Light Field Layer Matting

Juliet Fiss
University of Washington
jfiss@uw.edu

Brian Curless
University of Washington
curless@cs.washington.edu

Richard Szeliski
Microsoft Research
szeliski@microsoft.com

Abstract

In this paper, we use matting to separate foreground layers from light fields captured with a plenoptic camera. We represent the input 4D light field as a 4D background light field, plus a 2D spatially varying foreground color layer with alpha. Our method can be used to both pull a foreground matte and estimate an occluded background light field. Our method assumes that the foreground layer is thin and fronto-parallel, and is composed of a limited set of colors that are distinct from the background layer colors. Our method works well for thin, translucent, and blurred foreground occluders. Our representation can be used to render the light field from novel views, handling disocclusions while avoiding common artifacts.

1. Introduction

Many photographs of natural scenes have a layered composition, where a foreground layer partially occludes background content. The *natural image matting* problem addresses the separation and modeling of these layers. Most image matting algorithms accept a trimap, which segments the image into foreground, background, and unknown regions. However, for many common types of layered scenes, providing a trimap is not practical. For example, consider a scene photographed through a dusty or dirty window. In this example, the foreground layer is spatially complex, made of many small, irregular, translucent occluders.

In this paper, we propose a method for matting such layers from light fields. Our method assumes that the foreground plane is fronto-parallel. Instead of a trimap, our method takes as user input two depth parameters: d_f, which specifies the depth at which the foreground layer is most in focus, and d_τ, a threshold depth that separates the foreground layer from the background layer. To select these parameters, the user sweeps a synthetic focal plane through the scene (e.g. with a depth slider) and makes selections by visual inspection. These parameters are necessary to signal user intent: what content should be considered part of the foreground, and what content part of the background?

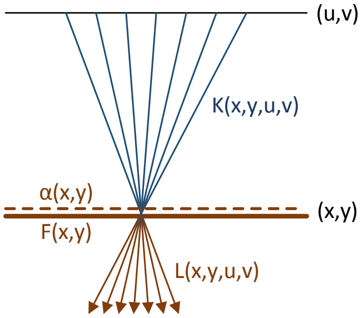

Figure 1. We model the input light field as foreground layer composited over a background light field.

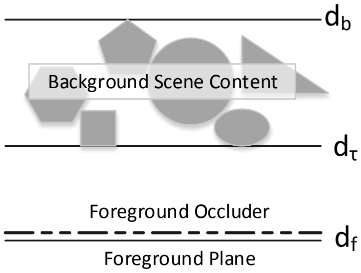

Figure 2. Depth parameters inform the algorithm where to place the foreground plane, and whether content should be considered part of the background or the foreground.

We model the input light field as a composite of two layers: a background light field, plus a foreground layer with constant depth, spatially varying color, and spatially varying alpha. Our method recovers the background light field, foreground color, and foreground alpha automatically.

1.1. Background

We model a light field L, captured by a plenoptic camera, as a composite of a background light field K with a foreground layer color map F and alpha matte α. Our goal is to estimate K, F, and α given L and some additional parameters describing the scene.

In this section, we use a two-plane ray parameterization

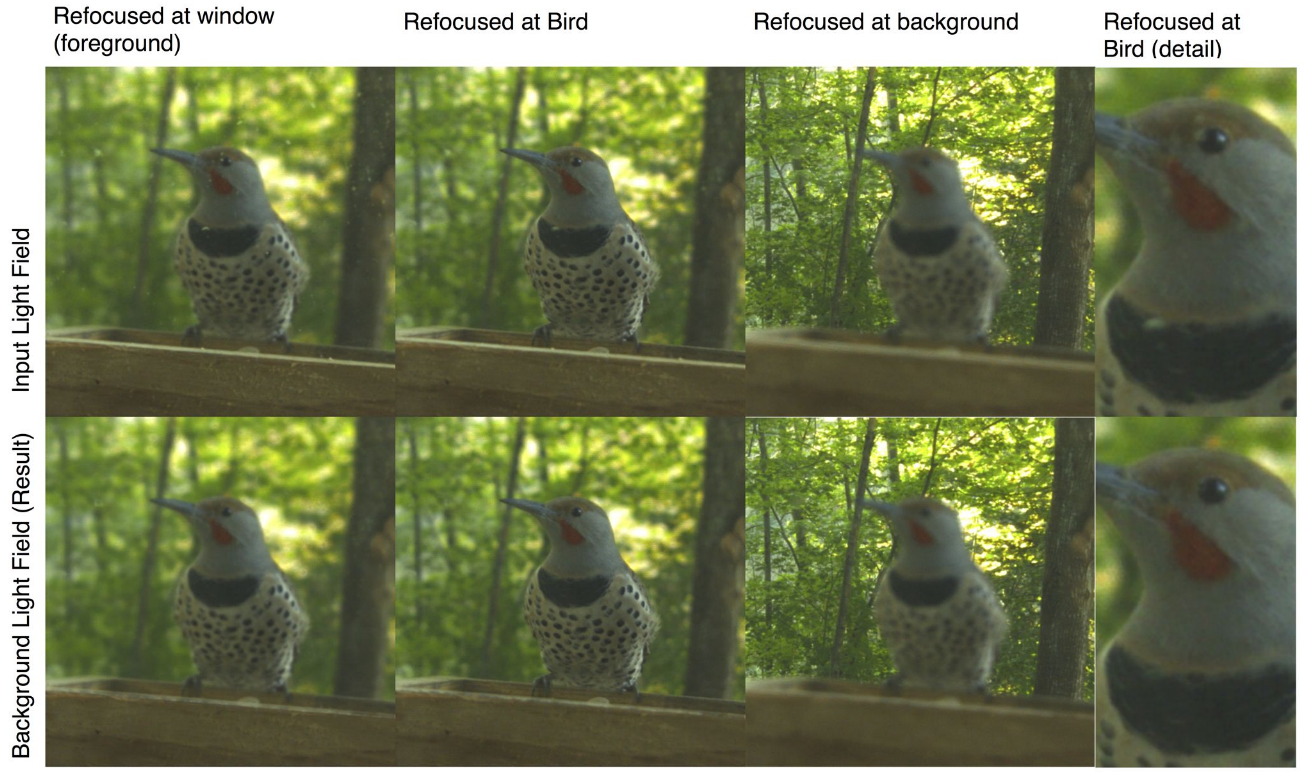
Refocused at window (foreground)
Refocused at Bird
Refocused at background
Refocused at Bird (detail)
Input Light Field
Background Light Field (Result)

Figure 3. In this scene, a bird is photographed behind a dusty window. Top: Input light field refocused at three depths. Bottom: Background light field, with the window dust removed, refocused at the same depths.

to describe the scene. We define (x, y) to be the plane where the foreground layer is located and (u, v) to be a plane in the background of the scene. According to our model,

$$L(x,y,u,v) = \alpha(x,y)F(x,y) + (1-\alpha(x,y))K(x,y,u,v) \tag{1}$$

We define image B as the background light field refocused at the foreground plane:

$$B(x,y) = \frac{1}{A}\int_A K(x,y,u,v)\,du\,dv \tag{2}$$

We define image C as the input light field refocused at the foreground plane:

$$C(x,y) = \frac{1}{A}\int_A L(x,y,u,v)\,du\,dv \tag{3}$$

Combining the above equations, we see that

$$\begin{aligned} C(x,y) = & \qquad (4)\\ =&\frac{1}{A}\int_A \left[\alpha(x,y)F(x,y) + (1-\alpha(x,y))K(x,y,u,v)\right]\,du\,dv\\ =&\alpha(x,y)F(x,y) + (1-\alpha(x,y))\frac{1}{A}\int_A K(x,y,u,v)\,du\,dv\\ =&\alpha(x,y)F(x,y) + (1-\alpha(x,y))B(x,y)\end{aligned}$$

Therefore, we can use matting to recover F and α from B and C.

1.2. Applications

After the layers of the scene have been recovered, they can be used for a number of applications. If the foreground layer represents contamination, such as in the dirty window example described above, the background light field can be used in place of the input light field for any standard application. Like the input the light field, the background light field can be refocused, and non-Lambertian effects such as specular reflections are preserved.

The foreground and background layers can also be used together to render novel views. In this paper, we compute our results on images from a Lytro camera, and compare our renderings to those from the Lytro perspective shift feature. The Lytro feature is fully automatic, with no additional information about the scene, but it suffers from artifacts on many types of images that are well suited to our method.

As Wanner and Goldluecke show in [22], light field depth estimation algorithms that estimate only one depth layer tend to fail in systematic ways on scenes with multiple physical layers. These depth estimation algorithms adopt the assumption proposed in *The Lumigraph* [6] and *Light*

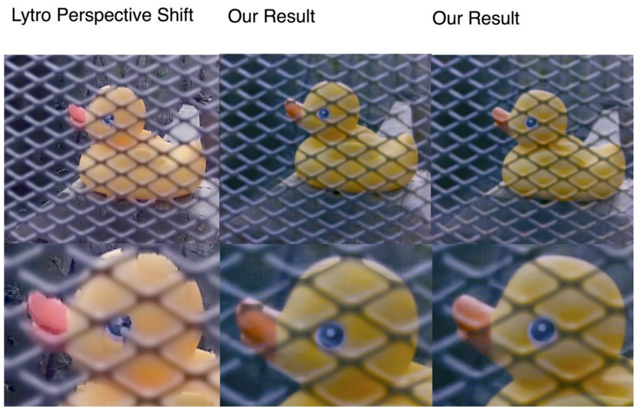

Figure 4. In this scene, a wire mesh occludes a rubber duck. We compare our novel view renderings to the Lytro perspective shift feature. Top: novel views. Bottom: detail.

Field Rendering [10] that rays reflect off of a single object and pass through a transparent medium, and therefore, radiance is constant along every ray. This assumption implies that each ray has one depth and one color. However, in layered scenes, ray color is often a mixture of foreground and background colors. Common cases of color mixing occur when the foreground occluders are translucent or blurry, at thin occluders, and at the edges of thick occluders. Depth estimation algorithms that cannot resolve the ambiguity between the foreground and background at such mixed pixels are forced to make hard decisions about which depth label to assign to any given ray (or pixel). The result is a depth map with hard, jagged object boundaries, and sometimes inaccurate breaks and tears. Such errors in the depth map create artifacts when rendering the scene from novel views.

Our method for computing novel views first removes the foreground layer from the background light field, then computes the novel view for the background light field, and finally composites the shifted foreground layer over the background rendering. This composite rendering appears smooth as the viewing angle changes. Another benefit of our approach is that the foreground layer can be composited over the background light field at any spatial location or depth. These parameters can be used to enhance the perspective shift effect or change the relative depths of the foreground and background layers.

2. Related Work

In this paper, we consider a sub-problem of a more general problem:

> Given images of an arbitrary scene containing both occluders and occluded content, cleanly separate and model the occluders (foreground layer(s)) as well as the complete occluded content (background layer). For the foreground layer(s), estimate color, alpha, and depth. For the background layer, estimate color and depth.

This general problem is underdetermined. For most arbitrary scenes, many alternative models could be used to describe the original input images. However, for applications such as matting and rendering novel views, the user typically desires only one of these solutions. Therefore, constraints and priors are typically imposed on the general problem to define a better-constrained sub-problem. In this section, we review some of these sub-problems, which have been analyzed in prior work.

2.1. Natural Image and Light Field Matting

The natural image matting problem is concerned with estimating foreground and alpha layers from images with arbitrary backgrounds, but does not typically estimate the complete background. Techniques such as Bayesian matting [3] use a single input view as well as a user-supplied trimap. Cho *et al*. [1] take as input a light field and a user-supplied trimap for the central view, and compute consistent foreground alpha mattes across the views of the light field.

Several techniques use multiple views of the scene to create a trimap automatically. Wexler *et al*. [24] assume the background is already known, or can be easily calculated by taking the median of all input views. That technique also uses several priors based on limiting assumptions about the foreground object. Joshi *et al*. [8] use multiple synchronized views to create a trimap automatically. McGuire *et al*. [14] use synchronized video streams with varying degree of defocus to create a trimap automatically.

2.2. Occluded Surface Reconstruction

The occluded surface reconstruction problem is concerned with reconstructing occluded surfaces, but is not typically concerned with reconstructing the occluders themselves. An analysis by Vaish *et al*. [21] compares methods for estimating the depth and color of occluded surfaces from light fields, using alternative stereo measures such as entropy and median. While those methods work very well in cases of light occlusion, they break down once occlusion exceeds 50 percent. Therefore, in this paper, we use the standard mean and variance methods typically used in multiview stereo for estimating color and depth of occluded surfaces. In Eigen *et al*. [4], small occluders such as dirt or rain are removed from images taken through windows. That method uses a single image as input and trains a convolutional neural network to detect and remove the occluders. The system must be trained on each specific type of occluder. In this paper, we demonstrate our technique on a similar application. However, we use a light field as input and do not use any machine learning or external datasets when computing our results.

The approach of Gu *et al*. [7] is closely related to our own. That paper uses multiple input images and defocus to separate foreground and background layers. Different constraints, priors, and input sets are required depending on the type of scene to be reconstructed. For removing dirt from camera lenses, the approach leverages either a calibration step or images of multiple scenes taken through the same dirty lens, plus priors on the natural image statistics of the scene. For removing thin occluders, the method requires knowledge of the PSF of the camera, plus images taken with the background in focus and the foreground at different levels of defocus. Foreground and background depth information is also required if only two of these images are supplied. Like our method, their method reconstructs not only the occluded surface, but also the occluder. However, their method of removing thin occluders is limited to occluders that do not add any of their own radiance to the scene (i.e. they are dark grey or black in color). Their method is also limited to static scenes, because multiple exposures must be captured.

2.3. Layered Image Representations

In *Layered Depth Images* [17], Shade *et al*. propose the Layered Depth Image representation, in which each pixel is assigned multiple depths and colors. Their representation can be used for image based rendering to avoid the appearance of holes where disocclusions occur. In their work and related subsequent work [19] [25], the foreground is assumed to be opaque (that is, the foreground alpha either 1 or 0). Zitnick *et al*. [26] use a similar layered representation for image-based rendering, adding a thin strip of non-binary foreground alpha (computed with Bayesian matting) to represent mixed pixels at the foreground-background boundary.

2.4. Layered Light Fields

Layered light field representations have been used in prior work for rendering synthetic scenes. Lischinski and Rappoport [12] propose the layered light field as a collection of layered depth images from different viewpoints. They use this representation for image-based rendering of non-Lambertian effects in synthetic scenes. Vaidyanathan *et al*. [20] propose partitioning the samples of a synthetic light field into depth layers for rendering defocus blur. Wanner and Goldluecke [22] estimate two layered depth maps for a light field of a layered scene.

2.5. Light Field Depth Estimation and Rendering

The method proposed in this paper builds on prior work on computing depth maps and digitally refocused images from light fields. Any light field processing system could be used to implement the proposed techniques, as long as it provides the following operations:

1. Given a light field L and a range of hypothesis depths $d_0 ... d_n$, compute a depth for every ray in a given light field, the per-ray depth map D. This is equivalent to having one depth map per view of the scene.

2. A refocusing operation $R\left(L, d, W\right)$, which projects all rays in light field L into the central view, focused at depth d. This operation takes an optional weight map W, which is used to weight each ray in the light field before refocusing. For standard refocusing, this weight is 1 for all pixels. If a light field is refocused through a range of depths, $d_0 ... d_n$, the result is a focal stack $Q = R\left(L, d_0 ... d_n, W\right)$. An all-in-focus image can be produced by refocusing each ray at its in-focus depth, as given by depth map D using $A\left(L, D, W\right)$.

3. An interpolation operation $L = S\left(Q, D\right)$, which interpolates colors L into the light field domain from a focal stack of images Q given by a per-ray depth map D. We assume that both the focal stack and the depth map sweep through the depths $d_0 ... d_n$.

4. For rendering the scene from novel views, we also require our rendering system to have a view-shifted refocusing operation $V\left(L, D, d, s, t\right)$, which refocuses light field L with depth map D at view (s, t), focused at depth d.

We capture our input light fields with a Lytro camera [13] and use the projective splatting-based rendering system developed for this camera by Fiss *et al*. [5]. The rendering technique used by this system is similar to the projective system described by Liang and Ramamoorthi [11], but adjusts splat size based on per-ray depth. This rendering system uses simple winner-take-all variance to compute a depth per ray. A more sophisticated depth estimation method [9] [18] [23] could be used instead; however each method of depth estimation will introduce its own biases. For rendering the scene from novel views, we simply shift the splat location of each ray proportional to its depth. Again, more sophisticated techniques have been developed for view-dependent rendering that could be used in place of the basic algorithm [16] [22] [23] .

3. Methods

As input, our method takes a light field image L, as well as three depth parameters: a frontmost depth d_f, a threshold depth d_τ, and a backmost depth d_b. The frontmost depth d_f specifies where the foreground layer is in focus. The threshold depth d_τ divides the foreground and background scene content. In general, because the problem is underdetermined, the solution will differ depending on the supplied depth parameters. For example, depending on the placement of d_τ, the algorithm will treat an object in the middle

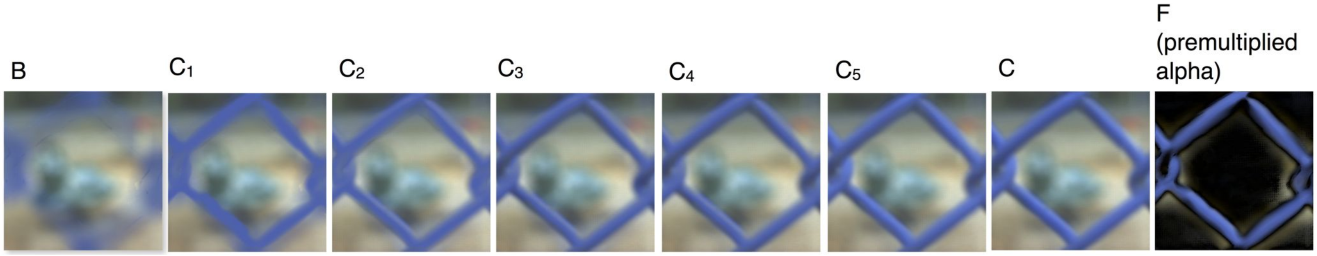

Figure 5. Multicolor matting: at each iteration, one color and alpha matte is estimated. We show the intermediate composite images.

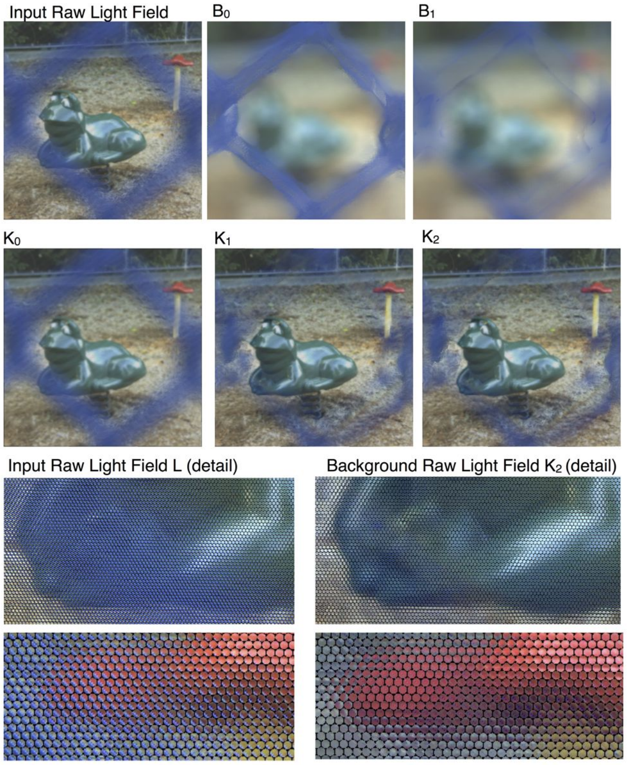
Input Raw Light Field
B0
B1
K0
K1
K2
Input Raw Light Field L (detail)
Background Raw Light Field K2 (detail)

Figure 6. In this scene, playground equipment is photographed through a blue fence. Top: B and K for two iterations of our algorithm. The occluder is thick, so it is not completely removed. Bottom: detail.

of the scene as either an occluder or an occluded object. The backmost depth is a system parameter, which specifies the farthest depth in the scene that includes background content. All scene content of interest is assumed to occur within the depth range $d_b...d_f$. The output of the algorithm is: (1) a 4D light field K representing the background layer only, with contamination from the foreground layer largely removed and (2) a foreground layer represented by a color layer F and an alpha matte α.

We assume that the foreground layer occurs within a narrow range of depths and is composed of a limited set of colors that are distinct from the colors in the background layer. The more closely the input light field matches these assumptions, the more cleanly the matting operation will be able to separate foreground from background.

Our method works iteratively, alternately estimating the foreground and background layers. We start by estimating an initial background light field, K_0. At each iteration of the outer loop i, we refocus the background light field K_i at the foreground plane d_f, to produce image B_i. If we compare B_i to image C, the original light field refocused at the foreground plane, we see that $RMSE(B_i, C)$ measures the amount of foreground content in C that is not in B_i. Next, in the matting step, we use B_i and C, to compute α_i and F_i.

The matting step is itself iterative. At each iteration of the matting step j, we estimate one foreground color $F_{i,j}$ and one alpha matte $\alpha_{i,j}$. From B_i and C, we compute $\alpha_{i,0}$ and $F_{i,0}$. We then compute the composite $C_{i,1} = \alpha_{i,0} + (1 - \alpha_{i,0})B_i$. On the next iteration, we use $C_{i,1}$ and C to compute $\alpha_{i,1}$ and $F_{i,1}$. $C_{i,j+1}$ builds upon $C_{i,j}$ to become closer to C, and $RMSE(C_{i,j}, C)$ decreases with each iteration of the matting step. We stop iterating when $RMSE(C_{i,j}, C) < 2$ (usually 1-5 iterations). One color is estimated per iteration j, so the number of color primaries used to model the foreground is the number of iterations in the matting step. The final reconstruction C_i is the composite of B_i with the final multicolor foreground layer. C_i is visually close to C. $F_{i,j}$ and $\alpha_{i,j}$ from all iterations are combined to produce F_i and α_i. Figure 5 illustrates this iterative multicolor matting step.

Finally, we use the estimate of F_i and α_i to improve our estimate of the background light field. The entire algorithm is repeated for several iterations. The outer loop terminates when $RMSE(B_i, B_{i-1})$ stops decreasing past some epsilon. Figure 6 illustrates some intermediate iterative results of our method.

3.1. Initial Foreground and Background Colors

The first step in our method is to compute an initial estimate of the background light field, K_0. We sweep a plane through the range of background depths $d_b...d_\tau$, computing focal stack Q:

$$Q = R(L, d_b ... d_\tau, 1) \tag{5}$$

Next, we compute a per-ray depth map $D_{b,0}$ for the input light field, where depth values are only allowed in the range $d_b ... d_\tau$. This depth map will have accurate depth values for rays that intersect textured objects or edges in the background of the scene, but unreliable depth values for rays that intersect an opaque occluder in the foreground region of the scene.

We estimate the background color of each ray by interpolating colors from the refocused images of the scene Q:

$$K_0 = S(Q, D_{b,0}) \tag{6}$$

Many high frequency details in the background image content, such as small specular highlights, are lost when computing K_0. However, these details are unimportant at this step, because K_0 will be rendered out of focus during the next step.

Next, we estimate the degree to which rays from the original light field L are in focus at the foreground plane d_f. We compute a per-ray depth map D_f for L, where depth values are only allowed in the range $d_\tau ... d_f$. This depth map will have accurate depth values for rays that intersect textured objects or edges in the foreground depth range, but unreliable depth values for rays that pass through transparent regions of the foreground depth range. We scale D_f between 0 and 1 to create weight map w_f, which is 0 for rays that are in focus near d_f. Finally, to compute B_0, we refocus K_0 at d_f, down-weighting any rays that are found to be in focus near d_f:

$$B_0 = R(K_0, d_f, w_f) \tag{7}$$

To compute C, we simply refocus the original light field at the foreground plane.

$$C = R(L, d_f, 1) \tag{8}$$

3.2. Single Color Matting

Some common types of foreground occluders, such as window screens and dust on windows, can be adequately described by a single (spatially uniform) RGB foreground color F with spatially varying alpha matte α [2] [7]. Given B and C as computed previously, we compute

$$(F_i, \alpha_i) = \arg\min_{F', \alpha'} \|\alpha' F' + (1 - \alpha') B_i - C)\|_2^2 \tag{9}$$

In this optimization, α is constrained between 0 and 1, and image colors are constrained between 0 and 255. F is initialized to medium gray (128, 128, 128), and α is initialized to 0 at all pixels. The optimization is solved using nonlinear least squares in the MATLAB optimization toolbox. We do not use any regularization or priors when solving for α and F, but these could be added if such information about the scene is known.

3.3. Multicolor Matting

For scenes with multicolor foregrounds, we would like to allow F to vary spatially. However, allowing F to be any color is very susceptible to noise. Therefore, we constrain F to be a linear combination of a small subset of colors. Rather than estimating the linear weights directly, we iteratively estimate color layers $F_{i,0...m}$ and $\alpha_{i,0...m}$. We initialize $C_{i,0} = B_i$ and solve

$$(F_{i,j}, \alpha_{i,j}) = \arg\min_{F', \alpha'} \|\alpha' F' + (1 - \alpha') C_{i,j} - C)\|_2^2 \tag{10}$$

$$C_{i,j+1} = \alpha_{i,j} F_{i,j} + (1 - \alpha_{i,j}) C_{i,j} \tag{11}$$

This matting procedure is iterated until the RMSE between $C_{i,j}$ and C drops below a threshold (we use threshold=2). $F_{i,0...m}$ and $\alpha_{i,0...m}$ are then simplified into single layers F_i and α_i using alpha compositing [15], where F_i is a spatially varying linear combination of $F_{i,0...m}$:

$$\alpha_i = 1 - \prod_{j=0}^{m} (1 - \alpha_{i,j}) \tag{12}$$

$$F_i = \frac{1}{\alpha_i} \Bigg[\alpha_{i,m} F_{i,m} + \alpha_{i,m-1} (1 - \alpha_{i,m}) F_{i,m-1} + ... + \alpha_{i,0} \prod_{j=1}^{m} (1 - \alpha_{i,j}) F_{i,0} \Bigg] \tag{13}$$

3.4. Refining Background Colors

Next, we matte out the foreground layer from the original light field to produce a decontaminated light field J_i.

$$J_i = \frac{L - \alpha_i F_i}{1 - \alpha_i} \tag{14}$$

J_i will have the influence of the front layer reduced, but likely not completely removed. Where α_i is close to 1, J_i will have amplified noise. Therefore, J_i is replaced with K_i where α_i is close to 1 (we use 0.9), and linearly blended when α_i exceeds a threshold (0.3).

Next, we use J_i to compute a weighted all-in-focus image $A_i = A(J_i, D_{b,i}, 1 - \alpha_i)$. A_i represents the background colors with the foreground rays down-weighted or excluded. We use the value of α_i for each ray as a measure of how much it is occluded by the foreground. If α_i exceeds a threshold (0.3), then the ray is not used to compute A_i. Finally, J_i is blended with interpolated values from A_i (using threshold $t = 0.4$) to compute the next estimate of the background light field.

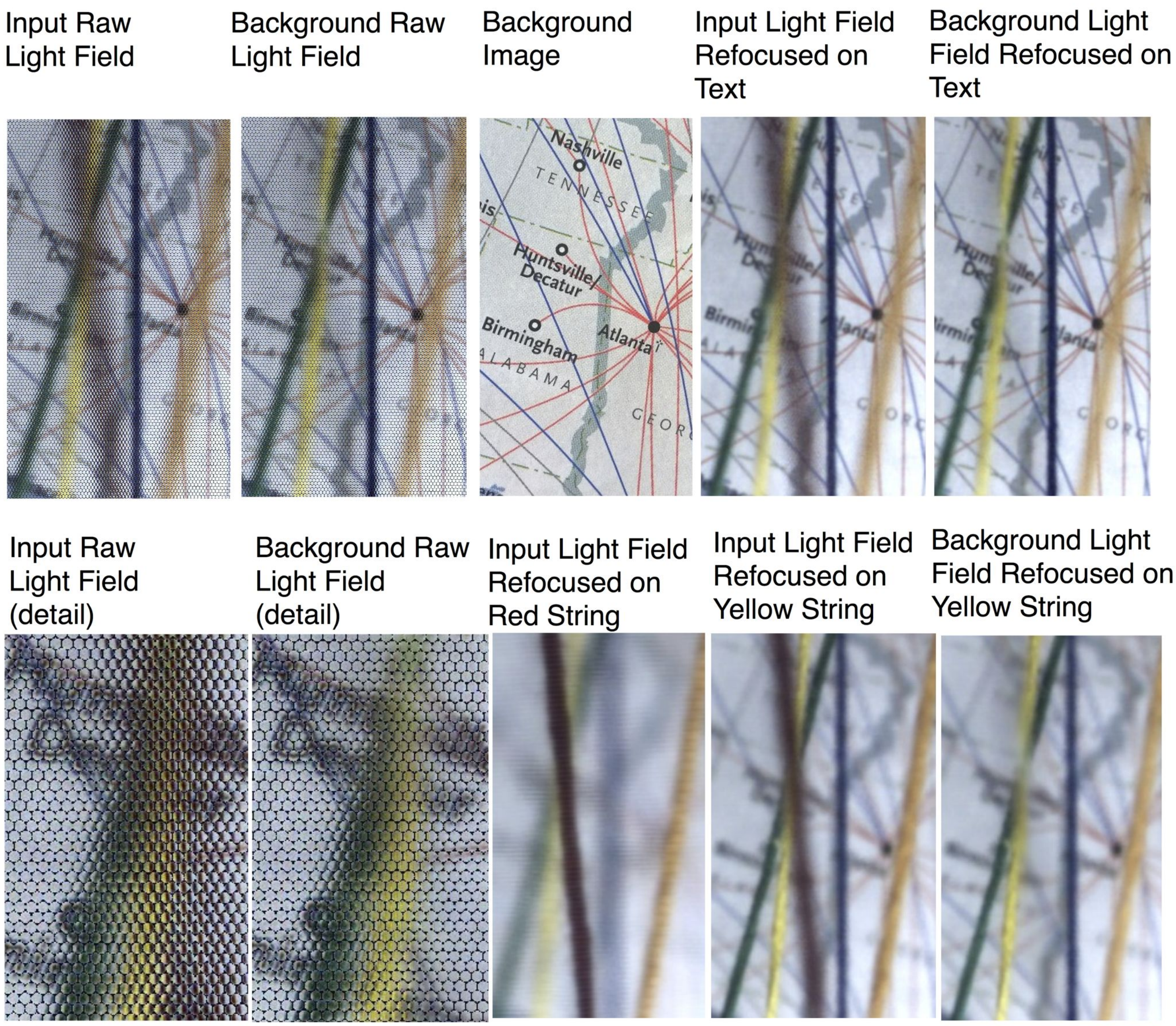
Input Raw Light Field
Background Raw Light Field
Background Image
Input Light Field Refocused on Text
Background Light Field Refocused on Text
Nashville
TENNESSEE
Huntsville/ Decatur
Birmingham
Atlanta
ALABAMA
GEOR
Input Raw Light Field (detail)
Background Raw Light Field (detail)
Input Light Field Refocused on Red String
Input Light Field Refocused on Yellow String
Background Light Field Refocused on Yellow String

Figure 7. In this scene, colored strings occlude a map. The red string is removed using our method. We compare the input and background light fields in their raw form, and refocused at two different depths. We also compare to an image of the map used in the background.

$$K_{i+1} = \begin{cases} \left(1 - \frac{\alpha}{t}\right) J_i + \left(\frac{\alpha}{t}\right) S(A_i, D_{b,i}), & \text{if } \alpha \le t \\ S(A_i, D_{b,i}), & \text{otherwise} \end{cases} \tag{15}$$

For the next iteration, we compute per-ray depth map $D_{b,i}$ using K_i. We also compute B_i, using α_i as a weight.

$$B_i = R(K_i, d_f, (1 - \alpha_i)) \tag{16}$$

4. Results

Examples of background estimation (removing the foreground layer) are shown in Figures 3, 6, and 7. In the case of large foreground occluders, where no rays see any part of the background, the matting operation will not have enough information to completely remove the foreground colors (Figure 6). In Figures 4, 8, and 9, we composite a shifted foreground layer over a novel view of the background, and compare to the Lytro perspective shift feature. Note that the Lytro rendering often has breaks an inaccuracies in the foreground layer, leading to artifacts. Our method is able to avoid these kinds of artifacts. Many of our results are best viewed as animations. Please see our project page for animations and additional results http://grail.cs.washington.edu/projects/lflm/.

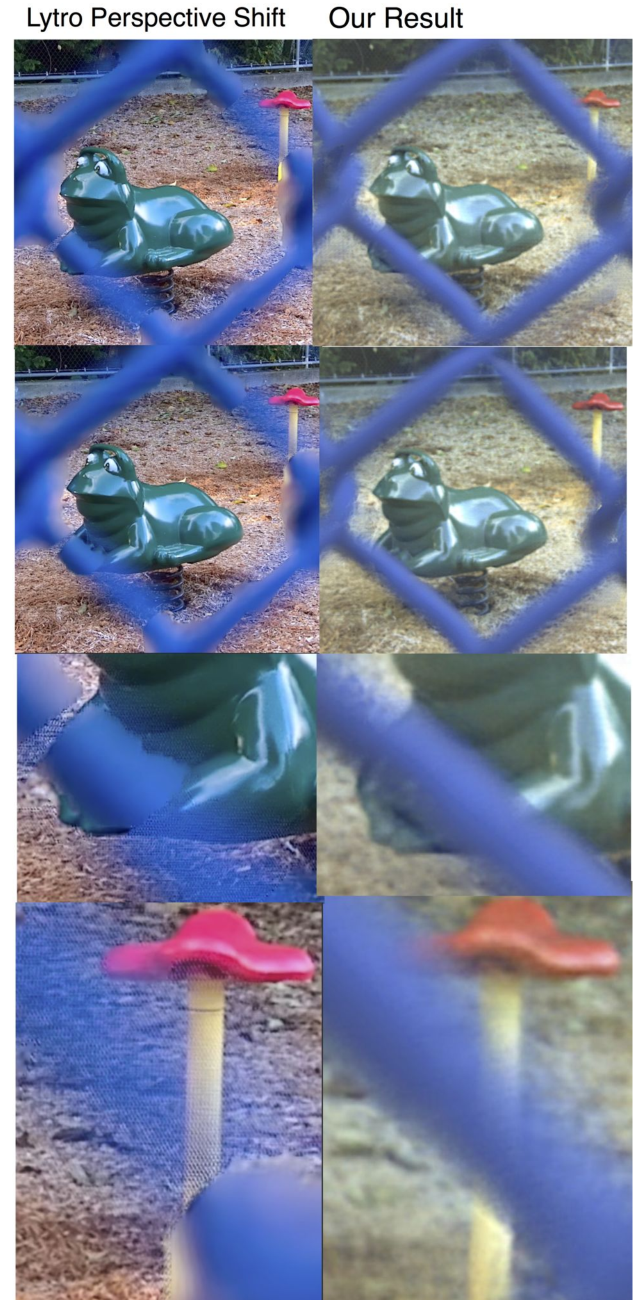

Figure 8. In this scene, playground equipment is photographed through a blue fence. We compare our novel view renderings to the Lytro perspective shift feature. We compare our novel view renderings to the Lytro perspective shift feature. Top: novel views. Bottom: detail.

5. Conclusion, Limitations, and Future Work

In this paper, we have presented a method to use matting to separate foreground layers from light fields captured with a plenoptic camera. Our method can be used both to pull a foreground matte and to estimate an occluded background light field. Our method works well for thin, translucent, and blurred foreground occluders. Our representation can be used to render the light field from novel views, handling disocclusions while avoiding common artifacts.

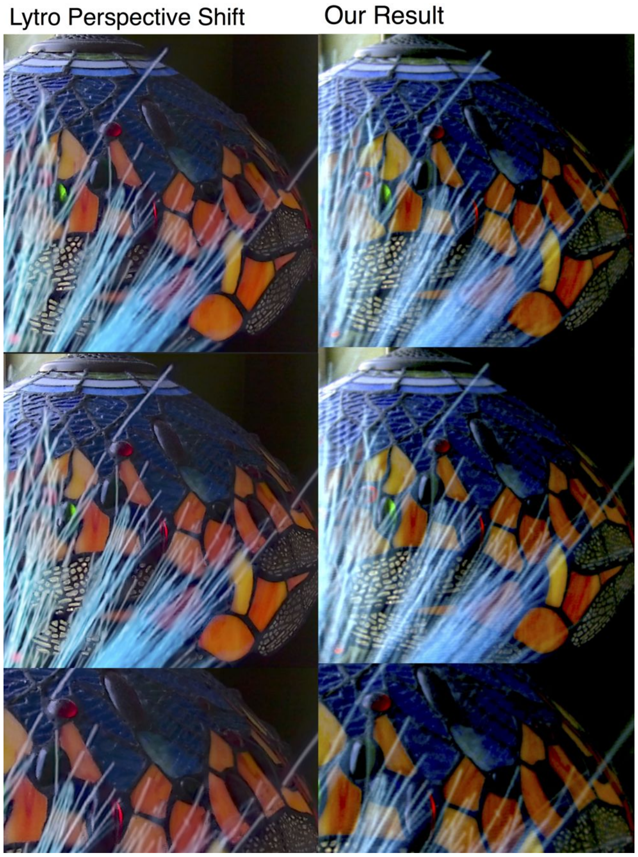

Figure 9. In this scene, blue hairs occlude a glass lamp. We compare our novel view renderings to the Lytro perspective shift feature. Top: novel views. Bottom: detail.

The technique we propose in this paper is limited in several ways, but we believe these limitations could be overcome in future work. Our method assumes that the foreground layer is thin, fronto-parallel, and at a known depth. In future work, we plan to extend our technique to work on foreground layers with complex or unknown depth. Also, in this paper, we consider only the case of a single foreground layer. We plan to extend our technique to work on light fields with multiple layers of foreground occluders. Finally, we assume that the foreground layer is composed of a small set of colors, which are distinct from the background layer colors. We plan to extend our method to work on more complex foregrounds.

References

[1] D. Cho, S. Kim, and Y.-W. Tai. Consistent matting for light field images. In *ECCV*, 2014. 3

[2] Y.-Y. Chuang, A. Agarwala, B. Curless, D. H. Salesin, and R. Szeliski. Video matting of complex scenes. *ACM Transactions on Graphics*, 21(3):243–248, July 2002. Special Issue of the SIGGRAPH 2002 Proceedings. 6

[3] Y.-Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski. A bayesian approach to digital matting. In *Proceedings of IEEE CVPR 2001*, volume 2, pages 264–271. IEEE Computer Society, December 2001. 3

[4] D. Eigen, D. Krishnan, and R. Fergus. Restoring an image taken through a window covered with dirt or rain. *Computer Vision, IEEE International Conference on*, 0:633–640, 2013. 3

[5] J. Fiss, B. Curless, and R. Szeliski. Refocusing plenoptic images using depth-adaptive splatting. In *Computational Photography (ICCP), 2014 IEEE International Conference on*, pages 1–9, May 2014. 4

[6] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *In Proceedings of SIGGRAPH 96*, pages 43–54. ACM, 1996. 2

[7] J. Gu, R. Ramamoorthi, P. Belhumeur, and S. Nayar. Removing image artifacts due to dirty camera lenses and thin occluders. *ACM Transactions on Graphics (SIGGRAPH ASIA 09)*, 28(5), Dec. 2009. 4, 6

[8] N. Joshi, W. Matusik, and S. Avidan. Natural video matting using camera arrays. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, pages 779–786, New York, NY, USA, 2006. ACM. 3

[9] C. Kim, H. Zimmer, Y. Pritch, A. Sorkine-Hornung, and M. Gross. Scene reconstruction from high spatio-angular resolution light fields. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*, 32(4):73:1–73:12, 2013. 4

[10] M. Levoy and P. Hanrahan. Light field rendering. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 31–42, New York, NY, USA, 1996. ACM. 3

[11] C.-K. Liang and R. Ramamoorthi. A light transport framework for lenslet light field cameras. *ACM Transactions on Graphics (TOG)*, 2015. 4

[12] D. Lischinski and A. Rappoport. Image-based rendering for non-diffuse synthetic scenes. In *Rendering Techniques? 98*, pages 301–314. Springer, 1998. 4

[13] Lytro. *The Lytro Camera*. http://lytro.com. 4

[14] M. McGuire, W. Matusik, H. Pfister, J. F. Hughes, and F. Durand. Defocus video matting. *ACM Trans. Graph.*, 24(3):567–576, 2005. 3

[15] T. Porter and T. Duff. Compositing digital images. *SIGGRAPH Comput. Graph.*, 18(3):253–259, Jan. 1984. 6

[16] S. Pujades, F. Devernay, and B. Goldluecke. Bayesian view synthesis and image-based rendering principles. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3906–3913, June 2014. 4

[17] J. Shade, S. Gortler, L.-w. He, and R. Szeliski. Layered depth images. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 231–242, New York, NY, USA, 1998. ACM. 4

[18] M. W. Tao, S. Hadap, J. Malik, and R. Ramamoorthi. Depth from combining defocus and correspondence using light-field cameras. In *International Conference on Computer Vision (ICCV)*, Dec. 2013. 4

[19] Y. Tsin, S. B. Kang, and R. Szeliski. Stereo matching with linear superposition of layers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):290–301, February 2006. 4

[20] K. Vaidyanathan, J. Munkberg, Petrik, Clarberg, and M. Salvi. Layered light field reconstruction for defocus blur. *ACM Transactions on Graphics (TOG)*, 2013. 4

[21] V. Vaish, R. Szeliski, C. L. Zitnick, S. B. Kang, and M. Levoy. Reconstructing occluded surfaces using synthetic apertures: Stereo, focus and robust measures. In *In Proc. Computer Vision and Pattern Recognition*, page 2331, 2006. 3

[22] S. Wanner and B. Goldluecke. Reconstructing reflective and transparent surfaces from epipolar plane images. In *German Conference on Pattern Recognition (Proc. GCPR, oral presentation)*, 2013. 2, 4

[23] S. Wanner and B. Goldluecke. Variational light field analysis for disparity estimation and super-resolution. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(3):606–619, March 2014. 4

[24] Y. Wexler, A. W. Fitzgibbon, and A. Zisserman. Bayesian estimation of layers from multiple images. In *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*, volume 3, pages 487–501. Springer-Verlag, 2002. 3

[25] K. Zheng, S. B. Kang, M. Cohen, and R. Szeliski. Layered depth panoramas. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, June 2007. 4

[26] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 600–608, New York, NY, USA, 2004. ACM. 4