Urban Street Lighting Infrastructure Monitoring Using a Mobile Sensor Platform

Sumeet Kumar, Ajay Deshpande, Stephen S. Ho, Jason S. Ku, and Sanjay E. Sarma

***Abstract*— We present a system for collecting and analyzing information on street lighting infrastructure. We develop a car-mounted sensor platform that enables collection and logging of data on street lights during night-time drive-bys. We address several signal processing problems that are key to mapping street illumination levels, identifying street lamps, estimating their heights, and geotagging them. Specifically, we highlight an image recognition algorithm to identify street lamps from the video data collected by the sensor platform and its subsequent use in estimating the heights of street lamps. We also outline a framework to improve vehicle location estimates by combining sensor observations in an extended Kalman filter framework. Our eventual goal is to develop a semi-live virtual 3-D street lighting model at urban scale that enables citizens and decision makers to assess and optimize performance of nighttime street lighting.**

***Index Terms*— Sensors, urban sensing, mobile sensing, machine vision, image recognition, machine learning, geotagging, automation.**

I. Introduction

THERE are nearly 40 million street lights in the US alone, and they consume 31 TWh of energy annually [1]. Worldwide numbers will increase with urbanization as billions of people move to urban centers. International lighting standards require a certain threshold of street lighting ranging from 1 lux of light at the surface of residential suburbs to 50 lux at road intersections [2], [3]. Oddly, the science of streetlight placement is relatively primitive today, and the means to monitor how much light reaches the street are very limited. The standards for measuring lighting are complicated, manual, and rarely implemented at a city-wide scale.

Furthermore, like much of the infrastructure in the developed word, lighting infrastructure has aged. Monitoring streetlights is a tedious manual task that relies on inspection and incident reports. Matters are not helped by the 2008 financial crisis, after which many cities have faced severe budget shortfalls, several of which are teetering on the brink of bankruptcy [4]. To offload infrastructure costs, some cities are now outsourcing lighting activity to private companies on fixed-price contracts, but have inadequate methods to measure the service level delivered to citizens. The contractors themselves are considering replacing old lights with LED units, but are struggling to figure out what the inventory of lighting is, how many lux units they deliver on the street level, and how to monitor streetlight condition over time. LED lights are dimmable, which opens up another avenue: modulating lights to ensure that energy is minimized without compromising safety or security. This too requires better assessment and measurement tools.

A common practice to monitor street lighting is to have a city official drive scheduled routes and observe the status of street lamps. This approach lacks scalability and scientific reliability. Another approach involves local government blocking of a specific road and measuring light levels using a lux meter. Though this approach provides accurate and reliable data, it is labor intensive, not scalable and may cause significant inconvenience to the citizens. Researchers have also explored deployment of static light sensor nodes on streetlight poles to monitor and control lamps remotely [5], [6]. Commercial systems such as GE LightGrid system and Philips CityTouch system have found some recent success. Such systems are expensive, for example, the GE LightGrid for connecting $26 - 50$ street lamps costs \$10 k [7] and may not be viable where an upgrade of the existing lighting infrastructure is difficult. Also, the static sensors measure the lighting levels accurately close to the lamps, while the decision makers are more interested in measuring street level illumination. Furthermore, such static deployment suffers from drawbacks, such as, cost, maintenance needs and node failures [8], [9].

There is a pressing need for a scalable way to measure and map street lighting. The information needs to be updated regularly enough so that timely operational and maintenance decisions are enabled, though the information update may not be real time. In short, imagine a form of "Google Street View" with a semi-live, updated view of lighting along city streets and its utility to both citizens and decision makers.

In this paper, we introduce a car-top sensor system designed to monitor urban street lighting. The inherent mobility of the system allows scalability of data collection. We discuss challenges and solutions on sensor integration, data management, algorithm development and data analysis. While we note that there are strict norms around how illuminance measurements need to be conducted from the regulations point of view [2], [3], and our system does not claim any of these, our

Manuscript received February 1, 2016; revised March 22, 2016; accepted March 29, 2016. Date of publication April 8, 2016; date of current version May 17, 2016. This work was supported by Ferrovial Servicios. The associate editor coordinating the review of this paper and approving it for publication was Dr. Wan-Young Chung.

S. Kumar, S. S. Ho, J. S. Ku, and S. E. Sarma are with the Mechanical Engineering Department, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: sumeetkr@mit.edu; ssh1@mit.edu; jasonku@mit.edu; sesarma@mit.edu).

A. Deshpande is with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (e-mail: ajayd@us.ibm.com).

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the authors. The Supplementary Material contains example data collection file generated by the sensors (except camera) on the car-mounted sensor platform. This material is 1 MB in size. The Supplementary Material also contains example video data collected by one of the cameras on the car-mounted sensor platform. This material is 234 MB in size.

Digital Object Identifier 10.1109/JSEN.2016.2552249

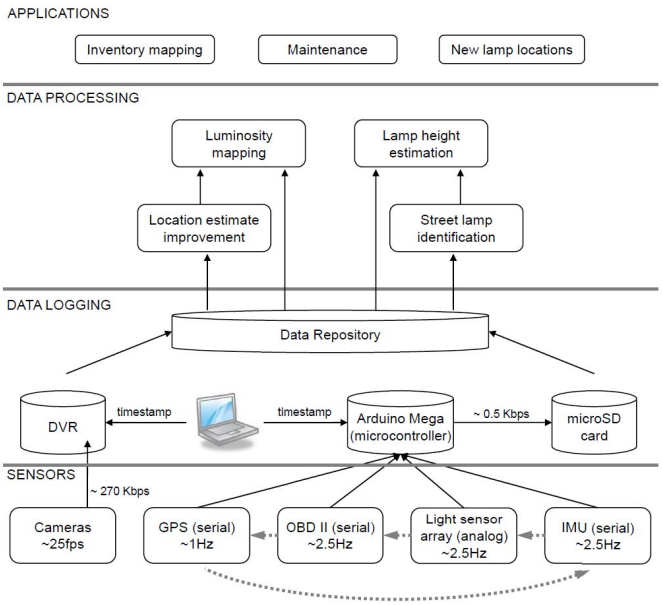


Fig. 1. Overall system architecture.

work demonstrates the feasibility of a scalable and reliable approach to monitoring, mapping and identifying failures.

In Section II we present a high-level architecture of our system, lay down our application goals and establish context for the various modules to be presented in the rest of the paper.

II. System Architecture

Figure 1 presents the overall system architecture organized in four layers. The bottom most layer consists of sensors collecting data of different modalities. All the sensors are mounted on a car-top sensor platform. The next layer consists of a data logging system which gathers and logs data from the sensors. The data processing layer addresses the processing of raw data to extract useful features and learn different models that serve as the building blocks for the potential applications in the top-most layer. The location estimate improvement model fuses data from GPS, IMU and OBD-II sensors to improve vehicle location estimates. The luminosity mapping model leverages improved locations and data from light sensors to create spatio-temporal heat maps of street illumination. The street lamp identification model leverages camera images to classify street lamps from other bright objects. The lamp height estimation model uses the classification results and data from OBD-II sensors to estimate the heights of street lamps.

A. Paper Organization

The rest of the paper is organized around the detailed description of the bottom three layers. In Section III we describe the bottom two layers including the sensors and the components of the data logging system. In the remaining sections we address the modules in the data processing layer. In Sections IV and V, we address the lamp identification model and the lamp height estimation model respectively. In VI, we briefly address the location estimate improvement model and luminosity mapping. In Section VII we discuss the related work that pertain to the different technical challenges addressed in the paper and in Section VIII we conclude.

TABLE I
Sensor Make and Description

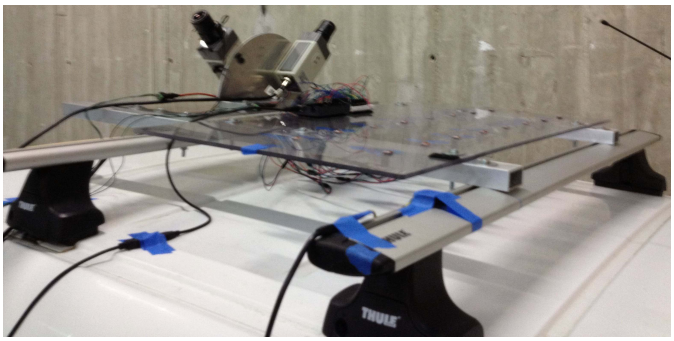
Sensor	Additional description	Sampling rate
TEMT6000 light sensor	Resolution: 0.25 lux, Range: [0,250] lux	–
BC-EL630 security camera with CCD	Field of view: $\sim 95.6^{\circ}$, 12 V DC powered through car charger	25 fps
uBlox LEA6 GPS from 3D Robotics	Accuracy: 2.5 m circular, UART capability for logging	5 Hz
UM6-LT Orientation Sensor from CH Robotics	Accelerometer and gyroscope measurements, broadcasts measurements at 20-300 Hz	1000 Hz
Sparkfun OBD-II UART	Car speed measurements, UART capability for logging	–

III. Hardware and System Integration

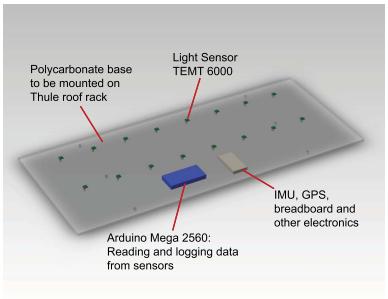
We develop a car-top platform mounted with sensors to gather information on street illumination levels and conditions of street lamps. We measure incident illumination using light/lux sensors. Light sensors are incapable of providing differentiating information about various sources of light. Since we aim to understand the condition of street lamps, we employ a system of video cameras that provides information on the presence or absence of street lamps in the environment. Furthermore, location information is crucial for any mobile sensing approach to perform geo-spatial analytics. We add a GPS to our sensor platform. Since GPS suffers from low reliability, especially in urban driving conditions we augment our sensor platform with an IMU that provides accelerometer, gyroscope and magnetometer (heading) data and an OBD-II reader which reads car speed data.

A. Sensors

In this section we describe the sensors, their description and make, and the manufacturer specified sampling rates for digital sensors. Table I above provides a consolidated summary. We used the TEMT6000 light sensor to measure light intensity. For collecting video data, we used BC-EL630, which is a security camera with CCD sensors. This particular camera was selected as it can operate effectively at low lighting conditions without the use of infrared and has a wide dynamic range that allows it to be used under varied lighting conditions. The system also had a power box that supplied 12 V DC to the cameras. During deployment, the power box was connected to a cigarette charger in the car. The cameras came along with a DVR, which was used to record videos during field experiments. We used the 3D Robotics uBlox LEA6 GPS module which allowed easy integration with a microcontroller through a serial port for data logging. For an IMU, we used the UM6-LT Orientation Sensor from CH Robotics. It provided accelerometer and gyroscope measurements. We used an OBD-II UART from Sparkfun to interface with a car's OBD-II bus providing car speed data. This sensor supports all major OBD-II standards, such as CAN and JBUS, and can be easily integrated with a microcontroller through its standard Rx/Tx ports. As a microcontroller, we chose Arduino Mega 2560



(a)



(b)

Fig. 2. (a) Assembled car-top sensor platform. (b) Light sensor array.

because of its speed, I/O capabilities, extensive open source documentation and code base and ease of use. We also used a standard 9 V battery pack for powering Arduino.

B. Car-Top Platform

Fig. 2a shows the assembly of the sensor platform on top of a car. We fabricated aluminum supports to mount the cameras. In order to ensure coverage of the half-plane above the car roof, we mounted the cameras at angles $+30°$ and $-30°$ with respect to the vertical direction. We created an array of light sensors to measure spatial variability of vertical illumination on a plane or the isolux contours. The isolux contours when combined with the lamp location information can help us to model the three-dimensional light field and estimate street level illumination. We designed a plexiglass board on which the light sensors were assembled in an 8×2 array as shown in Fig. 2b. The GPS, the IMU, the OBD-II sensor and the microcontroller were assembled on the plexiglass. Both the camera rig and the plexiglass were then attached to a roof rack which was then placed on top of a car. The DVR was placed inside the car and the BNC cables were passed through the window opening. Similarly, wires from the OBD-II reader were passed through the window opening to be connected to the car's OBD port.

C. Data Logging and Management

Figure 1 also depicts the architecture of the data logging system. We used the analog pins on Arduino Mega to read data from the light sensors and the Rx-Tx pins to read the serial data coming from the IMU, OBD-II reader and GPS (digital). For the analog data, Arduino uses a 10-bit analog-to-digital converter to convert the voltage reading in the range $0-5$ V to a binary number. The other three sensors use an asynchronous serial protocol to communicate with the Arduino over the Rx-Tx ports. We used the documentation available for each of the three sensors to develop a computer program to parse the serial data coming from the sensors and extract the relevant sensor measurements. We used a microSD shield, an adapter that enabled connecting a microSD card to the microcontroller, for direct logging of the sensor data as .txt files on the mounted microSD card. As seen in Figure 1 the video data was recorded on the DVR.

We had data coming from different sensors and it was important to have a common frame of reference for their timestamps to enable multi-sensor data analytics. The digital sensors, microcontroller and DVR have internal clocks which may not be synchronized. We decided to use a laptop's clock to provide a common time reference to all the sensors as shown in the bottom layer of Figure 1. At initialization we used a laptop to provide a reference time stamp to the DVR. Similarly, when the microcontroller was initialized, we sent the current time of the laptop to the Arduino via USB through a computer program that we developed. Note that the two time stamps may have different values but are referenced with respect to the laptop's clock. We then relied on the internal DVR and microcontroller clocks to advance time stamps from the initial reference times. We used the Arduino's time stamp when logging different sensors' data. All the sensor data on the Arduino were hence timestamped with respect to the laptop time just as the video on the DVR. This solution however required a user to manually initialize both the Arduino and the DVR and then the data was collected autonomously during deployment. The Arduino polled data from the different sensors sequentially and repeated the data collection loop throughout the field deployment as shown in Figure 1. The data collection scheme was a polling system where the microcontroller sequentially looked for a new data packet from each sensor and if the data packet was available, the microcontroller read and transferred the data packet to the microSD card. During testing, we identified that if the data packet was not available then moving to the next sensor reduced the idle time of the microcontroller and maximized the overall sampling rate of the sensor system.

D. Field Experiments

We tested our final prototype in four cities in three countries: Cambridge MA (USA), Malaga (Spain), Santander (Spain) and Birmingham (UK). The prototype worked robustly and reliably. In our experiments we found that the GPS was logged at a sampling rate of $\sim$1 Hz and the sampling rate had a standard deviation of $\sim$25%. The other three sensors were logged at $\sim$2.5 Hz and the rate had a standard deviation of $<$20%. The total data rate for the Arduino microcontroller was $\sim$1.71 MB/hr. The videos were recorded at 25 fps and the data rate was $\sim$0.96 GB/hr.

We found the driving speeds of 30 mph and less to have a minimum impact on the measurement process. As the minimum separation between lamp poles in our field experiments was greater than 25 m, at 30 mph our system collected at

(a) (b)

Fig. 3. Example images collected by the mobile sensor platform.

least 5 light sensor, OBD-II and IMU measurements, 2 GPS observations and 50 images per street lamp.

IV. An Algorithm for Street Lamp Identification

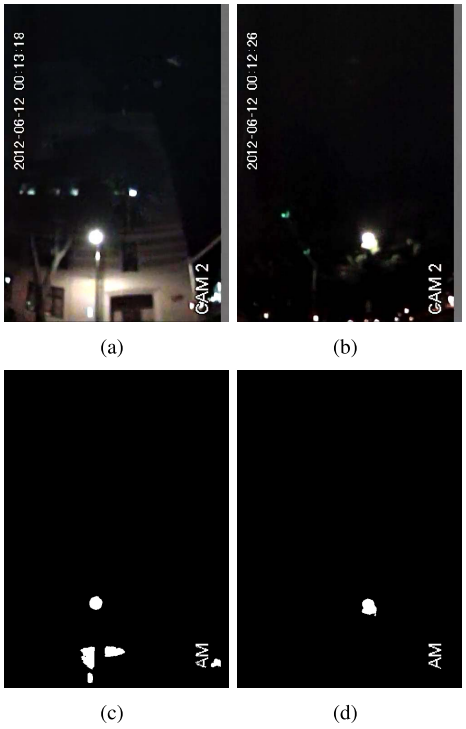
Our eventual goal is to develop a "Google Street View" of urban street lighting infrastructure that has a semi-live updated information on the location and performance of lamps and street illumination. For that, several data processing challenges have to be addressed as shown in Fig. 1. Identifying street lamps from the night time video data collected is the first step. In this section we present a street lamp identification algorithm that employs state of the art feature engineering and machine learning techniques. In the subsequent section we extend the results on lamp classification to estimate the heights of street lamps. Later, we discuss the importance of improving vehicle location and trajectory estimates in the context of luminosity mapping and present an extended Kalman filter framework to combine car speed measured by OBD-II reader with GPS data and inertial measurements.

A. Street Lamp Identification From Night Time Video Data

Fig. 3 shows example images collected by our mobile light scanning platform. As one expects, night time images comprise either of dark pixels or saturated pixels. Furthermore, the images suffer from bloom or glow which produces fringes of light extending from bright objects into the dark regions of the image and reduces the contrast and sharpness (*i.e.*, presence of edges). The aforementioned reasons make it hard to implement object identification algorithms.

As seen in Fig. 3, the bright, saturated regions in the images may also correspond to non-street lamps such as lit windows and doors and we need to differentiate them from the street lamps. We minimize imaging headlights of other cars by ensuring that the field of view of the camera system is above the car roof (at intersections cars on the roads perpendicular to the driving direction may come into view). Essentially, our goal is to develop an object identification technique which can identify street lamps from other bright objects in the background.

We now present an overview of our approach to developing the street lamp classifier. The details of the implementation can be found in the author's PhD thesis [10]. As mentioned before, the pixels corresponding to street lamps are saturated. A direct first step towards identifying the regions of an image that may correspond to a street lamp is to apply a thresholding-based image segmentation.



(a) (b) (c) (d)

Fig. 4. For the original images in the top row the bottom row contains the corresponding binary image.

B. Preprocessing

We first converted every color image $I_{N\times M\times 3}$ to a grayscale image $G_{N\times M}$ where every pixel had a value $G_{i,j} \in [0, 1]$. The grayscale image was then converted to a black and white image using thresholding segmentation where every pixel with $G_{i,j} \leq t$ was mapped to zero and every pixel with $G_{i,j} > t$ was mapped to 1. We chose the threshold to be $t = 0.975$ and denoted the binary image as $B_{N\times M}$. We further removed from the binary image all the bright objects that had less than 200 pixels as we experimentally observed that street lamps were typically larger than 200 pixels. Fig 4 shows examples of the original image (first row) and the converted binary image (second row).

A bounding box was drawn around each bright region and the corresponding region in the original image ($I_{N\times M\times 3}$) was cropped and saved as a possible street lamp candidate. We used the cropped image set to develop the street lamp identification algorithm. Fig. 5 shows example lamp images and Fig. 6 shows examples of non-lamp bright objects in the scene that we aim to correctly differentiate from the lamps. The cropped images were manually labeled as lamps and non-lamps to enable the subsequent supervised learning study.

We observe that there is a significant variation in shape, intensity distribution and background noise in the set of lamp images. When comparing Fig. 5 with Fig. 6, we observe the lamps are primarily oval in shape while the non-lamp objects

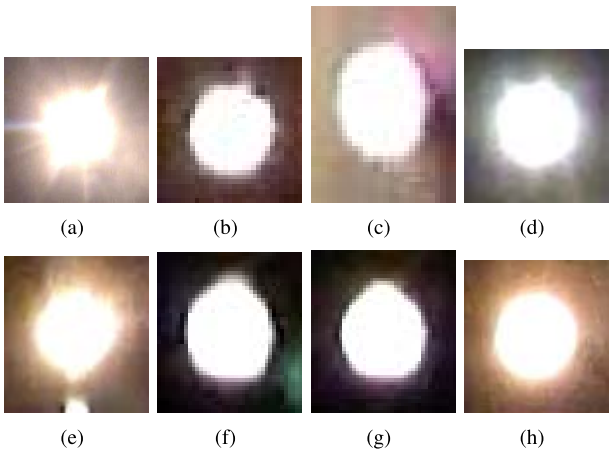


Fig. 5. Example images of lamps.

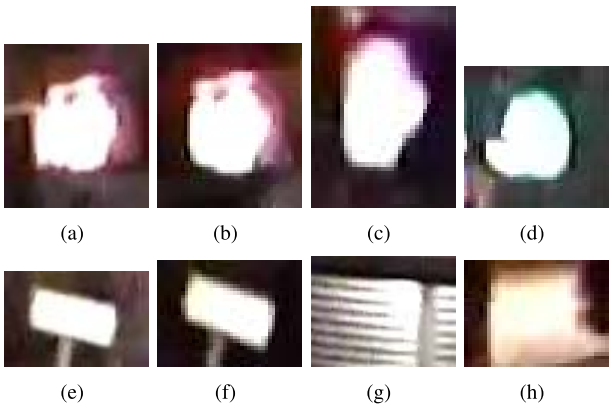


Fig. 6. Example images of other bright objects in the scene.

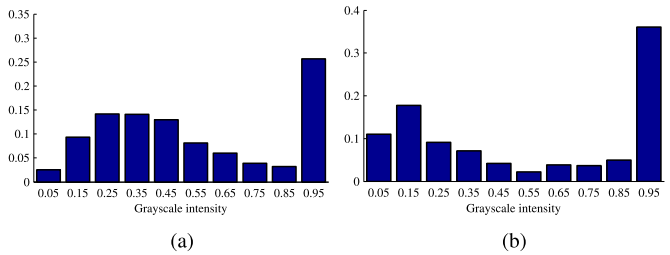


Fig. 7. Normalized histogram of grayscale intensity. (a) A lamp image. (b) A non-lamp image.

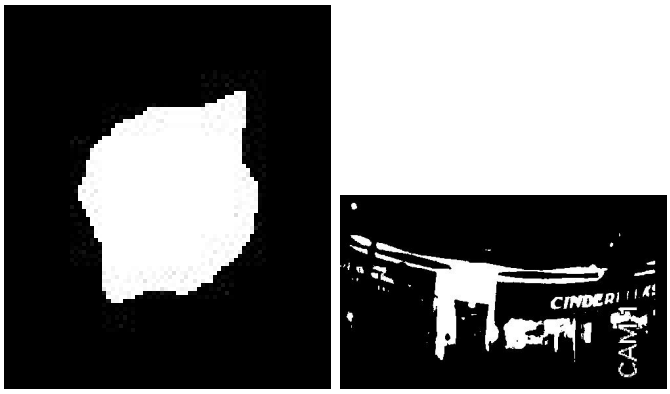


Fig. 8. Black and white binary image. (a) A lamp object. (b) A non-lamp object.

are primarily rectangular in shape. Furthermore, there is a difference in the nature of the spatial distribution of intensity gradients in the regions surrounding the saturated pixels between lamp and non-lamp objects. Note that identifying an appropriate edge or corner set representation of the lamp objects is hard due to both the variation in shape and the bloom effect.

C. Feature Representation

Our approach is to construct different categories of feature sets and then concatenate them to form a large feature vector. Thereafter, we use dimension reduction techniques to find a good subset of features from the large feature vector. The large feature vector is composed of the following categories:

- **Normalized Histogram of Grayscale Intensity**: We converted every cropped image to grayscale ($\in [0, 1]$) and calculated the histogram of the grayscale intensity with the following bin centers: $[0.05 : 0.1 : 0.95]^T$ and the bin size of 0.1 giving a 10 dimensional feature vector. Fig. 7 shows an example of the different histograms computed for a lamp and a non-lamp image.
- **Size and Total Intensity**: Two additional feature vectors which are the number of pixels in the cropped images and the sum of the grayscale intensity.
- **Shape Parameters**: We converted each cropped image to a black and white image by using thresholding segmentation with a threshold value of 0.975. Matlab allowed direct computation of several shape parameters using the function "regionprops" and we used the following:
  1) Number of Edges in the Convex Hull of the region: We computed the Convex Hull of the black and white image which is a $p \times 2$ matrix that specifies the smallest convex polygon that can contain the region. We took p as a feature vector which corresponded to the number of edges in the enclosing convex polygon of the region.
  2) Eccentricity: Eccentricity (ϵ) of an ellipse which has the same second-moment as the region.
  3) Euler Number: A scalar which specifies the difference between the number of objects in the image and the number of holes in the image. The rationale behind using the Euler number is that we expect lamp images to have no holes (Fig. 8a). On the other hand non-lamp objects may have holes within the bright regions (Fig. 8b).
  4) Extent: The ratio of the total number of pixels in the region to the total number of pixels in the smallest rectangle containing the bright/white pixels.
- **Histogram of Oriented Gradients (HOG)**: They are feature descriptors that count occurrences of gradient orientation in localized regions of an image. These descriptors have found good success in object

recognition [11]–[15]. The premise for the HOG descriptors is that local object appearance and shape can be described by the distribution of intensity gradients or edge directions. To compute the HOG descriptors we converted every cropped image to grayscale and then rescaled them to a standard template size of 50×50 using a bilinear transformation [16]. We then divided the image into 3×3 overlapping cells of pixel dimensions 24×24. We computed the x and the y derivative of the image intensity (I_x and I_y) using the centered derivative mask $[1, 0, -1]$ and $[1, 0, -1]^T$ respectively. The gradient orientation was then calculated by $\theta = tan^{-1}\left(\frac{I_y}{I_x}\right)$. The angular range of $(\pi, \pi]$ was divided into 18 discrete bins each of size 20°. The histogram for every cell was calculated by identifying all the pixels in the cell whose θ belong to a certain 20° bin interval and assigning the magnitude of the gradient, *i.e.*, $\sqrt{I_x^2 + I_y^2}$ as the contribution of the pixel in that bin. Hence, for every cell, an 18 dimensional histogram vector ($\mathbf{h}$) was created which was then contrast normalized by dividing it by $\|\mathbf{h}\| + 0.01$, where $\|.\|$ is the ℓ_2 norm of the vector. Combining the histograms from all the cells resulted in a 162 dimensional feature vector that represented the HOG.

- **Pixel Grayscale Intensity Values**: We converted every cropped image into a standard 50×50 template using a bilinear transformation, converted them to grayscale and concatenated the grayscale pixel intensity values to form a 2500 dimensional feature vector.

After concatenating all the feature vectors obtained above, we formed a 2678 dimensional feature vector, *i.e.*, $\mathbf{f}_i \in \mathcal{R}^{2678}$. Feature selection (aka variable reduction) is an important step towards developing a robust classifier. Our data set comprised of 13483 cropped images out of which 1689 were images of lamps. The preceding ratio of the number of lamps to non-lamps was the natural distribution obtained when the original video frames were cropped. Out of the 11794 non-lamp images 3288 were images of the texts added by the camera (see bottom right of Fig. 8b).

D. Feature Reduction and Lamp Classification

In this paper, we employed filtering-based feature selection techniques to reduce the dimension of the feature space because of their speed and computational efficiency [17]–[22]. Many filter based feature selection techniques are supervised and use the class labels to weigh the importance of individual variables. We used Fisher score and Relief score as supervised filters and principal component analysis (PCA) as an unsupervised filter.

- We computed the Fisher score [17] of a feature for binary classification using:

$$FS(f_i) = \frac{n_1(\mu_1^i - \mu^i)^2 + n_2(\mu_2^i - \mu^i)^2}{n_1\left(\sigma_1^i\right)^2 + n_2\left(\sigma_2^i\right)^2} = \frac{1}{n}\frac{(\mu_1^i - \mu_2^i)^2}{\frac{\left(\sigma_1^i\right)^2}{n_1} + \frac{\left(\sigma_2^i\right)^2}{n_2}}, \quad (1)$$

where n_j is the number of samples belonging to class j, $n = n_1 + n_2$, μ^i is the mean of the feature f^i, μ_j^i and σ_j^i are the mean and the standard deviation of f_i in class j. The larger the score, the higher the discriminating power of the variable is. We calculated the Fisher score for every variable and ranked them in a decreasing order and then selected the subset whose weights were in the top $p\%$ as the reduced feature set.

- We calculated the Relief score [18] of a feature by first randomly sampling m instances from the data and using:

$$RS(f_i) = \frac{1}{2}\sum_{k=1}^{m} d\left(f_k^i - f_{NM(\mathbf{x}_k)}^i\right) - d\left(f_k^i - f_{NH(\mathbf{x}_k)}^i\right), \quad (2)$$

where f_k^i denotes the value of the feature f_i on the sample $\mathbf{x}_k$, $f_{NH(\mathbf{x}_k)}^i$ and $f_{NM(\mathbf{x}_k)}^i$ denote the values of the nearest points to $\mathbf{x}_k$ on the feature f_i with the same and different class label respectively, and $d(.)$ is a distance measure which we chose to be the ℓ_2 norm. The larger the score, the higher the discriminating power of the variable is. Again, we calculated the Relief score for every variable and ranked them in a decreasing order. We then selected the subset whose weights were in the top $q\%$ as the reduced feature set.

- Given the observation matrix X, PCA computes a score matrix S which represents the mean subtracted observation matrix in the latent space L through:

$$S_{N\times M} = (X - X_{avg})_{N\times M} L_{M\times M}. \quad (3)$$

If we chose a k-dimensional subspace to represent our observations, using the first k columns of L (L_k) allows the optimal representation of the observation in the k-dimensional subspace as:

$$\hat{S}_k = (X - X_{avg})L_k. \quad (4)$$

Increasing k increases the percentage variance explained by the reduced subspace (var_k). We selected k according to the criterion $\frac{\text{var}_k}{\text{var}_M} \leq r$, where $r \in [0, 1]$.

The subsets selected by the three methods described above depend on the parameters p, q and r respectively. We used three different types of classifiers: Naive Bayes classifier [23]–[25], discriminant analysis [26], [27] and support vector machines [28]–[30].

We divided the data set into a training set and a test set where the test set had around 10% of the samples from each of the three classes. We denoted the set of observations in the training set as $\mathbf{f}_T$ and the set of observations in the test set as $\mathbf{f}_t$.

On the basis of the observation that the camera text was consistent, we decided to develop a three-class classification technique. In our proposed method we first aim to classify the cropped image as a camera text or the "others" (one-vs-all strategy for multi-class classification). Furthermore, text recognition has been studied extensively [31]–[33] and it would serve as a good control test for the proposed algorithm. If the image is classified as the "others" we then employ a two-class classification technique to identify if it is a lamp or

TABLE II

NUMBER OF VARIABLES $|S|$ IN THE SELECTED VARIABLE SUBSET

Fisher Score		Relief Score		PCA	
p	$\|S\|$	q	$\|S\|$	r	$\|S\|$
0.85	286	0.85	239	0.15	0
0.75	31	0.75	34	0.25	0
0.50	13	0.50	7	0.50	2
0.25	6	0.25	1	0.75	20
0.10	2	0.10	1	0.90	100

TABLE III

TEN-FOLD MISCLASSIFICATION RATE IN %

Fisher Score based feature selection, $p = 0.75$					
Naive Bayes Classifier					
Gaussian distribution		Kernel smoothing density estimate			
		Gaussian	Box	Triangle	Epanechnikov
2.70		1.46	2.17	1.83	1.90
Discriminant Analysis					
Linear	Quadratic	Diagonal Linear	Diagonal Quadratic	Pseudo Linear	Pseudo Quadratic
1.45	2.75	1.14	2.70	1.45	2.75
Support Vector Machines					
Linear		Quadratic	Polynomial (order 3)		Radial Basis Function
1.02		0.94	1.21		0.82

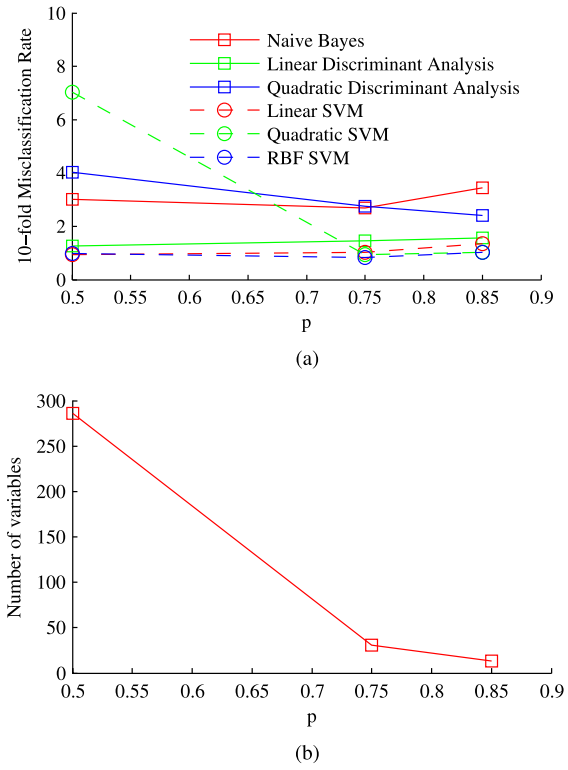


Fig. 9. (a) Variation of tenfold misclassification rate with p. (b) Variation of size of the variable set selected by the Fisher score ranking with p.

a non-lamp. The following details the steps involved in the development of the sequential binary classifiers:

- **Identifying Camera Text**: Given the training data set $\mathbf{f}_T$, we first constructed class labels, assigning 1 to examples that were camera text and 0 to the ones that belonged to the "others" category. As discussed in Section IV-D, choosing different values of p, q and r leads to different sizes of the selected variable sets ($|S|$) as enumerated in Table II.
  Matlab has various implementations of each of the three aforementioned classifiers and we studied their performance with different implementations. We employed 10-fold cross-validation to find the best implementation and the optimal parameters of a classifier. For the Fisher score based variable selection method, with $p = 0.75$, Table III enumerates the 10-fold misclassification rate for the different classifiers. For the SVMs, the optimal choice of C and γ (in the case of rbf kernel) in Table III was obtained by doing a grid search on the discrete set $C = [0.005, 0.01, 0.1, 1, 10, 50]^T$ and $\gamma = [0.005, 0.01, 0.1, 1, 10, 50]^T$ and choosing the parameter that gave the minimum 10-fold cross-validated misclassification rate. We observed that the SVMs outperformed the Naive Bayes and the discriminant analysis classifiers. From the grid search we found that the best values of C were 0.1 for the linear kernel, 0.005 for the quadratic and the polynomial kernel and 50 for the radial basis function kernel with the best $\gamma = 10$. Fig. 9 plots the variation of the 10-fold misclassification rate and the size of the selected variable set with p. Typically, a "U-shaped" curve is expected for the misclassification rate with the size of the feature set, where a high number of features lead to overfitting and higher cross-validation error while very low number of features miss relevant features that have good discriminative power. Our simulations show a similar trend and comparing with the size of the selected variable set (Fig. 9b), we chose $p = 0.75$ as a good choice of the parameter for the Fisher score based feature selection.
  We then tested the SVM classifiers on the test set ($\mathbf{f}_t$) and found that both the linear and the radial basis function kernel achieved an accuracy of 99% in correctly identifying the camera texts. Owing to its simplicity, we chose the linear kernel SVM over the radial basis function kernel SVM with $C = 0.1$.
  We followed a similar analysis for the Relief score based feature selection and found that $q = 0.75$ was a good choice for sub-selecting features. The parameters for SVMs were again identified by a grid search. We observed that the SVMs outperformed the other classifiers and when they were compared on the test set, the linear kernel SVM achieved an accuracy of 99% in correctly identifying the camera texts. Similarly, for PCA based feature selection we found $r = 0.75$ to be a good choice for feature reduction. On the test set we again found that the linear kernel SVM gave an accuracy of 99%.
  From the above discussion, the best linear SVM selected by either of the three feature selection methods achieved an accuracy of 99% in correctly classifying camera

texts on the test set. We selected all three of them as the final set of classifiers for identifying camera texts annotated on the images. The final prediction was made on the basis of a simple majority vote across the three classifiers. Note that in all the three cases the best value of the box margin parameter, C, was found to be 0.1.

- **Identifying Street Lamps**: In this section we discuss the development of the two-class classifier to identify lamps and non-lamps from the set "others". We followed a process similar to the one discussed above. Firstly, we identified the parameters (p, q and r) for the different feature selection methods by comparing the 10-fold misclassification rate and then we selected the classifier that gave the minimum classification error on the test set as the final classifier. Though, in this case, we observed that the PCA based feature selection did not work as the first principal component explained around 99.6% of the total variance and when we used this component to train the classifiers, we observed high misclassification rates ($\sim$15%).
  For the Fisher score based feature selection, we found $p = 0.75$ to be a good choice which led to a variable set of size 27. When the classifiers were used on the test set, the minimum classification error (false negative here) of 2.80% was achieved by the quadratic SVM while the minimum false positive error of 1.22% was achieved by the rbf SVM. Note that the classification error with rbf SVM was 8.34% while the quadratic SVM had a false positive error of 5.23%. We selected the quadratic SVM as the final choice of the classifier with $C = 0.01$.
  We found $q = 0.85$ with $|S| = 15$ to be a good choice for the Relief score based feature selection. On the test set, the minimum classification error of 3.35% was achieved by the polynomial SVM while the minimum false positive error of 1.45% was achieved by the rbf SVM. Also, the classification error with rbf SVM was 5.03% while the polynomial SVM had a false positive rate of 5.34%. We selected the polynomial SVM as the final choice of the classifier with $C = 0.01$.
  We included the quadratic SVM with the Fisher score based feature selection and the polynomial SVM with the Relief score based feature selection as the final set of classifiers. We then adopted a hybrid classification scheme where an image was classified as a lamp if both the classifiers predicted it to be a lamp otherwise it was classified as a non-lamp. This approach was successful in reducing the false positives and we obtained $\sim$5% classification error and $\sim$3.5% false positive error with the hybrid approach. Fig. 10 summarizes the steps involved in creating the final hybrid classifier.

As digital cameras are becoming inexpensive and pervasive, using images for infrastructure management has attracted significant attention. Here we have described an approach that employs systematic feature construction, reduction and subsequent classifier optimization to develop a lamp classification algorithm that addresses our goal of street lamp identification from the video data collected by the car-top sensor platform. As a future work, we would like to explore more sophisticated machine learning techniques, for example, deep learning that has the potential to automatically learn high level features from the image set and perform improved classification [34]. We would also like to make the data publicly available so that other researchers can explore novel image classification techniques for street lamp identification.

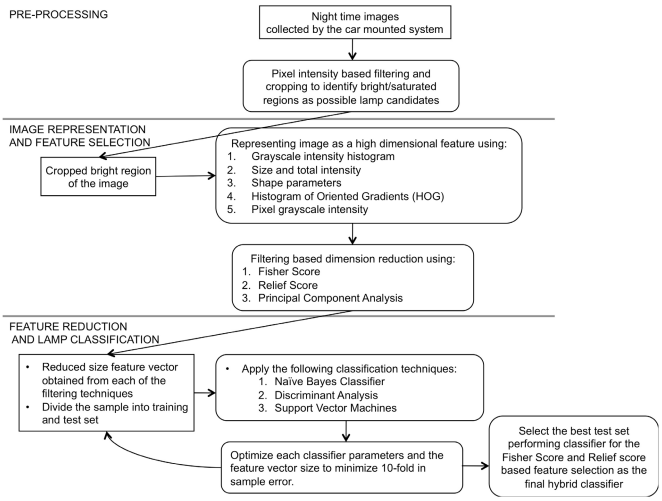


Fig. 10. Schematic of the steps involved in creating the hybrid lamp classifier.

V. A Height Estimation Algorithm Using Lamp Tracking and Perspective Geometry

Some cities have accurate data on streetlight inventory including their heights but many don't. Lamp height information is useful for modeling illumination of streets and walkways. Furthermore, this information is of interest to those responsible for the maintenance of city lighting assets. For example, when sending maintenance crews, differences in lamp height can factor into what equipment and/or personnel are required on site. In this section we propose a method to estimate the height of the street lamps from the collected video data.

Fig. 11a is a visual representation of the imaging performed by the car-mounted camera system. Here we make an assumption that the car travels parallel to the edge of the road, *i.e.*, along the x axis with a speed U. In this section, we present a lamp height estimation algorithm that employs perspective projection and optical flow on the two-dimensional images.

A. Perspective Projection

Perspective projection describes the mapping of a three-dimensional point in space to a two-dimensional coordinate on the image plane [16], [35]. As shown in Fig. 11b, $X-Y-Z$ is the frame of reference attached to the ground with the $Z-$axis pointing vertically up. The frame $X' - Y' - Z'$ is attached to the camera with the Y'axis aligned with the optical axis and at an angle $\theta = 60^\circ$ with respect to Y axis. The point P (in our case a street lamp) is represented by the coordinated (X', Y', Z') and its image p by (x, f, z). The axis X and X' are coincident.

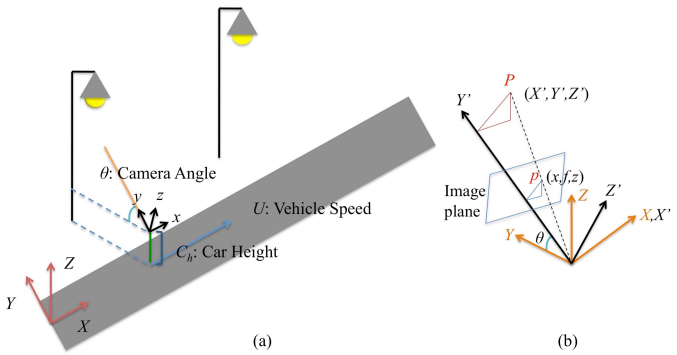


Fig. 11. Geometry of the car-mounted imaging system.

Fig. 12. (a) Example grayscale image frame. (b) Magnitude of x–direction intensity gradient. (c) Magnitude of y–direction intensity gradient.

Assuming an ideal pin hole camera model and using equations of coordinate transformation, for the point P we have

$$Z = \left(\sin(\theta) + \frac{z}{f}\cos(\theta)\right)\frac{X'f}{x}. \tag{5}$$

We have further assumed that the ground reference frame is attached to the roof of the car. We can now add the height of the car (C_h) to Z to obtain the height of the lamp post as

$$\begin{aligned} h_{\text{lamp}} &= C_h + \left(\sin(\theta) + \frac{z}{f}\cos(\theta)\right)\frac{X'f}{x} \\ &= C_h + \left(\sin(\theta) + \frac{z}{f}\cos(\theta)\right)Y'. \end{aligned} \tag{6}$$

B. Optical Flow

Optical flow is defined as the apparent motion of the brightness pattern on an image as the camera moves with respect to an object or vice versa [16]. Referring to our imaging geometry of Fig. 11a, let $I(x,z,t)$ be the irradiance at a time t at the image point (x,z). If $u(x,z)$ and $w(x,z)$ are the x and the z components of the optical flow vectors at that image point and if ∂t is a small time interval, we make the following assumption:

$$I(x,z,t) = I(x + u\partial t, z + w\partial t, t + \partial t). \tag{7}$$

Using the Taylor series expansion of (7) and ignoring higher order terms, we obtain the following optical flow equation at every pixel (i,j) of a digital image:

$$I_x u_{i,j} + I_z w_{i,j} + I_t = 0, \tag{8}$$

where we use a finite difference approximation to compute $I_x = \left(\frac{\partial I}{\partial x}\right)_{i,j}$, $I_z = \left(\frac{\partial I}{\partial z}\right)_{i,j}$ and $I_t = \left(\frac{\partial I}{\partial t}\right)_{i,j}$.

Lucas-Kanade is a popular method for estimating optical flow that divides the original image into smaller sections and assumes a constant velocity in each section [36]. It then solves a weighted least squares problem in each section by minimizing the following objective:

$$\sum_{(i,j)\in\Omega}\sum W^2\left(I_x u + I_z w + I_t\right)^2, \tag{9}$$

where W is a window function that emphasizes the constraint at the center of each section, Ω is the extent of the section and (u,w) are the constant optical flow velocities obtained by minimizing (9). Fig. 12 shows an example image and the x and y intensity gradient of the image. We observe that the magnitude of the intensity gradient is primarily zero throughout the image and the pixels with higher magnitude are confined in a small region surrounding the lamps and other bright regions. This observation prompts a modification of the Lucas-Kanade method where we first identify the region of the image where a street lamp is located using our lamp classification technique and then use that region to estimate the image velocity of the street lamp.

C. Algorithm for Height Estimation

The height of the lamp post as given by (6) is in terms of the unobserved coordinate X'. Using the pin hole camera model we have

$$X' = \frac{xY'}{f}. \tag{10}$$

We assume that as the car moves along X' axis, Y' or the distance of the lamp from the camera remains constant. Differentiating the left and the right side of (10) with respect to time gives

$$\begin{aligned} U &= \frac{u(x,z)Y'}{f} \\ \implies Y' &= \frac{Uf}{u(x,z)}, \end{aligned} \tag{11}$$

where U is the speed of the car along the X axis. Substituting (11) in (6) we get

$$h_{\text{lamp}} = C_h + \left(\sin(\theta) + \frac{z}{f}\cos(\theta)\right)\frac{Uf}{u(x,z)}, \tag{12}$$

where z is the image pixel corresponding to the street lamp and $u(x,z)$ is the x component of the optical flow corresponding to the lamp pixel.

As discussed in Section V-B, we find the optical flow velocity for lamp pixels by first using our lamp classification algorithm to identify the street lamp and then use the Lucas-Kanade formulation by assuming constant (u,w) over the region corresponding to the street lamp. Assuming $W = 1$

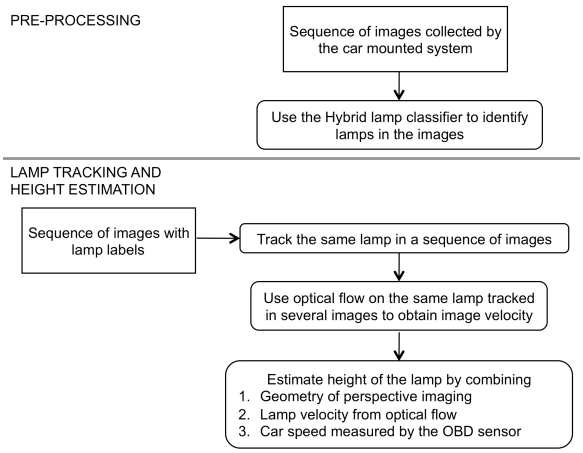


Fig. 13. Schematic of the steps involved in creating the lamp height estimation algorithm.

in (9), we obtain the following estimate of u:

$$u = -\frac{\left(\sum\sum I_y^2\right)\left(\sum\sum I_x I_t\right) - \left(\sum\sum I_x I_y\right)\left(\sum\sum I_y I_t\right)}{\left(\sum\sum I_x^2\right)\left(\sum\sum I_y^2\right) - \left(\sum\sum I_x I_y\right)^2}, \tag{13}$$

where the summations are taken over all the pixels $(i, j) \in \Omega_l$, which is the region of the image that is identified as a street lamp. Our lamp identification algorithm also provides the coordinates of the centroid (x_l, z_l) of the bounding box corresponding to a lamp object. The $z-$coordinate of the centroid, z_l, was used in height estimate. The car speed, U, was measured by the OBD-II reader. We then have the following estimate of the height of the street lamp:

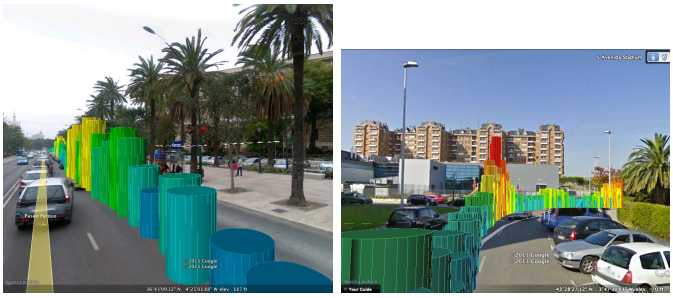
$$h_{\text{lamp}} = C_h + \left(\sin(\theta) + \frac{z_l}{f}\cos(\theta)\right)\frac{Uf}{u}. \tag{14}$$

We used the following parameters for simulations: $C_h = 2$ m, $\theta = 60°$, $f = 2.8 \times 10^{-3}$ m and the size of an image pixel to be 3.522×10^{-4} m (which was used to calculate z_l). Applying the above framework to the video data collected in Cambridge MA, we estimated the height of the Cobrahead street lamps at the intersection of Vassar street and Main street to be $\sim$12.9 m with a standard deviation of $\sim$1.5 m. Cobrahead street lamps are typically installed between $12 - 14$ m. Fig. 13 summarizes the height estimation algorithm.

VI. Luminosity Mapping: Improving Vehicle Location Estimates Through GPS-OBD Integration

Fig. 14 shows a visualization method where the height of the cylinder is proportional to the average illumination measured across the sensor array. The location data is provided by the GPS and it specifies the center of the circle. Note that the visualization overlays the night time data collected during our drive-by deployments on the day time images from Google Earth.

A key utility of our system is to compare our measurements against the lighting standards. For example, at the intersection of Portland St and Broadway in Cambridge MA, the standard recommends average illuminance of 34 lux with the ratio of average to minimum illuminance being 3 [3]. Across several measurements we found the average illuminance there to be $\sim$25 lux with the ratio of average to minimum illuminance being $\sim$4. Note that our measurements were taken at the car-top level while the standard recommends illumination values at street level. As a part of future work, we aim to develop techniques to estimate street level values from car top values.



(a) (b)

Fig. 14. Two examples of street lighting levels indicated by vertical cylinders.

GPS, which provides absolute geographic location, is typically used to obtain location information of driving vehicles and has an accuracy of a few meters [37]. Differential GPS can have an improved sub-meter accuracy [38]. GPS, in general, suffers from low reliability especially in urban driving conditions due to multipath effects and poor satellite signal [39]. Furthermore, as we discussed in Section III-D, we log the data from the GPS at a lower sampling rate as compared to the IMU and OBD-II scanner. A lower sampling rate limits the spatial resolution of location information. Hence in the context of our sensor platform, we needed to develop a robust method for using low frequency, noisy GPS data to accurately estimate vehicle trajectories. We achieved our objective by integrating on-board-diagnostics (OBD) car speed data with the GPS data and the accelerometer measurements from the IMU in an extended Kalman filter (EKF) framework, the details of which can be found in [40].

A different data collection system was used for the results presented in this section. The system had a GPS that allowed a maximum sampling frequency of 10 Hz and the data from the IMU and the OBD-II reader were directly logged into a laptop [40]. We collected the original GPS data collected at 10 Hz and then systematically downsampled it to assess the impact of a lower frequency GPS data. For each GPS downsampled sampling rate (f_s), we execute an on-line EKF estimation, implementing state updates at 10 Hz and storing the resulting state trajectory.

Fig. 15a and 15b depict two examples of vehicle trajectory reconstructions with GPS data at $f_s = 0.2$ Hz and 0.05 Hz, respectively, together with a $f_s = 10$ Hz reference trajectory in black. At a downsampling factor of 50 ($f_s = 0.2$ Hz) we achieve robust replication of the reference trajectory. At a downsampling factor of 200 ($f_s = 0.05$ Hz), a segment-wise distortion of the estimated trajectory is observed. These segments mostly exhibit errors in the vehicle heading estimate, while segment lengths show minor deviations.

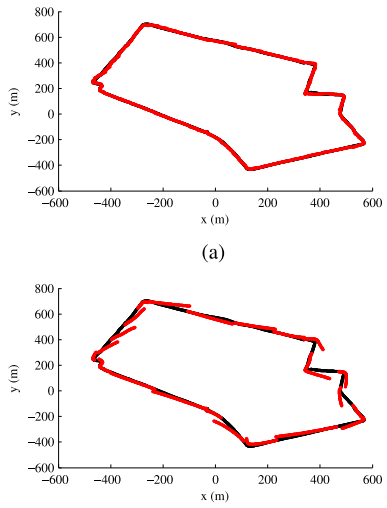


Fig. 15. Comparison of vehicle reference trajectory (in black) with EKF trajectory estimate (in red) for different GPS sampling times. (a) $f_s = 0.2$ Hz. (b) $f_s = 0.05$ Hz.

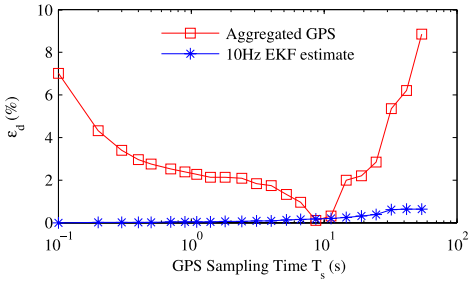


Fig. 16. Variation of ϵ_d with T_s.

We consider two error metrics: relative error in driven distance estimate (ϵ_d) and root mean squared error of position estimates (ϵ_r) [40]. In Fig.16 we report the variation of ϵ_d as a function of $T_s = 1/f_s$. For EKF-based trajectory reconstructions, ϵ_d is lower than 0.1% for $T_s < 1$ s and lower than 1% for $T_s < 50$ s. The aggregated distance increments exhibit a higher driven distance error. The U-shaped distribution of the errors can be attributed to GPS noise aggregation at low T_s and the failure to capture sufficient path curvature details at high T_s. On the other hand, our hybrid EKF method provides robust driven distance estimate that is largely unaffected by these sources of errors.

Furthermore, we extended our analysis to emulate random GPS outage. For every entry in the downsampled GPS-measurement vector, we determined its inclusion in the EKF estimation as the outcome of a Bernoulli trial with a certain probability of not observing the GPS data (denoted by p). We chose $p \in \{1/3, 1/2, 2/3\}$ and for each p ran 50 simulations to obtain mean and standard deviation of the error metrics. Fig.17a shows that ϵ_r is acceptable up to $T_s = 1$ s, after which we observe a sharp increase in ϵ_r. As expected, the error is higher for higher probabilities of GPS outage. Fig.17b shows a similar trend, though the driven distance error is acceptable even for higher values of T_s. For driving scenarios where the GPS availability may vary across the route, this suggests an adaptive sampling strategy.

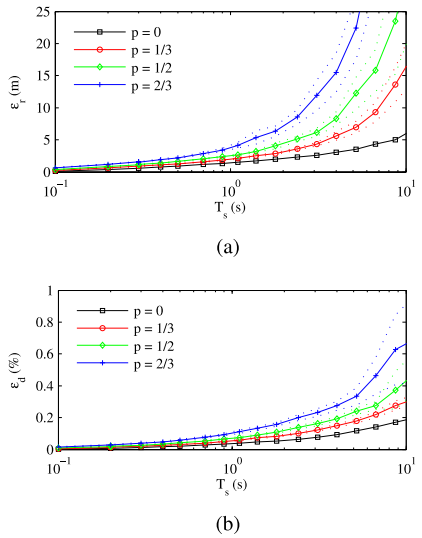


Fig. 17. Comparison of the performance metrics for different GPS outage probability (p): (a) ϵ_r and (b) ϵ_d. The solid lines indicate the mean of the metric while the dashed lines indicate mean +/− $\frac{1}{2}$ standard deviation.

Adjusting the GPS sampling rate in accordance with observed GPS outages will lead to higher accuracy in trajectory estimation.

VII. Related Work

In the recent past, mobile sensor systems have gained significant attention with the promise to implement large scale urban monitoring. Google Street View is a benchmark example where images collected by a car driving on the streets are stitched together to create 360° panoramic views [41]. The CarTel project has shown great promise in monitoring urban commute times, assessing Wi-Fi deployments and implementing automotive diagnostics with deployments spanning multiple years [42]. Researchers have also deployed vehicle mounted infrared thermography system to address identification of heat insulation leaks in residential and commercial buildings [43]. The efforts presented in this paper are in line with the above theme and vision.

Authors in [44] have recently investigated non-invasive vehicle mounted sensor system to measure street lighting levels. However their system does not include an OBD-II sensor and cameras for capturing street lamp images. Though they have discussed in detail the instrumentation challenges and data collection in their field experiments, they have not addressed signal processing and algorithmic challenges with real world noisy data.

The advancements in computer vision and image classification techniques have encouraged their use in vision based urban and environment monitoring applications. There are several challenges to image classification: the same object can be imaged with different viewpoints, illumination, occlusion, scale, deformation and background clutter [45]. Significant advancements have been made in the classification of images to identify human faces [46]–[50] which have found applications in the areas of surveillance and social networking. Other application areas that have been researched are text recognition [31]–[33] and car, motorbike or aeroplane

detection [51]–[53]. Researchers have as well explored computer vision techniques to identify street light blackouts using simulated data [54].

The first step in image classification involves converting the image to a feature vector on which a classification model can be learned. The feature vector should have discriminative power and at the same time should be robust to changes in pose, illumination, scaling and background clutter. Researchers have explored various object representation and feature construction schemes such as edge descriptors, part and structure models, constellation models, shape features, histogram of oriented gradients, bag of visual words, *etc*. [45], [55], [56].

The next step involves developing a learning algorithm which can be divided into generative and discriminative methods. In a generative model a joint distribution of the feature vector ($\mathbf{x}$) and the class label (c) is learned as $p(c, \mathbf{x})$ while in a discriminative model a conditional distribution $p(c|\mathbf{x})$ or an input-output relationship $c = f(\mathbf{x})$ is learned. Both generative models (*e.g.* mixture models, naive Bayes, topic models) and discriminative models (*e.g.* support vector machines (SVM), nearest neighbors, conditional random fields and neural networks) have been used for image classification applications [23]–[30], [45], [55].

Typically the size of the feature vector in an image classification task is large and a feature selection step is implemented before developing a learning algorithm. Feature selection or dimensionality reduction has been studied extensively in the machine learning community and some of its key benefits include: better understanding and visualization of data, faster training and developing computationally efficient classifiers, reducing the curse of dimensionality and providing better generalization performance. The feature selection techniques can be broadly divided into three different categories: wrappers, filters and embedded methods [17]–[22].

Optical flow for motion tracking has been studied extensively in the computer vision community and has found several applications in surveillance, three-dimensional shape reconstruction, video compression, medical imaging *etc*. [16], [57], [58].

Researchers have made significant progress in developing sensor fusion algorithms for combining GPS data with inertial measurement units (IMU) to improve location estimates. These methods typically involve a vehicle motion model, a sensor observation model and a Kalman filter like framework to improve GPS location estimates [40], [59], [60].

Use of urban road map information and map-matching techniques have been studied to further improve location estimates [61]–[64]. Map-matching has the potential to be integrated with our work to improve vehicle location errors. Furthermore, map data can be used to construct prior estimates on lamp locations which can reduce false negatives and positives. Lamp location errors can also be reduced through repeated measurements over time to increase the confidence interval of the estimated location.

VIII. Conclusion and Future Work

In this paper we have presented a car-top sensor system for monitoring urban street light infrastructure. Our sensor system is built of an array of lux meters for light intensity mapping and security cameras to identify street lamps from different light sources. Dedicated microcontroller and DVR systems help in logging data from these sensors. Additionally, we gathered and combined data from GPS, IMU and OBD-II sensors to obtain improved location estimates which enhance the quality of mapping and improve accuracy of lamp inventorying. The main challenge with the camera images is to separate street lamps from other shining objects. We developed a supervised learning method to identify street lamps in cropped images from other light sources. We used Fisher score and Relief score as supervised filters and principal component analysis (PCA) as an unsupervised filter in feature selection. We demonstrated a classification accuracy of 95% and a false positive rate of 3.5% with support vector machines. Once the lamps were identified, we developed a perspective geometry based methodology to estimate height of lamps, which can help in developing a stock of street lamp inventory. We further demonstrated average illumination mapping with the light sensor data and improved location estimates.

To summarize, we have described that our final prototype is capable of collecting data reliably under real driving conditions. We have also discussed the algorithms and their foundations that enable us to make inferences from the collected real world noisy data. Indeed our efforts went through several iterations, the details of which can be found in the PhD thesis [10].

The car-top sensor system presents several possible directions for future work. For example, light intensity mapping and street lamp inventory can help in identifying low light intensity areas and find candidate locations for adding street lamps. This system can aid to monitor current health of lighting infrastructure and maintenance needs. It can also be used to verify existing lamp inventory data and create new ones in a scalable way. As noted before, we would like to develop techniques to estimate street level illumination levels from car top measurements. Furthermore, we would like to establish accuracy and error bounds to compare our process with the measurement norms detailed in the lighting standards. Correlating light intensity maps with other social indicators such as crime frequency can recommend necessary actions. Our eventual goal is to develop a semi-live virtual three-dimensional street lighting model at urban scale that will enable citizens and decision makers to assess and optimize performance of nighttime street lighting.

Acknowledgment

The authors would also like to thank members of the Field Intelligence Lab, MIT for useful discussions.

References

[1] "Street lighting retrofit projects: Improving performance, while reducing costs and greenhouse gas emissions, Clinton Climate Initiative, Tech. Rep., 2010.

[2] *Road Lighting—Part 2: Performance Requirements*, European Standard EN 13201-2, European Committee for Standardization, 2003.

[3] *IES RP-8-14: Roadway Lighting (ANSI Approved)*, Illuminating Eng. Society Std. IES RP-8-14, Wall Street, NY, USA, 2014.

[4] *L.A. Faces 72-Million USD Budget Shortfall*. [Online]. Available: http://articles.latimes.com/2011/dec/03/local/la-me-city-budget-shortfall-20111203

[5] C. Jing, D. Shu, and D. Gu, "Design of streetlight monitoring and control system based on wireless sensor networks," in *Proc. 2nd IEEE Conf. Ind. Electron. Appl. (ICIEA)*, May 2007, pp. 57–62.

[6] X.-M. Huang, J. Ma, and L. E. Leblanc, "Wireless sensor network for streetlight monitoring and control," *Proc. SPIE*, vol. 5440, pp. 313–321, Apr. 2004.

[7] *Sign Up for the LightGrid Pilot Program Today!* [Online]. Available: http://www.gelighting.com/LightingWeb/na/images/CTRL012-GE-LightGrid-Pilot-Program-Sell-Sheet_tcm201-104308.pdf

[8] D. Bhadauria, O. Tekdas, and V. Isler, "Robotic data mules for collecting data over sparse sensor fields," *J. Field Robot.*, vol. 28, no. 3, pp. 388–404, 2011.

[9] M. Di Francesco, S. K. Das, and G. Anastasi, "Data collection in wireless sensor networks with mobile elements: A survey," *ACM Trans. Sensor Netw.*, vol. 8, no. 1, p. 7, Aug. 2011.

[10] S. Kumar, "Mobile sensor systems for field estimation and 'hot spot' identification," Ph.D. dissertation, Dept. Mech. Eng., Massachusetts Inst. Technol., Cambridge, MA, USA, 2014.

[11] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, vol. 1. no. 1, pp. 886–893.

[12] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan, "Fast human detection using a cascade of histograms of oriented gradients," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 2006, pp. 1491–1498.

[13] P. E. Rybski, D. Huber, D. D. Morris, and R. Hoffman, "Visual classification of coarse vehicle orientation using histogram of oriented gradients features," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2010, pp. 921–928.

[14] L. Hu, W. Liu, B. Li, and W. Xing, "Robust motion detection using histogram of oriented gradients for illumination variations," in *Proc. 2nd Int. Conf. Ind. Mechatron. Autom. (ICIMA)*, vol. 2, May 2010, pp. 443–447.

[15] O. Déniz, G. Bueno, J. Salido, and F. De la Torre, "Face recognition using histograms of oriented gradients," *Pattern Recognit. Lett.*, vol. 32, no. 12, pp. 1598–1603, 2011.

[16] B. K. P. Horn, *Robot Vision* (MIT Electrical Engineering and Computer Science). Cambridge, MA, USA: MIT Press, 1986.

[17] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Jan. 2003.

[18] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.

[19] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[20] J. Fan and J. Lv, "A selective overview of variable selection in high dimensional feature space," *Statist. Sinica*, vol. 20, no. 1, pp. 101–148, 2010.

[21] M. Pal and G. M. Foody, "Feature selection for classification of hyperspectral data by svm," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 5, pp. 2297–2307, May 2010.

[22] I. A. Gheyas and L. S. Smith, "Feature subset selection in large dimensionality domains," *Pattern Recognit.*, vol. 43, no. 1, pp. 5–13, 2010.

[23] D. D. Lewis, "Naive (Bayes) at forty: The independence assumption in information retrieval," in *Mach. Learning: ECML*. Berlin, Germany: Springer-Verlag, 1998, pp. 4–15.

[24] J. Ren, S. D. Lee, X. Chen, B. Kao, R. Cheng, and D. Cheung, "Naive Bayes classification of uncertain data," in *Proc. 9th IEEE Int. Conf. Data Mining (ICDM)*, 2009, pp. 944–949.

[25] T. Calders and S. Verwer, "Three naive Bayes approaches for discrimination-free classification," *Data Mining Knowl. Discovery*, vol. 21, no. 2, pp. 277–292, 2010.

[26] C. J. Huberty, *Applied Discriminant Analysis*. New York, NY, USA: Wiley, 1994.

[27] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. R. Mullers, "Fisher discriminant analysis with kernels," in *Proc. 9th IEEE Signal Process. Soc. Workshop Neural Netw. Signal Process.*, 1999, pp. 41–48.

[28] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[29] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, p. 27, 2011.

[30] G. Mountrakis, J. Im, and C. Ogole, "Support vector machines in remote sensing: A review," *ISPRS J. Photogram. Remote Sens.*, vol. 66, no. 3, pp. 247–259, 2011.

[31] Y. Amit and D. Geman, "Shape quantization and recognition with randomized trees," *Neural Comput.*, vol. 9, no. 7, pp. 1545–1588, 1997.

[32] Y. Amit and D. Geman, "A computational model for visual selection," *Neural Comput.*, vol. 11, no. 7, pp. 1691–1715, 1999.

[33] S. Espana-Boquera, M. J. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez, "Improving offline handwritten text recognition with hybrid HMM/ANN models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 4, pp. 767–779, Apr. 2011.

[34] Q. V. Le *et al.* (2011). "Building high-level features using large scale unsupervised learning." [Online]. Available: http://arxiv.org/abs/1112.6209

[35] R. Collins. (2007). *CSE/EE486 Computer Vision I: Introduction to Computer Vision*. [Online]. Available: http://www.cse.psu.edu/~rcollins/CSE486/

[36] S. Baker and I. Matthews, "Lucas–Kanade 20 years on: A unifying framework," *Int. J. Comput. Vis.*, vol. 56, no. 3, pp. 221–255, 2004.

[37] L. L. Arnold and P. A. Zandbergen, "Positional accuracy of the wide area augmentation system in consumer-grade GPS units," *Comput. Geosci.*, vol. 37, no. 7, pp. 883–892, Jul. 2011.

[38] L. S. Monteiro, T. Moore, and C. Hill, "What is the accuracy of DGPS?" *J. Navigat.*, vol. 58, no. 2, pp. 207–225, 2005.

[39] P. Zhang, J. Gu, E. E. Milios, and P. Huynh, "Navigation with IMU/GPS/digital compass with unscented Kalman filter," in *Proc. IEEE Int. Conf. Mechatron. Autom.*, vol. 3. Niagara Falls, Canada, Jul. 2005, pp. 1497–1502.

[40] S. Kumar, J. Paefgen, E. Wilhelm, and S. E. Sarma, "Integrating on-board diagnostics speed data with sparse gps measurements for vehicle trajectory estimation," in *Proc. SICE Annu. Conf. (SICE)*, Nagoya, Japan, Sep. 2013, pp. 2302–2308.

[41] D. Anguelov *et al.*, "Google street view: Capturing the world at street level," *Computer*, vol. 43, no. 6, pp. 32–38, Jun. 2010.

[42] B. Hull *et al.*, "CarTel: A distributed mobile sensor computing system," in *Proc. 4th Int. Conf. Embedded Netw. Sensor Syst.*, 2006, pp. 125–138.

[43] L. N. Phan, "Automated rapid thermal imaging systems technology," Ph.D. dissertation, Dept. Mech. Eng., Massachusetts Inst. Technol., Cambridge, MA, USA, 2012.

[44] H. C. Lee and H. B. Huang, "A low-cost and noninvasive system for the measurement and detection of faulty streetlights," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 4, pp. 1019–1031, Apr. 2015.

[45] L. Fei-Fei, R. Fergus, and A. Torralba. (2005). *Recognizing and Learning Object Categories*. [Online]. Available: http://people.csail.mit.edu/torralba/shortCourseRLOC/

[46] P. Viola and M. J. Jones, "Robust real-time object detection," *Int. J. Comput. Vis.*, vol. 4, pp. 1–30, Jul. 2001.

[47] M.-H. Yang, "Kernel eigenfaces vs. Kernel fisherfaces: Face recognition using Kernel methods," in *Proc. 5th IEEE Int. Conf. Autom. Face Gesture Recognit.*, May 2002, pp. 215–220.

[48] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Comput. Surv.*, vol. 35, no. 4, pp. 399–458, 2003.

[49] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004.

[50] S. Z. Li and A. K. Jain, Eds., *Handbook of Face Recognition*. Nagoya, Japan: Springer, 2011.

[51] H. Schneiderman and T. Kanade, "Object detection using the statistics of parts," *Int. J. Comput. Vis.*, vol. 56, no. 3, pp. 151–177, 2004.

[52] S. Agarwal and D. Roth, "Learning a sparse representation for object detection," in *Computer Vision—ECCV*. Berlin, Germany: Springer-Verlag, 2002, pp. 113–127.

[53] A. Vedaldi and A. Zisserman. (2011). *Image Classification Practical*. [Online]. Available: http://www.robots.ox.ac.uk/~vgg/share/practical-image-classification.htm

[54] P. N. Zanjani, M. Bahadori, and M. Hashemi, "Monitoring and remote sensing of the street lighting system using computer vision and image processing techniques for the purpose of mechanized blackouts (development phase)," in *Proc. 22nd Int. Conf. Exhibit. Electr. Distrib.*, 2013, pp. 1–4.

[55] A. Torralba. (2012). *6.869: Advances in Computer Vision*. [Online]. Available: http://6.869.csail.mit.edu/fa12/

[56] D. Zhang and G. Lu, "Review of shape representation and description techniques," *Pattern Recognit.*, vol. 37, no. 1, pp. 1–19, Jan. 2004.

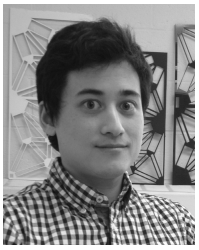
[57] T. Senst, V. Eiselein, and T. Sikora, "Robust local optical flow for feature tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 9, pp. 1377–1387, Sep. 2012.

[58] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, "Recent advances and trends in visual tracking: A review," *Neurocomputing*, vol. 74, no. 18, pp. 3823–3831, Nov. 2011.
[59] W. Ding, J. Wang, C. Rizos, and D. Kinlyside, "Improving adaptive Kalman estimation in GPS/INS integration," *J. Navigat.*, vol. 60, no. 3, pp. 517–529, 2007.
[60] J. Georgy, T. Karamat, U. Iqbal, and A. Noureldin, "Enhanced MEMS-IMU/odometer/GPS integration using mixture particle filter," *GPS Solutions*, vol. 15, no. 3, pp. 239–252, 2011.
[61] C. Smaili, M. E. E. Najjar, and F. Charpillet, "Multi-sensor fusion method using dynamic Bayesian network for precise vehicle localization and road matching," in *Proc. 19th IEEE Int. Conf. Tools Artif. Intell. (ICTAI)*, vol. 1, Oct. 2007, pp. 146–151.
[62] M. A. Quddus, W. Y. Ochieng, and R. B. Noland, "Current map-matching algorithms for transport applications: State-of-the art and future research directions," *Transp. Res. C, Emerg. Technol.*, vol. 15, no. 5, pp. 312–328, 2007.
[63] J. S. Greenfeld, "Matching GPS observations to locations on a digital map," in *Proc. Transp. Res. Board 81st Annu. Meeting*, 2002.
[64] P. Newson and J. Krumm, "Hidden Markov map matching through noise and sparseness," in *Proc. 17th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst. (GIS)*, New York, NY, USA, 2009, pp. 336–343.

Sumeet Kumar received the bachelor's degree from the Indian Institute of Technology Kanpur, the M.S. degree in mechanical engineering from the Massachusetts Institute of Technology (MIT), in 2009, and the Ph.D. degree in mechanical engineering from MIT in 2014. He is currently a Quantitative Researcher with Dean Capital Investments. His research interests include machine learning, statistical signal processing, data science, sensor systems, and thermal modeling. He received the Pappalardo Fellowship and the Academic Excellence Awards in 2005 and 2006.

Ajay Deshpande received the B.Tech. and M.Tech. degrees from the Indian Institute of Technology Bombay in 2001, and the master's degrees in mechanical engineering, and electrical engineering and computer science and the Ph.D. degree in mechanical engineering from the Massachusetts Institute of Technology (MIT), in 2006 and 2008, respectively. He was a Post-Doctoral Associate with the Laboratory for Manufacturing and Productivity, MIT. He is currently a Research Staff Member with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY. He was a recipient of the MIT Presidential Fellowship.

Stephen S. Ho received the B.Sc. degree from Cornell University, and the M.S. and Ph.D. degrees from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, all in mechanical engineering. He has been a Research Scientist with the Field Intelligence Laboratory, MIT, since 2007. His research interests center around the application of information toward improving systems, processes, and resource management. His current research includes sensors and inference of vehicles and vehicle traffic, lighting and municipal assets for urban environments, and energy efficiency. His Ph.D. research evaluated the impact rich RFID information has on warehouse operations and the opportunities for increased efficiency that stem from that additional information.



Jason S. Ku is currently pursuing the Ph.D. degree with the Department of Mechanical Engineering, Massachusetts Institute of Technology. He primarily studies computational geometry, particularly questions involving the relationship between two and three dimensions. His research topics include the design and fabrication of static and transformable folding structures from sheet-like material, 3-D object reconstruction algorithms from lower dimensional projections, and 3-D projection.

Sanjay E. Sarma received the bachelor's degree from the Indian Institute of Technology, the master's degree from Carnegie Mellon University, and the Ph.D. degree from the University of California at Berkeley. He was the Fred Fort Flowers (1941) and Daniel Fort Flowers (1941) Professor of Mechanical Engineering at MIT. He is the first Dean of Digital Learning at MIT. He co-founded the Auto-ID Center at MIT and developed many of the key technologies behind the EPC suite of RFID standards now used worldwide. He was also the Founder and CTO of OATSystems, which was acquired by Checkpoint Systems (NYSE: CKP) in 2008. He serves on the Boards of GS1, EPCglobal, and several startup companies, including Senaya and ESSESS. He also work with Schlumberger Oilfield Services, Aberdeen, U.K., and the Lawrence Berkeley Laboratories, Berkeley, CA. He has authored over 75 academic papers in computational geometry, sensing, RFID, automation, and CAD, and is a recipient of numerous awards for teaching and research, including the MacVicar Fellowship, the Business Week eBiz Award, and the Informationweek's Innovators and Influencers Award. He advises several national governments and global companies.