Depth Estimation with Occlusion Modeling Using Light-field Cameras

Ting-Chun Wang, Alexei A. Efros, Ravi Ramamoorthi, *Senior Member, IEEE*

Abstract—Light-field cameras have become widely available in both consumer and industrial applications. However, most previous approaches do not model occlusions explicitly, and therefore fail to capture sharp object boundaries. A common assumption is that for a Lambertian scene, a pixel will exhibit photo-consistency, which means all viewpoints converge to a single point when focused to its depth. However, in the presence of occlusions this assumption fails to hold, making most current approaches unreliable precisely where accurate depth information is most important – at depth discontinuities.

In this paper, an occlusion-aware depth estimation algorithm is developed; the method also enables identification of occlusion edges, which may be useful in other applications. It can be shown that although photo-consistency is not preserved for pixels at occlusions, it still holds in approximately half the viewpoints. Moreover, the line separating the two view regions (occluded object vs. occluder) has the same orientation as that of the occlusion edge in the spatial domain. By ensuring photo-consistency in only the occluded view region, depth estimation can be improved. Occlusion predictions can also be computed and used for regularization. Experimental results show that our method outperforms current state-of-the-art light-field depth estimation algorithms, especially near occlusion boundaries.

Index Terms—Light-fields, 3D reconstruction, occlusion detection

1 Introduction

Light-field cameras from Lytro [3] and Raytrix [20] are now available for consumer and industrial use respectively, bringing to fruition early work on light field rendering [10], [16]. An important benefit of light field cameras for computer vision is that multiple viewpoints or sub-apertures are available in a single light-field image, enabling passive depth estimation [4]. Indeed, Lytro Illum and Raytrix software produces depth maps used for tasks like refocusing after capture, and recent work [22] shows how multiple cues like defocus and correspondence can be combined.

However, very little work has explicitly considered occlusion before. A common assumption is that, when refocused to the correct depth (i.e., the depth of the center view), angular pixels corresponding to a single spatial pixel represent viewpoints that converge to the same point in the scene. If we collect these pixels to form an *angular patch* (Eq. 6), they exhibit photo-consistency for Lambertian surfaces, which means they all share the same color (Fig. 2a). However, this assumption is not true when occlusion occurs at a pixel; photo-consistency no longer holds since some viewpoints will now be blocked by the occluder (Fig. 2b). Enforcing photo-consistency on these pixels will often lead to incorrect depth results, causing smooth transitions around sharp occlusion boundaries.

In this paper, we explicitly model occlusions, by developing a modified version of the photo-consistency condition on angular pixels. Our main contributions are:

1) An occlusion prediction framework on light field images that uses a modified angular photo-consistency.
2) A robust depth estimation algorithm which explicitly takes occlusions into account.

We show (Sec. 3) that around occlusion edges, the angular patch can be divided into two regions, where only one of them obeys photo-consistency. A key insight (Fig. 3) is that the line separating the two regions in the *angular domain* (correct depth vs. occluder) has the same orientation as the occlusion edge does in the *spatial domain*. This observation is specific to light-fields, which have a dense set of views from a planar camera array or set of sub-apertures. Although a stereo camera also satisfies the model, the sampling in angular domain is not sufficient to observe an orientation of the occlusion boundary.

We use the modified photo-consistency condition, and the means/variances in the two regions, to estimate initial occlusion-aware depth (Sec. 4). We also compute a predictor for the occlusion boundaries, that can be used as an input to determine the final regularized depth (Sec. 5). These occlusion boundaries could also be used for other applications like segmentation or recognition. As seen in Fig. 1, our depth estimates are more accurate in scenes with complex occlusions (previous results smooth object boundaries like the holes in the basket). In Sec. 6, we present extensive results on both synthetic data (Figs. 9, 10), and on real scenes captured with the consumer Lytro Illum camera (Fig. 11), demonstrating higher-quality depth recovery than previous work [8], [22], [26], [30].

2 Related Work

(Multi-View) Stereo with Occlusions: Multi-view stereo matching has a long history, with some efforts to handle

- *T.-C. Wang and A. A. Efros are with the EECS Department at the University of California, Berkeley, in Berkeley, CA 94720. Email: {tcwang0509,efros}@eecs.berkeley.edu.*
- *R. Ramamoorthi is with the CSE Department at the University of California, San Diego, at La Jolla, CA 92093. E-mail: ravir@cs.ucsd.edu.*

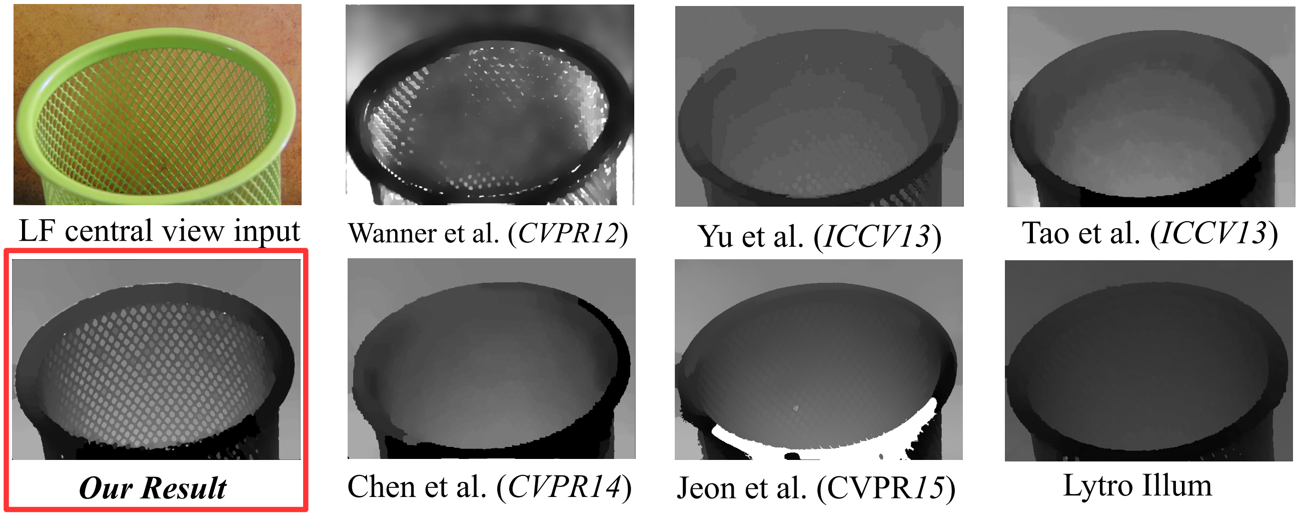
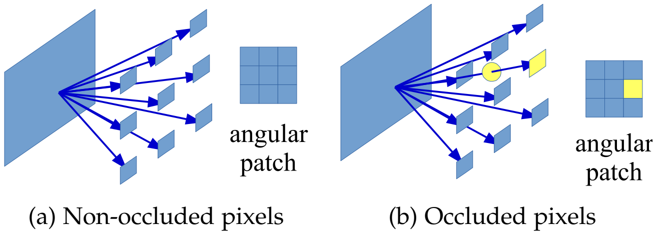


Fig. 1: *Comparison of depth estimation results of different algorithms from a light field input image. Darker represents closer and lighter represents farther. It can be seen that only our occlusion-aware algorithm successfully captures most of the holes in the basket, while other methods either smooth over them, or have artifacts as a result.*



(a) Non-occluded pixels (b) Occluded pixels

Fig. 2: *Non-occluded vs. occluded pixels. (a) At non-occluded pixels, all view rays converge to the same point in the scene if refocused to the correct depth. (b) However, photo-consistency fails to hold at occluded pixels, where some view rays will hit the occluder.*

occlusions. For example, the graph-cut framework [13] used an occlusion term to ensure visibility constraints while assigning depth labels. Woodford et al. [29] imposed an additional second order smoothness term in the optimization, and solved it using Quadratic Pseudo-Boolean Optimization [21]. Based on this, Bleyer et al. [5] assumed a scene is composed of a number of smooth surfaces and proposed a soft segmentation method to apply the asymmetric occlusion model [28]. However, significant occlusions still remain difficult to address even with a large number of views.

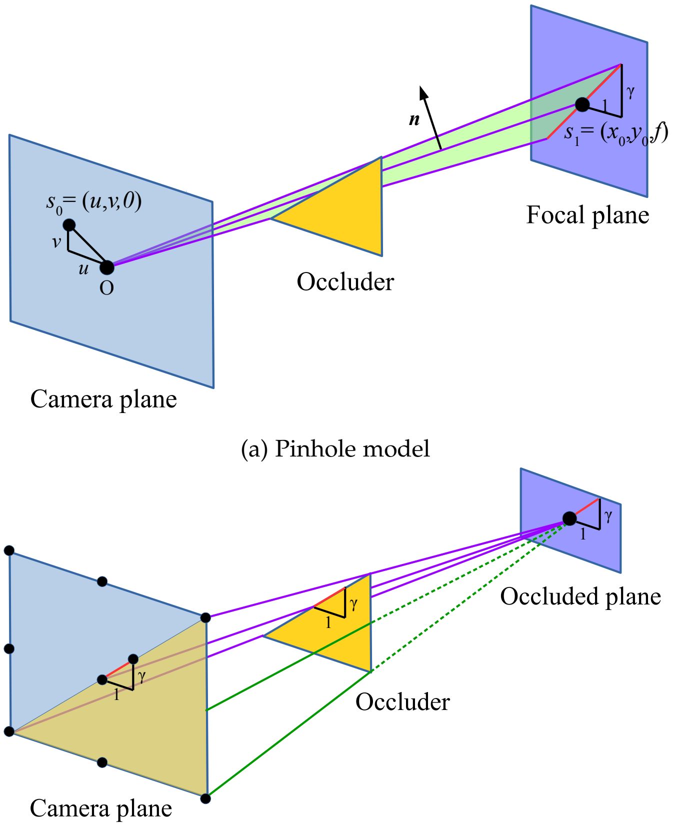
Depth from Light Field Cameras: Perwass and Wietzke [20] proposed using correspondence techniques to estimate depth from light-field cameras. Tao et al. [22] combined correspondence and defocus cues in the 4D Epipolar Image (EPI) to complement the disadvantages of each other. Neither method explicitly models occlusions. McCloskey [18] proposed a method to remove partial occlusion in color images, which does not estimate depth. Wanner and Goldluecke [26] proposed a globally consistent framework by applying structure tensors to estimate the directions of feature pixels in the 2D EPI. Yu et al. [30] explored geometric structures of 3D lines in ray space and encoded the line constraints to further improve the reconstruction quality. However, both methods are vulnerable to heavy occlusion: the tensor field becomes too random to estimate, and 3D lines are partitioned into small, incoherent segments. Kim et al. [12] adopted a fine-to-coarse framework to ensure smooth reconstructions in homogeneous areas using dense light fields. Jeon et al. [11] proposed a phase-based interpolation method to increase the accuracy for sub-pixel shift. We build on the method by Tao et al. [22], which works with consumer light field cameras, to improve depth estimation by taking occlusions into account. Although Tao et al. have a more recent method for depth estimation [23], it aims at combining the shading cue which is not applicable in our case. Compared to the work by Wang et al. [25], we added the comparisons to Jeon et al. [11] and results by Lytro Illum, which make the validation more complete.

Chen et al. [8] proposed a new bilateral metric on angular pixel patches to measure the probability of occlusions by their similarity to the central pixel. However, as noted in their discussion, their method is biased towards the central view as it uses the color of the central pixel as the mean of the bilateral filter. Therefore, the bilateral metric becomes unreliable once the input images get noisy. In contrast, our method uses the mean of about half the pixels as the reference, and is thus more robust when the input images are noisy, as shown in our result section.

3 Light-Field Occlusion Theory

We first develop our new light-field occlusion model, based on the physical image formation. We show that at occlusions, some of the angular patch remains photo-consistent, while the other part comes from occluders and exhibits no photo consistency. By treating these two regions separately, occlusions can be better handled.

For each pixel on an occlusion edge, we assume it is occluded by only one occluder among all views. We also assume that we are looking at a spatial patch small enough, so that the occlusion edge around that pixel can



(a) Pinhole model

(b) “Reversed” pinhole model

Fig. 3: *Light field occlusion model. (a) Pinhole model for central camera image formation. An occlusion edge on the imaging plane corresponds to an occluding plane in the 3D space. (b) The “reversed” pinhole model for light field formation. It can be seen that when we refocus to the occluded plane, we get a projection of the occluder on the camera plane, forming a reversed pinhole camera model.*

be approximated by a line. We show that if we refocus to the occluded plane, the angular patch will still have photo-consistency in a subset of the pixels (unoccluded). Moreover, the edge separating the unoccluded and occluded pixels in the angular patch has the same orientation as the occlusion edge in the spatial domain (Fig. 3). In Secs. 4 and 5, we use this idea to develop a depth estimation and regularization algorithm.

Consider a pixel at (x_0, y_0, f) on the imaging focal plane (the plane in focus), as shown in Fig. 3a. An edge in the central pinhole image with 2D slope γ corresponds to a plane P in 3D space (the green plane in Fig. 3a). The normal $\mathbf{n}$ to this plane can be obtained by taking the cross-product,

$$\mathbf{n} = (x_0, y_0, f) \times (x_0 + 1, y_0 + \gamma, f) = (-\gamma f, f, \gamma x_0 - y_0). \quad (1)$$

Note that we do not need to normalize the vector. The plane equation is $P(x, y, z) \equiv \mathbf{n} \cdot (x_0 - x, y_0 - y, f - z) = 0$,

$$P(x, y, z) \equiv \gamma f(x - x_0) - f(y - y_0) + (y_0 - \gamma x_0)(z - f) = 0. \quad (2)$$

In our case, one can verify that $\mathbf{n} \cdot (x_0, y_0, f) = 0$ so a further simplification to $\mathbf{n} \cdot (x, y, z) = 0$ is possible,

$$P(x, y, z) \equiv \gamma f x - f y + (y_0 - \gamma x_0) z = 0. \quad (3)$$

Now consider the occluder (yellow triangle in Fig. 3a). The occluder intersects $P(x, y, z)$ with $z \in (0, f)$ and lies on one side of that plane. Without loss of generality, we can assume it lies in the half-space $P(x, y, z) \geq 0$. Now consider a point $(u, v, 0)$ on the camera plane (the plane where the camera array lies on). To avoid being shadowed by the occluder, the line segment connecting this point and the pixel (x_0, y_0, f) on the image must not hit the occluder,

$$P(\mathbf{s}_0 + (\mathbf{s}_1 - \mathbf{s}_0)t) \leq 0 \quad \forall t \in [0, 1], \quad (4)$$

where $\mathbf{s}_0 = (u, v, 0)$ and $\mathbf{s}_1 = (x_0, y_0, f)$. When $t = 1$, $P(\mathbf{s}_1) = 0$. When $t = 0$,

$$P(\mathbf{s}_0) \equiv \gamma f u - f v \leq 0. \quad (5)$$

The last inequality is satisfied if $v \geq \gamma u$, i.e., the *critical slope on the angular patch $v/u = \gamma$ is the same as the edge orientation in the spatial domain*. If the inequality above is satisfied, both endpoints of the line segment lie on the other side of the plane, and hence the entire segment lies on that side as well. Thus, the light ray will not be occluded.

We also give an intuitive explanation of the above proof. Consider a plane being occluded by an occluder, as shown in Fig. 3b. Consider a simple 3×3 camera array. When we refocus to the occluded plane, we can see that some views are occluded by the occluder. Moreover, the occluded cameras on the camera plane are the projection of the occluder on the camera plane. Thus, we obtain a “reversed” pinhole camera model, where the original imaging plane is replaced by the camera plane, and the original pinhole becomes the pixel we are looking at. When we collect pixels from different cameras to form an angular patch, the edge separating the two regions will correspond to the same edge the occluder has in the spatial domain.

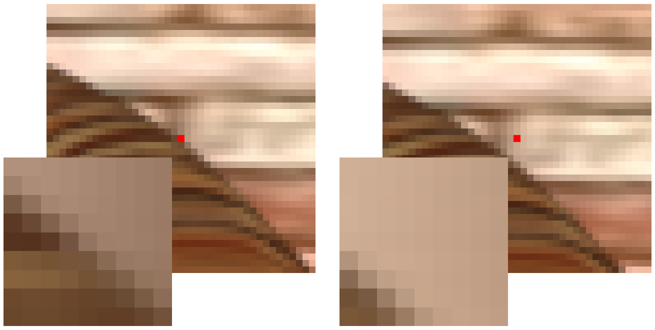
Therefore, we can predict the edge orientation in the angular domain using the edge in the spatial image. Once we divide the patch into two regions, we know photo consistency holds in one of them since they all come from the same (assumed to be Lambertian) spatial pixel.

4 Occlusion-Aware Initial Depth

In this section, we show how to modify the initial depth estimation from Tao et al. [22], based on the theory above. First, we apply edge detection on the central view image. Then for each edge pixel, we compute initial depths using a modified photo-consistency constraint. The next section will discuss computation of refined occlusion predictors and regularization to generate the final depth map.

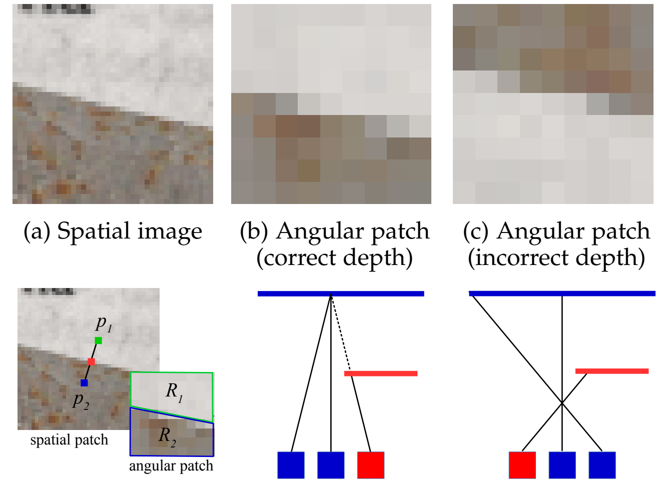
4.1 Edge detection

We first apply Canny edge detection on the central view (pinhole) image. Then an edge orientation predictor is applied on the obtained edges to get the orientation angles at each edge pixel. These pixels are candidate occlusion pixels in the central view. However, some pixels are not occluded in the central view, but are occluded in other views, as shown in Fig. 4, and we want to mark these as candidate occlusions as well. We identify these pixels by dilating the edges detected in the center view.



(a) Occlusion in central view (b) Occlusion in other views

Fig. 4: *Occlusions in different views. The insets are the angular patches of the red pixels when refocused to the correct depth. At the occlusion edge in the central view, the angular patch can be divided evenly into two regions, one with photo-consistency and one without. However, for pixels around the occlusion edge, although the central view is not occluded, some other views will still get occluded. Hence, the angular patch will not be photo-consistent, and will be unevenly divided into occluded and visible regions.*



(a) Spatial image (b) Angular patch (correct depth) (c) Angular patch (incorrect depth)

(d) Color consistency (e) Focusing to correct depth (f) Focusing to incorrect depth

Fig. 5: *Color consistency constraint. (b)(e) We can see that when we refocus to the correct depth, we get low variance in half the angular patch. However, in (c)(f) although we refocused to an incorrect depth, it still gives low variance response since the occluded plane is very textureless, so we get a "reversed" angular patch. To address this, we add another constraint that p_1 and p_2 should be similar to the averages of R_1 and R_2 in (d), respectively.*

4.2 Depth Estimation

For each pixel, we refocus to various depths using a 4D shearing of the light-field data [19],

$$L_\alpha(x,y,u,v) = L(x + u(1-\frac{1}{\alpha}), y + v(1-\frac{1}{\alpha}), u, v), \quad (6)$$

where L is the input light field image, α is the ratio of the refocused depth to the currently focused depth, L_α is the refocused light field image, (x,y) are the spatial coordinates, and (u,v) are the angular coordinates. The central viewpoint is located at $(u,v) = (0,0)$. This gives us an angular patch for each depth, which can be averaged to give a refocused pixel. In our implementation, we use a simple linear interpolation to perform the resampling in Eq. 6. However, more advanced resampling techniques, e.g. the phase-based method in [11], could be used and could potentially lead to better results.

When an occlusion is not present at the pixel, the obtained angular patch will have photo-consistency, and hence exhibits small variance and high similarity to the central view. For pixels that are not occlusion candidates, we can simply compute the variance and the mean of this patch to obtain the correspondence and defocus cues, similar to the method by Tao et al. [22].

However, if an occlusion occurs, photo-consistency will no longer hold. Instead of dealing with the entire angular patch, we divide the patch into two regions. The angular edge orientation separating the two regions is the same as in the spatial domain, as proven in Sec. 3. Since at least half the angular pixels come from the occluded plane (otherwise it will not be seen in the central view), we place the edge passing through the central pixel, dividing the patch evenly. Note that only one region, corresponding to the partially occluded plane focused to the correct depth, exhibits photo-consistency. The other region contains angular pixels that come from the occluder, which is not focused at the proper depth, and might also contain some pixels from the occluded plane. We therefore replace the original patch with the region that has the minimum variance to compute the correspondence and defocus cues.

To be specific, let (u_1, v_1) and (u_2, v_2) be the angular coordinates in the two regions, respectively. We first compute the means and the variances of the two regions,

$$\bar{L}_{\alpha,j}(x,y) = \frac{1}{N_j}\sum_{u_j,v_j} L_\alpha(x,y,u_j,v_j), \quad j = 1,2 \quad (7)$$

$$V_{\alpha,j}(x,y) = \frac{1}{N_j - 1}\sum_{u_j,v_j}\left(L_\alpha(x,y,u_j,v_j) - \bar{L}_{\alpha,j}(x,y)\right)^2, \quad (8)$$

where N_j is the number of pixels in region j. Let

$$i = \arg\min_{j=1,2}\left\{V_{\alpha,j}(x,y)\right\} \quad (9)$$

be the index of the region that exhibits smaller variance. Then the correspondence response is given by

$$C_\alpha(x,y) = V_{\alpha,i}(x,y) \quad (10)$$

Similarly, the defocus response is given by

$$D_\alpha(x,y) = \left(\bar{L}_{\alpha,i}(x,y) - L(x,y,0,0)\right)^2 \quad (11)$$

Finally, the optimal depth is determined as

$$\alpha^*(x,y) = \arg\min_\alpha\left\{C_\alpha(x,y) + D_\alpha(x,y)\right\} \quad (12)$$

4.3 Color Consistency Constraint

When we divide the angular patch into two regions, it is sometimes possible to obtain a "reversed" patch when we refocus to an incorrect depth, as shown in Fig. 5. If the

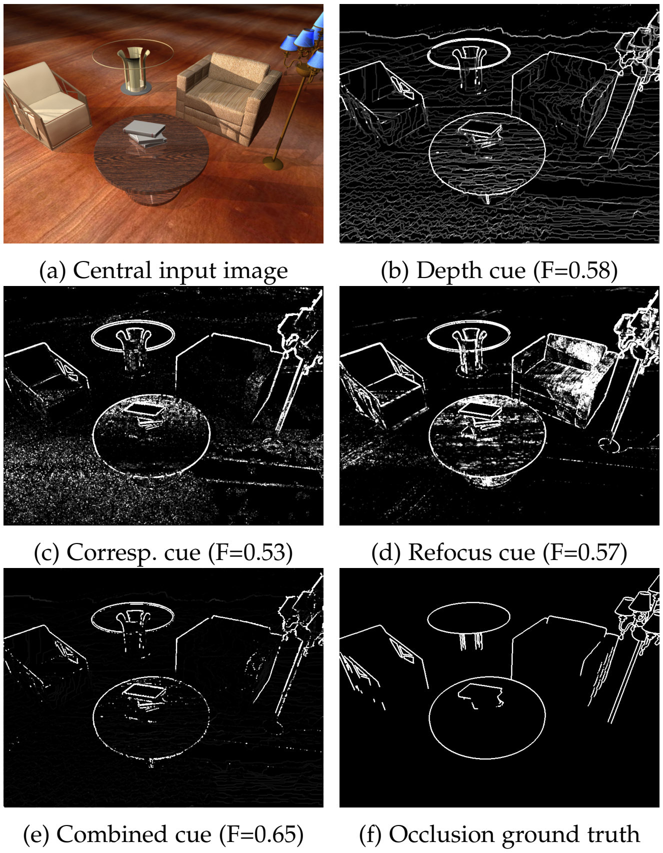

(a) Central input image
(b) Depth cue (F=0.58)
(c) Corresp. cue (F=0.53)
(d) Refocus cue (F=0.57)
(e) Combined cue (F=0.65)
(f) Occlusion ground truth

Fig. 6: *Occlusion Predictor (Synthetic Scene). The intensities are adjusted for better contrast. F-measure is the harmonic mean of precision and recall compared to the ground truth. By combining three cues from depth, correspondence and refocus, we can obtain a better prediction of occlusions.*

occluded plane is very textureless, this depth might also give a very low variance response, even though it is obviously incorrect. To address this, we add a color consistency constraint that the averages of the two regions should have a similar relationship with respect to the current pixel as they have in the spatial domain. Mathematically,

$$|\bar{L}_{\alpha,1}-p_1|+|\bar{L}_{\alpha,2}-p_2| < |\bar{L}_{\alpha,2}-p_1|+|\bar{L}_{\alpha,1}-p_2|+\delta, \quad (13)$$

where p_1 and p_2 are the values of the pixels shown in Fig. 5d, and δ is a small value (threshold) to increase robustness. If refocusing to a depth violates this constraint, this depth is considered invalid, and is automatically excluded in the depth estimation process.

5 Occlusion-Aware Depth Regularization

After the initial local depth estimation phase, we refine the results with global regularization using a smoothness term. We improve on previous methods by reducing the effect of the smoothness/regularization term in occlusion regions. Our occlusion predictor, discussed below, may also be useful independently for other vision applications.

5.1 Occlusion Predictor Computation

We compute a predictor P_{occ} for whether a particular pixel is occluded, by combining cues from depth, correspondence and refocus.

5.1.1 Depth Cues

First, by taking the gradient of the initial depth, we can obtain an initial occlusion boundary,

$$P^d_{\text{occ}} = f\left(\nabla d_{\text{ini}}/d_{\text{ini}}\right) \quad (14)$$

where d_{ini} is the initial depth, and $f(\cdot)$ is a robust clipping function that saturates the response above some threshold. We divide the gradient by d_{ini} to increase robustness since for the same normal, the depth change across pixels becomes larger as the depth gets larger.

5.1.2 Correspondence Cues

In occlusion regions, we have already seen that photo-consistency will only be valid in approximately half the angular patch, with a small variance in that region. On the other hand, the pixels in the other region come from different points on the occluding object, and thus exhibit much higher variance. By computing the ratio between the two variances, we can obtain an estimate of how likely the current pixel is to be at an occlusion,

$$P^{\text{var}}_{\text{occ}} = f\left(\max\left\{\frac{V_{\alpha^*,1}}{V_{\alpha^*,2}}, \frac{V_{\alpha^*,2}}{V_{\alpha^*,1}}\right\}\right), \quad (15)$$

where α^* is the initial depth we get.

5.1.3 Refocus Cues

Finally, note that the variances in both the regions will be small if the occluder is textureless. To address this issue, we also compute the means of both regions. Since the two regions come from different objects, their colors should be different, so a large difference between the two means also indicates a possible occlusion occurrence. In other words,

$$P^{\text{avg}}_{\text{occ}} = f(|\bar{L}_{\alpha^*,1} - \bar{L}_{\alpha^*,2}|) \quad (16)$$

Finally, we compute the combined occlusion response or prediction by the product of these three cues,

$$P_{occ} = \mathcal{N}(P^{\text{d}}_{\text{occ}}) \cdot \mathcal{N}(P^{\text{var}}_{\text{occ}}) \cdot \mathcal{N}(P^{\text{avg}}_{\text{occ}}) \quad (17)$$

where $\mathcal{N}(\cdot)$ is a normalization function that subtracts the mean and divides by the standard deviation. The threshold values of the f function for depth, correspondence and refocus cues are set to 1, 100, and 0.01, respectively.

5.2 Depth Regularization

Finally, given initial depth and occlusion cues, we regularize with a Markov Random Field (MRF) for a final depth map. We minimize the energy:

$$E = \sum_p E_{\text{unary}}(p, d(p)) + \lambda \sum_{p,q} E_{\text{binary}}(p, q, d(p), d(q)). \quad (18)$$

where d is the final depth p, q are neighboring pixels, and λ is a weight which we set to 5. We adopt the unary term similar to Tao et al. [22]. The binary energy term is defined as

$$E_{binary}(p,q,d(p),d(q)) = \frac{\exp\left[-(d(p)-d(q))^2/(2\sigma^2)\right]}{(|\nabla I(p)-\nabla I(q)|+k|P_{occ}(p)-P_{occ}(q)|)} \tag{19}$$

where ∇I is the gradient of the central pinhole image, and k is a weighting factor. The numerator encodes the smoothness constraint, while the denominator reduces the strength of the constraint if two pixels are very different or an occlusion is likely to be between them. The minimization is solved using a standard graph cut algorithm [6], [7], [14]. We can then apply the occlusion prediction procedure again on this regularized depth map. A sample result is shown in Fig. 6. In this example, the F-measure (harmonic mean of precision and recall compared to ground truth) increased from 0.58 (depth cue), 0.53 (correspondence cue), and 0.57 (refocus cue), to 0.65 (combined cue).

6 Results

In this section, we first show results of different stages of our algorithm (Sec. 6.1), and then demonstrate the superiority of our method by comparing to different state-of-the-art algorithms (Sec. 6.2). Finally, we show limitations and some failure cases of our method (Sec. 6.3).

6.1 Algorithm Stages

We show results of different stages of our algorithm in Fig. 7. First, edge detection is applied on the central pinhole image (Fig. 7a) to give all possible edge boundaries (Fig. 7b). As can be seen, although the output captures the occlusion boundaries, it also contains lots of false positives. We then compute the initial depth (Fig. 7c) and occlusion prediction (Fig. 7d) using the method described in Sec. 4. Note that the false positives in the obtained occlusion are dramatically reduced. Finally, using the initial depth and occlusion detection, we further regularize the depth (Sec. 5) to get the final depth (Fig. 7e) and occlusion detection (Fig. 7f). Note that the final occlusion detection realistically captures the true occlusion boundaries. For runtime, on a 2.4 GHz Intel i7 machine with 8GB RAM, our MATLAB implementation takes about 3 minutes on a Lytro Illum Image (7728 × 5368 pixels). This is comparable to [22], since all the additional steps are marginal to the computation.

6.2 Comparisons

We compare our results to the methods by Wanner et al. [26], Tao et al. [22], Yu et al. [30], Chen et al. [8], Jeon et al. [11], and the results by Lytro Illum. For Chen et al., since code is not available, we used our own implementation. Compared to Wang et al. [25], we added the comparisons to Jeon et al. [11] and results by Lytro Illum (for real images), which make the validation more complete. Since ground truth at occlusions is difficult to obtain, we perform extensive tests using the synthetic dataset created by Wanner et al. [27] as well as new scenes modeled by us. Our dataset is generated from 3dsMax [1] using models from the Stanford Computer Graphics Laboratory [9], [15], [24] and models freely available online [2]. Upon publication of this work, the dataset will be available online. While the dataset by [27] only provides ground truth depth, ours provides ground truth depth, normals, specularity, lighting, etc, which we believe will be useful for a wider variety of applications. In addition to synthetic datasets, we also validate our algorithm on real-world scenes of fine objects with occlusions, taken by the Lytro Illum camera.

6.2.1 Occlusion Boundaries

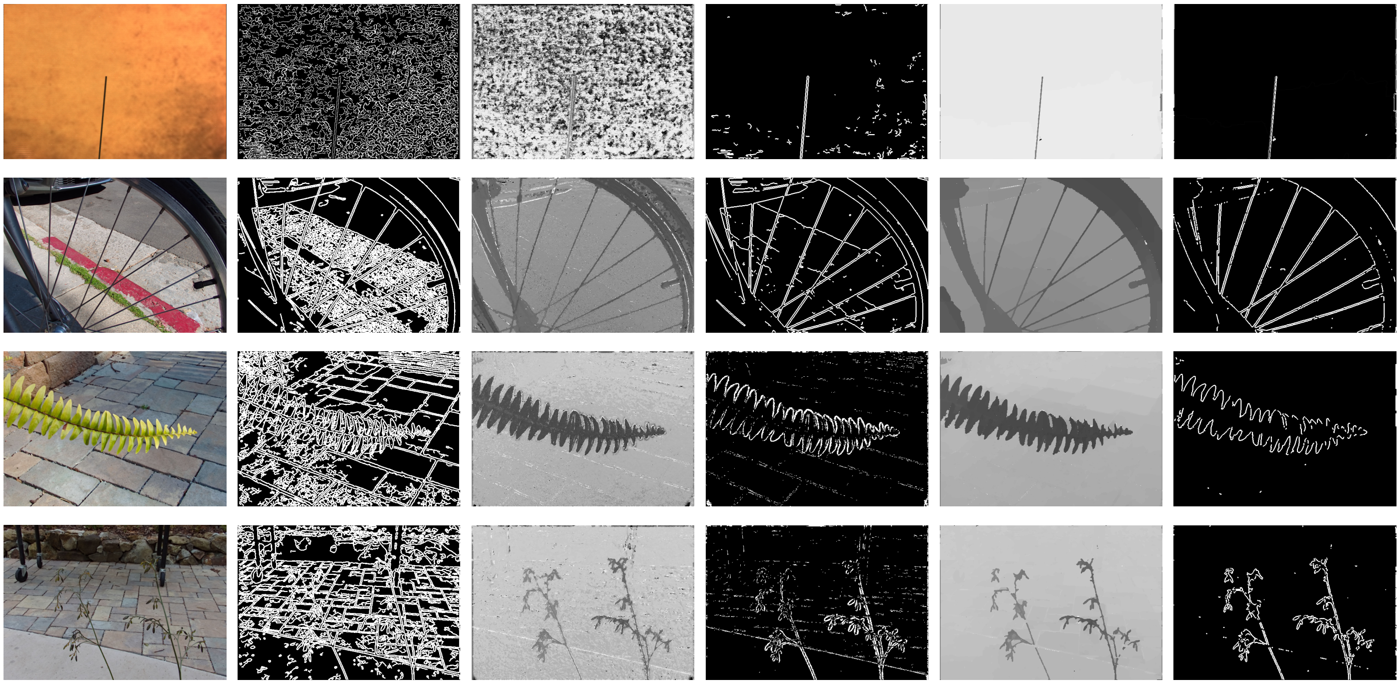
For each synthetic scene, we compute the occlusion boundaries from the depth maps generated by each algorithm, and report their precision-recall curves by picking different thresholds. For our method, the occlusions are computed using only the depth cue instead of the combined cue in Sec. 5, to compare the depth quality only. A predicted occlusion pixel is considered correct if its error is within one pixel. The results on both synthetic datasets are shown in Figs. 8a,8b. Our algorithm achieves better performance than current state-of-the-art methods. Next, we validate the robustness of our system by adding noise to a test image, and report the F-measure values of each algorithm, as shown in Fig. 8c. Although Chen et al. [8] performs very well in the absence of noise, their quality quickly degrades as the noise level is increased. In contrast, our algorithm is more tolerant to noise.

6.2.2 Depth Maps for Synthetic Scenes

Figure 9 shows the recovered depths on the synthetic dataset by Wanner et al. [27]. It can be seen that our results show fewer artifacts in heavily occluded areas. We obtain the correct shape of the door and window in the top row, accurate boundaries along the twig and leaf in the second row, and realistic antenna shape and wing boundaries in the bottom row. Other methods smooth the object boundaries and are noisy in some regions. Figure 10 shows the results on our synthetic dataset. Notice that we capture the boundaries of the leaves, fine structures like the lamp and holes in the chair, and thin shapes of the lamp and the chandelier. Other methods smooth over these occlusions or generate thicker structures. The RMSE of the depth maps compared to the ground truth are also shown in Table 1. However, note that RMSE is not the best metric for the improvements on thin occluded structures provided by our method.

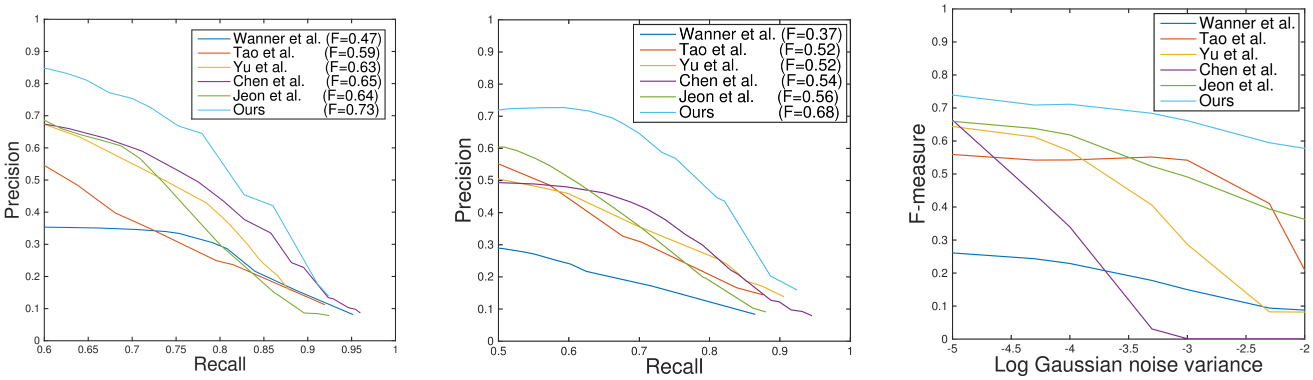
6.2.3 Depth Maps for Real Scenes

Figures 1 and 11 compare results on real scenes with fine structures and occlusions, captured with Lytro Illum light field camera. Our method performs better around occlusion boundaries, especially for thin objects. Ours is the only method that captures the basket holes in Fig. 1. In Fig. 11, our method properly captures the thin structure of the lead (first row), reproduces the fine petals of the flower (second row), captures the holes behind the leafs without over-smoothing (third and fourth row), obtains realistic shape of the stem(fifth row), and reproduces the complicated structure of the strap (final row).



(a) Light-field input (b) Edge detection (c) Initial depth (d) Initial occlusion (e) Final depth (f) Final occlusion

Fig. 7: *Real-world results of different stages of our algorithm. We first apply edge detection on the central input, run our depth estimation algorithm on the light-field image to get an initial depth and an occlusion response prediction, and finally use the occlusion to regularize the initial depth to get a final depth map. We can then run the occlusion predictor on this final depth again to get a refined occlusion.*

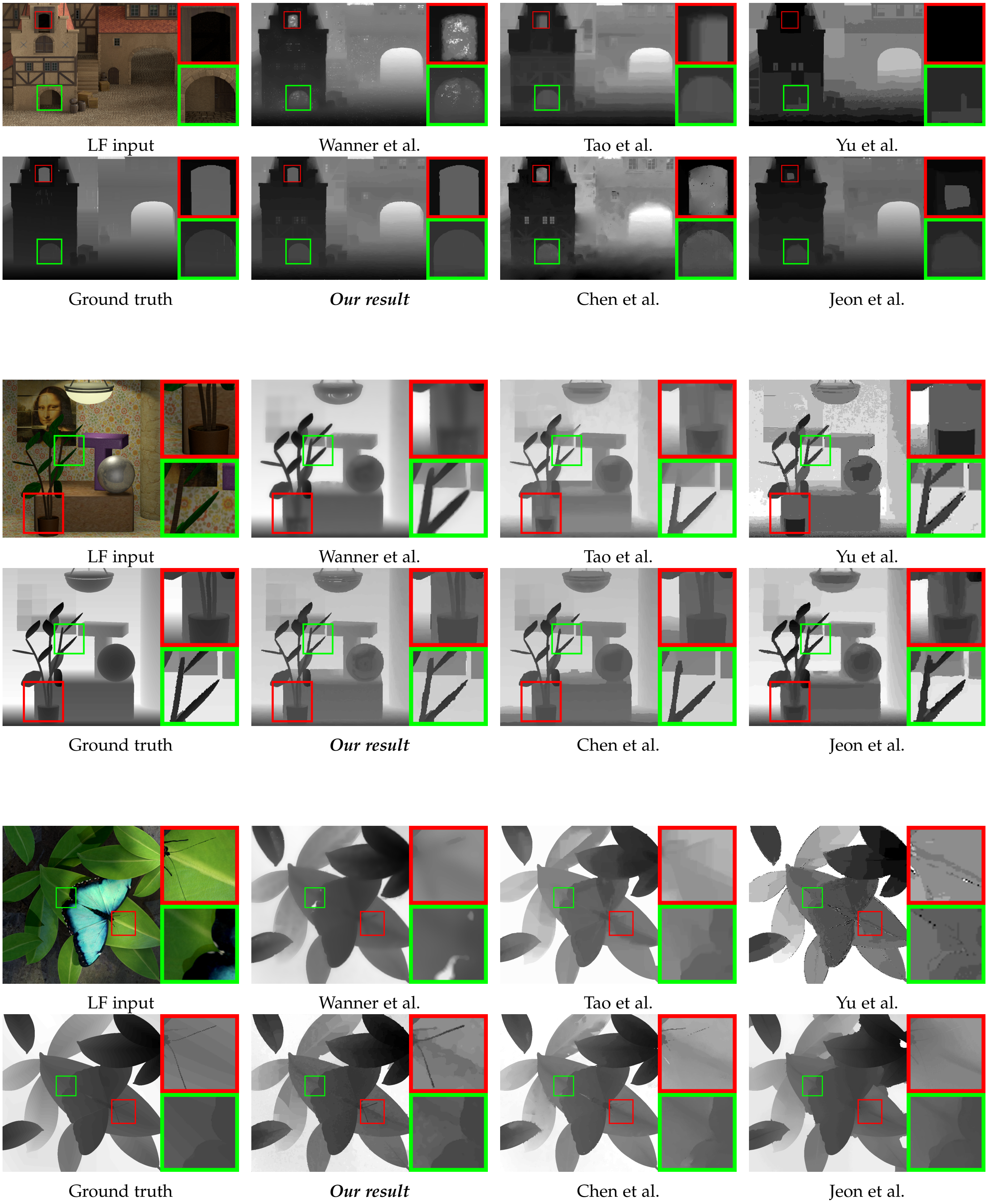


(a) PR curve (dataset by Wanner et al.) (b) PR curve (our dataset) (c) F-measure vs. noise

Fig. 8: *(a) PR-curve of occlusion boundaries on dataset of Wanner et al.* [27] *(b) PR-curve on our dataset. (c) F-measure vs. noise level. Our method achieves better results than current state-of-the-art methods, and is robust to noise.*

	Wanner et al.	Tao et al.	Yu et al.	Chen et al.	Jeon et al.	Our method
Dataset by Wanner et al.	0.0470	0.0453	0.0513	0.0375	0.0443	***0.0355***
Our dataset	0.1256	0.1253	0.1006	0.1019	0.1062	***0.0974***

TABLE 1: *Depth RMSE on synthetic scenes. Our method achieves lowest RMSE on both datasets. Note that RMSE is not the best metric for the improvements on thin occluded structures provided by our method.*



LF input

Wanner et al.

Tao et al.

Yu et al.

Ground truth

Our result

Chen et al.

Jeon et al.

LF input

Wanner et al.

Tao et al.

Yu et al.

Ground truth

Our result

Chen et al.

Jeon et al.

LF input

Wanner et al.

Tao et al.

Yu et al.

Ground truth

Our result

Chen et al.

Jeon et al.

Fig. 9: *Depth estimation results on synthetic data by Wanner et al.* [27]. *Some intensities in the insets are adjusted for better contrast. In the first example, note that our method correctly captures the shape of the door/window, while all other algorithms fail and produce smooth transitions. Similarly, in the second example our method reproduces accurate boundaries along the twig/leaf, while other algorithms generate smoothed results or fail to capture the details, and have artifacts. Finally, in the last example, our method is the only one which can capture the antennas of the butterfly, and preserve the boundary of the wings, while other methods fail or generate smoothed results.*

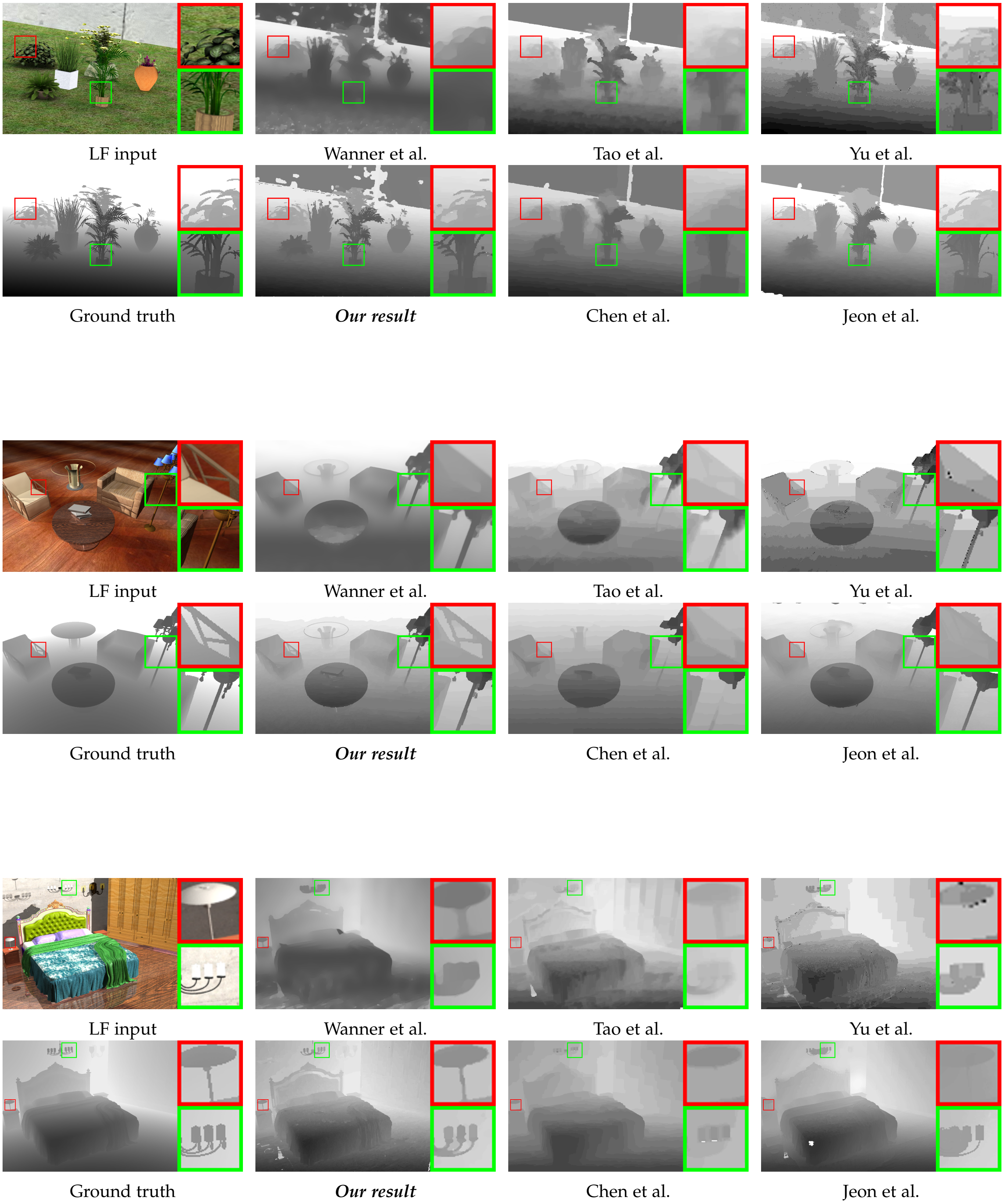


Fig. 10: *Depth estimation results on our synthetic dataset. Some intensities in the insets are adjusted for better contrast. In the first example, our method successfully captures the shapes of the leaves, while all other methods generate smoothed results. In the second example, our method captures the holes in the chair as well as the thin structure of the lamp, while other methods obtain smoothed or thicker structures. In the last example, our method captures the thin structure of the lamp and the chandelier, while other methods fail or generate thickened results.*

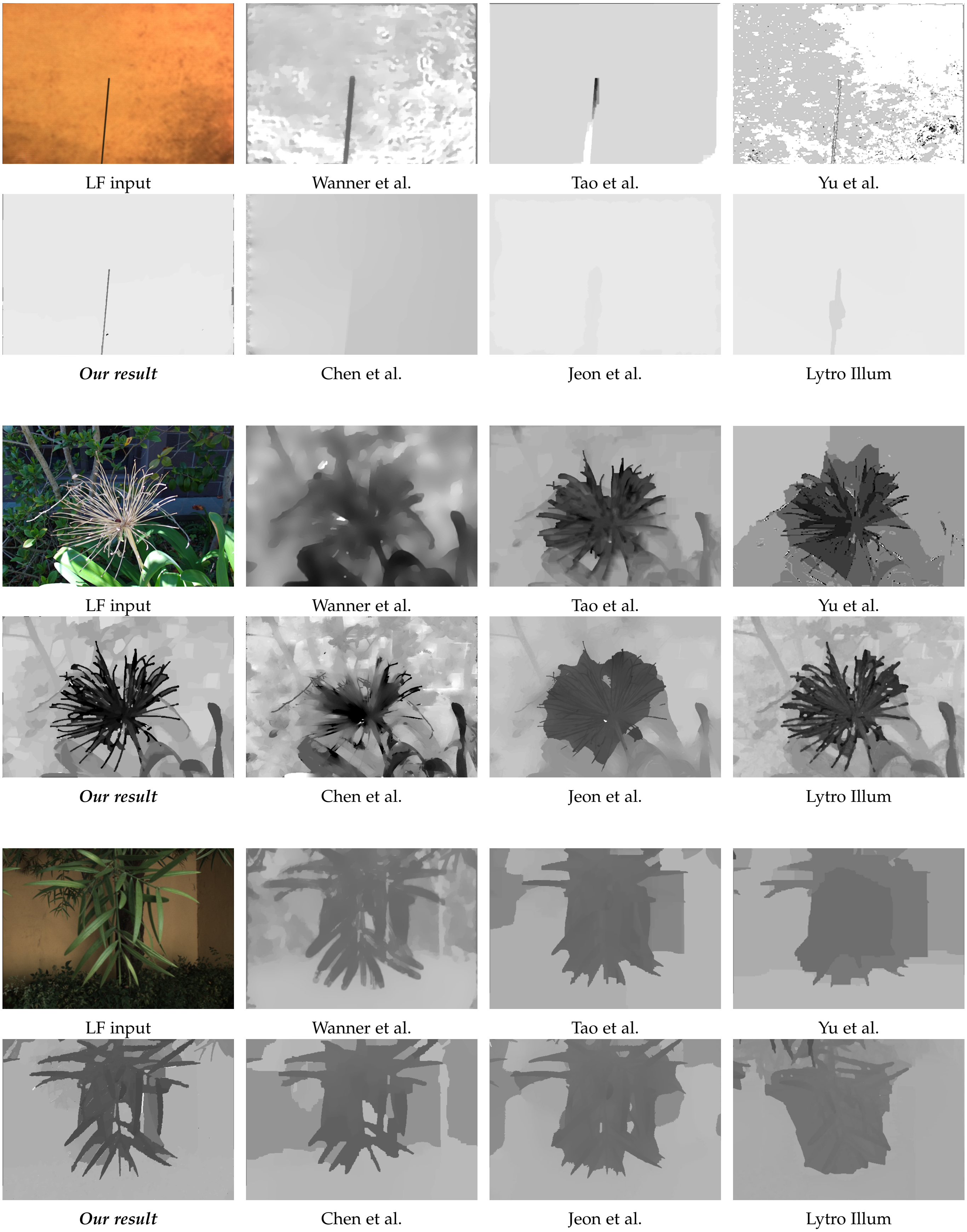
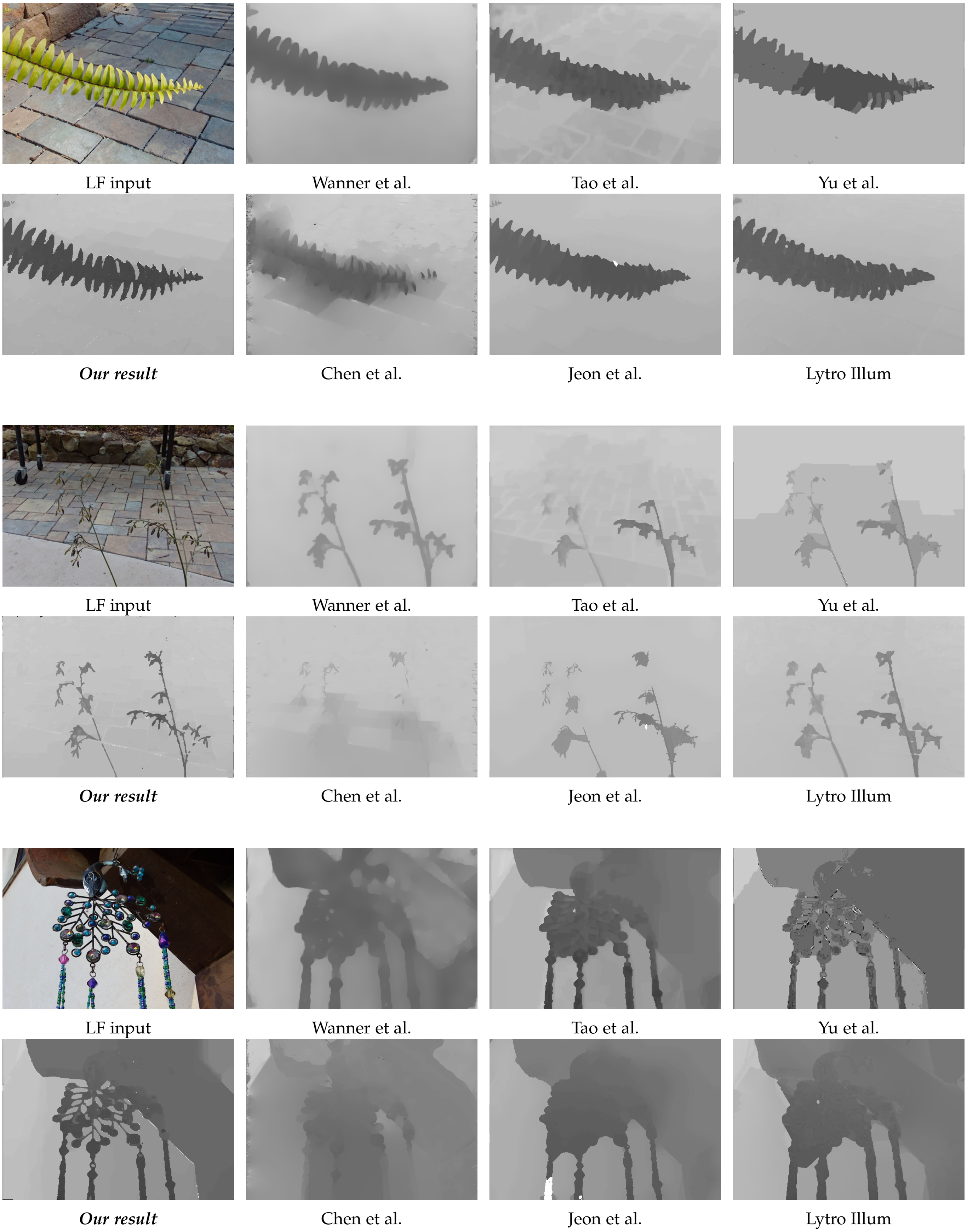


Fig. 11: *Depth estimation results on real data taken by the Lytro Illum light field camera. It can be seen that our method realistically captures the thin structures and occlusion boundaries, while other methods fail, or generate dilated structures.*



LF input | Wanner et al. | Tao et al. | Yu et al.

Our result | Chen et al. | Jeon et al. | Lytro Illum

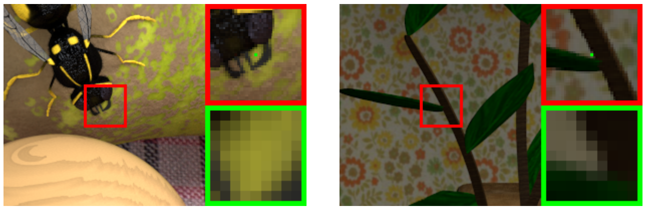
LF input | Wanner et al. | Tao et al. | Yu et al.

Our result | Chen et al. | Jeon et al. | Lytro Illum

LF input | Wanner et al. | Tao et al. | Yu et al.

Our result | Chen et al. | Jeon et al. | Lytro Illum

Fig. 11: *Depth estimation results on real data taken by the Lytro Illum light field camera (continued). It can be seen that our method realistically captures the thin structures and occlusion boundaries, while other methods fail, or generate dilated structures.*



(a) Small area occlusion (b) Multi-occluder occlusion

Fig. 12: *Limitations. The upper insets show close-ups of the red rectangle, while the lower insets show the angular patches of the green (central) pixels when refocused to the correct depth. (a) Our algorithm cannot handle occlusions where the occluded area is very small, so that there is no simple line that can separate the angular patch. (b) Also, if more than one occluder is present around the pixel, it is not enough to just divide the angular domain into two regions.*

6.3 Limitations and Future Work

First, our algorithm cannot handle situations where the occluded plane is very small relative to the angular patch size, or if the single occluder assumption fails to hold (Fig. 12). If the occluded area is very small, there is no simple line that can separate the angular patch into two regions. If we have multiple edges intersecting at a point, its angular patch needs to be divided into more than two regions to achieve photo consistency. This may be addressed by inspecting the spatial patch around the current pixel instead of just looking at the edges. Second, our algorithm cannot perform well if the spatial edge detector fails or outputs an inaccurate orientation. We also assume the light-field is bandlimited [17], so aliasing does not occur and we can always find consistent correspondences in the original light-field representation. Finally, similar to previous stereo methods, our algorithm cannot perform well at textureless regions. In addition, since we only use half the angular patch around edges, it might introduce some confusion in certain cases. For example, a special case would be a plane which is uniform on one side and textured on the other side. Using previous methods, the depth around the separating edge can be uniquely determined using the entire angular patch. However, no matter which depth we refocus to, the angular patch will be uniform on one side, and our method will not be able to find the correct depth. In this case, the unary cost will be indiscernible, and we will rely on neighboring pixels in the textured region to determine its depth (by smoothness constraint), just as previous methods rely on neighboring pixels to determine the depths in uniform regions.

7 Conclusion

In this paper, we propose an occlusion-aware depth estimation algorithm. We show that although pixels around occlusions do not exhibit photo-consistency in the angular patch when refocused to the correct depth, they are still photo-consistent for part of the patch. Moreover, the line separating the two regions in the angular domain has the same orientation as the edge in the spatial domain. Utilizing this information, the depth estimation process can be improved in two ways. *First*, we can enforce photo-consistency on only the region that is coherent. *Second*, by exploiting depth, correspondence and refocus cues, we can perform occlusion prediction, so smoothing over these boundaries can be avoided in the regularization. We demonstrate the benefits of our algorithm on various synthetic datasets as well as real-world images with fine structures, extending the range of objects that can be captured in 3D with consumer light-field cameras.

Acknowledgement

We acknowledge support from ONR grant N00014-15-1-2013, a Berkeley Fellowship, Intel, and funding from Draper, Nokia and Sony to the UC San Diego Center for Visual Computing.

References

[1] 3D modeling, animation, and rendering software. http://www.autodesk.com/products/3ds-max. 6

[2] Free 3ds models. http://www.free-3ds-models.com. 6

[3] Lytro redefines photography with light field cameras. Press release, Jun 2011. http://www.lytro.com. 1

[4] Edward H Adelson and John Y. A. Wang. Single lens stereo with a plenoptic camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 14(2):99–106, 1992. 1

[5] Michael Bleyer, Carsten Rother, and Pushmeet Kohli. Surface stereo with soft segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 2

[6] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26(9):1124–1137, 2004. 6

[7] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(11):1222–1239, 2001. 6

[8] Can Chen, Haiting Lin, Zhan Yu, Sing Bing Kang, and Jingyi Yu. Light field stereo matching using bilateral statistics of surface cameras. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1518–1525, 2014. 1, 2, 6

[9] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of Siggraph*, 1996. 6

[10] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proceedings of Siggraph*, 1996. 1

[11] Hae-Gon Jeon, Jaesik Park, Gyeongmin Choe, Jinsun Park, Yunsu Bok, Yu-Wing Tai, and In So Kweon. Accurate depth map estimation from a lenslet light field camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2, 4, 6

[12] Changil Kim, Henning Zimmer, Yael Pritch, Alexander Sorkine-Hornung, and Markus H Gross. Scene reconstruction from high spatio-angular resolution light fields. *ACM Transactions on Graphics (TOG)*, 32(4):73, 2013. 2

[13] Vladimir Kolmogorov and Ramin Zabih. Multi-camera scene reconstruction via graph cuts. In *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*. 2002. 2

[14] Vladimir Kolmogorov and Ramin Zabin. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26(2):147–159, 2004. 6

[15] Venkat Krishnamurthy and Marc Levoy. Fitting smooth surfaces to dense polygon meshes. In *Proceedings of Siggraph*, 1996. 6

[16] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of Siggraph*, 1996. 1

[17] Chia-Kai Liang and Ravi Ramamoorthi. A light transport framework for lenslet light field cameras. *ACM Transactions on Graphics (TOG)*, 34(2):16, 2015. 12

[18] Scott McCloskey. Masking light fields to remove partial occlusion. In *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*, pages 2053–2058, 2014. 2

[19] Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, and Pat Hanrahan. Light field photography with a hand-held plenoptic camera. *Computer Science Technical Report CSTR*, 2(11), 2005. 4

[20] Christian Perwass and Lennart Wietzke. Single lens 3D-camera with extended depth-of-field. In *Proceedings of IS&T/SPIE Electronic Imaging*, 2012. 1, 2

[21] Carsten Rother, Vladimir Kolmogorov, Victor Lempitsky, and Martin Szummer. Optimizing binary mrfs via extended roof duality. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. 2

[22] Michael W Tao, Sunil Hadap, Jitendra Malik, and Ravi Ramamoorthi. Depth from combining defocus and correspondence using light-field cameras. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013. 1, 2, 3, 4, 6

[23] Michael W Tao, Pratul P Srinivasan, Jitendra Malik, Szymon Rusinkiewicz, and Ravi Ramamoorthi. Depth from shading, defocus, and correspondence using light-field angular coherence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[24] Greg Turk and Marc Levoy. Zippered polygon meshes from range images. In *Proceedings of Siggraph*, 1994. 6

[25] Ting-Chun Wang, Alexei Efros, and Ravi Ramamoorthi. Occlusion-aware depth estimation using light-field cameras. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015. 2, 6

[26] Sven Wanner and Bastian Goldluecke. Globally consistent depth labeling of 4D light fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 1, 2, 6

[27] Sven Wanner, Stephan Meister, and Bastian Goldlücke. Datasets and benchmarks for densely sampled 4D light fields. In *Annual Workshop on Vision, Modeling and Visualization*, pages 225–226, 2013. 6, 7, 8

[28] Yichen Wei and Long Quan. Asymmetrical occlusion handling using graph cut for multi-view stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. 2

[29] Oliver Woodford, Philip Torr, Ian Reid, and Andrew Fitzgibbon. Global stereo reconstruction under second-order smoothness priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 31(12):2115–2128, 2009. 2

[30] Zhan Yu, Xinqing Guo, Haibing Ling, Andrew Lumsdaine, and Jingyi Yu. Line assisted light field triangulation and stereo matching. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013. 1, 2, 6

Ting-Chun Wang received his B.S. degree in 2012 at National Taiwan University and is currently pursuing a PhD at U.C. Berkeley, Electrical Engineering and Computer Science Department, advised by Ravi Ramamoorthi and Alexei Efros. His research interest is in computational photography and computer vision problems, particularly light-field technologies with both computer vision and computer graphics applications.

Alexei A. Efros received his BS degree in Computer Science from the University of Utah in 1997 and PhD from UC Berkeley in 2003. Following a postdoc at Oxford University, he was nine years on the faculty of Carnegie Mellon University, while also been affiliated with cole Normale Suprieure/INRIA. He is now an associate professor at EECS Department at UC Berkeley. He is a recipient of CVPR Best Paper Award (2006), NSF CAREER award (2006), Sloan Fellowship (2008), Guggenheim Fellowship (2008), Okawa Grant (2008), Finmeccanica Career Development Chair (2010), SIGGRAPH Significant New Researcher Award (2010), ECCV Best Paper Honorable Mention (2010), and the Helmholtz Test-of-Time Prize (2013).

Ravi Ramamoorthi received his BS degree in engineering and applied science and MS degrees in computer science and physics from the California Institute of Technology in 1998. He received his PhD degree in computer science from Stanford University Computer Graphics Laboratory in 2002, upon which he joined the Columbia University Computer Science Department. He was on the UC Berkeley EECS faculty from 2009-2014. Since July 2014, he is a Professor of Computer Science and Engineering at the University of California, San Diego and Director of the UC San Diego Center for Visual Computing. His research interests cover many areas of computer vision and graphics, with more than 100 publications. His research has been recognized with a number of awards, including the 2007 ACM SIGGRAPH Significant New Researcher Award in computer graphics, and by the white house with a Presidential Early Career Award for Scientists and Engineers in 2008 for his work on physics-based computer vision. He has advised more than 20 Postdoctoral, PhD and MS students, many of whom have gone on to leading positions in industry and academia; and he has taught the first open online course in computer graphics on the EdX platform in fall 2012, with more than 80,000 students enrolled in that and subsequent iterations.