237D Final Report

Alex Yen

June 10, 2021

Abstract

Electric grid infrastructures are often ad-hoc in developing countries. This leads to a wide range of future scalability and grid-related problems; if the grid infrastructure is not well established, then grid expansion and electrification is hard; if there is no documentation, then there are no easy solutions to problems that arise. Traditional documentation methods require a lot of manual labor, time, and money, to which these efforts might be negligible with the volatility of the grid foundation, due to its unstructured and ad-hoc nature. We propose GridInSight 2.0, a system that is able to autonomously collect and document electricity grid information via computational imaging; street lights contain electricity sensitive information, which can be captured through commodity cameras. By capturing and recording a variety of light bulb waveforms called Bulb Response Functions (BRFs) in the lab, we are able to extract phase information and classify these BRFs based on their signature waveform and bulb type. This project focuses on geolocating street lights, building off of being able to uniquely identify street lights in the real world; this will allow us to generate a detailed grid mapping of any city's electric grid infrastructure that is purely based upon street light information. This methodology will provide efficient problem solving and expansion of grid-related problems in developing countries, which can also be applied to developed countries.

1 Introduction

The grid infrastructure in developing countries is ill-maintained. Based upon an ad-hoc fashion, most of the electricity infrastructure is poorly documented and unstructured, which causes problems down the line for infrastructure expansion and grid-failure debugging. Solutions to this problem include the traditional method for electric grid documentation, which requires physical labor to measure and record the current health at various points within the grid. Yet doing so in the long term is unaffordable, manual, and unscalable; as the grid expands, the methods by which a grid is maintained need to evolve to provide a more efficient solution towards grid health monitoring and grid expansion.

One solution involves computational imaging on the electric grid – cheap computing that will passively monitor and document electrical properties of the grid infrastructure, ultimately providing knowledge of the grid structure that can be used to both fix problems that arise and expand on the current infrastructure. To do so, we intend to construct an electric grid phase mapping purely extracted from images and cameras, in which this mapping can be used to isolate the power lines running throughout the grid. All modern electricity systems are divided into three phases—sinusoidal-based electricity that is delayed by parts of three with respect to the grid frequency—which can be extracted from a rolling shutter camera in the real world. Our system consists of a pre-built database of light bulb signature waveforms, which is used to extract the phases of street lights. In addition to extracting the phases of street lights, we also intend to uniquely identify street lights and interpolate their geospatial location (i.e. GPS location); if we image street lights all throughout a city but can only link phase information with respect to the original image, how do we amass all this information between images into a cohesive grid mapping?

To this end, we intend to use commodity cameras to image street lights, which contain electric-grid-sensitive information; these cameras include the use of smartphone cameras that local people already possess or have access to. From this we can extract phase information based on a method developed in [1], which we have done so in [2]. In this paper, I tried to measure depth with camera images, with the intent to measure the distance away from the camera to the street light. Doing so will allow us to obtain GPS location (either through the smartphone GPS information or an external sensor), which can be used to generate a grid mapping from camera images. The information obtained from street lights can be used to create a phase mapping of the electric grid – acquiring ground truth information about the grid enables more efficient grid debugging as well as grid expansion, both of which are problematic for developing countries. Progressing the methods to ensure proper grid health and grid expansion is fundamental to progressing society as a whole, a particular issue that renders developing countries in a “developing” state. Furthermore, our proposed method to monitor the infrastructure in developing countries is applicable to developed countries as well; the issue is less prominent in developed countries, but is nonetheless still present. Our application aims to be applicable for modern and future grid infrastructures for years down the line.

2 Background, Related Work, and Project directions

The bulk of my work was based off of [3]; their application intends to extract the GPS location of irrigation canals through a stereo camera setup, just as I intended to do with street lights. I had conducted experiments very early on in the quarter regarding depth estimation in meters. With a rough estimate of how far away my intended objects were, in addition to the results from mathematical triangulation for depth estimation, my initial results seemed fairly accurate; later in the paper, I will discuss all the problems with this initial assumption. Thus, after reading the white paper and finding that my initial results were acceptable (without ground truth data), I felt ready to move onto a more accurate solution: image rectification.

Without going into the details, image rectification allows for more accurate depth estimation results from triangulation with a stereo camera setup. The setup by [3] includes a StereoPi [4] camera and external sensors for future GPS geolocation from image features (e.g. inertial measurement unit (IMU) sensor). The current work and analysis includes calibrated the cameras (e.g. undistorted images, rectified images), in addition to intermediary depth estimation results. Rather than replicating the entire system built in [3], I tried to take bits and pieces of their implementation and apply that to our application. The main difference initially was the stereo camera setup – [3] purchased a stereo camera kit [3], and our application intends to use smartphone cameras.

The reason for using smartphones is catering towards our application in developing countries – with the advent of smartphone accessibility, smartphones are more available to design a system around than a specific system that requires manual work to set up. In order to convince locals to participate in grid maintenance with their personal smartphones, the system requires low maintenance and low effort for setting up. Ideally (and currently unrealistically), we should be able to use any two smartphones that can download an app and run software that automates the grid mapping process. In practice and reality, that goal requires more nuanced factors that we have not yet accounted for. As a result, our initial directions included building a system that involved smartphones (and hence their cameras), such that this system can be deployed with the readily available smartphones that local people might possess. However, after speaking with Ben Ochoa, a computer vision expert at UCSD, we realized a need for system simplicity – *get the system working first*. This will be further discussed in Section section 5.

As a result, we initially attempted the more complex route with smartphone cameras, due to the nature of our application. [3] used a SterePi setup with an automated camera calibration system [5], which I could not rely on because we did not possess that setup. It is noted that the authors of [3] had offered to build an identical setup for us, but (naively) I had refused the offer because (1) I thought I knew what I was doing, (2) I was not confident enough to rely on another system and apply that system to our work (especially at the cost of lab funding and potentially wasting others' time), and (3) our application intended to use commodity smartphone cameras rather than specific kits. I attempted to pull the most relevant pieces of information from [3] for our application and became fixated on image rectification.

3 Technical Material

For my main results this quarter, I initially wanted to estimate the GPS coordinates of features in images, which was a far cry from what I was actually able to achieve. In the end, I was only able to conduct a distance estimation study in meters from a study I conducted really early on in the quarter, the reasons being that I got side tracked by a concept called "image rectification" in addition to realizing that my depth estimation test mentioned in my milestone report was a flop; this will be mentioned in Sections 4 and 5. As a result, my experimental setup consisted of taking images along a sidewalk; sidewalks are nice because they have relatively equally spaced squares that I can just line up a camera and tripod with. Furthermore, the sidewalks are relatively straight – this is important to make images as coplanar to each other as possible. Afterwards, I took images with a fixed distance away from each other. To estimate depth from two images, I used a trigonometric methodology as shown in Figure 1.

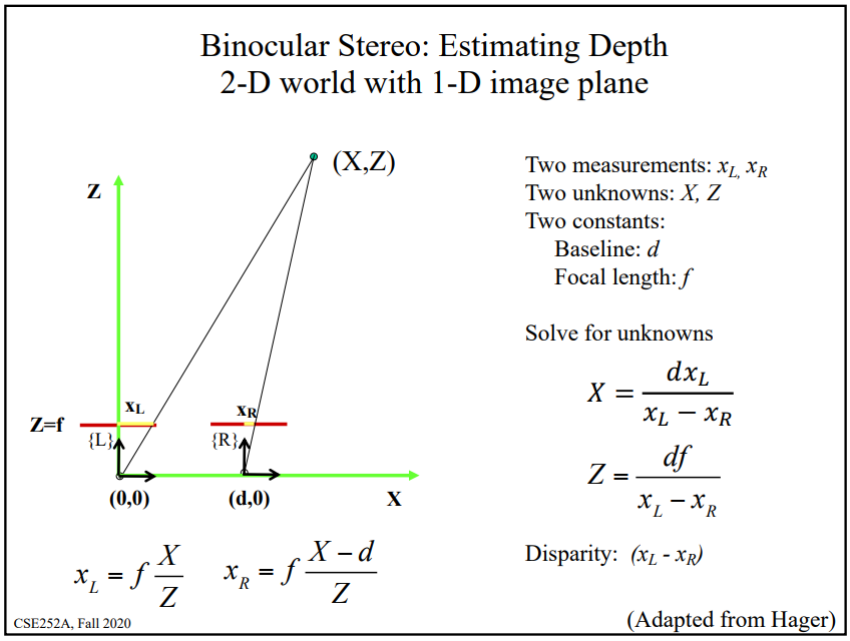

Figure 1: Trigonometric diagram of estimating the world coordinate of a feature from two images.

There are four measurements and constants to measure and calculate in order to estimate the point (X,Z):

- X_L – the column pixel value of the feature in the left image
- X_R – the column pixel value of the feature in the right image
- d – the baseline distance between the two cameras
- f – the focal length of the camera(s)

I found X_L and X_R manually through GIMP. d is measured by taking images with a fixed distance away from each other. f is found by converting the focal length in millimeters to pixel units. This is done through the following equation:

F(mm) = F(pixels) * SensorWidth(mm) / ImageWidth (pixel)

All the parameters for this equation can be found with an Android application called "DevCheck;" we pull the focal length (f), the sensor width (in millimeters), and the image width (in pixel units). We can then rearrange the equation to find the focal length in pixel units. After finding the constants, I wrote code to evaluate the consistency of my depth estimations. In my experimental setup, I took about nine consequent images, each 1.625 meters apart from each other. Then, in order to calculate depth, I pair images together. In my image dataset of about nine images, I pair images with respect to the first image (e.g. image_1 and image_2, image_2 and image_3, image_1 and image_4, etc.) and consequently (e.g. image_1 and image_2, image_2 and image_3, image_3 and image_4, etc.). I measured X_L and X_R for each image and then calculate the approximate distance away from the intended feature/object (near or far street light); the objects are shown in Figure 2. By measuring the ground truth distance away from the nearer and the farther street light, which are 12.5 meters and 28.8 meters away from the camera respectively, I was able to see errors as little as about 0.9 meters and as high as 5.6 meters. Considering that the distance between street lights are more than 10 meters apart from each other, these errors might be acceptable for our application.

However, after I spoke with Ben, I found that Figure 1 was not intended to be used the way I used it for this class; there steps such as image rectification/enforcing the epipolar constraint that I did not do, but as of now, I am still confused on whether my results are just wrong or inaccurate. However, I have new directions that will not rely on this methodology and do stereo vision the correct way.

Figure 2: Images from daytime dataset and nighttime dataset.

There are many issues that I encountered when using the rest of the calculations shown in Figure 1 to estimate the point (X,Z), which I will mention later in Section 5. For instance, if I use two different cameras, which focal length do I use to calculate Z? This issue did not occur when I did this calculation with just physically translated images (taking images by moving a camera side to side with minimal rotation), and the images were as coplanar as possible to each other with human error. However, the bulk of my work revolved around the math in Figure 1, which turned out to be (theoretically) wrong.

4 Milestones

I had set out many tasks to complete throughout the quarter via milestones. However, I think most of them were incomplete – a lot of the problems I tried to solve were either confusing (because I did not understand how to tackle the problem) or intimidating in the sense that I was (admittedly) afraid to try and go down a wrong path. In my original project proposal, I had read [3] and became obsessed with image rectification after I had initially done depth estimation tests with the nearer and farther street lights as shown in Figure 2. This stemmed from reading [3], in which they used an image rectification technique called Zhang's method [6]; Zhang's method is something I had not used before in a previous computer vision class.

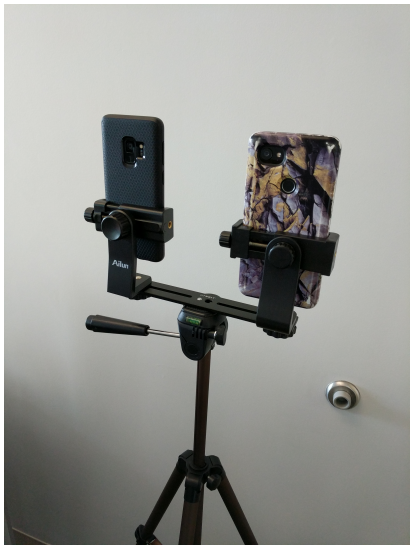

Figure 3: Smartphone stereo setup for testing purposes

So I started to go down the route of image rectification but did not get far – I was very lost in what I had to do and did not want to rely on my code from CSE 252A because (I think/thought) there were issues with it. However, I admit that I should have just tried it in the end. But before I could dive into that, I realized that one of my cameras has distortion issues, and I got distracted in fixing that; at the time, I was using a smartphone and an off-the-shelf camera purely for testing/working purposes. This was the main reason that made me realize that I went off on a tangent – *we need to get the system simply working!* Dealing with all these small issues is not the primary focus of my research project, and I need to ignore that and use equipment that will make my life easier by not dealing with all the hard nuances with our intended final system. The undistortion technique I used from OpenCV was not perfect, and it caused more problems that I would have to fix in order to incorporate an off-the-shelf camera into my testing.

After I realized that I started going down a rabbit hole of problems I was trying to solve, I realized that I needed to focus on my original goals, which was depth/GPS estimation of features in images. So for my milestone report, I proposed a redirection from my original milestone goals by doing an in depth study of depth estimation with ground truth depth measurements; I conducted a test with the setup shown in Figure 3. For this experiment, I used the tiled floor in a CSE hallway – each tile is a square foot, so I could easily count/measure the distance away from the camera to the intended object, which was just a flashlight; the

Table 1: Reference to First Image, Nearer Street Light Distance Estimations (in Meters)

	Img_1 and Img_2	Img_1 and Img_3	Img_1 and Img_4	Img_1 and Img_5	Img_1 and Img_6	Img_1 and Img_7	Img_1 and Img_8	Img_1 and Img_9
Daytime Dataset	14.69	13.65	13.67	13.5	13.2	13.19	13.30	13.37
Nighttime Dataset	17.12	15.71	14.41	11.56	11.83	11.99	12.16	12.2

Table 2: Reference to First Image, Farther Street Light Distance Estimations (in Meters)

	Img_1 and Img_2	Img_1 and Img_3	Img_1 and Img_4	Img_1 and Img_5	Img_1 and Img_6	Img_1 and Img_7	Img_1 and Img_8	Img_1 and Img_9
Daytime Dataset	32.02	30.09	29.5	28.71	27.38	27.65	28.29	28.46
Nighttime Dataset	49.95	40.94	32.86	27.15	30.76	27.65	28.02	27.98

center of the light is the feature for my tests. So, I took images in five foot increments away from the light (e.g. 5, 10, 15, ... , 90, 95, 100), and then recorded X_L and X_R of each image. However, I noticed a (lack of a) trend when recording these values. Calculating X_L and X_R and subtracting X_R from X_L (e.g. X_L - X_R) is actually calculating the disparity. I observed that when the images are taken from afar, human error will affect this disparity measurement; the differences in disparity between farther distances (e.g. 90, 95) will be mere pixels; human error from manually extracting the column pixel values in GIMP will completely mess that up. As a result, I realized that my proposed depth estimation study with ground truth depth distances was also a flop; in the real world, we should expect street lights to be between 10-30 meters away from the camera (100 feet is about 30 meters), and I realized that this (manual) study I proposed would do me no good. So instead, I relied on the original dataset from early on in the quarter, as shown in Figure 2, to complete my measurement study.

As a result, most of my milestones were incomplete, and semi-completed my depth estimation study proposed in my milestone report; image rectification and GPS coordinates were not touched. That being said, after chatting with Ben, I have some future directions on how to do things the right way.

5 Challenges

I encountered many challenges throughout this quarter, all of them being unexpected through my own naivete. In this section, I will go through all of the challenges I encountered unexpectedly and personally as well.

5.1 Coincidentally Right

After speaking with Ben, I found that all of my initial results were incorrect – for about a month, I led myself down the wrong path, thinking that I was on the right track. The results from Tables 1-4 show the depth estimations from pairs of images. Overall, the values showed consistency, which was something that threw me off.

Somehow, I did not realize the really high numbers for the farther street light depth estimation via the consecutive images method in Table 4, but the estimations overall seemed consistent; while I did not measure how these estimations compared to the ground truth originally, I found that statistics of error and percent

Table 3: Consecutive Images, Nearer Street Light Distance Estimations (in Meters)

	Img_1 and Img_2	Img_2 and Img_3	Img_3 and Img_4	Img_4 and Img_5	Img_5 and Img_6	Img_6 and Img_7	Img_7 and Img_8	Img_8 and Img_9
Daytime Dataset	14.69	12.74	13.72	13.01	12.12	13.14	14.03	13.87
Nighttime Dataset	17.11	15.71	14.41	11.56	11.82	11.99	12.16	12.2

Table 4: Consecutive Images, Farther Street Light Distance Estimations (in Meters)

	Img_1 and Img_2	Img_2 and Img_3	Img_3 and Img_4	Img_4 and Img_5	Img_5 and Img_6	Img_6 and Img_7	Img_7 and Img_8	Img_8 and Img_9
Daytime Dataset	32.02	28.38	28.38	26.57	23.13	29.04	32.86	29.73
Nighttime Dataset	49.95	34.69	23.56	17.84	65.73	18.36	30.46	27.75

error would have further convinced myself that the method was working properly. Thus, while my preliminary results appeared to be correct, they were wrong (or inaccurate) in the end.

I also realized that there were issues with this method regarding the focal length, as mentioned in Section 3. Coincidentally, I feel like a lot of things lined up, which tricked me into thinking that I knew what I was doing when in reality I got really lucky; I was (once again, more or less) blindly coming up with the next steps to solve.

5.2 Cameras on a Moving Vehicle

I massively underestimated difficulties with strapping this stereo camera system onto a moving vehicle, needless to say the difficulties that will occur when incorporating the other part of this research into this camera system. The main issue that I realized with taking stereo images on a moving vehicle is camera synchronization – images need to be taken at precisely the same time or else the pairs will be mismatched. Synchronization between two off-the-shelf cameras that can plug into a board might be feasible, but synchronizing two mobile phone cameras might be difficult; I have never done so, nor do I even know how I could do so. To accomplish this with mobile cameras, we will likely need an Android application, which will take more time to develop. A future approach might be to timestamp each image, and pair images with the closest time stamps. However, I currently do not have a solution to this problem yet.

5.3 "Team...?"

The subsection title is a joke towards modern gamers who might blame their teammates for failing/losing, but I have no one to blame but myself. There are many things I was disappointed in, the main being that I felt like I could have done more/better – there were certainly a lot of things I did not even try this quarter, and I feel abashed by that. I could make excuses saying that I might have been slightly burnt out earlier in the quarter, making me feel fearful of failure, but I know what I did and did not do in the end. Throughout the entire quarter, I essentially tried reinventing the wheel, or tried learning how to reinvent the wheel – I should have stuck more with relying on the expert (Ben), but I felt bad about reaching out to him all the time that I decided to take matters into my own hands. Especially after my preliminary depth estimation results, I felt confident with my knowledge and my ability to solve this problem. However, with paper submissions coming up, I somehow became afraid to try things; I had homework solutions for my CSE 252A class that I could have used for this class, but I was reluctant to do so! I honestly do not know what happened this quarter with me, but I became obsessed with image rectification, yet had trouble just setting forth an approach and attempting that approach.

So with the busyness of different paper submissions and the fear of failing, I felt like I accomplished very little; the part that hurts me the most is that I felt like I had wasted more time, like I had done so my senior year

of college. Computer vision has definitely been the bane of this entire project, and here I am still struggling about a year and a half later. The future hope that I have is that Ben agreed to help me informally for now – expert advice is necessary. I believe that the learning curve to become familiar with and explore more computer vision concepts is a bit steep. Sure, I admit to the fact that I did not read a single chapter from either textbooks in CSE 252A/B (oops), but I realized that that was probably one of the biggest mistakes I made as well.

From talking with Ben, I now have future directions, and I intend to catch up on some reading to fill in many of the gaps that I am missing with my computer vision understanding.

6 Conclusion and Project Takeaways

In conclusion, I conducted a basic depth estimation study via stereo vision. I found that my results were inaccurate, and that doing stereo vision the right way requires much more programming than manual work. All the challenges that I faced can be found in Section 5. In the end, I felt like the entire project was a flop – perhaps I should have followed exactly what the Columbia folks were doing, especially since I am in no position to direct myself in what steps I should take next to solve this problem. But also, I felt that I definitely got in over my own head – perhaps this was due to my preliminary results looking promising, but failing a lot this quarter has taught me to take a step back and not get too over my own head.

For advice to future students who are taking 237D for research purposes or for doing a project alone – something that really hit me in the end is that everyone had beautiful projects, and I had practically nothing. It's easier said than done, but I would say to not get discouraged. I was definitely discouraged a lot, but being a one person team means that the end result will likely be less than what a team of 2-3 students produce, and research will often produce negative results, which is something I am still getting used to.

Overall, I feel like I made okay use of my time this quarter – I definitely learned a lot about the problems with stereo vision/stereo cameras for my project, so maybe that's one of the main takeaways for myself. I also learned a lot about myself in terms of what I'm capable of, what I'm capable of, and when I need to get help. So overall, while I feel like this quarter has been a negative experience, it's probably a positive experience in reality for myself personally (I'm just having a hard time accepting that right now).

References

[1] Mark Sheinin, Yoav Y. Schechner, and Kiriakos N. Kutulakos. Rolling shutter imaging on the electric grid. In *2018 IEEE International Conference on Computational Photography (ICCP)*, pages 1–12, 2018.

[2] Zeal Shah, Alex Yen, Ajey Pandey, and Jay Taneja. GridInSight. In *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. ACM, November 2019.

[3] Weifan Jiang, Vivek Kumar, Nikhil Mehta, Jack Bott, and Vijay Modi. Irrigation Detection by Car: Computer Vision and Sensing for the Detection and Geolocation of Irrigated and Non-irrigated Farmland. https://qsel.columbia.edu/assets/uploads/blog/2020/publications/irrigation-detection-by-car-ghtc-fall-2020.pdf. Online; accessed 14-April-2021.

[4] StereoPi. https://stereopi.com/. Online; accessed 09-June-2021.

[5] StereoPi. https://github.com/realizator/stereopi-fisheye-robot. Online; accessed 09-June-2021.

[6] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.