Monocular Localization in Urban Environments using Road Markings

Yan Lu Jiawei Huang Yi-Ting Chen Bernd Heisele

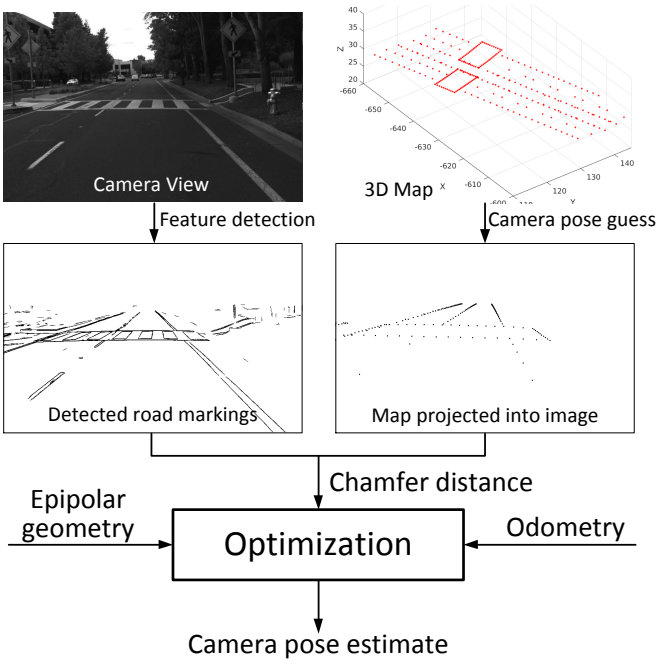
***Abstract*— Localization is an essential problem in autonomous navigation of self-driving cars. We present a monocular vision based approach for localization in urban environments using road markings. We utilize road markings as landmarks instead of traditional visual features (e.g. SIFT) to tackle the localization problem because road markings are more robust against changes in perspective , illumination, and across time. Specifically, we employ Chamfer matching to register edges of road markings against a lightweight 3D map where road markings are represented as a set of sparse points. By only matching geometry of road markings, our localization algorithm further gains robustness against photometric appearance changes in the environment. We take vehicle odometry and epipolar geometry constraints into account and formulate a non-linear optimization problem to estimate the 6 DoF camera pose. We evaluate the proposed method on data collected in the real world. Experimental results show that our method achieves sub-meter localization errors in areas with sufficient road markings.**

I. Introduction

The development of self-driving vehicles has made significant progress thanks to the advancements in perception, motion planning, and emerging sensing technologies. To achieve fully autonomous navigation, accurate localization is required. While GPS can provide global position information, it suffers from the notorious multipath effects in urban environments. Therefore, alternative methods are needed for localization in GPS-challenged environments.

The basic idea of localization is to match sensor observations against an *a priori* known map. Maps can be generated by human surveying or robotic mapping using a variety of sensors. LiDAR is a widely used sensor for mapping because it can provide accurate range measurements. A common approach is to use LiDAR in the mapping process as well as localization. However, the cost of LiDAR is too high for wide spread deployment. On the other hand, cameras are low-cost and lightweight, but visual mapping is challenging due to the lack of direct range measurement. Thus, an alternative solution is to adopt low-cost sensors (e.g., cameras) in localization and high-cost sensors (e.g., LiDAR) for mapping. The rationale is that maps need to be very accurate but do not need to be generated/updated as frequently as localization. The challenge is how to match measurements against maps that are constructed from different sensing modalities. In particular, researchers have studied monocular camera-based localization in 3D LiDAR based maps. For example, in [1] the map is augmented with surface reflectivity measurements such that synthetic views of road plane can be generated and matched against camera images; in [2] 3D points reconstructed by visual SLAM are matched against the maps generated by LiDAR SLAM.

In this work, we utilize a monocular camera to localize itself in a map which is not generated by cameras. The map is constructed by manually labeling landmarks in a 3D environment created by registering 3D LiDAR point clouds. As shown in Figure 1, our map consists of sparse 3D points



(a)

(b)

Fig. 1: (a) Overview of our proposed optimization based localization system. (b) Map (red points) projected into the camera view using the camera pose obtained by optimization.

Yan Lu, Jiawei Huang and Yi-Ting Chen are with Honda Research Institute USA, Mountain View, CA 94043, USA. {ylu,jhuang,ychen}@hra.com

Bernd Heisele is with Microsoft, Bellevue, WA 98004, USA beheisel@microsoft.com

representing landmarks (e.g., road markings). Unlike [1], we only match geometric features of the road rather than photometric features. The reason is twofold. First, our map does not contain much appearance information about landmarks; second, matching geometry allows robust localization against appearance or illumination changes. In this paper, we present a method of tracking the 6 DoF camera pose within a given map. For each image, the proposed system detects edges of road markings, and computes Chamfer distance between the detected edges and the projected road marking points in the image space. Then, we formulate a non-linear optimization problem to estimate the camera pose. The formulation takes into consideration the Chamfer distance, the vehicle odometry and epipolar constraints. Our system also detects localization failures and re-localizes itself after failures. Experimental results show that our method achieves sub-meter localization errors in areas with sufficient road markings.

II. Related work

Our work belongs to the area of robot localization [3], which refers to the process of inferring position and orientation of a robot within a given map. Maps can be generated by robotic mapping [4] or SLAM (simultaneous localization and mapping) [5].

The core of localization process is to match sensor measurements against maps. Therefore, localization approaches can be classified by sensing modalities and map representations. One category of localization methods utilizes the same type of sensor for localization and mapping, which can largely simplify the matching problem. 3D LiDAR is a popular choice due the high precision of range measurements. In [6] 3D LiDAR is employed to first map road surfaces and then localize a vehicle by correlating ground reflectivity. In [7] 3D LiDAR is used to generate 3D maps represented by Gaussian mixtures and localization is done by registering 3D point clouds with maps. The limitation of 3D LiDAR based approaches lies in the high sensor cost. In contrast, cameras are low-cost and lightweight. There is a large body of literature in visual localization using visual landmarks. For example, Se et al. create a database of visual landmarks from SIFT points [8] and then localize a camera by SIFT matching [9]. In [10] Cummins and Newman localize a camera by matching the current image against an image database using bag-of-words techniques [11]. The drawback of using camera for both localization and mapping is twofold. First, it is hard to obtain high accuracy in visual mapping/SLAM because cameras do not observe range information. Second, visual matching quality in localization can easily be affected by time, perspective and illumination changes.

To overcome the limitations mentioned above, researchers have studied the use of different sensing modalities in localization and mapping. In the mapping stage of [1], 3D LiDAR based SLAM is applied to reconstruct the 3D structure of the environment, and a dense ground-plane mesh augmented with surface reflectivity is constructed afterward; in the monocular camera based localization stage, synthetic views of the ground plane are generated and compared with the camera live view to infer the current pose. In contrast, our work does not require synthetic views generated by GPU as in [1] because our map is much more lightweight. In the visual localization process of [2], a local 3D map is reconstructed from image features using visual odometry (ORB-SLAM [12]), and then aligned with a given 3D LiDAR point could map using ICP-like techniques [13]. By only matching geometric structures, this method gains robustness against changes in the photometric appearance of the environment. Our approach is similar to [2] in the sense of merely relying on geometry matching. However, our approach can work with single static images while [2] requires a sequence of images with camera motion for reliable 3D reconstruction.

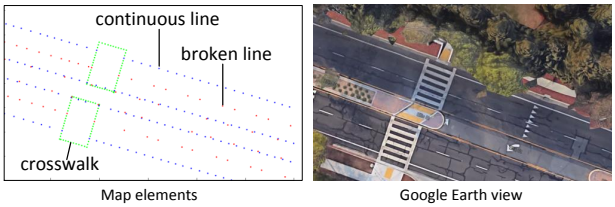
Besides sensing modalities, the selection of features or landmarks also plays a critical role in localization and SLAM. In LiDAR based approaches, landmarks can be plain point clouds or geometric structures like corners, ridges and planes. LiDAR data is typically matched using ICP [13] and its variants. In the domain of visual localization, interest points such as SIFT [14] are often used as landmarks. Matching of interest points is usually done by finding nearest neighbors [15] in high dimensional descriptor space. Alternative visual features like edges [16], lines [17], vanishing points [18] and their combinations [19], [20] have also been investigated for their robustness to illumination. In addition, higher level landmarks are also of great interest because they are closer to what humans exploit for navigation [21], [22]. In the work of [23], objects like tables and chairs are treated as landmarks for SLAM. In the context of vehicle navigation, road markings are a very useful type of high level landmarks. For example, Wu and Ranganathan [24] detect road markings such as arrows using MSER [25] and achieve robust localization against lighting changes. However, they use a stereo camera and assume a flat ground, whereas we employ a monocular camera and make no flat ground assumption.

To handle measurement noises and sensor fusion, various estimation frameworks are adopted in localization and SLAM. Kalman filtering is a popular approach (e.g. [26]) because of its simplicity and efficiency, though it assumes linear models and Gaussian noise. Particle filtering is also widely utilized [6] because it does not assume any noise model. However, particle filtering suffers from the curse of dimensionality. For visual SLAM, there are two dominant frameworks: filtering [27] and bundle adjustment [28], [29]. It is shown that bundle adjustment approaches can achieve higher accuracy due to the re-linearization when solving the non-linear optimization [30]. Thus, to obtain better localization accuracy our work adopts an optimization based framework for camera pose estimation.

III. Background and problem description

A. Map

Our maps are provided by a mapping company and consist of a variety of elements including road markings, curbs, traffic signs, etc. In this paper we only use two types of



(a)

Solid lines

[FeatureCode] Line_DivisionCont
[FeatureType] 3DLine
[FeatureTypeVer] 2012_04
[FormType] 3DLine
[FID] 000215
[3DPolyline]
-5460.2192,-35345.4338,40.7747
-5460.6251,-35345.8541,40.7708
[Width] 0.15
[ColorCode] W
[LineClass] 1
[RoadDivCode] 102
[RoadRegulationCode]
[RoadIndicationCode] 206
[FID] 000216
[3DPolyline]
-5461.6899,-35343.4173,40.7741
-5462.0432,-35343.8179,40.7802
[Width] 0.15
[ColorCode] W
[LineClass] 1
[RoadDivCode] 102
[RoadRegulationCode]
[RoadIndicationCode] 206

Broken lines

[FeatureCode] Line_DivisionBrk
[FeatureType] 3DLine
[FeatureTypeVer] 2013_04
[FormType] 3DLine
[FID] 000144
[3DPolyline]
1862485.5291,600529.2792,-12.8697
1862480.9886,600532.3923,-12.9198
1862477.5812,600534.6529,-12.9240
1862473.9232,600537.0613,-12.9419
1862471.2528,600538.7918,-12.9972
1862467.9661,600540.8946,-13.0289
1862465.3214,600542.5220,-13.0386
1862461.4600,600544.8773,-13.0873
1862458.8722,600546.4386,-13.1046
1862454.7321,600548.8735,-13.1440
1862452.3146,600550.2654,-13.1891
[Width] 0.15
[ColorCode] Y
[LineClass] 2
[RoadDivCode]103
[RoadRegulationCode]
[RoadIndicationCode]

(b)

Fig. 2: (a) Map elements used for localization. (b) Map features stored in texts. Grayed out parts are information not yet used in our system.

road markings: solid lines and broken lines. As illustrated in Figure 2(a), solid lines usually come from lane or crosswalk boundaries, while broken lines typically exist between lanes. The rationale of choosing this subset of map elements is twofold. First, they can be more frequently observed than other elements like speed limit signs and turning arrows. Second, they are relatively easy to detect from images due to their distinctive appearance and large sizes as opposed to curbs and traffic signs. In the rest of this paper, "road marking" only refer to the selected types of road markings.

The road markings are concisely stored in text files and grouped by geographic locations. As illustrated in Figure 2(b), a road marking feature is represented by a set of 3D points (sampled along its center line) along with other information such as width and color.

B. Notations

Denote a road marking by $\mathbf{M}_i := \{\mathbf{m}_{i,j} \in \mathbb{R}^3 | j = 1, \cdots, n_i\}$ with n_i being the total number of points.

We assume that the camera is pre-calibrated (i.e. intrinsic matrix K is known) with radial distortion removed. Let I_k be the image captured at time k. Define $\mathbf{P}_k := \{\mathbf{R}_k, \mathbf{t}_k\}$ to be the camera pose w.r.t. the map at time k, with $\mathbf{R}_k \in \mathrm{SO}(3)$ the rotation matrix and $\mathbf{t}_k \in \mathbb{R}^3$ the translation vector.

We assume that an odometer is available to provide relative motion between adjacent camera frames. Let $\mathbf{D}_k$ be the rigid transformation between I_{k-1} and I_k measured by the odometer.

The problem is given a map, images, and odometry up to time k, estimate the camera pose $\mathbf{P}_k$ w.r.t. the map.

IV. System

Let us assume that the system has been initialized (we defer system initialization until Section IV-E). As illustrated in Figure 1(a), at time k we detect edges of road markings in I_k. In the same time, we predict the camera pose $\mathbf{P}'_k$ using the pose estimate from the previous frame $\mathbf{P}_{k-1}$ and odometry data $\mathbf{D}_k$. We retrieve a subset of the map in neighborhood of $\mathbf{P}'_k$. Then we refine the camera pose through optimization and obtain $\mathbf{P}_k$. Let us start with feature detection.

A. Feature detection

We detect road markings by extracting their contours. However, generic edge detectors produce too many irrelevant edges (i.e. false positives). Here we adopt a random forest based edge detector [31] and retrain it with our own image data. A random forest is a collection of independent decision trees. Each tree is given the same input sample and classifies it by propagating it down the tree from the root node to a leaf node. By presenting an initial untrained decision tree with many input and output mappings, the parameters of its internal splitting functions will gradually evolve and produce similar input-output mappings. This learning process is made possible by defining an information gain criterion. Parameters resulting in maximum information gain are rewarded.

B. Feature matching

As mentioned in Section III-A, a road marking is represented by a small set of 3D points. From odometry information we can predict the camera pose $\mathbf{P}'_k$ at time k and then project the points of road markings into the image space. To evaluate how well the projected points match against the detected image feature, we use Chamfer matching which essentially associates each projected point to a nearest edge pixel. Chamfer distance can be efficiently computed from distance transform [32]. To account for orientation, we divide edge pixels into different groups by their gradient direction and compute the distance transform accordingly.

C. Map retrieval

Given a predicted camera position, we select a subset of road markings from the map that are within a certain distance (80 m) to the camera. On wide roads with center islands, road markings on the other side of road may be occluded and thus missed in the edge detection. Since we do not have information about the center island in the map, we can not predict occlusions. Our solution to this problem is that on wide roads (over 15 m) we only use road markings on the side where the vehicle drives. This is sufficient on wide roads which typically have multiple lanes in one direction. Narrow roads usually do not have center islands which means we can use lane markings from both driving directions.

As the map does not contain directional information of road markings, we need to identify the road markings on the

same side as the vehicle. From the local map, we first detect curbs on each side, and then figure out the road width as the distance between two curbs. If the road width is larger than 15 m, we forgo road markings located on the other side of the road. Note that curbs are not used for matching because they are more difficult to detect in images.

D. Optimization

To estimate the camera pose, we not only use Chamfer matching, but also take other constraints into account.

Chamfer distance. Let $\mathbf{C}_k$ be the distance transform computed from the edge image. For any point $\mathbf{x}$ on I_k, we can query its Chamfer distance $\mathbf{C}_k(\mathbf{x})$ from $\mathbf{C}_k$ by interpolation. Let $\pi(\mathbf{P}, \mathbf{X})$ be the projection function that projects a 3D point $\mathbf{X}$ from the world frame to the image with pose $\mathbf{P}$. Suppose $\mathcal{M}_k$ is the set of road marking points that are in the camera view according to the predicted camera pose $\mathbf{P}'_k$. We define

$$\mathcal{C}_{\text{chf}}(\mathbf{P}_k) = \sum_{\mathbf{X} \in \mathcal{M}_k} \mathbf{C}_k(\pi(\mathbf{P}_k, \mathbf{X})) \tag{1}$$

Road markings may not always pose sufficient constraints on camera pose estimation, for example, when there is only a straight solid lines in the view. We add the following extra constraints in the estimation process.

Epipolar constraint. Suppose $\mathbf{x}_{i,k-1} \leftrightarrow \mathbf{x}_{i,k}$ is a pair of image points from I_{k-1} and I_k, respectively, and they correspond to the same 3D point. The epipolor constraint is

$$\tilde{\mathbf{x}}_{i,k-1}^T \mathbf{F} \tilde{\mathbf{x}}_{i,k} = 0 \tag{2}$$

where $\mathbf{F}$ is the so-called fundamental matrix, and $\tilde{\mathbf{x}}$ denotes the homogeneous coordinates of $\mathbf{x}$. For a calibrated camera, $\mathbf{F}$ is determined by the relative pose between two views. Define

$$\begin{aligned} {}^{k-1}\mathbf{R}_k &:= \mathbf{R}_{k-1}^T \mathbf{R}_k \\ {}^{k-1}\mathbf{t}_k &:= \mathbf{R}_{k-1}^T (\mathbf{t}_k - \mathbf{t}_{k-1}). \end{aligned} \tag{3}$$

One can verify that $\{{}^{k-1}\mathbf{R}_k, {}^{k-1}\mathbf{t}_k\}$ is the relative rigid transformation between $\mathbf{P}_{k-1}$ and $\mathbf{P}_k$. The fundamental matrix can be computed

$$\mathbf{F} = K^{-T} [{}^{k-1}\mathbf{t}_k]_\times {}^{k-1}\mathbf{R}_k K^{-1} \tag{4}$$

where $[{}^{k-1}\mathbf{t}_k]_\times$ is the matrix representation of the cross product with ${}^{k-1}\mathbf{t}_k$.

Given a set of point correspondences $\{\mathbf{x}_{i,k-1} \leftrightarrow \mathbf{x}_{i,k}, i = 1, \cdots\}$ between I_{k-1} and I_k, we define

$$\mathcal{C}_{\text{epi}}(\mathbf{P}_{k-1}, \mathbf{P}_k) = \sum_i \tilde{\mathbf{x}}_{i,k-1}^T \mathbf{F} \tilde{\mathbf{x}}_{i,k}. \tag{5}$$

We use SURF points [33] in the epipolar constraints. Eq. (5) only poses constraints on 5 DoFs of camera pose because physical scale is not observable by a monocular camera. Therefore, we use the odometry for another constraint.

Odometry constraint. Recall that $\mathbf{D}_k$ is the rigid transformation between I_{k-1} and I_k measured by the odometer. Since epipolar constraint already covers 5 DoFs, we only use the translation magnitude of $\mathbf{D}_k$ as a constraint. Let d_k denote the magnitude of the translation component of $\mathbf{D}_k$. We define

$$\mathcal{C}_{\text{odm}}(\mathbf{P}_{k-1}, \mathbf{P}_k) = \left(d_k - \left|{}^{k-1}\mathbf{t}_k\right|\right)^2. \tag{6}$$

Optimization formulation. Given $\mathbf{P}_{k-1}$, we estimate $\mathbf{P}_k$ by minimizing the following cost function

$$\mathcal{C}(\mathbf{P}_k) = \mathcal{C}_{\text{chf}}(\mathbf{P}_k) + \mathcal{C}_{\text{epi}}(\mathbf{P}_{k-1}, \mathbf{P}_k) + \mathcal{C}_{\text{odm}}(\mathbf{P}_{k-1}, \mathbf{P}_k). \tag{7}$$

This problem can be solved using the Levenberg-Marquardt algorithm.

E. Initialization and reset

To initialize the system, we assume a rough estimate of the camera pose, which can be obtained from GPS or other sources. This initial estimate is usually too far from the true solution for the optimization to work. Instead, we need to exhaustively search for a good estimate. To do so, we randomly sample a large set of candidate poses around the rough estimate in the parameter space, and find one that minimizes $\mathcal{C}(\mathbf{P}_k)$. With the best candidate as an initial solution, we further minimize $\mathcal{C}(\mathbf{P}_k)$.

We also monitor the localization performance by checking the Chamfer distance. A large Chamfer distance can indicate a wrong localization estimate. We consider the system to be failed when consecutive large Chamfer distances occur. In case of system fail, we reset the system using the same strategy as for initialization. The only difference is that we sample candidates around the current pose estimate.

V. Experiments

We have implemented our system in C++ under Linux. We collect image data using a forward-looking camera (Point-Grey GS3-PGE-23S6M-C) mounted on our test vehicle. The image resolution we use is 864×540 (down-sampled from 1920×1200 for speed). We first evaluate our road marking detection.

A. Feature detection test

We compare our road marking detection results against the following algorithms: Canny edge detection [34], the original random forest based edge detector [31] denoted by RF-org, and a lane marker detection (LMD) algorithm [35]. We use RF-re to denote the random forest based edge detector re-trained using our road marking data.

We have collected a set of 675 images with manually annotated road markings (see Figure 3 for example), available to the public[1]. We randomly choose 457 images for training, and the rest for testing. Figure 3 shows four testing images and the outputs of all edge detection algorithms. In Figure 4, we plot the precision-recall curves for each method by varying their respective thresholds. Note that LMD only results in one point because it does not support easy thresholding. In the precision-recall plane, the upper right corner represents the ideal detection result. It is clear that RF-re outperforms the rest by a large margin. In fact, this

[1] http://datasets.honda-ri.com/roadmark

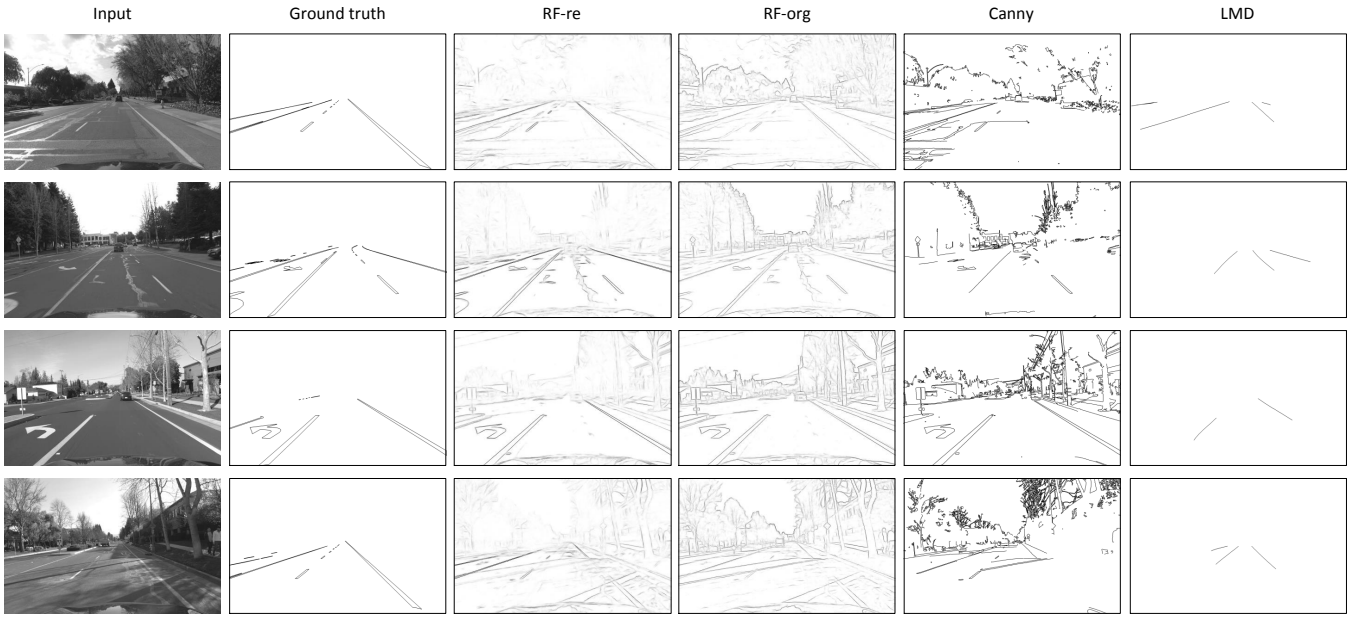


Fig. 3: Sample images for feature detection comparison. Ground truth are binary images with black indicating true edges. The results of RF-re and RF-org are grayscale images, where darker pixels have stronger belief of being edges. Canny outputs are obtained by setting a high threshold, but false positives still exist. LMD outputs are binary, showing a high false negative rate.

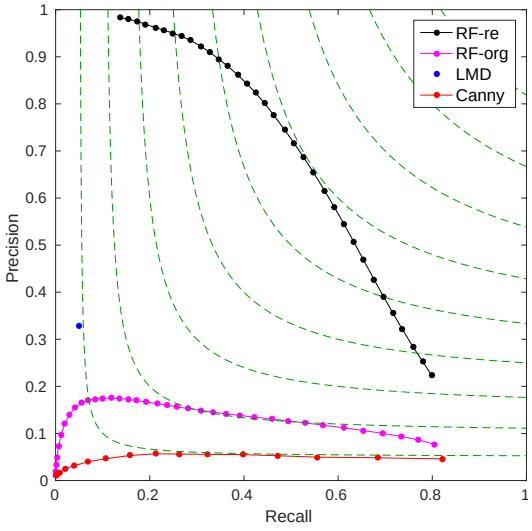


Fig. 4: Precision-recall curves for edge detection algorithms in comparison. Note that LMD only presents a single point because no easy thresholding is allowed by the algorithm.

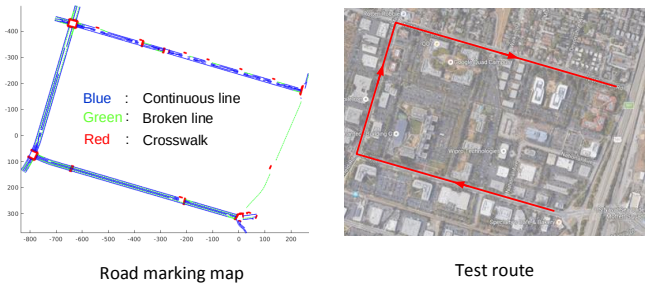


Fig. 5: Left: Localization test map. Right: The test route superimposed on Google map. Note that in one segment of the road, only a center broken line is available in the map, providing rather weak geometric constraints. In such case our algorithm is inapplicable and thus not tested.

is not surprising because generic edge detectors like Canny and RF-org produce too many false positives while LMD produces too many false negatives, as illustrated in Figure 3.

B. Localization test

We now evaluate our localization system using real map and data. Our current map covers an area of approximately 1000×500 m^2 as shown in Figure 5. We collected two sets of data: Dateset A in May 2016, Dataset B in August 2016. Both datasets traverse the same route, which is approximately 2.25 km long (see Figure 5). We use high grade GPS/INS system (OxTS RT3000) to provide ground truth.

Our test data contains many challenges, some of which are illustrated in Figure 6. For example, a new crosswalk was built after the map had been generated, some lane markings were almost invisible due to fading, vehicles caused severe occlusions of lane markings, strong shadows posed problems to the edge detector, and road work altered the appearance of the scene.

Due to the uniqueness of our sensor and map configuration, it is hard for us to benchmark the proposed method with existing algorithms. As a *baseline*, we also evaluate our system when using the generic edge detector RF-org

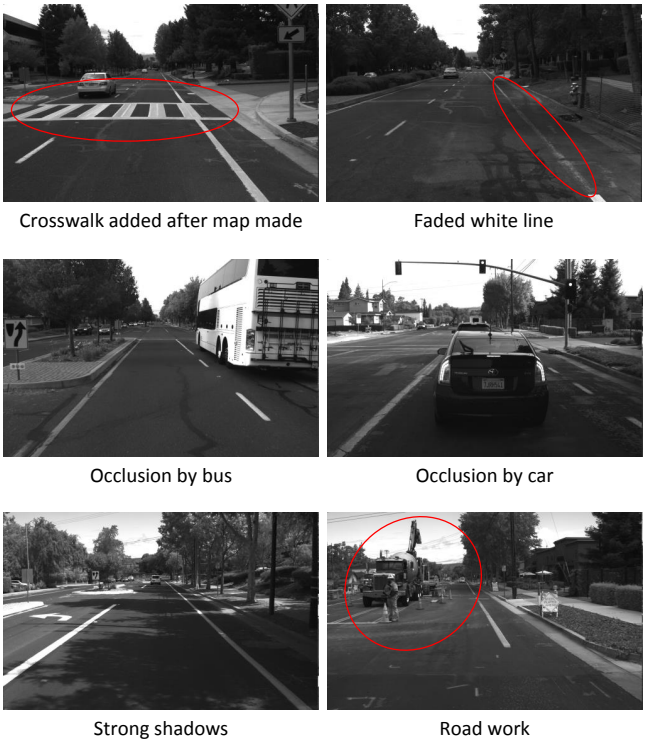


Fig. 6: Localization challenges presented in our test data.

for road marking detection as RF-org works better than Canny and LMD. Although our system estimates 6 DoF pose, we only compute longitudinal, lateral, and heading errors here because the GPS/INS estimate of altitude is not as reliable as longitude or latitude. The error distributions obtained by the proposed method are illustrated in Figure 7 and Figure 8 for each dataset, respectively. We see that the lateral error distributions seem to be biased, which can be attributed to the imperfection in the ground truth and map. If such imperfection did not exist, our localization errors should be even smaller because the projected map points in images usually align well with the road marking edges, see Figure 1(b) or supplemental materials for example.

In Table I, we report the root mean square (RMS) errors of the proposed method and the baseline. We can see that the proposed method achieves sub-meter localization errors on both datasets despite the fact that they are acquired months apart. This demonstrates the robustness of our system against changes over time. In addition, the proposed method clearly outperforms the baseline, which further justifies the necessity of re-training the random forest based edge detector with road marking data.

VI. Conclusion & Future work

Localization is a prerequisite for autonomous driving. In this paper we presented a monocular vision based localization algorithm for navigation in urban environments using road markings. We chose road markings as landmarks for localization instead of traditional visual features (e.g. SIFT) because road markings are more robust against time, perspective and illumination changes. We employed Chamfer matching to register the detected road markings in an image against their representations in a lightweight map. We further took vehicle odometry and epipolar geometry constraints into account and formulated a non-linear optimization problem to obtain the 6 DoF camera pose. We evaluated the proposed method on data collected in the real world. Experimental results showed that our method achieved sub-meter localization errors despite that the data were collected months apart. In the meantime, we are aware that the proposed method is not applicable when road markings are absent or sparse. Therefore, we will investigate using other types of landmarks in the future to achieve more robust localization.

TABLE I: RMS errors for localization.

Dataset	Method	Longitudinal(m)	Lateral(m)	Heading(°)
A	Baseline	0.426	0.857	1.03
	Proposed	0.239	0.595	0.84
B	Baseline	0.509	0.867	1.24
	Proposed	0.271	0.679	0.91

Despite various challenges presented in the test data, the proposed method obtains sub-meter level errors. Note that Dataset A and B are collected on the same route in May 2016 and August 2016, respectively.

References

[1] R. W. Wolcott and R. M. Eustice, "Visual localization within lidar maps for automated urban driving," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 176–183.

[2] T. Caselitz, B. Steder, M. Ruhnke, and W. Burgard, "Monocular camera localization in 3d lidar maps," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[3] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust monte carlo localization for mobile robots," *Artificial intelligence*, vol. 128, no. 1, pp. 99–141, 2001.

[4] S. Thrun, "Robotic mapping: A survey," in *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.

[5] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous localization and mapping with sparse extended information filters," *International Journal of Robotics Research*, vol. 23, no. 7-8, pp. 693–716, 2004.

[6] J. Levinson, M. Montemerlo, and S. Thrun, "Map-based precision vehicle localization in urban environments." in *Robotics: Science and Systems*, vol. 4. Citeseer, 2007, p. 1.

[7] R. W. Wolcott and R. M. Eustice, "Fast lidar localization using multiresolution gaussian mixture maps," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 2814–2821.

[8] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 4, pp. 91–110, November 2004.

[9] S. Se, D. G. Lowe, and J. J. Little, "Vision-based global localization and mapping for mobile robots," *IEEE Transactions on robotics*, vol. 21, no. 3, pp. 364–375, 2005.

[10] M. Cummins and P. Newman, "FAB-MAP: Probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[11] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 1470–1477.

[12] R. Mur-Artal, J. Montiel, and J. D. Tardós, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[13] P. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

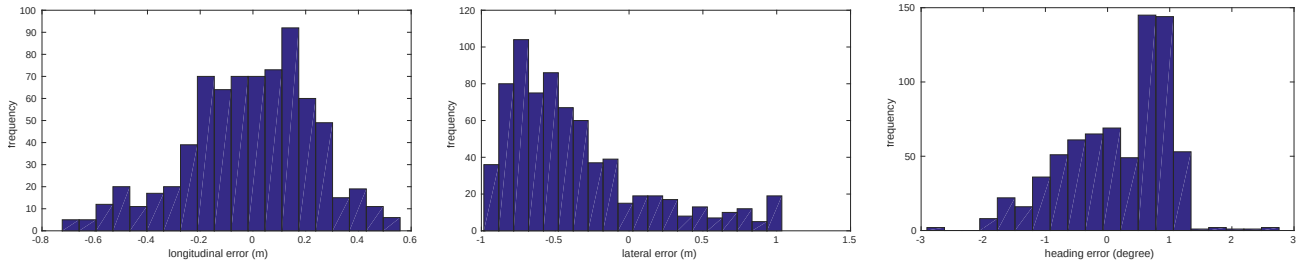


Fig. 7: Error histograms of Dataset A. From left to right: longitudinal (m), lateral (m), and heading (degree).

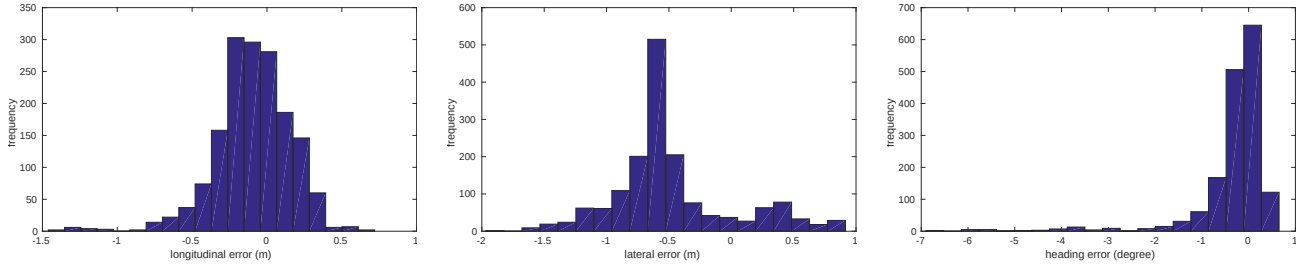


Fig. 8: Error histograms of Dataset B. From left to right: longitudinal (m), lateral (m), and heading (degree).

[14] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2, 1999, pp. 1150–1157.

[15] M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 11, pp. 2227–2240, 2014.

[16] E. Eade and T. Drummond, "Edge landmarks in monocular SLAM," *Image and Vision Computing*, vol. 27, no. 5, pp. 588 – 596, 2009.

[17] Y. Lu and D. Song, "Robustness to lighting variations: An RGB-D indoor visual odometry using line segments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 688–694.

[18] G. Zhang, D. H. Kang, and I. H. Suh, "Loop closure through vanishing points in a line-based monocular SLAM," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 4565–4570.

[19] Y. Lu and D. Song, "Robust RGB-D odometry using point and line features," in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 3934–3942.

[20] H. Li, D. Song, Y. Lu, and J. Liu, "A two-view based multilayer feature graph for robot navigation," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2012, pp. 3580–3587.

[21] Y. Lu, D. Song, and J. Yi, "High level landmark-based visual navigation using unsupervised geometric constraints in local bundle adjustment," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1540–1545.

[22] J. Lee, Y. Lu, Y. Xu, and D. Song, "Visual programming for mobile robot navigation using high-level landmarks," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 2901–2906.

[23] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, "Slam++: Simultaneous localisation and mapping at the level of objects," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1352–1359.

[24] A. Ranganathan, D. Ilstrup, and T. Wu, "Light-weight localization for vehicles using road markings," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 921–927.

[25] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, September 2004.

[26] J. Ziegler, H. Lategahn, M. Schreiber, C. G. Keller, C. Knöppel, J. Hipp, M. Haueis, and C. Stiller, "Video based localization for bertha," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE, 2014, pp. 1231–1238.

[27] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, June 2007.

[28] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment – a modern synthesis," in *Vision Algorithms: Theory and Practice*. Springer, 2000, pp. 298–372.

[29] Y. Lu and D. Song, "Visual navigation using heterogeneous landmarks and unsupervised geometric constraints," *IEEE Transactions on Robotics (T-RO)*, vol. 31, no. 3, pp. 736–749, June 2015.

[30] H. Strasdat, J. M. Montiel, and A. J. Davison, "Visual SLAM: Why filter?" *Image and Vision Computing*, vol. 30, no. 2, pp. 65–77, 2012.

[31] P. Dollár and C. L. Zitnick, "Fast edge detection using structured forests," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 8, pp. 1558–1570, 2015.

[32] H. Barrow, J. Tenenbaum, R. Bolles, and H. Wolf, "Parametric correspondence and chamfer matching: Two new techniques for image matching," in *International Joint Conference on Artificial Intelligence*, 1977, pp. 659–663.

[33] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *European Conference on Computer Vision (ECCV)*. Springer, 2006, pp. 404–417.

[34] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[35] M. Aly, "Real time detection of lane markers in urban streets," in *Intelligent Vehicles Symposium, 2008 IEEE*. IEEE, 2008, pp. 7–12.