Curb-intersection feature based Monte Carlo Localization on urban roads

The MIT Faculty has made this article openly available. Please share how this access benefits you. Your story matters.

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>As Published</td>
<td><a href="http://dx.doi.org/10.1109/ICRA.2012.6224913">http://dx.doi.org/10.1109/ICRA.2012.6224913</a></td>
</tr>
<tr>
<td>Publisher</td>
<td>Institute of Electrical and Electronics Engineers (IEEE)</td>
</tr>
<tr>
<td>Version</td>
<td>Author's final manuscript</td>
</tr>
<tr>
<td>Accessed</td>
<td>Sat Dec 22 13:11:26 EST 2018</td>
</tr>
<tr>
<td>Citable Link</td>
<td><a href="http://hdl.handle.net/1721.1/81465">http://hdl.handle.net/1721.1/81465</a></td>
</tr>
<tr>
<td>Terms of Use</td>
<td>Creative Commons Attribution-Noncommercial-Share Alike 3.0</td>
</tr>
<tr>
<td>Detailed Terms</td>
<td><a href="http://creativecommons.org/licenses/by-nc-sa/3.0/">http://creativecommons.org/licenses/by-nc-sa/3.0/</a></td>
</tr>
</tbody>
</table>
Curb-Intersection Feature Based Monte Carlo Localization on Urban Roads

B. Qin*, Z. J. Chong*, T. Bandyopadhyay†, M. H. Ang Jr.* E. Frazzoli§†, D. Rus§‡
* National University of Singapore, Kent Ridge, Singapore
† Singapore-MIT Alliance for Research and Technology, Singapore
§ Massachusetts Institute of Technology, Cambridge, MA., USA.

Abstract—One of the most prominent features on an urban road is the curb, which defines the boundary of a road surface. An intersection is a junction of two or more roads, appearing where no curb exists. The combination of curb and intersection features and their idiosyncrasies carry significant information about the urban road network that can be exploited to improve a vehicle’s localization. This paper introduces a Monte Carlo Localization (MCL) method using the curb-intersection features on urban roads. We propose a novel idea of “Virtual LIDAR” to get the measurement models for these features. Under the MCL framework, above road observation is fused with odometry information, which is able to yield precise localization. We implement the system using a single tilted 2D LIDAR on our autonomous test bed and show robust performance in the presence of occlusion from other vehicles and pedestrians.

I. INTRODUCTION

Intelligent Vehicle/Highway Systems (IVHS) have been one of the most popular research areas in robotics. By realizing autonomous navigation, intelligent vehicles enhance operational safety and efficiency of the transportation system. Localization is one fundamental requirement for vehicle autonomy. This paper investigates the ability to localize under minimal sensing, with only one LIDAR, odometry information and prior road network information.

In the past few years, researchers spent much effort on the fusion of Global Positioning System (GPS) and Inertial Navigation System (INS) to estimate vehicle position[12], [13], [3]. This approach usually achieves good accuracy in open areas; however, the performance deteriorates in dense urban environment, where GPS signal quality gets severely undermined by satellite blockage and urban multipath propagation due to high buildings, as discussed in [11]. To overcome this problem, road-matching method can be used. The basic idea underlying road-matching is to treat road constraint of vehicle motion as observation. The road is perceived as a line segment, with no lane width information. By checking the on-driving road segment with a road-map, additional localization information can be derived. In [5], Najjar et al. propose a road-matching localization algorithm, using Belief Theory for road selection and Kalman Filtering for recursive estimation. Some other similar studies can be found in [7], [6]. These road-matching algorithms achieve good localization in a global fashion. However, they are not designed to generate accurate position relative to the road. In this sense, the localization is at a coarse level, which may be inadequate for a vehicle performing complex tasks on road surface.

In other studies, researchers refer to local features for high precision localization. In [14], lane markers are extracted to reduce localization error. But due to the fact that lane makers just carry lateral position information, longitudinal localization error can be reduced only when the road curvature is big. In [8], a novel “Virtual 2D Scans” method is proposed by making use of the building outlines as features. A simplified 2D line feature map is generated beforehand as prior knowledge. However, possible lack of building features and slow update rate limits its effectiveness.

One of the most dominant features on an urban road is the curb. Serving as the road boundary, curb features carry much richer localization information than lane markers. In [9], [2], single side curb features are extracted by a vertical LIDAR to improve vehicle localization together with road information. While these algorithms reduce lateral localization error considerably, they help little in the longitudinal direction. Intersection features appear at junctions of roads, where no curb exists. While curb features mostly help localize vehicle laterally in the road, intersection features carry rich longitudinal information. The combination of curb and intersection features gives a complete picture of the urban road network. The complementary nature of these two kinds of features makes it well suited to improve the vehicle localization.

This paper proposes a Monte Carlo Localization (MCL) method using the curb-intersection feature on urban roads. Our main contribution is to propose a novel idea of “Virtual LIDAR”, where curb and intersection measurements are utilized to improve localization accuracy laterally and longitudinally. This paper adopts the MCL framework to fuse odometry information with road observation, which is able to yield precise pose estimation. We implement the system using a single tilted 2D LIDAR to detect the curb-intersection features and show robust performance in the presence of occlusion from other vehicles and pedestrians.

The remainder of this paper is organized as follows. In Section II, the extraction of curb and intersection features is introduced. Section III provides details of the curb-intersection based MCL method. Experimental results and analyses are presented in Section IV. Finally, Section V concludes the paper and discusses future work.
II. CURB-INTERSECTION FEATURE EXTRACTION

There are numerous studies for the detection of road boundary. One of the ways is to utilize a tilted-down LIDAR for curb detection. Cramer et al. [4] applied Hough Line for scan segmentation and feature extraction, while Kodagoda et al. [16], [10] achieved the same goal by using EKF filter. This paper presented an intuitive two-step method to detect both curb and intersection, which proves to be efficient and robust. Similar work can be found in [17].

A. Segmentation of Laser Scan

In the first step, one single laser scan is segmented into several pieces, by virtue of its typical laser range/angle characteristics on road. Fig. 1 shows the model of sensing on road. One LIDAR sensor is mounted at point O, with its laser beam angle as $\theta$. The distance between point P and O’ is the look-ahead distance of the tilted-down LIDAR. As presented in Fig. 1, laser beams from point O are cast onto different planes, i.e. road surface plane, curb plane, road shoulder plane, and so on. From this model, a piecewise function can be derived to represent the relationship between the beam angle and range value, with each interval corresponding to an individual plane. Without involving too many details, a simplified formula can be represented as:

\[
  r(\theta) = \begin{cases} 
  R_{leftCurb}(\theta) & \text{for } \theta_C \leq \theta \leq \theta_B \\
  R_{roadSurf}(\theta) & \text{for } \theta_B \leq \theta \leq \theta_C \\
  R_{rightCurb}(\theta) & \text{for } \theta_E \leq \theta \leq \theta_D \\
  \ldots & \ldots 
  \end{cases}
\]

Due to the piecewise fact of function $r(\theta)$, a second-order differential filter can be implemented to detect the edges:

\[
  r_f(\theta) = \sum_{i=-5}^{i=5} r(\theta + i \times \mu) + \sum_{i=-2}^{i=2} r(\theta + i \times \mu) - 5 \mu \]

where $\mu$ is angular resolution of the LIDAR sensor, and $\theta \in [-\pi/2 + 5\mu, \pi/2 - 5\mu]$. Boundary points are extracted as local maxima or minima in the filter response plot, and their values should exceed certain threshold, as shown in Fig. 2.

B. Classification of Scan Segments

In the second step, scan segments generated are fed into a sequential classification process.

1) Road surface segment, shown as line CD in Fig. 1, is selected first. It always locates between two edge points nearest to center of the sensor.
2) Curb lines, (BC and DE of Fig. 1), are searched subsequently, based on point C and D determined from the former step.
3) Rest segments are other features off the road.

Some restrictive criteria are applied during above steps, such as road width, curb height, etc. Specifically, to extract a valid curb feature, the length of segment CD should be bigger than the minimum value of road width, the curb height of segment BC (or DE) should be within certain range, and the number of laser beams on BC (or DE) should be over certain threshold, etc. Only when all these criteria get satisfied, classification result is thought to be valid. Thus most noise like vehicles and pedestrians get filtered. One typical classification result is shown in Fig. 3. Among these categories, curb is saved for further usage.

It should be clarified that, a fixed maximum detection range is defined for the curb extraction algorithm. If the distance of curb edge (C or D), to projected center O’ equals to or exceeds this range, the curb features are deemed unreliable, and will yield “no-curb” result. For this reason, above algorithm doesn’t apply to situations at intersection, where curb is too far away, or there may be no curb at all. However, the fact of “no-curb” also carries useful localization information. To embody this kind of information, a virtual “intersection feature” is introduced.

As shown in Fig. 4, intersection feature is represented by a virtual beam $\overrightarrow{PR}$ (or $\overrightarrow{PL}$), with its direction tangent to $O’P$ and distance as maximum detection range of curb extraction. It should be clarified that feature extraction of left and right sides are independent, enabling both curb and intersection...
III. MONTE CARLO LOCALIZATION ALGORITHM
A. MCL Overview
In this paper, Monte Carlo Localization (MCL) is applied to estimate the vehicle pose. MCL is a probabilistic localization algorithm based on Bayes’ Theorem and Monte Carlo method. A thorough study is made by Sebastian Thrun et al [15]. The belief $\text{bel}(x_t)$ in MCL is represented by a set of $M$ particles $\{x_t^m, w_t^m\}_{m=1}^M$, and each particle is paired with an importance weight $w_t^m$.

$$\text{bel}(x_t) \sim \{x_t^m, w_t^m\}_{m=1}^M$$  \hspace{2cm} (3)

MCL estimates the position of the vehicle recursively by repeating the following steps:
1) Prediction: a new set of particles $\{x_t^m, w_t^m\}_{m=1}^M$ for time $t$ is generated with $\{x_{t-1}^m, w_{t-1}^m\}_{m=1}^M$ and the control $u_t$, according to certain motion model $p(x_t | u_t, x_{t-1})$.
2) Correction: the importance weight of each particle in $\{x_t^m, w_t^m\}_{m=1}^M$ is adjusted with new measurements $z^t$, according to certain measurement model $p(z_t | x_t, m)$.
3) Resampling: the particle set $\{x_t^m, w_t^m\}_{m=1}^M$ will be resampled when necessary. After resampling, the distribution of the particles approximates $\text{bel}(x_t)$.

B. Pseudo-3D Odometry Motion Model
In prediction step, a motion model is applied to propagate particles for prior belief distribution $\text{bel}(x_t)$. Generally, a 2D motion model in [15] is enough. Even for a vehicle moving in 3D world, we solve the localization problem on its horizontal projection plane, as shown in Fig. 5. Here we extend the 2D motion model to a Pseudo-3D one, by introducing a pitch noise part.

Table I represents the Pseudo-3D Odometry Sample Motion Model. This model is used for sampling from $p(x_t | u_t, x_{t-1})$ with relative motion information on the horizontal plane. Here $\gamma$ denotes the pitch angle, $\delta_{\text{rot}1}$ the initial rotation on the projected plane, $\delta_{\text{trans}}$ the translation and $\delta_{\text{rot}2}$ the second rotation. More details can be found in [15].

C. Curb-Intersection Measurement Model
In correction step, importance weights of particles get adjusted based on measurements and related sensor models. Here we propose a novel idea of “Virtual LIDAR” to obtain the required models.

a) Virtual LIDARs on Horizontal Plane: As discussed in previous sections, measurements extracted here are curb segments (line $BC$ and $DE$), or virtual beams ($\overrightarrow{PR}$ and $\overrightarrow{PL}$). However, because curb-intersection features are extracted using a tilted-down LIDAR, building its measurement model is not an easy task. To simplify computation, we try to solve the problem on 2D horizontal plane ($z=0$), as illustrated by Fig. 6.
Together with their casting origins, the new virtual assembled LIDAR-V2, two virtual beams are recorded at different time, a new virtual LIDAR, denoted as LIDAR-VSA1. As for serving as endpoints of virtual laser beams casted from time are then translated into the latest LIDAR coordinate, retained for each scan. Curb points recorded at different computational cost, only two curb points (C and D) are and fed into the MCL processing. The assembled virtual scan can then be treated as a real scan, remains accurate within a short distance interval. The new virtual LIDARs carry much sparser information than real ones, it with planar LIDARs. However, because scans of the virtual intersection feature is reduced to a common MCL problem, it is always advisable to obtain curb-intersection feature met along their virtual light path. Only through ray tracing in “Beam Model” can this working manner gets properly interpreted.

b) Scan-Assembled Virtual LIDARs: With two virtual LIDARs established, the MCL problem using curb-intersection feature is reduced to a common MCL problem with planar LIDARs. However, because scans of the virtual LIDARs carry much sparser information than real ones, it is advisable to assemble several scans at different time into one. The assumption validating this operation is odometry remains accurate within a short distance interval. The new assembled virtual scan can then be treated as a real scan, and fed into the MCL processing.

During the scan assembling of LIDAR-V1, to reduce computational cost, only two curb points (C and D) are retained for each scan. Curb points recorded at different time are then translated into the latest LIDAR coordinate, serving as endpoints of virtual laser beams casted from a new virtual LIDAR, denoted as LIDAR-VSA1. As for LIDAR-V2, two virtual beams are recorded at different time, together with their casting origins. The new virtual assembled LIDAR is even more exotic: it is composed of several 2-beam range finders, with each finder mounted at different positions and angles. When the beam number of LIDAR-VSA1 (or LIDAR-VSA2) exceeds certain “Assembling Threshold”, one virtual measurement is published. Finally, we get two new virtual LIDARs: LIDAR-VSA1 for curb feature, and LIDAR-VSA2 for intersection feature, as shown in Fig. 7.

D. Practical Considerations

a) MCL Estimation Frequency: In this paper, MCL estimation loop is triggered by the arrival of virtual measurements. Whenever an assembled virtual scan from “LIDAR-VSA1” or “LIDAR-VSA2” is available, prediction step is performed in a retrospect manner, followed by a correction step with the new incoming measurement. In this sense, the frequency of virtual scan is quite important. To control the frequency of “LIDAR-VSA1” and “LIDAR-VSA2”, we can either control the frequency of curb-intersection feature extraction, or control their “Assembling Threshold”. We find it is always advisable to obtain curb-intersection feature when the vehicle moves, and suspend the process when stopping. “Assembling Threshold” is determined by trading off MCL response speed and robustness.

b) Algorithm Robustness: The robustness of MCL is one of key issues. A reasonably big “Assembling Threshold” will help the algorithm to resist measurement noise. Actually, before curb-intersection is fed into MCL, we adopt the temporal EKF method in [16] to reduce measurement noise. The temporal filter is applied after curb extraction and before scan assembling operation. In the filter update step, if the Mahalanobis distance between the detected curb and the predicted one exceeds certain threshold, the newly detected curb will be considered as noise and discarded. This EKF method helps to eliminate minor noise like pedestrians and small cars.
Another strategy that we apply to increase algorithm robustness is the injection of random particles in [15]. When the vehicle is locally lost or the measurement is badly corrupted for sometime, the short-term average of particle importance factors will be decreased remarkably. In this case, a fraction of particles will be generated around the predicted position, and spread according to a uniform distribution within certain range.

c) Map-Incorporated Prediction: For vehicle localization on urban roads, one assumption is that vehicles are not likely to drive off-road. This assumption allows us to penalize those erratic particles by decreasing their weight importance. In this way, map information is also incorporated into the prediction step, which makes our localization more robust.

IV. EXPERIMENTS

A. Experimental Setup

Our test bed is a Yamaha G22E golf cart with various sensors, as shown in Fig. 8. We use one SICK LMS 291 LIDAR for curb and intersection detection. It is mounted in the front, with a tilted-down angle of 18 degrees. One wheel encoder (Scancon-2RS) and one IMU (3DM-GX3-25) are mounted on the cart provide necessary odometry information (distance, pitch and yaw). The proposed algorithm is tested online. In the experiment, the golf cart is driven manually on a hilly road at the campus of National University of Singapore, from point S to G, as shown in Fig. 9. Several big slopes are involved along the way, with the maximum height difference over 10 meters. The average speed in test is about 3.5 m/s. The reference road map is an occupancy grid map manually generated from a vector-format road map provided by Land Transport Authority (LTA) of Singapore and satellite map. The size of this road map is 200 meter by 240 meter, with grid resolution of 0.1 meter, as shown by Fig. 10.

B. Experimental Results

In the test, the golf cart is given a rough initial position at S, and driven for about 430 meters to G. The localization results are shown in Fig. 9. The light-blue lines denote road boundary. The red line marks the localization result of the curb-intersection feature based MCL, and raw odometry trace is shown by yellow line. For comparison, we also give the localization result from one state of art GPS/INS module (Ublox EVK-6R) by green dotted line. From Fig. 9, it is apparent that dead-reckoning odometry drifts a lot after certain distance. Even when it is fused with GPS, INS/GPS trajectory tends to fall out of road boundary. Because our
algorithm incorporates road surface information, it helps
to correct the odometry and yield fairly decent estimation.
Fig. 10 shows the occupancy grid map of road boundary.
The green points represent the curb features detected in the
experiment, overlaid on top according to localization results.
Some unexpected points in the figure are measurement noise.

To evaluate the localization result, estimation errors of
position and attitude are calculated against ground truth
values. We rely on our occupancy grid map to get the ground
truth. When the ground truth is needed, vehicle position
relative to the road network is measured carefully and marked
onto the map image. By counting the pixel in the image,
the ground truth can be calculated easily. The vehicle was
driven manually to the selected points marked in Fig. 9 and
the errors in location estimate are plotted in Table II. It can
be seen that position error of our algorithm is usually small,
less than 0.6 meter; and the orientation estimation is quite
accurate, less than 3 degrees to the ground truth.

From Table II, one can also observe that position errors
at some critical points of intersections and turnings (like
A, C, D, F) are much smaller than that of the straight
road (like B). The phenomena can be explained from the
estimated variance of particles. Fig. 11 shows “estimation
variance” vs “driving distance” in road longitudinal and
lateral direction. During the whole test, lateral estimation
variance remains small, which means particles are confident
about the lateral position. However, the longitudinal variance
changes remarkably along the drive, which determines the
accuracy of localization.

During the trip from A to B, the longitudinal variance
increases first, due to consistency of road boundary. When
the road represents a small curvature, curb features embody-
ing this information will reduce the longitudinal variance.
Thanks to the look-ahead distance of the tilted-down LIDAR,

\[
\begin{array}{cccccc}
\text{Marked Points} & A & B & C & D & E & F & G \\
\text{Position Error (m)} & 0.20 & 0.55 & 0.06 & 0.20 & 0.32 & 0.06 & 0.08 \\
\text{Orientation Error (deg)} & < 3 & & & & & & \\
\end{array}
\]

the vehicle will sense this information before it actually
arrives there. When the vehicle is approaching the inter-
sections and turnings (like A, C, D, F), the particles are
condensed significantly by the detected intersection and the
tightly curved curb features. The longitudinal variance at
these points is usually less than 0.4 meter. Hereby it can
be concluded that, while curb features on straight roads help
to estimate the lateral position, the intersection and tightly
curved curb features contribute very much to the longitudinal
positioning. Fig. 12 shows typical particle behaviours around
point B, C, D. The red arrays are particles, with green lines as
detected curbs, and purple segments to visualize intersection
features.

In the experiment, there is one situation where measure-
ment noise becomes severe: when the vehicle is passing by
an intersection at F. As mentioned in Section III, injection
of random particles is performed to overcome this “noisy
situation”. This operation leads to an increase of the esti-
mation variance, as reflected in Fig. 11. As long as new
reliable measurements come in, particles quickly converge,
and localization quickly recovers from the bad situation.
Actually, although light measurement noise happens from
time to time in the test, the localization is hardly disturbed.
The robustness of this algorithm is proved.

Besides the manual drive, we conducted another simple
semi-autonomous drive to test our localization algorithm.
The vehicle is required to navigate from point S to G by
following a predefined route. While the throttle and brake
are controlled manually, the steering is controlled by an on-
board computer. It turns out that the localization is accurate
enough for the vehicle to reach its target smoothly.

C. Autonomous System Demonstration

As a part of the overall goal of attaining mobility on
demand, we conducted an autonomous system demonstration
in July 2011, where we had guests request the vehicle to
navigate from a pickup location to pre-specified drop-off lo-
cations shown in Fig. 13. More details and videos of the oper-
...influence much the control of vehicle motion. The look-ahead knowledge. From experiment results, our algorithm proves to occupancy grid map for road boundary is used as prior LIDAR" is applied to get the measurement models. An based on the curb-intersection feature, which is extracted events were detected as no-information case and recovered curb detection was hampered briefly by traffic. However such only localization failed at an inclined T-junction 2 times over demonstration, the autonomous vehicle serviced almost 10 requests from the guests, running over 7 km and the curb-only localization failed at an inclined T-junction 2 times over the whole demo. The reason for failure was determined to be lack of curb features and planar maps at the intersections and T-junctions, which prompted us to include such intersection features resulting in the localization scheme presented in this paper. Since then till date we have covered over 50km in autonomous runs during various demonstrations with onboard passengers without the localization failing in any segment of the route. This included situations where the curb detection was hampered briefly by traffic. However such events were detected as no-information case and recovered from once the sensory occlusion was overcome.

V. CONCLUSIONS AND FUTURE WORK

This paper proposes a Monte Carlo Localization algorithm based on the curb-intersection feature, which is extracted through a two-step procedure. A novel idea of "Virtual LIDAR" is applied to get the measurement models. An occupancy grid map for road boundary is used as prior knowledge. From experiment results, our algorithm proves to be accurate and robust. Although the longitudinal estimation variance may increase at a long straight road, it will not influence much the control of vehicle motion. The look-ahead distance in the feature extraction can help vehicles to localize accurately before they reach crossings and turnings.

One disadvantage that limits the proposed algorithm is its reliance on an occupancy grid map. It is laborious to generate this map manually, and its storage is also not efficient. We plan to substitute the occupancy grid map with a vector map. In future work, other features of urban roads like lane markings will also be exploited for better localization.

REFERENCES