Combined Bilateral Filter for Enhanced Real-Time Upsampling of Depth Images

Oliver Wasenmüller, Gabriele Bleser and Didier Stricker
Germany Research Center for Artificial Intelligence (DFKI), Trippstadter Str. 122, Kaiserslautern, Germany
{oliver.wasenmueller, gabriele.bleser, didier.stricker}@dfki.de

Keywords: Depth Image, Upsampling, Joint Bilateral Filter, Texture Copying, Depth Discontinuity, Sensor Fusion.

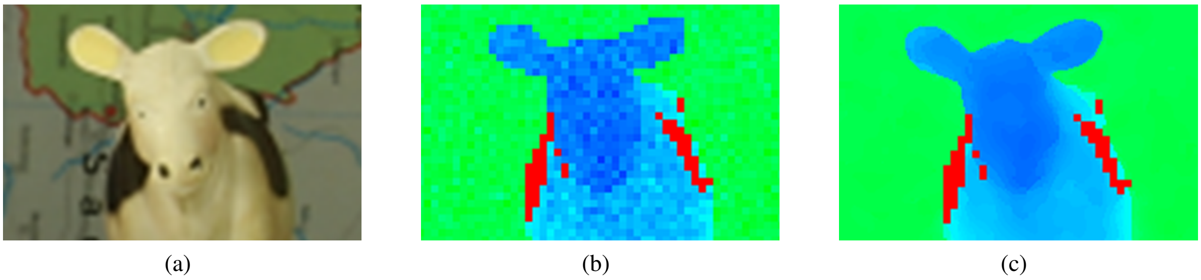
Abstract: Highly accurate depth images at video frame rate are required in many areas of computer vision, such as 3D reconstruction, 3D video capturing or manufacturing. Nowadays low cost depth cameras, which deliver a high frame rate, are widely spread but suffer from a high level of noise and a low resolution. Thus, a sophisticated real time upsampling algorithm is strongly required. In this paper we propose a new sensor fusion approach called Combined Bilateral Filter (CBF) together with the new Depth Discontinuity Preservation (DDP) post processing, which combine the information of a depth and a color sensor. Thereby we especially focus on two drawbacks that are common in related algorithms namely texture copying and upsampling without depth discontinuity preservation. The output of our algorithm is a higher resolution depth image with essentially reduced noise, no aliasing effects, no texture copying and very sharply preserved edges. In a ground truth comparison our algorithm was able to reduce the mean error up to 73% within around 30ms. Furthermore, we compare our method against other state of the art algorithms and obtain superior results.

1 INTRODUCTION

Within recent years low-cost depth cameras are widely spread. These cameras are able to measure the distance from the camera to an object for each pixel at video frame rate. There are two common technologies to measure depth. Time-of-Flight (ToF) cameras, such as the SwissRanger or the Microsoft Kinect v2, emit light and measure the time the light takes to return to the sensor. Structured light cameras, such as the Microsoft Kinect v1 or the Occipital Structure, project a pattern into the scene and capture the local disparity of the pattern in the image. In all these cameras high speed and low price are accompanied by low resolution (e.g. Kinect v2: $512 \times 484 px$) and a high level of noise. However, their application areas, such as 3D reconstruction of non-rigid objects, 3D video capturing or manufacturing, require both high quality depth estimates and fast capturing. High-end devices, such as laser scanners, provide extremely accurate depth measures, but are usually expensive and only able to measure a single point at a time. Thus, a sophisticated filter algorithm to increase the resolution and decrease the noise level of depth images in real-time is strongly required.

In this paper we propose an enhanced real-time iterative upsampling, which we call Combined Bilateral Filter (CBF). Our new algorithm is based on a fusion of depth and color sensor with the principle of Joint Bilateral Filtering (JBF) (Kopf et al., 2007). Such algorithms take as an input a low resolution and noisy depth image together with a high resolution color image as e.g. Kinect produces it. They process the two images to a high resolution noise-reduced depth image as shown in Figure 1. Thereby we especially focus on two drawbacks that are common in JBF based algorithms namely texture copying and no depth discontinuity preservation. Texture copying (see Figure 5(k)) is an artifact that occurs when textures from the color image are transferred into geometric patterns that appear in the upsampled depth image. Our proposed filter algorithm is a new combination of a traditional bilateral filter with a JBF to avoid texture copying. The new Depth Discontinuity Preservation (DDP) post processing step prohibits wrong depth estimates close to edges (see Figure 4), which occur in many bilateral filters.

The remainder of this paper is organized as follows: Section 2 gives an overview of existing methods for depth image upsampling. The proposed CBF algorithm and the DDP post processing are motivated and explained in detail in Section 3, while they are evaluated regarding quality and runtime in Section 4. The work is concluded in Section 5.



(a) (b) (c)

Figure 1: Proposed filter. (a) Color input image. (b) Depth input image. (c) Final result: significantly reduced noise, no aliasing effects, no texture copying, very sharply preserved edges and a higher resolution. Baby3 dataset taken from (Scharstein and Szeliski, 2003). Red pixels have no depth value.

2 RELATED WORK

Depth image upsampling is a well-known problem in the literature and its currently established solutions can be subdivided into three main categories. A rough classification is depicted in Figure 2.

In the first main category the upsampling is based on one single depth image. Besides simple upsampling, which is not considered here, this includes methods that apply machine learning approaches like for example in (Aodha et al., 2012). Additional information is here provided via training patches with pairs of low and high resolution depth images that show different scenes. After training the patches an upsampled image can be computed based on the training data. However, the training dataset needs to be representative, whereas runtime grows with the database size. Thus, a patch based approach is not applicable for real time upsampling.

In the second main category the upsampling is based on multiple depth images, which were captured either from a moving camera or multiple displaced cameras. This requires to register the input images and to find a suitable way to combine them to an upsampled image. These methods differ in the ability to cope with displacement. Thus, a classification into static methods with small displacement (e.g. (Schuon et al., 2009)) and dynamic methods with large displacement (e.g. (Cui et al., 2010)) is reasonable. However, none of these methods is able to process depth images in real time and thus they are not suitable for our purpose.

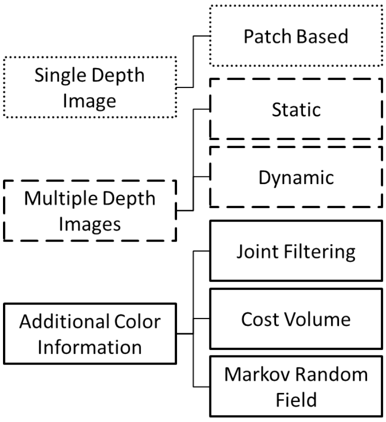


Figure 2: A rough classification of current methods for depth image upsampling.

In the third main category the upsampling is based on one depth image and one simultaneously captured color image. The methods are based on the assumption of coherent depth and color discontinuity, meaning that edges in the color image coincide with edges in the depth image and vice versa.

One approach for using color information in depth images is a Markov Random Field (MRF) as proposed in (Diebel and Thrun, 2005). Therefore, they define the posterior probability of the upsampled depth image as a MRF and optimize for the maximum a posteriori solution. However, this approach suffers from texture copying and high calculation time.

Another approach to use color information in depth images is a cost volume approach as proposed in (Yang et al., 2007). This is based on minimization of costs in a volume with different depth hypotheses. After initial construction of the volume, a filter is applied on each depth hypothesis and for each pixel the depth value with lowest cost is chosen. The filter is one of the crucial points in this method and often realized by an edge preserving filter. This principle suffers also from texture copying, but preserves sharp edges very well. However, it has a high complexity, since the filter must be applied to each single depth hypothesis. Thus, it has a high calculation time and is not suitable for our purpose.

The last common approach to use color information in depth images is Joint Bilateral Filtering (JBF). The basic idea comes from 2D image filtering, where the bilateral filter (BF) for smoothing images and at the same time preserving edges was introduced in (Tomasi and Manduchi, 1998). The key idea of this filter is that the influence of a pixel in the calculation of another pixel does not only depend on spatial neighborhood but also on similar color values. The BF is computed for a pixel p from the input image I in a neighborhood N around p by

$$BF_{Tomasi}(I)_p = \frac{1}{\sum_{q \in N} W_{p,q}} \cdot \sum_{q \in N} W_{p,q} \cdot I_q \qquad (1)$$
$$\text{with} \quad W_{p,q} = G_{\sigma_s}(p-q)\, G_{\sigma_r}(I_p - I_q),$$

where $W_{p,q}$ is a weighting factor, G_{σ_s} is the spatial filter kernel and G_{σ_r} is the range filter kernel. The spatial filter kernel G_{σ_s} is a Gaussian with variance σ_s centered at the pixel p and decreases the influence of distant pixels q. On the other hand the range filter kernel G_{σ_r} is a Gaussian with variance σ_r that decreases the influence of pixels q with distant color I_q to I_p.

This idea was extended in (Kopf et al., 2007) to the JBF, where a depth image D is filtered with the guidance of a color image I. A pixel in the depth image is then influenced by neighboring pixels depending on their distance and the difference of the corresponding color values. The JBF can be computed for a pixel p from the depth image D and the color image I by

$$JBF_{Kopf}(D,I)_p = \frac{1}{\sum_{q \in N} W_{p,q}} \cdot \sum_{q \in N} W_{p,q} \cdot D_q \qquad (2)$$
$$\text{with} \quad W_{p,q} = G_{\sigma_s}(p-q)\, G_{\sigma_r}(I_p - I_q).$$

The main assumption here is again coherent depth and color discontinuity leading to texture copying in regions, which are both flat and textured. This problem was addressed in some related work like e.g. in (Kim et al., 2011) by an extension of the JBF. They add a blending function α to the JBF in order to control the influence of the spatial kernel G_{σ_s} and the range kernel G_{σ_r}. The idea is to differentiate two cases and filter pixels of flat areas with the spatial kernel, in order to avoid texture copying, while pixels of edge areas are filtered with the range kernel. The filter can then be computed for a pixel p from the depth image D and the color image I by

$$JBF_{Kim}(D,I)_p = \frac{1}{\sum_{q \in N} W'_{p,q}} \cdot W'_{p,q} \qquad (3)$$
$$\text{with} \quad W'_{p,q} = \sum_{q \in N} [(1 - \alpha(\Delta_N)) G_{\sigma_s}(p-q) +$$
$$\alpha(\Delta_N) G_{\sigma_r}(I_p - I_q)] D_q$$
$$\text{and} \quad \alpha(\Delta_N) = \frac{1}{1 + e^{-\varepsilon(\Delta_N - \tau)}},$$

where $W'_{p,q}$ is a weighting factor. In the considered neighborhood N the difference between the maximum and minimum depth value is Δ_N, whereas ε and τ are constants. The min-max difference τ determines at which value the blending interval shall be centered, whereas ε controls the width of the transition area between the two cases. This method avoids texture copying quite well but performs poorly in edge regions (Figure 6).

3 METHOD

Our proposed filter is able to iteratively upsample a low resolution and noisy depth image in real time. Therefore, we use a new so-called Combined Bilateral Filter (CBF, see Section 3.1) composed of a Joint Bilateral Filter (JBF) and traditional Bilateral Filter (BF). With this combination the filter avoids texture copying. The new Depth Discontinuity Preservation (DDP) post processing step in Section 3.2 prohibits wrong depth estimates close to edges, which occur in many bilateral filters. The whole iterative filtering process is explained in Section 3.3. An overview of our filter is given in Figure 3, while its details are outlined below.

3.1 Combined Bilateral Filter (CBF)

The proposed CBF is driven by the observation that the JBF delivers good results for edge areas in depth images, but performs poorly in flat depth image regions by introducing artefacts like texture copying. These artefacts are mainly caused by the influence of the color information. Thus, we filter the depth image in flat regions without any guidance of color information by a traditional BF (see Equation 1). Note, in contrast to the filter of (Kim et al., 2011) in Equation 3 we still filter by an edge preserving filter in flat areas. This enables us to preserve also fine depth discontinuities in (almost) flat regions.

The crucial point is to determine the type of region (flat or edge) in a depth image for a given point. A straight forward approach with an edge detection algorithm like e.g. (Canny, 1986) as preprocessing is not suitable here, because our filter should run in real time. Thus, we need to determine the type of region while filtering.

In our approach two filters - the BF and the JBF - are applied in parallel for a given pixel p. The type of region is then estimated based on the difference Δ_p of these two filter results with

$$\Delta_p = |JBF(D,I)_p - BF(D)_p|. \qquad (4)$$

The difference Δ_p is low in flat regions and high in edge regions. Thus, the CBF can be computed for a pixel p from the depth image D and the color image I

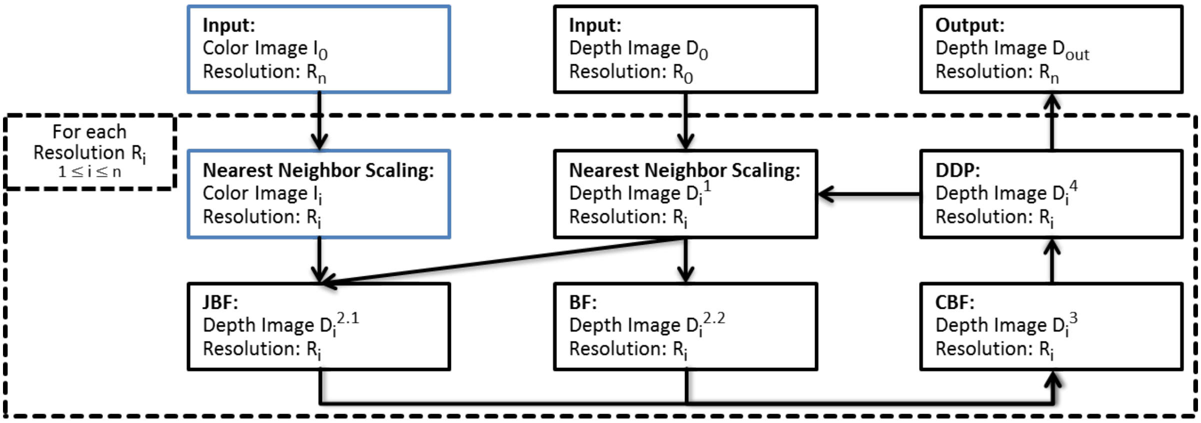


Figure 3: Schematic overview of proposed depth image upsampling filter.

by

$$CBF(D,I)_p = \begin{cases} JBF(D,I)_p, & \Delta_p > s \\ \cos^2\left(\frac{\pi}{2s}\Delta_p\right)BF(D)_p + & , \\ \sin^2\left(\frac{\pi}{2s}\Delta_p\right)JBF(D,I)_p, & \Delta_p \leq s \end{cases} \tag{5}$$

where s is a manually chosen threshold to determine above which value of Δ_p only the JBF is used anymore. With the decision to use sine and cosine respectively as blending function a normalization is not necessary, since $\forall x : \sin^2 x + \cos^2 x = 1$.

3.2 Depth Discontinuity Preservation (DDP)

The CBF is an edge preserving filter, since it is composed of the two edge preserving filters BF and JBF. However, after filtering fine blur still remains around edges in the depth image just as in the BF and JBF. For 2D images, where the bilateral filtering concept comes from, this has not a high impact, since fine blur around edges still looks appealing from a visual point of view. Certainly, fine blur around edges in depth images has a high impact as shown in Figure 4. If a depth sensor captures two distant but overlapping planes, it captures (nearly) no pixels between these two planes. However, by filtering with a JBF or a BF a transition consisting of pixels between the two planes is created. These pixels are of course substantially wrong, since in reality there is no connection between the planes and also the depth sensor did not capture any connection. This error occurs in many existing bilateral filter based depth image filtering algorithms.

Thus, in this paper we propose a new post processing filter called Depth Discontinuity Preservation (DDP) to avoid depth transitions between different depth levels. The goal is to adjust wrong filter results by shifting pixels to the correct depth level. The idea of DDP is that the edge preserving filter should only output a pixel p, whose depth value was already existing before filtering in a small neighborhood N around p. This idea is similar to a median filter, but in contrast DDP retains the edge preserving property of the input filter like e.g. the CBF. More formally, DDP filters a pixel p for a raw depth image D and a CBF result (cf. Equation 5) by

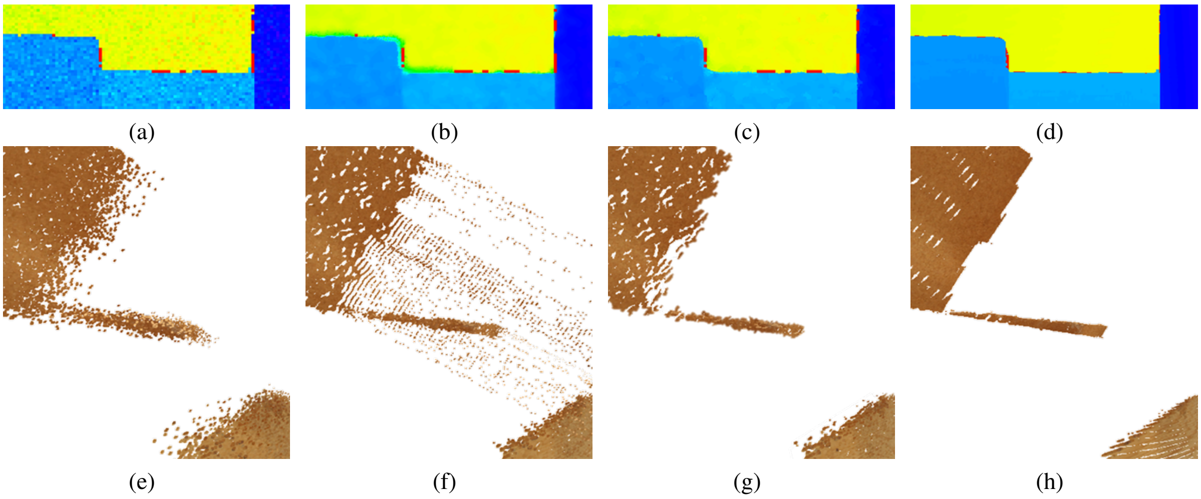
$$DDP(D,CBF)_p = \Big\{ CBF_j \Big| j \in N; \forall k \in N : \\ \|CBF_j - D_p\| \leq \|CBF_k - D_p\| \Big\}, \tag{6}$$

where N is a small neighborhood around p. The effect of DDP is discussed in Section 4 (see Figure 4).

3.3 Filtering Process

Our filter (see Figure 3) takes as an input a color Image I_0 with a high resolution R_n and a depth image D_0 with a low resolution R_0. The goal is to increase the resolution of D_0 from R_0 to R_n. Therefore, we use an iterative procedure with n iterations. Unlike other methods we do not directly upsample D_0 from R_0 to R_n and filter afterwards several times on the highest resolution, since filtering on high resolutions is time consuming. Our method converges to resolution R_n in each iteration in a linear way.

In the i-th iteration both the color image I_0 and the depth image D_{i-1} are scaled to resolution R_i resulting in I_i and D_i^1 respectively. The BF (Equation 1) and the JBF (Equation 2) are then applied to I_i and D_i^1 required for our CBF (Equation 5). The output of the CBF is the depth image D_i^3, which is refined by our DDP (Equation 6) filter to D_i^4. If the final resolution is reached ($i = n$) the depth image D_i^4 is the final result D_{out}, otherwise the next iteration starts. The effects of each step are illustrated in Figure 5 and are reviewed in Section 4.


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

(e) (f) (g) (h)

Figure 4: Depth Discontinuity Preservation (DDP). (a),(e): input depth image. (b),(f): joint bilateral filter (JBF) or combined bilateral filter (CBF). (c),(g): JBF or CBF with DDP postprocessing. (d),(h): ground truth. Top row: 2D depth image. Bottom row: depth image in 3D view. Wood2 dataset taken from (Scharstein and Szeliski, 2003).

4 EVALUATION

In this section we evaluate our CBF algorithm. Therefore, we first explain our evaluation setup in Section 4.1. In a second step we evaluate our CBF algorithm from a visual point of view in Section 4.2. At the end we compare in Section 4.3 our CBF algorithm against ground truth data and related algorithms.

4.1 Evaluation Setup

The evaluation of depth images is not trivial, since it is hard to produce ground truth data. Thus, we used - like many other authors - the disparity images of the stereo datasets of Middlebury (Scharstein and Szeliski, 2003) in the evaluation of our algorithm. A disparity image can also be seen as a depth image, because depth and disparity are directly related. Hence, the disparity images with a resolution of 1282×1110 are used as ground truth data.

For the simulation of depth sensor data we downsample the resolution by factor 4, since real depth data has a resolution in this range. To simulate the depth sensors noise we add a Gaussian noise with a standard deviation of 4 to the downsampled image. The downsampled and noisy depth image serves then as an input for both our developed and competitive depth image upsampling algorithms. In the ground truth comparisons of this section we compare the output of a given filter against the raw depth image. To demonstrate the effects of depth image upsampling algorithms we distinguish between flat and edge areas. The edges are detected by a Canny edge detector (Canny, 1986) and then dilated to cover the whole edge area. The flat areas are the complement of the edge areas. For the quantitative evaluation we choose two measures: For a depth image D and ground truth image G with n pixels the Mean Error (ME) can be estimated by

$$ME(D,G) = \frac{1}{n}\sum_{i=1}^{n} \|D_i - G_i\| \qquad (7)$$

and the Error Rate (ER) by

$$ER(D,G) = \frac{1}{n}\sum_{i=1}^{m} p_i, \;\; \forall p_m : \|D_{p_m} - G_{p_m}\| > 2. \qquad (8)$$

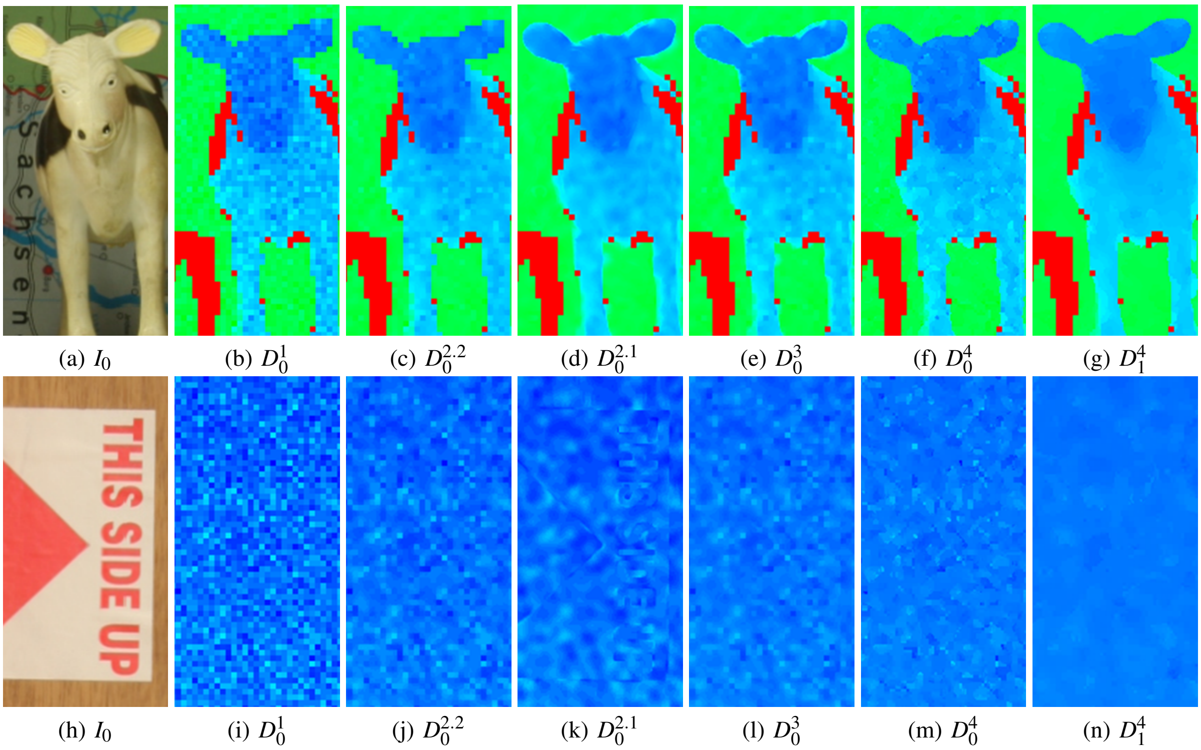
The ER gives the percentage of pixels, which have an error greater than 2. The quality of both our CBF algorithm and competitive algorithms depends on the choice of the corresponding parameters. We chose the parameters of Table 1 empirically in several tests to achieve the best results for each algorithm.

Table 1: Parameters of evaluated algorithms.

	$\|N\|$	σ_s	σ_r	s	ε	τ
CBF + DDP	49	3	2	18	-	-
JBF_{Kopf}	49	3	2	-	-	-
JBF_{Kim}	49	3	0.5	-	15	0.5
Cost Volume	49	3	3	-	-	-

4.2 Qualitative Evaluation

In this section we qualitatively evaluate the effects of our new filter. Therefore, Figure 5 shows the resulting depth image after each single filtering step. The top row (Figure 5 (a)-(g)) demonstrates the effects of edge



(a) I_0 (b) D_0^1 (c) $D_0^{2.2}$ (d) $D_0^{2.1}$ (e) D_0^3 (f) D_0^4 (g) D_1^4

(h) I_0 (i) D_0^1 (j) $D_0^{2.2}$ (k) $D_0^{2.1}$ (l) D_0^3 (m) D_0^4 (n) D_1^4

Figure 5: Illustration of each step in our filter for two exemplary zoomed datasets. (a),(h): input color image. (b),(i): input depth image. (c),(j): bilateral filter. (d),(k): joint bilateral filter. (e),(l): combined bilateral filter (CBF). (f),(m): depth discontinuity preservation (DDP). (g),(n): final result after two iterations. Top row: Baby3 dataset. Bottom row: Wood2 dataset. Each taken from (Scharstein and Szeliski, 2003). Red pixels have no depth value.

preserving filtering while removing noise, whereas the bottom row (Figure 5 (h)-(n)) illustrates the filtering of a flat and textured region. The first column shows the high resolution color input image I_i and the second column depicts the low resolution depth input image D_i^1 with lots of noise.

After filtering with a BF the noise in $D_i^{2.2}$ is reduced but still clearly visible. The edges around the ears show aliasing effects, but also no texture was copied into the depth image. In contrast, the JBF reduces much more noise in $D_i^{2.1}$ and also aliasing at the edges of the ear is not visible anymore. However, texture was copied into the depth image as visible in Figure 5 (k). Here, the 3D structure contains the borders of the characters in Figure 5 (h), the triangle and the rectangle. Furthermore, around the edges fine blue is visible.

Our CBF combines the results of BF and JBF as well as their particular advantages. The resulting depth image D_i^3 has a lower level of noise, shows no aliasing effects and contains no texture in its 3D structure. There is only one drawback left: around the depth discontinuities, e.g. at the ears, fine blur is still visible.

Our DDP post processing in D_i^4 removes the fine blur in edge regions and still aliasing effects are avoided. This is even better visible in 3D in Figure 4. Without DDP a non-existing transition between the two distant depth levels is introduced, whereas with DDP this connection is completely corrected. Note, the wrong pixels in the transition area are not removed but corrected in their position. However, the DDP slightly reduces the effect of smooth filtering in flat regions (Figure 5 (m)). This is due to the fact that DDP outputs only values, which are present in a neighborhood and these values must not be optimal for smoothing, especially in flat regions.

In our evaluation setup the whole filter runs two iterations - each doubling the resolution - to achieve the final resolution. The output D_{out} is a high resolution depth image with essentially reduced noise, no aliasing effects, no texture copying and very sharply preserved edges.

4.3 Quantitative Evaluation

After qualitatively evaluating our results in the previous section, we quantify the quality of our algorithm in this section. Therefore, we conduct a ground truth comparison as specified in Section 4.1 and compare the results with competitive algorithms.

Figure 6 shows the Mean Error (ME) for three kinds of image regions of all 42 datasets of (Scharstein and Szeliski, 2003). In the whole image both JBF_{Kopf} and JBF_{Kim} can halve the ME compared to the input, whereas the Cost Volume approach reduces the ME by around 60%. Our algorithm on the other hand is able to reduce the ME by up to 73%. In edge regions our algorithm ($ME \approx 2.5$) performs slightly worse than the Cost Volume ($ME \approx 2.1$), but still better than JBF_{Kopf} ($ME \approx 3.2$) and JBF_{Kim} ($ME \approx 3.9$). However, the Cost Volume has a high complexity, meaning that our algorithm is the best performing of the real-time capable algorithms. In flat regions, which contain most pixels, our algorithm is superior compared to all other evaluated algorithms. In Table 2 the Error Rate (ER) in % is used as quality measure and the results are similar to the ME.

To enable the real-time performance of our CBF filter including the DDP post processing we ported our code to the GPU. Since the GPU offers a fast parallel computing and our algorithms are calculated for each pixel independently, we achieve without excessive GPU optimization an performance gain of up to 2000× compared to the CPU. In our evaluation setup (see Section 4.1) we are able to upsample a depth image with our CBF including DDP within around 30 milliseconds on a NVIDIA GeForce GTX 660 Ti. The parallel computation of both the BF and the JBF is very effective and takes in our implementation only around 10% more calculation time than one single filter. As long as the GPU provides enough parallel processing units, the processing time does not increase for higher resolutions. However, in general the processing time is linear to the resolution and quadratic to the radius of the neighborhood N around a processed pixel.

5 CONCLUSION

The proposed Combined Bilateral Filter (CBF) together with the new Depth Discontinuity Preservation (DDP) post processing is able to upsample a noisy depth image in real time with the guidance of a color image. Our CBF filter explicitly avoids texture copying and the DDP preserves edges very sharply. The output of our algorithm is a high resolution depth image with essentially reduced noise and no aliasing effects. Compared to existing algorithms such as JBF_{Kopf} (Kopf et al., 2007), JBF_{Kim} (Kim et al., 2011) and Cost Volume (Yang et al., 2007) we overall achieve superior results with our algorithm. Our algorithm is able to reduce the mean error within around 30ms up to 73% in a ground truth comparison. Furthermore, our algorithm can be used as a standalone pre processing for existing algorithms whenever depth images are needed as an input.

ACKNOWLEDGEMENTS

This work was partially funded by the Federal Ministry of Education and Reseach (Germany) in the context of the projects ARVIDA (01IM13001J) and DENSITY (01IW12001). Furthermore, we want to thank Vladislav Golyanik for his advise in GPU programming.

REFERENCES

Aodha, O. M., Campbell, N. D., Nair, A., and Brostow, G. J. (2012). Patch based synthesis for single depth image super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 71–84. Springer.

Canny, J. (1986). A computational approach to edge detection. In *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, volume 6, pages 679–698. IEEE.

Cui, Y., Schuon, S., Chan, D., Thrun, S., and Theobalt, C. (2010). 3D shape scanning with a time-of-flight camera. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1173–1180. IEEE.

Diebel, J. and Thrun, S. (2005). An application of markov random fields to range sensing. In *Advances in neural information processing systems (NIPS)*, pages 291–298. NIPS.

Kim, C., Yu, H., and Yang, G. (2011). Depth super resolution using bilateral filter. In *International Congress on Image and Signal Processing (CISP)*, volume 2, pages 1067–1071. IEEE.

Kopf, J., Cohen, M. F., Lischinski, D., and Uyttendaele, M. (2007). Joint bilateral upsampling. In *ACM Transactions on Graphics (TOG)*, volume 26, page 96. ACM.

Scharstein, D. and Szeliski, R. (2003). High-accuracy stereo depth maps using structured light. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 1–195. IEEE.

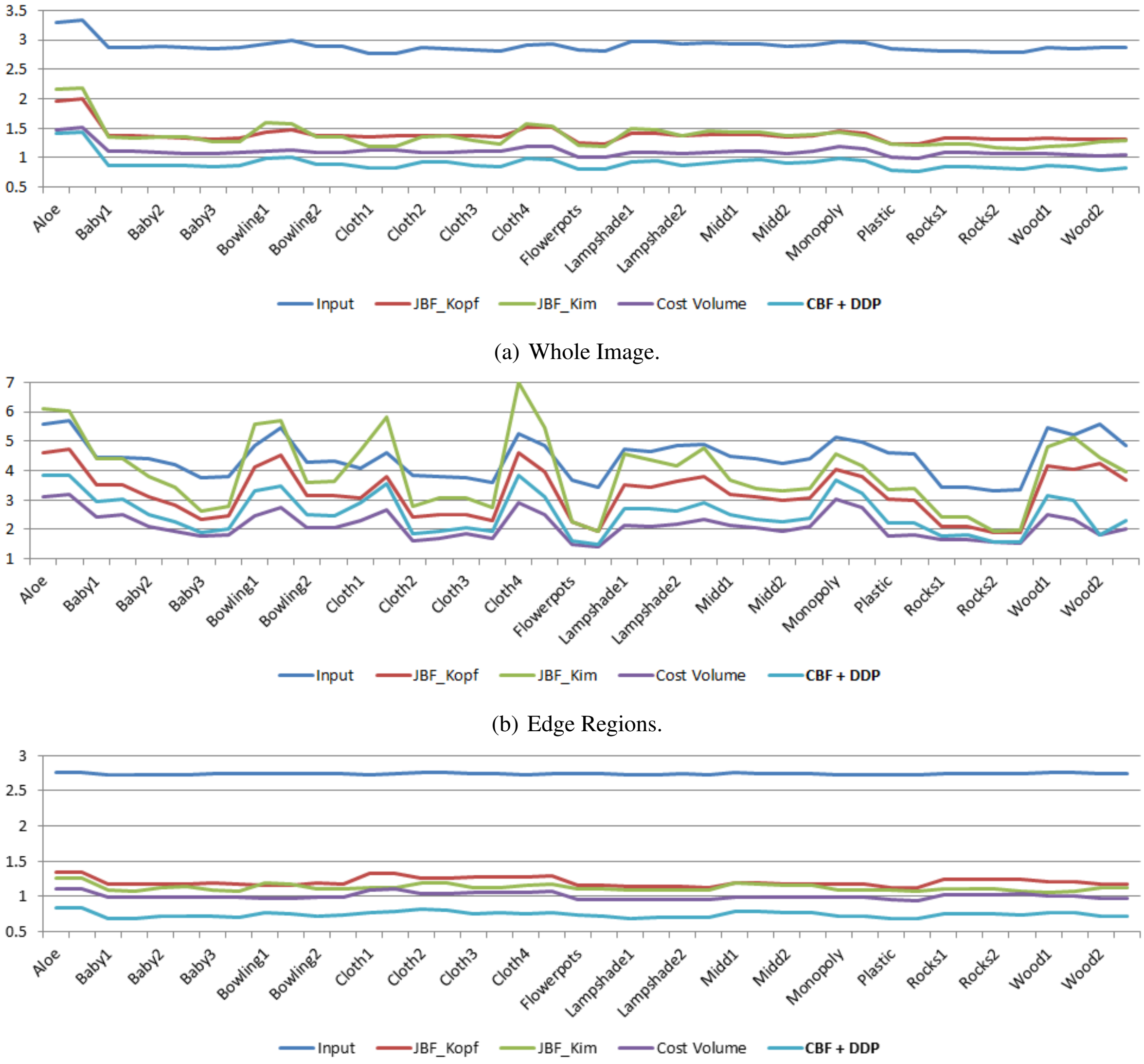
Schuon, S., Theobalt, C., Davis, J., and Thrun, S. (2009). LidarBoost: Depth superresolution for ToF 3D shape scanning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 343–350. IEEE.

Tomasi, C. and Manduchi, R. (1998). Bilateral filtering for gray and color images. In *International Conference on Computer Vision (ICCV)*, pages 839–846. IEEE.

Yang, Q., Yang, R., Davis, J., and Nistér, D. (2007). Spatial-depth super resolution for range images. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE.

Table 2: Error Rate (see Equation 8) in % for a representative selection of datasets of (Scharstein and Szeliski, 2003) in denoted regions.

Dataset	Input			JBF_{Kopf}			JBF_{Kim}			Cost Volume			CBF + DDP		
	all	edge	flat	all	edge	flat	all	edge	flat	all	edge	flat	all	edge	flat
Aloe	46.4	49.7	45.7	18.2	35.1	14.3	14.2	44.4	7.2	9.2	14.8	7.9	9.3	27.4	5.2
Baby3	45.9	47.6	45.7	11.8	24.6	10.1	5.9	25.4	3.4	6.2	13.6	5.2	5.0	17.2	3.5
Bowling2	45.7	48.4	45.4	11.7	26.6	10.2	5.7	26.2	3.6	6.0	12.5	5.3	4.9	19.8	3.3
Cloth4	45.7	49.4	45.4	14.3	33.4	12.7	7.0	44.9	4.1	7.6	14.7	7.0	5.4	30.7	3.4
Flowerpots	45.6	46.9	45.5	10.3	21.4	9.2	4.5	18.6	3.1	5.2	10.7	4.6	3.8	14.0	2.8
Midd2	45.9	48.2	45.6	11.5	25.8	10.0	6.9	26.2	4.9	5.8	10.5	5.4	5.8	18.2	4.5
Rocks1	45.9	48.9	45.5	12.9	24.3	11.7	5.8	25.3	3.7	6.9	14.7	6.1	4.7	17.3	3.3
Wood2	45.6	46.7	45.6	10.5	24.8	9.9	3.8	24.4	2.8	5.2	8.8	5.0	3.2	14.9	2.6



(a) Whole Image.

(b) Edge Regions.

(c) Flat Regions.

Figure 6: Mean Error (see Equation 7) for all datasets of (Scharstein and Szeliski, 2003) in denoted regions.