Computational Photography
Assignment #1: Epsilon Photography

Josh Adams

Fall 2019

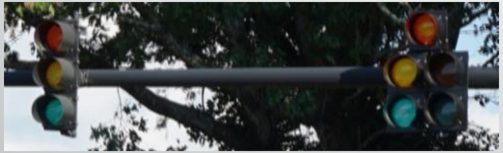
Epsilon Project Overview

Image 1

Image 2

Image 3

Image 4



Final Artifact

Description:

I took multiple pictures of a stop light at the various moments when different segments were lit. I wanted to combine the red light, green light and yellow light, into one image. The reason I wanted to do this was to make it appear as though the lights were all turned off at same time.

Project Discussion

- What is your epsilon parameter? *The traffic signal light*
- Location of pictures? *Asheville, North Carolina*
- Date and time? e.g. *August 26, 2019 from 5:00pm – 6:00pm*
- How did you control the settings, the environment, and the camera to meet your epsilon requirement?

I controlled the settings of my camera by setting putting my camera in a manual mode, where settings will not change on their own. The environment was the most difficult aspect to control as I needed to capture the traffic lights in the various stages. Where I was taking these pictures it was very unlikely that my light would ever change from 'red'. I managed to force the lights to change by utilizing the cross-walk button, although it required a very long wait time.

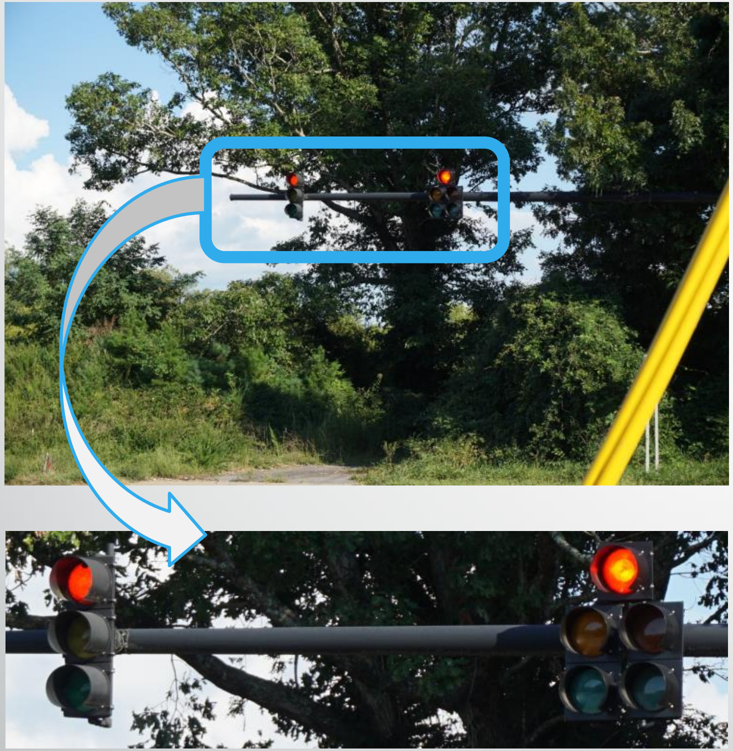


Image 1

Description:

Image of a traffic light where the ‘red’ light is illuminated. These pictures were taken by placing my camera on a large metal box near the road.

Camera settings:

Exposure: 1/160sec
Exposure Bias: 0 step
Aperture: 4.96875
ISO Speed: ISO-160
F-Stop: f/8
Focal Length: 50mm
Original Dimensions: 6000x4000 pixels
Focused Dimensions: 1676x504 pixels

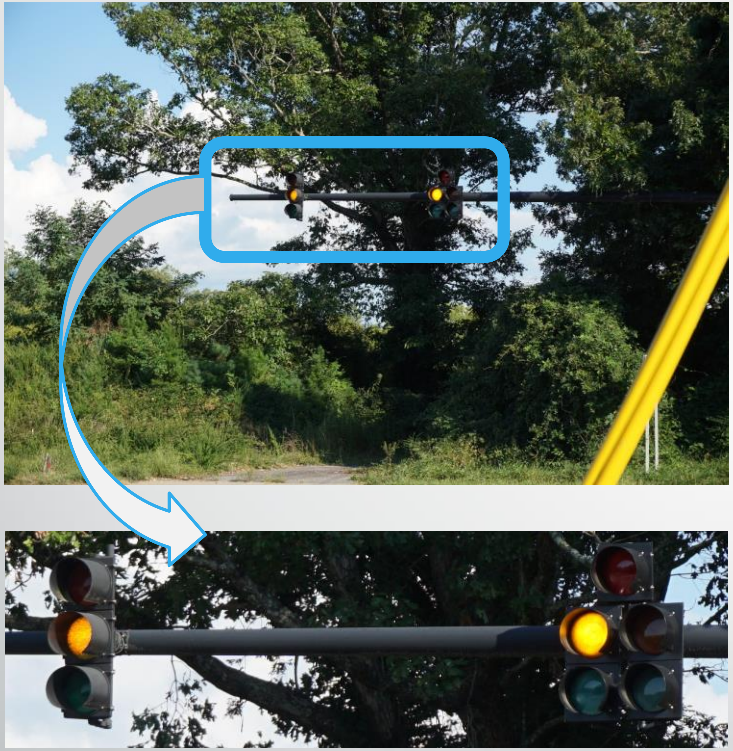


Image 2

Description:

Image of a traffic light with the ‘yellow’ light illuminated. It was a slightly windy day, as such the background vegetation moved in the various images.

Camera settings:

Exposure: 1/160sec
Exposure Bias: 0 step
Aperture: 4.96875
ISO Speed: ISO-320
F-Stop: f/8
Focal Length: 50mm
Original Dimensions: 6000x4000 pixels
Focused Dimensions: 1676x504 pixels

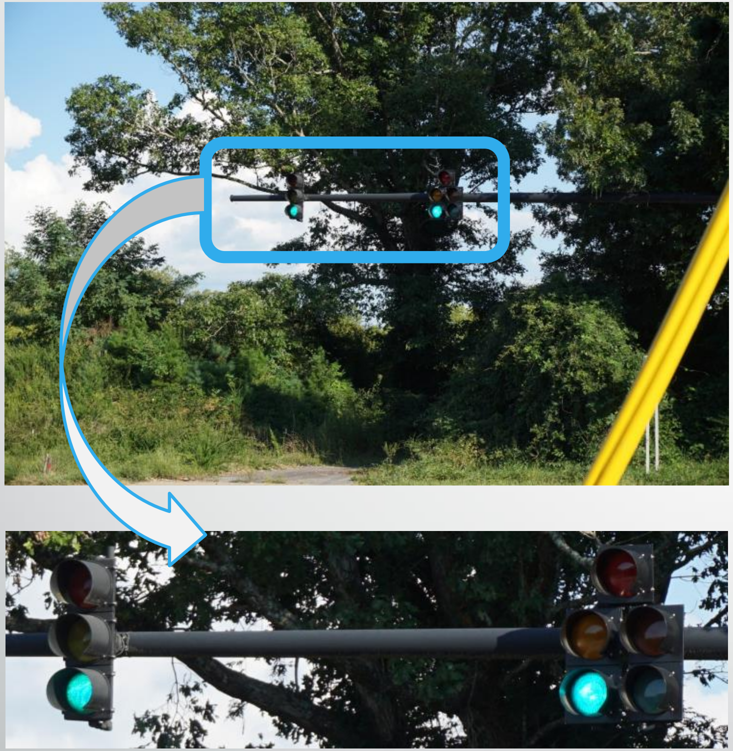


Image 3

Description:

This is an image of a traffic light with the ‘green’ light illuminated.

Camera settings:

Exposure: 1/160sec
Exposure Bias: 0 step
Aperture: 4.96875
ISO Speed: ISO-640
F-Stop: f/8
Focal Length: 50mm
Original Dimensions: 6000x4000 pixels
Focused Dimensions: 1676x504 pixels

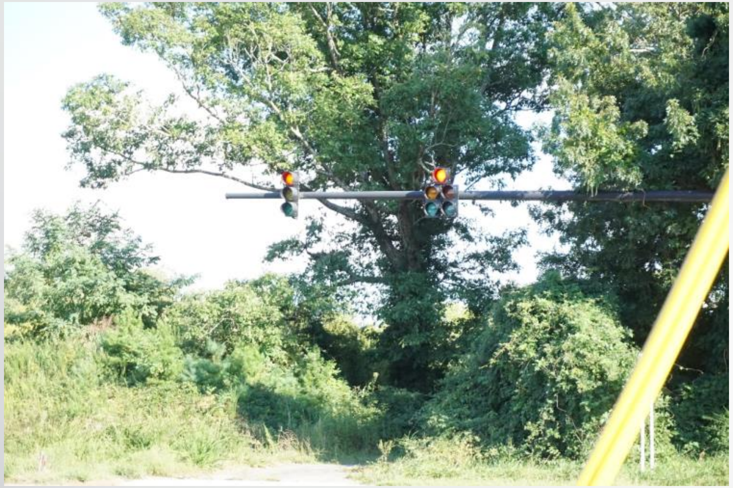


Image 4

Description:

This was my attempt to get an image of the traffic light in the middle of it changing lights. I was unsuccessful and the extremely long wait time restricted my attempts to only a handful, this being my best attempt.

Camera settings:

Exposure: 1/2000sec
Exposure Bias: 0 step
Aperture: 4.96875
ISO Speed: ISO-3200
F-Stop: f/8
Focal Length: 50mm
Original Dimensions: 6000x4000 pixels

Final Artifact

“Lights Out”

Final Artifact Details

- Discuss how your final artifact demonstrates your epsilon. I extracted the lights from each of the images. I did this by comparing the changes between the images and obtaining those which changed in the various images. This would produce the ‘red’, ‘yellow’ and ‘green’, since those are the ones that change

- Did you do anything to prepare your N images? (e.g. cropping, alignment) My code does all the preparing of the images I needed.

- Walk through the code you wrote to create the final artifact. You may provide code snippets to help in your discussion. Do NOT just copy and paste your code with in-line comments. **Explain** the purpose of the code that you’re presenting.

Final Artifact Details cont.

- **Create Bounding Box**
  a) *bounding_box = {"x1": 2225, "x2": 3901, "y1": 1357, "y2": 1861}*
     i. The bounding box was arbitrarily created by just looking at the coordinated of the initial image and finding some 'box' which contains the traffic lights in the images.

- **Crop Image using Bounding Box to a much more 'focused' area**
  a) *red_light_focus = red_light[bounding_box["y1"]:bounding_box["y2"], bounding_box["x1"]:bounding_box["x2"]]*
     i. I use the bounding box to crop out the area of the images I am wanting to work with as to reduce computation and complexity.

- **Process focused images**
  a) *red_light_channel_zero = ndimage.convolve(red_light_focus[1:-1, 1:-1, 0], weights=kernel)*
     i. I used my kernel to convolve over the image in the 3 channels independently.
     ii. The reason I did this is that when running convolution on the pixels it will ultimately take the average the channels which will result in a grayscale image.

- **Combine and Align Images**
  a) This was accomplished using two nested for-loops as I still am not extremely familiar with NumPy and this was the straightest forward method for me to get the result I was wanting.

Project Retrospective

- In what ways was your project successful?

My project was successful because I set out to accomplish the task of manually combining my images and I was able to do this. I also found some cool ways to process images, which I consider a success even if they do not directly apply to this project.

- If you were to repeat the project, is there anything you would do differently knowing what you do now?

If I were to repeat my project, I would do many things differently. I would first choose a different topic, other than traffic lights because it is hard to have a controlled environment. I had to consider many aspect which were constantly changing such as the background flora moving in the wind. It added a layer of complexity I was not expecting. The next issue was the lights taking very long to change.

Resources

Record your sources here. We accept all reasonable formats that would allow us to verify your sources. Our class lectures do not have to be referenced. If you are reusing your work from a prior semester, mention it here.

- I (Josh Adams) took the pictures
- I (Josh Adams) wrote the code for processing the images
- OpenCv
- NumPy