SELF-SIMILARITY MATCHING WITH PREDICTIVE LINEAR UPSAMPLING FOR DEPTH MAP

*Norishige Fukushima**

Nagoya Institute of Technology, Japan

Kouta Takeuchi, Akira Kojima

Nippon Telegraph and Telephone Corporation, Japan

ABSTRACT

We propose a real-time upsampling scheme for depth maps. The proposed scheme contains two upsampling stages; one is self-similarity matching (SSM), and the other is predictive linear upsampling (PLU). SSM accelerates cost volume filtering by using a variant of joint bilateral upsampling, which utilizes high-dimensional vectors, which is neighborhoods of an RGB image and a depth map. The high-dimensional upsampling suppresses edge blurring and scattering problems. PLU generates smooth surfaces with keeping edges guided by the results of SSM. Experimental results show that the proposed scheme has higher accuracy than the state-of-the-art upsampling. Additionally, the proposed method has real-time performance on a multi-core CPU.

Index Terms **—** depth map upsampling, edge-aware upsampling, cost volume filtering, cost volume upsampling

1. INTRODUCTION

RGB-D camera has important for computer vision and image processing researches. Depth sensors usually have lower-resolution size than RGB cameras; thus, depth map upsampling is necessary. Moreover, with subsampling, depth map upsampling is utilized for acceleration of depth map estimation, and coding of 3D video [1].

Recently, many depth upsampling algorithms, which jointly utilize RGB and depth images, are proposed; thus, we classify these methods into three types based on computational complexity; the upsampling type, the refinement filtering type, and the cost volume filtering/optimization type. The upsampling type [2, 3, 4, 5, 6] is the fastest approaches among the three types. The computational complexity depends on the low-resolution size S_L of depth maps, i.e., $O(S_L)$. The accuracy of these methods, however, are insufficient. Upsampled flat regions are contaminated by transferring textures in an RGB image, and upsampled edges are blurring or scattering due to incoherence between the RGB image and the depth map. The refinement filtering type [5, 7, 8] mitigates these problems. The computational order is usually $O\left(S_H\right)$, where S_H is the high-resolution size of RGB images. The cost volume filtering/optimization type [9, 10, 11, 12, 13, 14, 15] performs sophisticated filters or optimizations for the cost volume constructed from low-resolution depth maps. These algorithms further improve accuracy, but the order is $O(T \cdot S_H)$, where T is the number of the depth candidates. Consequently, they are off-line processing. Usually, upsampling is one of preprocessing; thus, reducing computational costs is important. The upsampling type is suitable for this purpose; however, the type has not enough accuracy in the flat surface and the object edge regions.

*This work was partially supported by JSPS KAKENHI Grant Number 15K16023.

In this paper, we propose an accelerated method for cost volume filtering to reduce the order to $O(T \cdot S_L)$. Instead of upsampling initial depth maps before cost volume construction, a low-resolution cost volume is built, and then the cost volume is jointly upsampled. The proposed scheme contains two upsampling methods, named self-similarity matching (SSM) and predictive linear upsampling (PLU). SSM upsamples a low-resolution cost volume by high-dimensional Gaussian upsampling, which is edge-aware upsampling with neighborhood vectors. Then, we can generate a piecewise-flat depth map without edge blurring, texture copying, and edge scattering problems. In addition, PLU refines the results of SSM to have a piecewise-linear depth map. Moreover, cooperating with depth map denoising, we can reduce a kernel radius and can implement efficiently by utilizing sparsity of the cost volume.

2. RELATED WORK

In this section, we briefly review related works. Joint bilateral upsampling (JBU) [2] is an early work of edge-aware upsampling. JBU utilizes a high-resolution image as guidance and then performs weighted averaging by using bilateral weights from the image. While JBU has a texture transferring problem, an extension of noise-aware filtering for depth map upsampling [3] realizes robust upsampling for noises. As an acceleration, upsampling with guided filtering [4] is one of the fastest upsampling. These approaches are low complexity; however, edge blurring is inevitable. Recently, a weighted histogram-based approach of weighted mode filtering [5] solves the blurring issue, and joint geodesic upsampling [6], which uses geodesic distance instead of Euclidean distance, prevents mixing pixels. Both methods estimate accurate depth maps but these methods consume computational costs; however, these approach is relatively faster than the refinement filtering and the cost volume filtering/optimization types.

The refinement filtering type uses fast edge-preserving filtering [4, 8] for refining pre-upsampled depth maps. Joint bilateral/ trilateral filtering with a weight map [7, 16] is also proposed for suppressing texture-transferring and edge-blurring problems. The weighted mode filtering [5] can also solve these problems.

With the approach of the cost volume filtering/optimization type, the cost volume is constructed from a temporally upsampled depth map, and then the cost volume is filtered or optimized. The bilateral cost volume filtering [9] aggregates costs by using bilateral filtering. After that, we can obtain a sharp depth map by minimizing the cost volume with a winner-takes-all manner. The variant that uses guided filtering [10] and the general framework of the cost volume refinement is also proposed [11]. The Markov random field (MRF) optimization for the cost volume is firstly proposed by [12]. Numerous approaches improve the MRF optimization for depth map upsampling [13, 14, 15]. MRF approaches are accurate but much slower than the filtering approaches.

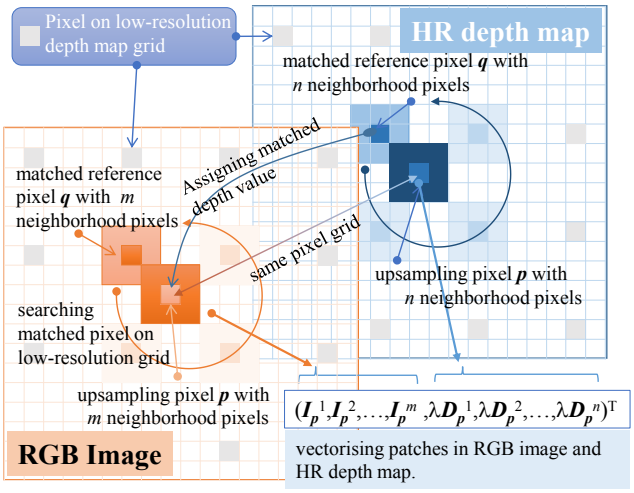

Figure 1. Self-similarity matching. Neighborhood pixels around a current pixel in an RGB image and a high-resolution (HR) depth map is vectorized. HR depth map is temporally upsampled by bicubic. Then the vector is compared with a vector of reference pixels at the nearest samples on a low-resolution grid to obtain a matched value.

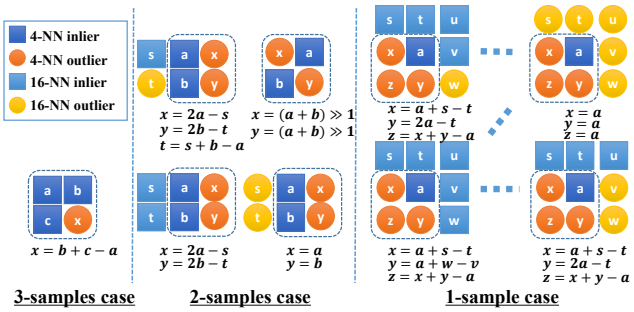

Figure 2. Examples of prediction of outliers samples in predictive linear upsampling. We predict samples by using 4-NN or 16-NN pixels. In the 3-samples case, we have other three rotated cases. In the 2-sample case, we have other three rotated cases for the line pattern, and two cases for the diagonal pattern. In the 1-sample case, we cannot show all cases due to the page limit.

3. DEPTH MAP UPSAMPLING SCHEME

In this section, we present the proposed scheme of self-similarity matching (SSM) and predictive linear upsampling (PLU). Also, denoising for depth maps is discussed.

3.1. Self-Similarity Matching

With SSM, we search a similar patch in the high-resolution grid of the self-image, and then we assign an almost depth value of the matched patches to the upsampling pixel. The idea is inspired from cost volume filtering [11, 17]. Figure 1 shows the overview of SSM. Next, we introduce SSM according to cost volume filtering manners.

Let $D_{\downarrow}$ be a low-resolution depth map. Also, let $\boldsymbol{p}$ be a current pixel position on the high-resolution grid. The symbol of $\downarrow$ is a downsampling operator; hence, $\boldsymbol{p}_{\downarrow}$ is the relative pixel position of $\boldsymbol{p}$ on the low-resolution grid. In a step of cost generation, a matching cost function is defined in the low-resolution grid:

$$C_{\downarrow \boldsymbol{p}_{\downarrow}}(d) = 1 - \exp\left(\frac{\|D_{\downarrow \boldsymbol{p}_{\downarrow}} - d\|_2^2}{-2\sigma_c^2}\right), \tag{1}$$

where d is a depth value candidate, and $\|\cdot\|_2$ denotes a L_2 norm operator. $D_{\downarrow \boldsymbol{p}_{\downarrow}}$ is a depth value of the low-resolution depth map $D_{\downarrow}$ on a pixel $\boldsymbol{p}_{\downarrow}$.

Then, we upsample the low-resolution cost volume with edge-aware aggregation. Let $\boldsymbol{q}$ and $\boldsymbol{q}_{\downarrow}$ be supported pixel positions on the high-resolution grid and the relative pixel position of $\boldsymbol{q}$ on the low-resolution grid. Similar to joint bilateral upsampling [2][1], cost slices in the cost volume are upsampled. Note that we use high-dimensional Gaussian distribution [18, 19] to prevent the edge scattering issue. To generate the high-dimensional signals, we utilize patches of the RGB image and also utilize patches of the depth map. The high-resolution cost is:

$$C_{\boldsymbol{p}}(d) = \sum_{\boldsymbol{q}_{\downarrow} \in S} f^{\mathrm{SSM}}(\boldsymbol{p}, \boldsymbol{q}) C_{\downarrow \boldsymbol{q}_{\downarrow}}(d), \tag{2}$$

$$f^{\mathrm{SSM}}(\boldsymbol{p},\boldsymbol{q}) = \exp\left(\frac{\|\boldsymbol{p}-\boldsymbol{q}\|_2^2}{-2\sigma_s^2}\right) \exp\left(\frac{\|\boldsymbol{v}(\mathcal{N}_{\boldsymbol{p}}) - \boldsymbol{v}(\mathcal{N}_{\boldsymbol{q}}))\|_2^2}{-2\sigma_h^2}\right), \tag{3}$$

[1]JBU is defined as: $D_{\boldsymbol{p}}^{\mathrm{jbu}} = \sum_{q_{\downarrow} \in N} f^{bi}_{\sigma_s,\sigma_c}(\boldsymbol{p}, \boldsymbol{q}) D_{\downarrow \boldsymbol{q}_{\downarrow}}$, where f^{bi} is a bilateral Gaussian weight. JBU performs weighted averaging for depth maps directly; thus, edge-blurring is inevitable.

where $\sigma_{s,h}$ are control parameters for spatial and similarity Gaussian distribution, respectively. $\mathcal{N}_{\boldsymbol{p},\boldsymbol{q}}$ are sets of neighborhood pixels around the pixel $\boldsymbol{p}, \boldsymbol{q}$. A vectorized function $\boldsymbol{v}(\cdot)$ gathers pixels in a patch of the image I and of the depth map $\hat{D}$ on the high-resolution grid. $\hat{D}$ is temporally upsampled in advance by using bicubic. For this upsampling simple and smooth upsampling is desirable. Note that the depth values are balanced by the parameter λ between the intensity of the RGB image and the depth map. Thereby, the vector using m neighborhoods in the RGB image and n neighborhoods in the depth map is represented as:

$$\boldsymbol{v}(\mathcal{N}_{\boldsymbol{p}}) = (I_{\boldsymbol{p}}^1, I_{\boldsymbol{p}}^2, ..., I_{\boldsymbol{p}}^m, \lambda \hat{D}_{\boldsymbol{p}}^1, \lambda \hat{D}_{\boldsymbol{p}}^2, ..., \lambda \hat{D}_{\boldsymbol{p}}^n)^T. \tag{4}$$

Finally, the resulting depth map of SSM is obtained by minimizing the upsampled cost:

$$D_{\boldsymbol{p}}^{\mathrm{ssm}} = \arg\min_{d \in T} C_{\boldsymbol{p}}(d), \tag{5}$$

where T is a set of depth value candidates, e.g., [0:255].

The matching process of SSM completely prevents blurring and moderates texture-copying, however, the resulting depth map is piecewise-flat. In the next section, we refine the depth map to be piecewise-smooth.

3.2. Predictive Linear Upsampling

In PLU, we do not use an RGB image as guidance to completely prevent texture-transferring artifacts. In addition, we use predictive linear upsampling guided by the resulting SSM to have smooth surfaces with sharp object edges.

Let $D_{\boldsymbol{p}}^{\mathrm{lt}}, D_{\boldsymbol{p}}^{\mathrm{rt}}, D_{\boldsymbol{p}}^{\mathrm{lb}}, D_{\boldsymbol{p}}^{\mathrm{rb}}$ be depth values of the 4-nearest neighborhood pixels (4-NN) for the upsampling pixel $\boldsymbol{p} = (x, y)$. In principle, we use simple linear upsampling with the neighborhoods for estimating value $D_{\boldsymbol{p}}^{\mathrm{PLU}}$:

$$D_{\boldsymbol{p}}^{\mathrm{PLU}} = \alpha\beta D_{\boldsymbol{p}}^{\mathrm{lt}} + \bar{\alpha}\beta D_{\boldsymbol{p}}^{\mathrm{rt}} + \alpha\bar{\beta} D_{\boldsymbol{p}}^{\mathrm{lb}} + \bar{\alpha}\bar{\beta} D_{\boldsymbol{p}}^{\mathrm{rb}}, \tag{6}$$

where $\alpha, \bar{\alpha}, \beta, \bar{\beta}$ are coefficients for the linear upsampling; $\alpha = (x - x^{\mathrm{lt}})/(x^{\mathrm{rt}} - x^{\mathrm{rt}})$, $\bar{\alpha} = 1 - \alpha$, and $\beta = (y - y^{\mathrm{lt}})/(y^{\mathrm{lt}} - y^{\mathrm{lb}})$, $\bar{\beta} = 1 - \beta$. $x, y^{lt,rt,lb,rb}$ are the x, y coordinates of 4-NN pixels.

The 4-NN pixels are the nearest pixels from $\boldsymbol{p}_{\downarrow}$ in the low-resolution depth map $D_{\downarrow}$, only if the target and neighborhood pixels are on the same object. When the 4-NN pixels contain outliers in the region, we use predicted values $\hat{D}_{\boldsymbol{p}}^{pr}$ instead of the outliers for linear upsampling. The event will happen around object boundaries. To detect outliers in the high-resolution grid, we utilize the upsampled depth map with SSM as guidance. When we regard pixels as outliers, the difference between the depth value of the target pixel $\boldsymbol{p}$ and the 4-NN pixels in the low-resolution depth

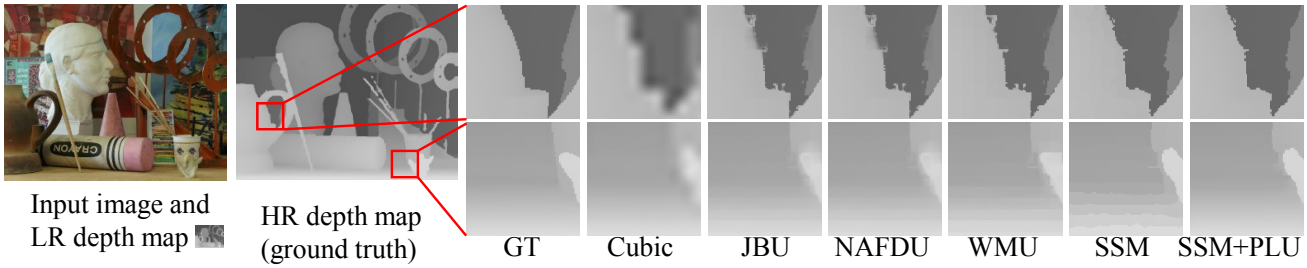

Figure 3. Depth map upsampling results (*Art* dataset). Ratio of upsampling is $\times 8$.

Table 1. Bad pixel ratio in each upsampling [20]. Depth error threshold is 1. The size of datasets is half resolution (704×560).

noise	method	Aloe			Art			Moebius			Dolls			Reindeer		
		x2	x4	x8	x2	x4	x8	x2	x4	x8	x2	x4	x8	x2	x4	x8
σ=0	JBU	6.23	8.41	9.51	1.49	6.82	10.05	2.54	5.19	9.96	2.54	5.41	12.51	2.51	4.66	8.87
	NAFDU	1.62	5.75	8.93	1.35	3.25	5.28	2.28	3.16	9.56	2.28	4.80	7.06	1.17	2.88	6.37
	WMU	0.97	2.60	6.56	0.67	1.55	6.45	0.88	2.24	5.37	1.16	1.88	5.71	0.58	1.57	3.77
	SSM	1.03	2.62	6.29	0.69	2.76	6.61	0.89	2.32	5.11	1.26	2.07	6.07	0.58	1.47	3.79
	PROP	**0.74**	**1.68**	**4.33**	**0.65**	**1.28**	**3.83**	**0.71**	**1.60**	**4.69**	**0.70**	**1.54**	**4.29**	**0.45**	**1.03**	**2.77**
σ=3	JBU	7.76	13.87	12.40	6.94	11.07	23.14	5.31	9.33	15.99	4.32	8.93	17.66	3.55	6.62	13.50
	NAFDU	2.63	8.72	11.81	3.31	7.02	14.82	4.53	6.82	14.17	3.98	8.19	15.71	1.93	4.54	10.18
	WMU	2.13	6.89	11.78	2.91	6.91	14.62	4.59	6.68	14.25	3.61	7.85	16.04	1.99	3.75	9.67
	BRF+PROP	**1.76**	**4.89**	**10.87**	**1.48**	**5.03**	**9.17**	**2.03**	**4.93**	**9.89**	**2.80**	**6.06**	**13.93**	**1.31**	**3.38**	**8.16**

map is over a depth threshold of the pixel $T_{\boldsymbol{p}}^D$. Thereby, the depth values of the 4-NN pixels are defined by:

$$D_{\boldsymbol{p}}^l = \begin{cases} D_{\downarrow \boldsymbol{p}\downarrow}^l & |D_{\boldsymbol{p}}^{\mathrm{SSM}} - D_{\downarrow \boldsymbol{p}\downarrow}^l| < T_{\boldsymbol{p}}^D \\ \hat{D}_{\boldsymbol{p}}^{pr} & otherwise \end{cases} \tag{7}$$
$$l \in \{\mathrm{lt}, \mathrm{rt}, \mathrm{lb}, \mathrm{rb}\}.$$

The depth threshold $T_{\boldsymbol{p}}^D$ switches two state of thresholds T_{flat}^D and T_{edge}^D for flat and edge areas according to the second derivative of the low-resolution depth map, which indicates possibilities of object edges. Therefore, we switch the depth threshold by using a threshold for the second derivative of the low-resolution depth map $T^{D''_{\downarrow}}$. The depth threshold $T_{\boldsymbol{p}}^D$ is defined by:

$$T_{\boldsymbol{p}}^D = \begin{cases} T_{\mathrm{flat}}^D & \max(|\frac{\partial^2}{\partial^2 x} D_{\downarrow \boldsymbol{p}\downarrow}|, |\frac{\partial^2}{\partial^2 y} D_{\downarrow \boldsymbol{p}\downarrow}|) < T^{D''_{\downarrow}} \\ T_{\mathrm{edge}}^D & otherwise \end{cases}, \tag{8}$$

where $\frac{\partial^2}{\partial^2 x} D_{\downarrow \boldsymbol{p}\downarrow}$ and $\frac{\partial^2}{\partial^2 y} D_{\downarrow \boldsymbol{p}\downarrow}$ are the second derivative of depth map in the x and y directions, respectively. We use the maximum value of the derivatives.

After detecting outliers in the 4-NN pixels, we predict values for outliers. The prediction method varies with the number of outliers. The examples of the predictions are depicted in Fig. 2. If we have full samples, we do not need a prediction. In the case of 3 samples case, we predict a missing sample by a plane prediction. If we have 1 or 2 sample(s), we use 16-nearest neighborhoods for the prediction. All predictions can be made only with the addition and bit-shift operators.

3.3. Depth Map Denoising

The conventional works convolute noisy depth maps by using joint images. However, innovation of time of flight (ToF) depth sensors, such as Kinect v2, reduces noises of depth maps dramatically. A half decade ago, noise standard deviation of depth maps, which are ranged in 0 to 255, were from 10 to 20; however, current sensor's one is under 3. The fact indicates that direct filtering is better than joint filtering for denoising because self-signals has higher S/N and correlations.

Therefore, we use direct filtering of binary weighted range filter (BRF) [21] as prefiltering for depth maps. BRF is a simplified bilateral filter or a variant of ϵ-filter [22]. The spatial kernel of

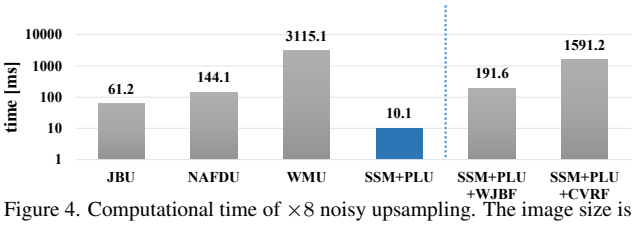

Figure 4. Computational time of $\times 8$ noisy upsampling. The image size is 88×70 to 704×560 upsampling. We use Xeon X5690 3.47Ghz (12 core CPU)

BRF is a box type, and range kernel is binary defined by a threshold. The BRF is quite fast and enough denoising performance for depth maps.

With BRF, we can separate the denoising issue and the upsampling issue. Thus, after denoising, we can set small kernel for cost volume upsampling. In our setup, we use only 4-NN samples and 5 depth candidates ($\sigma_c = 1$ with clipping 3σ) for cost volume upsampling. Using voxels in the cost volume is very sparse; thus, we compute only valid voxels instead of filtering the whole volume. The number of filtering samples is the almost same as JBU with 16-NN samples; thus, computational cost is quite low.

4. EXPERIMENTAL RESULTS

We compare the proposed scheme (PROP: SSM+PLU), SSM only and three upsampling methods with/without noise conditions. They are joint bilateral upsampling (JBU) [2], noise-aware filter for depth upsampling (NAFDU) [3] and weighted mode filter upsampling (WMU) [5]. These competitive algorithms belong to the upsampling type; consequently, the computational cost is low. We use five test images in Middlebury stereo datasets [23, 24]. For our simulation, we generate low-resolution depth maps from the ground truth by using nearest neighbor sampling. The parameters in this experiments are as follows; $\sigma_h = 13$, $\sigma_c = 1$, $\sigma_s = 0.5r$, $\lambda = 2$, $m = 5$, $n = 1$, $T^{D''} = 5 + \log r$, $T_{flat}^D = 10 + \log r + 1$, $T_{edge}^D = 5 + \log r + 1$, where r is a mugnification factor.

Table 1 shows the error ratio between each upsampling and the ground truth depth maps with/without noises. The proposed upsampling has the best performance. Figure 3 shows a visual comparison in *Art*. With SSM, we can keep the edge without blurring and scattering pixels, while we cannot smooth the slant sur-

faces. SSM+PLU can smooth the part with keeping the edge. In the noisy case, the proposed method with BRF prefiltering has the best performance.

Figure 4 shows the computational time of each upsampling, and also show the methods in the other type; weighted joint bilateral filtering (WJBF) [7] and cost volume refinement filtering (CVRF) [9]. Note that the y-axis is a logarithmic scale. The proposed scheme is the fastest in the upsampling type and has real-time performance. As well, the proposed scheme is quite faster than the other, because, the conventional works need large kernel from joint images to handle denoising issues while BRF takes only 0.8 ms with two iterations of 3×3 kernel filtering.

Replacing an initial depth value from cubic upsampling to the proposed scheme for refinement filtering or cost volume filtering further improves the accuracy of depth maps. For example, in the *Art* dataset with the $\times 8$ upsampling case, bad pixel ratio was 3.71 with WJBF, and 3.69 with CVRF. Using bicubic for initial depth map reported in [9], the results of WJBF and CVRF were 14.7 and 12.8, respectively. Thus, our upsampling result is effective for an initial depth map in these refinement and cost volume filtering/optimization.

5. CONCLUSION

In this paper, we proposed real-time depth map upsampling, which accelerates cost volume filtering. The upsampling had two steps; one was self-similarity matching, and the other was predictive linear upsampling. Experimental results showed that SSM could upsample depth maps without blurring and scattering edges, and then PLU guided with SSM results was the best performance among state-of-the-art methods with/without noise conditions. The proposed method run in real-time with a multi-core CPU. It took about only 10.1 ms in $\times 8$ upsampling.

6. REFERENCES

[1] K. Muller, H. Schwarz, D. Marpe, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, P. Merkle, F. H. Rhee, G. Tech, M. Winken, and T. Wiegand, "3d high-efficiency video coding for multi-view video and depth data," *IEEE Transactions on Image Processing*, vol. 22, no. 9, pp. 3366–3378, 2013.

[2] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, "Joint bilateral upsampling," *ACM Transactions on Graphics*, vol. 26, no. 3, 2007.

[3] D. Chan, H. Buisman, C. Theobalt, and S. Thrun, "A noise-aware filter for real-time depth upsampling," in *Proc. Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*, 2008.

[4] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1397–1409, 2013.

[5] D. Min, J. Lu, and M.N. Do, "Depth video enhancement based on weighted mode filtering," *IEEE Transactions on Image Processing*, vol. 21, no. 3, pp. 1176–1190, 2012.

[6] M.-Y. Liu, O. Tuzel, and Y. Taguchi, "Joint geodesic upsampling of depth images," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[7] T. Matsuo, N. Fukushima, and Y. Ishibashi, "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.

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

[9] Q. Yang, R. Yang, J. Davis, and D. Nister, "Spatial-depth super resolution for range images," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[10] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 504 – 511, 2013.

[11] S. Fujita, T. Matsuo, N. Fukushima, and Y. Ishibashi, "Cost volume refinement filter for post filtering of visual corresponding," in *Proc. SPIE*, 2015.

[12] J. Diebel and S. Thrun, "An application of markov random fields to range sensing," in *Proc. Advances in neural information processing systems (NIPS)*, 2005.

[13] J. Lu, D. Min, R.S. Pahwa, and M.N. Do, "A revisit to mrf-based depth map super-resolution and enhancement," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011.

[14] D. Kim and K.-J. Yoon, "High-quality depth map upsampling robust to edge noise of range sensors," in *Proc. IEEE International Conference on Image Processing (ICIP)*, 2012.

[15] K.-H. Lo, K.-L. Hua, and Y.-C.F. Wang, "Depth map super-resolution via markov random fields without texture-copying artifacts," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

[16] K.-H. Lo, Y.-C.F Wang, and K.-L. Hua, "Joint trilateral filtering for depth map super-resolution," in *Proc. IEEE Visual Communications and Image Processing (VCIP)*, 2013.

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

[18] A. Adams, N. Gelfand, J. Dolson, and M. Levoy, "Gaussian kd-trees for fast high-dimensional filtering," *ACM Transactions on Graphics*, vol. 28, no. 3, pp. 21, 2009.

[19] A. Adams, J. Baek, and M. A. Davis, "Fast high-dimensional filtering using the permutohedral lattice," *Computer Graphics Forum*, vol. 29, no. 2, pp. 753–762, 2010.

[20] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[21] N. Fukushima, T. Inoue, and Y. Ishibashi, "Removing depth map coding distortion by using post filter set," in *Proc. IEEE International Conference on Multimedia and Expo (ICME)*, 2013.

[22] H. Watabe, Y. Arakawa, and K. Arakawa, "Nonlinear filters for multimedia applications," in *Proc. IEEE International Conference on Image Processing (ICIP)*, 1999.

[23] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.

[24] H. Hirschmüller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.