REAL-TIME DEPTH MAP OCCLUSION FILLING AND SCENE BACKGROUND RESTORATION FOR PROJECTED-PATTERN-BASED DEPTH CAMERAS

Yuriy Berdnikov

Moscow State University
Graphics & Media Lab
yberdnikov@graphics.cs.msu.su

Dmitriy Vatolin

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

ABSTRACT

In this paper we present our approach to real-time filtering of depth maps taken using projected-pattern-based depth cameras and to restoration of scene backgrounds for images taken using aligned RGB and depth cameras. An original depth map contains a numerous occlusions, and stereo-from-depth-map video generation leaves many uncovered areas. To solve this problem we propose an adaptive occlusion-filling algorithm for depth map processing and for restoration of scene backgrounds using depth map. Our goal is to accurately fill occlusions while maintaining real-time processing speed using common workstations.

Index Terms— depth map, disparity map, kinect, projected patterns, depth camera, background restoration, depth restoration

1. INTRODUCTION

In November, 2010 Microsoft released the first widely available and relatively inexpensive depth camera: Microsoft Kinect. This depth camera uses projected patterns technology for depth estimation: an infrared light source projects a dot pattern on the scene and an infrared camera captures an image. Because of the relative displacement of projector and camera, the displacement of the projected dots in the shot depends on the distance between the camera and the scene point. Such cameras have many drawbacks: they cannot work with shiny or self-illuminating surfaces, and all foreground objects cause occlusions. Moreover, a non-trivial problem crops up when filling occlusions for stereo pairs because of different foreground and background depths.

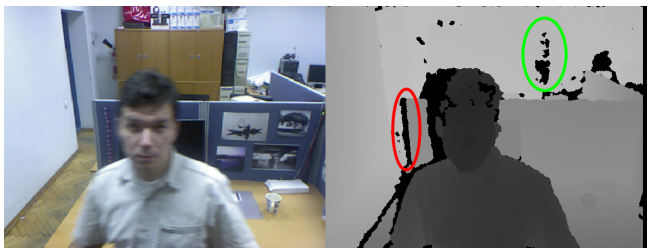
Approaches to filling occlusions filling usually work slowly, if they are to produce high-quality results. For example, the approach described in [1] uses normal maps and the AdaBoost classification algorithm, which requires training and extensive computational resources. The approaches of [2] and [3] require no training and work much faster, but they still cannot support real-time processing. Patch-based approaches [4] are highly parallel and could potentially be implemented using GPUs to achieve real-time speeds, but they have a common drawback: either they use only spatial information, or they use temporal information improperly? creating visual artifacts in moving scenes.

The approaches described in [5] and [6] are much more suitable for the present task because of simplicity. [5] uses a weighted combination of simple spatial and temporal inpainting; [6] uses fast spatial filling. Neither method. however, uses the specific characteristics of a depth map created using a projected patterns based depth camera, resulting in a priori decreasing their efficiency. To avoid such problems? we developed a new algorithm.

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

2. PROPOSED METHOD

2.1. Depth Map Occlusion Classification



(a) RGB frame (b) Depth frame

Fig. 1. Combined video and example of the occlusion classes.

There are two occlusion classes :

1) Occlusions caused by the edges of foreground objects (example marked on 1 with red color)

2) Occlusions caused by shiny surfaces and other object characteristics and random factors (example marked on 1 with green color)

To separate this cases, we developed the following separation criteria:

1) Occlusions on the foreground object edges exist only if the foreground is right of background (this is true for IR projector place in the right of IR camera)

2) The occlusion width equals the disparity difference between the foreground and background

$$W_{occ} = Disp(D_{foreground}) - Disp(D_{background})$$

Here $Disp(x)$ is a conversion function between the depth indicated by the camera and the pattern disparity. This function is unique to each exact camera. We approximate this function using a polynomial function

$$Disp(x) = a_n x^n + ... + a_1 x + a_0$$

where the coefficients a_n, ..., a_1, a_0 are obtained from the camera calibration datasheet or via manual calibration using sum-of-square-difference (SSD) optimization.

Analyzing each occlusion horizontally, we determine the most probable width for the occlusion

$$W_{mp} = Disp(D_f) - Disp(D_b)$$

$$D_f = \frac{\int_{\Omega_1} w_i D_i}{\int_{\Omega_1} w_i}$$

$$D_b = \frac{\int_{\Omega_2} w_i D_i}{\int_{\Omega_2} w_i}$$

Here, $D(i)$ is the depth of the current pixel, Ω_1 and Ω_2 are the foreground and background areas neighboring the current occlusion, and w_i is the weight of current pixel; this weight depends on the pixel's distance from the occlusion.

Comparing W_{mp} and W_{curr}, we can determine which class this occlusion area belongs to.

2.2. Depth Map Occlusion Filling

For occlusion class 1 we propose using "deepest neighbor" method:

$$D_i = max(D_f, D_b)$$

This method is physically correct, because such occlusions are caused by the object's IR shadow, and these areas must be threated as a background.

For occlusion class 2, we propose using the interpolation method:

$$D_i = \frac{\alpha D_f + \beta D_b}{\alpha + \beta}$$

where α and β are the distances to the nearest foreground and background pixels.

2.3. Background Estimation and Scene Occlusion Filling

Assuming that the camera is static or that the camera motion has been compensated using some appropriate method, we can use the "deepest in history" method for background reconstruction: if the depth of the current pixel is greater than the depth of the current background, we update the background depth and RGB values.

To avoid missing changes in the background color (like camera noise or a dynamic background such as TV-set), we update the background depth values only if $D_{new} > D_{current}$, and we update the RGB values when $D_{new} > \delta D_{current} - \gamma$, where $\delta \leq 1$ and $\gamma \geq 0$

To generate an image using a virtual displaced camera, we preform object displacement on the basis of the object's depth for the current scene and the restored background, and we fill occlusions in the current scene using background data.

3. RESULTS

First video sequence is 300 frames long with moderate foreground motion and static background, 180th frame was taken as an example. Second video sequence is 200 frames long, contains very slow motion and camera shaking and 150th frame was taken as an example.

Both video sequences are taken using Microsoft Kinect, 640x480 @ 30 fps.

Algorithm has been implemented in c++ and show real-time performance on Intel Core I5 2.56 GHz.

Fig. 2. Source RGB frame with corrected zoom and camera misalignment

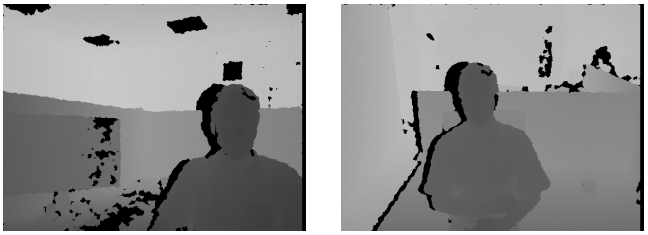


Fig. 3. Source depth frame

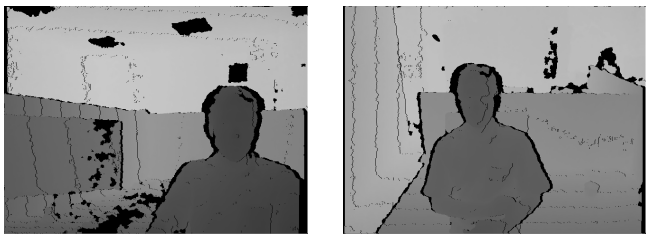


Fig. 4. Depth frame with corrected camera displacement

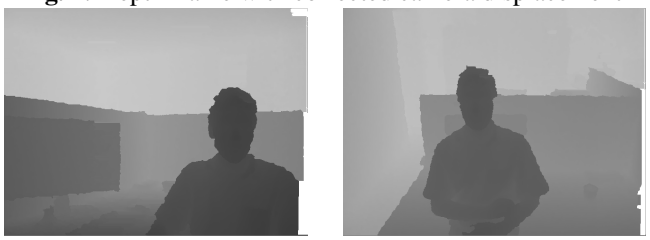


Fig. 5. Depth frame after occlusion filling

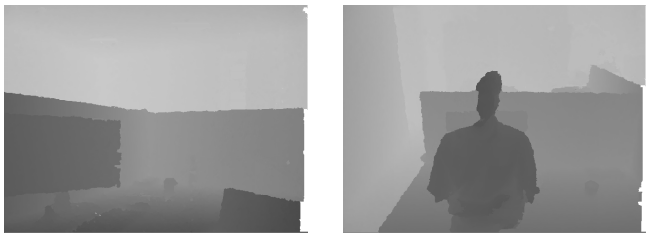


Fig. 6. Estimated scene background depth

4. FURTHER WORK

In the short term, we plan to improve the occlusion filling algorithm by considering temporal information and by using fast motion-compensation algorithms to enable the use of non-static cameras.

5. CONCLUSION

In this work we presented our approach to real-time depth map filtering and occlusion filling as well as background restoration for video that combiner RGB and depth information, where the depth was determined using a projected-patterns-based depth camera. Lastly, we described our intended directions for future work.

6. ACKNOWLEDGMENTS

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

7. REFERENCES

[1] Christoph Strecha Pascal Fua Engin Tola, Andrea Fossati, "Large occlusion completion using normal maps," *ACCV*, November 2010.

[2] C. Rother J. Shotton A. Criminisi, A. Blake and P. H. S. Torr, "Efficient dense stereo with occlusions for new view-synthesis by four-state dynamic programming," *International Journal of Computer Vision*, vol. 71, num. 1, pp. 89–110, November 2010.

[3] Atanas Gotchev Lucio Azzari, Federica Battisti, "Comparative analysis of occlusion-filling techniques in depth image-based rendering for 3d videos," *ACM Multimedia*, 2010.

[4] Kentaro Toyama Antonio Criminisi, Patrick Perez, "Object removal by exemplar-based inpainting," *WI Proc. IEEE Computer Vision and Pattern Recognition*, 2003.

[5] Barenburg B. Magalhaes J. P. Gunnewick R. Klein, Berretty R-P. M., "Coherent spatial and temporal occlusion generation," *SPIE, the International Society for Optical Engineering*, ISSN 0277-786X 2009 2010.

[6] Richard McKenna Yu-Sung Chang Manuel M. Oliveira, Brian Bowen, "Fast digital image inpainting," *Proceedings of the International Conference on Visualization, Imaging and Image Processing*, 2001.

Fig. 7. Estimated scene background RGB

Fig. 8. 3D visualization without background restoration

Fig. 9. 3D visualization with background restoration

Fig. 10. Source RGB frame with corrected zoom and camera mispointing

Fig. 11. Source depth frame

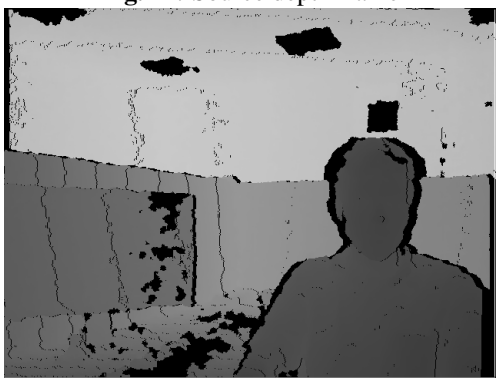


Fig. 12. Depth frame with corrected camera displacement

Fig. 13. Depth frame after occlusion filling

Fig. 14. Estimated scene background depth

Fig. 15. Estimated scene background RGB

Fig. 16. 3D visualization without background restoration

Fig. 17. 3D visualization with background restoration