Plenoptic depth map in the case of occlusions

Zhan Yu[a], Jingyi Yu[a], Andrew Lumsdaine[b],

Todor Georgiev[c]

[a]University of Delaware, Newark, DE 19716, USA;
[b]Indiana University, Bloomington, IN 47405, USA;
[c]Qualcomm, San Diego, CA 92121, USA;

ABSTRACT

Recent realizations of hand-held plenoptic cameras have given rise to previously unexplored effects in photography. Designing a mobile phone plenoptic camera is becoming feasible with the significant increase of computing power of mobile devices and the introduction of System on a Chip. However, capturing high numbers of views is still impractical due to special requirements such as ultra-thin camera and low costs. In this paper, we analyze a mobile plenoptic camera solution with a small number of views. Such a camera can produce a refocusable high resolution final image if a depth map is generated for every pixel in the sparse set of views. With the captured multi-view images, the obstacle to recovering a high-resolution depth is occlusions. To robustly resolve these, we first analyze the behavior of pixels in such situations. We show that even under severe occlusion, one can still distinguish different depth layers based on statistics. We estimate the depth of each pixel by discretizing the space in the scene and conducting plane sweeping. Specifically, for each given depth, we gather all corresponding pixels from other views and model the in-focus pixels as a Gaussian distribution. We show how it is possible to distinguish occlusion pixels, and in-focus pixels in order to find the depths. Final depth maps are computed in real scenes captured by a mobile plenoptic camera.

Keywords: Depth Estimation, Computational Photography, Plenoptic Cameras

1. INTRODUCTION

Recent realizations of hand-held plenoptic cameras have given rise to previously unexplored effects in photography. A plenoptic camera [1–4] uses a microlens array to capture the 4D radiance of a scene. The acquired radiance, as an integral image, can be processed for either 3D scene reconstruction or synthesizing dynamic depth of field (DoF) effect. There are numerous applications for this emerging camera technology, ranging from entertainment (Lytro[5]) to depth recovery for industrial and scientific applications (Raytrix[6]).

Essentially, a plenoptic camera is a single-shot, multi-view acquisition device. Even though a high resolution sensor is commonly used for overcoming the spatio-angular tradeoff issue ,[4] the resulting images, however, are still at a disappointingly low resolution. The Adobe light field camera[4] captures 20 different views of a scene with a 10 megapixel sensor. The rendered 700×700 images have visible artifacts at occlusion boundaries. Ng[1] proposed a different design with a 296×296 microlens array covering a 16 megapixel sensor. The dense angular resolution greatly suppressed artifacts with higher refocusing power. Nevertheless, the image resolution is as low as the number of microlenses in the camera (296×296). The commodity Lytro light field camera uses a 11 megapixel sensor to acquire the radiance. Following Ng's design, the images generated from the camera still suffer from a low resolution of 0.7 megapixel, with some visible artifacts around thin objects and sharp edges.

To generate high quality images, in this paper, we explore the solution of increasing the spatial resolution in the design and interpolating the angular resolution in post processing. The reason we choose this trend is that capturing high number of views is still impractical due to special requirements such as ultra-thin camera and low costs. However, we can take advantage of the cheap but high resolution sensor of low cost cameras and

Further author information: (Send correspondence to Zhan Yu)
Zhan Yu: E-mail: yshmzhan@udel.edu, Telephone: +1(302)740-3485
Todor Georgiev: E-mail: todorg@qualcomm.com

easily build a multi-view system on mobile devices. In this paper, we analyze a plenoptic camera solution with small numbers of views. Such a camera can produce refocusable high resolution final image if a depth map is generated for every pixel in the sparse set of views.

With the captured multi-view images, the obstacle to recover a high-resolution depth is occlusions. To robustly resolve these, we first analyze the behavior of pixels in such situations. We show that even under severe occlusion, one can still distinguish different depth layers based on statistics. We estimate the depth of each pixel by discretizing the space in the scene and conducting plane sweeping. Specifically, for each given depth, we gather all corresponding pixels from other views and model the in-focus pixels as a Gaussian distribution. To resolve the occlusion issue among different views, we apply an iterative process to accurately estimate depth layers from the closest to the furthest, so that the occlusion pixels will be masked out when estimating local minima. However, pixels on constant color surfaces tend to choose small disparity since they will lead to small variance. To avoid these trivial solutions, we propose a global optimization solution and an edge mask solution. Experimental results show that our algorithm is able to recover accurate depth map from the integral image of real scenes captured by a plenoptic camera.

2. RELATED WORK

Radiance Acquisition. Integral or light field photography has its roots in the methods introduced by Lippmann[7] and Ives[8] over 100 years ago. During the last twenty years, numerous integral cameras have been built.[3,9,10] Early camera-array based systems[11] can capture high spatial resolution radiance but are bulky and expensive. Recent approaches aim to capture the radiance using a single commodity camera. Ng[12] designs a hand-held LF camera which places a microlens array in front of the camera sensor for separating converging rays. This design has led to the commercial LF camera Lytro.[5] Lumsdaine et al.[13] introduce a slightly different design by focusing the microlens array on a virtual plane inside camera. In this case each microlens image captures more spatial samples but fewer angular samples on the refocusing plane. Both Ng's and Lumsdaine's designs match the F-numbers of the main lens and each microlens to avoid cross-talk among microlens images.

Depth Estimation from Radiance. The plenoptic cameras have renewed the interest on multi-view depth estimation. The seminal work by Kolmogorov and Zabih[14] extends binocular stereo to multi-view stereo using the graph-cut framework. The solution, in essence, adds an additional occlusion term to the data term and smoothness term to handle complex occlusions. Using imagery from the light field camera, Georgiev et al.[15] apply a window based algorithm for producing coarse disparity maps to guide digital refocusing. Most recently, Wanner and Goldlücke[16] develop a structure tensor based approach to measure each pixel's direction in 2D EPI. They then encode the estimated edge directions into dense depth estimation for global optimization. Despite these advances, very few multi-view stereo matching algorithms managed to recover highly accurate depth maps under severe occlusion conditions, which is very common in captured radiance by plenoptic cameras. In this paper, we demonstrate how to iteratively resolve the occlusion problem when computing local minimum for depth estimation.

3. DEPTH ESTIMATION FROM RADIANCE

To estimate the scene depth based on the captured integral image, without loss of generality, we consider the behavior of a pixel p_0 in the integral image. This pixel represents view v_0, and is related to different sets of views when assigning different depth d to it.

3.1 No occlusion

Consider Figure 1, where we assume all the surfaces in the scene are Lambertian. If p_0 is assigned the correct depth d, it maps to a point P on a surface. All rays emitted from P have constant color. Therefore, rays captured by any other view v_i at pixel p_i will have the same color as p_0. On the other hand, if incorrect depth d' is assigned to p_0, then the corresponding p_i will tend to have different color than p_0.

With this observation, when assigning a depth d_j to a pixel, we model the distribution of color over all pixels from different views as a unimodel Gaussian distribution to further compensate for the vignetting effect and

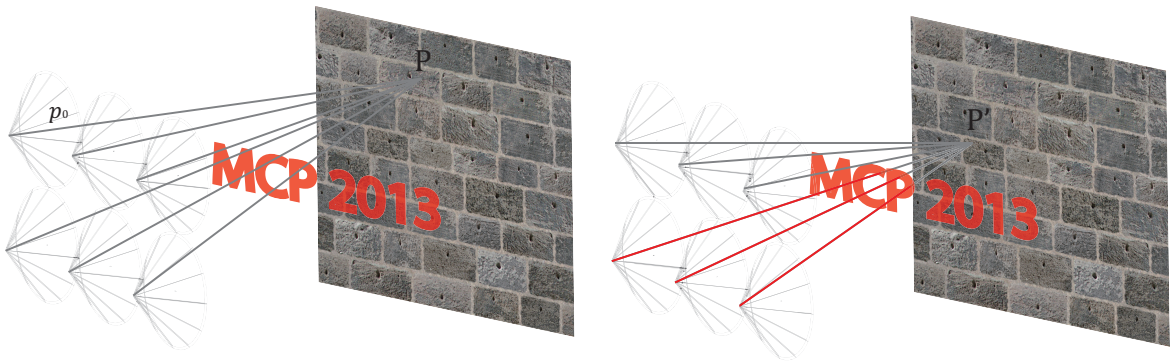

Figure 1. Color sampled by cameras without (left) or with (right) occlusion.

camera noise. And the variance of the distribution defines the possibility of the p_0 actually lying on d_i. It is computed by:

$$V_{p_0,d_j} = \frac{\sum (I_{p_i} - \bar{I})^2}{N}, \tag{1}$$

where I_{p_i} is the intensity of p_i and N is the number of pixels associated. If V_{p_0,d_j} is small, meaning all the pixels have almost the same color, the probability of p_0 having depth d_i is high.

3.2 Depth estimation with occlusion

Consider Figure 1, where some of the views looking at P' are occluded. In this case, even with a correctly assigned depth, due to occlusion, some rays emitted from the front surfaces replace the correct rays from the back surface, resulting in high variance in our Gaussian model.

To resolve this issue, Yu et al.[17] assume occlusion surfaces have similar color and model the problem with a bimodel Gaussian distribution. One can easily extend this approach to a N-model Gaussian distribution but deciding N is rather difficult. However, having similar color on all occlusion surfaces is a rather extreme assumption. Moreover, under a small number of views, sometimes there are not enough pixels to form Gaussian distribution. The state of the art globally consistent depth labeling method[16] proposed global labeling constraints on epipolar plane images (EPI). But it requires a densely sampled radiance (at least 17×17) in order to estimate local direction on the EPI. Therefore it does not fit our sparse sampling situation. However, to show the robustness of our algorithm, we still compare our result with this algorithm by providing more views.

Next, we analyze the distribution of pixel intensities. In the regular case, images of P are still captured by some of the views. In this case, the Gaussian distribution still holds, but with noise around the region far from the mean. It is possible to explicitly separate out the occlusion samples or implicitly model this distribution as N-modal. However, in the extreme case where most samples are from occlusion surfaces, it is almost impossible to tell which samples are from the in-focus plane from a single observation.

Instead of trying to point out which samples are outliers directly from a single observation under a given depth, we propose to iteratively mask out the layers that are in front of P. For each iteration, we still loop over all the depth values to check for the minimum variance for each pixel. The difference is that starting from the second interaction, we make use of the current depth map and when testing depth d_i on P, we ignore pixels that have smaller depth than d_i.

Now we analyze this idea in detail. We assume that the scene depth is from d_{min} to d_{max}, and that there are sufficient number of views to form different distributions when assigned with different disparities. It is also reasonable to assume that if all the occlusion pixels are masked out, the intensity distribution will achieve minimum variance at the correct depth value. In the first iteration, we can successfully find the local minimum for the closest depth since no occlusion will occur on those pixels.

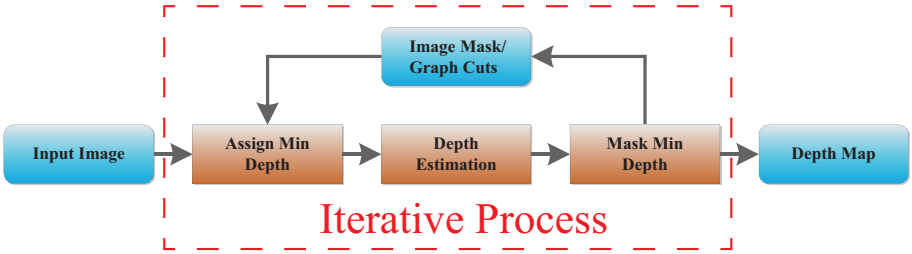

Figure 2. Our depth estimation pipeline.

The next iteration, we mask out those pixels when computing the depth for all pixels since they are considered as occlusions. Note that pixels at d_{min} may also be assigned depth d_{min} during the first iteration due to occlusion problems. However, by masking out all the pixels at d_{min}, our algorithm is able to guarantee that no pixels from d_{min} will affect the converges of pixels at d_{min+1}. Therefore during the second iteration, all pixels on d_1 will be computed under no occlusion condition. And so on.

Now we prove that in each iteration, our estimation is occlusion free.

Base case. In iteration 0, all the depth are computed directly using the unimodel Gaussian distribution. In this case, all the pixels on d_0 will be marked out correctly.

Induction. Suppose in iteration n, depths smaller than d_n are all computed correctly, in iteration $n+1$, we ignore pixels with depths up to smaller than d_{n+1}. So that pixels with d_{n+1} can be computed with no occlusion.

However, as mentioned above, at each iteration i, pixels not lying on depth d_i may be incorrectly assigned with d_i due to occlusion. In this case, in a later iteration, more than necessary pixels may get masked out, so that with a small disparity, a trivial solution with small variance could be produced for pixels on the constant color surfaces. In this case, another process would be required to either use the boundary pixels to regulate these pixels (global optimization) or use a edge mask to ignore the pixels on the surfaces in a later iteration (edge mask).

4. AVOIDING THE TRIVIAL SOLUTION

Note that in each iteration, not only pixels lower than current depth will be masked out; pixels incorrectly marked as having depths lower than the current depth will also be masked out due to occlusion. To resolve this issue, we further propose two solutions: (1) an edge mask is required to mark those surfaces, and (2) a global optimization process using GC.

4.1 Edge Mask

The second part is done iteratively on the edge pixels only. Usually, more than 2 iterations are conducted for an input image. To avoid occlusions, we sweep through all trial depths from low to high and compute the variance at each by not using pixels that have disparity higher than the current disparity. In each iteration, we update the depth at each pixel to be the disparity at lowest variance and continue with another iteration. Typically 5 iterations are sufficient to produce good results.

4.2 Global Optimization

Constant color surfaces in the scene are always a problem since it is almost impossible to estimate depth directly from them. Traditional global optimization methods such as graph cuts or belief propagation use a smoothness constraint to compensate for this issue. We embed our algorithm into the graph cuts framework and let the smoothness constraint to resolve the trivial solution issue. Specifically, in each iteration, we minimize the energy function by constructing a graph with data term (variance of pixel intensities) as the links to source/target and smoothness term (depth difference between neighboring pixels) as links to neighboring pixels. In this case, we can reuse min-cut/max-flow algorithm to minimize the energy function.[14,18] Note that the data term in our case is occlusion free because we do not consider pixels with depth lower than the current depth.

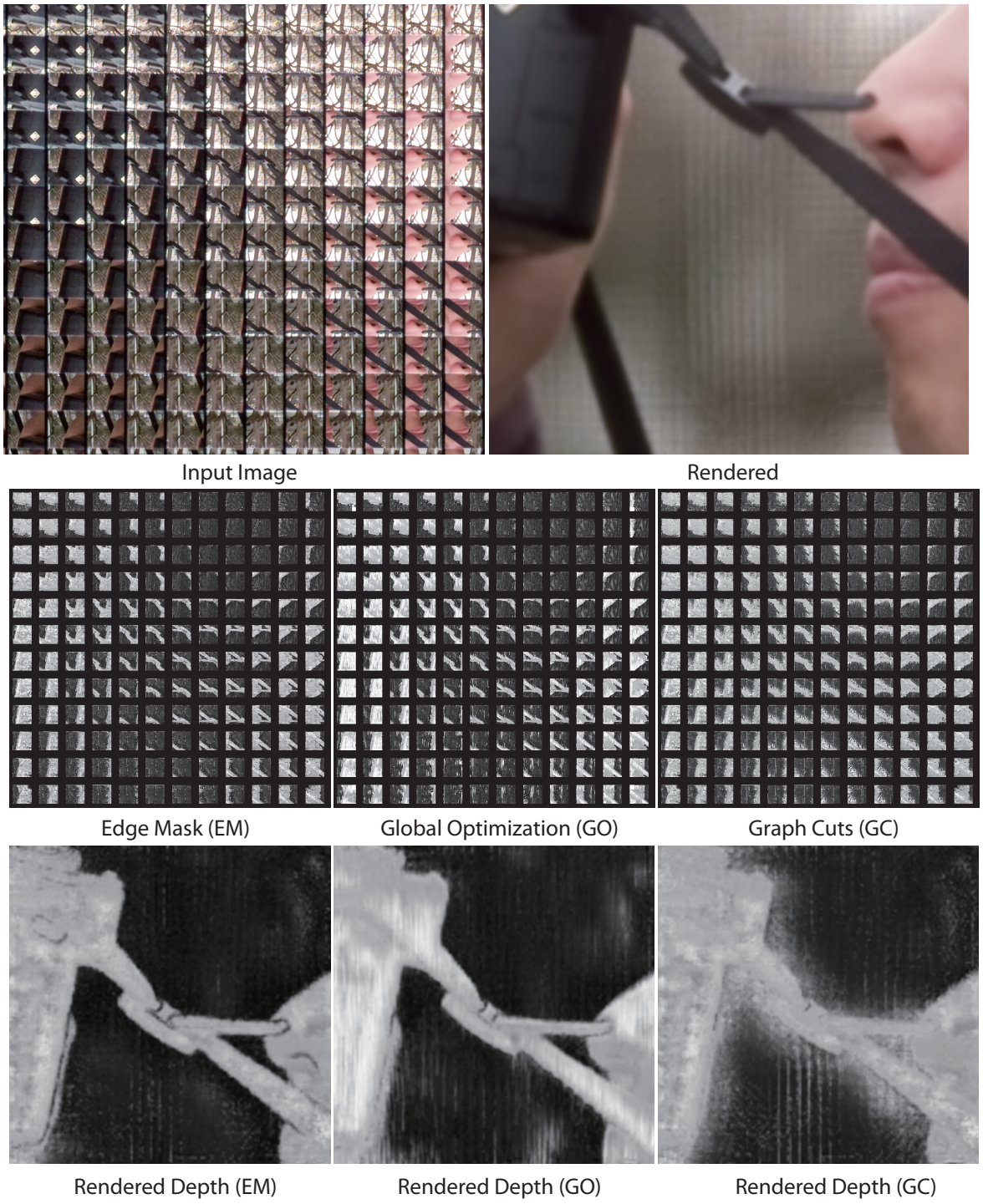

Figure 3. Estimated depth map using different methods based on the input integral image of the camera scene.

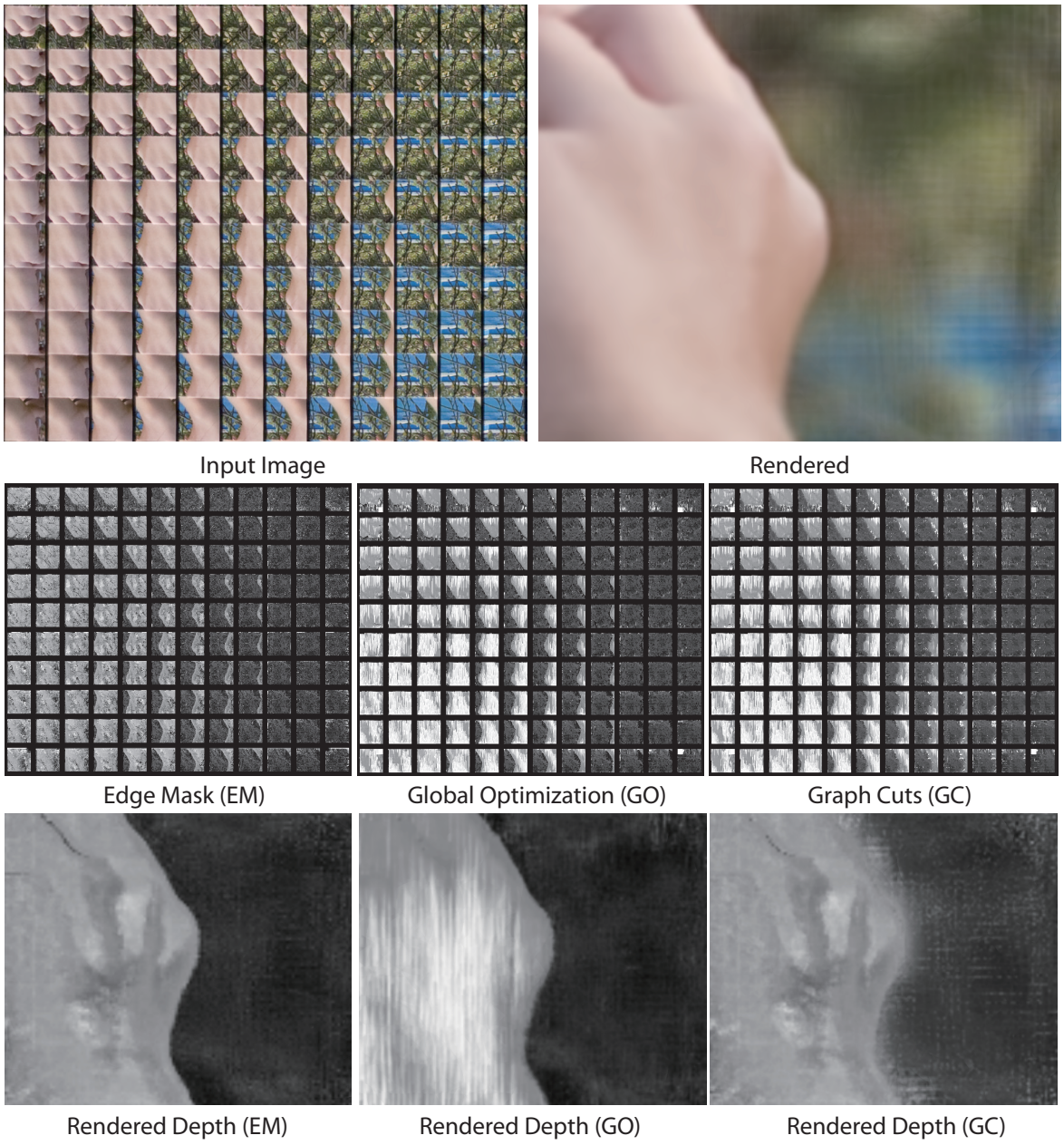

Figure 4. Estimated depth map using different methods based on the input integral image of the hand scene.

5. EXPERIMENTS

All experiments were conducted on a PC with Intel Core i7 3.2GHz CPU and 8GB memory.

The second row of Fig. 3 and Fig. 4 show the depth maps of the captured radiances of a camera scene and a hand scene using our method with edge mask (EM), global optimization (GO) and brute force graph cuts (GC). On the third row, we render the depth map using the plenoptic rendering. GC has very noisy occlusion boundaries such as edges of the belt in Fig. 3 and the edges of the hand in Fig. 4 due to the the severe occlusion conditions. In contrast, GO and GC both accurately recover fine details and robustly handle the occlusion boundaries. However, the result of EM appears a little bit more variant on surfaces with constant depth but GC better preserves the smoothness of surfaces.

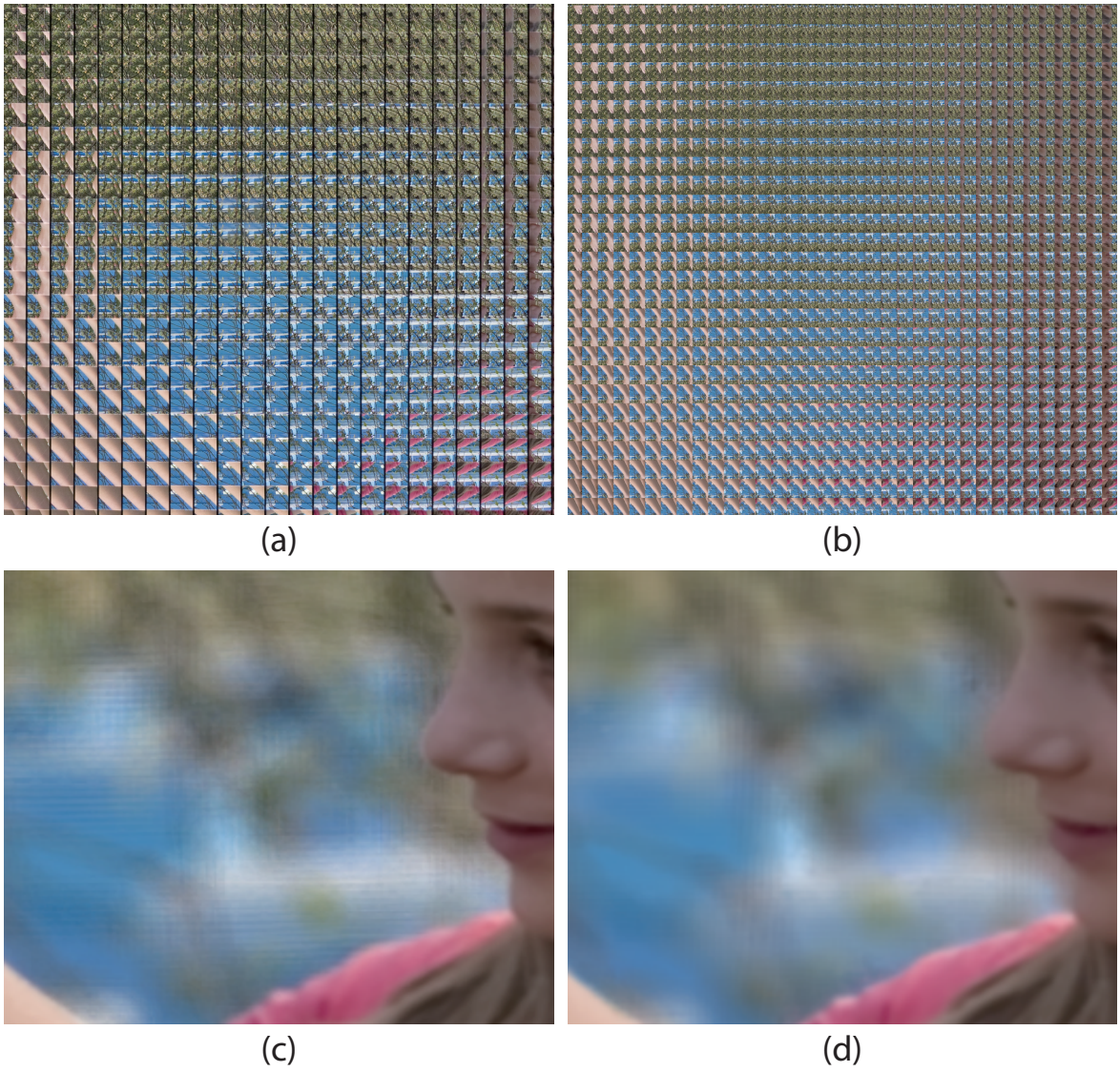

Figure 5. Different applications using the estimated depth. (a) Input views (captured integral image). (b) Synthesized views. (c) Rendering using input views. (d) Rendering using synthesized views.

5.1 Synthesizing novel views

Figure 5 (b) shows our result of using depth estimated from the input integral image to synthesize arbitrary views representing a new, denser integral image with more views. Given the input image of 25×25 views of a girl scene, we synthesize a new integral image with 25×25 views that are concentrated in the central area. With the correctly estimated occlusion boundaries, we are able to faithfully recover the edges of the arm, wrinkles on the shirt on the foreground and thin branches and leaves of the trees, cover of the bee hives in the background. Note that our boundaries sometimes appears to be noisy. This is because of our algorithm assigns a single depth value for each pixel and is not capable of handling translucent pixels on the edges. We will discuss this issue in Section 6.

5.2 Rendering aliasing reduced images

Aliasing in the rendered image is usually caused by under-sampling of the radiance. To conduct anti-aliasing, we use our estimated depth for the radiance to synthesize a densely sampled radiance of 100×100 views. We then render the dynamic depth of field effect using the new radiance. As shown in Figure 5 (d), compared with the result using the original captured radiance, when focusing on the foreground, we are able to greatly reduce the aliasing artifacts on the background and simulating a D-SLR quality image.

6. DISCUSSIONS AND LIMITATIONS

It is known that boundary pixels require matting to resolve the translucency. Since our algorithm explicitly defines one depth for each pixel, the depth for the translucent pixels could not be correctly computed. As shown in Figure 6. It is our immediate future work to explore a model of multiple depths per pixel in our algorithms. In our edge map algorithm, the threshold for the edge map is empirically defined. In the future, we plan to

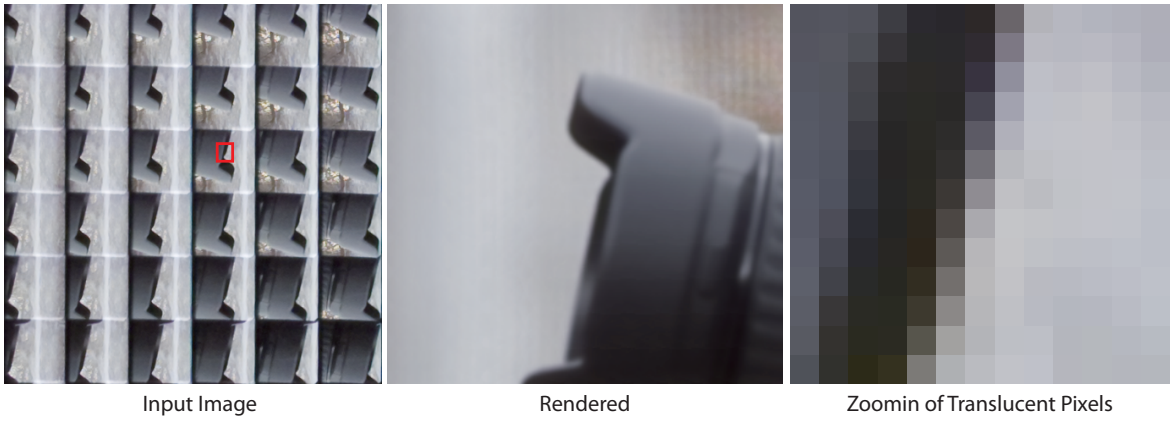

Figure 6. Translucent pixels appear near occlusion boundaries in the captured image.

analyze the statistics of the image and automatically choose the thresholds.

REFERENCES

[1] Ng, R., Levoy, M., Brdif, M., Duval, G., Horowitz, M., and Hanrahan, P., "Light field photography with a hand-held plenoptic camera," *Stanford University Computer Science Tech Report* **2**(2005-02), 1–11.

[2] Lumsdaine, A. and Georgiev, T., "The focused plenoptic camera," in [*In Proc. IEEE ICCP*], (2009).

[3] Adelson, E. and Wang, J., "Single lens stereo with a plenoptic camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**, 99–106 (1992).

[4] Georgeiv, T., Zheng, K. C., Curless, B., Salesin, D., Nayar, S., and Intwala, C., "Spatio-angular resolution tradeoff in integral photography," in [*Proceedings of Eurographics Symposium on Rendering*], 263–272 (2006).

[5] Lytro, "www.lytro.com."

[6] Raytrix, "www.raytrix.com."
[7] Lippmann, G., "La photographie intgrale," *Comptes-Rendus, Acadmie des Sciences* **146**, 446–451 (1908).
[8] Ives, F., "Parallax stereogram and process of making same, patent us 725567," (1903).
[9] Gortler, S., Grzeszczuk, R., Szeliski, R., and Cohen, M., "The lumigraph," in [*Proceedings of ACM SIGGRAPH*], 43–54 (1996).
[10] Isaksen, A., McMillan, L., and Gortler, S., "Dynamically reparameterized light fields," in [*Proceedings of ACM SIGGRAPH*], 297–306 (2000).
[11] University, S., "Stanford light field."
[12] Ng, R., Levoy, M., Brdif, M., Duval, G., Horowitz, M., and Hanrahan, P., "Light field photography with a hand-held plenoptic camera," *Stanford University Computer Science Tech Report* **2**(2005-02), 1–11.
[13] Lumsdaine, A. and Georgiev, T., "The focused plenoptic camera," in [*In Proc. IEEE ICCP*], 1–8 (2009).
[14] Kolmogorov, V. and Zabih, R., "Multi-camera scene reconstruction via graph cuts," in [*Proceedings of the ECCV*], (2002).
[15] Georgiev, T. and Lumsdaine, A., "Focused plenoptic camera and rendering," *Journal of Electronic Imaging* **19** (2010).
[16] S. Wanner, B. G., "Globally consistent depth labeling of 4d light fields," in [*Proceedings of IEEE CVPR*], (2012).
[17] Yu, J., McMillan, L., and Gortler, S., "Scam light field rendering," in [*IN: PACIFIC GRAPHICS*], (2002).
[18] Kolmogorov, V. and Zabih, R., "Computing visual correspondence with occlusions using graph cuts," in [*Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*], **2** (2001).