TEMPORAL POST-PROCESSING METHOD FOR AUTOMATICALLY GENERATED DEPTH MAPS

Sergey Matyunin, Dmitriy Vatolin
Graphics & Media Lab, Moscow State University, Leninskiye Gory, Moscow, Russian Federation
smatyunin@graphics.cs.msu.ru, dmitriy@graphics.cs.msu.ru

Maxim Smirnov
YUVsoft Corp.
ms@yuvsoft.com

Keywords: Depth map, filtering, temporal post-processing, 3D video.

Abstract: Methods of automatic depth maps estimation are frequently used for 3D content creation. Such depth maps often contains errors. Depth filtering is used to decrease the noticeability of the errors during visualization. In this paper, we propose a method of temporal post-processing for automatically generated depth maps. Filtering is performed using color and motion information from the source video. A comparison of the results with test ground-truth sequences using the BI-PSNR metric is presented.

1 INTRODUCTION

Accurate and reliable depth information plays an important role in 3D video creation and processing. Creating a depth map for a conventional 2D video is a laborious process, so methods of automatic generation are under development. One of the promising approaches is depth reconstruction using object motion (Kim et al., 2007). In (Saxena et al., 2005), the authors propose a method of spatial structure analysis based on neural networks and machine learning.

Estimation of depth on the basis of stereo video (Ogale and Aloimonos, 2005) can be used for parallax tuning and nonlinear editing of 3D content.

The problem of definite depth reconstruction without additional information is generally unsolvable. For automatic depth reconstruction, methods 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 (Lee and Ho, 2009). In (Tam and Zhang, 2004), 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 (Chen et al., 2005). To increase the quality of the results, edge direction is taken into account.

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 spatial and temporal enhancement for depth maps captured by depth sensors was proposed in (Kim et al., 2010). Motion information is used to minimize depth flickering on stationary objects. This approach only considers the presence of the motion rather than the magnitude of the motion.

In (Zhang et al., 2009), 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 (Zhang et al., 2008) 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 (several minutes for each frame).

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

For the filtering of the current depth map D_n, the algorithm uses the neighboring source frames $I_{n-m},\dots,I_{n+m}$ and the depth maps $D_{n-m},\dots,D_{n+m}$. $I_i(x,y)$ denotes the intensity (or color) of pixel (x,y) in frame i. $I_i(x,y)$ is either a three-vector for a color image or a scalar for a grayscale image. The proposed method consists of four steps:

1. Motion estimation between the current frame I_n and neighboring frames $I_{n-m},\dots,I_{n-1}$, $I_{n+1},\dots,I_{n+m}$, where $m > 0$ is a parameter. The result of this stage is a field of motion vectors $MV_i(x,y) = (u^i(x,y), v^i(x,y))$. We define $MV_n(x,y) \equiv (0,0)$.
2. Computation of the confidence metric $C_i(x,y)$ for the resultant motion vectors $MV_i(x,y)$. Here, $C_i(x,y) \in [0,1]$. $C_i(x,y)$ quantifies the estimation quality for motion vector $MV_i(x,y)$.
3. Motion compensation for the depth maps and the source frames:

$$I_i^{MC}(x,y) = I_i(x+u^i(x,y), y+v^i(x,y)),$$

$$D_i^{MC}(x,y) = D_i(x+u^i(x,y), y+v^i(x,y)),$$

   where D_i^{MC} denotes the motion-compensated depth maps, and I_i^{MC} denotes the motion-compensated source frames.
4. Depth map filtering using the computed D_i^{MC}, C_i and I_i^{MC} values.

3.1 Motion Estimation

We used a block matching motion estimation algorithm based on the algorithm described in (Simonyan et al., 2008b). The algorithm uses 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.

3.2 Confidence metric

The motion estimation algorithm often produces wrong motion vectors, especially in the occlusion areas. Wrong motion estimation leads to artifacts. To reduce the influence of outliers we introduce confidence metric C for motion vectors. The metric is based on that described in (Simonyan et al., 2008a).

$$C = (1-\alpha) * C_{SAD} + \alpha * C_{MV},$$

where C_{SAD} responds to the motion-compensated interframe difference and C_{MV} responds to the spatial smoothness of motion vectors field in the spatial neighborhood of the current block. $\alpha \in [0,1]$ describes the smoothness of the current block.

3.3 Filtering

The filter consists of two stages. The first stage is temporal median filtering. Median filtering is often used to eliminate sharp discontinuities in the time domain. We apply filtering to the motion compensated frames to make our approach usable for the video sequences with fast motion. In order to reduce the influence of the errors of motion compensation we consider only those pixels (x,y) from the neighboring depth maps D_i^{MC} which have a good confidence metric value and small interframe difference $|I_i^{MC}(x,y) - I_n(x,y)|$:

$$\begin{aligned} D_n^{med}(x,y) = median\Big\{ & D_i^{MC}(x,y) \Big| \\ & i \in [n-m,\dots,n+m], C_i(x,y) > Th_C, \\ & |I_i^{MC}(x,y) - I_i(x,y)| < Th_{SAD} \Big\}. \end{aligned} \quad (1)$$

Thresholds Th_C and Th_{SAD} depend on the noise level of the source video.

The second stage of the filtering is temporal smoothing. We average the depth over the spatio-temporal neighborhood of the current pixel with weights $\omega(t,x,y,x',y')$:

$$D_n^{smooth}(x,y) = \frac{1}{k(x,y)} \sum_{t=n-m}^{n+m} \sum_{(x',y')\in\sigma(x,y)} (\omega(t,x,y,x',y') \cdot D_t^{med}(x',y')), \quad (2)$$

where $k(x,y)$ is a normalization term:

$$k(x,y) = \sum_{t=n-m}^{n+m} \sum_{(x',y')\in\sigma(x,y)} \omega(t,x,y,x',y'). \quad (3)$$

$\sigma(x,y)$ denotes the small spatial neighborhood of pixel (x,y). The size of $\sigma(x,y)$ is chosen as a tradeoff

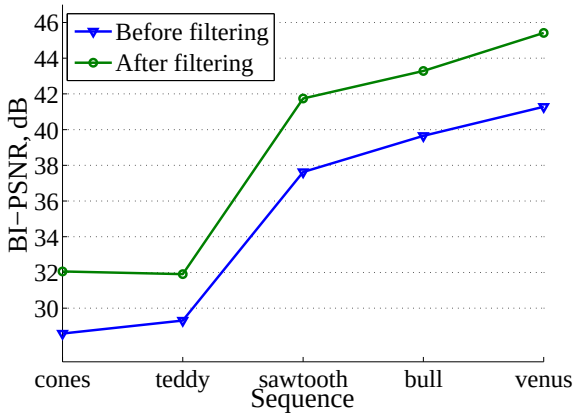


Figure 1: Results of the objective quality assessment. Depth maps were compared with ground truth depth before and after filtering. The comparison was performed using the Brightness Independent PSNR metric.

between computation speed and processing quality. Smaller $\sigma(x,y)$ produces worse results for noisy video.

Previous fully processed depth D_t^{smooth} can be used instead of D_t^{med} in (2). This approach yields a more stable result, but at the expense of quality of processing for small details.

Weighting function ω is given by

$$\omega(t,x,y,x',y') = f(t,x',y') \cdot C_t(x',y') \cdot g(x-x',y-y'),$$

where function $f(t,x',y')$ denotes the quadratic function of the motion-compensated inter-frame difference $|I_n(x',y') - I_t^{MC}(x',y')|$; $C_t(x',y')$ is the confidence metric of motion estimation of pixel (x',y') in frame t; $g(x,y)$ is a Gaussian function.

If the neighboring pixel (x',y') belongs to the area with good confidence and has a color similar to the color of current pixel (x,y), then the depth of $(x',y)'$ has more influence on the resulting depth of pixel (x,y). Thus, the algorithm averages the depth in the neighborhood of each pixel using the information about the motion-compensated interframe difference for the source video, the confidence metric for motion vectors, and the spatial proximity.

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. A time of the depth map estimation was not included in the measurement. The most time consuming stage of the algorithm is the motion estimation. Without regard for the motion estimation, the proposed method allows to filter the depth map in linear time with respect to the size of the image.

All the source depth maps for quality evaluation were obtained using the depth-from-motion method based on that described in (Ogale and Aloimonos, 2005). We used five consecutive frames for filtering ($m = 2$) in our experiments. $m > 3$ gives satisfactory results only on the stationary video sequences because of unreliable motion estimation for distant frames.

For an objective evaluation, the standard sequences "Cones," "Venus," "Sawtooth," "Teddy" and "Bull" were used (Scharstein and Szeliski, 2002; Scharstein and Szeliski, 2003). Each data set is a multi-baseline stereo sequence, i.e., the frames are taken from equally-spaced viewpoints along the horizontal axis. The latter allows to consider the ground-truth disparities as the representation of the depth. The comparison with ground truth was performed using the Brightness Independent PSNR metric (Vatolin et al.,). Fig. 1 shows the results.

For a subjective evaluation, Fig. 2 depicts the results of the algorithm for the sequence "Teddy". 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). Fig. 3 shows the results for the sequence "Cones".

One of the main drawbacks of the proposed algorithm is producing undesirable texture on some depth maps. This problem can be solved using spatial filtering. Cross bilateral post-processing (Petschnigg et al., 2004) gives rather good results (Fig. 4). The proposed method allowed to reduce artifacts on the resulting rendered view and improve visual quality.

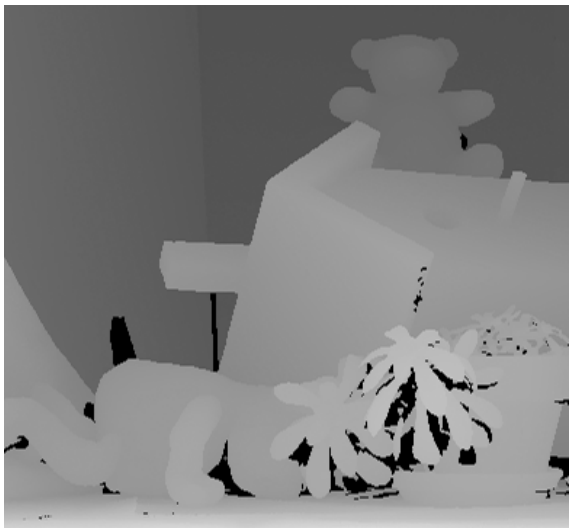
5 Conclusion

In this paper, we described a post-processing method algorithm for depth maps that are automatically generated from video. The proposed method shows high quality of filtering, uses single-pass processing, and doesn't require complex calculations (e.g., energy minimization or color segmentation). The algorithm's integration with a spatial post-filtering can be used as a high-performance processing method for the depth maps.

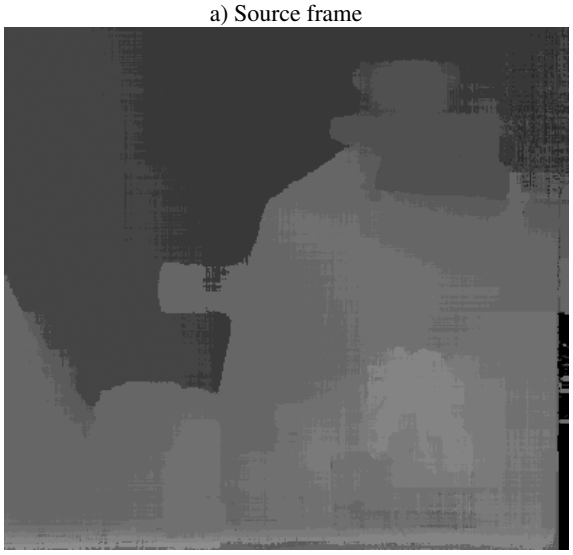
ACKNOWLEDGEMENTS

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

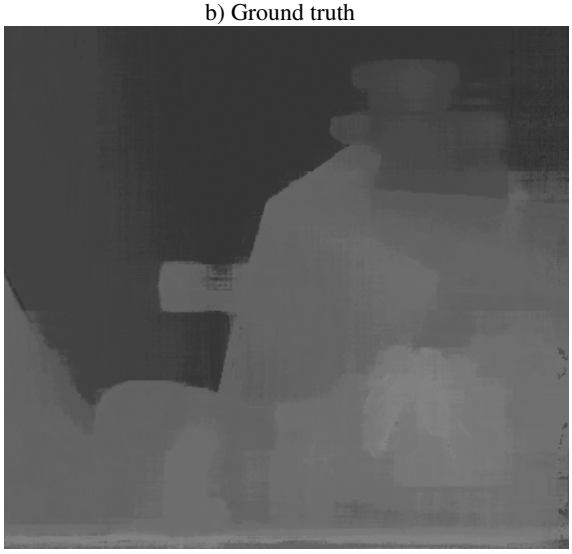
a) Source frame



b) Ground truth



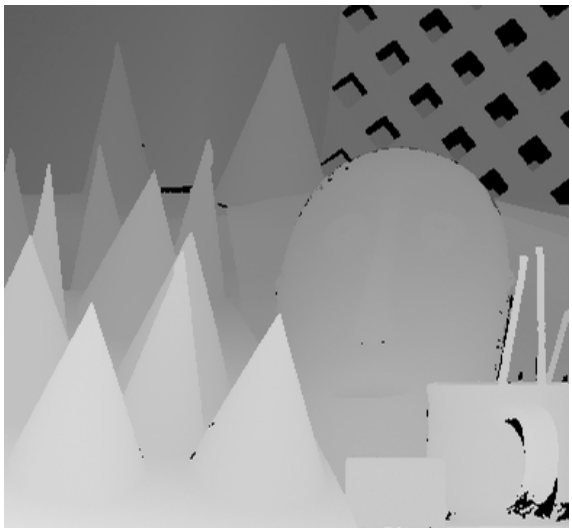
c) Estimated depth map



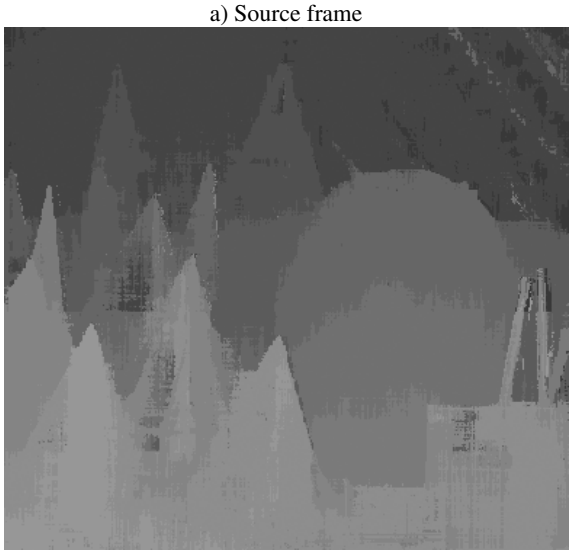
d) Filtered depth map

Figure 2: Sequence ”Teddy”.

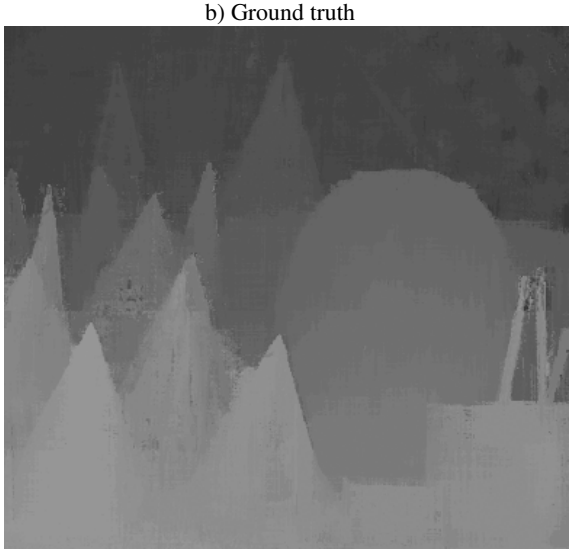
a) Source frame



b) Ground truth



c) Estimated depth map



d) Filtered depth map

Figure 3: Sequence ”Cones”.

a) Original 2D image

b) Rendered view for the estimated depth map

c) Rendered view for the filtered depth map

Figure 4: Depth based rendered view. Segment of "Road" sequence. The rendered view based on the filtered depth seems more natural. The depth map was filtered using the proposed method and the cross bilateral filtering. Occlusions are not processed.

REFERENCES

Chen, W.-Y., Chang, Y.-L., Lin, S.-F., Ding, L.-F., and Chen, L.-G. (2005). Efficient depth image based rendering with edge dependent depth filter and interpolation. *IEEE International Conference on Multimedia and Expo*, 0:1314–1317.

Kim, D., Min, D., and Sohn, K. (2007). Stereoscopic video generation method using motion analysis. In *3DTV Conference*, pages 1–4.

Kim, S.-Y., Cho, J.-H., Koschan, A., and Abidi, M. A. (2010). 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.

Lee, S.-B. and Ho, Y.-S. (2009). Discontinuity-adaptive depth map filtering for 3d view generation. In *Proceedings of the 2nd International Conference on Immersive Telecommunications*, pages 1–6.

Ogale, A. S. and Aloimonos, Y. (2005). Shape and the stereo correspondence problem. *International Journal of Computer Vision*, 65(3):147–162.

Petschnigg, G., Agrawala, M., Hoppe, H., Szeliski, R., Cohen, M., and Toyama, K. (2004). Digital photography with flash and no-flash image pairs. In *SIGGRAPH*, pages 664–672.

Saxena, A., Chung, S. H., and Ng, A. Y. (2005). Learning depth from single monocular images. In *Advances in Neural Information Processing Systems 18*, pages 1161–1168. MIT Press.

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

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

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

Simonyan, K., Grishin, S., Vatolin, D., and Popov, D. (2008b). Fast video super-resolution via classification. In *International Conference on Image Processing*, pages 349–352. IEEE.

Tam, W. J. and Zhang, L. (2004). 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.

Vatolin, D., Noskov, A., and Grishin, S. MSU Brightness Independent PSNR (BI-PSNR). `http://compression.ru/video/quality_measure/metric_plugins/bi-psnr_en.htm` .

Zhang, G., Jia, J., Wong, T.-T., and Bao, H. (2008). Recovering consistent video depth maps via bundle optimization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1 – 8.

Zhang, G., Jia, J., Wong, T.-T., and Bao, H. (2009). Consistent depth maps recovery from a video sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):974–988.