Using Unknown Occluders to Recover Hidden Scenes

Adam B. Yedidia[1] Manel Baradad[1] Christos Thrampoulidis[2] William T. Freeman[1,3]
Gregory W. Wornell[1]
{adamy, mbaradad, billf, gww}@mit.edu, cthrampo@ucsb.edu
[1] Massachusetts Institute of Technology, [2] U.C. Santa Barbara, [3] Google Research

Abstract

We consider the challenging problem of inferring a hidden moving scene from faint shadows cast on a diffuse surface. Recent work in passive non-line-of-sight (NLoS) imaging has shown that the presence of occluding objects in between the scene and the diffuse surface significantly improves the conditioning of the problem [2]. However, that work assumes that the shape of the occluder is known a priori. In this paper, we relax this often impractical assumption, extending the range of applications for passive occluder-based NLoS imaging systems.

We formulate the task of jointly recovering the unknown scene and unknown occluder as a blind deconvolution problem, for which we propose a simple but effective two-step algorithm. At the first step, the algorithm exploits motion in the scene in order to obtain an estimate of the occluder. In particular, it exploits the fact that motion in realistic scenes is typically sparse. The second step is more standard: using regularization, we deconvolve by the occluder estimate to solve for the hidden scene.

We demonstrate the effectiveness of our method with simulations and experiments in a variety of settings.

1. Introduction

Imaging scenes that are not directly visible, also called non-line-of-sight (NLoS) imaging, is a difficult and often ill-posed problem. Recently, it has become an area of active study, with broad applications including search-and-rescue, anti-terrorism, and traffic [3, 4, 2].

Past methods that rely on human-visible light to image hidden scenes can be divided into one of two categories. *Active* methods introduce light into the scene and make use of known or measured properties of the introduced light, such as time of return, to image the hidden scene [28, 23, 27, 10]. *Passive* methods, on the other hand, rely exclusively on ambient light from the scene, such as secondary reflections on the observation plane, to infer the contents of the hidden scene [25, 4, 2]. Both of these types of methods generally presume that there is something directly visible to both the observer and the hidden scene (see Fig. 1). In this work, we refer to this visible area as the *observation plane*.

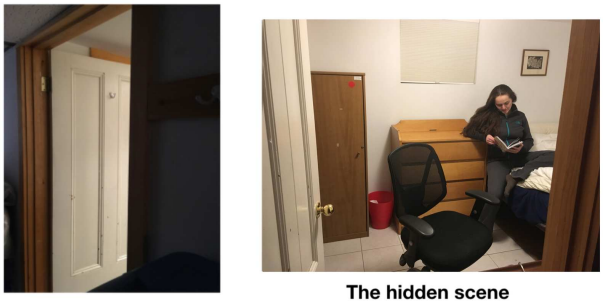

Figure 1: This figure shows a hypothetical scenario to which our method could be applied. On the left, we can see a door, but we don't know anything about what's in the room it leads to (shown on the right). Using the method we present in this paper, an onlooker could try to reconstruct an image of the room by observing the door, perhaps by using the occlusion provided by the chair and the motion provided by the person. This is true even though neither the chair nor the person is visible to the onlooker!

Passive methods suffer from the fact that in real-world settings, only a two-dimensional surface can be observed (see e.g. Fig. 1), but the scene producing those observations is three-dimensional. Hence, the problem is inherently ill-posed. Past methods have resolved this issue by either assuming the scene lies on a lower-dimensional manifold, thereby only reconstructing only a lower-dimensional projection of the scene [25, 4], or making use of a strong spatial prior over realistic scenes when reconstructing [2]. Our method falls into the former category, as we assume that both the scene and occluder lie on parallel, flat planes. This allows us to model the shadows cast on the observation planes as a simple convolution of these two planes.

Although there has by now been plenty of previous work demonstrating that it is possible to use the presence of an oc-

cluder to infer the structure of a hidden scene [25, 4, 24, 2], this work, to our knowledge, is the first to do so in a *blind* manner, meaning that we know nothing *a priori* about the structure of the occluder. Past work that exploits occlusion either uses scene calibration to get a precise picture of the occluder before the system can work [24, 2, 21] or is limited to situations in which the occluder has some basic, common shape, like a pinhole, pinspeck, or edge [25, 4]. The blind nature of this problem compounds the already daunting challenge of non-line-of-sight imaging. However, we hope that this will make our method widely applicable in situations where occluders are complex but pre-calibration is not an option, such as traffic or search and rescue [3].

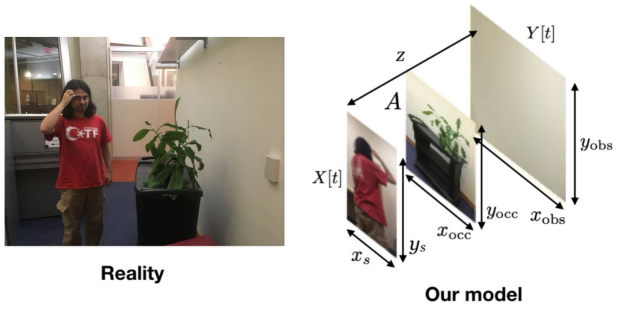

Figure 2: Left: a real-world scenario with a moving scene, an occluder, and an observation wall. Right: our model of the scenario.

2. Background

This work draws inspiration from past work belonging to two broad categories: the first is NLoS imaging, particularly occlusion-based NLoS imaging, and the second is past work in blind deconvolution. To our knowledge, this work is the first to synthesize these two well-studied areas of research into an algorithm that does something novel: getting a two-dimensional view of a hidden scene, with only minimal assumptions about the hidden scene and unknown occluder.

2.1. Non-line-of-sight imaging

2.1.1 Active methods

Past work in active NLoS imaging has demonstrated the possibility of resolving structure in hidden scenes using time-of-flight (ToF) cameras [28, 23, 13, 10], including counting people [27], inferring size and motion of objects [12, 19, 9], and object tracking in 2D space [14]. In particular, Pandharkar et al. [19] used ToF cameras to resolve moving scenes in a partially uncalibrated setting, and Velten et al. [26] used them to recover 3D structure from behind corners. Thrampoulidis et al. [24] showed that occluder-based imaging could be applied to the active setting, as well.

2.1.2 Passive methods

Recently there has been a surge of interest in occluder-based imaging methods. Torralba and Freeman [25] were among the first to notice that objects the environment could form "accidental" cameras, making resolving hidden structure simpler. In particular, they used the fact that many common objects behave approximately like pinspecks or pinholes. Additionally, in 2019, Saunders et al. [21] used pinspeck occluders to resolve 2D scenes.

Occluder-based methods have also been used to see around corners [4] and infer light fields [2]. Coded-aperture photography [16, 31, 20] can also be thought of as a kind of occluder-based imaging.

2.2. Blind deconvolution

Past work in the field of blind deconvolution has largely focused on image deblurring [17, 6, 15, 5], although there has also been work applied to communication [7], and control systems [1]. Recently, interest in blind deconvolution for image deblurring has been revived by Fergus et al. in [8].

Past approaches to blind deconvolution have generally made use of local search methods, including total variation minimization [6] and alternating projections [29]. There has also been past work in multi-frame blind deconvolution with applications to deblurring astronomical images [22, 11, 18].

3. Scenario

3.1. Setup

Our model of the scenario consists of three elements: a hidden moving scene, an occluder, and the observation plane. We model each of these elements as parallel 2D planes. See Fig. 2 for an illustration.

The hidden scene is presumed to be a collection of diffuse reflectors, shining light uniformly in all directions and towards the occluder and observation plane. The hidden scene is also presumed to contain some motion. The unknown occluder is presumed to be a set of perfectly opaque objects lying on a common plane. We assume the hidden scene, unknown occluder, and observation planes to each be a substantial distance apart, relative to their sizes. This allows us to invoke paraxial imaging assumptions, like in [30, 4].

The observation plane is presumed to be perfectly Lambertian. In simulations, we also presume the observation plane to be white and uniform, and that all of the light reaching the observation plane comes from the scene; in exper-

iment, we use mean-subtraction to account for non-white, non-uniform observations with ambient "nuisance" light sources, a method also employed in other work (e.g. [4]). This allows us to apply our method to most realistic scenarios with minimal adaptations to the core algorithm. We explore the effect of other deviations from the idealized scenario we present here in Section 6.

3.2. Light Propagation

The assumptions we describe in Subsection 3.1 imply that translating a light source in the scene will correspond to a simple translation of the shadow it casts on the observation plane in the opposite direction. For a more detailed explanation of why that is, and how that model deviates from reality when those assumptions are violated, see Section 6.

We model the propagation of light through the system as a 2D convolution of the scene with the occluder. This follows from the fact that a translation of an impulse light source will simply translate the shadow cast by the occluder, and from the fact that the observed light can be modeled as a linear combination of light emanating from different sources in the scene. See e.g. [30], who use the same convolution-based model of light propagation that we do. In Section 6, we go into some detail on how robust this model is, and in Section 7 we present the results of experiments, including real-world experiments.

In the simulations presented in this paper, we assume that we see the full convolution of the scene and the occluder on the wall. If the scene is a plane of size $x_s \times y_s$ and the occluder a plane of size $x_{\text{occ}} \times y_{\text{occ}}$, this corresponds to an observation of $(x_s + 2x_{\text{occ}}) \times (y_s + 2y_{\text{occ}})$. However, in practical settings, it may not be possible to see the full convolution of the scene and occluder on the wall. In the experimental case, therefore, we express the size of the observed part of the wall as $x_{\text{obs}} \times y_{\text{obs}}$. It is easy to adapt our algorithm to the case when only part of the convolution between scene and occluder is visible, as explained in Sections 4 and 5. But of course, the larger x_{obs} and y_{obs} are, the more information about the hidden scene will be available, and the better the reconstructions will be.

4. Occluder Estimation

Our blind deconvolution algorithm consists of two steps. The first step estimates the 2D occluder from the observation movie and is the primary contribution of this paper. We describe this step in this section. In Section 5, we describe the more standard second step, which recovers the movie using the estimated occluder.

4.1. Preliminaries

Let $Y = \{Y[0], \ldots, Y[T]\}$ be the observed video, with each $Y[t]$ corresponding to a video frame. Let $\bar{Y} = \frac{1}{T}\sum_t Y[t]$ be the "average frame" of the video, i.e., the frame such that each of its pixels is equal to the temporal average of that pixel across the entire video. Also, consider: (a) the *mean-subtracted video* $Y_\mu = \{Y[0] - \bar{Y}, \ldots, Y[T] - \bar{Y}\}$, i.e., the video of differences from the mean of the original video; (b) the *difference video* $Y_D = \{Y[1] - Y[0], \ldots, Y[T] - Y[T-1]\}$, i.e., the discrete temporal derivative of the observed video. Similarly, let $X = \{X[0], \ldots, X[T]\}$ be the ground-truth video of the scene, and let X_μ and X_D be the mean-subtracted ground-truth video of the scene and the discrete temporal derivative of the ground-truth video, respectively. X_μ and X_D are defined relative to X in the same way as above.

At this point, note the subtle but important difference between the mean-subtracted video and the difference video, which will play different roles in the algorithm. We use the observed difference video to estimate the occluder, and we use the observed mean-subtracted video when reconstructing the moving scene. What makes the difference video preferable for occluder estimation is the fact that most realistic moving scenes have just a few moving objects in them; thus each frame of the difference video is sparse.[1]

Finally, we let A be the occluder. Each element of the occluder is either 0 or 1, with 1 being no occlusion, 0 being occlusion.

As explained in Section 3, we can express the observation as the convolution of the scene and the occluder[2]. Thus, for all $0 \leq t \leq T$,

$$Y[t] = A * X[t], \;\; Y_\mu[t] = A * X_\mu[t], \;\; Y_D[t] = A * X_D[t]. \tag{1}$$

Given Y (and by extension Y_μ and Y_D), our goal is to learn both X and A. We will exploit the fact that each of the $X_D[t]$ is sparse and the fact that A is binary-valued. Next, we describe an algorithm that uses $Y_D[t]$ to infer an estimate of A, which we denote $\hat{A}$.

4.2. Algorithm Description

Informally, we estimate the occluder by successively multiplying together randomly-chosen difference frames of the observation video with each other. Before doing so, we want to shift them such that their dot product is maximized.

[1] The sparsity of the difference video is necessary for our algorithm to work. Note, however that taking temporal derivatives amplifies the noise relative to the signal. Therefore, in situations in which the mean-subtracted ground-truth video is sparse, it is preferable to use the mean-subtracted observation video instead of the difference observation video for the task of occluder estimation. Sparse mean-subtracted ground-truth video would occur for example when most of the light in the scene is being emitted by a single source.

[2] In the case that the observation is in color, Equation 1 will be true for each color channel individually. Then, we can run the same algorithm as otherwise, but choosing at each step a single color channel of each difference frame.

We can efficiently compute the set of all possible dot products of two frames, up to shifts, by computing the correlation between the two frames. Thanks to the sparsity of the difference frames, the aggregated overlap between the random difference frames that we choose will likely correspond to the shape of the occluder.

The algorithm's pseudocode is given as Algorithm 1. Therein and onwards, we use superscripts (such as $X^{i,j}$) to denote a single pixel of an image or a single entry of a matrix, and single bars (such as $|X|$) to denote the elementwise absolute value of a matrix. Note that the occluder estimate $\hat{A}$ evolves over the course of the algorithm. For the reader's convenience, we provide a detailed illustration of the first two rounds of the algorithm in Fig. 3.

Our algorithm consists of three steps which we repeat until a maximum iteration count is reached. First, there is a pre-processing step, the goal of which is to select observed difference frames corresponding to sparse ground-truth difference frames. In particular, Algorithm 1 performs the preprocessing step by randomly selecting frames. We have empirically observed that this simple solution produces satisfactory results.

Next comes the alignment step. In the first iteration, we randomly select an absolute difference frame to be our first estimate of the occluder $\hat{A}_1$. At each iteration $k = 2, \ldots$ that follows, we treat $\hat{A}_{k-1}$ as a video frame which we align with a randomly selected new frame to obtain a refined estimate of the occluder $\hat{A}_k$.

In order to better understand the details of the alignment procedure and the reason why it yields an estimate of the occluder, it is instructive to consider the simple example of "ideally sparse" frames. Suppose we had a difference ground-truth frame that was a perfect impulse at (i_1, j_1), i.e., $X_D[1]^{i,j} = \delta(i - i_1, j - j_1)$, where δ is the 2D Kronecker-δ function. Then, clearly, $Y_D[1]$ is nothing but a shift of the occluder A by (i_1, j_1). In this ideal case, we immediately obtain a good picture of the occluder just by looking at a single difference observation frame. Unfortunately, in practice ground-truth video frames are only approximately sparse. We therefore model the difference observation frames as noisy shifts of the occluder. In particular, for two such frames let $Y_D[1]^{i,j} = A^{i-i_1,j-j_1} + n_1$ and $Y_D[2]^{i,j} = A^{i-i_2,j-j_2} + n_2$, where n_1 and n_2 denote noise. The goal of the alignment step is to create "aligned" versions of $Y_D[1]$ and $Y_D[2]$, which we will call $Z_D[1]$ and $Z_D[2]$, and for which:

$$Z_D[1]^{i,j} = Y_D[1]^{i,j} = A^{i-i_1,j-j_1} + \tilde{n}_1, \qquad (2)$$
$$Z_D[2]^{i,j} = Y_D[2]^{i-(i_2-i_1),j-(j_2-j_1)} = A^{i-i_1,j-j_1} + \tilde{n}_2.$$

This is achieved in Algorithm 1 by cross-correlating $Y_D[1]$ and $Y_D[2]$, finding where the max of the correlation occurs and appropriately shifting the original frames. This will approximately minimize the noise terms $\tilde{n}_1$ and $\tilde{n}_2$. See also Figure 3.

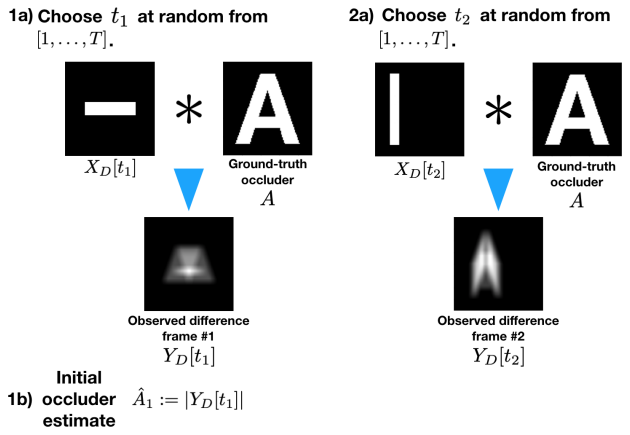

Figure 3: A worked example of initialization of Algorithm 1, followed by a single pass through the for-loop. Continued in Fig. 4.

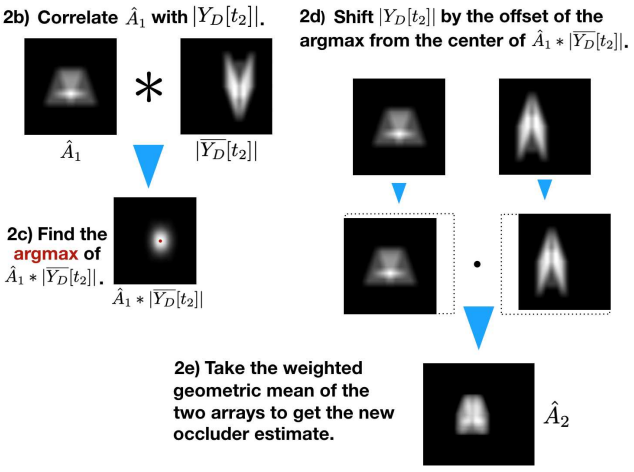

Figure 4: A worked example of Algorithm 1.

The goal of the third step is to reduce the noise terms in (2) and improve the estimate of the hidden A matrix. The simplest de-noising rule would be to return the average of $Z_D[1]$ and $Z_D[2]$. We have found instead that performing the average on the logarithms of the absolute values of the frames performs better. This explains the "geometric-mean" step in Algorithm 1.

Running the full occluder-estimation algorithm by sampling 100 frames takes a few minutes on a laptop.

4.3. Comparison to other methods

It is instructive to describe the differences between our application and that of most of the previous literature in blind deconvolution. Past work in blind deconvolu-

Algorithm 1 Our algorithm for estimating the occluder.

Set t to a random integer in $[0, T]$. Set $\hat{A}_1 = |Y_D[t]|$.
for k *in* $[2, \text{NumIter}]$ **do**
  Set t_k to a random integer in $[0, T]$.
  Compute $C = \hat{A}_{k-1} * |Y_D[t_k]|$.
  Find $(i, j) := \text{argmax}_{i,j} C^{i,j}$, where by convention we take $C^{0,0}$ to be the central element of $C^{i,j}$.
  Let $S_{i,j}(|Y_D[t_k]|)$ be $|Y_D[t_k]|$ shifted horizontally by i pixels and vertically by j pixels.
  $\hat{A}_k := (\hat{A}_{k-1})^{(k-1/k)} \cdot (|Y_D[t_k]|)^{(1/k)}$.
  ▷ In the line above, the superscripts denote elementwise exponentiation.
  Crop the zero-valued entries of the resulting $\hat{A}_k$ until it is the same shape as $\hat{A}_{k-1}$.
end

tion has largely focused on applications in image deblurring [17, 6, 15, 5]. Typically, this means that, given a single blurry image taken with a shaky camera, we would like to express the blurry image as the convolution of an unknown sharp image and an unknown blur kernel.

This problem differs from ours in three ways. First, unlike in our problem, one can assume that the unknown blur kernel is not only sparse but localized to a small region. Second, in our problem, we have additional information about the occluder: in particular, we assume it to be binary-valued. Finally, in our problem, we have many frames, each of which is a different (unknown) sparse kernel convolved with the occluder, which gives us much more information to work with.

The first difference means we have many more potential degrees of freedom to handle in our reconstruction algorithm; local search algorithms, used for deblurring in [6, 29], encounter difficulties when the potential size of the kernel is greatly increased. This makes it challenging to directly port blind-deconvolution algorithms used for image deblurring to our application.

Moreover, the many extra frames we have give us more information to work with. In particular, each frame shows us the occluder convolved by a different sparse kernel. This gives us many different "views" of the same occluder; it seems natural that as the length of the video goes to infinity, we should, in principle, be able to precisely characterize the shape of the occluder, even in the presence of arbitrary finite noise. How this intuition should extend to actual videos with a fixed number of frames is unclear, of course, but the nature of the problem (a fixed occluder with a non-fixed moving scene) lends itself naturally to an approach in which we estimate the occluder first, and then attempt deconvolution by the occluder estimate to recover the scene, rather than vice-versa.

Before settling on the method we used in this paper, we tried a variety of other methods, all of which failed. We tried a root-finding approach to the blind-deconvolution problem, a phase-retrieval-based approach (using ADMM), and we tried a simple gradient descent over the scene and the occluder jointly. For more details on these other approaches we tried, and why we believe they failed, please read our supplementary materials. We recommend that any reader hoping to improve upon our result should read the supplement to avoid repeating our mistakes.

5. Scene Reconstruction

This section describes our method for reconstructing the moving scene, given an estimate of the occluder. In general, we reconstruct the moving scene from the mean-subtracted observation movie Y_μ[3].

To perform the reconstruction, we first formulate the matrix $\hat{\mathbf{A}}$, which describes the linear transformation corresponding to convolution by the estimated occluder $\hat{A}$. If Y_μ is of size $x_{\text{obs}} \times y_{\text{obs}}$, and the part of the scene containing movement is of size $x_s \times y_s$, then by necessity, $\hat{\mathbf{A}}$ will be a matrix of size $(x_{\text{obs}} y_{\text{obs}}) \times (x_s y_s)$[4].

Once we've formulated the forward model $\hat{\mathbf{A}}$, we can reconstruct the moving scene simply by inverting $\hat{\mathbf{A}}$ with regularization:

$$\hat{X}_\mu = \lambda(\hat{\mathbf{A}}^T \hat{\mathbf{A}} + \lambda I)^{-1} \hat{\mathbf{A}}^T Y_\mu \tag{3}$$

Note that in Equation 3, $\hat{X}_\mu$ and Y_μ have both been flattened into vectors; that is, instead of being matrices of size (x_s, y_s) and $(x_{\text{obs}}, y_{\text{obs}})$, respectively, they are vectors of size $x_s y_s$ and $x_{\text{obs}} y_{\text{obs}}$.

If we are reconstructing an RGB image, we do the calculation of Equation 3 for each of the three color channels individually, and then assemble them into a single image.

[3] If the observation plane is perfectly white and uniform, and there are no "nuisance light" sources from anywhere besides the scene, the raw observation movie may be used instead.

[4] In an experimental setting, it's possible that the size of the moving scene (x_s, y_s) will be unknown. In this case, we recommend tuning the size of the scene by hand, erring on the side of larger $(\hat{x}_s, \hat{y}_s)$. If the chosen $(\hat{x}_s, \hat{y}_s)$ are too small, the reconstruction will be overconstrained and will produce nonsense; if, on the other hand, the chosen $(\hat{x}_s, \hat{y}_s)$ are too large, the expanded area will contain noise, but the subset of the scene corresponding to the signal will remain intelligible.

In Equation 3, the regularization parameter λ can be tuned for optimal performance. We generally found that a value of λ between 1 and 10 yielded the best reconstructions in experimental settings.

6. Deviations

In Section 3, we described assumptions that we made in order to guarantee that the observation would reflect the convolution of the occluder with the scene. For clarity, we repeat these assumptions here. First, we assume that the scene, occluder, and observation lie on parallel 2D planes. Second, we assume that the scene, occluder, and observation are far apart relative to their size. And third, we assume that the observation plane is perfectly Lambertian, white, or uniform.

Naturally, in most real-world settings, few, if any, of these assumptions will hold. So is the algorithm we present here useless in practice? No, in fact. If nothing else, in Section 7, we present the results of our algorithm in experimental settings in which all of these assumptions are violated, and these results demonstrate that our algorithm can be used in real-world settings to approximately recover hidden scenes and occluders.

We do, however, consider it instructive to describe in more detail the distortions introduced by violating the aforementioned assumptions.

6.1. Non-planar or non-parallel objects

Consider the following example of an incorrect planarity assumption: suppose that we assume the occluder to be a disk, but it is in fact a sphere.

As explained in Sec. 3, the observation is the convolution of the occluder with the scene because translating an impulse light source in the scene corresponds to translating its corresponding shadow on the observation. This will be true if the occluder is a disk, but not if it is a sphere. In general, the shadow of a parallel disk on the observation plane will be a circle, but the shadow of a sphere will be an ellipse whose eccentricity will vary with the (x, y)-position of the light source. Figure 5 illustrates this, and shows a reconstruction of a simple scene using when incorrectly assuming the occluding sphere to be a disk.

6.2. Nearby objects

Assuming that the scene, occluder, and observation are far apart from each other relative to their size is common in occluder-based imaging [30, 4], and is generally called *the far-field assumption*. The benefit of making the assumption is that it lets you ignore the effects of distance attenuation. The ray optics model tells us that if a small, flat surface of area dA is a distance r from a light source of intensity I, and the surface normal is at an angle of θ from the incident light, then the contribution c of the light source to the light intensity on the surface will go as:

$$c \sim I \frac{dA \cos(\theta)}{r^2} \tag{4}$$

Suppose we have two parallel planes, p_1 and p_2, a distance z apart. In that case, we can use Eq. 4 to derive the contribution of a light source of intensity I at $(0, 0)$ on one of the two planes to a small patch at (x, y) with area dA on the other. In this case, the contribution simplifies to:

$$c \sim I \frac{zdA}{(x^2 + y^2 + z^2)^{3/2}} \tag{5}$$

Now we can see what is meant more precisely by the scene, occluder, and observation being "far apart relative to their size." When $z \gg \sqrt{x^2 + y^2}$ for all (x, y) on either plane, then Eq. 5 simplifies to $c \sim I \frac{dA}{z^2}$, and the contribution of a source of light at any point on p_1 to any point on p_2 will be the same, irrespective of their locations on either plane. This is a necessary condition for a translation of a light source in the scene to simply translate its observed shadow, which is in turn a necessary condition for the observation plane to reflect the convolution of the scene and occluder, as discussed in Section 3.

See Fig. 5 for a simulated reconstruction of a nearby scene while incorrectly assuming it to be far away.

6.3. Imperfections on the observation plane

Most surfaces are not perfectly uniform and white. Subtracting the mean frame from the observation video will help to reduce the effects of imperfections on the observation plane, to an extent. Color variations on the observation plane will still cause visible artifacts, however, because a darker region of the observation plane will respond less to overall increases in luminosity than a brighter region.

Non-Lambertian surfaces pose even more of a challenge. If the observation plane is sufficiently non-Lambertian, then the most important reflections off the surface will not be diffuse, but will vary strongly as a function of the angle of the incident light. This will confuse our algorithm, and probably render its output useless. However, a sufficiently non-Lambertian surface may also make the problem much easier to solve, if the observation plane is mirror-like!

We don't show a simulated example corresponding to imperfections on the observation plane in Fig. 5, because their effect isn't much different from simple noise, which we account for using regularization (as explained in Section 5). However, in Section 7, we show experimental results for which the observation plane includes visible imperfections.

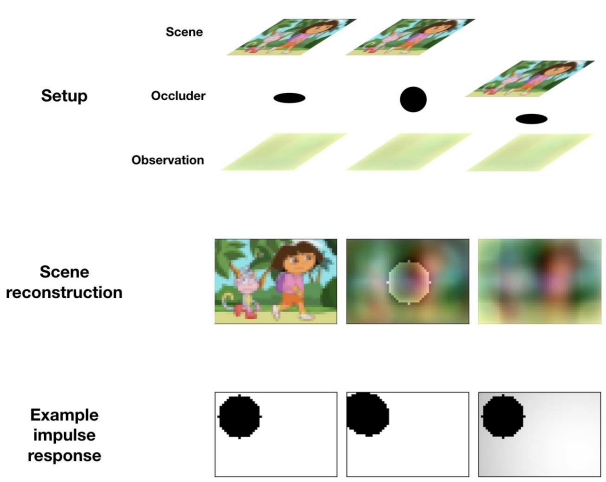

Figure 5: An illustration of the effects of the planarity assumption and the far-field assumption on the reconstruction. The top row shows a sketch of the true setup; in all three cases, the assumed setup is the one on the left. The middle row shows what reconstructions, generated using the approach described in Sec 5, of the leftmost image look like when the assumptions used for that approach are violated. The bottom row shows example impulse responses for each of the three scenarios. All data shown here is simulated, with no noise, to isolate the effect of each assumption.

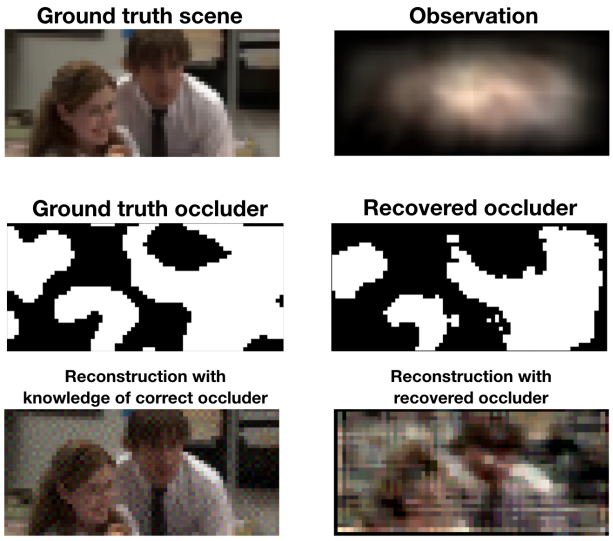

Figure 6: The output of the occluder-recovery and scene-reconstruction algorithms presented in Secs. 4 and 5, using the difference frames of a simulated observation at 25dB. See the supplementary materials for a full recovered video.

7. Results

In this section we present a summary of our results, both simulated and experimental. We show our reconstructions of occluders, along with a few still frames of reconstructed video. We leave the bulk of our results to the supplementary materials, however, as reconstructions of moving scenes are best seen in video form.

7.1. Simulations

In this section we show the result of simulations in an ideal scenario (all of the assumptions explored in Section 6 are assumed to hold perfectly). The moving scene is the introduction to a popular television show. The ground-truth occluder was generated via a random correlated process. The observation plane is assumed to display the full convolution of the moving scene with the occluder, plus additive IID Gaussian noise. The signal-to-noise ratio on the observation plane is 25 dB.

Figure 6 shows the result of occluder recovery, as well as a recovered still frame from the moving scene.

7.2. Experiments and Comparisons to Past Work

There has been surprisingly little past work as of this writing that does computational periscopy with the aim of recovering a head-on (as opposed to top-down) full-color 2D image of a scene in the passive setting (that is, without making use of active, directed illumination). Until 2019, the closest would have been the work of Bouman et al. in [4], but the full-color reconstructions shown in that work focus on 1D scene reconstructions, not 2D.

In 2019, however, Saunders et al. [21] showed that high-fidelity 2D full-color images could be recovered using a pinspeck occluder. Their experimental results differ from ours in two important ways. Firstly, they presume knowledge of the shape of the occluder (a pinspeck), though not its location in three-dimensional space. This gives them full knowledge of the imaging system, up to translation and scaling of the output. Second, their results are gathered from a still image with 3.5s of effective exposure time, whereas ours are drawn from a 100-FPS video of a moving scene (although the reconstruction shown is averaged over 5 frames of the ground-truth video, representing about 0.05s of exposure time). This implies a difference in signal strength. Only results from LCD monitor scenes are shown in [21]; to make it easy to compare our results with theirs, we include scenes from a cartoon shown on an LCD monitor as well as real-life scenes under heavy illumination in Figure 8.

As we can see in Figure 8, our monitor-based reconstructions are substantially lower-quality than those of [21]. We believe that this difference in reconstruction quality is primarily due to our system's imperfect knowledge of the occluder's form, which is a problem that the system of [21] does not have. We believe that the difference in SNR between the two settings may play a minor role as well.

Figure 7 shows the result of occluder recovery along-

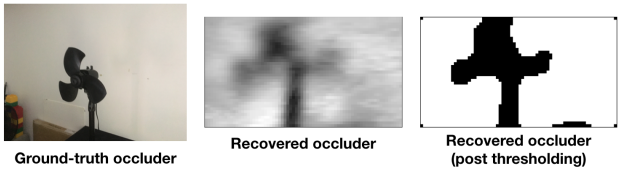

Figure 7: The output of the occluder-recovery algorithm presented in Section 4 in the experimental setting, alongside the ground-truth occluder. This is the occluder recovery used in the reconstruction shown in the second row of Figure 8.

Figure 8: Still frames from reconstructed videos under a variety of different experimental settings. Top row: the scene is a cartoon video, playing on an LCD monitor. Middle row: the scene is a moving man, illuminated by 200W of directed lighting. Bottom: the results of Saunders et al. [21], presented for comparison. The results of Saunders et al. demonstrate the potential improvement over our result when the form of the occluder is known. See Subsec. 7.2 for further discussion.

side its ground-truth counterpart. This recovered occluder is used for scene reconstruction in the live-action experiment in Figure 8.

8. Conclusion

This work is the first, to our knowledge, to study blind, fully uncalibrated non-line-of-sight imaging in the passive setting and produce experimental results. We believe that many of the most practical NLoS systems in the future will be uncalibrated, because in a wide variety of settings, most notably automatic driving systems in traffic, one cannot rely on having detailed preknowledge of the surrounding environment. Real-world scenes are constantly changing; we believe that work on NLoS systems should reflect that reality.

However, we believe that an enormous amount of progress is still possible on this new problem. In particular, our results remain quite low-resolution, and to achieve them we had to introduce a high level of illumination (about 200W of lighting over an area of 2-3m—we believe this is comparable to a scene in direct sunlight with the observation in direct shadow, but at this moment the comparison remains speculative, and this represents much stronger lighting than would be typical of an indoor scene). We hope that future work will be able to solve the problem of blind occluder-based imaging in real-world settings with much dimmer lighting; indeed, we believe that such progress will be necessary for these kinds of systems to prove useful in practice.

Additionally, we note that our occluder-recovery algorithm had more difficulty recovering more complex occluders—which is unfortunate, since past work [30, 16] shows more complex occluders to lead to more favorable reconstructions in general. Whether or not this represents a fundamental tradeoff, or whether future blind occluder-recovery algorithms will be able to recover complex occluders as easily as simple ones, remains to be seen.

Moreover, the algorithm we present here could probably be adapted to a wider variety of settings without too much difficulty. In particular, it could be very interesting to see our algorithm adapted to the setting in which the occluder and observation plane are perpendicular (such as a tree casting shadows on the ground). Our algorithm as presented won't work in this case, but we don't think that this natural extension of the problem is any harder in principle.

It would also be very natural to extend this work to the scenario where the scene is fixed, but the occluder is moving. We explored this possibility, but found it difficult to get good results from the first step of the algorithm, i.e. estimating the fixed scene. This fixed hidden scene in the modified problem is substantially less constrained than the fixed occluder in the original problem, since it can take on a wide range of values across three color channels, making it harder to infer. However, we consider this to be a promising direction for future research.

References

[1] S.-i. Amari, S. C. Douglas, A. Cichocki, and H. H. Yang. Multichannel blind deconvolution and equalization using the natural gradient. In *Signal Processing Advances in Wireless Communications, First IEEE Signal Processing Workshop on*, pages 101–104. IEEE, 1997. 2

[2] M. Baradad, V. Ye, A. B. Yedidia, F. Durand, W. T. Freeman, G. W. Wornell, and A. Torralba. Inferring light fields

from shadows. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6267–6275, 2018. 1, 2

[3] P. V. Borges, A. Tews, and D. Haddon. Pedestrian detection in industrial environments: Seeing around corners. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4231–4232. IEEE, 2012. 1, 2

[4] K. L. Bouman, V. Ye, A. B. Yedidia, F. Durand, G. W. Wornell, A. Torralba, and W. T. Freeman. Turning corners into cameras: Principles and methods. In *International Conference on Computer Vision*, volume 1, page 8, 2017. 1, 2, 3, 6, 7

[5] M. Cannon. Blind deconvolution of spatially invariant image blurs with phase. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(1):58–63, 1976. 2, 5

[6] T. F. Chan and C.-K. Wong. Total variation blind deconvolution. *IEEE transactions on Image Processing*, 7(3):370–375, 1998. 2, 5

[7] F. C. C. De Castro, M. C. F. De Castro, and D. S. Arantes. Concurrent blind deconvolution for channel equalization. In *Communications, 2001. ICC 2001. IEEE International Conference on*, volume 2, pages 366–371. IEEE, 2001. 2

[8] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. In *ACM transactions on graphics (TOG)*, volume 25, pages 787–794. ACM, 2006. 2

[9] G. Gariepy, F. Tonolini, R. Henderson, J. Leach, and D. Faccio. Detection and tracking of moving objects hidden from view. *Nature Photonics*, 10(1):23–26, 2016. 2

[10] F. Heide, L. Xiao, W. Heidrich, and M. B. Hullin. Diffuse mirrors: 3d reconstruction from diffuse indirect illumination using inexpensive time-of-flight sensors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3222–3229, 2014. 1, 2

[11] M. Hirsch, S. Sra, B. Schölkopf, and S. Harmeling. Efficient filter flow for space-variant multiframe blind deconvolution. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 607–614. IEEE, 2010. 2

[12] A. Kadambi, H. Zhao, B. Shi, and R. Raskar. Occluded imaging with time-of-flight sensors. *ACM Transactions on Graphics (ToG)*, 35(2):15, 2016. 2

[13] A. Kirmani, T. Hutchison, J. Davis, and R. Raskar. Looking around the corner using transient imaging. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 159–166. IEEE, 2009. 2

[14] J. Klein, C. Peters, J. Martín, M. Laurenzis, and M. B. Hullin. Tracking objects outside the line of sight using 2d intensity images. *Scientific reports*, 6:32491, 2016. 2

[15] D. Krishnan, T. Tay, and R. Fergus. Blind deconvolution using a normalized sparsity measure. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 233–240. IEEE, 2011. 2, 5

[16] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM transactions on graphics (TOG)*, 26(3):70, 2007. 2, 8

[17] A. Levin, Y. Weiss, F. Durand, and W. Freeman. Understanding and evaluating blind deconvolution algorithms. 2009. 2, 5

[18] C. L. Matson, K. Borelli, S. Jefferies, C. C. Beckner Jr, E. K. Hege, and M. Lloyd-Hart. Fast and optimal multiframe blind deconvolution algorithm for high-resolution ground-based imaging of space objects. *Applied Optics*, 48(1):A75–A92, 2009. 2

[19] R. Pandharkar, A. Velten, A. Bardagjy, E. Lawson, M. Bawendi, and R. Raskar. Estimating motion and size of moving non-line-of-sight objects in cluttered environments. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 265–272. IEEE, 2011. 2

[20] R. Raskar, A. Agrawal, and J. Tumblin. Coded exposure photography: motion deblurring using fluttered shutter. *ACM Transactions on Graphics (TOG)*, 25(3):795–804, 2006. 2

[21] C. Saunders, J. Murray-Bruce, and V. K. Goyal. Computational periscopy with an ordinary digital camera. *Nature*, 565(7740):472, 2019. 2, 7, 8

[22] T. J. Schulz. Multiframe blind deconvolution of astronomical images. *JOSA A*, 10(5):1064–1073, 1993. 2

[23] S. Shrestha, F. Heide, W. Heidrich, and G. Wetzstein. Computational imaging with multi-camera time-of-flight systems. *ACM Transactions on Graphics (ToG)*, 35(4):33, 2016. 1, 2

[24] C. Thrampoulidis, G. Shulkind, F. Xu, W. T. Freeman, J. Shapiro, A. Torralba, F. Wong, and G. Wornell. Exploiting occlusion in non-line-of-sight active imaging. *IEEE Transactions on Computational Imaging*, 2018. 2

[25] A. Torralba and W. T. Freeman. Accidental pinhole and pinspeck cameras. *International Journal of Computer Vision*, 110(2):92–112, Nov 2014. 1, 2

[26] A. Velten, T. Willwacher, O. Gupta, A. Veeraraghavan, M. G. Bawendi, and R. Raskar. Recovering three-dimensional shape around a corner using ultrafast time-of-flight imaging. *Nature communications*, 3:745, 2012. 2

[27] L. Xia, C.-C. Chen, and J. K. Aggarwal. Human detection using depth information by kinect. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, pages 15–22. IEEE, 2011. 1, 2

[28] F. Xu, D. Shin, D. Venkatraman, R. Lussana, F. Villa, F. Zappa, V. K. Goyal, F. Wong, and J. Shapiro. Photon-efficient computational imaging with a single-photon camera. In *Computational Optical Sensing and Imaging*, pages CW5D–4. Optical Society of America, 2016. 1, 2

[29] Y. Yang, N. P. Galatsanos, and H. Stark. Projection-based blind deconvolution. *JOSA A*, 11(9):2401–2409, 1994. 2, 5

[30] A. Yedidia, C. Thrampoulidis, and G. Wornell. Analysis and optimization of aperture design in computational imaging. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4029–4033. IEEE, 2018. 2, 3, 6, 8

[31] C. Zhou, S. Lin, and S. K. Nayar. Coded aperture pairs for depth from defocus and defocus deblurring. *International journal of computer vision*, 93(1):53–72, 2011. 2