

Visual Localization within LIDAR Maps for Automated Urban Driving

Ryan W. Wolcott and Ryan M. Eustice

Abstract— This paper reports on the problem of map-based visual localization in urban environments for autonomous vehicles. Self-driving cars have become a reality on roadways and are going to be a consumer product in the near future. One of the most significant road-blocks to autonomous vehicles is the prohibitive cost of the sensor suites necessary for localization. The most common sensor on these platforms, a three-dimensional (3D) light detection and ranging (LIDAR) scanner, generates dense point clouds with measures of surface reflectivity—which other state-of-the-art localization methods have shown are capable of centimeter-level accuracy. Alternatively, we seek to obtain comparable localization accuracy with significantly cheaper, commodity cameras. We propose to localize a single monocular camera within a 3D prior ground-map, generated by a survey vehicle equipped with 3D LIDAR scanners. To do so, we exploit a graphics processing unit to generate several synthetic views of our belief environment. We then seek to maximize the normalized mutual information between our real camera measurements and these synthetic views. Results are shown for two different datasets, a 3.0 km and a 1.5 km trajectory, where we also compare against the state-of-the-art in LIDAR map-based localization.

I. INTRODUCTION

Over the past several years, fully autonomous, self-driving cars have grown into a reality with progress in the simultaneous localization and mapping (SLAM) research community and the advent of consumer-grade three-dimensional (3D) light detection and ranging (LIDAR) scanners. Systems such as the Google driverless car use these LIDAR scanners, combined with high accuracy GPS/INS systems, to enable cars to drive hundreds of thousands of miles without user control [1].

In order to navigate autonomously, these robots require precise localization within an *a priori* known map. Rather than using the vehicle’s sensors to explicitly perceive lane markings, traffic signs, etc., metadata is embedded into a prior map, which transforms the difficult perception task into a localization problem. State-of-the-art methods [2], [3] use reflectivity measurements from 3D LIDAR scanners to create an orthographic map of ground-plane reflectivities. Online localization is then performed with the current 3D LIDAR scans and an inertial measurement unit (IMU).

The cost of 3D LIDAR scanners is prohibitive for consumer grade automobiles. Quite likely the greatest near-term

*This work was supported by a grant from Ford Motor Company via the Ford-UM Alliance under award N015392; R. Wolcott was supported by The SMART Scholarship for Service Program by the Department of Defense.

R. Wolcott is with the Computer Science and Engineering Division, University of Michigan, Ann Arbor, MI 48109, USA
 rwolcott@umich.edu.

R. Eustice is with the Department of Naval Architecture & Marine Engineering, University of Michigan, Ann Arbor, MI 48109, USA
 eustice@umich.edu.

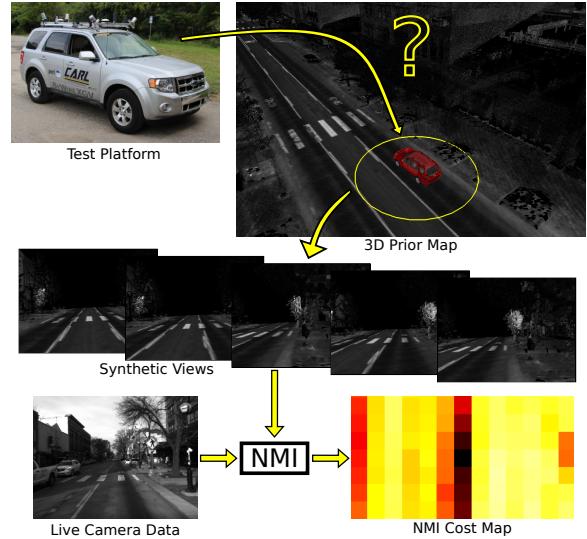


Fig. 1: Overview of our proposed visual localization system. We seek to localize a monocular camera within a 3D prior map (augmented with surface reflectivities) constructed from 3D LIDAR scanners. Given an initial pose belief, we generate numerous synthetic views of the environment, which we then evaluate using normalized mutual information (NMI) against our live view from camera imagery.

enabler for self-driving cars is the increased use of camera systems in place of expensive LIDAR scanners. Cameras provide a low-cost means to generate extremely rich, dense data that is suitable for localization.

Our approach leverages a graphics processing unit (GPU) so that we can generate several synthetic, pin-hole camera images, which we can then *directly* compare against streaming vehicle imagery. This differs from other visual localization approaches, like [4], which rely on sophisticated feature sets. This significantly simpler approach avoids over-engineering the problem by formulating a slightly more computationally expensive solution that is still real-time tractable on a mobile-grade GPU and capable of high accuracy localization.

In this paper, we propose exploiting 3D prior maps (augmented with surface reflectivities) constructed by a survey vehicle equipped with 3D LIDAR scanners. We localize a vehicle by comparing imagery from a monocular camera against several candidate views, seeking to maximize normalized mutual information (NMI) (as outlined in Fig. 1). The key contributions of our paper are:

- We present a multi-modal approach that allows us to use LIDAR-based ground maps, which accurately depicts the metric and surface reflectivity of the ground.
- We demonstrate that our projective framework can pre-

dict and evaluate appearance with a single, monocular camera.

- We benchmark our visual localization method with state-of-the-art LIDAR-based localization strategies.
- We show a GPU implementation that can provide real-time localization at ~ 10 Hz.

II. RELATED WORK

Early visual SLAM methodologies employ filtering frameworks in either an extended Kalman filter (EKF) [5] or Fast-SLAM framework [6], to generate a probability distribution over the belief pose and map of point features. In order to accurately localize within these point feature maps, one relies on co-observing these features. However, these features frequently vary with time of day and weather conditions, as noted in [7], and cannot be used without an intricate observability model [8].

In the context of autonomous vehicles, Wu and Ranganathan [4], [9] try to circumvent this by identifying and extracting higher fidelity features from road markings in images that are far more robust and representative of static infrastructure. Their method is able to densely and compactly represent a map by using a sparse collection of features for localization. However, their method assumes a flat ground, whereas our projective registration allows for more complex ground geometries and vertical structures.

Rather than relying on specific image *features* in our prior map (and complicated, hand-tuned feature extractors), our method is motivated by the desire to circumvent point features *entirely* and do whole image registration onto a static, 3D map captured by survey vehicles.

In work by Stewart and Newman [10], the use of a 3D map for featureless camera-based localization that exploits the 3D structure of the environment was explored. They were able to localize a monocular camera by minimizing normalized information distance between the appearance of 3D LIDAR points projected into multiple camera views. Further, McManus et al. [11] used a similar 3D map with reflectivity information to generate synthetic views for visual distraction suppression.

This approach has been previously considered, but methods thus far rely on the reconstruction of the local ground plane from a stereo camera pair. Senlet and Elgammal [12] create a local top-view image from a stereo pair and use chamfer matching to align their reconstruction to publicly available satellite imagery. Similarly, Napier and Newman [7] use mutual information to align a live camera stream to pre-mapped local orthographic images generated from the same stereo camera. With both of these methods, small errors in stereo pair matching can lead to oddly distorted orthographic reconstructions, thus confusing the localization pipeline. Further, our multi-modal approach allows us to take advantage of LIDAR scanners to actively capture the true reflectivity of our map, meaning our prior map is not susceptible to time of day changes in lighting and shadows.

The use of mutual information for multi-modal image registration has been widely used in the medical imaging

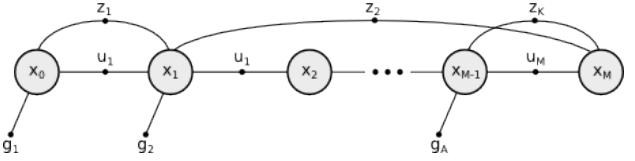


Fig. 2: Factor graph of the pose-graph SLAM problem that we solve in the off-line mapping stage. Here, x_i represents states of the robot, u_m represents incremental odometry measurements, z_k represents laser scan-matching constraints, and g_a are GPS prior measurements.

domain for several decades [13], [14]. More recently, the idea has been transferred to robotics for calibration of visual cameras to LIDAR scanners [15], [16]. This sensor registration has mostly been considered an offline task due to the expense of generating synthetic views for calibration.

To move this into real-time localization, we propose using a GPU to generate synthetic views, which we can then use a normalized measure of mutual information to optimize over our vehicle's pose. The GPU has been frequently used in robot localization for precisely this reason, including: Kinect depth-SLAM [17], image feature correspondence search for SIFT features [18], and line features [19].

III. PRIOR MAP

The first part of our localization framework is the offline mapping stage, which generates the map to be used for online localization. Our goal here is to generate a map that is metrically accurate to the surrounding structure. Prior to the offline mapping stage, our survey vehicle has no *a priori* knowledge of the environment, thus, we employ SLAM to build a model of the environment.

We use the state-of-the-art in nonlinear least-squares, pose-graph SLAM and measurements from our survey vehicle's 3D LIDAR scanners to produce a map of the 3D structure in a self-consistent frame. We construct a pose-graph to solve the full SLAM problem, as shown in Fig. 2, where nodes in the graph are poses (X) and edges are either odometry constraints (U), laser scan-matching constraints (Z), or GPS prior constraints (G). These constraints are modeled as Gaussian random variables; resulting in a nonlinear least-squares optimization problem that we solve with the incremental smoothing and mapping (iSAM) algorithm [20].

Since map construction is an offline task, we do not have to construct our pose-graph temporally. Instead, we first construct a graph with only odometry and global positioning system (GPS) prior constraints. With this skeleton pose-graph in the near vicinity of the global optimum, we use Segal et al.'s generalized iterative closest point (GICP) [21] to establish 6-degree of freedom (DOF) laser scan-matching constraints between poses; adding both odometry constraints (temporally neighboring poses) and loop closure constraints (spatially neighboring poses) to our pose-graph. Moreover, we augment our GPS prior constraints with an artificial height prior ($z = 0$) to produce a near-planar graph. Constraining the graph to a plane simplifies localization to a 3-DOF search over x , y , and θ .

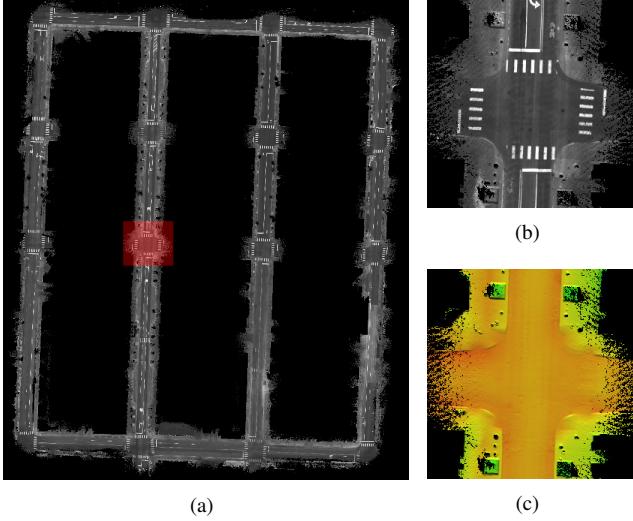


Fig. 3: Sample ground mesh used to generate synthetic views of the environment. In (a), we show a $400 \text{ m} \times 300 \text{ m}$ ground mesh colored by surface reflectivity, with a zoomed in view shown in (b) (this region is highlighted in red in (a)). We show the same zoomed view, colored by z -height to demonstrate the height variation we are able to capture with our ground-mesh in (c); yellow-to-red represents $\Delta z = 30 \text{ cm}$.

Algorithm 1 Pose-Graph to Ground-Mesh

Input: Optimized pose-graph, $G = \{x_0, x_1, \dots, x_{M-1}, x_M\}$
Output: Triangle ground-mesh, $T = \{t_0, \dots, t_N\}$

- 1: Initialize 10 cm sparse grid, $grid$
- 2: **for** x_i in G **do**
- 3: // Extract ground point cloud (p_j and r_j correspond to
- 4: // metric location and reflectivity, respectively)
- 5: $\{\{p_0, \dots, p_n\} \{r_0, \dots, r_n\}\} = \text{ExtractGround}(x_i)$
- 6:
- 7: // Drop extracted ground points into surface grid
- 8: **for** $j = 0 \rightarrow n$ **do**
- 9: Add $\{p_j, r_j\}$ to running mean at $grid[p_j]$
- 10: **end for**
- 11: **end for**
- 12: Spatially connect $grid$ to form 10 cm triangle mesh, T

From the optimized pose-graph, we construct a dense ground-plane mesh using Algorithm 1. Our algorithm is logically equivalent to extracting the ground-plane at each pose and draping an orthographic texture over a varying z -height map. A sample prior map can be seen in Fig. 3.

Note that our system is not limited to ground-only maps. We originally intended to incorporate the full 3D structure in our prior map, including buildings, street poles, etc., but found that the added structure did not appreciably increase registration quality enough to warrant the additional rendering cost (the 3D structure *doubled* scene prediction time). However, we did find that it was extremely important to use a mesh-surface as opposed to a strict planar texture because the planar texture did not accurately depict the curvature of the road (e.g., gutters sunken), as can be seen in the map colored by z -height in Fig. 3(c).

IV. PROJECTIVE IMAGE REGISTRATION

The goal of our image registration problem is to, given some initial pose prior x_k , find some relative offset Δx_i that optimally aligns the projected map, P_i , against our camera measurements, C_k . This optimization is framed as a local search problem within the vicinity of x_k and could be done in a brute-force manner by generating a predicted view for the entire $\text{dom}(x) \times \text{dom}(y) \times \text{dom}(\theta)$ search volume to avoid local maxima of hill-climbing searches. The remainder of this section details our method for generating these predicted views (P_i) and our NMI evaluation metric.

A. Generating Predicted Views

Given a query camera pose parameterized as $[R|t]$, where R and t are the camera's rotation and translation, respectively, our goal is to provide a synthetic view of our world from that vantage point. We use OpenGL, which is commonly used for visualization utilities, in a robotics context to simulate a pin-hole camera model, similar to [17].

All of our ground-mesh triangles are drawn in a world frame using indexed vertex buffer objects. These triangles are incrementally passed to the GPU as necessary as the robot traverses the environment—though the maps in our test set can easily fit within GPU memory. We pass the projection matrix,

$$P = M \cdot K \cdot \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}, \quad (1)$$

to our OpenGL Shading Language (GLSL) vertex shader for transforming world vertex coordinates to frame coordinates. Here,

$$M = \begin{bmatrix} \frac{2}{w} & 0 & 0 & -1 \\ 0 & -\frac{2}{h} & 0 & 1 \\ 0 & 0 & -\frac{2}{z_f - z_n} & -\frac{z_f + z_n}{z_f - z_n} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

and

$$K = \begin{bmatrix} f_x & \alpha & -c_x & 0 \\ 0 & f_y & -c_y & 0 \\ 0 & 0 & z_n + z_f & z_n \times z_f \\ 0 & 0 & -1 & 0 \end{bmatrix}, \quad (3)$$

where w and h are the image's width and height, z_n and z_f are the near and far clipping planes, and the elements of K correspond to the standard pinhole camera model. Note that the negative values in K 's third column are the result of inverting the z -axis to ensure proper OpenGL clipping.

For efficient handling of these generated textures, we render to an offscreen framebuffer that we then directly transfer into a CUDA buffer for processing using the CUDA-OpenGL Interoperability. Sample synthetic views can be seen in Fig. 4.

B. Simplified Rotational Search

A naïve approach to this local search problem would be to use the OpenGL pipeline to generate a synthetic view for each discrete step within the search volume, $\text{dom}(x) \times \text{dom}(y) \times \text{dom}(\theta)$. However, this would result in generating $n_x \times n_y \times n_\theta$ synthetic views. Because the

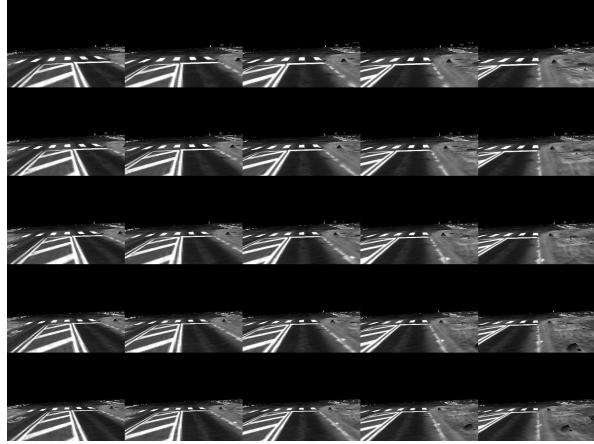


Fig. 4: Sample synthetic views generated by our OpenGL pipeline. These views were generated by varying longitudinal and lateral translation around the optimally aligned image (*center*).

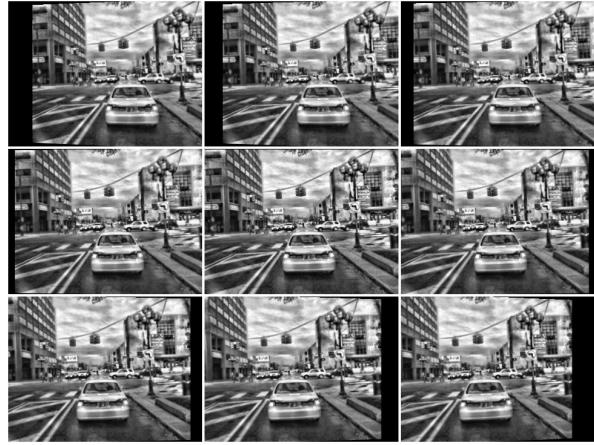


Fig. 5: Sample pre-warping applied to images to reduce the overall search space for image registration; pictured here are warps of $\theta = \{-6^\circ, -4.5^\circ, -3^\circ, -1.5^\circ, 0^\circ, 1.5^\circ, 3^\circ, 4.5^\circ, 6^\circ\}$. By rotating each *source* image in place, we can optimally pull out as much information from a single OpenGL rendered image.

predicted view rasterization is the primary bottleneck of the system (taking nearly 1 ms for each render), here we propose an alternative method of pre-warping the camera measurement to explore the θ space (warpings can be performed at 0.1 ms instead and can be parallelized with the serial OpenGL rasterizations).

We can leverage the infinite homography, $H_\infty = KRK^{-1}$, and apply a bank of precomputed rotational mappings to the source image,

$$\underline{u}^i' = KRK^{-1}\underline{u}^i. \quad (4)$$

This technique allows us to use the OpenGL pipeline to generate only $n_x \times n_y$ synthetic views, *first*, then compare each against n_θ (warped) measurements. We still evaluate the same number of candidate pairs, though we significantly reduce our OpenGL pipeline overhead. A sample of these rotations can be seen in Fig. 5.

C. Normalized Mutual Information Image Registration

Mutual information has been successfully used in various fields for registering data from multi-modal sources. Mutual

information provides a way to statistically measure the mutual dependence between two random variables, A and B . Most commonly, mutual information is defined in terms of the marginal and joint entropies of each:

$$MI(A, B) = H(A) + H(B) - H(A, B), \quad (5)$$

where these entropies can be realized by evaluating the Shannon entropy over the random variables A and B :

$$H(A) = - \sum_{a \in A} p(a) \log p(a), \quad (6)$$

$$H(B) = - \sum_{b \in B} p(b) \log p(b), \quad (7)$$

$$H(A, B) = - \sum_{a \in A} \sum_{b \in B} p(a, b) \log p(a, b). \quad (8)$$

This mutual information formulation clearly demonstrates that maximization of mutual information is achieved through the minimization of the joint entropy of A and B . This optimality coincides with minimizing the dispersion of the two random variable's joint histogram.

By viewing the problem in this information theoretic way, we are able to capture more interdependency between random variables than with simple similarity or correlation-based measures. For example, tar strips in the road frequently appear dark in LIDAR reflectivity, yet bright in visual imagery. Correlative methods can only measure either a negative or positive correlation and often fails under varying illumination. However, because maximization of mutual information is concerned with seeking tightly compact joint distributions, we can successfully capture this mutual dependence (see Fig. 6(b)). Note that it would be quite difficult to create a hand-tuned feature detector that could identify this type of information for localization.

Because our source imagery and predicted views have varying amount of overlap (largely due to our pre-warping technique), we instead employ a normalized mutual information measure. The amount of overlap between two candidate images can bias the standard mutual information measure toward *lower* overlap image pairs [22]. To avoid these effects, Studholme et al. proposed an overlap invariant measure of mutual information, normalized mutual information (NMI):

$$NMI(A, B) = \frac{H(A) + H(B)}{H(A, B)}. \quad (9)$$

This measure shares the same desirable qualities of the typical mutual information shown in (5), but is more robust to overlap changes.

In summary, our image registration amounts to the following optimization:

$$\left(\hat{x}_k, \hat{y}_k, \hat{\theta}_i \right) = \underset{(x_k, y_k, \theta_i)}{\operatorname{argmax}} NMI(C_i, P_k), \quad (10)$$

where θ_i spans the pre-warping of source imagery, C_i , and (x_k, y_k) explores the local search around our prior belief by generating synthetic views, P_k .

V. FILTERING FRAMEWORK

Our image registration is fairly agnostic to filtering framework, so here we briefly present an EKF localization framework. Due to the near-planar surface model, we are able to treat localization as a 3-DOF optimization, with the state vector $\mu_k = \{x_k, y_k, \theta_k\}$.

We define a discrete time process model and incorporate *only* image registration corrections into our state filter.

$$\text{Predict} \quad \bar{\mu}_k = F_{k-1}\mu_{k-1} \\ \bar{\Sigma}_k = F_{k-1}\Sigma_{k-1}F_{k-1}^\top + Q_{k-1}$$

$$\text{Update} \quad K_k = \bar{\Sigma}_k H_k^\top (H_k \bar{\Sigma}_k H_k^\top + R_k)^{-1} \\ \mu_k = \bar{\mu}_k + K_k (z_k - h_k(\bar{\mu}_k)) \\ \Sigma_k = (I - K_k H_k) \bar{\Sigma}_k (I - K_k H_k)^\top + K_k R_k K_k^\top$$

Here, F_{k-1} represents our plant model that integrates measurements from an Applanix IMU with uncertainty Q_{k-1} , H_k is a linear observation model (identity matrix), and K_k is the corrective Kalman gain induced by our image registration measurement z_k (with uncertainty R_k). The measurement z_k is exactly the output of our image registration in (10) and R_k is estimated by fitting a covariance to the explored cost surface, as is done in [23].

Our filter is initialized in a global frame from a single dual-antenna GPS measurement with high uncertainty, which provides a rough initial guess of global pose with orientation. We adaptively update our search bounds to ensure that we explore a 3σ window around our posterior distribution. This dynamic approach allows us to perform an expensive, exhaustive search to initially align to our prior map while avoiding local maxima, then iteratively reduce the search space as our posterior confidence increases. We restrict the finest search resolution to be 20 cm over ± 1 m. Note that aside from using GPS for initializing the filter, this proposed localization method *only* uses input from inertial sensors, a wheel encoder, and a monocular camera.

VI. RESULTS

We evaluated our theory through data collected on our autonomous platform, a TORC ByWire XGV, as seen in Fig. 1. This automated vehicle is equipped with four Velodyne HDL-32E 3D LIDAR scanners, a single Point Grey Flea3 monocular camera, and an Applanix POS-LV 420 inertial navigation system (INS).

Algorithms were implemented using OpenCV [24], OpenGL, and CUDA and all experiments were run on a laptop equipped with a Core i7-3820QM central processing unit (CPU) and mid-range mobile GPU (NVIDIA Quadro K2000M).

In collecting each dataset, we made two passes through the same environment (on separate days) and aligned the two together using our offline SLAM procedure outlined in §III. This allowed us to build a prior map ground-mesh on the first pass through the environment. Then, the subsequent pass would be well localized with respect to the ground-mesh, providing sufficiently accurate ground-truth in the experiment (accuracy an order of magnitude greater than

our localization errors). Experiments are presented on two primary datasets:

- *Downtown*: 3.0 km trajectory through downtown Ann Arbor, Michigan in which multiple roads are traversed from both directions and the dataset contains several dynamic obstacles.
- *Stadium*: 1.5 km trajectory around Michigan Stadium in Ann Arbor, Michigan. This dataset presents a complicated environment for localization as half of the dataset is through a parking lot with infrequent lane markings.

A. Image Registration

Since our odometry source has significantly low drift-rates, image registration deficiencies can be masked by a well-tuned filtering framework. Thus, we first look directly at the unfiltered image registration within the vicinity of ground-truth results.

To evaluate our image registration alone, we took our ground truth pose belief over the *Downtown* dataset and tried to perform an image registration to our map once a second. Ideally, we should be able to perfectly register our prior map, however, due to noise or insufficient visual variety in the environment, we end up with a distribution of lateral and longitudinal errors.

We present these results in two ways. First, we show our vehicle’s trajectory through the prior map in which we color our longitudinal and lateral errors at each ground-truth pose, shown in Fig. 8. In this figure, larger and brighter markers indicate a larger error in registration at that point. One can immediately notice that we are not perfectly aligned longitudinally on long, straight stretches; during these stretches, the system frequently relies on a double, solid lane marking to localize off of. To maintain accuracy, the system requires occasional cross-streets, which provide more signal for constraining our pose belief.

Second, we show the same results in histogram form, as can be seen in Fig. 9, where we see that our registration is primarily concentrated within ± 30 cm of our ground-truth. A common mode can be found in the tails of the histograms. This is caused by areas that are visually feature poor or obstructed by significant obstacles; for example, lane markings can often be perceived by the survey vehicle’s LIDAR scanners and captured in our prior map, yet the subtle transition between pavement and faded lane markings cannot be observed by our camera. In these scenarios, the optimal normalized mutual information will try to pull the registration toward the edges of our prior map—the edges are often feature poor as well, and this alignment minimizes the joint entropy of the two signals.

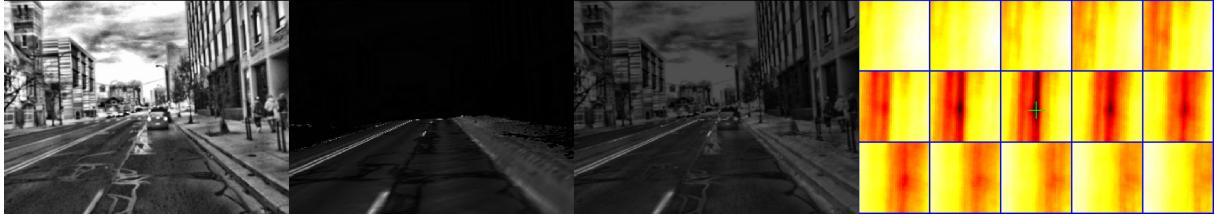
Finally, we present several scenarios of our image registration succeeding (Fig. 6) and common causes of failure (Fig. 7). These figures were generated by exploring within a local window around known ground truth.

B. Filtered Localization

We next looked at the filtered response of our system that incorporates the projective image registration into an



(a) Typical observation, unaffected by dynamic obstacle



(b) Negative correlation captured (bright tar strips in imagery aligns with dark in prior map)



(c) Our method demonstrates robustness to shadows

Fig. 6: Successful image registrations. From left to right, we show the source image, the best predicted image by our image registration, an alpha-blending of the source and predicted image, and the normalized mutual information cost map. Each cell in the cost map represents a different θ -slice of the 3D cost surface (the maxima is marked with a green '+').



(a) Our method is not robust to all dynamic obstacles, as shown in this figure

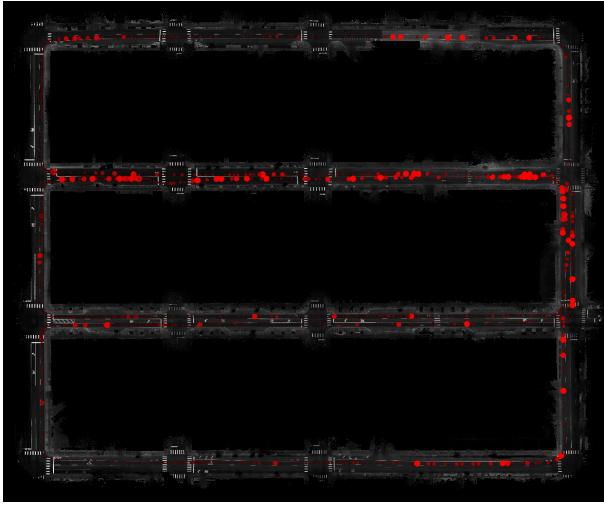


(b) Poor imagery relative to prior map prevents registration

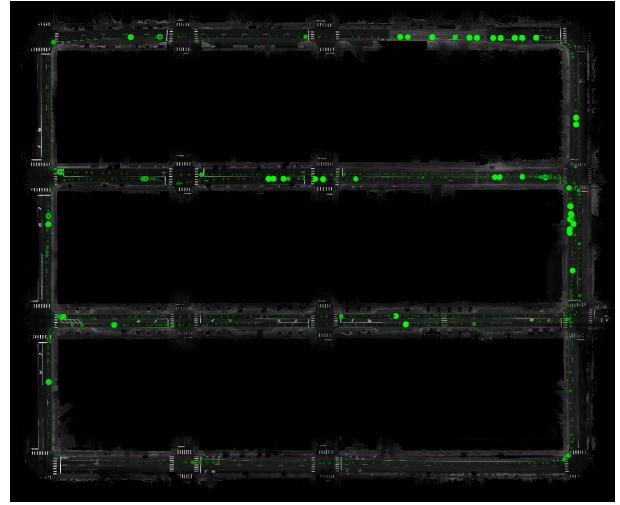


(c) Only constrained laterally by double lane marker

Fig. 7: Failure modes of our image registration (see Fig. 6 for image descriptions).



(a) Longitudinal Errors



(b) Lateral Errors

Fig. 8: Longitudinal and lateral errors in our image registration, sampled at each second of our trajectory; larger and brighter markers indicate regions where image registration produced higher errors longitudinally (a) or laterally (b). In (a), we see that, especially on the third street from the bottom, we are only well constrained longitudinally in and around intersections; quite often, we are only constrained laterally due to a double, solid lane divider being the only feature in view. In (b), we see that our method provides good lateral registration—the few bright spots seen are when the vehicle is stopped at an intersection with a vehicle obstructing our view. Note that in these two figures, perfect longitudinal or lateral registration is indicated by *dark red* or *green*, respectively.

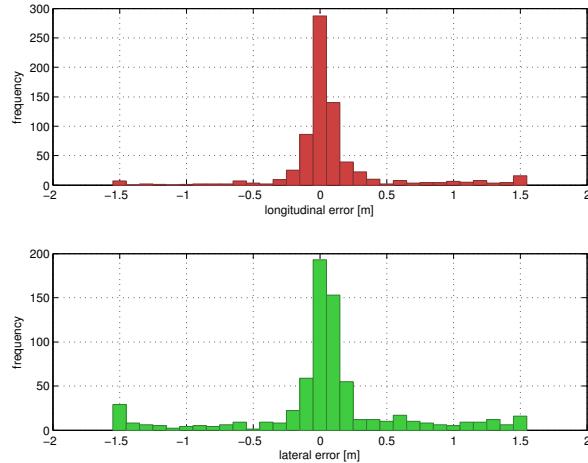


Fig. 9: Histograms of longitudinal and lateral error that our projective image registration produces (i.e., we are not using our localization filter to generate this). Most frequently, our proposed method is able to stay within ± 30 cm of ground truth. Large frequency in the tails of the histograms are caused by feature poor regions where the NMI biases towards the edges of the search limits (which was ± 1.5 m here).

EKF localization framework (see also the accompanying video attachment). Moreover, we compare our localization performance against our own implementation of the state-of-the-art LIDAR-based localization proposed by Levinson et al. in [2], [3]. Our LIDAR-based localizer builds orthographic ground images using the four Velodyne HDL-32E's onboard; these orthographic ground images can then be aligned to an orthographic prior map built using an accumulation of these scans.

We present longitudinal and lateral errors over time for GPS, LIDAR-based localization, and our proposed single

camera algorithm within the *Downtown* and *Stadium* datasets (see Fig. 10). Our proposed solution is able to maintain error levels at a similar order of magnitude as the LIDAR-based options, while using a sensor that is several orders of magnitude cheaper.

Note that the *Stadium* results show a rather large variance in longitudinal error; this is because half of the dataset is through a parking lot containing little visual variation. Also, we are slow to initially converge longitudinally because the first 20 s of the run is on a two-lane road containing only a double, solid lane marker.

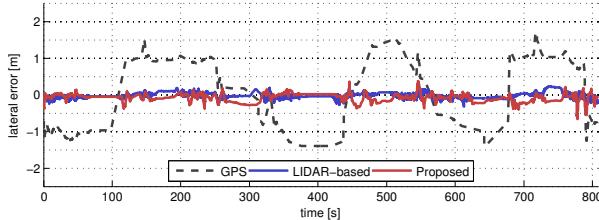
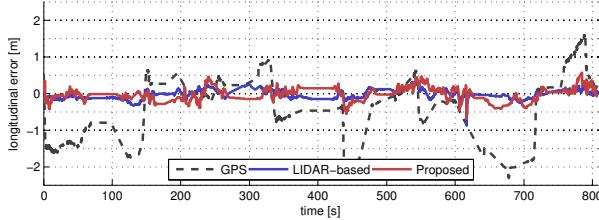
These results are also summarized in Table I. Here we show that we are able to achieve longitudinal and lateral root mean square (RMS) errors of 19.1 cm and 14.3 cm, respectively, on the *Downtown* dataset. Further, we obtain longitudinal and lateral RMS errors of 45.4 cm and 20.5 cm, respectively, on the *Stadium* dataset.

Method	<i>Downtown</i> RMS Error		<i>Stadium</i> RMS Error	
	Longitudinal	Lateral	Longitudinal	Lateral
GPS	91.0 cm	100.5 cm	81.7 cm	73.4 cm
LIDAR-based	12.4 cm	8.0 cm	14.3 cm	10.9 cm
Proposed	19.1 cm	14.3 cm	45.4 cm	20.5 cm

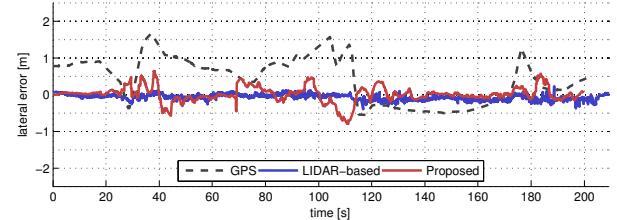
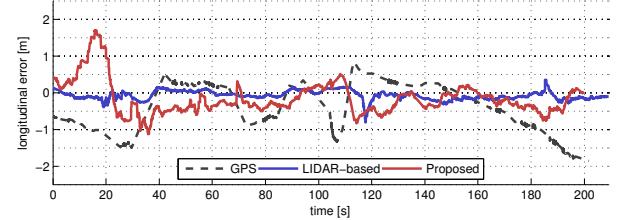
TABLE I: Comparison of RMS errors for GPS, LIDAR-based localization, and our proposed vision-only localization. Our method is able to maintain sufficiently well localized for use in an automated vehicle.

VII. CONCLUSION

In this paper, we showed that a single monocular camera can be used as an information source for visual localization in a 3D LIDAR map containing surface reflectivities. By maximizing normalized mutual information, we are able to register a camera stream to our prior map. Our system is



(a) *Downtown*—Filtered Results.



(b) *Stadium*—Filtered Results.

Fig. 10: Here we present our localization accuracy in terms of longitudinal and lateral error relative to SLAM-optimized ground-truth over time. Our proposed solution achieves a similar order of magnitude performance as the state-of-the-art LIDAR-based solutions while being several orders of magnitude cheaper. GPS alone is presented to show that it cannot provide reliable localization for automated vehicles. Despite significant longitudinal errors in the *Stadium* dataset, we are still able to maintain lateral alignment, which is critically important for lane-keep.

aided by a GPU implementation, leveraging OpenGL to generate synthetic views of the environment; this implementation is able to provide corrective positional updates at ~ 10 Hz. Moreover, we compared our algorithm against the state-of-the-art LIDAR-only automated vehicle localization, revealing that our approach can achieve a similar order of magnitude error rate, with a sensor that is several orders of magnitude cheaper.

REFERENCES

- [1] J. Markoff. Google Cars Drive Themselves, in Traffic. *The New York Times*, 10 October 2010. [Online]. Available: <http://www.nytimes.com/2010/10/10/science/10google.html>
- [2] J. Levinson, M. Montemerlo, and S. Thrun, “Map-based precision vehicle localization in urban environments.” in *Proc. Robot.: Sci. & Syst. Conf.*, Atlanta, GA, June 2007.
- [3] J. Levinson and S. Thrun, “Robust vehicle localization in urban environments using probabilistic maps,” in *Proc. IEEE Int. Conf. Robot. and Automation*, Anchorage, AK, May 2010, pp. 4372–4378.
- [4] T. Wu and A. Ranganathan, “Vehicle localization using road markings,” in *IEEE Intell. Vehicles Symp.*, June 2013, pp. 1185–1190.
- [5] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, “MonoSLAM: Real-time single camera SLAM,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 1052–1067, 2007.
- [6] E. Eade and T. Drummond, “Scalable monocular SLAM,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, vol. 1, June 2006, pp. 469–476.
- [7] A. Napier and P. Newman, “Generation and exploitation of local orthographic imagery for road vehicle localisation,” in *Proc. IEEE Intell. Vehicles Symp.*, Madrid, Spain, June 2012, pp. 590–596.
- [8] N. Carlevaris-Bianco and R. M. Eustice, “Learning temporal co-observability relationships for lifelong robotic mapping,” in *IROS Workshop on Lifelong Learning for Mobile Robotics Applications*, Vilamoura, Portugal, Oct. 2012.
- [9] A. Ranganathan, D. Ilstrup, and T. Wu, “Light-weight localization for vehicles using road markings,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, Tokyo, Japan, Nov. 2013.
- [10] A. Stewart and P. Newman, “LAPS — localisation using appearance of prior structure: 6-dof monocular camera localisation using prior pointclouds,” in *Proc. IEEE Int. Conf. Robot. and Automation*, Saint Paul, MN, May 2012, pp. 2625–2632.
- [11] C. McManus, W. Churchill, A. Napier, B. Davis, and P. Newman, “Distraction suppression for vision-based pose estimation at city scales,” in *Proc. IEEE Int. Conf. Robot. and Automation*, Karlsruhe, Germany, May 2013, pp. 3762–3769.
- [12] T. Senlet and A. Elgammal, “A framework for global vehicle localization using stereo images and satellite and road maps,” in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Barcelona, Spain, Nov. 2011, pp. 2034–2041.
- [13] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens, “Multimodality image registration by maximization of mutual information,” *IEEE Trans. Med. Imag.*, vol. 16, no. 2, pp. 187–198, 1997.
- [14] J. P. Pluim, J. A. Maintz, and M. A. Viergever, “Mutual-information-based registration of medical images: A survey,” *IEEE Trans. Med. Imag.*, vol. 22, no. 8, pp. 986–1004, 2003.
- [15] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice, “Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information,” in *Proc. AAAI Nat. Conf. Artif. Intell.*, Toronto, Canada, July 2012, pp. 2053–2059.
- [16] Z. Taylor, J. I. Nieto, and D. Johnson, “Automatic calibration of multimodal sensor systems using a gradient orientation measure,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, Tokyo, Japan, Nov. 2013, pp. 1293–1300.
- [17] M. F. Fallon, H. Johannsson, and J. J. Leonard, “Efficient scene simulation for robust Monte Carlo localization using an RGB-D camera,” in *Proc. IEEE Int. Conf. Robot. and Automation*, St. Paul, MN, May 2012, pp. 1663–1670.
- [18] B. Charmette, E. Royer, and F. Chausse, “Efficient planar features matching for robot localization using GPU,” in *IEEE Workshop on Embedded Computer Vision*, San Francisco, CA, June 2010, pp. 16–23.
- [19] A. Kitanov, S. Bisevac, and I. Petrovic, “Mobile robot self-localization in complex indoor environments using monocular vision and 3D model,” in *IEEE/ASME Int. Conf. Advanced Intell. Mechatronics*, Zurich, Switzerland, Sept. 2007, pp. 1–6.
- [20] M. Kaess, A. Ranganathan, and F. Dellaert, “iSAM: Incremental smoothing and mapping,” *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1365–1378, 2008.
- [21] A. Segal, D. Haehnel, and S. Thrun, “Generalized-ICP,” in *Proc. Robot.: Sci. & Syst. Conf.*, Seattle, WA, June 2009.
- [22] C. Studholme, D. L. Hill, and D. J. Hawkes, “An overlap invariant entropy measure of 3d medical image alignment,” *Pattern Recognition*, vol. 32, no. 1, pp. 71–86, 1999.
- [23] E. Olson, “Real-time correlative scan matching,” in *Proc. IEEE Int. Conf. Robot. and Automation*, Kobe, Japan, June 2009, pp. 4387–4393.
- [24] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly, 2008.