15643 Computational Photography

Denise Yang

November 2022

1 Video Processing

I first loaded all the frames and subtracted the shadow image to acquire an array of delta images.

1.1 Finding Shadow Edges

In order to find the edge of the shadow in each frame, I cropped the vertical and horizontal unobstructed regions from each frame. I then took the regions in each frame of the delta images and shifted it tho the right by 1 pixel. Then I subtracted the shifted image from the original frame such that the edge of the shadow, where the shadow ends, has the highest positive value. After aggregating all of the points I used least squares to fit a line of form $ay+bx+c = 0$ to the pixels using homogeneous coordinates in order to avoid the trivial solution of $a, b, c = 0$. To do this I divided a and b by c and rearranged the equation as such $\frac{a}{c}y + \frac{b}{c}x = -1$. Since I cropped my images I also added an x offset to account for the 0 index corresponding to starting x value of the unobstructed region. After that I recovered the slope of the line via $m = \frac{a}{c} * \frac{c}{b}$ and the constant offset via $c = -1 * \frac{c}{b}$¿ Note that I solve the line given the y values i.e. my equations take the form $x = my + b$.

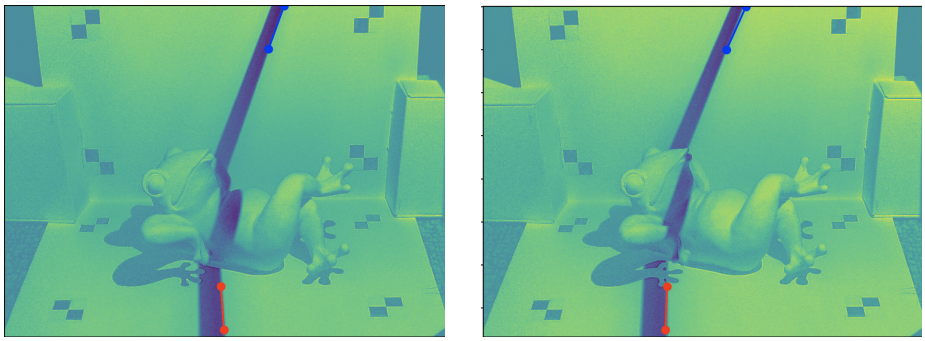
1.2 Finding Shadow Times

In order to find the shadow times, I subtracted the ith delta frame from the i+1th delta frame and also checked where sign flips existed between corresponding pixels between the two images. I then find the time where that value is the smallest. After accumulating all of these shadow times I masked out the regions where the pixel values were below a threshold of 0 most of the time meaning that that that area was likely shadowed across all the frames.

2 Calibration

2.1 Intrinsic and Extrinsic Calibration

I used the CalibrationDemo.py on the calibration images in the /calib directory.



(a) (b)

Figure 1: Recovered shadow edges

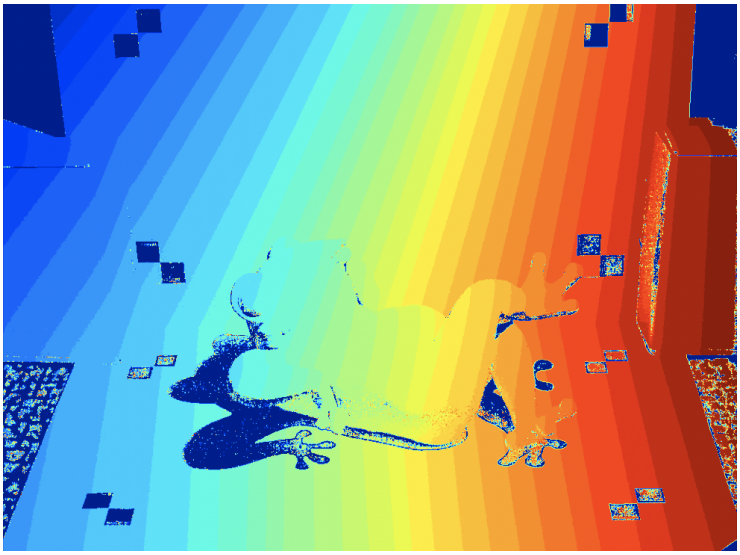


Figure 2: Recovered shadow times quantized to 32 times

2.2 Shadow Plane Calibration

I retrieved the linear equations for all of the shadow planes, and then found 2 points within each of the unobstructed regions. I then called pixel2ray on these points in order to translate them into 3d coordinates with the translation and rotation matrices from the prior portion. Afterwards I solved the equation $(r_{origin} + r_{dir} * t - P_0) \cdot n = 0$ for t in order to find the time of intersection with the shadow plane. $r_{origin} and r_{dir}$ are the respective origin and direction of the ray, n is a vector normal to the plane P_0 is a point on the plane. Solving for t we get $t = \frac{(P_0 - r_{origin}) \cdot n}{r_d ir \cdot n}$. Since we translated the ray to camera coordinates we can simplify our equation by choosing (0,0,0) and (0,0,1) as P_0 and n respectively thus reducing it to $\frac{-r_{origin} \cdot z}{r_{dir} \cdot z}$. We then multiplied the recovered the point of intersection by doing $pt = t * r_{dir} + r_{origin}$ and then converting it back to camera coordinated via $translateVector + rotationMatrix * pt.T$. We did this for each set of shadow edges that we found.

3 Reconstruction

For each pixel I convert them to their respective camera rays via pixel2ray. Then I use the shadow time of that corresponding pixel to get the respective shadow plane. I calculate the ray plane intersection of those rays and points via the same point intersection equation $(r_{origin} + r_{dir} * t - P_0) \cdot n = 0$ except now $(r_{origin} = 0$ since we're in camera coordinates. I acquire the normal by taking the cross product of the 4 plane points. To visualize my 3D reconstruction I used a 3D scatter plots of the points. In order to improve the results I clipped the depths to -2000 and 1000, and also did not equalize the aspect ratio since that resulted in my reconstruction looking more like a line.

4 Own Reconstruction

Our set up is shown below. The dimensions of the x's are 37.2cm by 32.1cm and the I camera ISO was 100 and the shutter speed was 1/10'.

Overall my results were not too great since the calibration images were rather dark and our horizontal x's were cut off leading to incorrect calibrations.

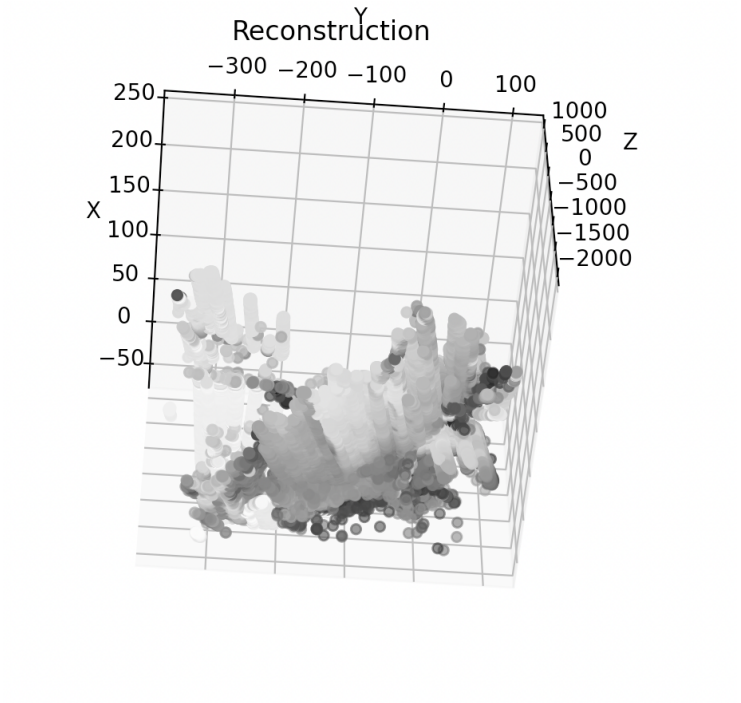
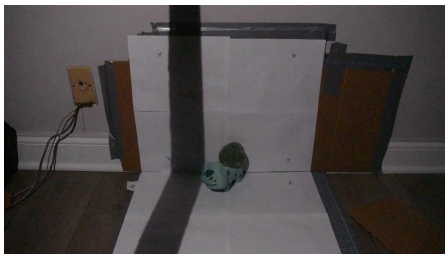
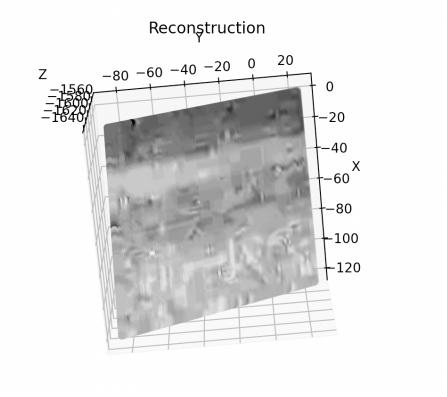


Figure 3: Setup

Figure 4: Reconstructed Frog



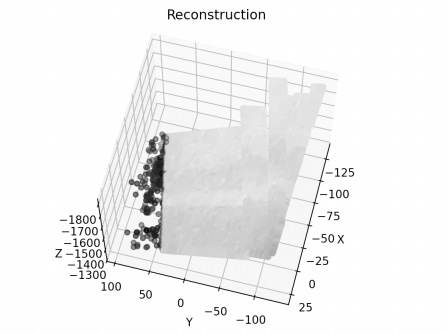
(a) Bulbasaur setup



(b) Bulbasaur reconstruction

Figure 5: Bulbasaur

(a) Snorlax setup



(b) Snorlax reconstruction

Figure 6: Snorlax