Color Segmentation Based Depth Image Filtering

Michael Schmeing and Xiaoyi Jiang

Department of Computer Science, University of Münster
Einsteinstraße 62, 48149 Münster, Germany,
{m.schmeing|xjiang}@uni-muenster.de

Abstract. We present a novel enhancement method that addresses the problem of corrupted edge information in depth maps. Corrupted depth information manifests itself in zigzag edges instead of straight ones. We extract the depth information from an associated color stream and use this information to enhance the original depth map. Besides the visual results, a quantitative analysis is conducted to prove the capabilities of our approach.

1 Introduction

Video-plus-depth is an important 3D scene representation format [1]. It consists of a color stream describing the texture of the scene and an associated depth stream describing for each pixel its distance to the camera. From this representation, arbitrary new views can be generated to enable stereo [1, 2], multi view [3] or free viewpoint video [4].

An important presumption for high quality rendering is a high quality depth map. However, there exists at the moment no depth map generation technique that is able to produce a perfect depth map, i.e., a depth map that is free of artifacts, holes, which is temporally stable and has video resolution all together.

Different depth map enhancement methods have evolved to address different aspects of depth map corruption [5–9] . In this paper, we propose a novel enhancement algorithm that takes associated color information into account to enhance the quality of edges in a depth map. We use depth maps generated by the Microsoft Kinect depth camera for our approach, though our algorithm is not restricted to depth maps generated with this camera. The Kinect camera is a structured light depth sensor which suffers from quite poor edge reproduction. Figure 1 shows an example. We use edge information found in the corresponding color stream via a superpixel segmentation and compute a new representative depth map D^r which stores robust edge information corresponding to the color

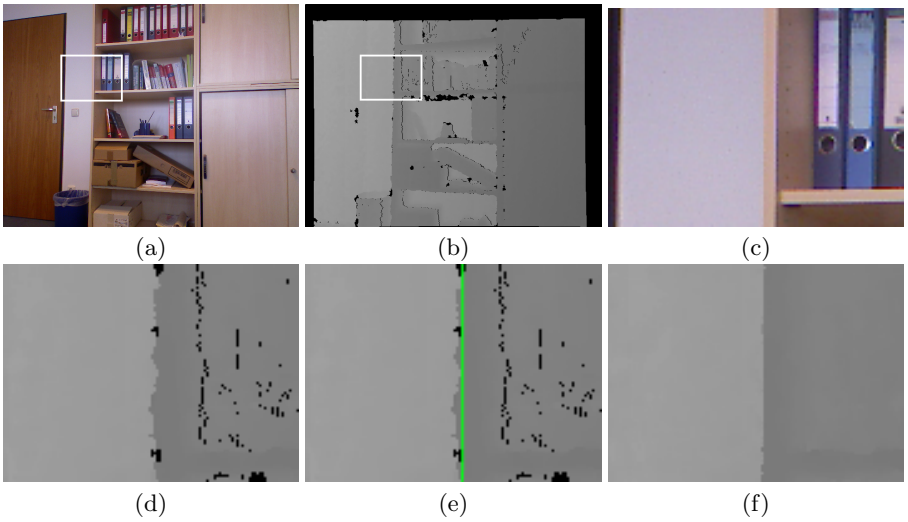

Fig. 1. (a) Example color image. (b) Associated depth map. (c) Magnification of (a). (d) Magnification of (b). (e) The green line marks the edge from the color stream. (f) Result of our approach.

stream. D^r is then used to enhance the source depth map D. A quantitative analysis shows that our method outperforms common depth enhancement algorithms in terms of edge restoration.

The rest of our paper is organized as follows. Section 2 discusses previous work in this field. We describe our algorithm in Section 3 and present some results in Section 4. Section 5 concludes this paper.

2 Related Work

There exist several methods that enhance depth maps. In [5], depth maps are filtered using a temporal median filter. Holes are filled using a spatial median. The work reported in [6] addresses special issues of the Microsoft Kinect depth camera. Holes that occur due to the offset between color and depth camera are closed by background extraction and other holes by row-wise linear interpolation. No temporal processing is applied.

In [7], the authors port simple filters like Gaussian-weighted hole filling and temporal smoothing as well as edge-preserving denoising on the GPU and achieve very high framerates of 100fps. The framework is modal and not dependent on a certain depth camera technology. The method looks promising and is said to be applicable to dynamic scenes although no special evaluation was given in this case.

3 Our Method

In our algorithm we address the common problem of corrupted edge information in depth images. Figure 1 shows an example. In the color image, the edge of the foreground object (the shelf) is a straight line, whereas it is corrupted in the depth stream. The corrupted edges in the depth image do not correspond with the edges of actual objects. When using this depth map for view synthesis (e.g. Depth Image Based Rendering [1]), artifacts will occur. Therefore, it is important to have edges in the depth map that correspond closely to the edges of the objects in the scene.

Our method works for scenes in the *video-plus-depth* format. We assume the depth stream to have the same resolution as the video stream. Depth upsampling [10], which is another field of depth video enhancement, can be applied as a preprocessing step if necessary.

There are two kinds of possible edge defects in a depth map. First, the edge is not straight but rather forms a zigzag line or consists of other visible steps. This can happen through the nature of the sensor (like the Kinect sensor) or through inadequate depth upsampling. The second defect is global misalignment, i.e. the complete edge of the depth map is shifted with respect to the edge in the color image. This defect can arise from insufficient registration between video camera and depth camera.

Let I be a frame of the video sequence and D the corresponding depth map. Our goal is to process D in a way that the edges in D align with the edges (of objects) in I.

As a first step, we perform *normalized convolution* [9] to fill holes in the depth map. A hole pixel x is filled with a weighted sum of the depth values of its non-hole neighboring pixels:

$$D^{nc}(x) = \frac{\sum_{x' \in N_x^*} D(x) g(x, x')}{\sum_{x' \in N_x^*} g(x, x')} \tag{1}$$

where N_x^* is the set of neighboring pixels of x that have a valid depth value and $g(x, x')$ a Gaussian function with parameter σ:

$$g(x, x') = \exp\left(-\frac{||x - x'||^2}{\sigma^2}\right). \tag{2}$$

In the next step, we identify edges in the color image. Instead of finding edges directly with common methods like the Canny operator, we use the implicit edge information given by a segmentation, more precisely, an over-segmentation of the color image. While a normal segmentation divides the image into "meaningful" areas (usually guided by edges), an over-segmentation further divides these areas. Those areas can nevertheless be recovered by combining areas of the over-segmentation. Particularly, the over-segmentation respects the edges of objects in the color image.

With an over-segmentation of the depth map, we can compute representative depth values for each segment, for example by taking the median or the average

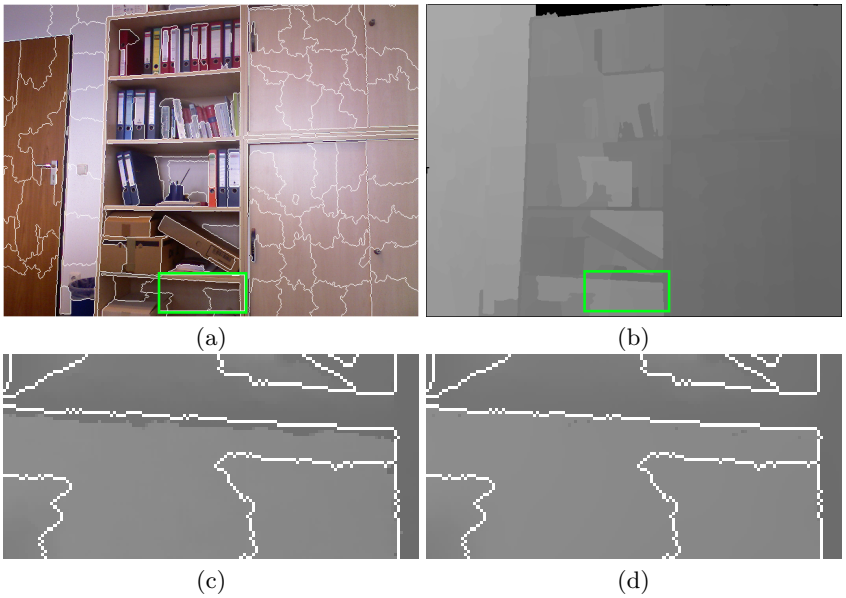

Fig. 2. (a) An example over-segmentation. (b) A representative depth map D^r with marked magnification region. (c) Cutout of original depth with projected segmentation. (d) Cutout of our method with projected segmentation: Depth and color edges align.

of all the depth values of pixels in this segment. These representative values are more robust to noise than one single pixel. The representative depth map D^r will be generated by filling each segment with its representative depth value. Since the segmentation respects the color image edges, the edges in the representative depth map will respect these edges, too. Using D^r, we can later discard pixels as corrupted that are too dissimilar to the representative depth value of their segment. See Figure 2(c) for an example: In the upper part of the main light region, the dark depth region overlaps into the light region which means that the depth edge and the color edge (i.e., the segment border) do not correspond. Figure 2(d) shows the corrected edge produced by our algorithm.

We tried different superpixel-segmentation methods including Mean-Shift and the method of [11] but in the end we used a simple watershed segmentation [12] because it delivers sufficient results for our purpose at a very high speed (more than 30fps at 640×480 resolution. We also tried different marker distributions for the watershed segmentation: randomly, on a regular grid, and skewed on a regular grid (which means that the markers of two consecutive rows do not lie in the same column but are slightly shifted). We tested these distributions with the additional constraint that markers are not placed on edges in the color image. The best results where obtained with the regular, skewed grid and no additional constraint.

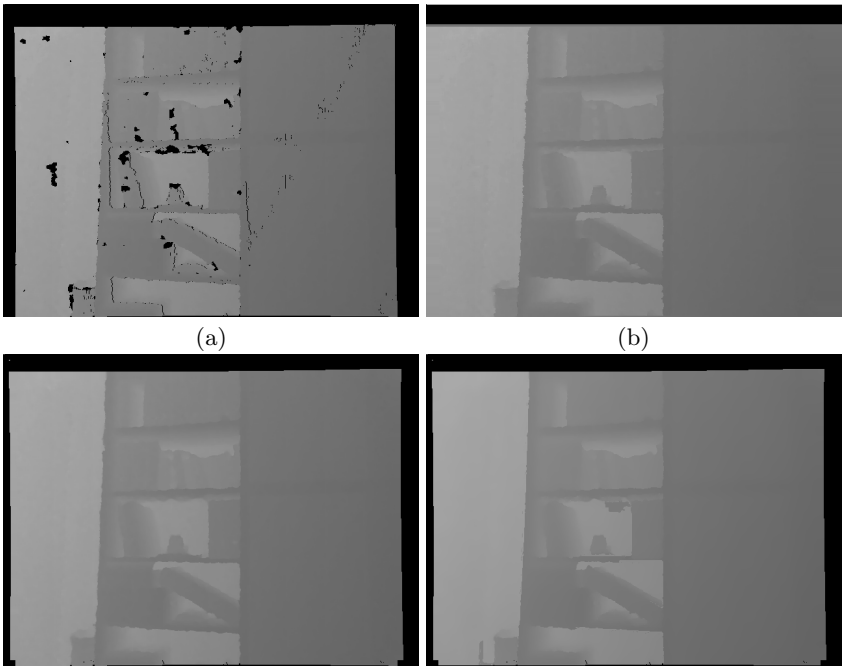

Fig. 3. (a) Example frame of the unfiltered depth generated by the Kinect depth sensor. (b) Frame filtered with method of Brednikov et al. [6]. (c) Frame filtered with method of Wasza et al. [7]. (d) Our method.

Figure 2(a) shows an example segmentation. The quality of the segmentation can be degraded by a high amount of image noise, so we first apply a bilateral filter to reduce the noise while simultaneously protect edges in the color image. The filtered color value $I(p)$ of a pixel p is given by:

$$I(p) = \frac{\sum_{q \in N} K_s(||p-q||) K_c(||p-q||) I(q)}{\sum_{q \in N} K_s(||p-q||) K_c(||p-q||)} \tag{3}$$

with K_s and K_c being Kernel functions, typically Gaussian distributions.

The obtained over-segmentation is then projected into the depth stream. In an ideal depth map, the edges of the over-segmentation would coincide with the edges in the depth map. Figure 2(c) shows what happens in real world depth maps (taken from a Kinect): Some areas overlap into neighboring segments.

Using a sufficient segment size, though, we can ensure that at least half of the depth pixels in a segment have correct depth (this is clearly the case in Figure 2(c)). We build the representative depth map D^r from this segmentation by computing for each segment the median depth value:

$$D^r(x,y) = \{d_k : (x,y) \in S_k,\ d_k = \underset{(x',y') \in S_k}{\text{median}}\ d(x',y')\}$$

Fig. 4. Sample color and depth frame from the *edge test* sequence.

where S_k is a segment in the color image. Figure 2(b) shows an example representative depth map. This depth map corresponds very well with the edges in the color image but suffers of course from the fact that it cannot represent smooth depth transitions but rather consists of discrete patches.

The final filtered depth map D^f uses the depth values of D^r only, if D exhibits corrupted depth values. D^f is obtained in the following way:

$$D^f(x,y) = \begin{cases} D^r(x,y) & \text{if } |D(x,y) - D^r(x,y)| > \theta \\ D(x,y) & \text{otherwise} \end{cases} \tag{4}$$

with θ being a threshold.

4 Experimental Results

4.1 Qualitative Results

Figure 3 shows some results of our method compared with other methods. Berdnikov et al.[6] address special issues of the Microsoft Kinect depth camera. Holes that occur due to the offset between color and depth camera are closed by background extraction and other holes by row-wise linear interpolation. The focus of Wasza et al.[7] lies on porting simple filters like Gaussian-weighted hole filling [9] and temporal smoothing as well as edge-preserving denoising on the GPU to achieve very high frame rates.

Our method closes all holes and in contrast to other method, it can restore straight edges. This behavior can also be seen in Figure 2(c) and (d).

4.2 Quantitative Results

It is difficult to obtain quantitative quality results for depth filtering algorithms. This is due to the fact that usually no ground truth depth map is available to compare the filtered depth map with. However, we designed a test method to assess the ability of our algorithm to restore edges.

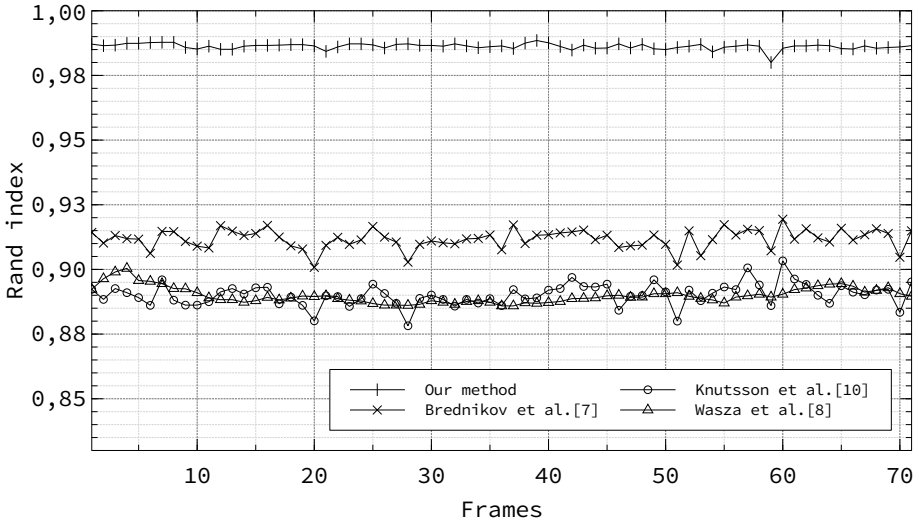

Fig. 5. Rand index values for different depth filtering algorithms. A value near to 1 means a very good correlation with the color stream.

Recall Figure 1(e) for an example of a corrupted edge in the original depth map. We see that the edge does not correspond very well with the edge in the color stream (green line). Our algorithm, see Figure 1(f), performs way better and we want to quantify this result. To do this, we recorded a test sequence *edge test* of 71 frames (color and depth, see Figure 4) with very simple geometry: It depicts a homogeneous (in terms of depth values) foreground object with a straight edge in front of a homogeneous background. The foreground object also has different color texture than the background.

In this situation, we can define two clusterings that divide the scene into foreground and background : C_D is a 2-means clustering of the depth map and C_C is a 2-means clustering of the color stream. If the depth map is aligned with the color stream and does not exhibit cracks or other corruption, then clustering C_D and C_C should be the same.

To determine how similar the clustering C_D and C_C are, we compute the Rand index[13]. The Rand index $\mathbb{R}(\cdot,\cdot) \in [0,\dots 1]$ is a very popular measure to describes how similar two clusterings are. A Rand index of 1 means they are the same whereas 0 means they are completely different. In our situation a high Rand index indicates a very good correlation between the color stream and the (filtered) depth stream. Figure 5 shows the Rand indices for all frames of the test sequence. We can see that our algorithm clearly outperforms all other algorithms.

5 Conclusion

We have presented a method to increase the spatial accuracy of depth maps using edge information of the associated color stream. Our method can reliably enhance corrupted edges in the depth stream and outperforms common algorithms.

Future work aims at the inclusion of inter-frame information to enforce time-consistency and further reduce edge artifacts.

References

1. Fehn, C., de la Barre, R., Pastoor, R.S.: Interactive 3-DTV-Concepts and Key Technologies. Proceedings of the IEEE **94** (2006) 524–538
2. Schmeing, M., Jiang, X.: Depth Image Based Rendering: A Faithful Approach for the Disocclusion Problem. In: Proc. 3DTV-Conf.: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON). (2010) 1–4
3. Zitnick, C.L., Kang, S.B., Uyttendaele, M., Winder, S., Szeliski, R.: High-quality Video View Interpolation using a Layered Representation. ACM Transactions on Graphics **23**(3) (2004) 600–608
4. Smolic, A., Mueller, K., Merkle, P., Fehn, C., Kauff, P., Eisert, P., Wiegand, T.: 3D Video and Free Viewpoint Video - Technologies, Applications and MPEG Standards. In: 2006 IEEE International Conference on Multimedia and Expo. (2006) 2161 –2164
5. Matyunin, S., Vatolin, D., Berdnikov, Y., Smirnov, M.: Temporal Filtering for Depth Maps generated by Kinect Depth Camera. In: Proc. 3DTV Conf.: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON). (2011) 1–4
6. Berdnikov, Y., Vatolin, D.: Real-time Depth Map Occlusion Filling and Scene Background Restoration for Projected-Pattern-based Depth Camera. In: 21th International Conference on Computer Graphics and Vision (GraphiCon2011). (2011) 1–4
7. Wasza, J., Bauer, S., Hornegger, J.: Real-time Preprocessing for Dense 3-D Range Imaging on the GPU: Defect Interpolation, Bilateral Temporal Averaging and Guided Filtering. In: IEEE International Conference on Computer Vision Workshops (ICCV Workshops). (2011) 1221 –1227
8. Min, D., Lu, J., Do, M.: Depth Video Enhancement Based on Weighted Mode Filtering. IEEE Transactions on Image Processing **21**(3) (2012) 1176 –1190
9. Knutsson, H., Westin, C.F.: Normalized and Differential Convolution. In: 1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1993. (1993) 515 –523
10. Diebel, J., Thrun, S.: An Application of Markov Random Fields to Range Sensing. In: Advances in Neural Information Processing Systems 18. MIT Press, Cambridge, MA (2006) 291–298
11. Liu, M.Y., Tuzel, O., Ramalingam, S., Chellappa, R.: Entropy Rate Superpixel Segmentation. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. (2011) 2097 –2104
12. Beucher, S., Lantuejoul, C.: Use of Watersheds in Contour Detection. In: International Workshop on Image Processing: Real-time Edge and Motion Detection/Estimation, Rennes, France. (1979)
13. Rand, W.M.: Objective Criteria for the Evaluation of Clustering Methods. Journal of the American Statistical Association **66**(336) (1971) pp. 846–850