but the performance of edge keeping and noise reduction is trade-off. When the image has large noises, the performance of edge keeping becomes low to remove the noises. In addition, only Gaussian noise can be removed by the filter, although depth map contains spike, speckle/blob and non-Gaussian noises. Moreover, the inaccuracy around object boundaries cannot be correct well. Now, a variant called joint bilateral filter [26, 27, 28] relaxes bilateral filter's problem well by using guided information of original RGB image used in stereo matching. This method regards the depth map as a filtering target, and the original RGB image as a kernel computation target. The filter makes the kernel by color or intensity values of the RGB image instead of depth values. The filter can smooth small non-Gaussian noises. In addition, the filter can fit edges in the depth map around an object to edges in the natural image's object.

The joint depth and image processing make a new problem. The joint filters spreads blurring to the outside of the depth map due to mixed pixels, and outliers. Mixed pixels in natural images occur on foreground and background boundaries, and they are caused by CCD sensor's aliasing and optical lens blur. The joint bilateral filter transfers the blur to the depth map. In addition, impulse outliers and large size noises on the depth map are diffused by the filtering. Some methods can reduce this type of the blur in some degree. Multilateral filter [29] has three weights that are space, color and additional depth weight. The depth weight can keep the shape of the edge of the depth map so that the weight suspends blurring. On the contrary, the ability of edge-keeping loses a characteristic of the object boundary recovering.

Cost volume refinement filter [30, 31] and its speed-up approximation [32] have better performance than the edge-preserving filtering for depth maps directly. The methods can correct depth edge and can remove spike noises and small size speckle noises without diffusion of depth values. In addition, the method hardly generates blur at object boundaries. The cost volume refinement filter is inspired by the cost volume filtering of the stereo matching. The filter regenerates a cost volume from an estimated map. The cost volume consists multiple of cost slices, which indicate every possibility of a depth value at each pixel, and all slices of each depth level are stacked. Each slice is then filtered slice by slice at each depth level with edge-preserving filtering. The possibility is computed by a difference between an initial depth and each depth level values. These methods perform d times, the number of depth search range, edge-preserving filters, and also iterate the process; thus the method consumes much time.

The authors propose a solution for the trade-off in accuracy and computational cost between direct edge-preserving filtering and cost volume refinement filter. That is the weighted joint bilateral filter with slope depth compensation filter. The approach reduces blurred areas by using weight maps, which indicates unreliable points or around edges, and then completely remove the slope depth compensation filter. The accuracy of the filters can reach cost volume refinement filters with saving computational cost. Distinguishing the cost volume refinement filtering, we call such directly filtering methods as direct edge-preserving filtering. However, common problems in the refinement of depth maps is still remained; that is largely incorrect matching before post filtering steps cannot flip to correct depth values.

Leveraged by the previous works, we present a new stereo matching framework of the in-loop feedback matching for low computational matching with accurate matching. The new framework utilizes computationally efficient approaches in cost aggregation, depth map computation/optimization and depth map refinement processes. We use box filtering for the cost aggregation, winner-takes-all for depth map computation-optimization, i.e. local stereo matching, and direct edge-preserving filtering for depth map refinement. Then, the refined depth map is used for iteratively feedback in matching cost computation.

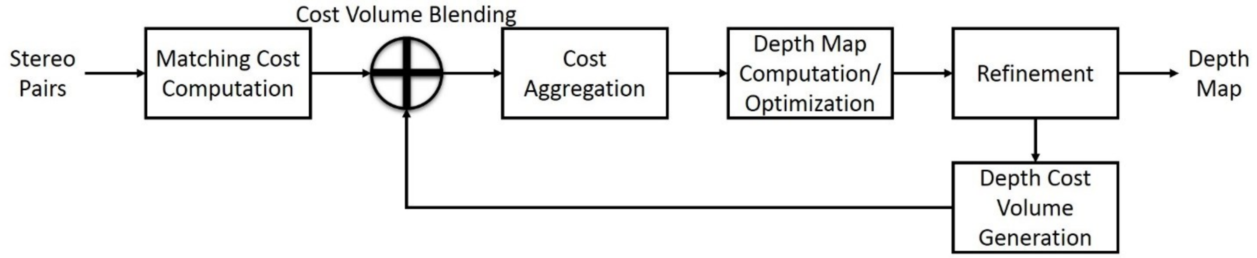

Cost Volume Blending
Stereo Pairs
Matching Cost Computation
Cost Aggregation
Depth Map Computation/ Optimization
Refinement
Depth Map
Depth Cost Volume Generation

Figure 1 Diagram of in-loop feedback matching for stereo correspondence problem.

3. IN-LOOP FEEDBACK MATCHING

In this section, we present a new stereo matching framework of in-loop feedback matching (IFM). We show the diagram of the IFM framework in Fig. 1. In the IFM, two additional steps are added to the usual stereo matching framework, i.e. the cost aggregation, depth map computation/optimization and the refinement step. These are a cost volume regeneration step and a cost volume blending step. Note that the additional steps make a loop for the stereo matching framework.

According to Fig. 1, at the first step, input stereo pairs are input and then its matching cost is computed for building a cost volume that is stacks of a cost slice. The computed matching cost are, then, aggregated. Next, the initial depth map is generated by computing or optimizing the cost volume. Finally, the initial depth map are refined by using filtering or other refining methods; then refined depth map is regenerated. These processes are the traditional stereo matching framework. The proposed framework of the IFM does the additional processes. The refined depth map is converted to a new cost volume, which represents the possibility of each depth level, via a regeneration process. Then the regenerated cost volume is blended to the first cost volume generated at the matching cost computation step, and then looped steps proceed iteratively. Details of each step are described as follows.

3.1 Matching Cost Computation, Cost Aggregation and Depth Map Computation

In this subsection, we describe the step of matching cost computation, the cost aggregation and depth map computation in the parts of the traditional stereo matching.

In the matching cost computation step, pixel-based matching cost between stereo pairs are computed with several methods, e.g. an absolute difference (AD), square intensity difference (SD), AD or SD of Sobel filtered images, Census transform, rank transform and so on. In this paper, we use the fused cost of AD and AD of edge images that is Sobel filtered, denoted *XSobel*. The matching cost computation is defined as:

$$Cost(d) = AD(d) + \alpha \cdot XSobel(d),$$

where d is a depth level, *Cost*(d), *AD*(d) and *XSobel* (d) are a matching cost, absolute difference of input images and absolute difference of edge images at each depth level, respectively. α is a blending parameter between each distance.

In the cost aggregation step, the computed matching cost are aggregated by using various method reported in the related work section. In this paper, box filtering is used for aggregation. In other words, we use block matching. The box filter is defined as follow:

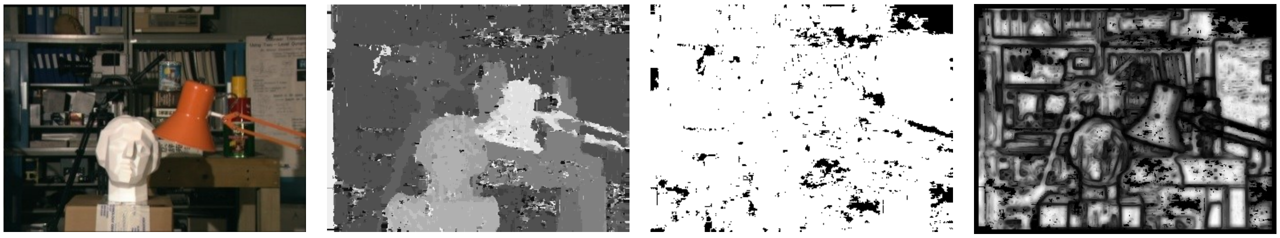
$$Cost_{agr}(p, d) = \sum\nolimits_{q \in N} Cost(q, d),$$

where p is coordinates of the current pixel, q is coordinates of support pixel around pixel p, N is rectangle aggregating set of support pixel s, *Cost* is the matching cost and $Cost_{agr}$ is a aggregated one. Usually, local stereo matching uses edge-preserving filters for improving accuracy and keeping object edges, however, the cost of this approach is higher than the box filter, obviously.

Then depth maps are computed with winner-takes-all (WTA) approach. Optimization step is ignored for real-time computation in this paper.

$$D_p = \underset{d}{argmin}\, Cost_{agr}(p, d),$$

The approach is simple and fast, although the accuracy of the depth map is not high. For iterative processing, however, the simple procedures are required.



(a) Natural image (b) Input depth map (c) Invalid mask (d) Weight map

Figure 2 Example of speckle mask and weight map.

3.2 Depth Map Refinement

To improve the accuracy of the estimated depth map, the depth map is refined in this step. We use a combination of the typical refinement approaches for detecting invalid region and merge them by using a weighed-edge-preserving filter. These stacks of post processing remove noises and reshape object boundary correctly.

To detect invalid region, we utilize well-known approaches of the LR consistency check [10], the uniqueness filtering [6] and the speckle or blob detection [33]. Usually, the LR consistency check judges whether the depth value is correct or not with the left and right depth maps. Then the method projects the depth map to another one and compares the difference between them. If the difference of a pixel is over the threshold, we mark the pixel invalid. We use an approximated version of the LR consistency check that requires only a left depth map. The implementation is in the OpenCV library [34]. Please see it for detail implementation. The uniqueness filter removes ambiguous matches via the aggregated cost volume. The approach checks difference between the minimized cost and the second minimized cost. If the difference under a threshold, the pixel is ambiguous; then we mark the pixel invalid. The speckle detection finds small speckles or blobs and then removes these regions. In texture-less region, the speckles of wrong matching tend to occur because of ambiguous matching. These matches are unnatural; thus we mark these region invalid. We use speckle detection function implemented in the OpenCV library. To make dense depth maps, the invalid regions are filled by valid value in neighborhoods. We search valid pixel for left and right scanline directions, and then we utilize the minimum one for filling value[10].

Finally, we use weighted joint bilateral filtering (WJBF) and as the depth map refinement filter [33]. The WJBF is a variant of the joint bilateral filter [26, 27]. The filter can set weight for each pixel. The WJBF is defined by:

$$D_p = \frac{\sum_{s\in N} R_s s(p,s) c\left(I_p, I_s\right) D_s}{\sum_{s\in N} R_s s(p,s) c\left(I_p, I_s\right)},$$

$$s(x,y) = exp\left(-\frac{||x-y||_2^2}{2{\sigma_s}^2}\right), c(x,y) = exp\left(-\frac{||x-y||_2^2}{2{\sigma_c}^2}\right),$$

where *p* and *s* are coordinates of a current pixel and of a support pixel, *I* is a natural image, *D* is a depth map, *N* is an aggregating set of support pixels, *s*() and *c*() are spacial and range kernels, respectively, and each weight function is the Gaussian distribution (σ_s;σ_c: const.). $\|\cdot\|_2$ denotes L2 norm, and R_s is a weight map value at the pixel *s*. The value of the weight *R* controls amount of the influence of each pixels in this image filtering. The weight map is defined by:

$$R_s = \sum\nolimits_{s\in N} M_s \cdot s(s,q) c\left(I_s, I_q\right) exp\left(-\frac{||D_s - D_q||_2^2}{2{\sigma_d}^2}\right),$$

where *q* is coordinates of support pixel around pixel *s*, M_s is an invalid region mask that is described in this section. The invalid mask has binary weight that is 0 or 1. We show the example of the invalid mask and the weight map in Fig. 2. Accelerating the weighted averaging filtering, the separable approximation [35] can be utilized for faster processing.

To remove subtle blurs around the object edges, we perform the joint nearest filter (JNF) [33]. With the JNF, blurred values are replaced by neighborhood non-blurred values. The JNF is defined by:

$$D_p^{JNF} = D_v^{Input},$$

$$s.t. \qquad v = \underset{s\in W}{argmin} \left\| D_p^{WJBF} - D_s^{Input} \right\|_2,$$

where D^x is the depth map estimated by a method $X \in \{Input, WJBF, JNF\}$, W is the aggregation set of a support pixel, and *v* is a pixel positon which minimizes the function. The JNF replaces the value blurred the WJBF as the nearest value in the support region on the before filtered version of D^{Input}.

3.3 Depth-Cost-Volume Generation

This step is a novel step in our proposed framework of the IFM for making a loop in the stereo matching framework. To feedback the initial cost volume in the matching cost step, we convert the refined depth map to a new cost volume named depth-cost-volume. The depth-cost-volume V consist of cost slices V_n $(n \in \{0, \dots, N-1\})$. N is the number of depth levels, i.e. 255 in the case of 8 bits. We define $V_{p,n}$ which is cost value on the pixel p with coordinates (*x, y*) in the n-th level cost slice V_n as follows:

$$V_{p,n} = L_x(n, l_p, \tau), 0 \leq n \leq N-1, x \in \{L1, L2, exp\},$$

$$\begin{cases} L_{L1} \coloneqq \frac{1}{\tau} \min\left(\left\|n - l_p\right\|_1, \tau\right), \\ L_{L2} \coloneqq \frac{1}{\tau^2} \min\left(\left\|n - l_p\right\|_2, \tau^2\right), \\ L_{exp} \coloneqq 1 - exp\left(-\frac{\left\|n - l_p\right\|_2}{2\tau^2}\right), \end{cases}$$

where L_x is a cost function for constructing each cost slice (e.g. L_{L1}, L_{L2} and L_{exp} are L1 norm, L2 norm and exponential function for cost slice generation, respectively), l_p is a depth value of the refined depth map at the pixel p, τ is a parameter for truncation threshold. A monotonically increasing function from the refined depth value is suitable for representing probability of depth map; thus we review L1 norm, L2 norm and exponential function as a representative. Other examples of the monotonically increasing functions are a sigmoid function and a cubed difference function. With preliminary experiments show that the L2 norm function for the depth cost volume generation step is the best and we utilize the L2 norm function it in this paper.

3.4 Cost Volume Blending

With this step, the depth cost volume is blended to the initial cost volume and then the steps become loopy. In this step, the initial and depth cost volumes are blended. In other word, we perform alpha blending each slice in the volumes:

$$Cost'_{p,n} = \alpha \cdot Cost_{p,n} + (1 - \alpha) \cdot V_{p,n}, 0 \leq n \leq N-1,$$

where *Cost'* is a blended matching cost, *Cost* is the initial cost volume, *V* is the depth cost volume, *p* is a pixel position, *n* is a depth value, and α is a blending parameter. The initial matching cost has usually ambiguity, e.g. in edge area and occlusion area. This blending feedback refines the ambiguity of initial matching; especially edge and occlusion area where is improved in the refinement step. At this time, the blend parameter is 0.5 in the paper.

3.5 Iterative Processing and Post Processing

In our frame fork of the IFM, each step is iterated with changing parameter. At the aggregation step, the kernel size of the box filter is smaller than the previous loop; therefore, the influence of neighborhood pixels is flooding with a coarse-to-fine manner.

In the last depth map refinement, named post processing, we add a subpixel interpolation step and binary range filtering step introduced by Yang [30]. The subpixel interpolation is defined as follow formula:

$$d' = d - \frac{Cost(d_+) - Cost(d_-)}{2(Cost(d_+) + Cost(d_-) - 2Cost(d))},$$

where *d'* is the interpolated depth value, *d* is the depth value of before interpolation, d_+ and d_- is (*d*+1) and (*d*-1). *Cost* is a matching cost volume function defined in Sec. 3.1. The binary range filtering is defined as follow formula:

$$D_p = \frac{\sum_{q \in N} g(p,q) D_q}{\sum_{q \in N} g(p,q)},$$

$$g(p,q) = \begin{cases} 1.0 & if \left|D_p - D_q\right| < 1 \\ 0 & else \end{cases}$$

In these steps, the depth map has sub-pixel accuracy. The cost of this stereo matching is linearly increasing by increasing handling depth level; thus we perform the sub-pixel interpolation step once as the final step.

(a) Tsukuba (b) Venus (c) Teddy (d) Cones

Figure. 3 Middlebury benchmark data sets.

Table. 1 Error rate of competitive method and proposed method with each relation. The row of "Before refinement" indicates the depth map generated after the first cost aggregation. The row of "After refinement" is the first refined depth map. In other words, 0-th loop of the proposed framework. The rows of number 1 to 5 is the *n*-th loop depth maps with the IFM.

Number of iteration	SGM	Before refinement	After refinement	1	2	3	4	5
Tsukuba	3.594	5.272	3.143	**2.724**	2.748	2.835	2.856	2.841
Venus	0.187	1.849	0.544	**0.080**	0.118	0.117	0.121	0.123
Teddy	7.39	9.556	7.098	6.287	4.81	4.151	3.807	**3.557**
Cones	3.63	4.392	3.201	2.825	2.748	2.678	**2.644**	2.686

4. EXPERIMENTAL RESULT

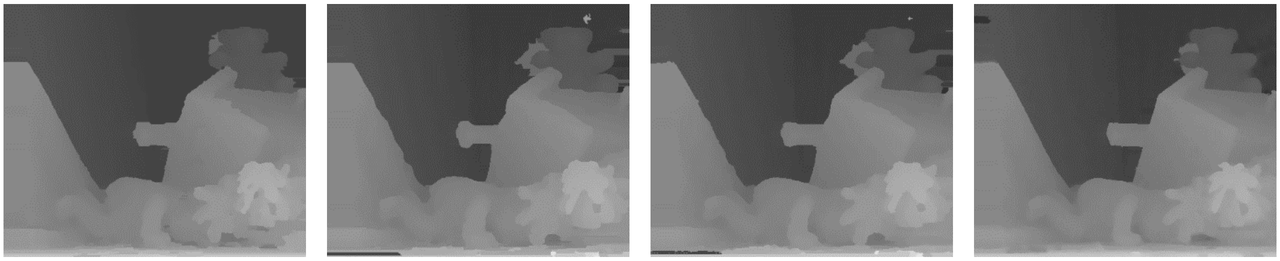
In our experiments, we evaluate the accuracy of the proposed framework of the IFM by using the Middlebury stereo benchmark [6]. Data sets are Tsukuba, Venus, Teddy and Cones (Fig. 3). Each image resolution and depth level is denoted in following manner (resolution; depth level): Tsukuba (384 × 288; 16), Venus (434 × 383; 32), Teddy and Cones (450×375; 64), respectively. We use the error rate [1] as the objective evaluation method and set to 1 for the error threshold in our experiments. Note that we only evaluate non-occluded regions in the depth maps.

The kernel radius for the cost aggregation is initially set to 5, and the radius gradually becomes small. We set the radiuses at each loop; 5, 3, 1, 1, 1. The kernel sizes of WJBF and JNF are 7 × 7 and 5 × 5, respectively. The other parameters are experimentally determined to maximize the resulting accuracy.

Table 1 shows the relations of the number of iteration and the error rate. SGM is Semi-Global Matching [10] that is a computationally efficient algorithm in the optimization method. We show that around 1.8 to 2.4 percent of the error rate decreases from the initial depth map by the IFM. Note that the number of iteration count of the max performance is different depending on the dataset. In addition, the error rate of the IFM is lower than the SGM results in any data sets.

In Fig. 4, Teddy's depth maps from each method are presented. The result of depth map (a) which is before refinement, i.e., just block matching, has fatting object edges and large error regions. After refinement (b), the error regions become small, and the object edges are corrected than the initial depth map. After 5 iteration via the IFM framework, the error regions become smaller, and object edges are more accurate than the previous two results. Comparing (a) and (d), the object edges of the IFM are more accurate than the SGM result. It is because that the feedback strategy to preserve the natural objects recursively.

The results of computational cost are shown in Tab. 2. Note that this result does not include the generation of the weight map. The implementations for our proposed and competition methods are written in C++ with Visual Studio 2010 on Windows 7 64 bit. The code is parallelized by Intel Threading Building Blocks and SSE vectorization. The CPU for the experiments is 3.50GHz Intel Core i7-3770K. As shown in Tab. 2, our method can compute faster than SGM until 4-th loop.



(a) SGM (b) Before refinement (c) After refinement (d) IFM (5-th)

Figure. 4 Results of depth map with each method.

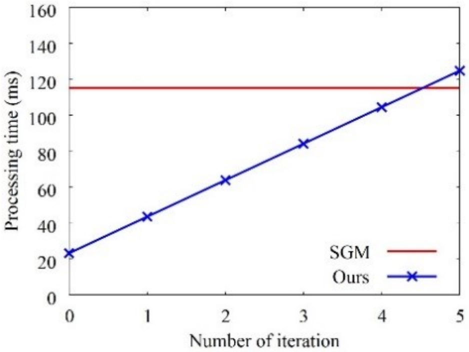


Figure. 5 Processing time of each method. The input image is Teddy.

4. CONCLUSION

In this paper, we proposed a novel framework called in-loop feedback matching (IFM) for edge-aware local stereo matching. Instead of using multiple edge-preserving filters for a number of slices as the usual approach, we use the edge-preserving filter for an estimated depth map and feedback the edge-awareness to the cost volume. To make the feedback, the proposed framework contains two additional steps; depth cost volume generation and cost volume blending steps; then, the IFM framework iterates each step. Experimental results show that the proposed framework generates accurate depth map with low cost.

REFERENCES

[1] Tanimoto, M., Tehrani, M.P., Fujii, T., and Yendo, T., "Free-Viewpoint TV," *IEEE Signal Processing Magazine*, 28(1), 67-76 (2011).

[2] Mori, Y., Fukushima, N., Yendo, T., Fujii, T., and Tanimoto, M., "View Generation with 3D Warping Using Depth Information for FTV," *Signal Processing: Image Communication*, 24(1-2), 65-72 (2009).

[3] Kodera, N., Fukushima, N., and Ishibashi, Y., "Filter based alpha matting for depth image based rendering," in *Proc. IEEE Visual Communications and Image Processing (VCIP)* (2013).

[4] Fukushima, N., Inoue, T., and Ishibashi, Y., "Removing Depth Map Coding Distortion by Using Post Filter Set," in *Proc. IEEE International Conference on Multimedia and Expo (ICME)* (2013).

[5] Inoue, T., Fukushima, N., and Ishibashi, Y., "Non-essentiality of Correlation between Image and Depth Map in Free Viewpoint Image Coding: Accurate Depth Map Case," in *Proc. 3DTV-CON* (2014).

[6] Scharstein, D., and Szeliski, R., "A taxonomy and evaluation of depth two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, 47(1), 7–42 (2002).

[7] Ohta, Y., and Kanade, T.,"Stereo by intra- and inter-scanline search using dynamic programming," *IEEE Trans. Pattern Analysis and Machine Intelligent*, 7(2), 139–154 (1985).

[8] Fukushima, N., Fujii, T., Ishibashi, Y., Yendo, T., and Tanimoto, M., "Real-time free viewpoint image rendering by using fast multi-pass dynamic programming," in *Proc. 3DTV-CON*, 1–4 (2010).

[9] Fukushima, N., Yendo, T., Fujii, T., and Tanimoto, M., "Free viewpoint image generation using multi-pass dynamic programming," in *Proc. SPIE, Stereoscopic Displays and Virtual Reality Systems XIV*, 64901F (2007).

[10] Hirschmuller, H., “Stereo processing by semiglobal matching and mutual information,” *IEEE Trans. on Pattern Analysis and Machine Intelligent*, 30(2), 328–341 (2008).

[11] Sun, J., Zheng, N. N., and Shum, H. Y., “Stereo matching using belief propagation,” *IEEE Trans. on Pattern Analysis and Machine Intelligent*, 25(7), 787–800 (2003).

[12] Boykov, Y., Veksler, O., and Zabih, R., “Fast approximate energy minimization via graph cuts,” *IEEE Trans. on Pattern Analysis and Machine Intelligent*, 23(11), 1222–1239 (2001).

[13] Wildeboer, M. O., Fukushima, N., Yendo, T., and Tehrani, M. P., Fujii, T. and Tanimoto, M., “A semi-automatic multi-view depth estimation method,” in *Proc. SPIE, Visual Communications and Image Processing*, 7744 (2010).

[14] Tanimoto, M., Fujii, T., Suzuki, K., Fukushima, N., and Mori, Y., “Reference softwares for depth estimation and view synthesis.” *ISO/IEC JTC1/SC29/WG11 MPEG 20081*, M15377 (2008).

[15] Pham, T.Q., and Van Vliet, L.J., “Separable bilateral filtering for fast video preprocessing,” in *Proc. IEEE International Conference on Multimedia and Expo (ICME)*, (2005).

[16] Yang, Q., Han, K.H., and Ahuja, N., “Real-time O(1) bilateral filtering,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 557-564 (2009).

[17] He, K., Sun, J., and Tang, X., “Guided image filtering,” *IEEE Trans. on Pattern Analysis and Machine Intelligent*, 35(6), 1397-1409 (2013.)

[18] Gastal, E.S.L., and Oliveira, M.M., “Domain transform for edge-aware image and video processing,” *ACM Trans. on Graphics*, 30(4) (2011).

[19] Zhang, K., Lu, J., and Lafruit, G., “Cross-Based Local Stereo Matching Using Orthogonal Integral Images,” *IEEE Trans. on Circuits and Systems for Video Technology*, 19(7), 1073-1079 (2009).

[20] Yoon, K.-J., and Kweon. I.-S., “Adaptive support-weight approach for correspondence search.” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(4), 650-656 (2006).

[21] Hosni, A., Bleyer, M., Gelautz, M., and Rhemann, C., “Local stereo matching using geodesic support weights.” in *Proc. IEEE International Conference on Image Processing (ICIP)*, 2093-2096 (2009).

[22] Rhemann, C., Hosni, A., Bleyer, M., Rother, C., and Gelautz, M.. “Fast cost-volume filtering for visual correspondence and beyond,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3017 - 3024 (2011).

[23] Yang, Q., “A non-local cost aggregation method for stereo matching,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1402-1409 (2012).

[24] Pham, C.C., and Jeon, J. W., “Domain Transformation-Based Efficient Cost Aggregation for Local Stereo Matching,” *IEEE Trans on Circuits and Systems for Video Technology*, 23(7), 1119-1130 (2013).

[25] Tomasi, C., and Manduchi, R., “Bilateral filtering for gray and color image,” in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 839–846 (1998).

[26] Pestschnigg, G., Szeliski, R., Agrawala, M., Cohen, M., Hoppe, H., and Toyama, K., “Digital photography with flash and no-flash image pairs,” *ACM Trans. on Graphics*, 23(3), 664–672 (2004).

[27] Eisemann, E., and Durand, F., “Flash photography enhancement via intrinsic relighting,” *ACM Trans. on Graphics*, 23(3), 673–678 (2004).

[28] Kopf, J., Cohen, M., Lischinski, D., and Uyttendaele, M., “Joint bilateral upsampling,” *ACM Trans. on Graphics*, 26(3) (2007).

[29] Lai, P. L., Tian, D., and Lopez, P., “Depth map processing with iterative joint multilateral filtering,” in *Proc. Picture Coding Symposium (PCS)*, 9–12 (2010).

[30] Yang, Q., Yang, R., Davis, J., and Nister, D., “Spatial depth super resolution for range images,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–8 (2007).

[31] Fujita, S., Matsuo, T., Fukushima, N., and Ishibashi, Y., “Cost volume refinement filter for post filtering of visual corresponding,” in *Proc. SPIE, Image Processing: Algorithms and Systems XIII* (2015).

[32] Yang, Q., Wang, L., and Ahuja, N., “A constant-space belief propagation algorithm for stereo matching,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1458–1465 (2010).

[33] Matsuo, T., Fukushima, N., and Ishibashi, Y., “Weighted joint bilateral filter with slope depth compensation filter for depth map refinement,” in *Proc. International Conference on Computer Vision Theory and Applications (VISAPP)* (2013).

[34] Bradski, G., and Kaehler, A., [Learning OpenCV: Computer vision with the OpenCV library,] *O'Reilly Media, Inc.* (2008).

[35] Fukushima, N., Fujita, S., and Ishibashi, Y., “Switching dual kernels for separable edge-preserving filtering,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2015).