Non-Line-Of-Sight Imaging with an Ordinary Digital Camera

Final Report

April 29th, 2024

CIS 6930/4930 - Computational Methods For Imaging

Dr. John-Murray Bruce

University of South Florida

Problem and Motivation

Non-Line-Of-Sight (NLOS) imagining is a practice that is commonly executed with very expensive imaging systems containing femtosecond lasers and specialized setups. To avoid having to incurr such a heavy cost we include an occluder, an opaque object partially blocking the scene, that allows for better conditioning by the addition of penumbra that allow better recovery of both the position of said occluder and the hidden scene. Only needing a high-resolution ordinary camera, and a single measurement.

NLOS has great value in observing hazardous environments, with a wide array of applications in autonomous driving, search and rescue, and medical imaging. Our setup allows not only for cost reduction in these areas but also adapting to their conditions outside of a laboratory.

Hardware Configuration

Our experimental setup is fairly straightforward yet critical for our success. It consists of a digital camera, an occluder, a hidden scene, a controlled light source, and computational tools for processing the data. First, the digital camera, we use a high-resolution digital camera mounted on a tripod to keep our images stable and consistent. Second, the occluder, consisting of a flat board placed between the camera and the hidden scene serves as our occluder. It blocks the direct line of sight, allowing us to capture only the light scattered from the hidden objects. We can adjust its position and orientation to see how these changes affect our final images. Third, the hidden scene, we place the objects we're trying to image so they're out of the camera's direct view. Fourth, the light Source: We have a controlled light source aimed at the hidden scene to ensure it is well-illuminated, avoiding direct exposure that might skew our results. Finally, computational resources, A dedicated computer equipped with MATLAB controls the camera, processes the images, and reconstructs the hidden scenes. We also simulate real-world conditions by introducing controlled noise into our data to test the robustness of our reconstruction methods.

Forward Problem

With the stated setup, we take a picture with our ordinary camera. This picture is basically a picture of the Lambertian wall. This camera measurement, measures the irradiance of the wall described through the formula:

$$I(\boldsymbol{p}_{\mathrm{w}}) = \int_{\boldsymbol{x}\in S} \left\{ \frac{\cos[\measuredangle(\boldsymbol{p}_{\mathrm{w}} - \boldsymbol{x}, \boldsymbol{n}_x)]\cos[\measuredangle(\boldsymbol{x} - \boldsymbol{p}_{\mathrm{w}}, \boldsymbol{n}_{\mathrm{w}})]}{\left\|\boldsymbol{p}_{\mathrm{w}} - \boldsymbol{x}\right\|_2^2} \times V(\boldsymbol{x}; \boldsymbol{p}_{\mathrm{w}}; \boldsymbol{p}_{\mathrm{o}})\mu(\boldsymbol{x}, \boldsymbol{p}_{\mathrm{w}}) f(\boldsymbol{x}) \right\} \mathrm{d}\boldsymbol{x} + b(\boldsymbol{p}_{\mathrm{w}})$$

It is of importance to pay close attention to the V(…) function that is a Boolean-valued function that equals 0 if the path is occluded. This adds sufficient dependency from a factor outside of the weak dependence of x that allows for better conditioning of the inverse problem. We thus now take this equation above and seek to discretize it in a similar way to this: $\mathbf{y} = \mathbf{A}(\mathbf{p}_0)\, \mathbf{f} + \mathbf{b}$, and as such obtain:

$$[\mathbf{A}(\boldsymbol{p}_{\mathrm{o}})]_{i,j} \approx \frac{1}{L}\sum_{k=1}^{L} \frac{\cos(\angle(\boldsymbol{p}_{\mathrm{w}} - \boldsymbol{x}_{j,k}, \boldsymbol{n}_{\boldsymbol{x}_{j,k}}))\cos(\angle(\boldsymbol{x}_{j,k} - \boldsymbol{p}_{\mathrm{w}}, \boldsymbol{n}_{\mathrm{w}}))}{\|\boldsymbol{p}_{\mathrm{w}} - \boldsymbol{x}_{j,k}\|_2^2} V(\boldsymbol{x}_{j,k}, \boldsymbol{p}_{\mathrm{w}}; \boldsymbol{p}_{\mathrm{o}})\mu(\boldsymbol{x}_{j,k}, \boldsymbol{p}_{\mathrm{w}})$$

That is the computed light-transport matrix for the forward model when calculated on all three color channels. Due to the presence of the occluder, this is not a Linear Shift Invariant model.

Inverse Problem

For the scene reconstruction process, we will de-interleave the Bayer pattern camera measurements y into RGB data with averaged green channels. From this, we can recover both the position of the occluder p_0 and the hidden scene of interest f in the spatial domain of the inverse model. This is a nonlinear problem, because the number of camera measurements is large relative to the recoverable resolution of the scene reconstruction.

First, we estimate the position of the occluder using the camera measurements y for each color channel and average to get a single estimate. A(p-$_0$) is the computed light-transport matrix for an occluder position p_0 with size of y as the number of rows and the number of columns depends on the resolution of attempted reconstruction f. The multiplication of $A(p_0)$ and its pseudo inverse with y decomposes y into 2 components: a component in range $A(p_0)$ and a component in the orthogonal complement, which represents the amount of model mismatch using the current occluder position. Minimizing the Euclidean square of this component, or the model mismatch, maximizes the estimation of the occluder position. Background noise b from the forward model is negligible to the reconstruction estimation accuracy.

$$\hat{\boldsymbol{p}}_o = \underset{\boldsymbol{p}_o}{\operatorname{argmax}} \left|\left| A(\boldsymbol{p}_o)[A(\boldsymbol{p}_o)^{\mathrm{T}} A(\boldsymbol{p}_o)]^{-1} A(\boldsymbol{p}_o)^{\mathrm{T}} \boldsymbol{y} \right|\right|_2^2$$

Next, we will compute the estimated hidden scene from the estimated occluder position. From the estimated occluder position, we can estimate the light-transport matrix using the left formula and compute its Moore-Penrose pseudo inverse using the right formula:

$$\hat{A} = A(\widehat{p_0}) \quad \hat{A}^{\dagger} = (\hat{A}^{\mathrm{T}}\hat{A})^{-1}\hat{A}^{T}$$

To produce an image of the computational FOV with robustness to ambient light prior to the scene reconstruction, we can compute the differences of measurements y for neighboring 16x16 pixel blocks vertically and corresponding rows of A using:

$$y_{i+1} - y_i \approx (\boldsymbol{a}_{i+1}^{\mathrm{T}}\boldsymbol{f} + b) - (\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{f} + b) \approx (\boldsymbol{a}_{i+1} - \boldsymbol{a}_i)^{\mathrm{T}}\boldsymbol{f}$$

In ideal conditions, where the estimated occluder position is exactly accurate and the model mismatch and background noises are insignificant, we can calculate the least-squares estimates of the hidden scene RGB content for each color channel using:

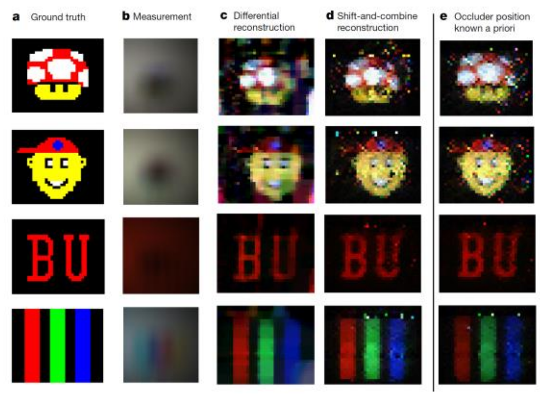
$$\hat{\boldsymbol{f}} = \hat{A}^{\dagger}\boldsymbol{y}$$

However, to improve the inverse model's robustness to noise and model mismatch in real-world conditions, we can promote sparsity in the scene's gradient using total variation (TV) regularization, making images appear smoother while preserving edges. The regularization parameter λ is determined through optimization to balance the accuracy of the reconstructed scene with the desired level of sparsity, to control the smoothness of the image:

$$\hat{\boldsymbol{f}} = \underset{\boldsymbol{f}}{\operatorname{argmin}} \; \|\hat{A}\boldsymbol{f} - \boldsymbol{y}\|_2^2 + \lambda \, \|\boldsymbol{f}\|_{\mathrm{TV}}$$

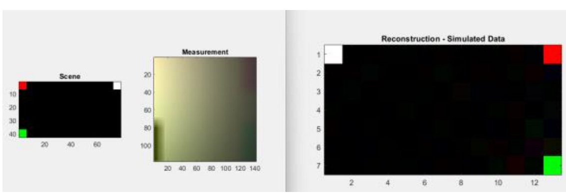
Findings

The researchers tested their system on 5 different images. They then took measurements of the visible wall, and computed the estimated occluder position and from there attempted a reconstruction of the images. To improve the reconstruction, the researchers then decided to run the reconstruction of the image 49 times using slightly varying occluder positions close to the estiamte the calculatednd combined them . Finally, they used the ground truth occluder position to get the best reconstruction possible.

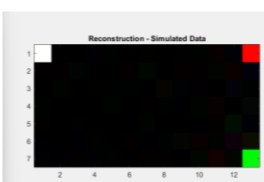


We then used the code provided to simulate the experiment in MatLab. The code gave a perfect reconstruction as in simulated environments the measurement and occluder position was perfect. We modified first the measurement itself by adding noise. Below are our results:

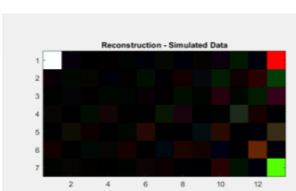
No noise:



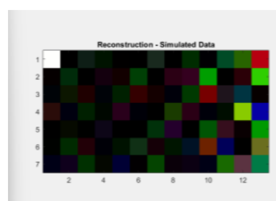
Signal to Noise Ratio (SNR) = 40:



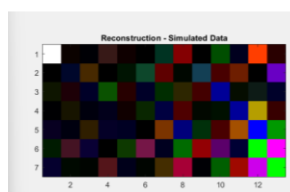
SNR = 25:



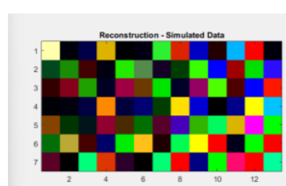
SNR = 15:



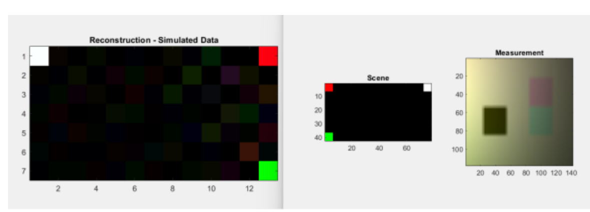
SNR = 10:



SNR = 0:



Next, we experimented with moving the occluder position to different locations. For this, we kept the noise at 0. Most positions resulted in bad reconstructions but we did find one position which gave a close to a perfect reconstruction:



From our experiments with the code, we learned that the occluder position is extremely important to the accuracy of the reconstruction of the scene. Clearly, in real experiments applying this the occluder position will need to be tuned to produce usable results. This results in a key limitation of the system in that it only works under very precise conditions.

Conclusion

In the context of Non-Line-of-Sight Imaging, the system performs well when the occluder is rectangular and properly positioned. However, any deviation in the occluder's shape or position significantly reduces reconstruction efficiency. Real-world scenarios often involve less-than-ideal conditions, such as poorly shaped or poorly positioned occluders. To enhance image quality under these circumstances, the system needs improvements. The paper proposes passive methods for Non-Line-of-Sight Imaging, addressing the ill-conditioned nature of the problem by incorporating the occluder. By employing noise reduction techniques and experimenting with various occluders and positions, reconstruction quality can be enhanced. Future work should focus on making these methods more adaptable to diverse environments without relying on precise occluder tuning. Additionally, increasing computational power would facilitate more effective implementation, especially beyond the limitations of MATLAB-based approaches.

Reference