Masking Light Fields to Remove Partial Occlusion

Scott McCloskey
Honeywell ACS Labs
Golden Valley, MN 55422
Email: scott.mccloskey@honeywell.com

***Abstract*—We address partial occlusion due to objects close to a microlens-based light field camera. Partial occlusion degrades the quality of the image, and may obscure important details of the background scene. In order to remove the effects of partial occlusion post-capture, previous methods with traditional cameras have required the photographer to capture multiple, precisely registered, images under different settings. The use of a light field camera eliminates this requirement, as the camera simultaneously captures multiple views of the scene, making it possible to remove partial occlusion from a single image captured by a hand-held camera. Relative to past approaches for light field completion, we show significantly better performance for the small viewpoint changes inherent to a handheld light field camera, and avoid the need for time-domain data for occlusion estimation.**

I. Introduction

When capturing a landscape through a fence, proper composition can be difficult. If the spacing between fence posts is smaller than the camera's entry pupil, it may be impossible to record the distant scene without attenuation. Beyond reducing an image's aesthetic quality, its utility for forensic purposes may also be reduced. Surveillance cameras, for instance, may accumulate paint on protective covers that creates a blind spot over part of the scene. Indeed, the placement of surveillance cameras is restricted by the need to avoid nearby occlusions.

This need not be the case. It has long been noted that large apertures can 'see through' thin occluding objects. When focused on a distant scene, the large aperture induces a blur disc at the occluder's depth which is larger than the object, meaning that some rays of light from the background can still reach the entrance aperture of the lens and ultimately contribute to the exposure. While there are several methods which to this, they are impractical for hand-held capture because they require multiple images with specific settings.

While multiple images are needed to remove partial occlusion from traditional images, light fields captured during a single exposure provided the data needed for removal. The key feature of a light field camera is that it captures intensity as a function of incident angle at the focal plane, allowing us to render an image equivalent to one captured with custom, per-pixel apertures designed to block the foreground occluder. While light field cameras have been available to high-end users for some time, the release in 2012 of the Lytro camera has made light field imaging more wide-spread. However, previous methods for removing partial occlusion from light fields can't be applied to hand-held Lytro images because they estimate occlusions from temporal information or use computational methods which are better suited for large, stationary camera arrays. We demonstrate partial occlusion removal with real images from a hand-held Lytro camera.

II. Related Work

While the basic light field concept may go back as far as Leonardo da Vinci, modern methods were introduced by Levoy and Hanrahan [1] and Gortler et al. [2]. In his thesis, Ng [3] reviews these and other works, and describes the design and use of the microlens-based light field camera which would become the Lytro. While the microlens implementation is preferred for consumer light field photography, much of the past light field research has used data captured by large camera arrays. Unfortunately, methods for removing partial occlusion do not translate from the camera array implementation to a hand-held, microlens-based implementation. In particular, the method of Wilburn et al. [4] identifies occluded pixels in the light field by capturing a video and designating as occluded any pixel whose intensity is constant over time. This assumes that both it is possible to capture multiple frames from a stationary camera, and that all objects in the background move; neither assumption holds in the cases we consider here. The light field completion work of Yatziv et al. [5] uses a median filter on aligned images which, as we demonstrate, does not perform well on the relatively small apertures used in the Lytro camera. Of the various approaches demonstrated for removing partial occlusion from hand-held images, the best fit is probably coded aperture [6], [7]. However, a single image captured through a coded aperture must balance the fundamental trade-off between depth estimation and deblurring performance [8], whereas a light field does not.

Many methods for the removal of blurred foreground occluders have used conventional cameras, whose lack of angular resolution require that multiple images be captured. Favaro and Soatto [9] demonstrate a multiple image method which reconstructs multiple layers of an artificially textured scene. McCloskey et al. [10] demonstrate a single image method for attenuating occluders, but does not address thin occluders. They also present a method [11] which relaxes the need for complete occlusion, but which uses a second input image taken with a different aperture. Gu et al. [12] provides an iterative method using three images with different apertures to resolve the background. Yamashita et al. [13] remove blurred fences from an image using information from two additional images: a flash/no-flash pair in which the fence is well-focused, so that it can be composited using flash matting [14]. While multiple image methods can produce nice results, they are impractical for hand-held image capture, as small camera motions result in divergent parallax motions of the foreground and background layers. Automatic single image methods are limited to texture in-painting, e.g. Park et al.'s de-fencing [15], which presents a detection method for regular, symmetric lattices.

In contrast to previous work, we present a single image

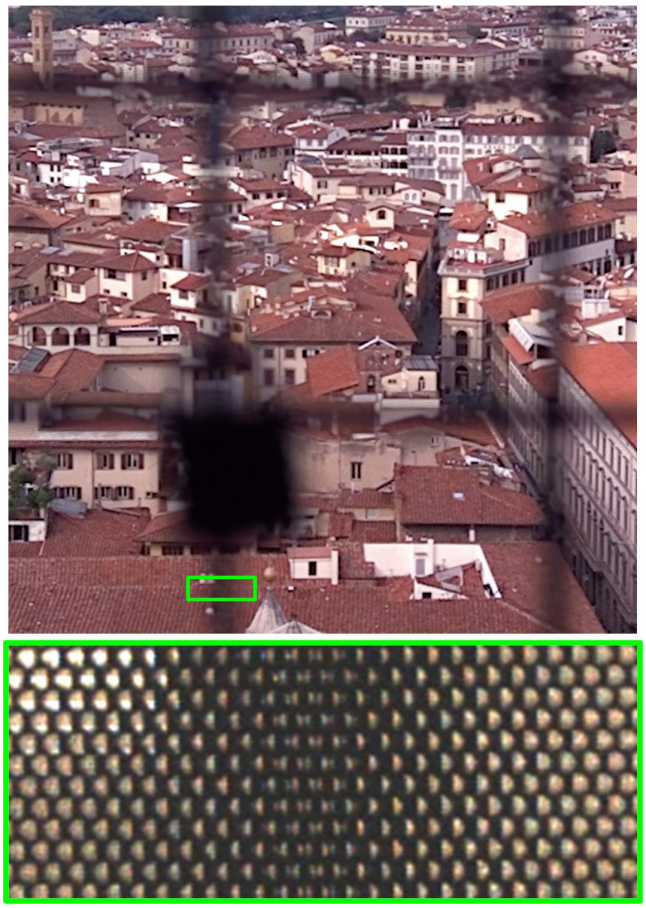

Fig. 1. (Top) A light field rendering with occlusion due to an attenuating foreground object (a metal grating). (Bottom) A part of the Lytro sensor image from the scene region outlined in green.

method suitable to hand-held image capture, to remove contributions from foreground occluders which both attenuate the background and contribute significant radiance. We do this by automatically detecting occluded rays at each microlens location of a light field camera, and we mask them out when rendering a sharply-focused image of the background.

III. Masking of Foreground Occluders

Fig. 1 shows an image of a cityscape taken through a metal grating to which someone has attached a padlock. This image demonstrates both partial occlusion (the darkened linear regions), where the background can still be seen, and complete occlusion (the black rectangle of the lock) where it can't. Because the grating is poorly lit, it attenuates light without contributing radiance to the image. While the image is one rendering of it, the light field structure contains more information than is shown in this rendering. The green inset corresponds to the outlined region of the scene, and shows the image recorded on the Lytro CMOS sensor, which is captured through a hexagonal lenslet array. The far left and right regions of the inset show that, away from the occluder, the lenslet produces a circular spot from background radiance. Moving toward the center of the image, the vertical occluder blocks an increasing proportion of the incident lighting directions. Even in the center, though, the occluder does not completely block

Fig. 2. The central 9 sub-aperture images of our illustrative example. While they appear quite similar, the foreground region (the black pixels) shift by about 7 pixels between neighboring images. Note: these images images look low quality because they are from raw sensor values; they have not been transformed to SRGB space, tone-mapped by the camera response function, or white balanced.

any of the lenslets, meaning that the camera captures some rays originating from all parts of the background in this region.

Fig. 1's inset also provides an intuitive illustration of how the contribution of the foreground occluder could be removed when rendering an image. With respect to the occluder's role contributing illumination to the sensor, we must identify and mask out that illumination from the light field. With respect to the occluder's role as an attenuator of background light, identifying the parts of the light field which are blocked also tells us the fraction of incoming rays are occluded by the foreground. Then, for those image pixels rendered using less than the full complement light field samples (due to the masking), amplification can be applied to the intensity rendered from the background radiance. So, the problem of rendering a background image without the attenuation or contribution of the foreground object is reduced to identifying those parts of the light field which represent the foreground occluder.

IV. Parallax for Foreground Estimation

While the raw sensor image shown in Fig. 1 is one illustration of the captured light field, it is more useful (for our purposes) to represent the light field as a collection of *sub-aperture images*, as defined in [3]. In this representation, the pixels from the raw sensor image are re-arranged to create an array of images, with the pixels in each coming from the same position under every micro-lens. Each sub-aperture image can be thought of as capturing a traditional image of the light passing through only a small sub-aperture region of the camera's lens. Given the spot size of the Lytro camera's microlenses, the raw image can be re-arranged into a 9-by-9 array of such images; the central 3-by-3 region is shown in Fig. 2. While these images are quite similar, they each have

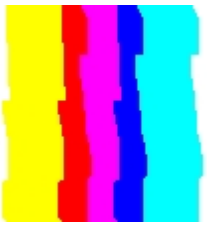

Fig. 3. A false-color image illustrating foreground parallax within the inset region of Fig. 1. The red channel indicates the foreground and background regions (intensities of 0 and 255, respectively) of the left-most sub-aperture image. The green/blue channels show the same for the central/right-most sub-aperture images, respectively. The lack of black pixels illustrates that there is no complete occlusion within this region of the image.

a unique optical center within the camera's aperture plane, so the same real-world object will appear in different locations of each sub-aperture image. That is, the sub-aperture images provide parallax on the captured scene: foreground objects (such as the grating) have relatively higher displacements between adjacent sub-aperture images than background objects (the buildings). The displacement of the foreground region from Fig. 1's inset is illustrated in false color by Fig. 3.

In order to estimate the foreground and background regions of the light field, then, we use parallax between the sub-aperture images. Foreground regions are characterized by relatively higher displacements between sub-aperture images, and background regions by small displacements.

V. Method

This section provides details of our method to estimate the mask, and then to render a focused image of the background without the contribution or attenuation of the occluder. We also discuss implementation details of how the Lytro camera is used. Throughout, we use a simplified version of the notation from [3], referring to sub-aperture images as $L^{(u,v)}$ for $u, v \in \{-4, -3, ...4\}$. We also refer to $L^{(0,0)}$ as the central sub-aperture image since it represents the rays passing through the center of the aperture.

Fig. 4 shows the steps of our method, illustrated using our example image. First, the lenslet image is converted to the sub-aperture image representation of the light field. Next, the foreground mask is estimated for the central sub-aperture image, using graph cuts for energy minimization, and is then shifted to generate masks for the other sub-aperture images. Then, the masks are used to determine an amplification map needed to counter the attenuating effect of the foreground. Finally, the amplification map, foreground masks, and sub-aperture images are combined to render a clean background image. For reference, Fig. 4 also shows a refocusing applied to the same image *without* masking out foreground occlusion.

A. Foreground Estimation

From the sub-aperture images $L^{(u,v)}$, we estimate a set of binary masks $\mathcal{M}^{(u,v)}$ indicating foreground regions in those images using parallax as a cue. We assume that the scene consists of a distant background which lies in an equi-focal plane, and a nearby object. Note that our assumption that the background is a *equi-focal plane* does not require that it be an *equal-depth plane*, as a distant focus plane may include a wide range of depths, such as the cityscape (ranging from a few hundred meters to several kilometers) shown in Fig. 1. With this assumption, areas of the background will shift uniformly between sub-aperture images; we refer to this shift as $\delta_b^{(u,v)}$, and note that shifts are linear with respect to sub-aperture image indices u and v, i.e.

$$\delta_b^{(2u,2v)} = 2\delta_b^{(u,v)}. \tag{1}$$

We estimate the background shift using SIFT features [16] extracted from $L^{(u,0)}\ \forall\ u$, the sub-aperture images with $v = 0$. The features from each non-central sub-aperture image are matched to those from the central sub-aperture image. Since the displacement between sub-aperture images with $v = 0$ must be horizontal, we remove erroneous matches by discarding those with a significant vertical shift (5 or more pixels, in our experiments). Because of under-illumination and defocus of the foreground (which is *not* assumed to lie in a focal plane), foreground regions generally lack the features necessary to be matched. As a result, the distribution of shifts represents the background only. We estimate $\delta_b^{(1,1)}$ as the median of the distribution, and extrapolate to find the other shifts using eq. 1.

We estimate the foreground/background label of each pixel in the central sub-aperture image using graph cuts which, because there are only two labels, is known to produce the globally-optimal solution regardless of the initial condition. We specify per-pixel costs of assigning the foreground or background label. In plain English, this cost is the maximum difference between the central sub-aperture image's pixel intensity and the intensities of the other sub-aperture images' pixels that would correspond to it *if it were a background pixel.* Mathematically, our cost of assigning a pixel to the background is

$$C_b^{(0,0)}(x) = \max_{u,v} \|L^{(0,0)}(x) - V(u,v)L^{(u,v)}(x+\delta_b^{(u,v)})\|. \tag{2}$$

Here, $V(u,v) \geq 1$ is a multiplicative term to account for vignetting of the microlenses. Because the central sub-aperture image is made up of pixels on each microlens's optical axis, it is brighter than other sub-aperture images. The brightness of the other sub-aperture images decreases with $\|u, v\|$, so V is proportional it. We determine V empirically by calibration, as described in Sec. VI.

Since the foreground generally lacks the features needed to establish its shift, we can't reliably estimate $\delta_f^{(u,v)}$ and the analogous cost function to eq. 2 can't be used. Instead, since the foreground appears as a region of nearly uniform intensity, we compute the cost of a foreground label at x as the intensity difference between it and an estimate of the foreground's intensity.

$$C_f^{(0,0)}(x) = \|L^{(0,0)}(x) - I_f\|, where \tag{3}$$

$$I_f = \frac{\sum_x \Omega(x) L^{(0,0)}(x)}{\sum_x \Omega(x)}, \tag{4}$$

and Ω is an indicator which is one for pixels above the 99th percentile of C_b and zero elsewhere.

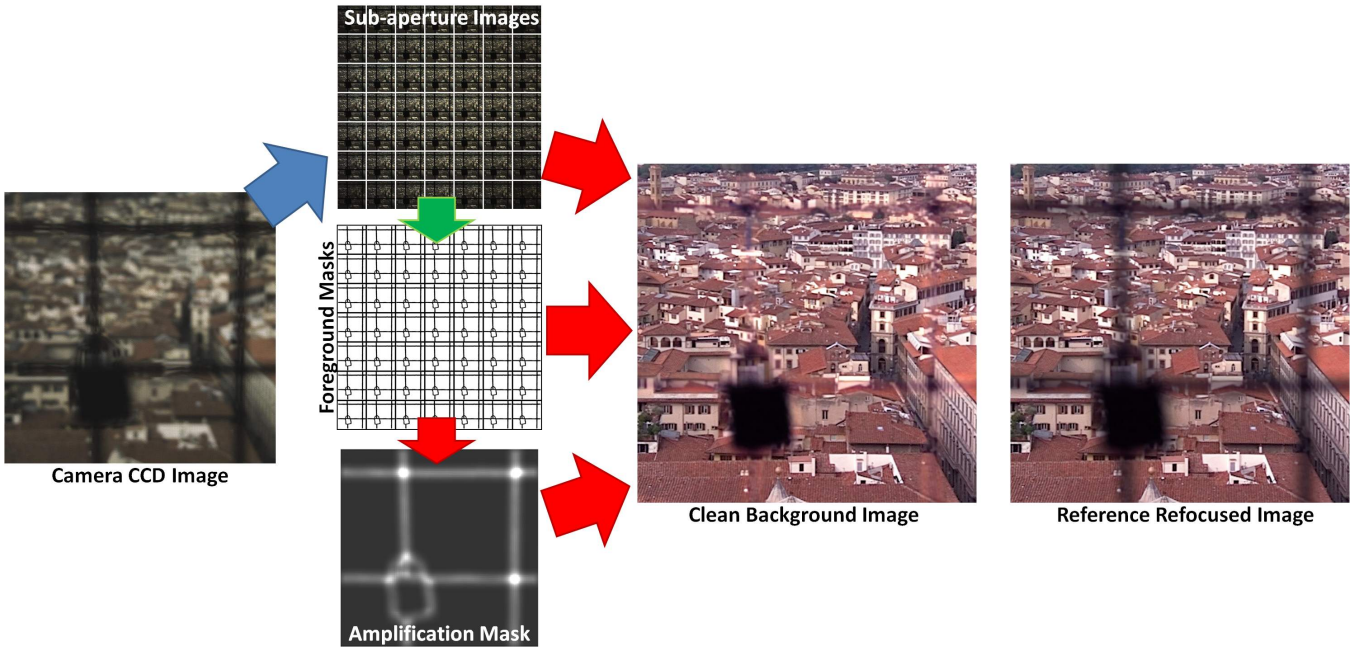

Fig. 4. **Method overview.** Starting from the raw sensor image extracted from the Lytro image file format, we generate the sub-aperture image representation of the light field (blue arrow; details in Sec. VI). From this, we estimate the foreground and background regions using parallax (green arrow; Sec. V-A). From the foreground masks, we estimate the amount of attenuation at each pixel, and then render the unoccluded regions of the sub-aperture images to generate a clean, focused background image (red arrows; Sec. V-B). For comparison, a rendering is shown without masking.

Mask Propagation Once the foreground mask $\mathcal{M}^{(0,0)}$ is estimated by graph cuts for the central sub-aperture image, it is propagated to the other sub-aperture images. We search for the shift between $L^{(0,0)}$ and $L^{(0,1)}$ that minimizes the RMS difference in the masked pixels, i.e.

$$\delta_f^{(0,1)} = argmin_{\hat{\delta}} \sum_x (L^{(0,0)}(x)\mathcal{M}^{(0,0)}(x) - L^{(0,1)}(x+\hat{\delta})\mathcal{M}^{(0,1)}(x+\hat{\delta}))^2. \quad (5)$$

This value is then extrapolated to find the remaining $\delta_f^{(u,v)}$ using eq. 1. Once the full set of shifts are known, the shifted masks are computed as

$$\mathcal{M}^{(u,v)}(x) = \mathcal{M}^{(0,0)}(x + \delta_f^{(u,v)}). \quad (6)$$

B. Rendering a Clean Background

In order to render a focused background while removing the contribution of the foreground occluder, we modify the shift-and-add re-focusing method from [3]. Ng's shift-and-add method renders the light field to an image focused on particular object by shifting the sub-aperture images so that they are all aligned with respect to the object. To focus on the background, for example, the rendered image

$$I(x) = \sum_{u,v} L^{(u,v)}(x + \delta_b^{(u,v)}). \quad (7)$$

Given a set of binary masks $\mathcal{M}^{(u,v)}$ indicating foreground regions, and the background's displacement $\delta_b^{(u,v)}$, we render an unoccluded ('clean') background image

$$\hat{I}(x) = A(x) \sum_{u,v} (1 - \mathcal{M}^{(u,v)}(x + \delta_b^{(u,v)}))L^{(u,v)}(x + \delta_b^{(u,v)}). \quad (8)$$

Terms of the summation where a pixel might get a contribution from the foreground in eq. 7 are exactly those where $\mathcal{M}^{(u,v)} = 1$, causing cancellation in eq. 8. The amplification term A accounts for the loss of signal due to the occlusion within sub-aperture images. In order to account for the vignetting of the microlenses causing the intensity of sub-aperture images decreases with $\|(u,v)\|$,

$$A(x) = \frac{\sum_{u,v}(1 - \mathcal{M}^{(u,v)}(x + \delta_b^{(u,v)}))1/V(u,v)}{\sum_{u,v} 1/V(u,v)}, \quad (9)$$

where V, again, quantifies microlens vignetting. After refocusing, we apply the camera response function and sensor RGB to SRGB color correction specified in image metadata.

VI. Implementation and Experiments

Though Lytro has pledged to release an API[1], they have yet to do so. As such, it was necessary to recreate parts of the Lytro processing chain in order to test our method. In order to extract the raw sensor image from the Lytro files, we use the lfpsplitter tool [17], and demosaic the Bayer BGGR array using Matlab's `demosaic` function. Constructing the sub-aperture images from the demosaiced sensor image turned out to be surprisingly difficult despite the availability of sensor metadata in the LFP files. Because the microlens array is

[1]Per Kurt Akeley at the Int'l Conf. on Computational Photography 2012.

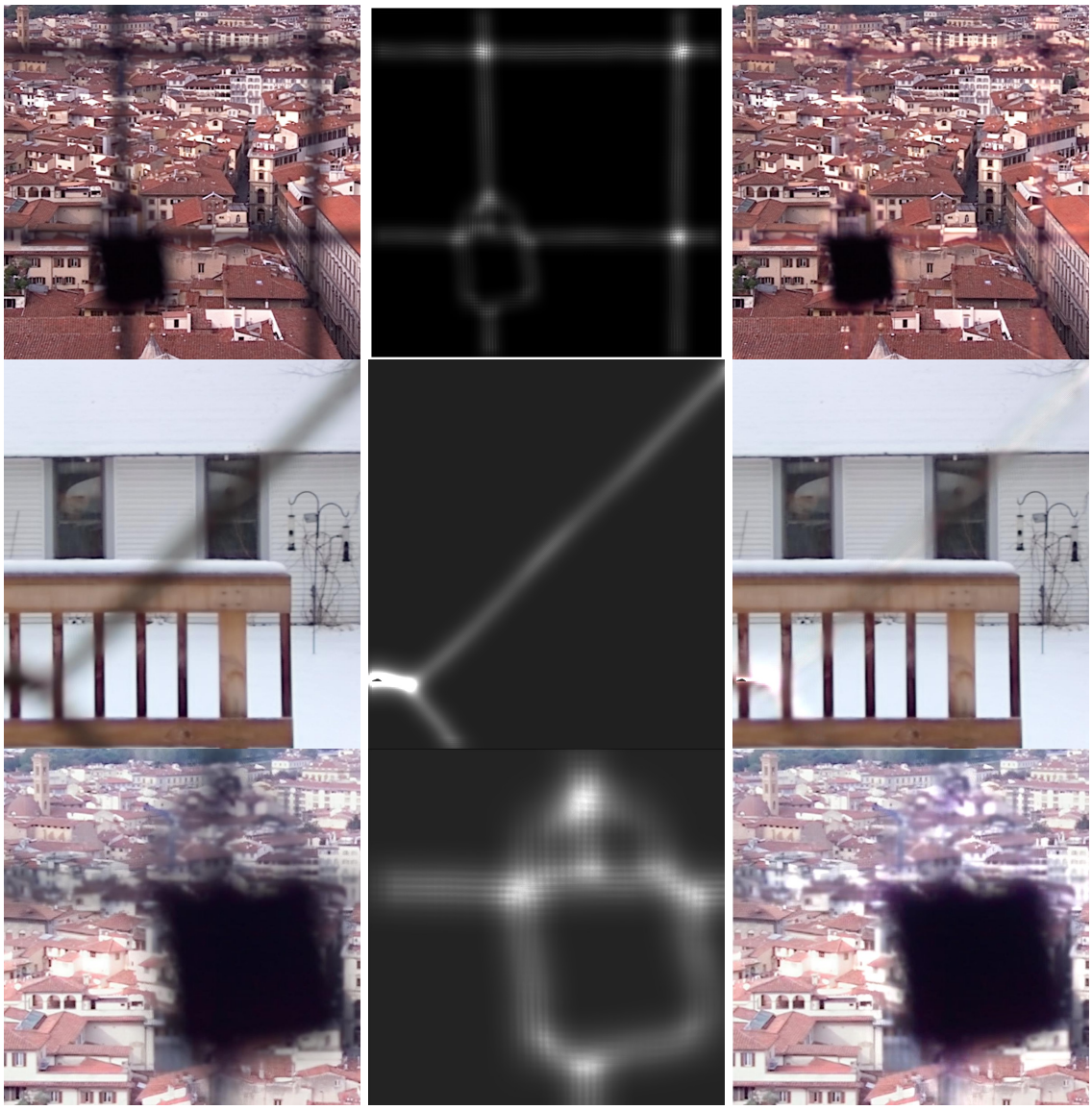

Fig. 5. Results for attenuating occluders. The left column shows, for reference, a rendering of the light field for sharp focus on the background, with partial occlusion. The center column shows the degree of amplification needed to cancel the attenuation due to the occluder. The right column shows the output of our method, with partial occlusion removed and a clearer rendition of the background.

rotated with respect to the sensor, because the microlens pitch is a non-integer multiple of the pixel pitch, and because the microlenses are on a hexagonal array rather than a rectangular one, finding the spot centers behind the microlenses is non-trivial. After failing to recover the spot centers directly from the metadata, we instead calibrate them by capturing several images of a uniformly-lit white wall. After averaging the demosaiced sensor images (to reduce noise), we empirically found the brightest pixel behind each microlens. The locations of the brightest pixels were then saved in order to indicate the locations of the (hexagonal) sub-aperture image $L^{(0,0)}$. Other sub-aperture images are found by displacements from these locations, and each sub-aperture image is interpolated to a rectangular pixel grid using Matlab's `TriScatteredInterp` functions. Using these displacements, we find the sub-aperture images $L_w^{(u,v)}$ for the white wall image in order to characterize the microlens vignetting. Specifically,

$$V(u,v) = \frac{\sum_x L_w^{(0,0)}(x)}{\sum_x L_w^{(u,v)}(x)}. \tag{10}$$

Fig. 5 shows the results of our method applied to several images with attenuating occluders (fences and metal gratings). The reconstructed contrast uniformity of the images are much improved by our method though, due to amplification of signal from a decreasing number of sensor pixels, rendered pixels in the regions of high partial occlusion exhibit increased noise. While this is fundamental to all methods that remove partial occlusion, the effect is less here than for methods that work with traditional images, since the light field rendering step adds the intensity from several sensor pixels, reducing noise.

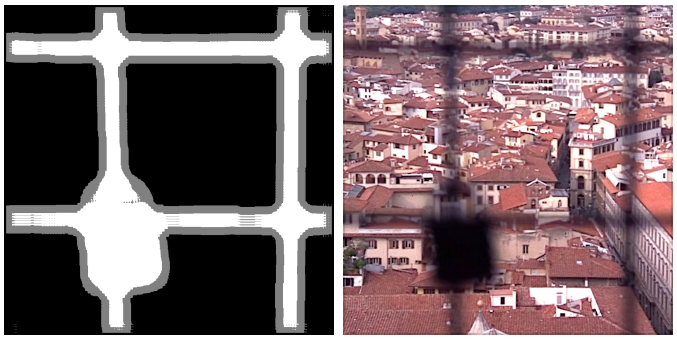

Fig. 6. Using the median of registered images [5] improves image quality when half of the views are unoccluded (gray pixels), but does nothing when more than half are occluded (white). The median filter result on our lock example (right) is little improved from normal refocusing.

A. Median Filter Comparison

While the median filter approach for light field completion [5] was designed for camera arrays, it can be adapted to lenslet-based capture by accounting for the vignetting effect, as

$$\hat{I}_{med}(x) = \text{median}_{u,v}\{V(u,v)L^{(u,v)}(x + \delta_b^{(u,v)})\}. \quad (11)$$

Conceptually, the median filter approach should produce a good estimate of the background for those pixels that are occluded in fewer than half of the sub-aperture images. As we show in Fig. 6, this provides little benefit for our images. This is because the aperture of the Lytro is significantly smaller than the camera arrays used previously, so foreground shifts are substantially smaller and most partially occluded pixels are occluded in more than half of the sub-aperture images.

VII. Assumptions and Limitations

While our method to render images free of partial occlusion can be used on single images from a hand-held camera, and thus requires less information than previous methods, certain assumptions are necessary. First, that the background lie in a focal plane, though we note that it may have significant depth variation within it (e.g., the landscapes in our examples). Our method is also limited to cases where the shift of the background layer between sub-aperture images is non-zero. There are cases where the background depth results in no shift across sub-aperture images, in which case the cost functionals used for segmentation are all zero. While differences in pixel intensity may help separate the foreground and background layers in such cases, this ability will be limited.

More fundamentally, our method to remove *partial* occlusion does not address regions of *complete* occlusion. In our thoroughgoing example, the lock attached to the metal grating is larger than the front aperture, and thus there are image regions which can't be recovered. Texture in-painting may be used in these regions to improve aesthetics, as in [5], but the true scene cannot be recovered.

As with all lenslet-based light field capture, Lytro's angular resolution is captured at the expense of spatial resolution. This is a fundamental tradeoff of single-shot light field acquisition, and is needed to address the hand-held case. Given the diminishing utility of high-resolution sensors with moderately-priced optics, we view the trade of spatial resolution for applicability to a hand-held camera as one worth making.

VIII. Conclusions

We have presented a method to automatically remove partial occlusion from a light field captured using a hand-held, microlens-based camera. By removing the requirement that the photographer capture several images using different settings without moving the camera, or that a light field video be captured, we enable everyday photographers to improve the quality of images taken through intermediate layers. The method is based on a modification of light field rendering which defines a custom aperture mask for each microlens, with the mask designed to block light from the foreground occluder. We estimate the aperture mask by exploiting parallax causing different shifts of objects in the foreground and background.

Having demonstrated a method to remove the contribution of opaque occluding objects, we plan to extend our method to translucent occluders such as smudges or dust on the lens. While such cases are harder, we believe that they can be addressed by realizing that the translucent occluder must lie on a known focal plane (the lens surface).

References

[1] M. Levoy and P. Hanrahan, "Light field rendering," *ACM Trans. on Graphics (Proc. of SIGGRAPH)*, 1996.

[2] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," *ACM Trans. on Graphics (Proc. of SIGGRAPH)*, pp. 43–54, 1996.

[3] R. Ng, "Digital light field photography," Ph.D. dissertation, Stanford University, Stanford, CA, USA, 2006.

[4] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy, "High performance imaging using large camera arrays," *ACM Trans. on Graphics (Proc. of SIGGRAPH)*, Jul. 2005.

[5] L. Yatziv, G. Sapiro, and M. Levoy, "Lightfield completion," in *Int'l Conf. on Image Processing*, 2004.

[6] A. Veeraraghavan, R. Raskar, A. Agrawal, A. Mohan, and J. Tumblin, "Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing," *ACM Trans. on Graphics (Proc. of SIGGRAPH)*, vol. 26, 2007.

[7] A. Levin, R. Fergus, R. Fergus, F. Durand, and W. T. Freeman, "Image and depth from a conventional camera with a coded aperture," in *SIGGRAPH*, 2007.

[8] C. Zhou, S. Lin, and S. Nayar, "Coded Aperture Pairs for Depth from Defocus and Defocus Deblurring," *Int'l J. of Computer Vision*, vol. 93, no. 1, p. 53, May 2011.

[9] P. Favaro and S. Soatto, "Seeing beyond occlusions (and other marvels of a finite lens aperture)," in *CVPR*, 2003.

[10] S. McCloskey, M. Langer, and K. Siddiqi, "Removal of partial occlusion from single images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 647–654, 2011.

[11] ——, "Removing partial occlusion from blutted thin occluders," in *Int'l Conf. on Pattern Recognition*, 2010.

[12] J. Gu, R. Ramamoorthi, P. Belhumeur, and S. Nayar, "Removing Image Artifacts Due to Dirty Camera Lenses and Thin Occluders," *Proc. of SIGGRAPH Asia*, Dec 2009.

[13] A. Yamashita, A. Matsui, and T. Kaneko, "Fence removal from multi-focus images," in *Int'l Conf. on Pattern Recognition*, 2010.

[14] J. Sun, Y. Li, S. B. Kang, and H.-Y. Shum, "Flash matting," *ACM Trans. on Graphics (Proc. of SIGGRAPH)*, Jul. 2006.

[15] M. Park, K. Brocklehurst, R. T. Collins, and Y. Liu, "Image de-fencing revisited," in *Asian Conf. on Computer Vision*, 2011.

[16] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int'l J. of Computer Vision*, vol. 60, no. 2, Nov. 2004. [Online]. Available: http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94

[17] N. Patel, "lfpsplitter," https://github.com/nrpatel/lfptools.