

Road vehicle localization with 2D push-broom LIDAR and 3D priors

Ian Baldwin
Department of Engineering Science
Oxford University
Oxford, UK
Email: ib@robots.ox.ac.uk

Paul Newman
Department of Engineering Science
Oxford University
Oxford, UK
Email: pnewman@robots.ox.ac.uk

Abstract—In this paper we describe and demonstrate a method for precisely localizing a road vehicle using a single push-broom 2D laser scanner while leveraging a prior 3D survey. In contrast to conventional scan matching, our laser is oriented downwards, thus causing continual ground strike. Our method exploits this to produce a small 3D swathe of laser data which can be matched statistically within the 3D survey. This swathe generation is predicated upon time varying estimates of vehicle velocity. While in theory this data could be obtained from vehicle speedometers, in reality these instruments are biased and so we also provide a way to estimate this bias from survey data. We show that our low cost system consistently outperforms a high caliber integrated DGPS/IMU system over 26 km of driven path around a test site.

I. INTRODUCTION

This paper is about the precision localization of a road vehicle using laser. Our focus is the cheap run time exploitation of prior 3D maps using a single 2D push-broom laser - a fixed scanning LIDAR sensor which generates data from being “pushed” through the world by a vehicle. Our motivation here is a vision of many low cost vehicles leveraging, and sharing, a potentially expensive but one-shot survey. To this end our localization scheme has no reliance on 3D laser, differential GPS or integrated inertial systems. Indeed, concerning the latter, we shall show how our system can outperform a commercial grade IMU system over 26 km of driven distance.

An immediate question is why bother using a 3D map at all - does simple 2D scan matching not suffice? Our experience leads us to answer with a resounding no. The combinations of roll and pitch, ground strike and grazing effects from hedgerows, verges, cambers and slopes makes vanilla techniques fragile. Indeed it was in the act of trying to accommodate ground strike and non-prismatic workspaces within conventional scan matching techniques that led us to what we propose and demonstrate here. We intentionally orient our single laser downwards - seeking out ground strike. By integrating estimates of the vehicle motion we can generate a 3D swathe of laser data. This dense “recent history” can be continually aligned within our survey 3D point-cloud by considering and matching the population statistics of the swathe and survey patch. The problem now is how to estimate vehicle motion.

Triaxial MEM’s gyroscopes are now ubiquitous and provide a cheap way to estimate angular rates. A velocity profile though is harder to come by. Immediately we can reach for indicated velocity coming from a road vehicle’s speedometer which, by law, they all possess. However, for obvious reasons, these devices typically overestimate velocity

and so in this paper we also admit the possibility that we must perform an online calibration of the velocity feed. We do this by detecting the passage of road markings through a fan of laser sweeps and recover vehicle velocity by deducing (by dynamic programming) a warping between a detected reflectance signal and that predicted given the prior survey.

One could argue that DGPS and IMU systems will, in the end, become inexpensive and therefore ubiquitous. While this is true, one still needs to worry about the operating envelope of such systems - in particular in areas deprived of good DGPS reception. As an illustration, consider Figure 1. This shows the Begbroke test site around which we gathered the 26 km of data used in this paper:

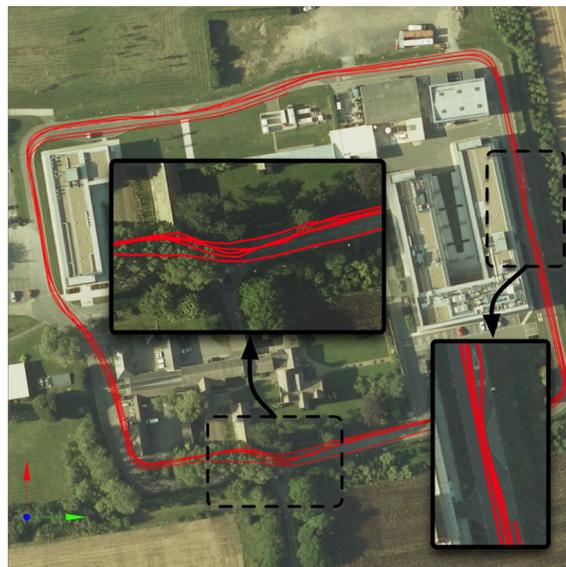


Fig. 1: Loops around the Begbroke site in Oxford, contrasting areas of good GPS reception (right), with poorer areas, (bottom section, amongst heavy foliage), and hence larger INS pose estimation errors. The global origin is shown in the lower-left. The 26km of data used in the paper was gathered around this site.

Shown in Figure 1 are the indicated trajectories from a high caliber OxTS INS system as it traverses the science park. Note that in the sections unobstructed by foliage and buildings (predominantly the north-east and western sections), the performance is good and the estimated vehicle path follows the road shape. However in the southern section a small copse of trees inhibits DGPS corrections and the INS-estimated

path diverges wildly from the road - no such issues arise in our system.

II. RELATED WORK

Laser-based outdoor localization of road vehicles using prior maps has recently been addressed by several authors. In [1] the authors make use of an actuated 3D laser sensor (the Velodyne) in addition to GPS, IMU and odometry data to generate a precise map offline. Localization within this map is performed with a particle filter. Although this work is the most similar to our approach in spirit, we do not require an expensive, actuated sensor or offline relaxation in order to generate a map. Furthermore the nature of our laser data requires us to adopt a quite different objective function. The authors extend this work in [2], developing a long-term probabilistic framework, again employing a Velodyne. Kummerle et. al. [3] develop an autonomous system that utilizes multi-level surface maps obtained from a Velodyne to represent the environment with a 2.5D structure. Most recently, [4] utilize the Velodyne in a SLAM framework. The authors in [5] utilize a 2D road-network map and a laser in conjunction with a DGPS-enabled receiver to estimate pose. Again, we seek to eliminate the requirement for an expensive, calibration-intense device (such as the Velodyne) or a persistent GPS signal in order to conduct long-term navigation.

In [6], the authors utilize a robust ICP [7] algorithm to perform local scan-matching, and incorporate a histogram-feature representation to match local submaps. In [8], the same authors utilize a rotating, scanning laser in order to generate 3D point-clouds maps. For this work, we take an approach of intentionally declining the laser to observe ground-strike, and therefore eschew the use of ICP methods which, given the scanning pattern of the beam, gives rise to multiple local minima.

Airborne Laser Swath Mapping (ALSM) [9] is a related concept in the field of geoinaging, that utilizes a similar swath-generation technique in order to generate consistent environment models - however, the system is not used for localization, instead using INS data for map fusion.

Visual Teach and Repeat [10] is a comparable vision-based system in which a stereo-camera is used to build a manifold map consisting of previously-visited submaps (the *teach* phase), that is then used in subsequent traversals for localization (*repeat*). Our approach differs in terms of leveraging active-sensing to reduce susceptibility to lighting/illumination change, in addition to alleviating the need for feature identification. VT&R also exhibits a relatively small convergence basin, which is not the case with our approach.

III. OVERVIEW

We will refer to the survey pass through the workspace as an experience, consisting of laser-sweeps, poses, velocities (both angular and linear) and accelerations. We adopt this terminology to emphasize that we do not need a globally correct metric map for our road localization task. We simply wish to localize ourselves with respect to the trajectory and a point-cloud defined by a prior experience - which in this case is a drive around urban roads. Of course this does not preclude the case of localizing against a trajectory that is metrically perfect - having been massaged by a global optimization with loop closures etc.- we just don't *require*

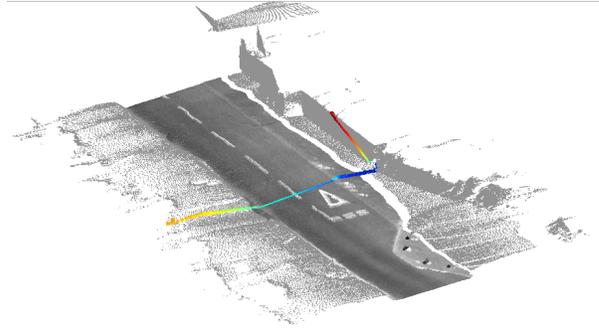


Fig. 2: A perspective view of a single scan of the declined 2D laser (in laser-light) as it moves through a section of Begbroke and produces a swath (motion is upper-left to lower-right). The high-lighted scan (colored by height) clearly shows how this orientation captures the relief of the road and surrounding verges of the vehicle. The swath has been colored using the reflectance data gathered by the 2D scans. As expected the highly-reflective (by design) road markings can be clearly seen. This signal will be exploited later in Section V to deduce vehicle velocity and a bias term in the speedometer readings.

this degree of preparation. We assume that it is possible to take the survey vehicle motion estimates and produce a 3D point-cloud that we will term \mathcal{P} .

In Section IV we will describe in detail how a stream of 2D laser data can be used to deduce vehicle motion. Before that it serves us to illustrate upfront the kind of detail that can be acquired with careful calibration and timing. Figure 2 illustrates how successive declined views can be accumulated to produce a detailed local view of the workspace. Note also the highly-reflective road markings, being clearly visible in laser-light. The figure also highlights the obvious point that a single 2D scan will be insufficient to localize the vehicle - there being an obvious forwards backwards ambiguity despite the marked curbside relief. We will need to build a swath of laser light which captures sufficient detail to anchor ourselves in the survey experience.

IV. LOCALIZATION

Figure 3 shows the overhead view of the 3D point-cloud \mathcal{P} developed during the experience (overlaid on Begbroke), alongside an exemplar swath, \mathcal{Q} , produced by developing the motion of the vehicle over a retrospective window $[t_k, t_{k-N}]$ (where N is the window length, in seconds). The tracking problem is - given the point-cloud \mathcal{P} , and the swath developed during runtime, \mathcal{Q} - to establish a transformation \mathcal{T} that best aligns the clouds. But how do we generate \mathcal{Q} if we do not know the vehicle motion?

1) *Swath generation*: In order to generate the swath \mathcal{Q} at runtime, we need to be able to reconstruct the motion of the vehicle over the windowing period. We define a laser-scan at time t to be:

$$\mathbf{s}(t) = \{r_1, \dots, r_{541}, i_1 \dots i_{541}\} \quad (1)$$

where r_n denotes the laser range reading (in meters) for beam n of scan $\mathbf{s}(t)$, i_n is the intensity of the beam, and $\mathcal{S} = \{\mathbf{s}(1), \dots, \mathbf{s}(n)\}$ is a collection of such scans. In addition we observe, through the gyroscope, the rotational velocities $\mathbf{w}(t)$ around the three ordinate axes of the vehicle (roll, pitch,

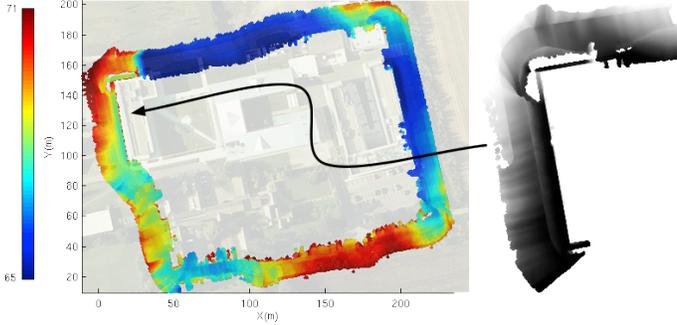


Fig. 3: Overhead view of the 3D point-cloud \mathcal{P} developed during the original experience (left), overlaid on the Begbroke site. Points are colored by height, blue (darker tones) being lowest, and red (lighter tones) highest. Shown on the right is an example swathe, \mathcal{Q} . Note the different color map, emphasizing the relative nature of \mathcal{Q} . Our task is to determine the transformation that best aligns \mathcal{Q} with \mathcal{P} . Note that the scales across these images differ for the purposes of illustration, and must be viewed in color.

yaw). Given the rotational velocities and the forward velocity of the vehicle, we can integrate the state equation:

$$\dot{\mathbf{x}}(t) = v(t) \begin{bmatrix} \cos(\int_{t_0}^t \omega_z(t) dt) \\ \sin(\int_{t_0}^t \omega_z(t) dt) \\ \sin(\int_{t_0}^t \omega_x(t) dt) \end{bmatrix} \quad (2)$$

to produce the *Special Euclidean* ($\mathbb{SE}2$) pose $\mathbf{x}(t)$ (consisting of Cartesian position, and orientation). We can then project the laser data points $\mathcal{S}(t)$ from $\mathbf{x}(t)$, thereby generating the swathe \mathcal{Q} . This task is in principle easy, but in practice details dominate. In particular one has to ensure very accurate time synchronization between gyro, and laser - this is something we discuss in Section V-A. Furthermore note we have a dependence on vehicle velocity v - the determination of this is also a focus of this paper. Putting aside these issues for now, the transformation that best aligns \mathcal{Q} with \mathcal{P} will be the current pose, $\mathbf{x}(t)$. We must now define “best”.

2) *Point-cloud alignment*: Once we have developed the swathe over the window period, it is now necessary to solve for the alignment with the survey point-cloud \mathcal{P} . We seek the transformation $\hat{\mathcal{T}}$ that brings the point-clouds \mathcal{P} and \mathcal{Q} into optimal alignment by minimizing an objective function f :

$$\hat{\mathcal{T}} = \underset{\mathcal{T}}{\operatorname{argmin}} f(\mathcal{P}, \omega, v, \mathcal{T}) \quad (3)$$

The swathe, referenced within the survey, is - as we have seen - a function of angular rotation rate profiles, velocity profile, laser scans and the $\mathbb{SE}2$ transformation we seek. Its generation can be factored as

$$\mathcal{Q} \mapsto g(\omega, v, \mathcal{S}) \cdot \mathcal{T} \quad (4)$$

where Equation 4 develops the swathe point-cloud \mathcal{Q} , and applies the transformation \mathcal{T} to project it into the global frame. Both \mathcal{P} and \mathcal{Q} are distributions of points in space. The Kullback-Leibler divergence offers a natural way to compare two such distributions and forms the basis of our objective function f . We define the operator $\mathcal{H}(\cdot)$ to represent the

histogramming function, and - applying this operation to our survey and hypothesized swathe (a function of \mathcal{T}) - the KL-divergence is:

$$f(\mathcal{P}, \mathcal{Q}) = \sum_{i=1}^N \mathcal{H}(\mathcal{Q})(i) \log \frac{\mathcal{H}(\mathcal{Q})(i)}{\mathcal{H}(\mathcal{P})(i)} \quad (5)$$

where N is the cardinality of the distribution and i is an iterator over bins. Note that if we only require a translation and rotation, then we can simply project points down into the global XY plane in order to generate the histograms.

Representing the point-clouds by their projected probability distributions is a natural way of capturing the structure of the environment. Prismatic structures, such as walls, are represented by a higher density as they are observed more often. Points on the ground have a naturally lower density, but are still informative about the shape and structure of the road. Using this approach, the KL divergence is a natural way of representing the distance between these distributions, provided that they are well defined - and this is a detail that we now must consider.

To prevent infinite divergences we apply absolute discounting to the probability distributions as follows. For any two probability distributions \mathbf{P} and \mathbf{Q} obtained by counts, with the sets of non-zero bins defined as $S_{\mathbf{P}}$ and $S_{\mathbf{Q}}$ respectively, we define the smoothed probability distribution \mathbf{P} to be:

$$\mathbf{P}(i) = \begin{cases} \mathbf{P}(i) - \epsilon & \text{if } i \in S_{\mathbf{P}} \\ \epsilon & \text{otherwise for } i \in S_{\mathbf{P}} \setminus S_{\mathbf{Q}} \end{cases} \quad (6)$$

Absolute discounting reduces the probability mass in distribution \mathbf{P} in all the non-zero bins (Equation 6), and this mass is reapportioned into bins that have mass under \mathbf{Q} but not \mathbf{P} (which is a set difference, denoted by operator \setminus in Equation 7). This is necessary in order for the divergence measure to be properly defined. As a final step, driven by our experience in optimizing this objective function, we apply a discrete Gaussian convolution to both \mathbf{P} and \mathbf{Q} during the optimization procedure:

$$\begin{aligned} \mathbf{P}'(x, y) &= \mathbf{P}(x, y) \star g(x, y) \\ &= \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} \mathbf{P}(n_1, n_2) \cdot g(x - n_1, y - n_2) \end{aligned} \quad (9)$$

and similarly for \mathbf{Q} , where $g(x, y) = \mathcal{N}(0, \sigma)$. This operation contributes significantly to the smooth nature of the cost function (discussed in Section VI).

Algorithm 1 details the described optimization procedure. The algorithm takes, as input, the survey experience point-cloud data, a candidate swathe and an initial estimate of the desired transformation \mathcal{T}_{guess} (an $\mathbb{SE}2$ pose).

The histogram granularity, c , is initialized to a default value, and the halting criterion for the optimization set to infinity. Line 7 makes use of the histogramming operation to produce a discrete probability density function (pdf) of the input point-cloud data with the number of bins determined by the second parameter. Line 11 defines the cost function used in the optimization procedure. This function takes as input two probability distributions, and returns the KL-divergence between them.

Algorithm 1 Cost function optimization

```

1: procedure ESTIMATETRANSFORMATION( $\mathcal{P}, \mathcal{Q}, \mathcal{T}_{guess}$ )
2:    $c \leftarrow c_{init}$ 
3:    $\delta \leftarrow \infty$ 
4:    $\mathcal{T} \leftarrow \mathcal{T}_{guess}$ 
5:   while  $\delta > \text{TOL}$  do
6:      $cost_{min} \leftarrow \infty$ 
7:      $\mathbf{P} \leftarrow \mathcal{H}(\mathcal{P}, c)$ 
8:      $\mathbf{P}' \leftarrow \mathbf{P} \star \mathcal{N}(0, \sigma)$ 
9:      $\mathcal{F} \leftarrow \text{BuildHistogramFilter}(\mathcal{T})$ 
10:    for  $\hat{\mathcal{T}}$  in  $\mathcal{F}$  do
11:       $cost \leftarrow \text{KL}(\mathcal{H}(\text{Transform}(\mathcal{Q}, \hat{\mathcal{T}}), c), \mathbf{P})$ 
12:      if  $cost < cost_{min}$  then
13:         $\delta \leftarrow \|\hat{\mathcal{T}} - \mathcal{T}\|$ 
14:         $\mathcal{T} \leftarrow \hat{\mathcal{T}}$ 
15:      end if
16:    end for
17:     $c \leftarrow c + c_{delta}$ 
18:  end while
19:  return( $\mathcal{T}$ )
20: end procedure

```

The granularity is increased by a quantity c_{delta} at every iteration (Line 17), to provide an annealing effect. The halting measure, δ , is the difference between the previous SE2 estimate and the current estimate, and the optimization halts once this measure has reached a predefined value. The difference between the two poses is measured as given by the metric in [11], in which the orientation in a SE2 pose is expressed with a complex number representation:

$$\mathcal{T} \rightarrow (x_t, y_t, a, b) \in \mathbb{R}^4 \quad (10)$$

where a and b are the complex components of the angle. The euclidean metric is now valid for comparing two poses in SE2.

3) *Pose prediction*: At the next discrete interval, we will have observed more rotational and linear velocity data, and require a pose seed for Algorithm 1 to initiate the search procedure. To obtain this estimate, we apply the velocities, through the system state equations, to $\mathbf{x}(t_{k-1})$ which we then use as a pose guess.

V. CALIBRATION ISSUES

A. Timing

Of vital importance is the timing calibration between the time as perceived by the clocks on-board the gyroscope, the laser sensor, and the speedometer. Disagreement between these clocks will result in point-clouds that exhibit “smear”. Thinking of the information content of \mathbf{Q} , this smearing or blurring will flatten the objective function making optimization harder.

We use the TICSynC [12] timing algorithm, which learns a probabilistic mapping between clocks, and is able to recover both skew and offset. Shown in Figure 4 are the TICSynC-corrected data (left), and the same data with a normally distributed 50ms error in timing (right). Visible in the right-hand figure is the ghosting in laser-space that increases with increased timing error.

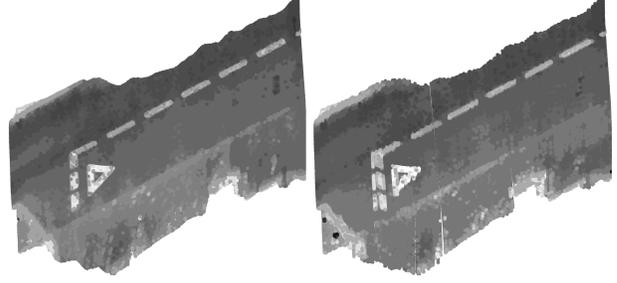


Fig. 4: Overhead view of a set of road markings around the Begbroke site. The point-cloud generated with TICSynC-corrected timings is shown on the left, as compared against a similar point-cloud with normally-distributed 50ms error. The corrected point-cloud does not exhibit the ghosting (visible in the right image).

B. Speedometer bias: offline calibration

Up until this point we have assumed that all of the terms on the right hand side of Equation 4 are known - $v(t)$ in particular being taken from the onboard vehicle speedometer. In general, vehicle speedometers are calibrated to overestimate the vehicle speed, and we will need to compensate for a constant bias term. As such, we will discuss a calibration process, in which we try to recover the bias factor from the observed velocity data. Immediately we reach for our already developed cost function and make this a function of unknown velocity bias. In this process, we will estimate this factor by scaling the observed velocities over a test section and generating correspondingly scaled candidate point-clouds \mathcal{Q} . Our calibration task is then to find the transformation $\hat{\mathcal{T}}$ and bias $\hat{\tau}$ that will produce the best alignment with the reference cloud \mathcal{P} :

$$\hat{\mathcal{T}}, \hat{\tau} = \underset{\mathcal{T}, \tau}{\operatorname{argmin}} f(\mathcal{P}, \omega, v, \mathcal{T}, \tau) \quad (11)$$

Figure 5 shows the results of this process:

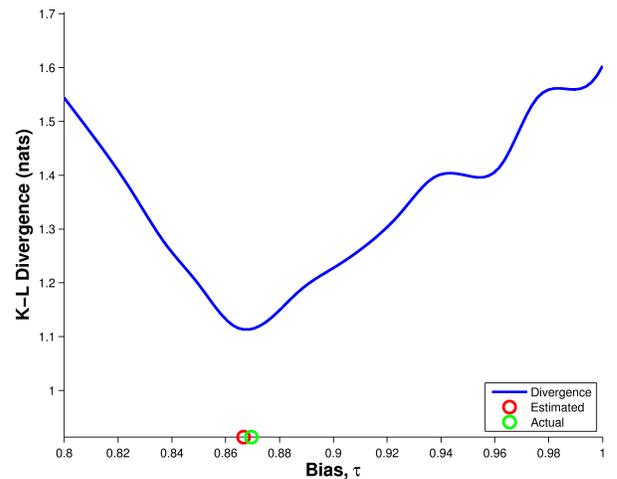


Fig. 5: An illustration of the speedometer bias calibration results. We apply varying correction factors, τ , to the observed velocities - and hence swathe sizes - over a training period, and record the minimum cost of the best pose match found for each factor. We use the best estimate over all bias values to correct the speedometer bias.

We intentionally corrupt ground truth velocities with a known bias, and use the above procedure to estimate the correcting factor. For the results presented in this paper, the true bias was set at 1.15, and therefore the value required to compensate for this over-estimate is $\sim .87$. We use the estimated correction factor of .865 in our analysis, and show that despite the resulting velocity error we can still track successfully over 26 kilometers of data. Note that this calibration technique only works when driving at a constant velocity, which is somewhat prescriptive.

C. Speedometer bias: online calibration

There is another alternative to the approach just described - one which, in the long term, might help us do away with requiring a velocity feed at all and in the short term dispenses with the need to drive at a constant speed. Furthermore it offers the opportunity to refine the bias calibration continually over the lifetime of the vehicle. The idea is that we can infer the velocity of the vehicle from the passage of environmental cues and hence form our bias estimate. We are fortunate here in that roads are replete with features specifically designed to be highly visible. As can be seen in Figure 6 - an intensity plot of the per-beam laser reflectances - the road-markings are highly-visible features in laser light. Given the repeatability of these features, and the fact that they tend to occur in similar regions in beam space (angular position in the scan), we can easily track the signal over time.

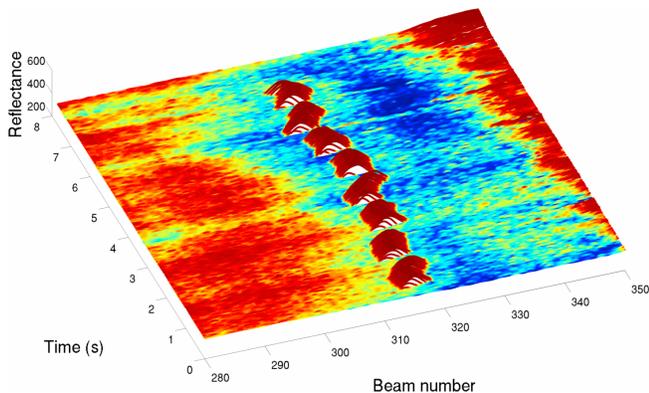


Fig. 6: Per-beam laser reflectances for a subset of the beams along a straight section of Begbroke, plotted against time. This image shows clearly the center-divide marking in the middle section of the road. (Lighter (blue) colors are less reflective, red (darker) are more reflective).

Figure 7 (top) shows a 1D representation of the (normalized) intensity values shown in Figure 6. This signal is observed during the generation of the swathe \mathcal{Q} . Figure 7 (bottom) shows the normalized intensity of center-lane divides in the experience point-cloud \mathcal{P} along a spline model of the centerline of the road network of Begbroke. The warping function that aligns these two signals is the velocity of the vehicle.

To extract this warping function, we use Dynamic Time Warping [13], a well-known technique in speech-analysis for comparing audio signals. Given two signals, $\mathbf{x} \in (x_1, \dots, x_N)$ and $\mathbf{y} \in (y_1, \dots, y_N)$, the distortion between the signals $\mathcal{D}(\mathbf{x}, \mathbf{y})$ is based on a sum of local distances

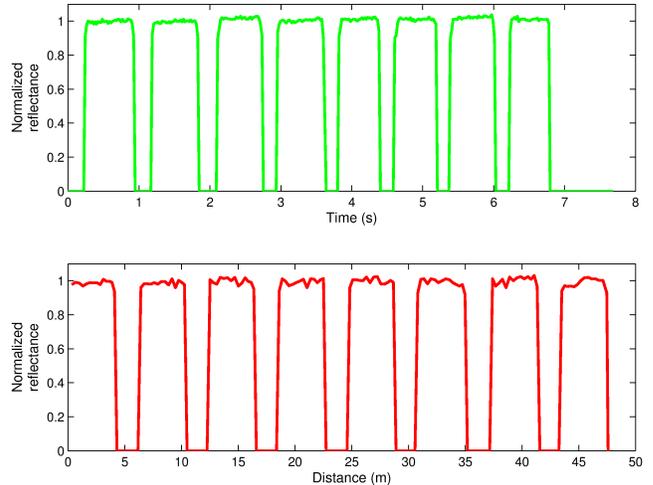


Fig. 7: Normalized reflectance values of the road markings during the swathe generation in the time domain (top), and from a section of a spline model of the road-network around Begbroke (bottom). The warping function that aligns these two signals is the velocity of the vehicle.

between elements $d(x_i, y_i)$, with the optimal alignment minimizing the distortion. This is posed as a minimization problem:

$$\mathcal{D}_{\Phi}(\mathbf{x}, \mathbf{y}) = \min_{\Phi} \frac{1}{K_{\Phi}} \sum_{k=1}^{K_{\Phi}} d(\mathbf{x}_{\Phi_t}(k), \mathbf{y}_{\Phi_r}(k)) \quad (12)$$

that is solved with a dynamic-programming approach. K_{Φ} represents the warping distance of path Φ . Having estimated the distortion, we can align the signals, determining the timing correspondence over the lane markers, hence inferring the velocity. Figure 8 shows the results of applying this procedure over the data shown in Figure 6.

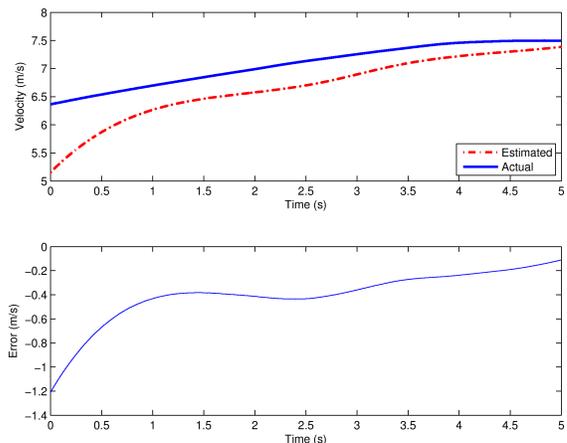


Fig. 8: The estimated and true velocities over the straight section of road shown in Figure 6, using the described warping method. The ground-truth velocity is shown in the top image in (solid) blue, with the estimated velocity in (dotted) red. Even with these relatively coarse markers, the estimated velocity tracks the true velocity. The lower figure illustrates the error in the velocity estimation over this swathe.

Sadly due to the current modest declination of the laser, it is not feasible to use this approach to infer velocity around areas of high curvature - the tracking becomes troublesome. Current work is addressing this problem by declining the laser even further, maintaining all the properties described so far, in addition to allowing for continuous velocity updates thus removing a dependence on speedometer data. However, as it stands we are able to demonstrate quantifiably good localization with just a single push-broom laser. The next section will present these results.

VI. RESULTS

A. Trajectory analysis

In order to evaluate the performance of the localization procedure, we require some method of comparing the resultant trajectories with those of the INS system (which itself can be in substantial error as shown in Figure 1). We define this as a displacement function, which for pose $\mathbf{x}(t)$ is defined to be:

$$\delta(\mathbf{x}(t)) = \|\mathbf{x}(t) - \hat{\mathbf{x}}_e\| \quad (13)$$

where \mathbf{x}_e is the closest pose of the trajectory in the survey loop, as measured by the metric in [11]. This displacement will capture both the true deviation from trajectory to trajectory in addition to the localization error. If we had traversed exactly the same route as the survey vehicle, we would expect this displacement to be zero. As we never traverse the exact same route twice, we expect - in a well behaved system - the displacement to be well bounded.

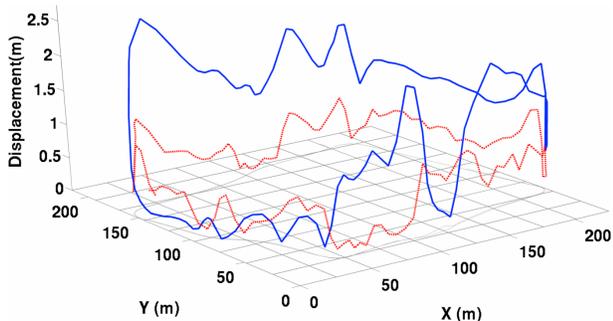


Fig. 9: Relative displacement of both the localization procedure (shown in red, dotted) and the INS (blue, solid) as measured against the experience data over the Begbroke site. The INS exhibits a high displacement in areas of poor GPS signal quality.

Figure 9 depicts this displacement function for the INS and the localization procedure for the same loop, as measured against the reference trajectory. As we can see from the figure, the displacement for both the INS and the localization procedure are bounded - however, the average displacement over the trajectory is substantially lower for our method, as compared to the INS. The areas in which the INS exhibits a large deviation (particularly the northern and southern sections) are results of the paucity of GPS signal in those regions. However, the displacement of our localization procedure is relatively constant over the site. To validate our technique, we tested the performance of the algorithm over 26 kilometers of trajectory data, spanning a period of three months. These datasets were taken at various times of the

day, ranging from early morning (9 a.m) to early evening (7 p.m).

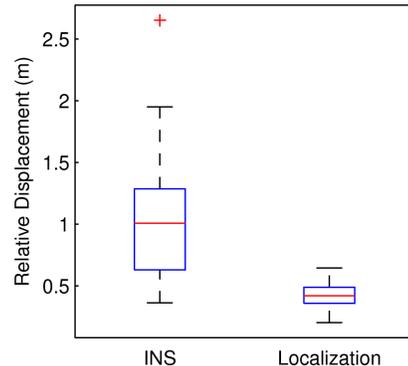


Fig. 10: A comparison of the mean displacement from the experience map for both the INS and the described localization procedure over 26 km worth of trajectory data spanning three months. The boxplot shows the median, 25th and 75th percentiles, with outliers plotted individually as points. As can be seen from the figure, the standard deviation of this displacement is substantially lower for our method.

Figure 10 shows a comparison of the mean displacement per trajectory, collected over this data. Depicted are the median, 25th and 75th percentiles of the mean displacement per trajectory, for all the trajectories considered. Not only is the median displacement distinctly lower for our method, but outliers have been eliminated and the variance of the displacement substantially reduced. This is compelling evidence that localization using our method is far more repeatable over the long term than relying on the estimates from the INS.

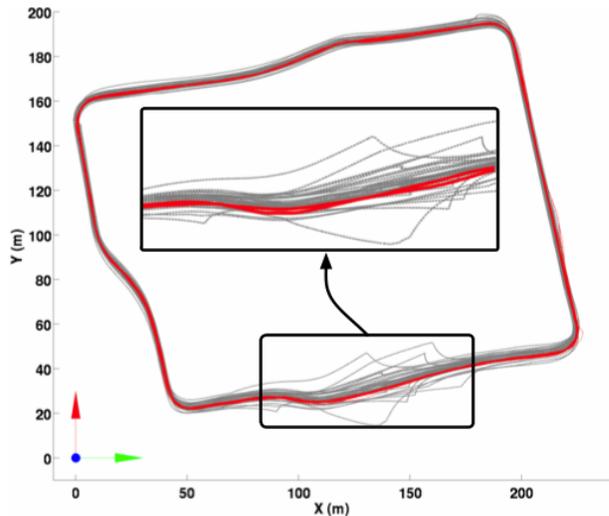


Fig. 11: A comparison of the original trajectories (gray, dotted) vs. the corrected trajectories (red, solid). As can be seen from the image, the INS data drifts substantially in areas of poor reception, which is avoided by using the described method (must be viewed in color).

Figure 11 contrasts the trajectories from the INS system (shown in gray) and the trajectories obtained from the localization procedure. Examining the INS trajectories, we can see a distinct wander over time. As expected, the localized trajectories are more consistent of all the datasets, and do

not exhibit the gross errors that are clearly visible in the trajectories from the INS.

B. Cost-function degeneracy

The KL-divergence typically produces a cost-function with a smooth gradient, as can be seen for example in Figure 12. However it also shows the effect of aliasing - the “corridor” problem, in which the valley in the figure lies along the forward trajectory of the vehicle.

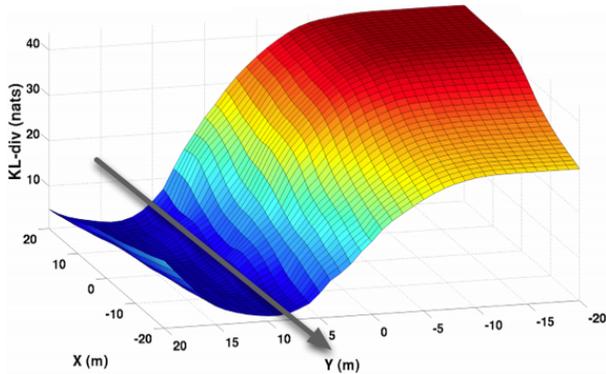


Fig. 12: An exemplar cost surface using the KL divergence between the swathe \mathcal{Q} and the experience map \mathcal{P} . Visible is the “corridor” problem along the forward trajectory of the vehicle (indicated by the arrow).

In current work we are looking at ways in which we can detect (via the appearance of the swathe) such a degeneracy but in the 26 km of data used here it did not cause a localization failure - the issue is not critical.

VII. DISCUSSION

We again highlight the performance of the system in the GPS-degenerate southern section of the test site. Although GPS-integrated INS systems are the current state-of-the-art in terms of outdoor navigation, these systems cannot operate for extended periods in GPS-denied areas without experiencing drift, as is shown in Figure 11. Use of a prior survey and swathe-matching using the proposed framework allows for robust long-term navigation, irrespective of GPS attenuation.

As the test site is a closed course, relatively little traffic was experienced during data collection. However, fluctuations in the number of vehicles in the car-parks (located in the west and south-east) present a constantly changing scenario, but were aptly dealt with by the proposed framework.

Although the current implementation does not run in real-time, there are no impediments to producing an implementation that does - the sliding window property requires an **initialization** period, however the algorithm is *constant time* ($O(1)$). The overall complexity of the algorithm is determined by point query in the prior map, which is (as a function of map size n and requested points k) $O(\sqrt{n} + k)$.

VIII. CONCLUSION

To conclude, we have presented a push-broom laser localization system which leverages a prior 3D survey for use on road vehicles. We demonstrated and evaluated its performance on 26 kilometers of real-world data spanning a

3 month period. We believe the work presented here can be extended to reduce dependence on angular and linear motion rate measurements. However as it stands, the system is an indication of what you can do without a Velodyne.

IX. ACKNOWLEDGMENTS

This work was funded by the Office of Naval Research (ONR) Grant N00140810337, and EPSRC Grant EP/I005021/.

REFERENCES

- [1] J. Levinson, M. Montemerlo, and S. Thrun, “Map-based precision vehicle localization in urban environments,” in *Proceedings of the Robotics: Science and Systems Conference*, 2007.
- [2] J. Levinson and S. Thrun, “Robust vehicle localization in urban environments using probabilistic maps,” in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 4372–4378.
- [3] R. Kummerle, D. Hahnel, D. Dolgov, S. Thrun, and W. Burgard, “Autonomous driving in a multi-level parking structure,” in *International Conference on Robotics and Automation, 2009*. IEEE, 2009, pp. 3395–3400.
- [4] F. Moosmann and C. Stiller, “Velodyne slam,” in *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, pp. 393–398.
- [5] Y. Morales, T. Tsubouchi, and S. Yuta, “Vehicle localization in outdoor mountainous forested paths and extension of two-dimensional road centerline maps to three-dimensional maps,” *Advanced Robotics*, vol. 24, no. 4, pp. 489–513, 2010.
- [6] M. Bosse and R. Zlot, “Map matching and data association for large-scale two-dimensional laser scan-based slam,” *The International Journal of Robotics Research*, vol. 27, no. 6, p. 667, 2008.
- [7] P. Besl and N. McKay, “A method for registration of 3-d shapes,” *IEEE Transactions on pattern analysis and machine intelligence*, pp. 239–256, 1992.
- [8] M. Bosse and R. Zlot, “Continuous 3d scan-matching with a spinning 2d laser,” in *Robotics and Automation, 2009. ICRA’09. IEEE International Conference on*. IEEE, 2009, pp. 4312–4319.
- [9] W. Carter, R. Shrestha, G. Tuell, D. Bloomquist, and M. Sartori, “Airborne laser swath mapping shines new light on earth’s topography,” *EOS, Transactions American Geophysical Union*, vol. 82, no. 46, pp. 549–555, 2001.
- [10] P. Furgale and T. Barfoot, “Visual teach and repeat for long-range rover autonomy,” *Journal of Field Robotics*, vol. 27, no. 5, pp. 534–560, 2010.
- [11] S. LaValle, *Planning algorithms*. Cambridge Univ Pr, 2006, ch. 5: Sampling-based Motion Planning, p. 189.
- [12] A. Harrison and P. Newman, “Ticsync: Knowing when things happened,” in *Proc. IEEE International Conference on Robotics and Automation (ICRA2011)*, Shanghai, China, May 2011, 05.
- [13] H. Strik and L. Boves, “Averaging physiological signals with the use of a DTW algorithm,” *Proceedings SPEECH*, vol. 88, no. 7, pp. 883–890, 1988.