Chapter 6
Dense 3D Reconstruction in Multi-camera Systems

Soonmin Bae

Abstract The 3D information is useful in the security services and media production markets. However, it is difficult to achieve a reliable and dense 3D reconstruction from images. In this paper, we introduce a new approach to generating a dense 3D map from stereo images. We estimate the epipolar geometry and find matches using the geometric model. Given matches, we triangulate the 3D points and propagate the 3D information to neighboring pixels. For propagation, we use segmentation for a plane fitting and non-homogeneous optimization for an edge-preserving smoothing.

Keywords 3D reconstruction • Stereo

6.1 Introduction

Multi-camera systems become popular in recent years. Surveillance systems use them to monitor specific areas. When multi-camera systems need to track a moving target, it is essential to pass the 3D information to each other (Fig. 6.1).

Media production uses multi-camera techniques to broadcast sports. The growing 3D TV market will demand more advanced techniques.

However, it is difficult to achieve a reliable and dense 3D reconstruction from images. This paper introduces a new approach to generating a 3D map from non-rectified stereo images. We estimate the epipolar geometry find matches using the geometric model. Given matches, we triangulate the 3D points and propagate the 3D information to neighboring pixels. For propagation, we segment the input image and apply a plane fitting. In addition, we perform non-homogeneous optimization for an edge-preserving smoothing.

S. Bae (✉)
Intelligent Control Technology Team, Advanced Technology Center, Samsung Techwin, Seongnam, South Korea
e-mail: soonmin.bae@samsung.com; soonminb@gmail.com

R. Tutsch et al. (eds.), *Progress in Optomechatronic Technologies*,
Lecture Notes in Electrical Engineering 306, DOI: 10.1007/978-3-319-05711-8_6,

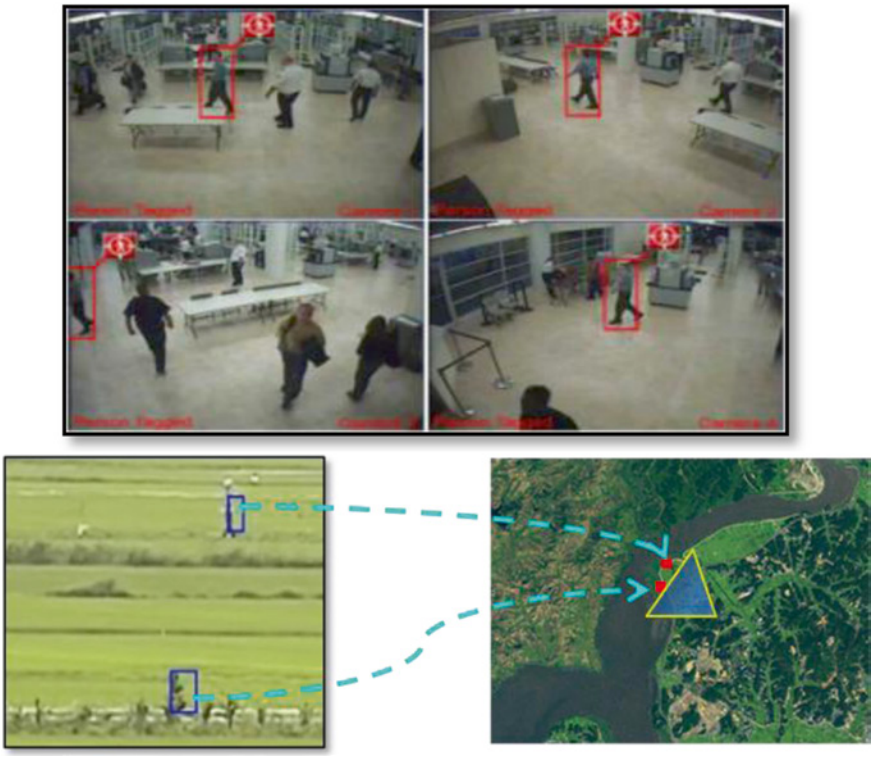

Fig. 6.1 Multi-camera surveillance systems

Although there exist previous methods which reconstruct 3D structures using stereo and segmentation [1, 4, 7], they deal with rather simple depth maps and rely on quite complicated algorithms.

6.2 Algorithm Description

For each pixel, we estimate a depth value. We call our depth estimation the depth map. We estimate the depth map in three steps. First, we estimate the depth value at corners. Then, we segment the image and apply a 3D plane-fitting to each segment given the sparse depth values. Finally, we propagate this depth measure to the rest of the image.

6.2.1 Epipolar Geometry Estimation and Triangulation

Given two non-rectified stereo images (Fig. 6.2), we estimate the epipolar geometry [3]. Since we assume the images are calibrated, we estimate the essential matrix. We lower the threshold for the corner detection to detect more matches (Fig. 6.3).

Fig. 6.2 Input images. They are not rectified, but calibrated

Fig. 6.3 Corner detection results (*red* and *green points*) and resulting matches (*blue lines*) satisfying the epipolar geometry

We triangulate the resulting matches and compute the initial 3D points. Although we increased the number of matches, they are points in the corners and do not cover the whole image. Therefore, we propagate the initial 3D points to the whole image (Fig. 6.4).

6.2.2 Segmentation and Plane-Fitting

To propagate the initial sparse depth value to the neighboring pixels, we segment the input image. After segmentation [2], we explicitly enforce the depth discontinuities in the segmentation boundaries and the depth continuities inside each segment using a plane-fitting. This results in sharp discontinuities and smooth planes (Figs. 6.5 and 6.6).

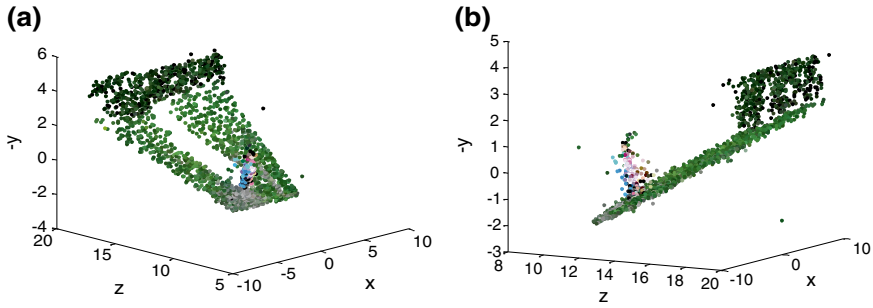

Fig. 6.4 The initial sparse 3D from two different views of the first example in Figs. 6.2 and 6.3. The horizontal rotation and vertical elevation in degrees are **a** −45 and 20, and **b** 60 and −10 respectively

Fig. 6.5 Segmentation results

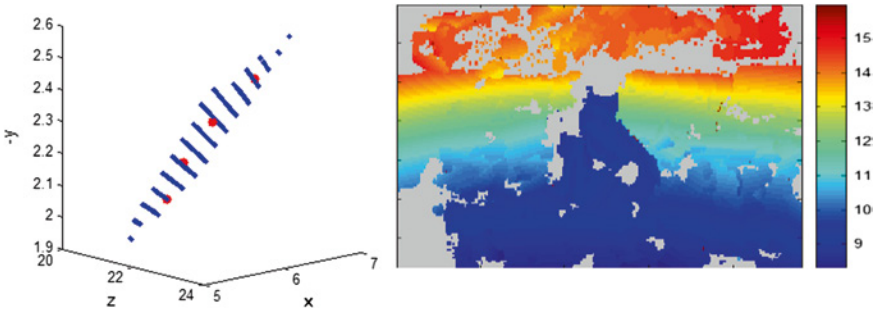

Fig. 6.6 Plane fitting example and plane fitting results. The *red values* means far and *blue* means near

To find a best-fitting plane, we use RANSAC (RANdom SAmple Consensus) [3] and PCA (Principal Component Analysis). We randomly select four initial depth values in a segment and estimate a corresponding plane using SVD (Singular Value Decomposition). We iterate the above process and find a best-fitting plane with a

smallest fitting error. We register the plane if the plane satisfies more than 60 % of the initial depth values with less than 0.1 fitting error in the segment and if the area containing the initial depth values projected to the selected plane covers more than 20 % of the area of the segment projected to the plane.

However, to fit a plane, there should be at least three matches and there exist segments with less than three matches. To resolve this issue, we use an edge-preserving smoothing.

6.2.3 Edge-Preserving Smoothing

For the segments where a plane cannot be fitted, we propagate the initial 3D maps to the whole image using non-homogeneous optimization [5]. Our assumption is that the depth varies smoothly over the image except where the color is discontinuous. We propagate the depth value to the neighbors with similar color.

We impose the constraint that neighboring pixels p, q have similar depth if they have similar colors. We minimize the difference between the depth $D(p)$ and a weighted average of depth of neighboring pixels:

Non-homogeneous optimization:

$$\begin{aligned} E(D) &= \sum (D(p) - \sum_{q \in N(p)} w_{pq} D(q))^2 + \sum \alpha_p (D(p) - ID(p))^2 \\ \text{with: } w_{pq} &\propto \exp\left(\frac{-(C(p) - C(q))^2}{\sigma_p^2}\right) \end{aligned} \tag{6.1}$$

where $ID(p)$ is the initial depth measure from the plane fitting and σ_p is the standard deviation of the colors of neighboring pixels in a window around p. The window size used is 21×21. We have experimented both with setting the second term as hard constraints versus a quadratic data term, and have found that the latter is more robust to potential remaining errors in the depth measure.

6.3 Results

We have implemented our dense 3D reconstruction using Matlab. We crop the upper and lower 2 % of the depth map.

6.3.1 Depth Map Results

Figure 6.7 presents the depth map of the input images in Fig. 6.2. Figure 6.8 shows an additional example and result.

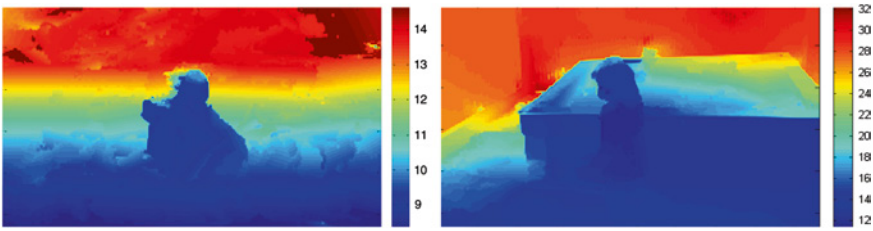

Fig. 6.7 The depth map results of the input images in Fig. 6.2. The *red* values means far and *blue* means near

Fig. 6.8 An additional result. The *red* values means far and *blue* means near

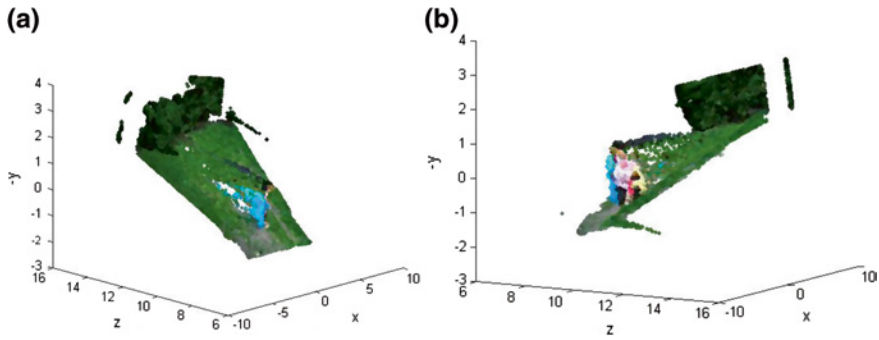

Fig. 6.9 The final dense 3D from two different views of the example in Fig. 6.4. The depth map clearly shows the depth discontinuities between the girl and bush and continuities in the grass and bush. The horizontal rotation and vertical elevation in degrees are **a** −45 and 20, and **b** 60 and −10 respectively

6.3.2 3D Results

Since we assume the camera is calibrated, we can compute the normalized coordinate for each pixel. As we multiply the depth value to the normalized coordinate, we achieve the 3D values for each pixel (Fig. 6.9).

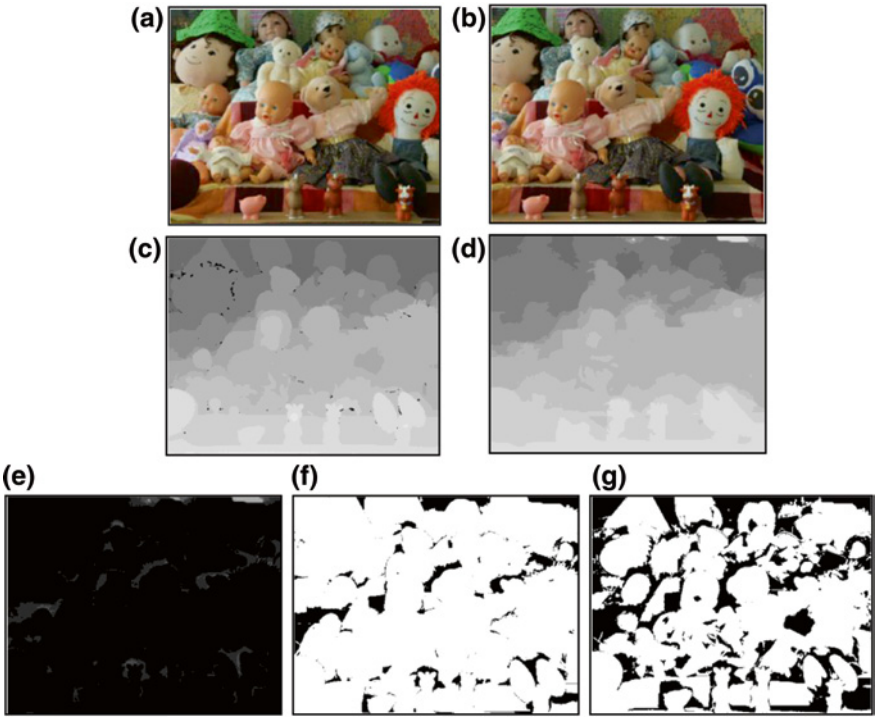

Fig. 6.10 Evaluation results. **a** Left image, **b** right image, **c** ground-truth disparities, **d** our result, **e** disparity differences, **f** pixels with less than 3 pixel disparity error, and **g** pixels with less than 1 pixel disparity error. 92 % pixels have less than a 3 pixel-error, and 85 % pixels have less than a 1 pixel-error. The median error ratio is 4.2 %

6.3.3 Evaluation

We evaluate the accuracy of our 3D using the Middlebury stereo datasets at http://vision.middlebury.edu/stereo/data/ [6]. Although the datasets are rectified, we do not rely on the fact that there is no rotation between the left and right cameras. The Middlebury stereo datasets provide the ground-truth disparities from a structured-lighting approach. We convert our depth map into disparity map using the following equation.

Conversion between depth and disparity:

$$Depth = \frac{focal\ length}{disparity} \tag{6.2}$$

Figures 6.10 and 6.11 show the results. They present that the corresponding disparity map from our 3D is very close to the ground-truth.

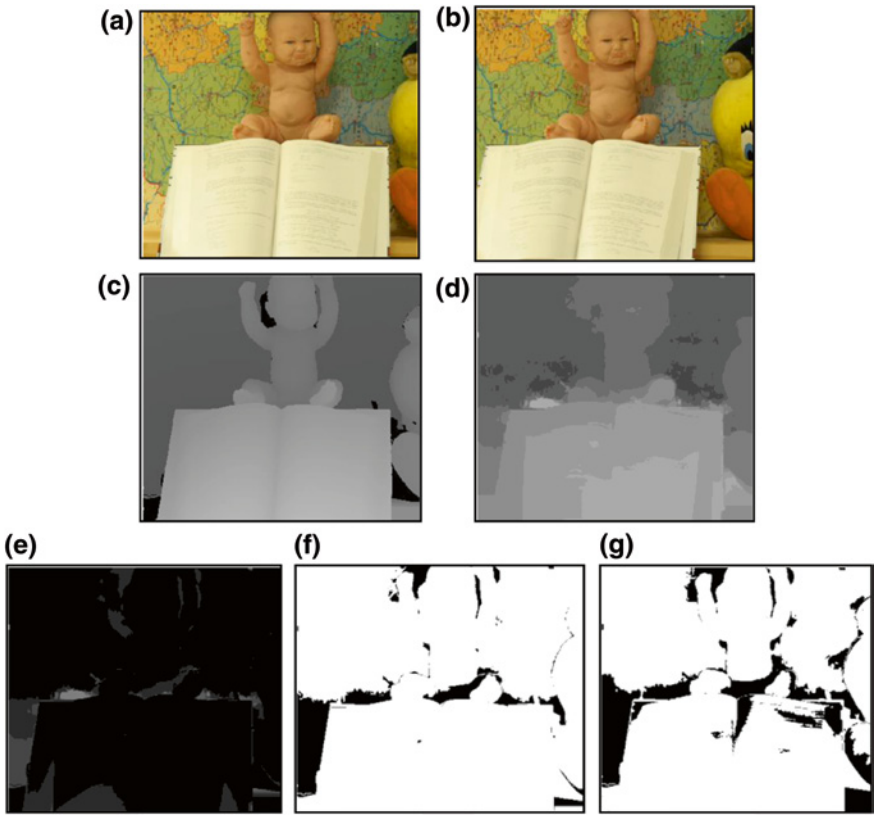

Fig. 6.11 More evaluation results. **a** Left image, **b** right image, **c** ground-truth disparities, **d** our result, **e** disparity differences, **f** pixels with less than 3 pixel disparity error, and **g** pixels with less than 1 pixel disparity error. 91 % pixels have less than a 3 pixel-error, and 72 % pixels have less than a 1 pixel-error. The median error ratio is 1.9 %

In Fig. 6.10, 92 % pixels have less than a 3 pixel-error, and 85 % pixels have less than a 1 pixel-error. The median error ratio is 4.2 %. In Fig. 6.11, 91 % pixels have less than a 3 pixel-error, and 72 % pixels have less than a 1 pixel-error. The median error ratio is 1.9 %.

6.4 Conclusion

This paper introduces a new approach to reconstructing a dense 3D from a sparse 3D. Given a pair of non-rectified images, we estimate sparse depth values at corners and propagate the 3D information to the neighboring pixels using a plane-fitting and edge-preserving smoothing. Our assumption is that depth is piece-wise smooth

and particularly smooth where colors are similar. We verify the accuracy of our 3D using the ground-truth disparity maps.

We envision that the resulting dense 3D reconstruction will be of benefit to the security services and media production markets.

References

1. Bleyer M, Gelautz M (2005) Graph-based surface reconstruction from stereo pairs using image segmentation. Proc SPIE 5665:288–299
2. Comaniciu D, Meer P (2002) Mean shift: a robust approach toward feature space analysis. IEEE Trans Pattern Anal Mach Intell 24:603–619
3. Hartley R, Zisserman A (2000) Multiple view geometry in computer vision. Cambridge Press, Cambridge
4. Hong L, Chen G (2004) Segment-based stereo matching using graph cuts. In: IEEE Computer Society conference on computer vision and pattern recognition
5. Levin A et al (2004) Colorization using optimization. ACM Transactions on Graphics, In: Proceedings of ACM SIGGRAPH conference 23(3):689–694
6. Scharstein C, Pal C (2007) Learning conditional random fields for stereo. In: IEEE Computer Society conference on computer vision and pattern recognition
7. Strecha C et al (2003) Dense matching of multiple wide-baseline views. In: IEEE international conference on computer vision