Fast Temporal Filtering of Depth Maps

Sergey Matyunin
Moscow State University
Graphics & Media Lab
smatyunin@graphics.cs.msu.ru

Dmitriy Vatolin
Moscow State University
Graphics & Media Lab
dmitriy@graphics.cs.msu.ru

Maxim Smirnov
YUVsoft Corp.
ms@yuvsoft.com

ABSTRACT

In this paper, we propose a method of filtering depth maps that are automatically generated from video sequences using optical flow, 3D reconstruction and scene analysis methods. To attain better quality, information from both the source video and depth map is used. The proposed algorithm uses motion estimation to take into account temporal information, but the algorithm's structure permits use of optical flow to improve quality, but at the expense of greater computation time. The method can be applied as a preprocessing stage for enhancement of multi-view or stereo video. Joint temporal and spatial processing can yield further improvements in quality. A comparison of the results with test ground-truth sequences using the BI-PSNR metric is presented.

Keywords

Depth map, temporal filtering, stereo, 3D, video.

1 INTRODUCTION

Depth maps are widely used for 3D video production. Creating a depth map is a laborious process, so methods of automatic generation are under development. One of the promising approaches is depth map reconstruction using object motion [KMS07]. In [SCN05], the authors propose a method of spatial structure analysis based on neural networks and machine learning.

A relatively simple research direction involves calculating depth using stereo reconstruction [OA05]. Despite its simplicity, this approach also encounters many unsolved problems. Estimation of depth on the basis of stereo data aids parallax tuning for different types of screens and showing rooms, and it allows to take the parallax of neighboring scenes into account during nonlinear editing.

The problem of definite depth reconstruction without additional information is generally unsolvable. For automatic depth reconstruction, approaches that are based on local criteria minimization can be applied. This approach, however, leads to errors in the depth map. Such depth maps cannot be used for 3D image creation owing to temporal instability and errors. A specific type of preprocessing is required to increase the temporal and spatial stability of the results. This paper proposes such a method of depth map processing using color and motion information.

2 RELATED WORK

Depth processing is often used to decrease the noticeability of depth map errors during visualization. Modified forms of Gaussian blur are applied in occlusion areas. In [TZ04], the authors propose asymmetric blurring: the filter length is larger in the vertical direction than in the horizontal direction. They also propose changing the size of the symmetric smoothing filter depending on the local values in the depth maps. An edge-dependent depth filter was proposed in [CCL+05]. To increase the quality of the results, edge direction is taken into account. Artifacts are more noticeable in occlusion areas. An adaptive method that responds to occlusions was proposed in [LH09].

The above-mentioned approaches only use data from the current frame, and they only use a portion of the color information from the source video (for example, only edges location).

A method of minimizing depth flickering for stationary objects was proposed in [KCKA10]; this method, however, only considers the presence of motion rather than the magnitude of the motion.

In [ZJWB09], the authors propose a method of reducing temporal instability by solving the energy minimization problem for several consecutive frames using graph cut and belief propagation. Another approach to depth map post-processing proposed in [ZJWB08] is iterative refinement. For each frame,

the algorithm refines the depth maps of neighboring frames. The refinement procedure is also reduced to the energy minimization problem. Such approaches produce good results, but owing to computational complexity, they require a long time to process the entire video.

The proposed approach uses several neighboring frames to refine the depth map. Filtering is performed by taking into account the intensity (color) similarity of pixels and the spatial distance. The algorithm takes information about object motion into account using motion compensation.

3 PROPOSED METHOD

The proposed algorithm uses frames from the source video sequence along with a depth map generated from this video. We denote $I_i(x)$ as the intensity (or color) of pixel x in frame i. This intensity is either a three-vector for a color image or a scalar for a grayscale image. n denotes the current frame number, and $D_i(x)$ represents the depth for the ith frame in position x . The proposed method consists of four steps:

1. Motion estimation (ME) between the current frame I_n and neighboring frames I_{n+d}, where $d = -m,\ldots,-1,1,\ldots,m$, and $m > 0$ is a parameter. The result of this stage is a field of motion vectors $MV_{n+d}(x)$. We define $MV_n(x) \equiv 0$.

2. Computation of the confidence metric $C_{n+d}(x)$ for the resultant motion vectors $MV_{n+d}(x)$. Here, $C_{n+d}(x) \in [0,1]$. $C_i(x)$ quantifies the estimation quality for motion vector $MV_i(x)$.

3. Motion compensation for the depth map and source frames. Here, D^{MC}_{n+d} denotes the motion-compensated depth maps, and I^{MC}_{n+d} denotes the motion-compensated source frames. Both the I^{MC}_i and D^{MC}_i images are computed using motion vectors MV_i, which are estimated from the source video sequence:

$$I^{MC}_i(x) = I_i(x + MV_i(x)),$$

$$D^{MC}_i(x) = D_i(x + MV_i(x)).$$

4. Depth map filtering using the computed D^{MC}_i, $C_i(x)$ and I^{MC}_i values.

3.1 Motion Estimation

The results described in this paper were obtained using a block matching motion-estimation algorithm based on the algorithm described in [SGVP08]. We used macroblocks of size 16×16, 8×8 and 4×4 with adaptive partitioning criteria. Motion estimation is performed with quarter-pixel precision. Both luminance and chroma planes are considered. A confidence metric is calculated for quality assessment of the resultant motion vector field. The metric is similar to that described in [SGV08].

3.2 Depth Filtering

We use $2m+1$ consecutive frames for filtering. The first step is temporal median filtering.

$$D^{med}_n = \underset{\substack{i=n-m,\ldots,n+m \\ C_i > Th_C \\ |I^{MC} - I| < Th_{Diff}}}{med} D^{MC}_i.$$

The median is calculated over those pixels from the current and neighboring depth maps which have sufficiently small interframe difference $|I^{MC} - I|$ and well-estimated motion vectors. Median filtering eliminates sharp discontinuities in the time domain.

The next processing step is temporal smoothing:

$$D^{smooth}_n(x) = \frac{\sum\limits_{t=n-m}^{n+m} \sum\limits_{y \in \sigma(x)} \omega(t,x,y) \cdot D^{input}_t(y)}{\sum\limits_{t=n-m}^{n+m} \sum\limits_{y \in \sigma(x)} \omega(t,x,y)},$$

where $\omega(t,x,y)$ is a weight function, and D^{input}_t is the input depth map for this processing step. The source depth map D_i serves as the input depth map D^{input}_i for neighboring frames, and the filtered result D^{med}_n serves as the input D^{input}_n for the current frame. Previous fully processed depth maps D^{smooth}_{n-d} can be the input when processing current frame. This latter approach yields a smoother resulting depth map, but it is less accurate for small details. $\sigma(x)$ denotes the spatial neighborhood of pixel x. The size of $\sigma(x)$ involves a tradeoff between computation speed and processing quality. The weighting function ω is

$$\omega(t,x,y) = f(|I^{MC}_t(y) - I_n(y)|) \cdot C_t(y) \cdot g(x,y),$$

where function f describes the dependence on the inter-frame difference; $C_t(y)$ is the confidence for the motion compensation of pixel y in frame t; and g denotes the dependence of weight on the spatial distance between x and y. In the simplest case, g is constant. To improve quality, we tested other relationships between the spatial distance and weight: linear, polynomial, and exponential. Function f is given by the formula

$$f(x) = max\left(0, min\left(1, \sum_{i=0}^{3} \mu_i \cdot \left(\frac{x}{\nu}\right)^i\right)\right),$$

where μ_i and ν are parameters of the algorithm.

Thus, we average the depth value in the neighborhood of each pixel using information about interframe differences for the source video, the confidence metric for motion vectors, and spatial proximity.

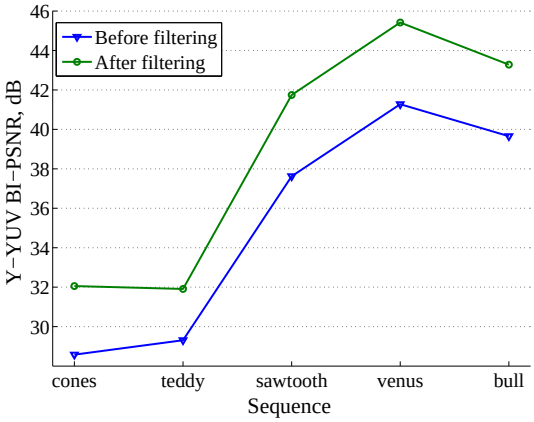
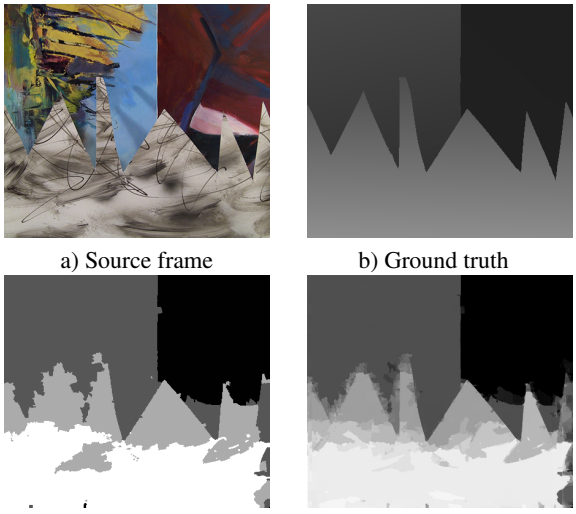


Figure 1: Results of the objective quality assessment. Depth maps were compared with ground truth depth before and after filtering. The comparison was performed using the Brightness Independent PSNR metric.



a) Source frame b) Ground truth

c) Estimated depth map d) Filtered depth map

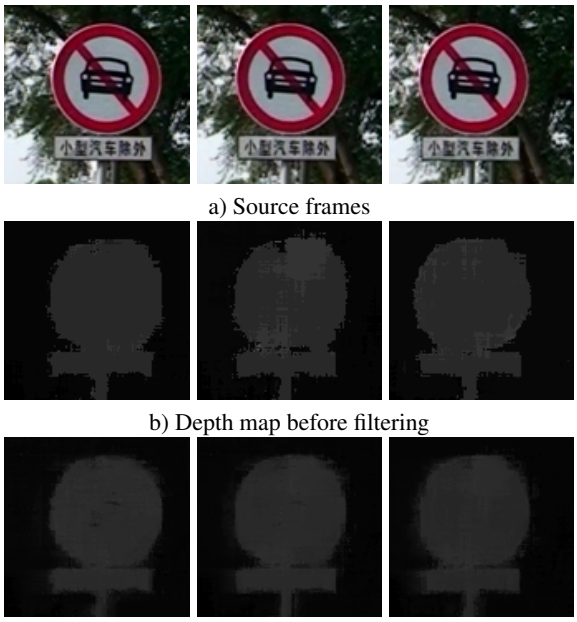
Figure 2: Comparison of results for ”Sawtooth” sequence.

4 RESULTS

The proposed algorithm was implemented in C as a console application. The algorithm uses one-pass processing, and it can be conveniently implemented in hardware. The algorithm's performance on an Intel Core2Duo T6670 processor running at 2.20 GHz is 7.6 fps for 448×372 video resolution.

For an objective evaluation, the standard sequences ”Cones,” ”Venus,” ”Sawtooth,” ”Teddy” and ”Bull” were used [SS02, SS03]. The comparison with ground truth was performed using the Brightness Independent PSNR metric [VNG]. Fig. 1 shows the results. The source depth maps were obtained using the depth-from-motion method based on that described in [OA05].

For a subjective evaluation, Fig. 2 depicts the results of the algorithm. The depth map (Fig. 2b) generated from the source video sequence (Fig. 2a) was filtered using the proposed method. The filtering process restored some lost details and fixed depth-estimation errors on object boundaries (Fig. 2d).



a) Source frames

b) Depth map before filtering

c) Depth map after filtering

Figure 3: Segments of three consecutive frames for ”Road” sequence.

a) Original depth map b) Filtered depth map

Figure 4: Depth-based rendered view for a segment of frame 112 for ”Road” sequence. The view based on filtered depth seems more natural.

For the test sequence ”Road” [ZJWB09], the proposed method improved the temporal stability of the depth map and recovered some details (see Fig. 3). The source depth map (Fig. 3b) has significant depth flickering. The proposed method improved the depth magnitude stability of objects in the scene (Fig. 3c).

We evaluated the visual quality of the rendered 3D image on the basis of an automatically generated depth map. The rendered view is shown in Fig. 4. The proposed method reduced artifacts in the resulting view and improved visual quality.

5 FURTHER WORK

In the proposed approach, we do not use occlusion detection and processing. In occlusion areas, motion estimation produces incorrect motion vectors, thus leading to artifacts. We intend to improve our confidence metric by implementing an algorithm for processing occlusion areas.

Better depth map quality can be achieved through spatial post-processing using information from the source video. Such filtering can be based on the assumption that uniform areas in the source video have uniform depth.

6 CONCLUSIONS

In this paper, we described a method of depth map filtering and presented a quality evaluation. The proposed algorithm improves the visual quality of depth maps and can simplify the manual work of 2D-to-3D video conversion. The described method allows the use of simpler and faster methods of automatic depth map generation without significant quality loss.

ACKNOWLEDGEMENTS

This research was partially supported by grant number 10-01-00697-a from the Russian Foundation for Basic Research.

REFERENCES

[CCL+05] Wan-Yu Chen, Yu-Lin Chang, Shyh-Feng Lin, Li-Fu Ding, and Liang-Gee Chen. Efficient depth image based rendering with edge dependent depth filter and interpolation. *IEEE International Conference on Multimedia and Expo*, 0:1314–1317, 2005.

[KCKA10] Sung-Yeol Kim, Ji-Ho Cho, Andreas Koschan, and Mongi A. Abidi. Spatial and temporal enhancement of depth images captured by a time-of-flight depth sensor. In *International Conference on Pattern Recognition (ICPR)*, pages 2358–2361. IEEE, 2010.

[KMS07] D.H. Kim, D.B. Min, and K.H. Sohn. Stereoscopic video generation method using motion analysis. In *3DTV Conference*, pages 1–4, 2007.

[LH09] Sang-Beom Lee and Yo-Sung Ho. Discontinuity-adaptive depth map filtering for 3d view generation. In *Proceedings of the 2nd International Conference on Immersive Telecommunications*, pages 1–6, 2009.

[OA05] Abhijit S. Ogale and Yiannis Aloimonos. Shape and the stereo correspondence problem. *International Journal of Computer Vision*, 65(3):147–162, 2005.

[SCN05] Ashutosh Saxena, Sung H. Chung, and Andrew Y. Ng. Learning depth from single monocular images. In *Advances in Neural Information Processing Systems 18*, pages 1161–1168. MIT Press, 2005.

[SGV08] K. Simonyan, S. Grishin, and D . Vatolin. Confidence measure for block-based motion vector field. In *GraphiCon*, pages 110–113, 2008.

[SGVP08] Karen Simonyan, Sergey Grishin, Dmitriy Vatolin, and Dmitriy Popov. Fast video super-resolution via classification. In *International Conference on Image Processing*, pages 349–352. IEEE, 2008.

[SS02] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002.

[SS03] Daniel Scharstein and Richard Szeliski. High-accuracy stereo depth maps using structured light. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:195, 2003.

[TZ04] W. J. Tam and L. Zhang. Non-uniform smoothing of depth maps before image-based rendering. In *Proceedings of Three-Dimensional TV, Video and Display III (ITCOM'04)*, volume 5599, pages 173–183, 2004.

[VNG] Dmitriy Vatolin, Alexey Noskov, and Sergey Grishin. MSU Brightness Independent PSNR (BI-PSNR). `http://compression.ru/video/quality_measure/metric_plugins/bi-psnr_en.htm`.

[ZJWB08] Guofeng Zhang, Jiaya Jia, Tien-Tsin Wong, and Hujun Bao. Recovering consistent video depth maps via bundle optimization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1 – 8, 2008.

[ZJWB09] Guofeng Zhang, Jiaya Jia, Tien-Tsin Wong, and Hujun Bao. Consistent depth maps recovery from a video sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):974–988, 2009.