

Fast, Automated Indoor Light Detection, Classification, and Measurement

Craig Hiller



Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2016-212

<http://www2.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-212.html>

December 16, 2016

Copyright © 2016, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

Acknowledgement

I would like to thank my parents for their constant support and guidance throughout all of my endeavors.

**Fast, Automated Indoor Light Detection, Classification, and
Measurement**

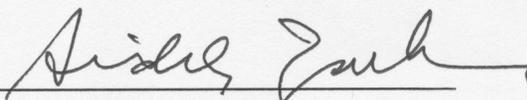
by Craig Mitchell Hiller

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences,
University of California at Berkeley, in partial satisfaction of the requirements for the
degree of **Master of Science, Plan II.**

Approval for the Report and Comprehensive Examination:

Committee:



Professor A. Zakhor
Research Advisor

12/15/16

(Date)



Professor A. von Meier
Second Reader

Dec 15, 2016

(Date)

Fast, Automated Indoor Light Detection, Classification, and Measurement

Craig Hiller

University of California, Berkeley

December 15, 2016

Abstract

Lighting is one of the largest uses of energy in buildings in the United States and around the world. A building lighting audit is one step to reducing the total energy consumption of a facility. This report presents a fast and automated method to detect, classify, and measure lighting in buildings without installation of any monitoring devices and without a building floor plan. With this system, the manual, tedious, and error-prone process of counting and marking all lights on a pre-given floor plan is replaced by two quick walkthroughs of the area with a custom collection device. The device consists of a ceiling-facing Canon DSLR camera, a Google Project Tango tablet, and an Ocean Optics Spark-VIS spectrometer, all mounted together in an easy to carry package. Using the data collected with this device, lights are detected through a series of image processing operations on pictures acquired by the camera and then tracked through multiple frames. Using the precise position and pose information from the Tango tablet and correspondences between sequential images, the dimensions of each light are estimated. Finally, using a neural network, the lights are classified based on their emission spectra. Operation was tested with four data sets recorded in Cory Hall and the Valley Life Sciences Building on the University of California, Berkeley campus. This system detected lights with a 7% error rate, classified them with a 14% error rate, and estimated surface area within a factor of two.

1 Introduction

Lighting is one of the largest consumers of power in buildings around the world. In California, interior commercial lights use more than 25% of a building's total energy [1]. Therefore, when creating a model of a building's energy use in tools such as EnergyPlus [2], it is imperative that lighting be taken into account. This project aims to automate the counting, classification and estimation of the lights' physical dimensions in a building to assist with energy modeling.

Current methods of light auditing start with the floor plan and then require an auditor to walk through the area to manually classify, count, and mark the lights on the floor plan [3]. This is a lengthy and error-prone process. Snapcount [4], a leader in this field, provides a tablet application to mark the locations of the lights on the digitized floor plan, and the ability to take a photograph of the light and record audio notes about it. Even with this software, light auditing is a very laborious process since the operator still needs to determine where the light is on the floor plan. For a large building, this process can take more than a day to complete [5].

Another approach used to perform lighting audits is to install an energy monitoring device to record and compare the power usage when the lights are on versus off. This method, detailed in Jain et. al. [6], requires invasive installation changes to the building's electrical system. Once the power usage for the lights is estimated, night and day power consumption is compared and the auditors can make recommendations on energy-saving actions.

Prior work in automated object-based building energy modeling includes window detection [7] and computer detection [8]. In [7], windows are detected in point clouds constructed from multiple LiDAR sensors and cameras mounted on a human carried and operated backpack system [9]. To do this, first the walls are detected and extracted from the 3D point clouds generated by the system. Then, each wall is processed through feature extractors based on multiple assumptions about windows. Finally, using a Markov Random Field and maximum a posteriori (MAP) inference, labels of wall or window are generated. Using the same hardware system, the authors in [8] de-

tect computers and calculate energy consumption through IR images, surface normal maps, visible light images, and depth images. The images are processed through selective search [10] and then a regional-convolutional neural network [11] is used to label the objects in the images. Power is estimated through the IR images and a support vector regressor.

Laskowski [12] demonstrates a method of detecting light sources in images through finding points of maximal luminance in a photograph, then reducing the number of points through distance metrics and similarity to other potential light sources in the image. While this approach works well for finding bright light sources in typical scenes, such techniques are not necessary when scene exposure can be controlled.

Venable et. al. [13] describe a process for an unmanned aerial vehicle (UAV) carried camera, spectrometer, and distance sensor to autonomously detect and classify lights in a region inside of a building. Lights are found using thresholding and other image processing techniques. To estimate sizes of the lights, the distance from UAV to floor is used as the ceilings are fixed in height. Bay and Terrill et. al. [14,15] describe the fundamentals of this work and additionally create a stitched image of the entire ceiling. The lights are then counted on this stitched image. The authors note that global positions of the lights cannot be computed using this method. Similar approaches are used in this paper for filtering the raw images of lights and spectra classification.

While the promise of fully autonomous light auditing is exciting, there are many advantages to a human carried portable system. UAVs are constrained by load carrying capacity, limiting camera quality and system battery life. A portable, human operated system is less constrained by battery capacity and weight resulting in longer duration collection times. In addition, a heavier camera results in higher quality imagery which aids detection. The camera used in this project, a professional level digital single-lens reflex (DSLR), has much better optics, precise exposure control, and very low image noise. The human carried device ensures that the entire area is covered, even portions that would be impassable or difficult for a flying craft. Such portions include doorways,

both open and closed, narrow passageways, and places with variable height ceilings.

Since light sources are inherently the brightest objects in the scene, classical thresholding techniques for visible light imagery can be successfully used for detection. Specifically, since surface normal maps and point clouds are not necessary for light detection, the expensive LiDAR and camera systems used previously for window and plug load detection [7, 8] can be avoided. Rather, a single commercially available camera, spectrometer, and localization device can be combined into a portable unit to be carried by an ambulatory human operator.

This report describes a system to automate the light audit task. Using this system, all the lights can be counted, located, and classified by walking the area at a normal pace without installing any additional monitoring devices to the building. By dramatically speeding up the audit process compared to the traditional clipboard and notepad method, audit costs should decrease. If the audit recommendations are followed, building owners should see a faster payback on their investment. Ultimately, this is an economic win for the owners, and an environmental win through reduced carbon emission. Additionally, a floor plan or map of the area is not needed to mark lights on since one can be generated on the fly thanks to Google's Project Tango tablet. This tablet is a low-cost, commercially available Simultaneous Localization And Mapping (SLAM) system. It provides precise pose and location information, used by the other algorithms described in this report to locate lights accurately within an area. Since precise device location and orientation at every instant is known, a distance sensor to measure the height of the room is not needed for light size estimates. Hence, the system is able to work in situations with varying ceiling heights without tuning.

This report is organized into sections detailing each aspect of the work. Section 2 describes the custom-built hardware and data collection methods used in this project. Section 3 dives into the data processing algorithms and procedures for detecting, counting, and classifying lights. Then, in Section 4 details of the four collected datasets are presented. Results are presented in Section 5 and finally suggestions for improvements are in Section 6.



Figure 1: Data Collection Platform

2 Hardware and Data Collection

2.1 Hardware

In this section, an overview of the hardware system developed is given along with the data collection procedures. Since no existing platform combines a localization system with a ceiling-facing camera and a spectrometer, a system was built to combine these three elements, shown in Figure 1. For localization, a Google Project Tango tablet device [16] was used. This provided a small, lightweight form factor for determining orientation and location of the system as the human operator walks and controls the other components. A Canon EOS 5D Mark III DSLR camera [17] with a 16-35mm f/2.8 lens [18] was used to capture imagery. This was chosen because it can capture images at a high frame rate, up to 6 frames per second, until its memory card is full [19], and can be triggered externally. A 128 GB memory card will provide approximately 6 hours of recording time of 1 MB images. The wide angle lens was used at the 16mm setting to provide a 96.7-degree horizontal field of view and a 73.7-degree vertical field of view, which was sufficient for this application. As depicted in Figure 1, the camera and lens were positioned upwards to face the ceiling. The spectrometer was an Ocean Optics Spark-VIS [20] with a diffuser on top. This sensor has a narrow field of view and only accurately reads a spectrum when underneath the light

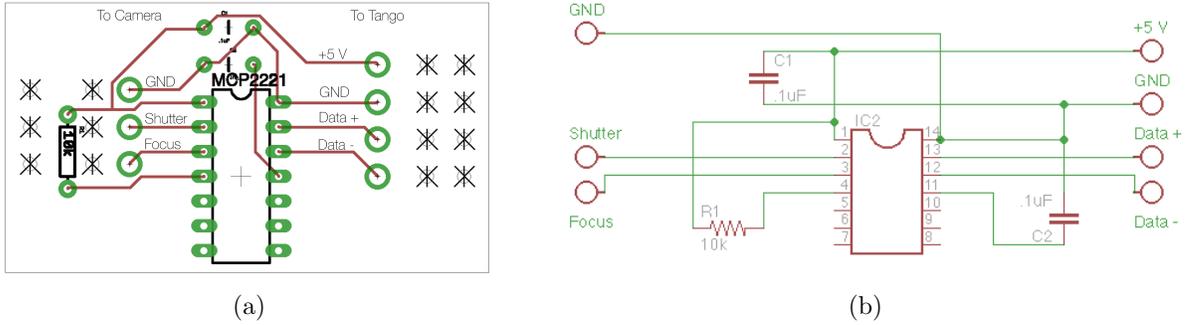


Figure 2: Camera controller board. (a) Board layout (b) Schematic

fixture of interest. These three devices were placed together on a wooden board and secured rigidly.

Since the camera stored all of its images to its own memory card, all timestamps associated with the images were based on the camera’s internal clock. However, the spectrometer was connected directly to the Tango device, so the clock on the Tango was used for time-stamping the spectrometer measurements. In order to synchronize the clocks between the Tango and the camera, the small circuit shown in Figure 2 based on the Microchip MCP221 USB Bridge [21] was built. This enabled the Tango to issue a USB command to the circuit which triggered the camera through its remote shutter release input at a known point in time. To detect lights, the fact that lights are significantly brighter than their surrounding areas was exploited and images were captured using a very low exposure value (EV). The camera settings used were a 1/4000 second shutter speed, f/22 aperture, and ISO 3200 sensitivity. By having such a fast shutter speed and small aperture, motion blur and focus issues were eliminated. There was no significant noise in the images at ISO 3200, so they were easily useable.

2.2 Data Collection

The location data is collected in two phases with the Google Project Tango tablet. This is a commercially available hardware device which can localize itself within an area and automatically output the path the tablet travels. Since the position measurements are outputted directly in meters, the detected light locations may be superimposed on the path, or floor plan, so the operator

can tell exactly where the lights are in the building.

The first phase generates an Area Description File (ADF) using Google’s Explorer App [22]. The ADF contains a representation of visual landmarks within a given area. These landmarks are stored so that when the file is loaded, the Tango recognizes it has already been in that place giving a frame of reference for future position updates. This process records image features from the target area, localized to a specific point in space. By adding loops to the walked path, Tango can adjust for areas it has already seen, improving the quality of the localization. A loop in the path is created when the operator walks around an area and again returns to the same starting point. This allows new measurements and features to be adjusted to the rest of the dataset, minimizing drift. This enables the Tango more quickly and accurately localize itself while providing better pose estimations during the second phase, i.e. the actual data collection. Without the initial ADF generation phase, position drifts of multiple meters are possible. Since the position is used to localize and de-duplicate lights, inaccurate positions would make accurate light localization impossible.

Anytime after the ADF is generated, the operator can begin the second phase of data collection. A custom application, developed for this project, loads the ADF and connects to the spectrometer and camera controller. Once the device is localized to the ADF, the user can start recording data. When the user presses “Start Collecting” in the application, the camera is triggered to start capturing photographs at 6 Hz and the Tango begins recording pose information at 100 Hz and spectrum information at approximately 15 Hz. During this time, pose and spectral information are saved to data files on the Tango and the images are recorded to the camera’s memory card.

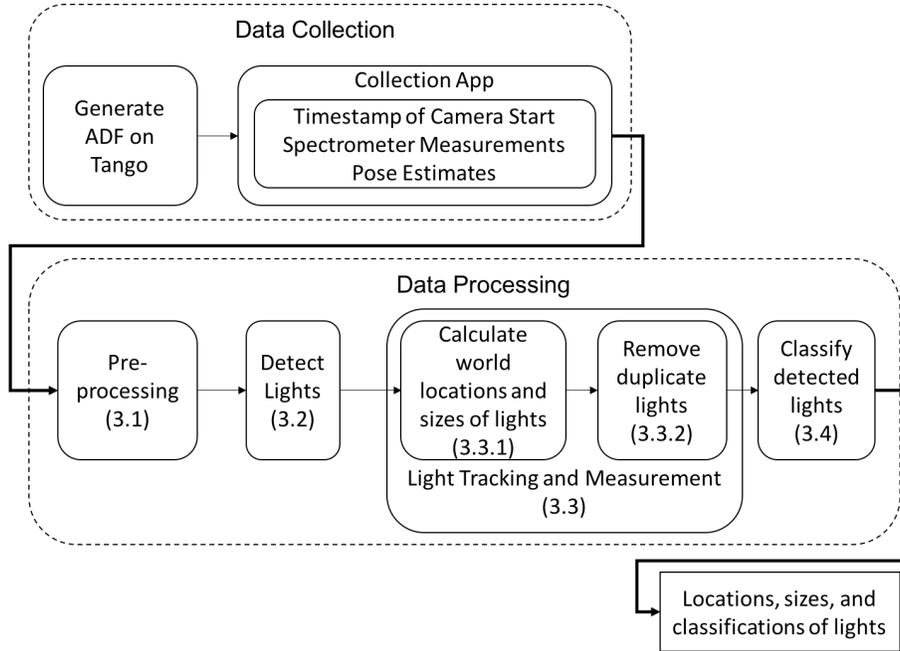


Figure 3: High-level diagram of the system. (x.y) refers to the section which describes that step.

3 Data Processing

A high level overview of the collection and data processing steps is shown in Figure 3. This section describes how all of the data are processed to accurately detect, locate, and classify lights.

3.1 Pre-processing

After all of the data is collected, the camera trigger time, spectrometer and pose data files from the Tango are converted into CSV format. The images are imported from the camera and resized to 1500×1000 pixels through ImageMagick and GNU Parallel [23]. In order to locate where the images were taken, first the timestamps between the camera and the Tango are aligned. To do this, the timestamp from the Tango when the camera was first triggered is used and then a 61 milliseconds shutter lag between when the shutter is pressed and the first image is taken is factored in. This shutter lag time is an intrinsic property of the camera, which has been well documented [24]. The EXIF information for each image provides timestamps with 10 millisecond precision. Using this timestamp, offset by the time difference computed in the previous step, each image is

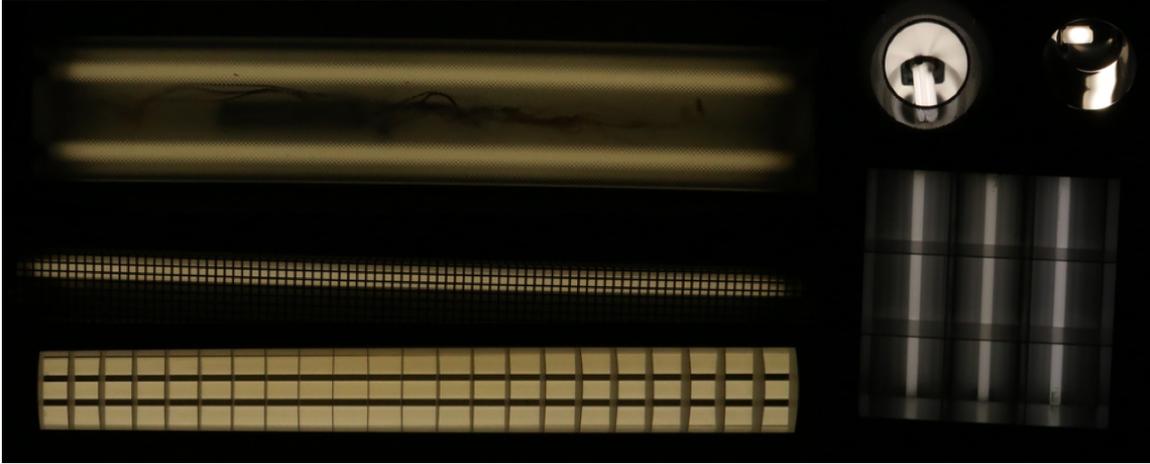


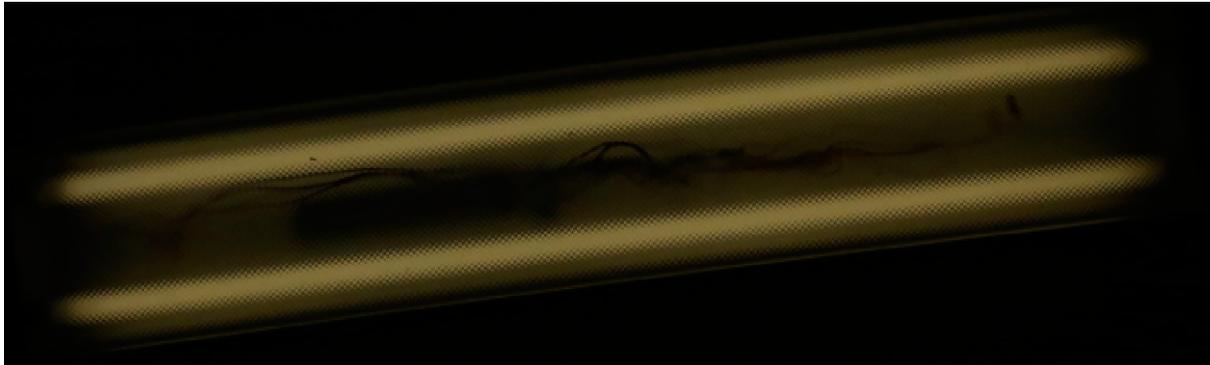
Figure 4: Examples of different light fixtures in the collected datasets.

matched to a pose recorded by the Tango by using the measurement nearest in time to the specific image. Similarly, the spectrometer measurements are associated with poses in the same way, but no temporal adjustment is needed since they were recorded by the Tango, and that is the time reference for the system.

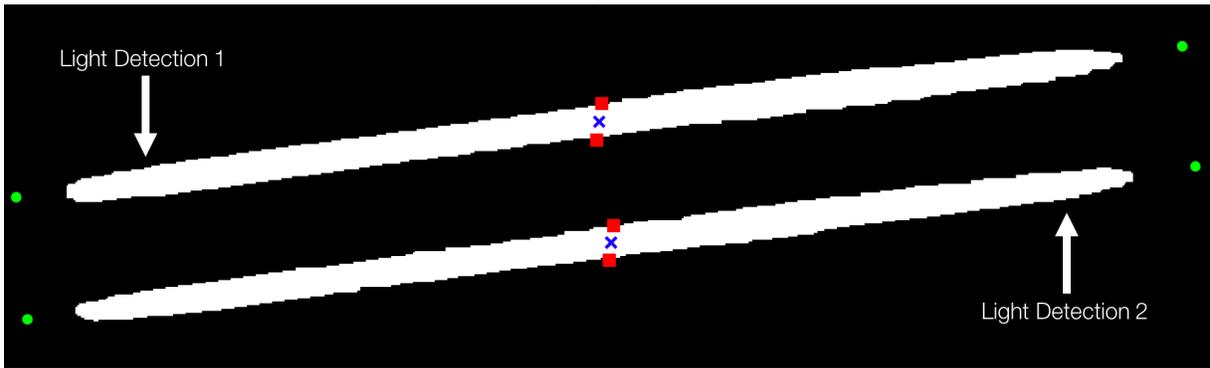
3.2 Light Detection

Since the images were captured with a low EV, lights can be detected simply by thresholding the images. However, due to variation in lighting fixtures, some of the bulbs may be partially obstructed or have bright reflections, as shown in Figure 4. To account for this, morphological operations to fill small gaps across the face of a light bulb are applied. The operations are, in order:

1. Opening with a 3×3 pixel rectangle
2. Closing with a 20×20 pixel rectangle
3. Filling of holes
4. Dilation with a 30×30 pixel rectangle
5. Erosion with a 30×30 pixel rectangle



(a)



(b)

Figure 5: Light detection example. (a) Input image of light - Fluorescent tubes 3rd Floor Cory Hall (b) Light mask shown with detected lights from blob analyzer. Centroids (blue x), end points of major axes (green dots), and minor axes (red squares) are marked for each light

The results of these operations are shown in Figure 5b as the underlying black and white image. There are some instances, however, where these operations will merge adjacent bulbs as in Figure 6. Therefore, there is no distinction between bulb and fixture. The resulting binary image from these operations is a mask of where the lights are, white pixel, and are not, black pixel. This mask is run through Matlab's blob analyzer function to compute its connected regions. Each of these regions is then treated individually as a separate light. For each detected blob, the centroid, orientation angle, and length of major and minor axis are returned. An example of the masking and blob detection is shown in Figure 5. If any part of the blob touches the boundary of the image, the blob is ignored and not used in tracking. This ensures that entire light is visible and captured by the blob detector.

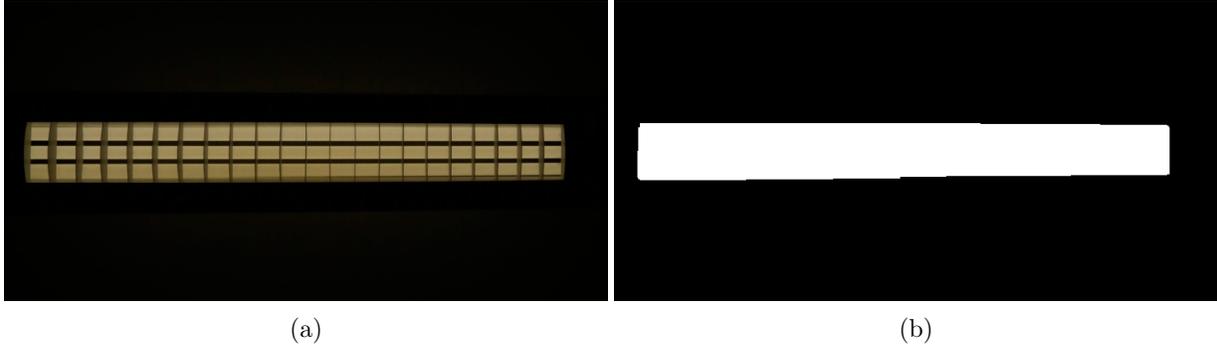


Figure 6: Merged light mask example. (a) Input image of light - Fluorescent tube fixture 3rd Floor Cory Hall with three tubes (b) Light mask result from thresholding and morphological operations

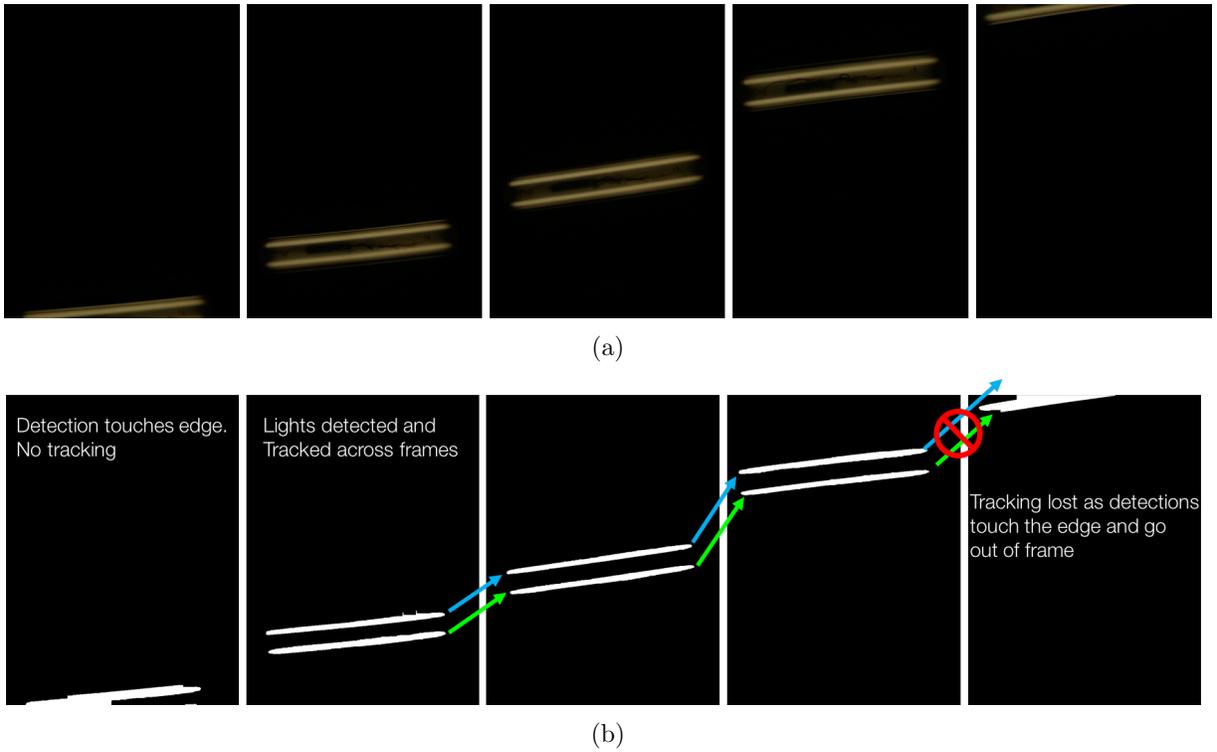


Figure 7: Light tracking example. (a) Sequence of raw images of lights. Sequence is shortened from the original 15 frames. (b) Detected lights and their tracks.

3.3 Light Tracking and Measurement

Since images are being captured at approximately six frames per second, each light will appear in multiple frames, shown in Figure 7a. In order to avoid over counting lights, each light detection from frame to frame is tracked using the approach specified in [25]. A Kalman filter is used to

track the lights between successive frames. Then, using the position and orientation information from the Tango tablet, the real world position and size of the lights are estimated as described in Section 3.3.1. Finally, since it is possible that the same light was visible multiple times throughout the data collection, a de-duplication procedure described in Section 3.3.2 is run to yield the final light detections.

Given the centroid of the light in pixel space, a Kalman filter is used to predict the location of the next detection of that light [25]. This filter takes prior centroid information and predicts a new location based on a constant velocity model. Then, the prediction is compared to the detected centroid of the next frame and the estimate is updated. The resulting position estimate is used for future iterations of the filter and is also stored as the centroid for the light. To implement this, Matlab’s constant velocity Kalman filter is used. It is configured with the first detected light centroid location, an initial estimate error with a 200 pixel location variance and a 50 pixel velocity variance, a motion noise with a 100 pixel location variance and a 50 pixel velocity variance, and a 100 pixel measurement noise. These values were used in Matlab’s tracking example and were found to perform well in the light tracking use case. Figure 7b shows the results of how lights are tracked.

Then, using Munkres’ variant of the Hungarian assignment algorithm [26], the predicted centroids from the Kalman filter are matched to the centroids in the input image. This algorithm yields an optimal assignment in time $O(n^3)$ where n is the number of lights to match between the frames. Munkres’ algorithm runs on an $n \times n$ matrix of costs, in this case, between the predicted lights from the Kalman filter and the newly detected lights. The cost for each assignment is the distance, in pixels, between the predicted centroid and the detected light’s centroid. However, there may not always be the same number of detected and predicted lights. This can occur when a new light comes into the frame or a light leaves the frame. To account for these situations, additional rows or columns are added with a high cost, 1000 pixels. If a light is not currently being tracked, or at any point leaves the frame, it is counted as a new light. The tracks are later analyzed in the de-duplication process, as described in Section 3.3.2, to detect any possible duplicates and the

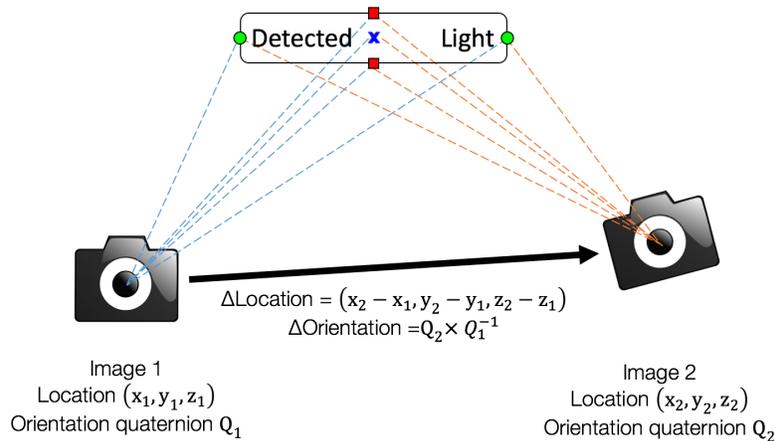


Figure 8: Diagram of stereo setup with two images. The corresponding points on the detected light are marked the same as in Figure 5b. Camera drawing from [27].

associated information consolidated.

3.3.1 Light Localizing and Sizing

A key aspect of this project was to determine the location of the lights are in a building. In order to provide the location of a light in real-world space, rather than in image space only, every pair of images where the light was identified was selected and the centroids of the detected light in each image used as corresponding points. Since the change in pose between the two images can be computed through subtraction and quaternion multiplication, respectively, this information can be used to create a stereo baseline as shown in Figure 8. Using MATLAB’s triangulate function, with the pair of centroids and the camera’s calibration matrix, the coordinates of the light with respect to the camera are computed. Then, by adding in the camera’s location at that frame, the position of the light with respect to the original starting point is found. This calculation is performed over every pair of images containing the same light. As a result, for a light to be correctly detected and measured, it must be fully in the camera’s field of view for three or more frames.

Using this same triangulation function, the physical size of the light fixtures can be measured. Again, starting with the centroid of the detected light and assuming the lights as ellipses, the end

points of the light along the major and minor axis of the ellipse can be calculated. For a given detection with centroid $c = (c_x, c_y)$, orientation θ , major axis length M and minor axis length m , the four end points are:

$$\begin{aligned} & \left(c_x - \frac{M \cos(\theta)}{2}, c_y + \frac{M \sin(\theta)}{2} \right) \\ & \left(c_x + \frac{M \cos(\theta)}{2}, c_y - \frac{M \sin(\theta)}{2} \right) \\ & \left(c_x - \frac{m \sin(\theta)}{2}, c_y + \frac{m \cos(\theta)}{2} \right) \\ & \left(c_x + \frac{m \sin(\theta)}{2}, c_y - \frac{m \cos(\theta)}{2} \right) \end{aligned}$$

Figure 5 shows these points overlaid on the detected light masks. These four points are then run through the same triangulation function as before and the Euclidean distances between the points along the major and minor axes are found. These distances are the real world length of each axis in meters. After processing all of the images for each light, the median major axis and minor axis lengths are returned as the estimated size of the light.

3.3.2 De-duplication of Lights

During the course of walking through a building, the same light may be encountered in non-successive frames. Heuristics are used to remove these duplicates. If two detections are within one meter of each other, they are either two bulbs in the same fixture or one is a duplicate of the other. Bulbs in the same fixture will appear together in subsequent frames whereas duplicates will not. These duplicates are identified as such and the associated images are treated as additional tracked frames of the original light to aid classification and measurement. Then, a second heuristic removes false detections which do not correspond to a light at all. This leverages the fact that, in at least some frames, the operator must be underneath a light for it to be properly identified. Therefore, if the detected center is more than 200 pixels from the image center in all candidate frames, the detection is discarded.

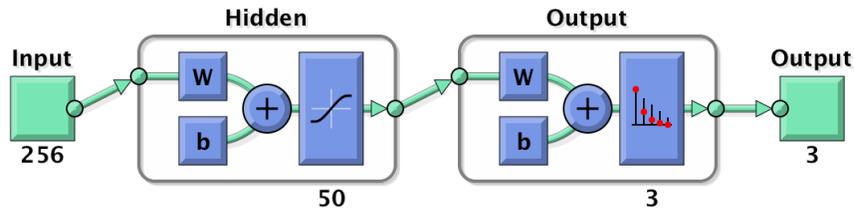


Figure 9: The neural network used for light spectrum classification.

3.4 Light Classification

The final aspect of this work is to classify the detected lights based on their light source: LED, fluorescent, or incandescent/halogen. The readings from the spectrometer are a 1024×1 vector of intensity values at varying wavelengths across the visible light spectrum. Since there can be slight variation in measurements of the same bulb, every four elements of this vector are averaged together, resulting in a 256×1 feature vector. This vector is used as input to a feed-forward neural network with a single hidden layer with 50 neurons, visualized in Figure 9. This network was trained on 7,037 spectra from multiple commercially available light bulbs of various styles, color temperatures, and brightness using MATLAB's scaled conjugate gradient neural net trainer. The network was trained for approximately 150 epochs, until 6 randomly selected validation elements were classified correctly. To avoid overfitting, different bulbs than present in the test buildings were used for training.

4 Datasets

This section introduces the four different datasets used for evaluation of this system. Details about each dataset’s lights and path length are in Table 1. The plots in Figures 10 and 11 show the collected paths and where lights were detected in each.

Dataset	# of Lights	Light Description	Path Length
Cory 325 Figs. 10a, 10b	4	LED bulbs	29 meters ^a
	2	Fluorescent soft white bulbs	
	2	Fluorescent daylight bulbs	
	2	Incandescent bulbs	
Cory 3rd Floor Figs. 10c, 10d	44	Four-foot long T-8 fluorescent tubes	300 meters
	46	Fluorescent light fixtures with up to 3 tubes ^b	
	2	Fluorescent spot lights	
Cory 5th Floor Figs. 11a, 11b	3	LED bulbs	173 meters
	6	Fluorescent bulbs	
	100	Two-foot long T-8 fluorescent tubes	
VLSB 4th Floor Figs. 11c, 11d	25	LED bulbs	435 meters
	83	Four-foot long T-8 fluorescent tubes	

Table 1: The four datasets collected for this project. Below each dataset name, Figs. x, y refer to x: path walked for collection, y: path with detected lights marked.

^aThe overall path length was 29 meters and each light was walked under twice to exemplify the de-duplication procedure.

^bThe multi-tube fixtures were each counted as a single large light because the tubes were very close together and the fixture baffles confounded the analysis.

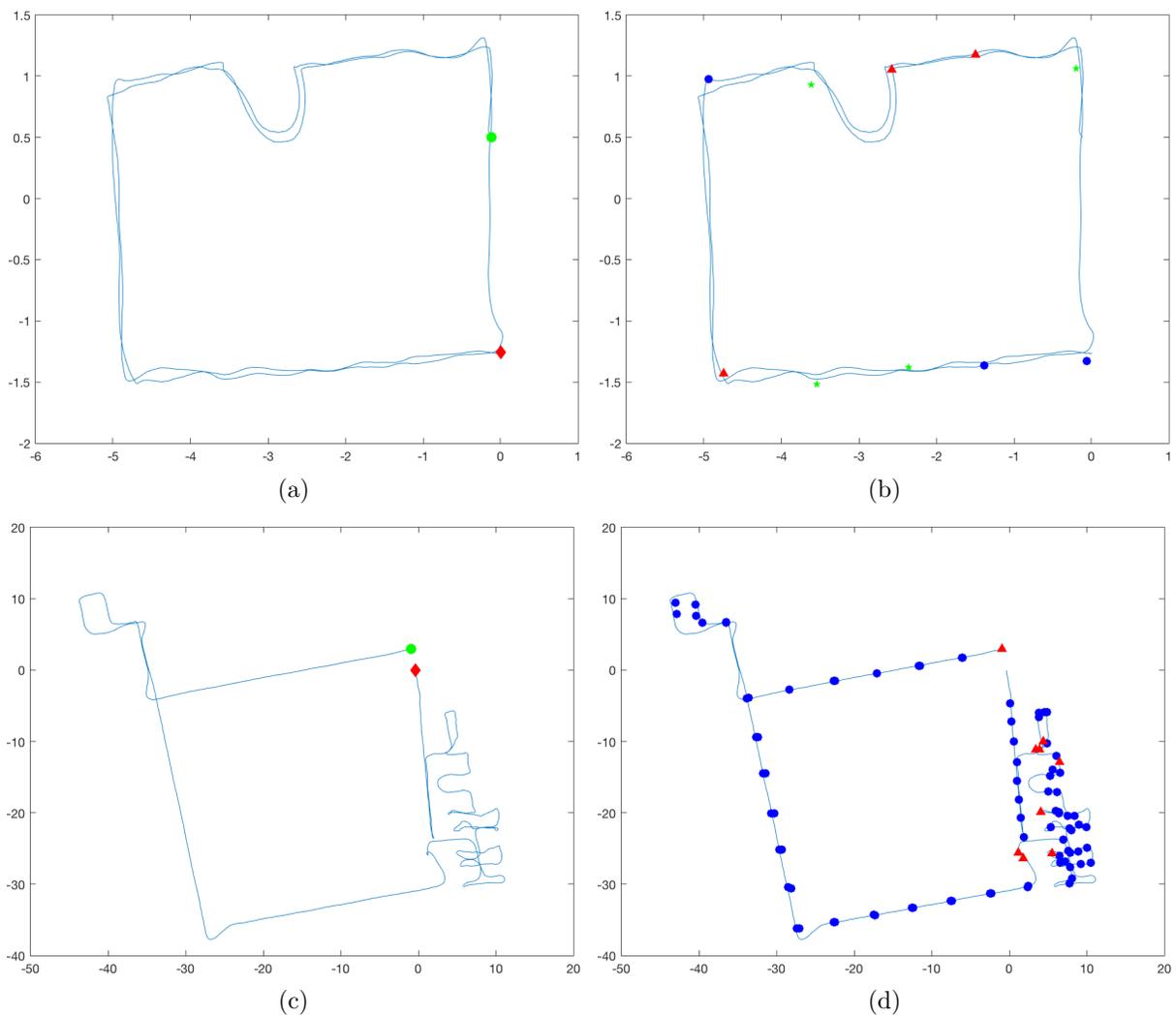


Figure 10: In the left plots (a, c), the green circle represents the start of the data collection path and the red diamond indicates the end of the path. The right plots (b, d) show detected lights. Fluorescent lights are marked with a blue circle, halogen/incandescent with a red triangle, and LED with a green star. (a) and (b) show the 325 Cory Hall dataset and (c) and (d) are 3rd floor Cory.

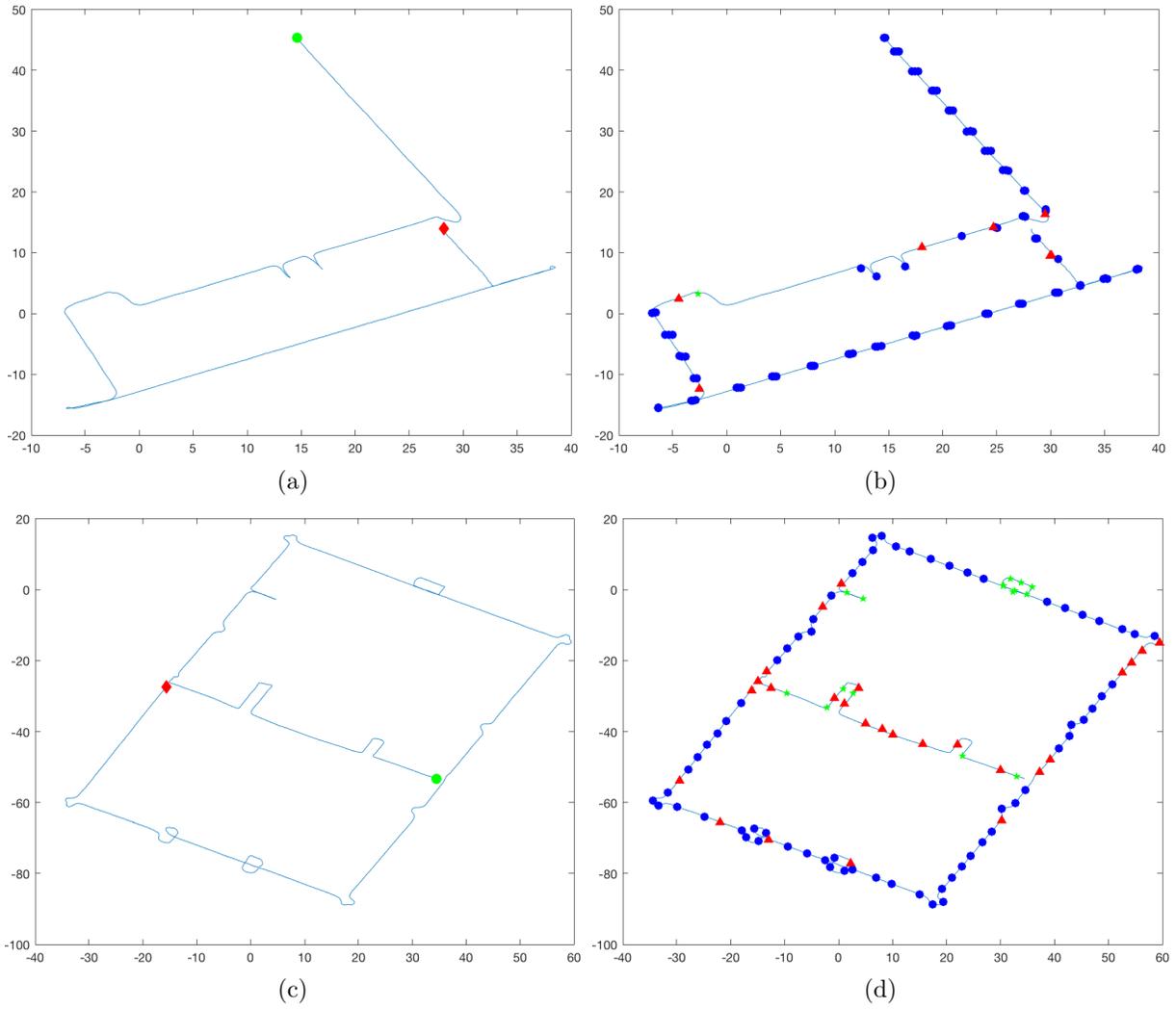


Figure 11: As in Figure 10 the left plots are the paths walked and the right plots show the detected lights. (a) and (b) are the 5th floor Cory dataset and (c) and (d) are 4th floor VLSB.

5 Results

The results for each of the problems addressed by this report are summarized in the following sections.

5.1 Detection

The results of the light detection algorithm across all datasets are shown in the Table 2. The algorithmically detected lights are those identified after all tracking and de-duplication procedures. For example, Figure 12a is the manually collected ground truth for the Cory 5th dataset showing rough locations of the lights, similar to what an auditor would create. The positions of the algorithmically detected lights in Figure 12b were visually compared against the ground truth to generate Table 2. A correct detection is defined as a visual correspondence with the ground truth. In the Cory 3rd dataset, error is higher due to maneuvering around office furniture. This is discussed further in Section 6. Error is calculated as:

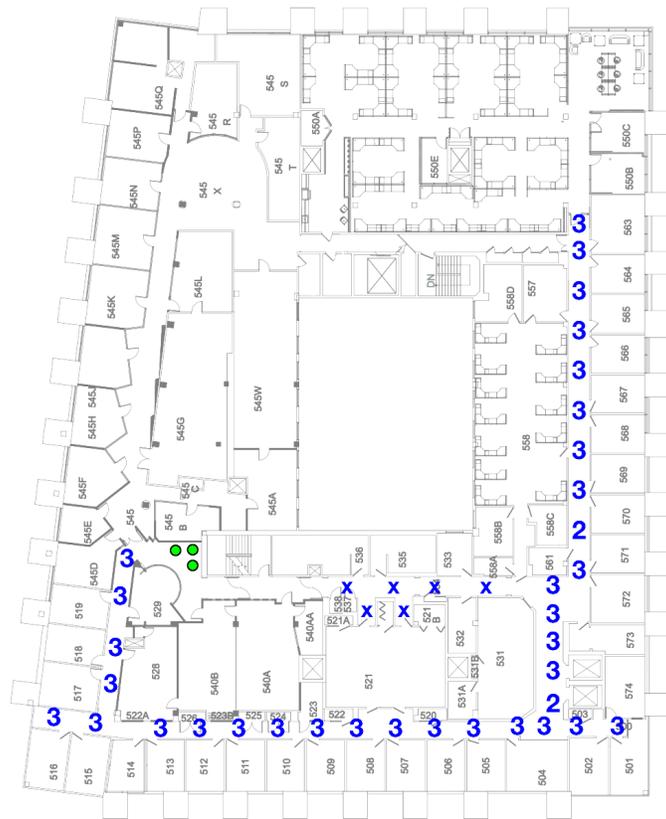
$$\text{Detection error rate} = \frac{\text{Missed Lights} + \text{False Positives}}{\text{Manually Counted Lights}}$$

Dataset	Manually Counted Lights	Algorithmically Detected Lights	Missed Lights	False Positives	Error (%)
Cory 325	10	10	0	0	0
Cory 3rd	92	93	6	7	14.1
Cory 5th	109	104	6	1	6.4
VLSB 4th	108	110	0	2	1.9
Total	319	317	12	10	6.9

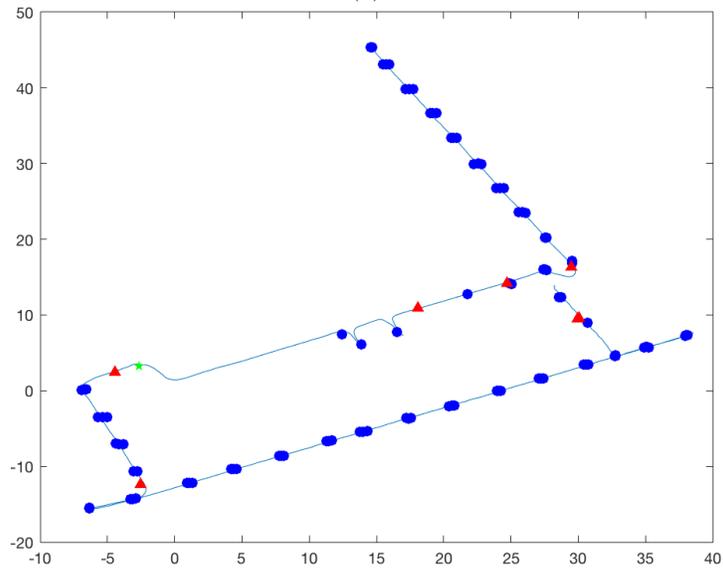
Table 2: Light detection results

5.2 Classification

Classification accuracy is based on the lights that were algorithmically detected, even if detected incorrectly. Results are shown in Table 3. Accuracy for the VLSB dataset is lower due to many of the lights being further away from the spectrometer due to higher ceilings as well as being near large windows during daylight. In addition to being distant from the spectrometer, these particular bulbs were smaller than typical, further hampering classification. The other datasets were collected



(a)



(b)

Figure 12: (a) manually labeled ground truth data for the 5th floor Cory dataset. Blue numbers indicate the number of fluorescent tubes at that location and a blue x indicates a fluorescent bulb. The green circles are locations of LED bulbs. (b) same figure as 11b showing the detected lights for 5th floor Cory.

at times with less outside light. Error is computed as:

$$\text{Classification error rate} = \frac{|\text{Correctly Classified} - \text{Total Detections}|}{\text{Total Detections}}$$

Dataset	Total Detections	Correctly Classified	Error (%)
Cory 325	10	9	10
Cory 3rd	93	83	10.8
Cory 5th	104	97	6.7
VLSB 4th	110	84	23.6
Total	317	273	13.9

Table 3: Light classification results

5.3 Measurement

The surface area of each light was calculated as the product of the calculated width and height, the lengths of the major and minor axes. The comparison between actual surface area and calculated surface area is shown in Table 4. Error is computed as:

$$\text{Measurement error rate} = \frac{|\text{Surface Area Calculated} - \text{Surface Area Actual}|}{\text{Surface Area Actual}}$$

Dataset	Surface Area Actual (m ²)	Surface Area Calculated (m ²)	Error (%)
Cory 325	0.049	0.032	34.7
Cory 3rd	6.80	16.74	146
Cory 5th	1.58	2.59	63.9
VLSB 4th	2.64	3.36	27.3
Total	11.06	22.72	105

Table 4: Light measurement results

A plot and histogram of the estimated sizes of the lights are shown in Figure 13. In the case of the third floor Cory dataset, three light measurements were greater than 2 m², causing the large error and therefore the large spikes in Figure 13. This was likely caused by difficulty maneuvering through the maze of cubicles in the room where this data set was collected. This limitation is further discussed in Section 6.

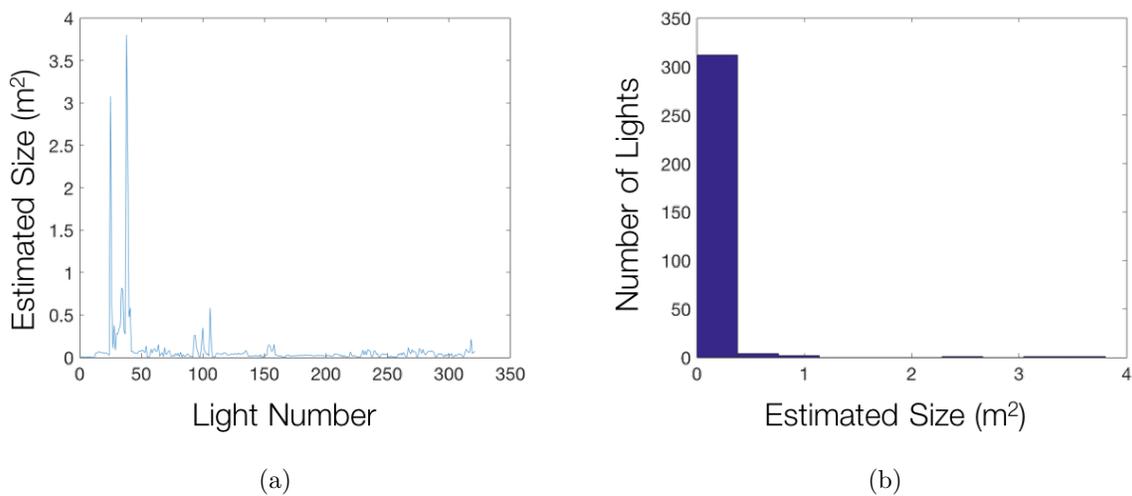


Figure 13: Estimated light sizes. (a) Plot of all lights and their estimated sizes (b) Histogram of light sizes

5.4 Visualization

A graphical user interface (GUI), screenshot in Figure 14, was developed to display all of the results from this project. In this GUI, the locations of all of the lights are displayed. When a light is clicked, the median spectrum and an image of the light is shown along with additional information such as surface area and dimensions.

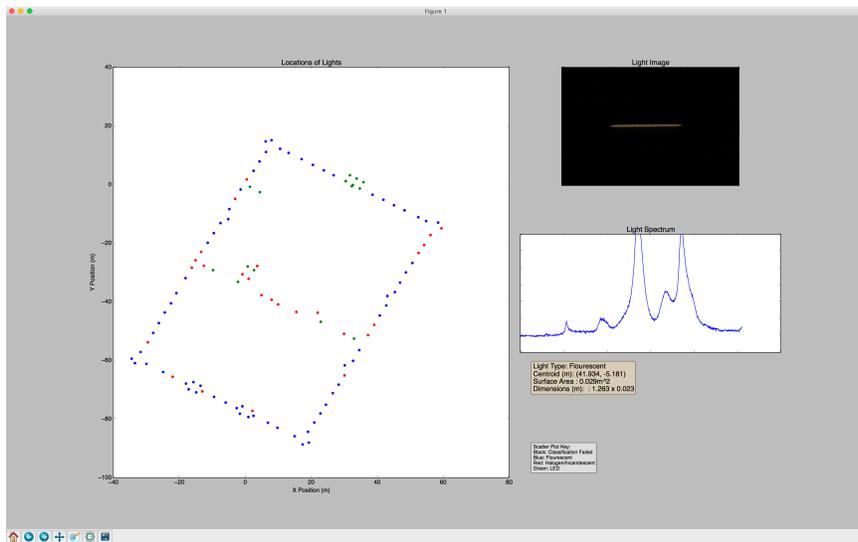


Figure 14: Visualization GUI for VLSB 4th Floor Dataset

6 Limitations and Future Improvement

Classification works best if the collection device is directly underneath a light. Therefore, for best results, the collection device must be maneuvered under each light. This is simple in open areas such as corridors or sparsely furnished rooms but difficult in the dense forest of office cubicles in the Trust and Chess work areas of 3rd floor Cory. In order to capture all of the lights and to ensure an accurate spectrum is recorded, the collection device must be held at arm's length to reach around the desks and filing cabinets. This is a less stable stance, resulting in some equipment shake. Since the clocks in the camera and Tango are only synchronized within 10 milliseconds, the pose at the moment the picture was taken could be significantly different from the pose assigned to that image. Additionally, even though an ADF was created for the Tango in the first phase of the collection process, when the device is in the same location, the poses may be slightly erroneous. These effects combined can cause lights to be detected in incorrect locations and, after the de-duplication procedures, have the algorithm return an incorrect number of lights.

To help mitigate these factors, a more stable and easier to maneuver platform could be used for collecting the data. Further, since better localization translates to better overall accuracy, walking extra loops during ADF generation improves the results.

While this project provides an estimate of total lighting surface area, type, and number of bulbs, the actual energy consumption of these could be determined with additional sensors, such as an infrared camera. A future extension of this work could involve additional sensors to yield more accurate detections and classifications, as well as measuring energy usage.

7 Conclusions

This report presents a method to detect, classify, and measure light bulbs/fixtures in buildings using commercially available hardware. The algorithm is accurate in detecting light sources, with an average error of 6.9% and the classifier to differentiate the types of bulbs has a 13.9% error rate. Measuring the surface area of lights is a more challenging problem since it relies on correctly identifying correspondences between the same light in successive frames, while using position information generated by clocks which are not perfectly synchronized. Surface area is estimated at approximately a factor of two, with 105% error rate across all datasets.

References

- [1] C. Jackson and K. Papamichael, May 2014. [Online]. Available: <http://www.energy.ca.gov/2014publications/CEC-500-2014-039/CEC-500-2014-039.pdf>
- [2] “Energyplus.” [Online]. Available: <https://energyplus.net/>
- [3] M. Krarti, *Energy audit of building systems: an engineering approach*. CRC press, 2016.
- [4] “Lighting audit software — why snapcount? — streamlinx.” [Online]. Available: <http://streamlinx.com/snapcount/>
- [5] “Commercial lighting energy audits.” [Online]. Available: <http://www.kmelectric.com/energy-audits/>
- [6] K. Keerthi Jain, N. Kishore Kumar, K. Senthil Kumar, P. Thangappan, K. Manikandan, P. Magesh, L. Ramesh, and K. Sujatha, *Lighting Electrical Energy Audit and Management in a Commercial Building*. Singapore: Springer Singapore, 2017, pp. 463–474. [Online]. Available: http://dx.doi.org/10.1007/978-981-10-1645-5_39
- [7] R. Zhang, S. A. Candra, K. Vetter, and A. Zakhor, “Sensor fusion for semantic segmentation of urban scenes,” in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1850–1857.
- [8] S. Mukherjee, P. Hariprasad, O. Oreifej, B. Pugh, E. Turner, and A. Zakhor, “Automatic computer detection and power estimation in indoor environments from imagery.”
- [9] N. Corso and A. Zakhor, “Indoor localization algorithms for an ambulatory human operated 3d mobile mapping system,” *Remote Sensing*, vol. 5, no. 12, p. 6611–6646, Dec 2013. [Online]. Available: <http://dx.doi.org/10.3390/rs5126611>
- [10] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, “Selective search for object recognition,” *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013. [Online]. Available: <https://ivi.fnwi.uva.nl/isis/publications/2013/UijlingsIJCV2013>

- [11] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” *CoRR*, vol. abs/1311.2524, 2013. [Online]. Available: <http://arxiv.org/abs/1311.2524>
- [12] M. Laskowski, “Detection of light sources in digital photographs,” in *In 11th Central European Seminar on Computer Graphics*, 2007.
- [13] K. Venable, K. McGee, D. McGarry, K. Wightman, C. Gutierrez, R. Weiser, B. Vanderford, J. Knight, B. Rasmussen, R. Coverick *et al.*, “Image recognition system for automated lighting retrofit assessment,” in *35th Industrial Energy Technology Conference*. Energy Systems Laboratory (<http://esl.tamu.edu>), 2013.
- [14] C. J. Bay, T. J. Terrill, and B. P. Rasmussen, “Autonomous lighting assessments in buildings: part 1—robotic navigation and mapping,” *Advances in Building Energy Research*, pp. 1–22, 2016.
- [15] T. J. Terrill, C. J. Bay, and B. P. Rasmussen, “Autonomous lighting assessments in buildings: part 2—light identification and energy analysis,” *Advances in Building Energy Research*, pp. 1–18, 2016.
- [16] “Tango.” [Online]. Available: <https://get.google.com/tango/>
- [17] “Eos 5d mark iii.” [Online]. Available: <https://www.usa.canon.com/internet/portal/us/home/support/details/cameras/dslr/eos-5d-mark-iii>
- [18] “Ef 16-35mm f/2.8l ii usm.” [Online]. Available: <https://www.usa.canon.com/internet/portal/us/home/products/details/lenses/ef/ultra-wide-zoom/ef-16-35mm-f-2-8l-ii-usm>
- [19] C. Inc, “Canon eos 5d mark iii instruction manual.”
- [20] “Spark-vis - ocean optics.” [Online]. Available: <http://oceanoptics.com/product/spark-vis/>
- [21] “Mcp2221,” 2015. [Online]. Available: <http://ww1.microchip.com/downloads/en/devicedoc/20005292b.pdf>

- [22] “Tango explorer — tango — google developers.” [Online]. Available: <https://developers.google.com/tango/tools/explorer>
- [23] O. Tange, “Gnu parallel - the command-line power tool,” *login: The USENIX Magazine*, vol. 36, no. 1, pp. 42–47, Feb 2011. [Online]. Available: <http://www.gnu.org/s/parallel>
- [24] S. Barnett, M. Tomkins, and Z. Weidelich, “Canon 5d mark iii review - performance,” Jan 2013. [Online]. Available: <http://www.imaging-resource.com/prods/canon-5d-mkiii/canon-5d-mkiii6.htm>
- [25] MathWorks, “Motion-based multiple object tracking.” [Online]. Available: <http://www.mathworks.com/help/vision/examples/motion-based-multiple-object-tracking.html>
- [26] J. Munkres, “Algorithms for the assignment and transportation problems,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957. [Online]. Available: <http://www.jstor.org/stable/2098689>
- [27] “File:high-contrast-camera-photo-2.svg.” [Online]. Available: <http://commons.wikimedia.org/wiki/File:High-contrast-camera-photo-2.svg>