Turning corners into cameras in Python

15-663 - Shane Li

1. Abstract

This is a final project report for 15-663 Computational Photography at CMU. In this project, I explore an implementation of a corner camera that allow users to any obstruction with an edge to reveal some scene on the other side of the obstruction. The data used to test this project was provided by Bouman et al. [1] on their website. The repo with the code and the data I gathered is at https://github.com/lishane31/15663-final.

Due to light scattering around the scene from global illumination, when objects move around the obstructing edge it creates subtle variations on the ground around the obstructing edge. Using these subtle variations, it is possible to extract 1-dimensional videos of the scene behind the obstructing edge. Using simple colored videos taken from consumer grade cameras, I show some reconstruction that reveal the number of people and their trajectories in the scene blocked by the obstructing edge.

2. Introduction

Being able to see around corners would prove extremely useful in many situations such autonomous vehicle's object detection for oncoming traffic around corners, as well as high pressure standoff or search and rescue situations where looking around the corner either possesses a danger or is physically impossible. Even though we are not able to see around corners, in many common environments light from the scene hidden behind an obstruction is scattered and reflected. Some of these light will be scattered into the observable scene, and cause minor changes in radiance. In this final project, I use these naturally occurring or man made obstructing edges in order to create a camera that allows us to grasp the number of people, and their trajectories, in the hidden scene. Since these obstructing edges are common all around us, this allows these corner cameras to be deployable in a wide range of environments.

The light emitted from the ground in our observable scene is determined by the light emitted on to it from the environment as well as its albedo, BRDF, and well the shape of its surface [1]. Of course, most of these light contributions from from the scene we can observe by standing next to this obstructing edge, but what might not be as obvious is that due to global indirect illumination, some of the light reflected on the ground comes from the hidden scene around the obstructing edge. This is what Bouman et al. calls a "penumbra" [1].

If someone were to stand at the base of the wall, they would hardly be able to see any scene around the hidden edge. However, as the person rotates about the obstructing edge in a circular fashion, they would be able to see more and more of the hidden scene. Due to this, the penumbra that the hidden scene casts on our observable scene increases as we stand farther and farther away from the wall. Thus, different portion of the ground integrates light from an increasing amount of the hidden scene. As demonstrated in Figure 1, as our angle away from the wall increases, the ground gets gradually brighter and we are able to observe more areas in the hidden scene.

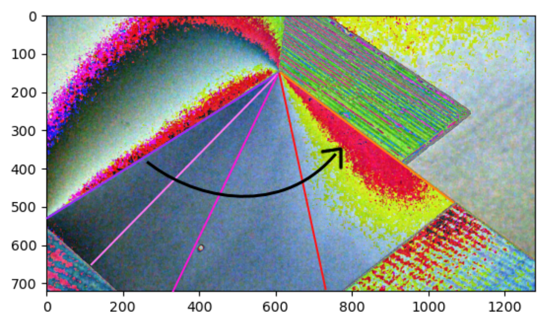

Figure 1. The magnified image of a frame from the video of the first scene provided by Bouman et al. [1]. In this video, a single person walks around the obstructing edge on the left side of the camera. The color is magnified by scaling each channel by a constant factor, and lines were added to demonstrate the relative increases in brightness of the ground as well as the motion of rotating around the obstructing edge away from the wall.

If someone were to enter the hidden scene, they would cause a minute change in the penumbra that would only be perceivable to our naked eyes if we were to magnify the image as shown in Figure 2 and Figure 3. However, by capturing a video of the penumbra using simple consumer grade cameras, we are able to use image processing techniques in order to extract these minute changes. Doing so, we can create 1 dimensional reconstructions of the hidden scenes.

In this paper, I will show how we can reconstruct the number and location of people in hidden scenes using obstructing edges, and demonstrate the results on data provided by Bouman et al. [1].

3. Methods

Using an edge camera with a hidden scene and an observation plane 4, we are able to extract minute changes in the

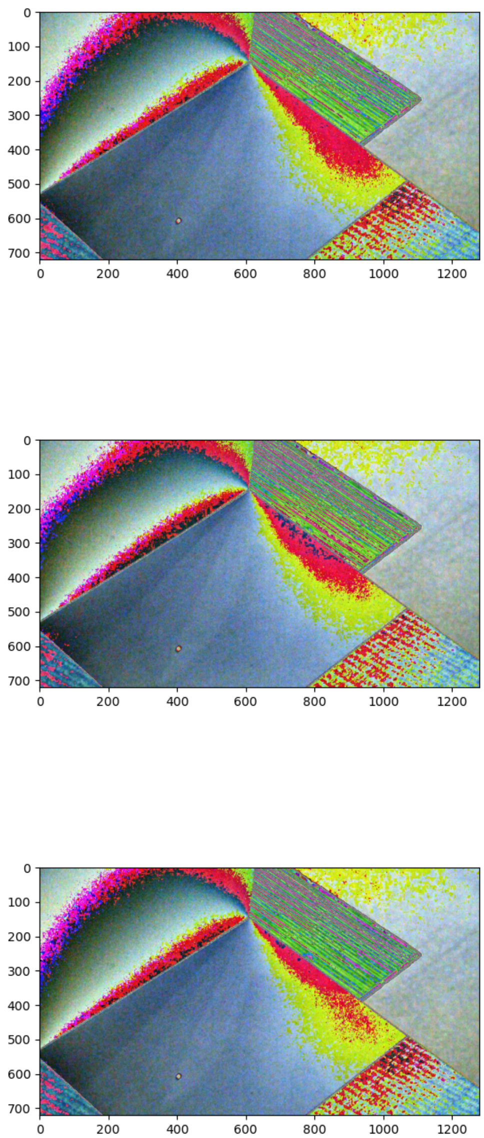

Figure 2. After magnifying the color by a constant factor, we can start to see minute differences in the observable scene in scene one caused by a person walking around in the hidden scene across different frames of the video capture.

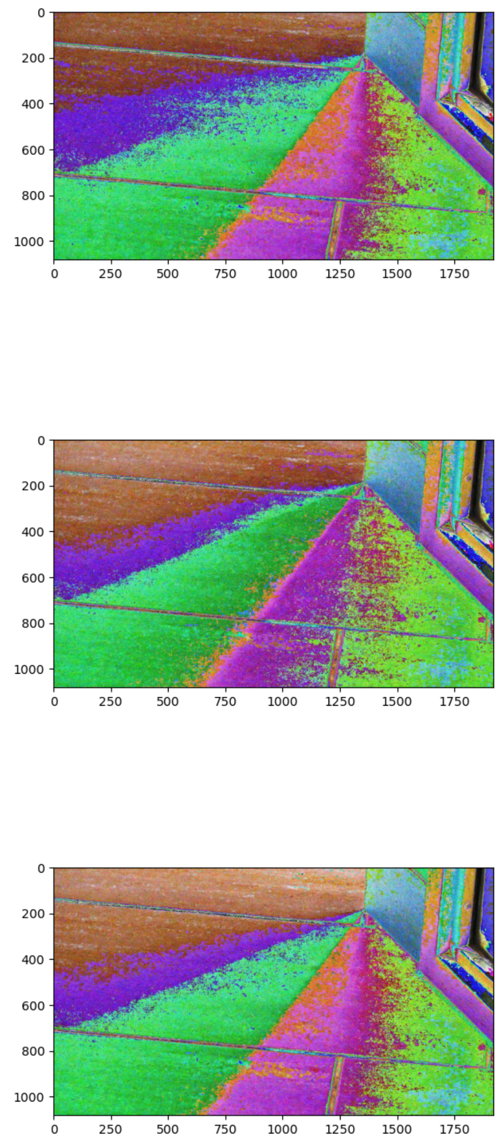

Figure 3. After magnifying the color by a constant factor, we can start to see minute differences in the observable scene in scene two caused by a person walking around in the hidden scene across different frames of the video capture.

penumbra and use these changes to infer about an object's position in the hidden scene.

The light reflected off of a surface at point p, with the unit normal $\hat{n}$, an incoming unit direction of $\hat{v}_i$ and an outgoing unit direction of $\hat{v}_o$, and a BRDF of β, is given by the equation:

$$L_o(p, \hat{v}_o) = a(p) \int L'_i(p, \hat{v}_i)\beta(\hat{v}_i, \hat{v}_o, \hat{n})\hat{v}_i \cdot \hat{n} d\hat{v}_i \quad (1)$$

Let us parameterize p in polar coordinates about the obstructing edge and θ increasing from 0 as we rotate away from the wall.

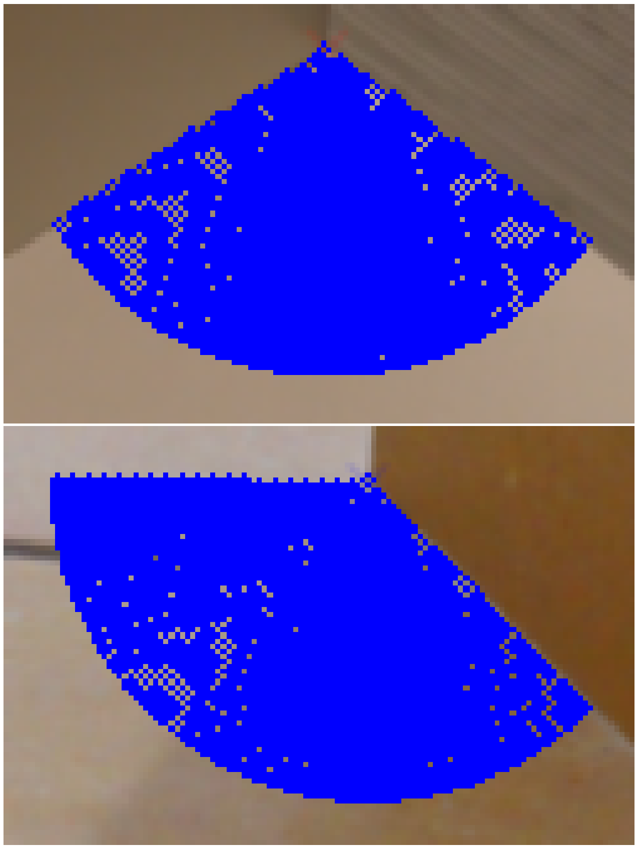

Figure 4. The observation plane that the recovery algorithm runs over. The top image shows the observation plane of scene one and the bottom image shows the observation plane of scene two.

Assuming that the observation plane is sufficiently Lambertian, then we can simplify Equation 1 by parameterizing by right ascension α and declination δ. Let $L_i = L_i' \hat{v}_i \cdot \hat{n}$, then:

$$L_o(r,\theta) = a(r,\theta)\int_{\alpha=0}^{2\pi}\int_{\delta=0}^{\frac{\pi}{2}} L_i(\alpha,\delta)d\alpha d\delta \qquad (2)$$

Given that the obstructing edge blocks light from $[\pi + \theta, 2\pi]$, let the contribution from the visible scene $L_v = \int_{\alpha=0}^{\pi}\int_{\delta=0}^{\frac{\pi}{2}} L_i(\alpha,\delta)d\alpha d\delta$ and contribution from the hidden scene $L_h(\phi) = \int_{\delta=0}^{\frac{\pi}{2}} L_i(\pi+\phi,\delta)d\delta$,

$$L_o(r,\theta) = a(r,\theta)[L_v + \int_{\phi=0}^{\theta} L_h(\phi)d\phi] \qquad (3)$$

Equation 3 shows that the light contribution to the penumbra is given by a constant term L_v denoting the light contribution from the visible scene, as well as the integral over light contribution from the hidden scene L_h. This integral is angle dependent on the parameterized angle θ and increases from only integrating light from the visible scene at $\theta = 0$ to fully integrating both scenes at $\theta = \frac{\pi}{2}$.

The 1 dimensional angular projection can then be recovered by taking the derivative:

$$\frac{d}{d\theta}L_o(r,\theta) = a(r,\theta)L_h(\theta) \qquad (4)$$

where we assumed $\frac{d}{d\theta}a(r,\theta)$ to be 0 and L_v goes to zero.

Then, if an object were to move around in the hidden scene, then the spatial derivative of the two different instances in time, $L_h^0(\theta)$ and $L_h^1(\theta)$, will give the angular change in the projection:

$$\frac{d}{d\theta}[L_o^1(r,\theta) - L_o^0(r,\theta)] = a(r,\theta)[L_h^1(\theta) - L_h^0(\theta)] \qquad (5)$$

Thus, the angular derivative of the differences between two frames of the video encodes the angular change between the two frames [1].

4. Implementation

Using the angular projections derived Section 3, I used the following methods to reconstruct the unknown hidden scene described in N discrete approximations x.

4.1. Discretization of the hidden scene

For each frame in the video, we can relate the pixels on the observation plane y^t to the angular projection $L_h^t(\theta)$ using:

$$y^t = L_v^t + Ax^t + w^t \qquad (6)$$

where $w^t \sim N(0, \lambda^2 I)$.

If there are M pixels on the observation plane, then the matrix A is an M x N matrix where each row denotes which discrete quantization of the hidden scene is observable from that pixel. In the dataset I used, the scenes revolve around the obstructing edge in a circle, and thus the matrix A is an upper triangular matrix where 0 values denote that the corresponding discrete quantization of the hidden scene can not be observed at that pixel due to the pixel lying at an angle θ too close to the base of the wall.

4.2. The estimation gain image

Then, according to Bouman et al. [1], the estimation gain image which computes an angular derivative over the observation plane is given by:

$$[\hat{L}_v^t \hat{x}^{tT}]^T = \sum^{t} \lambda^{-2}\tilde{A}^T y^t \qquad (7)$$

where $\tilde{A}$ is given by the augmented matrix [1 A]. Admittingly I was a little bit lost at this part, but an example of the estimation gain image I generated is show in Figure 5

Figure 5. The estimation gain image of the first scene with one person walking around.

4.3. Background subtraction

Then, in order to extract the differences between two frames of the hidden scene, we must first remove the influence of the visible scene's lighting. Our options here are to either capture a frame without the moving object, and subtract this reference frame from all frames of our video, or, given that in many real application scenarios we won't have access to this image, just use the mean image over all of the video's frames instead. In this project, I used the average frame across the video, and the results I gathered appeared to work well enough. This tells me that in the dataset the angular motions the person takes is approximately uniform.

4.4. Temporal smoothing

In order to further regularize the reconstructed image, a regularizer that imposes temporal smoothness is used. All results displayed are ran with the regularizer in effect.

4.5. Parameter selection

All results are ran with 200 discretization of the hidden scene, sampled at 20 concentric circles around the obstructing edge. Results not comparing different parameters are ran with noise parameter $\lambda = 10$, prior smoothing term $\beta = 100$, and interpolation strength $\alpha = 0.99$.

5. Results

Figure 6 shows the recovered trajectories of one person in indoor scene one.

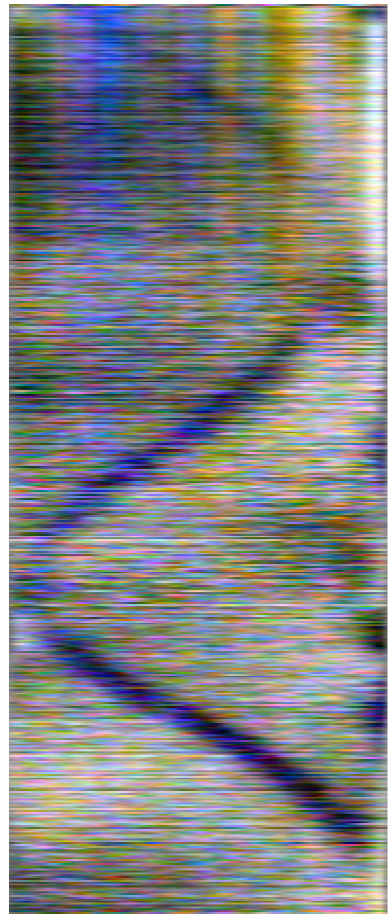

Figure 6. The recovered trajectory of the indoor hidden scene one with one person walking around.

Figure 7 shows the recovered trajectories of one person in indoor scene two.

Figure 8 shows the recovered trajectories of two people in indoor scene one.

Figure 12 shows differences between tuning the λ parameter from 1 to 10 to 50. As λ increases, the trajectory that we are able to capture becomes darker and more defined, but we start introducing more and more color noise all across the recovered projections.

Figure 13 shows differences between tuning the β parameter from 10 to 100 to 500. As β increases, the recovered trajectories gets more smudged out.

The same algorithm also works for outdoor and brightly lit environments. Figure 11 shows three different color magnified frames of an outdoor scene with one person walking around and Figure 10 shows the outdoor scene's observation plane. Figure 9 shows the recovered trajectories.

Figure 7. The recovered trajectory of the indoor hidden scene two with one person walking around.

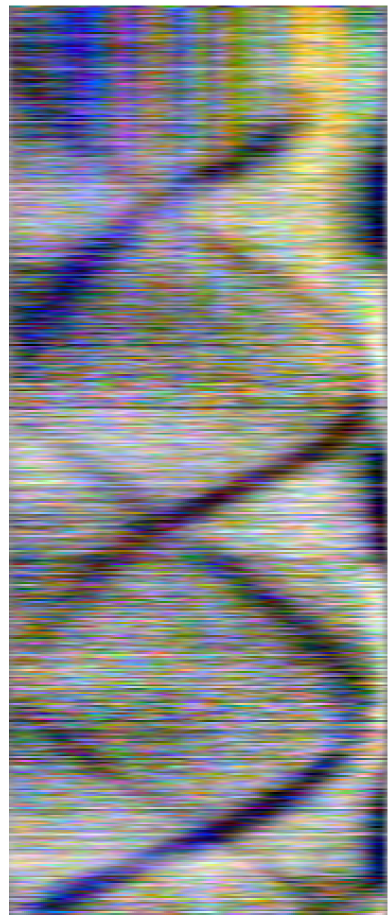

Figure 8. The recovered trajectory of the indoor hidden scene one with two people walking around.

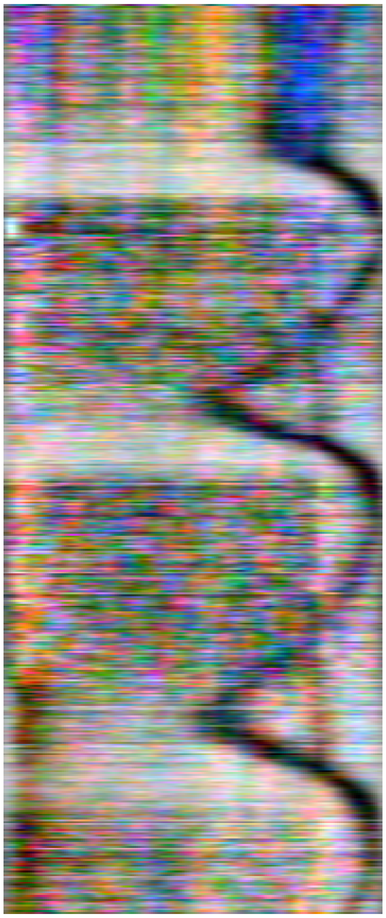

Figure 9. The recovered trajectory of the outdoor scene with one person walking around

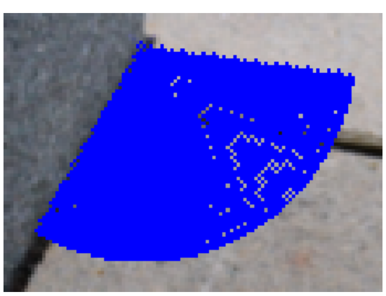

Figure 10. The observation plane that the recovery algorithm runs over for the outdoor scene.

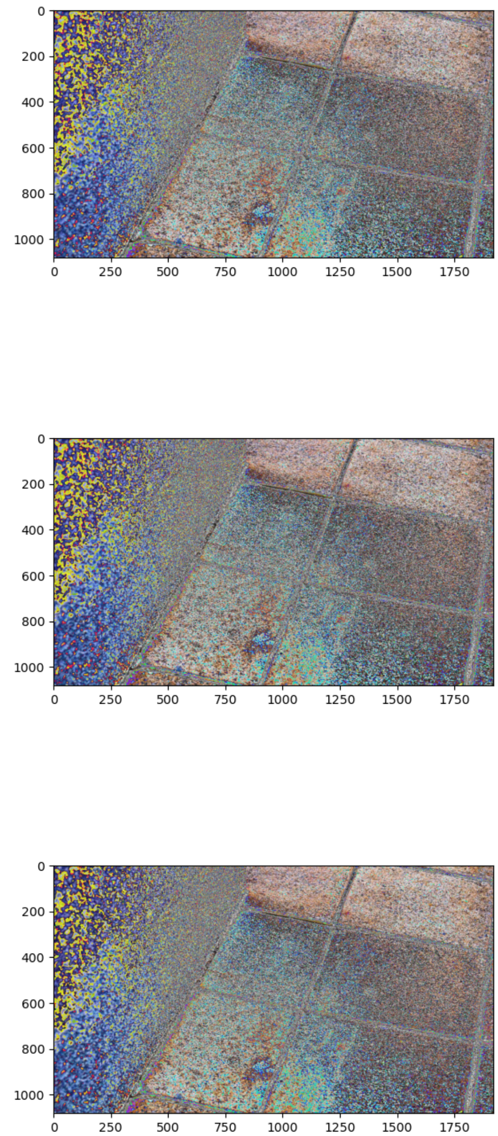

Figure 11. After magnifying the color by a constant factor, we can start to see minute differences in the observable scene in the outdoor scene caused by a person walking around in the hidden scene across different frames of the video capture.

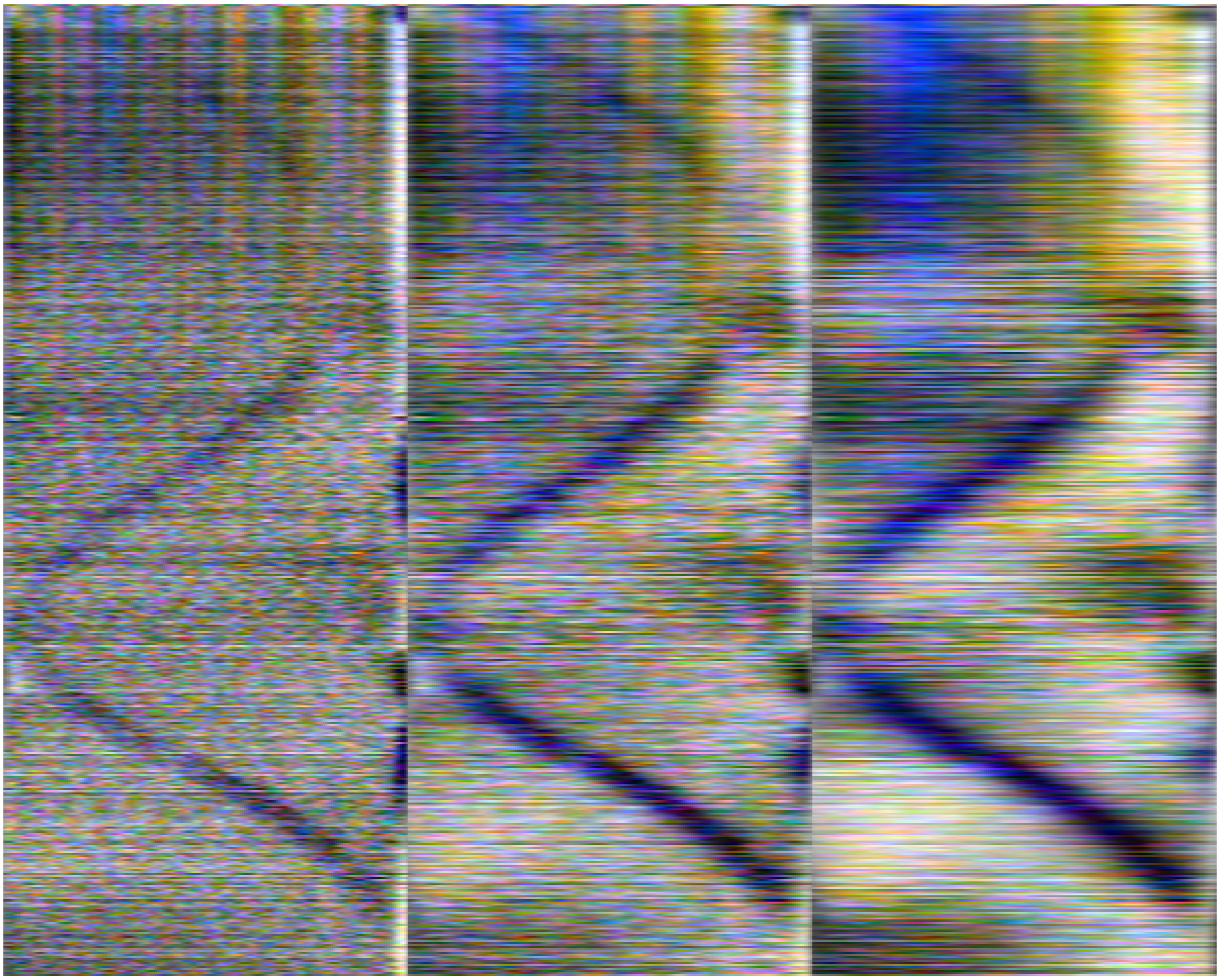

Figure 12. The recovered trajectories with $\lambda = 1$ (left), 10 (middle), and 50 (right).

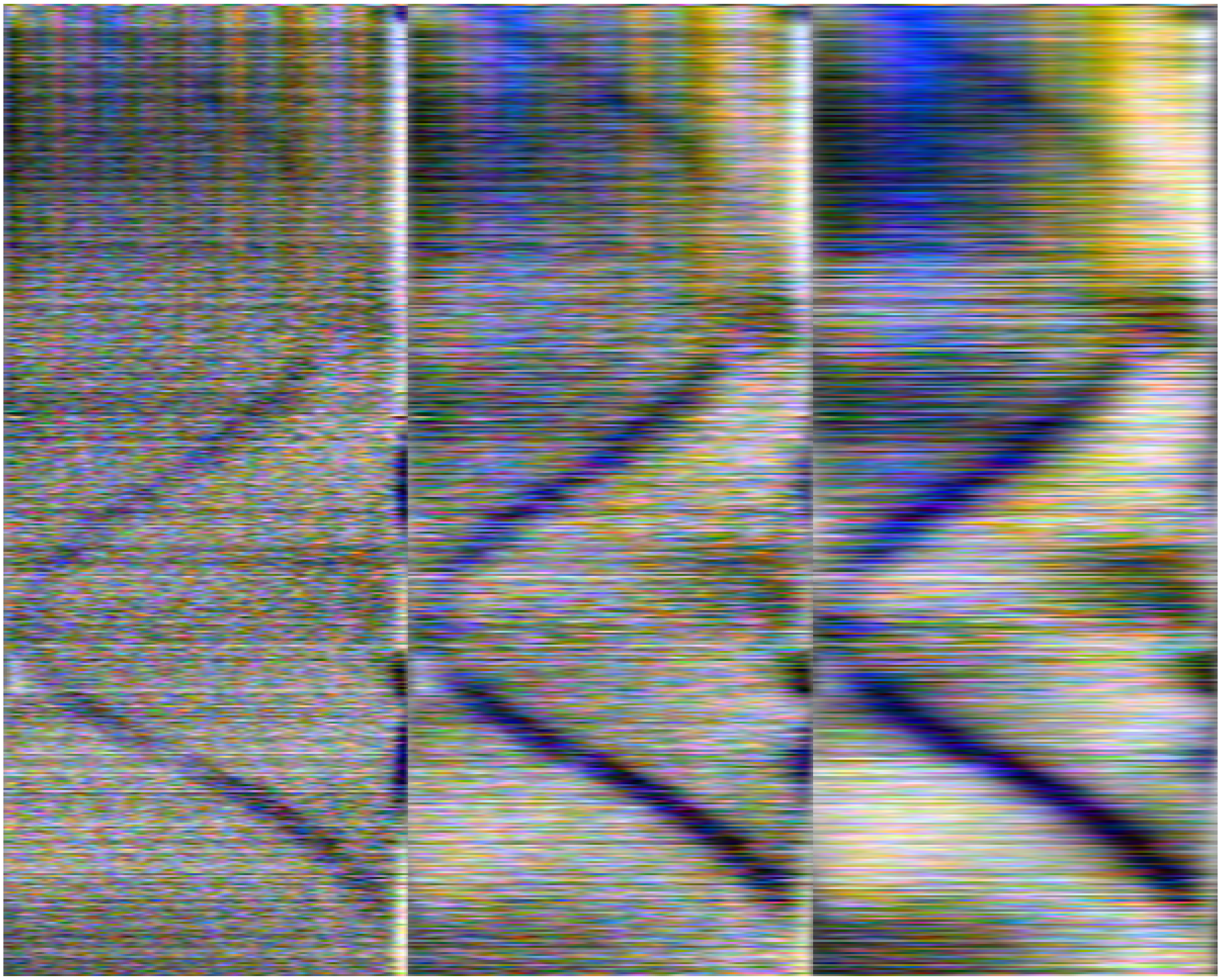

Figure 13. The recovered trajectories with $\beta = 10$ (left), 100 (middle), and 500 (right).

6. Final Words

This is my last assignment at CMU, as well as what might be my last homework assignment for the foreseeable future. Both Physics Based Rendering and this class were incredibly fun and interesting, and I hope that our paths will cross again someday!

References

[1] Katherine L. Bouman; Vickie Ye; Adam B. Yedidia; Fredo Durand; Gregory W. Wornell; Antonio Torralba; William T. Freeman. Turning corners into cameras: Principles and methods, 2017. Website with sample data. 1, 3