Estimation and navigation methods with limited information for autonomous urban driving

Jordan B. Chipka | Mark Campbell

Department of Mechanical and Aerospace Engineering, Cornell University Ithaca, New York, New York

Correspondence
Jordan B. Chipka, Department of Mechanical and Aerospace Engineering, Cornell University, New York, NY 14850. Email: jbc274@cornell.edu

Funding information
NSF GFRP, Grant/Award Number: DGE1144153; NSF, Grant/Award Number: IIS-1724282

Autonomous driving in dense urban areas presents an especially difficult task. First, globally localizing information, such as GPS signal, often proves to be unreliable in such areas due to signal shadowing and multipath errors. Second, the high-definition environmental maps with sufficient information for autonomous navigation require a large amount of data to be collected from these areas, significant postprocessing of this data to generate the map, and then continual maintenance of the map to account for changes in the environment. This paper addresses the issue of autonomous driving in urban environments by investigating algorithms and an architecture to enable fully functional autonomous driving with little to no reliance on map-based measurements or GPS signals. An extended Kalman filter with odometry, compass, and sparse landmark measurements as inputs is used to provide localization. Real-time detection and estimation of key roadway features are used to create an understanding of the surrounding static scene. Navigation is accomplished by a compass-based navigation control law. Experimental scene understanding results are obtained using computer vision and estimation techniques and demonstrate the ability to probabilistically infer key features of an intersection in real time. Key results from Monte Carlo studies demonstrate the proposed localization and navigation methods. These tests provide success rates of urban navigation under different environmental conditions, such as landmark density, and show that the vehicle can navigate to a goal nearly 10 km away without any external pose update at all. Field tests validate these simulated results and demonstrate that, for given test conditions, an expected range can be determined for a given success rate.

1 | INTRODUCTION

Autonomous driving in dense urban environments presents a unique challenge due to the lack of reliable information sources for localization and planning. For example, GPS measurements do not have guaranteed availability due to signal shadowing in urban canyons, tunnels, and parking structures. Furthermore, GPS measurements often experience multipath errors in urban environments, which cause positioning systems based on GPS to be an unreliable information source for autonomous agents. Even highly accurate, state-of-the-art positioning systems struggle to provide the level of localization needed for autonomous driving due to the difficulties that dense urban environments present. Furthermore,
the high-fidelity maps that enable autonomous driving are highly sensitive to environmental changes and expensive to obtain in regard to time, money, and resources. Storing these highly detailed maps on-board the vehicle is unfeasible for all maps in all locations or for environments that regularly change; for example, construction areas with resurfaced roads, repainted lane markings, and lane or road closings. One approach to overcome this difficulty is to rely on data connections to provide the vehicle access to these maps; however, these connections can be weak, spotty, or nonexistent in urban areas.3 Finally, security of autonomous vehicles is also a major concern, as both maps and GPS measurements can be spoofed, jammed, and/or altered.4 Given these challenges, this study investigates alternative architectures and sources of information that may be used for robust navigation of urban roadways, or to augment current and future navigation systems.

Currently, most autonomous driving systems use high precision GPS signals and vehicle sensors to localize within a high-definition (HD) environmental map.5 Many of the front-running companies in this field, such as Waymo, Uber, and Ford, dedicate entire teams to produce and maintain their HD environmental maps. These maps generally contain the road network, road features (such as painted road lines and markings, barriers, traffic lights, and traffic signs), and sensor-specific layers (such as a lidar point cloud layer) to be used in localization. Since so much information is needed to comprise an HD map, approximately 1 GB of data is typically required for every mile mapped.6 The primary method of obtaining such a map involves deploying sensory vehicles to acquire 360-degree lidar and/or camera data of a certain environment multiple times.7,8 It is only after this data is collected, followed by heavy postprocessing to create the final HD map, that a vehicle can drive autonomously using the precise map in the given environment. While recent developments utilizing a pose-based GraphSLAM estimator provide even more precise mapping, these approaches remain highly time, cost, and labor intensive.9 An alternative method of obtaining an HD map of the environment involves the use of unmanned aerial vehicles, equipped with lidar and camera systems, as a novel platform for photogrammetry.10 However, this method also requires time-consuming data collection with heavy data postprocessing. The GPS-fused simultaneous localization and mapping (SLAM) techniques have been proposed to reduce the complexity of the mapping task.11,12 However, the assumption of consistently receiving these GPS measurement updates is not valid for urban applications, such as in urban canyons like Manhattan and Chicago. Therefore, a localization solution that relies upon consistent accurate GPS measurements will ultimately prove to be unreliable. Overall, due to the extreme complexity and laborious nature of the HD mapping task, automakers such as General Motors, BMW, and Volkswagen have relied on third-party services, such as HERE, TomTom, and MobilEye, to provide these HD maps.13 Despite there being several drawbacks to HD maps for navigation, there is no alternative for intelligent mobility solutions in the current state of technology.14

There have been various recent studies that aim to allow for autonomous driving capabilities without the use of a priori HD mapping. Online SLAM can be applied to autonomous driving to alleviate the need for high-fidelity maps.15,16 For instance, FAB-MAP, a topological SLAM technique, allows for appearance-based navigation.17 Similarly, SeqSLAM is another visual SLAM technique and also aims to address the difficulty associated with changing environmental conditions.18 Although both methods do not rely on GPS measurements or an a priori HD map, they operate purely in appearance space and make no attempt to localize the vehicle in metric coordinates, either global or map based; in other words, these methods act as scene-matching algorithms. Without tracking the vehicle in metric coordinates, it is impossible to localize the vehicle in between two matched scenes. Therefore, a sufficiently dense map of images is required to adequately localize the vehicle. As the size of the environment grows, scalability becomes a major concern due to sharply rising computational requirements and rapidly diminishing performance.19

The problems of navigating without GPS and a highly detailed map can be addressed separately in a less computationally expensive manner. Techniques have been proposed in recent years based on accurate self-localization in mapped environments in an attempt to reduce the vehicle’s reliance upon GPS.20-24 However, these methods still rely heavily on a priori map information, as they require either a lidar or vision sensor layer to be included in the environmental map. Likewise, sufficient localization in an extended GPS blackout can be provided by a posterior pose algorithm, which augments GPS and an inertial navigation system with vision-based measurements of nearby lanes and stop lines referenced to a known map of environmental features.25 Through the use of this algorithm, a converged and accurate position estimate was retained during an 8-minute GPS blackout. Alternative methods, which take a perception approach, demonstrate the ability to automatically map the 3D position of traffic lights, enable real-time autonomous off-road navigation using semantic mapping, and allow for real-time joint semantic reasoning for road scene understanding.26-28 Additionally, a study has been shown to provide self-localization through the use of visual odometry along with map data from OpenStreetMap.29 These techniques address key mapping and localization challenges; however, they remain limited in terms of real-time mapping and do not provide the capability to infer the full roadway intersection scene required for navigation.
While recent research efforts have been made to address the challenges of driving either in GPS-denied circumstances or without a prior HD map, there is considerably less research to simultaneously address both challenges in a computationally-efficient manner. This work proposes a system architecture with novel localization, navigation, and scene estimation techniques to enable autonomous navigation without GPS measurements or prior HD map data. First, this paper proposes a novel scene estimation framework that relies on real-time sensor data, rather than prior map data, to enable robust autonomous navigation in urban environments. This method utilizes computer vision techniques to detect environmental cues which, when reasoned together in a probabilistic framework, are used to infer the static scene around the vehicle in real-time. The focus here is on inferring the minimum amount of information needed to navigate intersections since intersections offer unique challenges and are critical to overall navigation. This inferred information provides both the location and configuration of the intersection. Second, this paper explores how far a vehicle can travel with a minimal amount of prior map information; for the purposes of this study, the limited map information comes as a sparse landmark map. This results in a navigation solution with much better scalability than many other techniques in the research community.

Importantly, while the focus of this paper is on navigation with limited information, the proposed approach could also be used to supplement current navigation systems that use GPS, HD maps, SLAM techniques, or other self-localization methods. For example, the proposed work could provide vital support to autonomous vehicles for cases in which the navigation system experiences difficulties such as during a GPS blackout in a dense urban environment, or during a cybersecurity attack. This robustness to infrequent, yet crucial events, is critical for resilient long-term navigation solutions. Additionally, the use of a more sophisticated sensor suite could further improve the purely computer vision-based perception techniques described in this paper (e.g., lidar would offer the added benefit of lighting invariance). However, since this paper emphasizes the ability to drive autonomously despite a lack of commonly relied upon information, including rich sensor data provided by lidar and other expensive sensors, the inclusion of lidar data was considered out of scope for this study.

2 | SYSTEM ARCHITECTURE

The system architecture developed for this study is shown in Figure 1. This system diagram contains familiar elements in autonomous driving, such as steering and speed controllers, an object tracker, and a path generator. However, the pose estimator, navigation algorithm, and static scene estimator are updated from their typical form to face the difficulties associated with the lack of map information and GPS measurements. The proposed approach relies on local sensors to allow for real-time control of the vehicle (i.e., staying in a lane). Therefore, precise in-lane localization from HD maps or GPS is not required. Rather, high-level localization is provided by the lightweight sparse pose estimator, which utilizes only odometry measurements, compass measurements, and sparse map-based measurements, which come from an on-board sparse map of landmarks with corresponding coordinates. This estimator is termed lightweight due to the limited use of sensor measurements. The sparse map-based measurements are produced by matching features in the image obtained from the vehicle’s camera to features in a landmark image within the sparse map. Then, using the corresponding coordinates of the landmark image, the vehicle’s pose estimate is updated.

The static roadway scene estimator is used to detect and classify key roadway features in real time without the use of prior HD map data to infer only the map attributes required for autonomous navigation. This approach uses belief propagation to combine the information from many environmental cues to enhance robustness and minimize the mistakes that a single detector may provide. Such key features required for navigation include lane line location and type, intersecting roads, turn options at intersections, traffic lights, stop signs, and other navigation-critical road signs, such as one-way and do-not-enter signs. These key features supply the bare minimum information needed to navigate roadways (i.e., they define the viable routes of travel for the vehicle at a high-level). All additional low-level information needed for path planning and control can then be obtained from local sensors. The key roadway scene information, along with the inertial pose estimate, feed into an intersection navigation algorithm, which then probabilistically determines the best route to take to reach the goal. This high-level navigation scheme is provided by a compass-based navigation control law.

2.1 | Static roadway scene estimator

To satisfy the perception needs of autonomous navigation without prior detailed map data, a static roadway scene estimator is proposed. Figure 2 shows the detailed pipeline for the proposed real-time static roadway scene estimator. A raw image from the vehicle’s front-facing camera is the input to the pipeline. This image is then passed to several computer
vision components, which extract environmental cues to infer an understanding of the static scene to a sufficiently detailed level for autonomous navigation. Table 1 summarizes the environmental cues used for this study. The output of the static scene estimator are lane line locations and types (for low-level control), an estimated vehicle-relative intersection location, and probabilities for key intersection features. These features include the intersection itself, as well as its turning options (ie, left turn, right turn, or continue straight). A marginal probability is computed for each feature in real time. This information is passed into the intersection navigation algorithm, which directs the vehicle at a high level. The components for the static roadway scene estimator are described in further detail in the following sections.

2.1.1 Intersection estimator
Strong environmental cues are used to reliably detect and locate an intersection relative to the vehicle. These strong cues include stop sign and traffic light detections since these are most conspicuous for detecting intersections. Furthermore, due to their relatively close proximity to the intersection itself (sufficiently close to enable the high-level navigation proposed in this work), the estimated vehicle-relative location can be used as an approximate location for the intersection.
TABLE 1  Summary of environmental cues

| Environmental cues         | Detection method     | Contribution to overall scene understanding                                      |
|----------------------------|----------------------|-----------------------------------------------------------------------------------|
| Lane lines                 | Mobileye 560         | Lane line type, shape, and location relative to vehicle                           |
| Traffic light              | Detectron            | Presence of intersection, and approximate intersection location relative to vehicle |
| Stop sign                  | Detectron            | Presence of intersection, and approximate intersection location relative to vehicle |
| Road surface               | MultiNet             | Possible directions of travel at intersection                                      |
| Cross traffic vehicles     | Detectron            | Possible turning options at intersection                                           |
| Parked vehicles along crossroad | Detectron         | Possible turning options at intersection                                           |
| Oncoming/outgoing vehicles | Mobileye 560         | Possible direction of travel at intersection                                      |
| One-way signs              | Sliding window object detection | Unviable turning options at intersection                                           |
| Do-not-enter signs         | Sliding window object detection | Unviable direction of travel at intersection                                      |

To allow for vehicle-relative location estimation, the stop signs and traffic lights must be detected with a high level of precision. The proposed method uses Detectron, an open-source, state-of-the-art object detection platform developed by Facebook AI Research, which implements advanced object detection algorithms, such as Mask R-CNN.\(^{31,32}\) Using Detectron's implementation of Mask R-CNN trained on the COCO data set, common roadway objects, such as traffic lights, stop signs, and vehicles, can be accurately detected and classified within the proposed architecture.\(^{33}\) The output of Detectron includes a mask of pixels covering the detected object in the image, a label for the object, and a score between 0 and 1. This score value, while not a true probability, does give a level of confidence. For this study, only detections with a score greater than 0.9 were considered. The detector developed for this work outputs two binary variables, shown in Equation (1), indicating the presence \(z_{\text{TL}} = 1\) or \(z_{\text{SS}} = 1\) or absence \(z_{\text{TL}} = 0\) or \(z_{\text{SS}} = 0\) of a stop sign or traffic light.

\[
\begin{bmatrix}
  z_{\text{TL/SS}}
\end{bmatrix} = \begin{bmatrix}
  z_{\text{TL}} \\
  z_{\text{SS}}
\end{bmatrix} \tag{1}
\]

Using the 0.9 score threshold and comparing the detections found during testing with truth data, detection precision and recall values of 0.97 and 0.81 were found for stop signs and 0.95 and 0.62 for traffic lights. Due to these high precision values, false positive detections were not a major concern during testing; however, to mitigate the effects of false positive detections even further, a simple tracking method could be employed. This test data set comes from the 70 tests described later in this paper, in which a test vehicle collects camera data at 30 fps for 20 seconds while approaching an intersection. In these tests, the stop sign or traffic light was typically occluded or too far to be seen for the first 10 seconds of the test. Therefore, approximately half of all images in this data set were negative examples. Figure 3 shows example stop sign and traffic light detections that were found during testing.

To obtain a vehicle-relative intersection location estimate, the segmented shapes from Detectron for both traffic lights and stop signs are transformed from the image plane to a vehicle-relative position. These vehicle-relative locations are approximated by using a relationship between the segmented shape's height in the image plane and its relative position to the vehicle. This relationship was determined through a data-driven approach, in which traffic light and stop sign segmentations were computed at known distances from the object in thousands of images at various intersections under diverse conditions. This data, as well as the fitted curve that describes the relationship between segment height and distance from detected object, is shown in Figure 4. The predicted distance to a stop sign had a standard deviation of 1.7 m and the predicted distance to a traffic light had a standard deviation of 5.9 m. Given the scope of this study, these error values were considered acceptable. However, to more accurately predict these distance values, a more sophisticated technique could be used, such as incorporating a stereo camera or using a monocular depth estimation approach.\(^{34}\) For this study, the predicted distance to a stop sign was considered an approximate distance to the start of the intersection, and the predicted distance to a traffic light was considered an approximate distance to the end of the intersection. For measurements that returned multiple detections, the mean predicted distance was used. To generalize this approach to account for intersections with no stop sign or traffic light, the crossroad cues described in the following sections could be incorporated in a similar manner to infer the intersection location.

The raw predicted vehicle-relative intersection locations then pass into a 1D Bayes estimator, which fuses measurements from the vehicle’s odometry, to provide a vehicle-relative intersection location estimate in real-time. The relevant
FIGURE 3  Example detections using Detectron for (A) stop sign, with a score of 0.99, and (B) traffic light, with a score of 0.90.

FIGURE 4  Data, as well as fitted curve, relating segmented height to vehicle’s distance to (A) stop sign and (B) traffic light for various intersections under diverse conditions. The predicted distance for stop sign had a standard deviation of 1.7 m and the predicted distance for traffic light had a standard deviation of 5.9 m.

The equations for this Bayes estimator are shown in Equations (2) to (6).

\[
\begin{align*}
\hat{x}_k &= \hat{x}_{k-1} - dt \\
\hat{\sigma}_k^2 &= \sigma_{k-1}^2 + q \\
K_k &= \frac{\hat{\sigma}_k^2}{\hat{\sigma}_k^2 + r} \\
\hat{x}_k &= \hat{x}_k + K_k(z_k - \hat{x}_k) \\
\sigma_k^2 &= \hat{\sigma}_k^2 - K_k\hat{\sigma}_k^2
\end{align*}
\]
In Equation (2), $\hat{x}_{k-1}$ is the 1D vehicle-relative location estimate of the intersection from the rear axle of the vehicle at time step $k - 1$, $dt$ is the distance traveled by the vehicle between time steps (obtained from the vehicle’s odometry), and $\hat{x}_k$ is the predicted location estimate at time step $k$. In Equation (3), $\sigma^2_{k-1}$ is the location estimate error variance at time step $k - 1$, $\hat{\sigma}^2_k$ is the predicted location error variance at time step $k$, and $q$ is the odometry noise variance. The location estimate and error variance are updated in Equations (5) and (6) after receiving a measurement from the detected stop sign or traffic light, $z_k$. $K_k$ is a Kalman gain from Equation (4), and $r$ is the measurement noise variance, which was 1 m and assumed to be constant for simplicity of this calculation.

### 2.1.2 Road segmentation and direction prediction

As one of the cues to determine the possible directions of travel at an intersection, road segmentation is used to detect the road surface in an image. Based on the shape of the road segmentation, possible directions of travel are then inferred. MultiNet is used to perform real-time semantic reasoning of the roadway scene. A pretrained model using the Kitti data set was fine-tuned to be used in this study by adjusting threshold parameters and image characteristics. The output of MultiNet is a segmented image of all pixels, where road surface is detected. Based on the location of these pixels in the image, possible directions of travel at an intersection can be predicted. This prediction is performed by examining three regions of interest (ROIs) in the image, where road surface is expected to be for an intersection: straight, left, and right.

Figure 5 shows the output of MultiNet, as well as the raw RGB image, for two example cases; the ROIs are highlighted in all images. A specific road direction is predicted to exist if more than 25% of the pixels in the corresponding ROI are detected to be road. For instance, in Figure 5A, more than 25% of the pixels in the left and right regions are predicted to be road, whereas less than 25% of the pixels in the center region are predicted to be road. Therefore, for the case shown in Figure 5A, a road direction of left and right is predicted, whereas straight is not. The percentage threshold of 25% was determined by tuning it to minimize false detections while maximizing recall. This prediction does not attempt to measure any characteristics of the detected road but strictly predicts whether there exists road in a given direction that could be traveled upon. This condition is described in Equation (7), where $\text{ROI}(p_{ij} = 1)$ represents the count of pixels where road surface is detected in the ROI, and ROI represents the total number of pixels in the given ROI. The output of this detector, shown in Equation (8), are three binary variables representing the detected road directions at the intersection, ie, right, straight, and left. From comparing this detector’s output during testing to truth data, a detection precision of 0.77 and a recall of 0.48 were found. This comparison was made by checking truth data to see whether a positive detection of a road direction did, in fact, correctly reflect the true environment around the vehicle. Lighting
conditions and nearby driveways and parking lots were the primary factors for much of this detector’s difficulty. This detector could be improved by incorporating depth information through the use of a stereo camera or a monocular depth estimation approach, rather than relying on the proposed ROI-based approach. Depth information, combined with intrinsic and extrinsic calibration data for the camera, would allow pixels in the image plane to be projected into the real world. Although this approach would offer a more accurate location of the crossroad relative to the vehicle, this was considered to be outside the scope for this study.

\[
\frac{\text{ROI}(p_{ij} = 1)}{\text{ROI}} > 0.25 
\]

(7)

\[
\overrightarrow{z_{\text{road}}} = \begin{bmatrix}
  z_{\text{road}_x} \\
  z_{\text{road}_y} \\
  z_{\text{road}_z}
\end{bmatrix}
\]

(8)

### 2.1.3 Cross traffic vehicle detection and tracking

Detecting and tracking the movement of vehicles at an intersection is a cue that provides information regarding the manner in which the intersection can be navigated. For instance, if an autonomous vehicle is approaching an intersection and detects other vehicles traveling from left to right across its field of view, then a reasonable inference can be made that the autonomous vehicle has the option to turn right at that intersection. With this rationale, segmented vehicle shapes from Detectron are passed into a cross traffic vehicle tracker to determine the possible directions of travel at an intersection. Figure 6 shows vehicle detections from consecutive time steps that were obtained during testing. While any object tracker could be used, given the relatively simple goal of detecting nonzero velocity and heading only, a stand-alone tracker was developed by comparing consecutive segmented frames and inferring the direction of travel by the segmented vehicles. The correspondence between two segmented shapes in two consecutive frames is determined.

![FIGURE 6 Detectron cross traffic detections from consecutive time steps, (A) and (B). These detections resulted in a positive left cross traffic measurement due to the segmented vehicle’s predicted direction of travel, based on its size, geometry, and horizontal motion](image-url)
by comparing the shapes’ sizes and locations in the images. If the shapes’ sizes differed by less than 3% and its locations differed by less than 5% of the total image width, then the shapes are assumed to be from the same object. To mitigate false tracks, the tracker only considers vehicle segments with a length-to-height ratio greater than 2 and a confidence score above 0.95. The length-to-height ratio is used to limit the approach to only segmented vehicle shapes with a side profile, which generally indicates that the vehicle is within the cross traffic. The confidence score threshold is used to filter out partially occluded vehicles, which generally provide lower scores. To further mitigate false tracks, the tracker only considers tracks that indicate steady horizontal motion across the image between consecutive frames (i.e., the expected motion of vehicles belonging to the cross traffic). This filters out parked vehicles in parking lots, driveways, and along the road. All of the previously stated parameters were tuned to maximize the performance of the tracker, both precision and recall. The output of this cross traffic tracker, $\overline{z}_{ct}$, is shown in Equation (9).

$$\overline{z}_{ct} = \begin{bmatrix} z_{ctR} \\ z_{ctL} \end{bmatrix}$$  \hfill (9)

The output is given as two binary variables representing the presence of a right-moving vehicle and a left-moving vehicle for any two consecutive frames. These binary variables are 0 for no detection of cross traffic, and 1 when a segmented vehicle shape meets all previously described conditions for a given direction. From comparing the tracker output during testing to truth data, using 70 tests with up to 100 detections in a given test, a detection precision of 0.86 and a recall of 0.74 were found.

### 2.1.4 Parked vehicle detection and orientation prediction

Similar to detecting the direction of cross traffic at an intersection, detecting the orientation of parked vehicles along an intersecting road can give insight regarding the possible directions of travel at an intersection. For instance, the parked vehicles along the intersecting road seen in Figure 7 indicate that turning left is most likely a viable option at that intersection. The same size, geometry, and score constraints used for the cross traffic detector are also used for this detector to filter oncoming, occluded, and other undesired vehicle detections. To filter out moving vehicles, this detector only considers detections of vehicles that do not show evident horizontal or vertical motion across the image between consecutive detections. Once the segmented vehicle shapes meet these conditions, their orientation is predicted by evaluating their shape. Due to the standard aerodynamic shape of vehicles, the nose of the vehicle can predictably be differentiated from the tail of the vehicle by comparing their heights. The end of the vehicle with the lower height is assumed to be the nose of the vehicle. The output of this parked vehicle detector, $\overline{z}_{pc}$, are two binary variables, representing the presence of a right-facing parked vehicle and a left-facing parked vehicle on the crossroad, as shown in Equation (10).

$$\overline{z}_{pc} = \begin{bmatrix} z_{pcR} \\ z_{pcL} \end{bmatrix}$$  \hfill (10)

These binary variables are 0 for a given direction when no parked vehicles are detected in that direction, and 1 for a given direction when a segmented vehicle shape meets all previously described constraints (size, geometry, score, and motion) and predicts the corresponding orientation based on its geometry. Comparing this detector’s output during testing to truth data, which involved 70 tests at various intersections, a detection precision of 0.83 and a recall of 0.66 were found.

**FIGURE 7** Detections of parked vehicles using Detectron, and predictions of their orientation. These detections resulted in a positive left turn measurement due to the predicted orientation of the segmented vehicles, based on their geometry. In the figure, the green lines on the segmented shapes indicate the measured heights of the right and left end of the vehicle, and the blue line at the front of the segmented shape indicates the resulting prediction for the nose of the vehicle.
2.1.5 Lane line and oncoming/outgoing vehicle detection

Additional environmental cues that can be used to detect possible directions of travel at intersections are lane lines and oncoming or outgoing vehicles. For this study, Mobileye 560 was used to provide lane line and obstacle data in real time.\textsuperscript{36} The lane line data is given as a second-order polynomial function relative to the ego vehicle, and the obstacle data includes type (e.g., vehicle, pedestrian, and cyclist), relative longitudinal and lateral position, and relative longitudinal velocity. The lane line information (type and location) not only is primarily needed for the low-level path generator (see Figures 1 and 2) but also can be used to detect the option to continue straight at intersections that have easily visible lane lines from across the intersection. Furthermore, detecting oncoming or outgoing traffic beyond an intersection provides cues that there is a possibility to continue straight through an intersection. Mobileye detections are only used to detect oncoming or outgoing traffic, as opposed to detecting cross traffic, due to the device's narrow field of view. An example of this detector is shown in Figure 8. If lane lines are detected to continue straight past an intersection, the binary variable $z_{\text{lanes}} = 1$ is given. If an oncoming or outgoing vehicle is detected past an intersection, the binary variable $z_{\text{on/out}} = 1$ is given. According to Mobileye documentation, all detections have a precision of 0.99. However, since this study is using its detections to predict road direction at an intersection, this precision is lower due to false measurements caused by vehicles parked in driveways and parking lots. Comparing the predicted road direction from these measurements to truth data during testing (70 tests at various intersections), a detection precision of 0.96 and a recall of 0.78 were found.

2.1.6 One-way and do-not-enter sign detection

Not all directions of travel that are predicted using the previously described environmental cues are always possible options for travel. For instance, the vehicle can detect road surface to the left at an intersection and, therefore, predict that the intersection contains an option to turn left. However, if there exists a one-way sign in the opposite direction of this road, then this left-hand turn option is not viable. Therefore, one-way and do-not-enter sign detectors were developed for this work to ensure robust navigation capabilities. These detectors use the sliding window detection technique.

**FIGURE 8** Mobileye detections of oncoming and outgoing vehicles. (A) Raw RGB image of the intersection scene. (B) Top-down view of the Mobileye detections. Detections of the two white vehicles parked along the side of the road, as well as the outgoing silver vehicle, are shown by the yellow boxes.
by scanning an image using a pyramid representation and extracting ORB features. These extracted features from the sliding window are then matched with the features from a sample image of the sign using the ratio test. The threshold for the ratio test was tuned to ensure that the detection precision was above 0.95. Although tuning to this specification resulted in a lower recall of 0.38, this was deemed acceptable to maintain high precision. To increase computation speed, the top and bottom areas of the image are cropped out, leaving only the area where one-way and do-not-enter signs are typically seen. To improve the performance of this detector, a deep neural network could be used. However, this approach was not used in this work due to the need to first collect a large training set of images similar to the images that would be seen at test time. Since there was not an available and sufficiently large data set of one-way and do-not-enter signs on which to train, a classical classifier was instead used for this study. The output of the do-not-enter sign detector is a binary variable, \( z_{DNE} \), giving a value of 1 when a do-not-enter sign is detected and 0 when one is not. The output of the one-way sign detector are binary variables representing the detection of left or right one-way signs, as shown in the following in Equation 11:

\[
\begin{bmatrix}
  z_{ow_R} \\
  z_{ow_L}
\end{bmatrix}
\]

(11)

### 2.1.7 Belief propagation

Once the environmental cues are obtained from the detection techniques described in Sections 2.1.1 to 2.1.6, the presence of key intersection features to enable high-level navigation can be probabilistically determined. For this study, these features include the presence of an intersection itself, as well as its possible directions of travel (ie, left, right, or straight). Belief propagation is used to compute the probabilities associated with each intersection feature. Figure 9 shows a factor graph displaying the connections between the observed environmental cues, \( z \) (bottom of graph), and the latent intersection features, \( f \) (top of graph). Black edges connect cues and features that are directly related, whereas red dashed edges connect cues and features that are mutually exclusive. The factor graph is segmented into three independent sub-graphs and the corresponding probabilities are computed separately. The marginal probability of each intersection feature is initialized to 0.5. At each time step, a message, \( \mu_i \), is passed from each observed cue to its corresponding intersection feature. The values of the messages correspond to the detection probabilities described in the previous sections, whether the observed cue is detected and whether the cue and feature are directly related or mutually exclusive.

The marginal probability of a given feature is computed at each time step using the product of all incoming messages to the given feature node, as shown in Equation (12).

\[
p(f_j) \propto \prod_{i \in N(f_j)} \mu_i(f_j)
\]

(12)
In the aforementioned equation, \( p(f_j) \) is the marginal probability of feature \( f_j \), \( \mu(f_j) \) is the message from an observed cue to feature node \( f_j \), and \( N(j) \) represents the set of all cues that are linked to feature node \( f_i \).

For the parked vehicle, oncoming/outgoing vehicle, and cross traffic cues, each detection results in a sent message. Therefore, if multiple left-facing parked vehicles are detected, multiple messages are sent to the left turn node, resulting in a higher left turn probability. If an environmental cue is not detected, a neutral message (\( \mu_i = 0.5 \)) is sent, hence providing neither affirming nor dissenting information regarding the intersection feature. To mitigate false positives before an intersection is detected, the messages sent to the left turn, right turn, and straight intersection features are always neutral until the probability associated with the intersection feature exceeds 0.9 and the vehicle's estimated distance from the intersection is less than 30 m, indicating that an intersection is, in fact, present and the vehicle is close enough to detect the directions of travel. Finally, if an environmental cue is not detected after the vehicle comes within 20 m of the intersection, then a very weak dissenting message (\( \mu_i = 0.49 \)) is sent to the node for the corresponding direction of travel.

### 2.2 Lightweight sparse pose estimation

An inertial pose estimator is developed to enable high-level navigation assuming no detailed prior map information and no GPS measurements. To make the localization problem feasible without these critical information sources, the start and goal coordinates of the vehicle are assumed to be known. Second, the pose estimator employs dead reckoning using odometry and compass measurements to estimate the pose of the vehicle within a local frame of reference. Finally, in part of this study, map-based measurement updates to sparse, but known, landmarks are used to supplement the dead reckoning localization strategy. Since dead reckoning using odometry measurements alone diverges over time, a compass measurement update, which directly measures the heading of the vehicle in inertial space, is employed to improve the pose estimator accuracy notwithstanding relatively high compass uncertainty. To enable even longer driving, the pose estimator can be improved using sparse map-based measurement updates. The proposed approach then considers two key questions in this work: (1) How far can the vehicle travel and remain sufficiently localized during a GPS blackout and without prior map information? (2) After incorporating a map of landmarks, what level of landmark density enables the vehicle to successfully navigate to the goal?

On board the vehicle is assumed to be a sparse map of landmarks with corresponding image data and coordinates. Therefore, as the vehicle travels in the environment and collects camera data, it performs scene detection by comparing features in the collected images to the features in the landmark images within its database. For this study, ORB feature detection and matching is utilized to detect and describe features within the local scene of the vehicle and then match to a corresponding urban scene within the database of images on the vehicle reference.\(^{37}\) To reduce map storage size and run-time computation, ORB features are precomputed for the landmark database images and stored as the image data within the map. Therefore, only the lightweight ORB features and corresponding coordinates need to be stored within the vehicle's sparse map, rather than the entire landmark image. When a test image is found to produce ORB features that match the features of a landmark within the database, the corresponding landmark location is used as a measurement update within the extended Kalman filter (EKF) to refine the pose estimate of the vehicle. Using this definition of a landmark (ie, a set of ORB features and corresponding coordinates for a given location), then landmark density is simply defined as the number of landmarks per square kilometer.

This technique is implemented in simulation, as described in Section 3, and then in experiments, as described in Section 4. While the proposed work can utilize any scene detector or computer vision method without loss of generality, ORB feature detection was chosen over more complex approaches, such as the bag-of-words model used in FAB-MAP, due to its ease of implementation including training and tuning. Furthermore, by tracking the vehicle's position in metric coordinates in between landmark detections via dead reckoning, all landmarks beyond the 2-sigma uncertainty ellipse around the vehicle can be disregarded. This helps to reduce the run-time computation of the ORB landmark detector, as it needs only to compare the current test image to the landmarks nearby the vehicle. This approach also improved performance by eliminating many false positives for the experiments described in Section 4.

### 2.3 Intersection navigation algorithm

While a simple road network is often available for urban areas, it is not always accurate due to construction and road closings. Therefore, in an attempt to rely on a minimal amount of prior map information, this study assumes that no
information regarding the roads is known a priori. Under this assumption, navigating to a desired location in an urban environment becomes a challenging task. To make this difficult challenge feasible, certain assumptions need to be made. First, the initial location of the vehicle, as well as the goal, must be known within some defined uncertainty. Second, the vehicle is assumed to be equipped with cameras and able to detect common roadway objects and perceive the environment at the level required for the techniques described in Section 2.1. Furthermore, this detection capability would allow the vehicle to perform obstacle detection and low-level path planning. These two areas are considered outside the scope of this study, but the reader is referred to the work of Miller et al.\textsuperscript{35} for a suitable approach.

Under these specified assumptions, the low-level navigation problem (ie, stay on the road and avoid obstacles) becomes feasible, and a higher-level navigation problem can be formulated (ie, navigate from point A to point B via roads and intersections). High-level navigation to a goal location from a known starting location follows two basic principles in this framework. First, the vehicle decides which direction of travel to take at an intersection (eg, turn right, turn left, or continue straight) by minimizing a cost function, which is simply the difference in angle between the vehicle's heading and the direction to the goal. Second, the vehicle retains a list of intersections already visited along with the corresponding decisions made at each intersection. This list of intersections is generated by logging the vehicle's pose, as well as the uncertainty, each time a decision is made by the intersection navigation algorithm. Then, to determine whether an intersection was already visited, the current vehicle pose at the given intersection is compared to all the logged vehicle poses at previous intersections. If the current 2-sigma uncertainty ellipse of the vehicle pose overlaps with the 2-sigma uncertainty ellipse of the vehicle pose at a previous intersection, then that previously visited intersection is considered to be the same intersection as the current intersection. It is important to note that the locations of the previously visited intersections are based on the vehicle's pose and will therefore become less reliable as the vehicle's pose uncertainty increases. If a previously visited intersection is encountered, a penalty is applied to the previous intersection decision, thus making the vehicle less likely to perform the same action as before. This principle is similar to Tabu search, as it relaxes the basic rule of the algorithm and discourages the search to return to previously visited solutions to avoid getting stuck in suboptimal regions.\textsuperscript{39} The decision rule is general to any type of intersection (eg, different number of roads at different angles), although the focus of this study is on gridded roadways due to the intended application of dense urban areas. In a manner similar to traditional SLAM approaches, encountering previously visited locations can help to reduce pose uncertainty. However, unlike SLAM, revisiting a previously seen location is not desired for this study. In fact, it is undesirable as it indicates that a suboptimal route is being taken to the goal. Therefore, rather than increase complexity and use revisited intersections for localization (which could, at best, only reduce pose uncertainty to the level at which it was during the previous intersection visit), this study simply uses revisited intersections to better inform the intersection navigation algorithm.

The two principles for the navigation algorithm are summarized in the following optimization problem:

\[
\theta^* = \arg\min_{i \in N} \left| |\theta_i - \phi| + \gamma_i \right|, \quad (13)
\]

where \(\theta^*\) represents the intersection decision in the inertial coordinate frame, or direction of travel after the intersection, \(i\) is the index of turning options at an intersection (eg, turn right, turn left, or continue straight), and \(N\) denotes the total number of turning options at a given intersection. The direction of travel after the intersection is represented by \(\theta_i^+\). The direction from the vehicle to the goal after the intersection is given by \(\phi_i\). The penalty of making the same decision at the same intersection as before is denoted by \(\gamma_i\), which increases incrementally each time the same decision is made at the same intersection.

To clarify the navigation algorithm in Equation (13), Figure 10 portrays an example of an intersection decision. In this example, the vehicle navigates through a 3-way intersection. Based on the techniques described in Section 2.1, the vehicle can detect the type of intersection that it is approaching (eg, 3-way intersection with the option to turn right or left) using on-board sensor measurements in real time. With this information, the angle between the vehicle's heading and the direction to the goal is determined for each possible route. After applying any penalties based on the decisions made during previous visits to this particular intersection, the route associated with the minimum value is selected. In this example, assuming the intersection was not previously visited, the algorithm directs the vehicle to travel straight through the intersection.
FIGURE 10  Illustration of an autonomous vehicle navigating through a 3-way intersection using the proposed intersection navigation algorithm in Equation (13). The red star denotes the goal. A, The vehicle approaching the intersection; B, The vehicle after continuing straight through the intersection (decision $i=1$); C, The vehicle after turning right through the intersection (decision $i=2$). The navigation algorithm would choose to continue straight through the intersection in B, assuming this intersection was not previously visited.

SIMULATION RESULTS AND DISCUSSION

A set of Monte Carlo simulations were conducted to evaluate the feasibility and performance of the proposed high-level navigation and estimation approach as a function of several key parameters. The simulator models the vehicles dynamics using a four-state bicycle model\textsuperscript{40} and simulates its motion through a randomly-generated city road network. A predictive state model of the vehicle is given in Equation (14).

Equation (15) gives a more detailed representation of Equation (14) by explicitly expressing the individual vector components of the vehicle's state. In Equations (14) and (15), $\overline{x}_k$ denotes the vehicle state at time $k$, which consists of four components: the position of the center of the vehicle's rear axle ($x_k$ and $y_k$), the vehicle's heading ($\theta_k$), and the speed of the vehicle ($v_k$). The control inputs to the vehicle model are acceleration ($a_k$) and steering angle ($\phi_k$). The prediction time step is $\delta t$. In addition, the $\Delta$ terms are defined as follows:

\[
\Delta \theta = \frac{d_k}{\rho_k} \quad \text{(16)}
\]
\[
\Delta x = \rho_k \sin(\Delta \theta) \quad \text{(17)}
\]
\[
\Delta y = \rho_k (1 - \cos(\Delta \theta)) \quad \text{(18)}
\]
\[
\rho_k = \frac{l}{\phi_k} \quad \text{(19)}
\]
\[
d_k = \frac{1}{2} a_k \delta t^2 + v_k \delta t. \quad \text{(20)}
\]

Parameter $l$ represents the length between the front and rear axles, $\rho_k$ is the radius of curvature for the vehicle, and $d_k$ denotes the distance the vehicle travels over $\delta t$.

Simulated odometry, compass, and vision measurements are used to localize the vehicle as it travels. Steering and speed proportional-integral-derivative controllers are used to allow the vehicle to stay in its lane and on a specified path chosen by the navigation algorithm. The static roadway scene estimator was not implemented for this simulator, but rather, its measurements were simply assumed to be present. For every test in this Monte Carlo study, a randomly gridded map was generated. These maps ranged in size from 1 km\textsuperscript{2} to 100 km\textsuperscript{2}, and the block size ranged from 50 m to 300 m. To increase map complexity, the maps were randomly seeded with dead-end and one-way roads. Furthermore, start and end locations were selected randomly. Finally, for tests involving landmarks, the maps were randomly seeded with landmarks according to the given landmark density.
3.1 | Monte Carlo range tests without map information

Monte Carlo tests were performed to determine the maximum distance the vehicle can travel with no prior map information and without receiving any globally localizing information (ie, no GPS or landmark measurements) before it becomes lost. In other words, this is a worst-case scenario. On-board sensors are assumed to enable the vehicle to stay on the road despite the absence of prior map data. However, the limiting factor that may cause the vehicle to fail to reach the goal is the navigation algorithm's ability to differentiate between each intersection. Differentiating between intersections becomes ambiguous when the position uncertainty of the vehicle becomes large enough that it encompasses multiple intersections. The vehicle is presumed lost when the major axis of the 2-sigma uncertainty ellipse of the vehicle's x-y position exceeds 100 m, which is approximately the size of an average Manhattan city block. This study uses a 2-sigma uncertainty ellipse, rather than a 1-sigma uncertainty ellipse, to provide a high confidence region for the vehicle's position. While the pose uncertainty is below this threshold, the vehicle is considered sufficiently localized for the high-level navigation algorithm to function.

This work assumes a conservative odometry measurement uncertainty, which is common for automobiles. Both wheel slip and rotary encoder discretization errors are taken into account. These sources of uncertainty result in a maximum error of $\pm \frac{1}{2}$ of the angular rotation between two successive bits. Regarding heading measurements, the large amount of electrical hardware found on autonomous vehicles leads to high compass uncertainty. Thus, due to the difficulty of accurately modeling these interactions, this study investigates three conservative levels of compass uncertainty: 2-sigma values of $\pm 10^\circ$, $\pm 20^\circ$, and $\pm 30^\circ$. This simulated heading measurement error was modeled as Gaussian noise. Additionally, the fourth set of simulations were conducted in which heading measurements were not used to aid localization. A total of 4000 simulations were performed, with each simulation using a new random map, and new random start and end locations. Figure 11 shows the 2-sigma ellipse major axis as a function of distance traveled for each test in all four cases.

The results seen in Figure 11 follow a predictable trend for each set of tests. The major axis of the 2-sigma uncertainty ellipse increases rapidly at first, when the ellipse is small, but then its growth becomes more gradual as the vehicle travels further and the area of the ellipse becomes large. Within the scope of these simulations, a distance at which the vehicle becomes lost was not found for a 2-sigma compass uncertainty of $\pm 10^\circ$. To avoid a single test requiring substantially more time and computation than the rest of the tests, a maximum range of 50 km was set and the simulation was terminated if this range was surpassed. The vehicle was found to travel 20.8 km on average before becoming lost for a 2-sigma compass uncertainty of $\pm 20^\circ$. As for the tests using a 2-sigma compass uncertainty of $\pm 30^\circ$, the vehicle could travel 9.3 km on average before becoming lost. Finally, for the tests without the aid of heading measurements, the vehicle could only travel 300 m on average before becoming lost. In this case, only the wheel encoders are used to estimate the pose of the vehicle. Therefore, as the vehicle begins to drive, the initial uncertainty in the heading of the vehicle quickly results in a large lateral uncertainty in its pose.

A primary driver in this study, aside from compass uncertainty, is the number of turns that the vehicle made during its travel. A vehicle driving along a straight road will reach the goal with a larger uncertainty ellipse major axis than a vehicle...
traveling the same distance while taking many turns. This occurs because the ellipse grows along both axes in the case involving many turns but grows predominantly along only one axis in the case of driving straight. Therefore, since the ellipse’s growth is not shared between both axes, the major axis of the 2-sigma uncertainty ellipse grows at a faster rate for straight driving. Figure 11 shows this trend, as the majority of the data points above the fitted curve come from tests in which the vehicle took few turns, whereas the majority of the data points below the fitted curve come from tests in which the vehicle took many turns. A key take-away from this simulation study is that an autonomous vehicle can perform high-level navigation and localization for long periods of time without prior map information or GPS measurements using only an inexpensive compass and wheel encoders.

3.2 Monte Carlo landmark detection study

A Monte Carlo study was conducted to determine the extent to which sparse map-based measurements can improve localization, and, consequently, navigation and distance traveled. This study employed the same parameters as the prior Monte Carlo range tests; however, this study augmented pose estimation by incorporating a sparse map of landmarks with corresponding coordinates known a priori. For these tests, a simulated camera collects image data of its environment and then attempts to correlate the image with a sparse map of landmark images. In the case of a positive landmark detection, a map-based pose measurement update is performed using the landmark’s known location to a specified degree of uncertainty. The effects of landmark density and landmark detection recall are also studied. The simulated landmark detection, a map-based pose measurement update is performed using the landmark’s known location to a specified degree of uncertainty. The effects of landmark density and landmark detection recall are also studied. The simulated landmark detection recall is implemented by simply disregarding a specified percentage of the landmark detections, which assumes an operational design domain that excludes extreme weather, low lighting, or other situations that would make detection very difficult. A constant 2-sigma compass uncertainty of ±30° is assumed, which is the most conservative value from the Monte Carlo range study with no prior map information.

Furthermore, this study investigates three variants of the heading-based navigation function in Equation (13). For the first method, the vehicle attempts to drive straight to the goal and does not actively seek out landmarks (i.e., only fortuitous landmark measurement updates). This approach is termed the straight-to-goal navigation method. For the second method, the vehicle actively seeks out the nearest landmark that will also bring it nearer to the goal. This approach is termed the landmark-to-landmark navigation method. Finally, for the third approach, the vehicle attempts to drive straight to the goal until its pose uncertainty exceeds a specified threshold, at which point the vehicle then actively seeks out a nearby landmark to update its pose estimate. This approach is termed the hybrid navigation method. In this study, the specified threshold is 50 m for the size of the 2-sigma pose uncertainty ellipse’s major axis. Further investigation of these navigation methods could allow for an optimal navigation technique to be determined given additional environmental assumptions.

Given the five different landmark density values, five different landmark detection recall values, and three different navigation methods considered in this study, a total of 75 distinct combinations of test conditions were studied. For every combination, 700 simulation trials were performed to determine the success rate as a function of Euclidean distance from starting point to goal. Success rate is defined as the rate at which the vehicle reaches the goal before its 2-sigma uncertainty ellipse major axis exceeds the lost threshold of 100 m. The performance results from more than 50,000 Monte Carlo simulations are summarized in Figure 12. Note that the range values in Figures 11 and 12 are not directly comparable, as Figure 11 shows the Manhattan distance traveled, whereas Figure 12 shows the range as Euclidean distance. This distinction is made because different routes to the goal were taken for each different navigation method. Therefore, to show a fair comparison between navigation methods, Figure 12 shows the range as the Euclidean distance from start point to goal, rather than the total Manhattan distance traveled.

In this study, an 80% success rate of reaching the goal is examined. This success rate is chosen for easy comparison with the experimental results presented in Section 4. In Figure 12, the Euclidean range that the vehicle could reach with an 80% success rate is shown for every test condition. Intuitively, the range and overall success of the navigation method is expected to increase as both the landmark density and landmark detection recall increase. This trend is predominantly seen in the results. A clear upward trend can be seen in the data for dense landmark maps, with the trend becoming subtler as the density decreases. However, this trend is less apparent for the straight-to-goal navigation method, as the vehicle does not actively seek out landmarks but only fortuitously encounters them, and therefore receives far fewer landmark measurement updates. Note that the range values in Figure 12 never exceed 8 km. This is because the maximum distance between the starting point and goal for the maps generated in this study was limited to 8 km to prevent a single test from requiring substantially more time and computation than the rest of the tests. In all, the results from this Monte Carlo
FIGURE 12  Monte Carlo landmark detection study results for the (A) straight-to-goal navigation, (B) landmark-to-landmark navigation, and (C) hybrid navigation method. Note that this figure shows range as Euclidean distance from start point to goal, rather than Manhattan distance traveled by the vehicle.

study allow for an expected range with a given success rate to be determined for all combinations of landmark density values, landmark detection recall values, and navigation methods.

Figure 12 also gives insight to the effectiveness of each navigation method. The performance of the straight-to-goal navigation method, in which the vehicle drives straight to the goal and neglects landmarks, is notably poor compared to the alternative navigation methods. The landmark-to-landmark and hybrid navigation methods perform similarly in terms of robustness (i.e., the vehicle is able to reach the goal with similar reliability for given test conditions). The Euclidean range of the straight-to-goal navigation method is generally 1 to 2 km less than the corresponding range for the alternative navigation methods, excluding the 10 landmarks per km² case (in which many landmarks are reached regardless of the navigation method). However, the landmark-to-landmark and hybrid navigation methods, which actively seek out landmarks, also increase the average range of the vehicle. The landmark-to-landmark navigation method increases the vehicle’s range by 31% on average compared to the straight-to-goal navigation method. Likewise, the hybrid navigation method increases the vehicle’s range by 15% on average compared to the straight-to-goal navigation method. Therefore, in general, there is a tradeoff between overall distance traveled and reliability of reaching the goal for the various navigation methods examined in this study. A key take-away from this simulation study is that even just a few landmark updates, with an intelligent navigation algorithm, can greatly improve navigation duration and robustness.

4 | EXPERIMENTAL RESULTS AND DISCUSSION

Experimental studies to test the performance of the proposed methods were conducted in Ithaca, NY with Cornell University’s autonomous test vehicle, as shown in Figure 13. The first set of tests was conducted to experimentally validate the proposed static scene estimation technique. The following two sets of tests were conducted to understand the maturity of the theory for the proposed pose estimation and navigation methods, as well as to verify the simulation results. According to current New York State traffic laws, vehicles are not permitted to operate autonomously on public roads. Therefore, during this experimental study, the pose estimation and navigation techniques were used as a high-level navigation guide for a human driving the vehicle. In other words, the human driver acted as the inner-loop controller maintaining the vehicle on the road and following the high-level navigation decisions.

The vehicle’s wheel encoders provided the odometry measurements via CAN bus for this experimental study. A low-cost compass was installed in the vehicle to provide heading measurements. This instrument’s 2-sigma uncertainty value was empirically determined to be approximately ±25°. This high uncertainty value is a result of the large amount of electromagnetic interference within the vehicle. Moreover, this uncertainty is dependent on the vehicle’s location within the city and can improve or worsen based on its surroundings. For instance, the compass uncertainty worsened in construc-
tion areas due to the presence of heavy machinery and electromagnetic interference. Finally, a front-facing ZED stereo camera and a Point Grey Ladybug3 360-degree camera were used to capture visual data around the vehicle for detecting landmarks, as well as detecting visual environmental cues.

4.1 Static roadway scene estimation tests

An experimental study was conducted to evaluate the performance of the proposed static roadway scene estimator (refer to Figure 2, Figure 9, and Equation (12)). For this experimental study, 70 tests were performed in which the test vehicle approached an intersection while gathering image and odometry data and running the proposed algorithm to estimate the static roadway scene. The 70 tests included 10 tests from 7 different intersections. These field tests included both 3-way and 4-way intersections, as well as various roadway features, such as traffic lights, stop signs, and one-way signs. A summary of the 7 test intersections is shown below in Table 2. The 10 tests performed at each intersection were carried out under diverse environmental and vehicle conditions, such as lighting, weather, traffic, and vehicle speed. In addition to nominal test conditions, at least one test at each intersection was performed under each of the following circumstances: sunny (often with glare), rainy, at dawn, at dusk, heavy cross traffic, and no cross traffic. Computations for the detection and segmentation algorithms were performed on TITAN X and 1080 Ti graphics cards.

Figure 14 displays a summary of the experimental results for detecting the presence of an intersection and its possible directions of travel. In this figure, two key metrics are plotted for each trial. The first metric is the vehicle’s distance from the intersection at which the intersection is detected and its approximate relative position is estimated. An intersection is considered to be detected when its probability exceeds 0.9. These data points are shown in red. The mean distance of the vehicle from the intersection is 34.6 m. The second metric is the vehicle’s distance from the intersection at which the full intersection is detected. The full intersection is considered to be detected when the probabilities of all present intersection features (i.e., the intersection itself and all possible directions of travel) exceed 0.9. These data points are shown

| Table 2 | Descriptions of seven test intersections |
|---------|------------------------------------------|
| Name    | Description                              |
| Intersection A | 3-way intersection with stop sign and typically busy cross traffic |
| Intersection B | 3-way intersection with stop sign and vehicles typically parked along intersecting road |
| Intersection C | 4-way intersection with stop sign and vehicles typically parked along both roads |
| Intersection D | 4-way intersection with stop sign |
| Intersection E | 4-way intersection with traffic light and typically busy cross traffic |
| Intersection F | 3-way intersection with stop sign, right one-way sign, and typically busy cross traffic |
| Intersection G | 4-way intersection with stop sign and occluded views due to protruding vegetation |
in black. The mean distance of the vehicle from the intersection is 18.6 m. The data points in this figure are separated by intersection location, as indicated by the annotations in the figure. Thus, the 10 red data points and 10 black data points in each segment correspond to the same intersection although the test conditions differed, such as lighting, weather, traffic, and vehicle speed.

Figure 14 demonstrates that the proposed scene estimation method (shown in Figure 2) can successfully detect an intersection and estimate its location relative to the vehicle from approximately 35 m away on average. Detection and localization of the intersection at this distance gives the autonomous vehicle sufficient time to prepare to turn at the intersection. Note that the New York State Law mandates that a driver must turn on their turn signal and prepare to turn 30 m before entering an intersection. Figure 14 also exhibits the ability to successfully detect the full intersection from nearly 20 m away on average, a detection capability that permits the vehicle to successfully navigate the intersection. Furthermore, Figure 14 shows that the full intersection is successfully detected in all cases from at least 7 m away. Although false positive measurements of the various environmental cues did occur, they were not sufficiently prevalent to cause a false positive in the inferred intersection features (ie, the probability of an absent intersection feature never exceeded 0.9 for these tests). Although this failure was never seen in the experimental study, no guarantee can be made that this failure will never occur. Therefore, additional measures could be taken to mitigate failures as much as possible. For instance, the individual detectors used for the various environmental cues could be further improved, the operational design domain could be restricted to only include favorable conditions, and additional forms of redundancy could be added to the detection process.

Individual tests provide further details describing the capability of this technique. Figure 15A shows the results for test 1 at Intersection E (see Figure 14 for reference). In this test, the vehicle was able to successfully detect the intersection at 38.6 m away, and then detect the full intersection at 19.8 m away. Note that the left turn probability increases early due to a detection of a left-facing parked vehicle on the intersecting road. Figure 15B shows the results for test 5 at Intersection F. The intersection was detected at 41.2 m away and the full intersection was detected at 29.4 m away. It is interesting to note that the left turn probability begins to increase midway through the test, but then sharply drops when a one-way sign is detected, which had been hidden behind tree branches prior to that moment. The detection and tracking of cross traffic allows the right turn probability to begin to increase early in this test. Figure 15C shows the results for test 7 at Intersection A. In this test, the intersection was detected at 75.2 m away and the full intersection was detected at 30.1 m away. This test had clear conditions, which contributed to the high detection range for this test. Moreover, the detection of cross traffic again allowed the right turn probability to increase early in the test. Finally, Figure 15D shows the results for test 10 at Intersection B. The intersection was detected at 47.7 m away and the full intersection was detected at 17.6 m away. This test was conducted at dusk, and therefore, due to unfamiliar lighting conditions compared to the training data for this detector, road segmentation did not perform as strongly as the other cases. However, due to the strong detection of left-facing parked vehicles along the intersecting road, the left turn probability increased sharply early in the test.

Figure 16 shows the results from individual tests in which the proposed technique did not perform as well as the other cases. Figure 16A shows the results for test 3 at Intersection G. The intersection in this test was detected at 43.5 m away and the full intersection was detected at only 9.8 m away. Several trees and bushes along the right side of the road obstructed the view of the right turn option, and therefore, the right turn was detected late. Figure 16B shows the results for test 7 at Intersection E. For this test, the glare from the sun prevented the traffic light from being detected until the vehicle was nearly stopped at the intersection. Without detecting the traffic light early, the intersection probability did not exceed 0.9 until late in the test. This caused the messages being sent by the environmental cues associated with the directions of travel to remain neutral until the intersection was detected. Once the intersection was finally detected, the directions of travel could also be detected. In this test, the intersection was detected at only 8.2 m away and the full intersection was detected...
20 of 26

CHIPKA AND CAMPBELL

FIGURE 15  Data from individual tests demonstrating strong performance of the proposed technique. The 0.9 probability threshold is indicated by the dashed red line in the graphs on the left side. A, Test 1 at Intersection E; B, Test 5 at Intersection F; C, Test 7 at Intersection A; D, Test 10 at Intersection B at 7.6 m away. Figure 16C shows the results for test 10 at Intersection D. In this test, the intersection was detected at 30.1 m away and the full intersection was detected at 11.8 m away. This test was conducted at dusk and, therefore, experienced poor performance from road segmentation. This test also had limited cross traffic, oncoming/outgoing traffic, and vehicles parked along the intersecting road, which all contributed to the overall difficulty of this particular test. Overall, the vast majority of the poor performance trials seen in Figure 14 can be attributed to one of the following unfavorable conditions: obstruction of signs and roads by vegetation, missed detections of traffic lights due to sun glare (as seen in tests 7 and 8 at Intersection E in Figure 14), and missed detections of road surfaces due to poor lighting conditions at night (as seen in test 10 at Intersection D in Figure 14). The key take-away from this experimental study is that multiple environmental cues from vision can be fused probabilistically to detect intersections, as well as their key features, at an adequate proximity in a variety of challenging conditions.

4.2 Navigation range tests without map information

One series of experiments sought to validate the simulated results from the Monte Carlo range study without map information, as shown in Figure 11. During these tests, the vehicle was driven for a specified amount of time and the final position uncertainty from the pose estimator, along with the total distance traveled, was recorded. Thirty-three field tests
FIGURE 16 Data from individual tests with weaker performance. The 0.9 probability threshold is indicated by the dashed red line in the graphs on the left side. A, Test 3 at Intersection G; B, Test 7 at Intersection E; C, Test 10 at Intersection D

FIGURE 17 Experimental results from 33 field tests shown with simulated Monte Carlo results to verify the Manhattan distance an autonomous vehicle can travel without any external pose measurement updates before getting lost were conducted for this experimental study. The results from these tests are shown in Figure 17, with the Monte Carlo simulation results overlaid for comparison.

The experimental results shown in Figure 17 demonstrate that the vehicle can travel nearly 10 km without map information before becoming lost. Given that the low-cost compass used during this study had a 2-sigma uncertainty of approximately ±25°, the experimental data were expected to fall between the high compass uncertainty and medium compass uncertainty data points from the simulation study. In all, this trend is predominantly seen in Figure 17, which
demonstrates that the experimental results reinforce the simulation results. However, as the distance traveled increases, the experimental data align more closely with the high compass uncertainty data points from the Monte Carlo study. This is likely a result of locations within the city where the compass experienced abnormally high electromagnetic interference. The effect of these specific regions within the city on compass accuracy was first observed during compass calibration tests. As the distance traveled by the vehicle increases, it became more likely to drive through one of these areas with high magnetic interference, thus leading to a higher average compass uncertainty. The key take-away from this study is that the Monte Carlo simulation results were validated, indicating that long-term resilient navigation can be performed in urban areas with an inexpensive compass.

4.3 Navigation with visual landmark detection tests

A second set of experiments sought to validate the proposed high-level navigation method utilizing visual landmark detections. As the first step to validate the Monte Carlo landmark study results, an ORB feature-based landmark detector was developed and implemented on the vehicle to augment the existing pose estimator. This landmark detector had an average detection recall of approximately 60%, as evaluated by preliminary tests; it is envisioned that a higher performing visual detector could be developed, but the overall navigation results would trend in similar ways. Furthermore, 126 images from downtown Ithaca, NY intersections were obtained to populate the vehicle’s landmark database. These landmark images were chosen due to the prominent ORB features that they contained. For instance, street corners with prominent buildings often provided the most feature-rich landmark images due to the strong gradients from the building’s edges and corners. For each landmark image, the 20 most prominent 2D ORB features were used to describe the landmark. Only 20 features were utilized to reduce the computational cost of the landmark detector while also providing enough information to adequately describe the landmark. For more robust landmark detection in future studies, a more sophisticated landmark selection method could be used.\(^{43}\) Due to the specific landmark density that was chosen to be tested, as well as the small size of downtown Ithaca, only a few of these landmark images were used during the experiments. Finally, the landmark-to-landmark navigation method was used in these tests due to the increased reliability of reaching the goal, as seen in the prior Monte Carlo study.

For each trial, the vehicle was guided to a randomly chosen goal location by the navigation algorithm, which relied on the sparse pose estimator for localization. Once the goal was reached, a new random goal was spawned and the vehicle then proceeded to navigate to the new goal. This process was repeated until the vehicle could no longer reliably navigate to the goal due to an exceedingly large pose uncertainty, or the maximum experiment duration of 75 minutes was reached, at which point the experiment was terminated. Either a single landmark or no landmarks were placed within the map at a random location for each goal generated. This was done to achieve a landmark density of less than 1 landmark per km\(^2\) (since downtown Ithaca is approximately 1 km\(^2\) and a fraction of a landmark could not be placed within the map). One landmark was placed within the map with a probability of 0.6 for each goal generated. Then, when the vehicle reached its corresponding goal, the landmark was cleared from the map. Thus, no more than one landmark and one goal were in the map concurrently. This approach of traveling to many subsequent goals, as opposed to one goal, is needed to be taken due to the small size of downtown Ithaca, NY. Simulations show that the method of subsequently traveling to many short-range goals produce analogous results compared to traveling to one long-range goal.

For this experimental study, 10 tests were conducted with an average landmark density of 0.55 landmarks per km\(^2\) and an average landmark detection recall of 60%. The number of tests for this study was chosen based on the time required to perform each experiment, approximately 1 hour. In 8 of the 10 experiments, the vehicle successfully reached a final goal with a Euclidean distance of at least 6.9 km from the starting point. This result is plotted as the black dot in Figure 18, along with the Monte Carlo simulation results for comparison.

This experimental study demonstrates that a vehicle can reliably travel to a much further goal by incorporating a sparse map of landmarks. In many of the experiments, the vehicle could drive a Manhattan distance of 20 km, which is more than twice the distance traveled in the tests without map information. This study demonstrated that the vehicle could drive nearly 10 km between landmark measurement updates without its pose uncertainty becoming so large that the navigation algorithm breaks down. Thus, a key take-away from these experiments is that the proposed lightweight sparse pose estimator can sufficiently localize the vehicle to allow it to successfully reach the goal, as long as the landmark density and detection recall are high enough such that a landmark measurement update occurs at least every 10 km. This was seen in three tests, as the vehicle received many landmark measurement updates, and therefore, did not experience
any difficulty in navigating to the subsequent goal locations. These tests, however, were eventually terminated after 75 minutes.

Overall, the results from this experimental study match the simulation results relatively well, albeit some variation, therefore validating the simulation results. As shown in Figure 18, the experiments demonstrated that a range of 6.9 km could be reached with a success rate of 80%. Compared to the Monte Carlo results, the experimental range value is slightly below the expected value for the given landmark density and detection recall. This small amount of variation can, at least in part, be attributed to the small amount of field tests compared to the large number of simulation trials. It is expected that the experimental results would more closely reflect the Monte Carlo results with additional field tests. However, for this study, 10 experiments were deemed sufficient to demonstrate and validate the capabilities of the proposed techniques, given the large amount of time required to conduct each experiment. Moreover, the average landmark detection recall varied from test to test. This high uncertainty in the detection recall could also factor into the slight discrepancy between the experimental result and the corresponding simulation results, as shown in Figure 18. The detection recall was dependent on many environmental conditions including the prominence of the features at the intersection, weather, lighting, and traffic. For example, the vehicle would be less likely to receive a landmark measurement update if the landmark did not have prominent features (e.g., an open field or empty parking lot), or if the weather or traffic was very different from the database images. An example database image and test image of a landmark taken during testing is displayed in Figure 19.
A novel system architecture is presented to address the challenge of autonomous driving within an urban environment when reliable GPS measurements and map information are limited. This paper proposes a real-time static roadway scene estimation method that fuses several computer vision techniques to obtain detections of environmental cues, including traffic lights, stop signs, road surface, cross traffic, parked vehicles, oncoming/outgoing traffic, lane lines, one-way signs, and do-not-enter signs. These cues are then used to estimate the vehicle-relative location of the intersection, as well as the probabilities associated with the key intersection features needed for navigation through belief propagation. Furthermore, this paper proposes a pose estimation method that utilizes odometry, compass, and sparse map-based measurements to estimate the pose of the vehicle as it autonomously navigates urban roadways with limited map information and GPS measurements.

Static scene estimation was validated using an experimental study, which involved 70 tests at 7 different intersections under diverse conditions of lighting, weather, traffic, and vehicle speed. Results from this experimental study exhibit the capability to detect the intersection and estimate its vehicle-relative location from an average distance of about 35 m away, which gives an autonomous vehicle sufficient time to make a turn. Furthermore, the results show that the full intersection, with all key features needed for navigation, can be detected from an average distance of nearly 20 m away, and from at least 7 m away in 100% of the tests. This detection capability provides the information required to successfully navigate an intersection without GPS or prior detailed map information.

Monte Carlo simulation studies provide evidence to resolve key issues concerning navigating without GPS or detailed map information. These simulation studies demonstrate that a vehicle can travel nearly 10 km without any external pose measurement update. In addition, when a sparse landmark-based map was incorporated, these studies provided a relationship between the vehicle’s range for a given success rate and landmark density. Experimental results verify these simulation results, as they reproduce the nearly 10-km range without any external pose measurement update, and also provide an empirical range that matches the simulation results for a given success rate, navigation method, landmark detection recall, and landmark density. Overall, these studies demonstrate that an autonomous vehicle can travel for long periods of time without GPS measurements or prior map information using only local sensors and an inexpensive compass. Moreover, these studies demonstrate that, with only a few landmark updates, navigation duration and robustness improve greatly.

In general, this study addresses critical holes in the current state of autonomous driving technology, as most autonomous driving systems rely on high precision GPS signals and an HD environmental map. Obtaining an HD environmental map proves to be highly time, cost, labor, and computationally intensive, as a fleet of sensory vehicles must be deployed to acquire rich sensor data of a certain environment multiple times. After heavy postprocessing of this data, the final HD map typically requires 1 GB for every mile mapped. Despite these drawbacks, there is no alternative to HD maps for navigation in the current state of technology. In regards to localizing the vehicle without high precision GPS signals, techniques have been proposed in recent years but still rely heavily on prior detailed map information, or remain limited in terms of real-time mapping of the full roadway scene. Finally, SLAM techniques, such as FAB-MAP and SeqSLAM, can be used to alleviate the need for HD maps and GPS signals. However, these methods do not attempt to localize the vehicle in metric coordinates, and therefore require a dense map of images to adequately localize the vehicle. This dense map of images results in rapidly growing computational demands and diminishing performance as the environment increases in size. Overall, this study addresses the previous issues by demonstrating the capability to perform autonomous navigation, localization, and real-time scene estimation with limited a priori information, capabilities for which other existing methods must rely heavily on GPS measurements and a prior HD environmental map.

ACKNOWLEDGMENTS
This work was supported by the NSF Graduate Research Fellowship Program (GRFP) Grants DGE1144153 and NSF IIS-1724282.

DISCLOSURE STATEMENT
The authors have no conflict of interest relevant to this article.
REFERENCES

1. MacGougan G, Lachapelle G, Klukas R, et al. Performance analysis of a stand-alone high-sensitivity receiver. GPS Solutions. 2002;6(3):179-195.

2. Shen M, Sun J, Peng H, Zhao D. Improving localization accuracy in connected vehicle networks using Rao-Blackwellized particle filters: theory, simulations, and experiments. IEEE Trans Intell Transp Syst. 2018;20:2255-2266.

3. Rodriguez I, Nguyen HC, Sorensen TB, et al. Analysis of 38 GHz mmWave propagation characteristics of urban scenarios. In: Proceedings of European Wireless 2015; 21th European Wireless Conference; 2015; Budapest, Hungary.

4. Petit J, Shladover SE. Potential cyberattacks on automated vehicles. IEEE Trans Intell Transp Syst. 2015;16(2):546-556.

5. Litman T. Autonomous Vehicle Implementation Predictions. Victoria, Canada: Victoria Transport Policy Institute; 2017.

6. Wheeler MD, Wu XJ, Luo W, investors; Google Patents. High definition map updates based on sensor data collected by autonomous vehicles. US patent 15/859,011. 2018.

7. Levinson J, Thrun S. Robust vehicle localization in urban environments using probabilistic maps. Paper presented at: 2010 IEEE International Conference on Robotics and Automation; 2010; Anchorage, AK.

8. Moreno F-A, Gonzalez-Jimenez J, Blanco J-L, Esteban A. An instrumented vehicle for efficient and accurate 3D mapping of roads. Comput Aided Civ Infrastruct Eng. 2013;28(6):403-419.

9. Mutz F, Veronese LP, Oliveira-Santos T, de Aguiar E, Cheein FAA, De Souza AF. Large-scale mapping in complex field scenarios using an autonomous car. Expert Syst Appl. 2016;46:439-462.

10. O’Neill-Dunne J. Image boost: drones produce sharper images for mapping. Roads & Bridges. 2015;53(9).

11. Trehard G, Pollard E, Bradai B, Nashashibi F. On line mapping and global positioning for autonomous driving in urban environment based on evidential slam. Paper presented at: 2015 IEEE Intelligent Vehicles Symposium (IV); 2015; Seoul, South Korea.

12. Carlson J. Mapping Large, Urban Environments With GPS-Aided SLAM [dissertation]. Pittsburgh, PA: The Robotics Institute Carnegie Mellon University; 2010.

13. Kent L. HERE introduces HD maps for highly automated vehicle testing. 2015.

14. Seif HG, Hu X. Autonomous driving in the iCity—HD maps as a key challenge of the automotive industry. Engineering. 2016;2(2):159-162.

15. Durrant-Whyte H, Bailey T. Simultaneous localization and mapping: part i. IEEE Robot Autom Mag. 2006;13(2):99-110.

16. Bailey T, Durrant-Whyte H. Simultaneous localization and mapping (slam): Part ii. IEEE Robot Autom Mag. 2006;13(3):108-117.

17. Cummins M, Newman P. FAB-MAP: probabilistic localization and mapping in the space of appearance. Int J Robot Res. 2008;27(6):647-665.

18. Milford MJ, Wyeth GF. SeqSLAM: visual route-based navigation for sunny summer days and stormy winter nights. Paper presented at: 2012 IEEE International Conference on Robotics and Automation; 2012; Saint Paul, MN.

19. Cummins M, Newman P. Appearance-only SLAM at large scale with FAB-MAP 2.0. Int J Robot Res. 2011;30(9):1100-1123.

20. Vivacqua RPD, Bertozzi M, Cerri P, Martins FN, Vassalto RF. Self-localization based on visual lane marking maps: an accurate low-cost approach for autonomous driving. IEEE Trans Intell Transp Syst. 2018;19(2):582-597.

21. Levinson J, Montemerlo M, Thrun S. Map-based precision vehicle localization in urban environments. Paper presented at: Robotics: Science and Systems; 2007; Corvallis, OR.

22. Kim H, Liu B, Goh CY, Lee S, Myung H. Robust vehicle localization using entropy-weighted particle filter-based data fusion of vertical and road intensity information for a large scale urban area. IEEE Robot Autom Lett. 2017;2(3):1518-1524.

23. Ziegler J, Lategahn H, Schreiber M, et al. Video based localization for bertha. Paper presented at: 2014 IEEE Intelligent Vehicles Symposium Proceedings; 2014; Dearborn, MI.

24. Paton M, Pomerleau F, MacTavish K, Ostafew CJ, Barfoot TD. Expanding the limits of vision-based localization for long-term route-following autonomy. J Field Robot. 2017;34(1):98-122.

25. Miller I, Campbell M. Particle filtering for map-aided localization in sparse GPS environments. Paper presented at: 2008 IEEE International Conference on Robotics and Automation; 2008; Pasadena, CA.

26. Fairfield N, Urmson C. Traffic light mapping and detection. Paper presented at: 2011 IEEE International Conference on Robotics and Automation; 2011; Shanghai, China.

27. Maturana D, Chou P-W, Uenoyama M, Scherer S. Real-time semantic mapping for autonomous off-road navigation. In: Field and Service Robotics. Cham, Switzerland: Springer; 2018.

28. Teichmann M, Weber M, Zoellner M, Cipolla R, Urtasun R. MultiNet: real-time joint semantic reasoning for autonomous driving. Paper presented at: 2018 IEEE Intelligent Vehicles Symposium (IV); 2016; Changshu, China.

29. Brubaker MA, Geiger A, Urtasun R. Map-based probabilistic visual self-localization. IEEE Trans Pattern Anal Mach Intell. 2015;38(4):652-665.

30. Chipka JB, Campbell M. Autonomous urban localization and navigation with limited information. Paper presented at: 2018 IEEE Intelligent Vehicles Symposium (IV); 2018; Changshu, China.

31. He K, Gkioxari G, Dollár P, Girshick R. Mask r-cnn. Paper presented at: 2017 IEEE International Conference on Computer Vision (ICCV); 2017; Venice, Italy.

32. Girshick R, Radosavovic I, Gkioxari G, Dollár P, He K. Detectron. 2018.
33. Lin T-Y, Maire M, Belongie S, et al. Microsoft COCO: common objects in context. Paper presented at: European Conference on Computer Vision; 2014; Zurich, Switzerland.

34. Ma F, Cavalheiro GV, Karaman S. Self-supervised sparse-to-dense: self-supervised depth completion from LiDAR and monocular camera. Paper presented at: 2019 International Conference on Robotics and Automation (ICRA); 2018; Montreal, QC.

35. Miller I, Campbell M, Huttenlocher D. Efficient unbiased tracking of multiple dynamic obstacles under large viewpoint changes. IEEE Trans Robot. 2011;27(1):29-46.

36. Mobileye. Applications. 2018.

37. Rublee E, Rabaud V, Konolige K, Bradski GR. ORB: an efficient alternative to sift or surf. Paper presented at: 2011 IEEE International Conference on Computer Vision (ICCV); 2011; Barcelona, Spain.

38. Lowe DG. Distinctive image features from scale-invariant keypoints. Int J Comput Vis. 2004;60(2):91-110.

39. Glover F. Tabu search—part i. ORSA J Comput. 1989;1(3):190-206.

40. Schubert R, Adam C, Obst M, Mattern N, Leonhardt V, Wanielik G. Empirical evaluation of vehicular models for ego motion estimation. Paper presented at: 2011 IEEE Intelligent Vehicles Symposium (IV); 2011; Baden-Baden, Germany.

41. DRC. Techniques for Digitizing Rotary and Linear Motion. Wilmington, MA: Dynamic Research Corp; 1976.

42. New York State Law. Vehicle and traffic law. Title VII, Article 28, Section 1163b. 2018.

43. Mühlfellner P, Bürki M, Bosse M, Derendarz W, Philippson R, Furgale P. Summary maps for lifelong visual localization. J Field Robot. 2016;33(5):561-590.

How to cite this article: Chipka JB, Campbell M. Estimation and navigation methods with limited information for autonomous urban driving. Engineering Reports. 2019;1:e12054. https://doi.org/10.1002/eng2.12054