

Modelling Scene Change for Large-Scale Long Term Laser Localisation

Dan Withers and Paul Newman

Abstract—This paper addresses a difficulty in large-scale long term laser localisation - how to deal with scene change. We pose this as a distraction suppression problem. Urban driving environments are frequently subject to large dynamic outliers, such as buses, trucks etc. These objects can mask the static elements of the prior map that we rely on for localisation. At the same time some objects change shape in a way that is less dramatic but equally pernicious during localisation - for example trees over seasons and in wind, shop fronts and doorways. In this paper, we show how we can learn in high resolution, the areas of our map that are subject to such distractions (low value data) in a place-dependent approach. We demonstrate how to utilise this model to select individual laser measurements for localisation. Specifically, by leveraging repeated operation over weeks and months, for each point in our map pointcloud we build distributions of the errors associated with that point for multiple localisation passes. These distributions are then used to determine the legitimacy of laser measurements prior to their use in localisation. We demonstrate distraction suppression as a front-end process to large scale localisation by incrementally adding 50km of error data to our base map and show that robustness is improved over the base system with a further 10km of urban driving.

I. INTRODUCTION

This paper addresses the problem of laser based localisation for an autonomous vehicle operating in typical outdoor environments where things change. It is not about a localisation technique per se, that is well understood, but about how to build an objective function *within* the localiser which is sympathetic to the time varying nature of the scene. Ours like the majority of laser based localisation approaches require a prior pointcloud map. These maps are often built by fusing the measurements of one or more laser scanners with estimates of the vehicle’s ego-motion [1] [2]. The resulting pointcloud is a snapshot of the environment at the time of survey. The instantaneous nature of these maps invariably results in dynamic objects being captured: cars, buses, people etc as in Figure 1. Equally, the map is also likely to contain elements of foliage whose shape is seasonal. More radical change is seen when buildings are demolished, extended or shrouded and this happens surprisingly often in a city.

Metric localisation involves finding a pose that aligns a description of the local environment with a prior map. In our case, the local description is generated from the range measurements of a LIDAR. The vehicle used in this paper is equipped with a 2D LIDAR in a pushbroom configuration. Aligning a single scan from one of these 2D sensors to a 3D map is often an under constrained problem. For this reason, we project a cache of scans with the motion estimates from

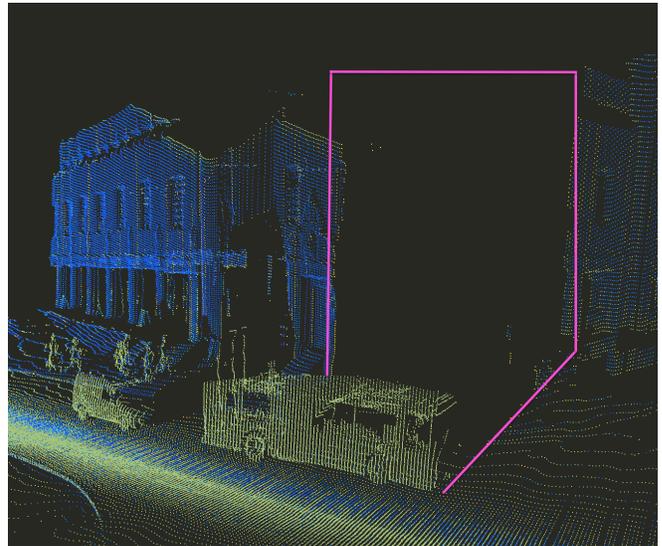


Fig. 1: In urban environments large dynamic obstacles frequently occlude the static environment from an autonomous vehicle’s sensors. Here a bus casts a shadow more than 10m wide onto the corner of a building (hole highlighted in pink). What will we do when the bus isn’t there any more? Measurements of the wall used at subsequent localisation will be poorly explained by the map and distort localisation as in Figure 2. A drastic enough scene change can result in catastrophic localisation failure.

odometry into a single coordinate frame. Thereby producing a less ambiguous 3D “swathe” of laser measurements and replicating the pointclouds produced by 3D sensors.

In Section III the method of alignment used for the local description of the environment (herein denoted swathe) with the map pointcloud is a vanilla derivative Iterative Closest Point (ICP) algorithm [3]. This algorithm finds the transform that minimises the sum of squared errors between corresponding points of two pointclouds. The correspondences for each swathe point are chosen by finding the nearest neighbour in the map pointcloud. Our implementation of the algorithm allows the error (or cost) and corresponding map point of each swathe point to be queried and logged for offline processing.

In Section IV we show how post-processing localisation data can help to augment the map with distraction suppression data - a simple (therefore fast to use) point by point model which captures the expected value of that point in subsequent localisation operations. Having successfully localised against a map we can aggregate all the error-

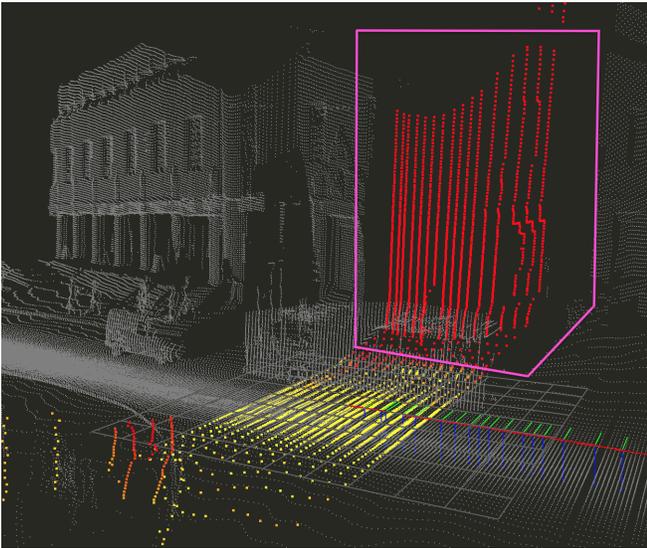


Fig. 2: Subsequent localisation with the map shown in Figure 1 results in measurements of the building landing in the bus’s shadow (highlighted in pink). These measurements have high error as shown by the red colour, in contrast to low error yellow measurements of the road. Post-processing of localisation data can identify areas susceptible to these outliers. Allowing the rejection of these measurements in subsequent localisation in order to improve accuracy and speed.

correspondence pairs that occurred per map point. For each map point, a distribution of the errors is built. More error samples are added after each localisation run.

In Section V explains that by inspecting the distribution of errors we are able to label map points which have been subject to high cost swathe points as being of low value to localisation. By further aggregating these point-specific distributions over all the points in a section of the map we are able to infer the likely errors in a place dependent manner and tailor the distraction suppression. In subsequent localisation, swathe points that associate with map points labelled as untrustworthy can be discarded on the assumption that they are outliers.

Finally, in Section VI we show that increasing the number of datasets used to estimate the error distributions improves localisation robustness with only a minor increase in the time taken for localisation.

II. RELATED WORK

The use of spinning 3D LIDAR sensors is common in the autonomous driving field. These sensors are often expensive but offer a 360° view around the robot and a rich local description. Levinson and Thrun [4] and Levinson et al. [5] use a Velodyne sensor mounted on the roof of a car and localise by aligning reflectance information in the local scan with a map. Whereas, Baldwin and Newman [6] [7] use a pushbroom LIDAR to build a 3D swathe description of the local environment. The points in this pointcloud are binned into a histogram on the ground plane. The variable density

of the histogram corresponds to multiple measurements of a wall falling in to the same bin. The swathe and map histograms are considered to be probability density functions and are aligned by minimising the KL divergence between the two. Maddern et al. [2] also build a swathe from a pushbroom LIDAR. They discretise the points and match the swathe and map by aligning histograms of the point heights. Wolcott and Eustice [8] also use the distribution of measurement heights. They build a Gaussian mixture map of height distributions which are used for scan matching with the measurements from a 3D spinning LIDAR.

Rusinkiewicz and Levoy [9] evaluate the convergence properties for the ICP algorithm with various point selection techniques. They show that using points uniformly distributed according to their estimated normal is superior to other selection techniques in most scenarios: using all points as in the original formulation [3], using a uniformly sampled subset of the points [10], randomly sampling points at each iteration of the algorithm [11] and sampling points based on intensity gradient where available [12].

Rather than rely on precise data association, Sunderhauf and Protzel [13] use switchable constraints in their SLAM formulation to turn possible outliers off from within the optimisation. Biber and Duckett [14] present dynamic maps that handle temporal change by storing multiple maps at different timescales. At localisation the map that best explains the sensor data is used for scan matching. After each localisation iteration local short term maps are updated online, whilst long term maps are updated offline. McManus et al. [15] use distraction suppression, to ignore dynamic objects in stereo images, to improve visual odometry. They do this by masking parts of the image deemed to be from dynamic objects. First a 3D pointcloud map is produced. The map is presumed to be static. At a known camera pose, the pointcloud can be projected into the image to produce a depth image. This depth image is compared with a depth image calculated from images from a stereo camera. Disagreement between the two depth maps is considered to be result of non-static objects in the live images. Pixels in disagreement are subsequently marked as distractions and not used in the visual odometry pipeline.

III. LOCALISATION

A. Aligning the swathe and map pointclouds

We frame localisation as a geometric alignment problem, where the pose of the robot minimises the sum of the squared alignment errors between the swathe pointcloud S and the map pointcloud M as in Equation 1. The function $g(\mathbf{x}, \mathbf{s}_i)$ transforms the swathe point \mathbf{s}_i with the SE(3) transform characterised by \mathbf{x} and produces the point \mathbf{q}_i which is referenced to the map co-ordinate frame. $f(\mathbf{q}_i, M)$ finds the distance from the transformed swathe point \mathbf{q}_i to the nearest point in the map pointcloud M , which is treated as the alignment error for the swathe point. Minimising the sum of the square of these errors is the typical formulation of ICP

and can be solved using gradient based optimisation.¹ A well aligned swathe and map can be seen in Figure 3.

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} \sum_{s_i \in S} f(g(\mathbf{x}, \mathbf{s}_i), M)^2 \quad (1)$$

B. Problem robustness

As we expect our swathe pointcloud to include measurements from outliers that do not feature in our map pointcloud and that are subject to measurement noise, we add a Huber kernel loss function $\rho(x)$ to each point for increased robustness [17]. Note that the objective function is robustified by the Huber kernel $\rho(x)$ and this is able to defend against some outliers in the solve. However, it is not sufficient for the drastic spatial change seen in examples such as Figure 4. The objective function contains a weight w_i that corresponds to the probability that s_i is an outlier. The selection of this weight is the purpose of our distraction suppression system and is detailed in Section V. Our new robust objective function is given in Equation 2.

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} \sum_{s_i \in S} w_i \rho(f(g(\mathbf{x}, \mathbf{s}_i), M)^2) \quad (2)$$

C. Nearest neighbour search

To find the distance between each swathe point and the map, a k-d tree of the map pointcloud is built. At each iteration of the minimisation problem, the nearest neighbour of each swathe point is found. A k-d tree search is on average an $O(\log n)$ operation. Searching for nearest neighbours is the most time-consuming element of the problem minimisation. For this reason, the fewer swathe points added to the problem and the fewer iterations that have to occur before convergence the quicker localisation can occur.

IV. ESTIMATING ERROR DISTRIBUTIONS ON A POINT BY POINT BASIS

A. Aggregating swathe errors

Logged data contains the correspondences and distances between each swathe point and the map pointcloud in the termination state of the optimisation. First, we filter the logged data to extract non-overlapping swathes, this ensures that each swathe measurement is used just once. The errors of each swathe-map correspondence are then aggregated in a histogram for each map point. The histogram bins correspond to errors for point-to-point separation of 0-10cm, 10-20cm, 20cm-30cm, 30-40cm, 40-50cm, 50cm+. Further localisation runs update the bin counts of these histograms. This allows us to add many localisation runs without increasing the memory footprint of the error distribution.

¹To solve this non-linear least squares problem we use the Ceres-Solver C++ library [16]

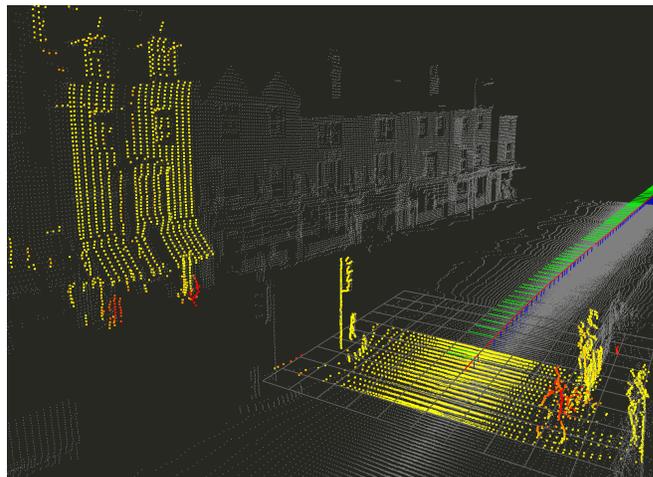


Fig. 3: A localisation event from an urban scene. The map pointcloud can be seen in grey and the swathe pointcloud in colours yellow-orange-red to show increasing cost as dictated by its error according to the map. In this scene we can see good alignment between the static elements of the swathe and map but pedestrians of the scene have high cost. The red, green and blue lines trailing into the distance are coordinate frame axes indicating the robot's previous poses.

B. Bayesian smoothing of error distributions

Each map point now has a histogram of errors attached to it. Some map points will have had no correspondences because they are from dynamic elements that have since left the environment. This is an example of the *zero count problem* or *sparse data problem*[18]. A Bayesian approach says that these map points have some non-zero probability of being used in the future. We apply Bayesian smoothing to the histograms to produce predictive distributions using a Dirichlet-multinomial model. This ensures that every error for a map point has non-zero probability.

The Dirichlet distribution is the multivariate generalisation of the Beta distribution [18]. In our case the dimensionality corresponds to the number of error bins, so is 6. For a map point i we observe N_i errors, each of which corresponds to a bin in the error histogram. The bin counts $\mathcal{D}_i = \{x_1, \dots, x_{N_i}\}$, where $x_i \in \{1, \dots, 6\}$. As the errors are from different swathe measurements we can assume they are iid, so the likelihood is given by the multinomial distribution given in Equation 3, where N_k is the number of times error k occurred.

$$p(\mathcal{D}_i | \boldsymbol{\theta}) \propto \prod_{k=1}^K \theta_k^{N_k} \quad (3)$$

The Dirichlet distribution is conjugate to the multinomial distribution and has support over K-dimension probability simplex. The form of our prior with $K = 6$ is

$$\text{Dir}(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{k=1}^K \theta_k^{\alpha_k-1} \mathbb{I}(\mathbf{x} \in S_K) \quad (4)$$

Due to the Dirichlet distribution being the conjugate prior of the multinomial distribution our prior is also a Dirichlet distribution

$$p(\boldsymbol{\theta}|\mathcal{D}_i) \propto p(\mathcal{D}_i|\boldsymbol{\theta})p(\boldsymbol{\theta}) \quad (5)$$

$$\propto \prod_{k=1}^K \theta_k^{\alpha_k+N_k-1} \quad (6)$$

$$= \text{Dir}(\boldsymbol{\theta}|\alpha_1 + N_1, \dots, \alpha_K + N_K) \quad (7)$$

For a Dirichlet distribution $\text{Dir}(\mathbf{x}|\alpha_1, \dots, \alpha_K)$ the expected value of a value x_k is given by Equation 8.

$$\mathbb{E}(x_k) = \frac{\alpha_k}{\alpha_0} \text{ where } \alpha_0 = \sum_k \alpha_k \quad (8)$$

Considering Equations 7 and 8 the posterior predictive is given by Equation 9.

$$p(X = i|\mathcal{D}) = \mathbb{E}(\theta_j|\mathcal{D}) = \frac{\alpha_i + N_i}{\sum_j \alpha_j + N_j} \quad (9)$$

Initially we assume each map point has errors of equal probability and set the concentration parameters $\alpha_j = 1 \forall j$ the posterior predictive becomes Equation 10, where $K = 6$.

$$p(X = i|\mathcal{D}) = \mathbb{E}(\theta_j|\mathcal{D}) = \frac{1 + N_i}{K + \sum_j N_j} \quad (10)$$

Equation 10 gives a probabilistically rigorous method of generating likelihood distributions for the errors of map points in a simple manner. The low complexity of this technique is attractive when a map pointcloud can contain hundreds of thousands of points. These distributions provide a basis for labelling map points as inliers or outliers.

As well as calculating these likelihood distributions on a per point basis we also calculate a distribution for all points in a local region of the map. In this way, we have a place dependent expectation of point errors.

V. SUPPRESSING DISTRACTIONS

The motivation for suppressing distractions can be seen in Figure 2, which shows outlier measurements from a hole in the map caused by a bus. These high cost outliers not only distort the minimisation problem, sometimes disastrously, but they also slow the solve down by increasing the total number of k-d tree searches. We should remember that each point search in a k-d tree is on average an $O(\log n)$ operation. Ideally we would not add these outliers to the objective function at all.

In Equation 2 we introduced a weight w_i that allows us to model the probability of a swathe point being an outlier. This weight is calculated by finding the map point closest to the swathe point and examining its error distribution. Strictly,

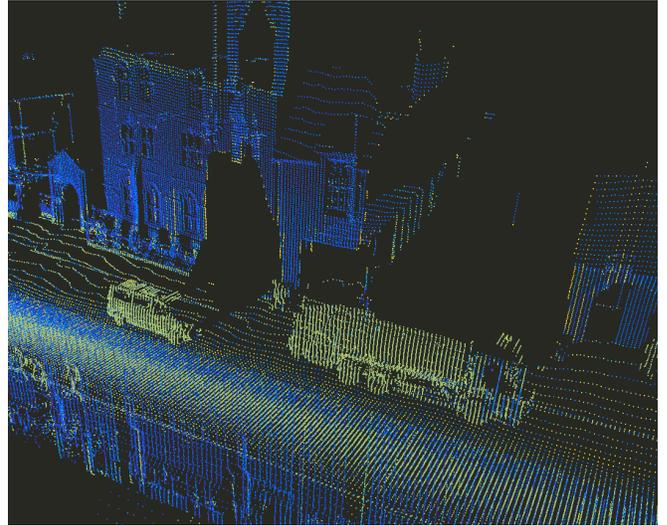


Fig. 4: A section of our map pointcloud that has been localised against with 5 datasets collected over the course of a month. The localisation error data has been aggregated across the datasets and the error distributions have been calculated. The colour of each point indicates the skew of the point’s error distribution. Blue colour indicates a posterior skewed towards low error correspondences and is typical for static elements like the walls. Lime green indicates a uniform posterior and is typical for dynamic elements that have not been seen since the map survey e.g. the van and refuse lorry. Gold indicates a posterior skewed towards high error correspondences and can be seen surrounding the shadows cast by the refuse lorry and the van.

this correspondence between map and swathe points varies as the swathe is transformed via the estimated pose. However, we have found that calculating the weight for each swathe point once per minimisation is preferential, as it drastically reduces the number of required nearest neighbour look-ups and the increased speed counteracts the reduced modelling. As a result, the likelihood of each swathe point being an outlier is calculated prior to optimisation. The swathe is first transformed by the predicted pose. Each point’s nearest neighbour in the map is found using a k-d tree (this adds some computational cost over the baseline system but we will see later that it is an acceptable overhead). We now use the error distributions (estimated in Section IV) to decide the likelihood of a map point’s trustworthiness and, in turn, infer the credibility of the swathe point.

First, for each map point, we calculate the median error $\mu_p = \text{med}(p(X|\mathcal{D}))$ of the posterior predictive in Equation 10. Then we calculate the population median μ_s of the error of all points that *could* be associated to the swathe in a local region. Note therefore that μ_s varies as the vehicle moves. A simple and fast method of calculating the weight w_i for the swathe point \mathbf{s}_i and thereby partitioning it (inlier / outlier) is reached via

$$w_i = p(\mathbf{s}_i = \text{outlier} | \mathbf{p}_j) = \begin{cases} 0.0 & \text{if } \mu_p > \mu_s \\ 1.0 & \text{otherwise} \end{cases} \quad (11)$$

Equation 11 shows how map points are labelled as outliers. Any swathe point \mathbf{s}_i that corresponds to an outlier map point \mathbf{p}_j is rejected from the optimisation by being given weight zero (Equation 2). Figure 4 illustrates the error distributions for a map pointcloud, swathe points corresponding to gold points will be rejected, while swathe points corresponding to blue points will be kept.

VI. EVALUATION

Evaluating the performance of state-of-the-art localisation is non-trivial as ground truth does not exist. Often measurements from GPS/INS systems are used to assess the performance of localisation. However, the reason that local sensing is required for autonomous vehicles is because GPS/INS systems are unreliable and suffer gross error in urban environments. Whilst we believe that the improved robustness that modelling map outliers improves performance everywhere it is easiest to quantify the performance in the areas which were subject to catastrophic error with a baseline system.

A. Characterising localisation error

We characterise localisation failure by large discrepancies between the input and output pose from localisation. This captures the localisation noise that is being passed to an autonomous vehicle’s control system. We are specifically interested in errors that correspond to lateral translation (in our co-ordinate frame this corresponds to y) and errors in yaw angle. These two elements have been chosen as they dominate the ability of the vehicle to follow a path.

In the results presented, we arbitrarily characterise localisation failure as translation jumps of magnitude 10cm or more and rotational jumps corresponding to 1° or more. The following results show that the addition of distraction suppression to our baseline system allows us to localise with much greater robustness for a minor increase in computational overhead.

B. Robotcar 10km datasets

The seven datasets used in this work were collected using the Mobile Robotics Group’s Robotcar platform in Oxford city centre [19]. The data used in these experiments is available for preview (and are now available to download) at <http://robotcar-dataset.robots.ox.ac.uk/>. The particular datasets used are listed in Table I. The data was collected over the course of a month at different times of day in varying weather. One dataset was used to produce the map. Five were then localised against the map to produce error data. This error data was then attached to the map in six layers such that we could test the effect of adding more error data over time. In effect, we end up with six maps that each share a base pointcloud but with increasing amounts of error data appended to it. Map

0 has no error data attached to it so reverts to the baseline system. The map is made up of multiple files, each of which contributes to a total map of 3.9GB for 10km of driving. The remaining dataset was used for localisation with the augmented map error distributions and is used to evaluate the proposed system.

0	2015-06-23-13-06-07	Map
1	2015-06-26-08-09-43	Error data
2	2015-06-30-13-31-45	Error data
3	2015-07-03-15-23-28	Error data
4	2015-07-10-10-01-59	Error data
5	2015-07-14-16-17-39	Error data
6	2015-07-29-13-09-26	Localisation analysis

TABLE I: Datasets used for mapping, error data aggregation and localisation experiments.

C. Results

Table II shows the number of localisation failures for each of the six levels of map augmentation. As the amount of error data attached to the map is increased the number of localisation failures decreases. The map with the most error data attached to it (Map 5) exhibited no localisation failures. As a result the following analysis will use the poses estimated by the localiser with Map 5 as the baseline poses that the other systems will be considered against.

Map Number	Datasets appended	Number of localisation failures
0	-	14
1	1	8
2	1,2	6
3	1,2,3	2
4	1,2,3,4	1
5	1,2,3,4,5	0

TABLE II: Summary of localisation failures for varying levels of error augmentation.

Using the poses estimated by the localiser using distraction suppression with Map 5, we can quantify the effect of adding more error data to the base map. Figure 5 shows the error histograms for the five levels of map augmentation. We can see from the figure that errors are small generally but also that as the amount of error data used is increased that errors condense around zero for translational and rotational components. This is confirmed by Figure 6 which shows the errors are consolidating around zero as the number of localisation runs (datasets) used in the error distribution increases.

Table III shows that the time taken for optimisation increases with swathe length. This is due to the increased number of points being solved for. We can also see that adding distraction suppression system increases the time required for localisation. This is due to the overhead of more nearest neighbour searches in determining outliers. In our domain this overhead is acceptable given the increased localisation robustness afforded by distraction suppression. Adding more error data to the distributions does not increase the memory footprint nor the time taken to label an outlier

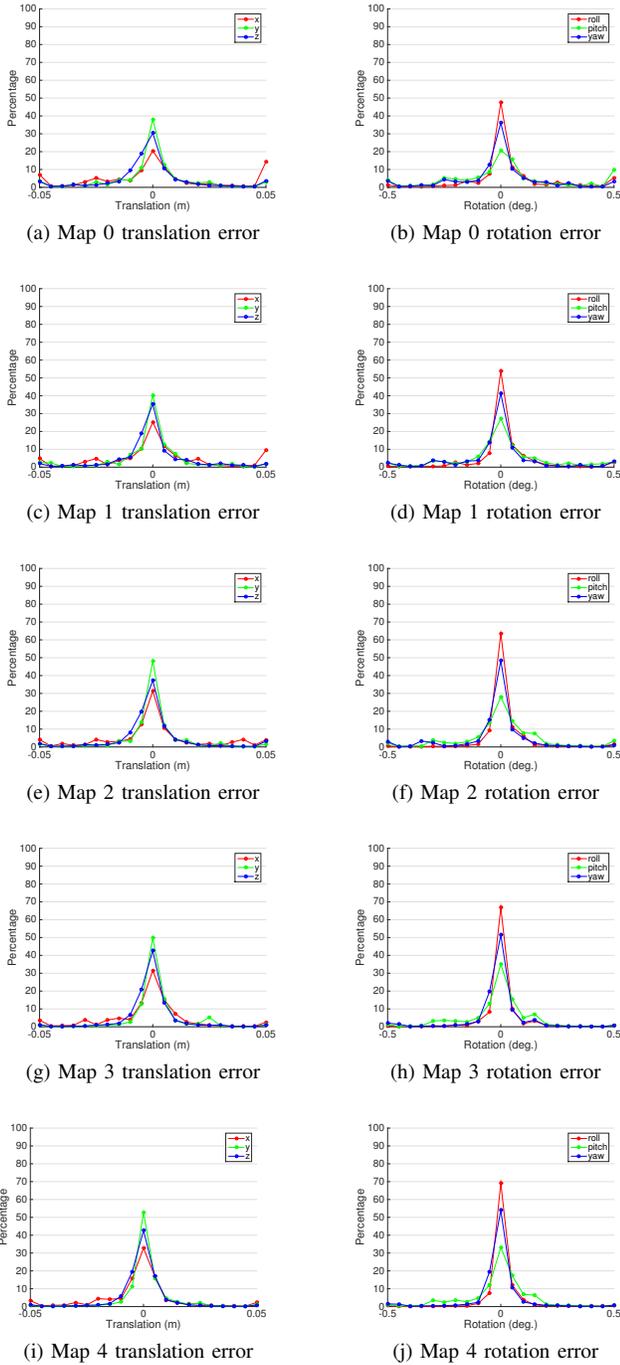


Fig. 5: Histograms of translational and rotational errors for the five maps used. Errors are calculated using the Map 5 poses. We can see that each system has relatively small error but a more-peaked distribution for Map 4 shows the superior agreement. Note we are considering small errors here. Translational errors smaller than 5cm and rotational errors smaller than 0.5 degrees.

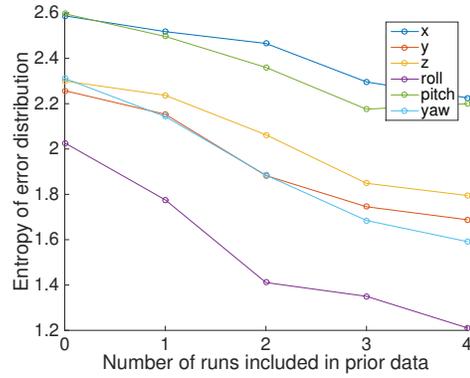


Fig. 6: Entropy of the error histograms shown in Figure 5. The decreasing entropy shows that the errors are becoming more ordered. In this case, it shows that very small errors are more likely as the number of localisation runs used in error aggregation increases.

due to us storing the errors with a histogram. All mapping and experiments were carried out on a 2012 Macbook 2.6 GHz Intel Core i7 laptop.

Swathe length (m)	Mean localisation time (ms)	
	Baseline system	+Distraction Suppression
2	33	50.5
3	36.4	59.6
4	73.8	84.5
5	107.5	116.0

TABLE III: Timing statistics for the baseline system and baseline with distraction suppression enabled. The swathe length is the distance over which laser measurements are cached in the swathe. Simply, the longer the swathe length the more points that a swathe will contain and the greater the computational burden.

Figure 7 provides a descriptive analysis of a catastrophic failure of the base system while the system with distraction suppression is unaffected. The left column corresponds to the baseline system and the right column to distraction suppression enabled system for a swathe of length 3m. Rows correspond to a moment in time. Row 0 (top) shows the vehicle approaching the hole cast by a refuse truck as detailed in Figure 4. Both systems approach well localised. Row 1 shows the vehicle level with the hole. With distraction suppression the measurements of the un-mapped wall correspond to map points identified as outliers and have been rejected (blue colour). The vehicle remains well localised by features on the opposite wall. In the baseline system (left) the measurements of the un-mapped wall are of high cost (see red colour) and have distorted the localisation. Row 2 shows the same instant as Row 1 but from above. In this view we can see that the baseline system has been “twisted” by the red wall points resulting in a lateral and yaw error. Row 3 (bottom) shows the baseline system lost and unable to recover whereas the distraction suppression continues to localise and reject outliers.

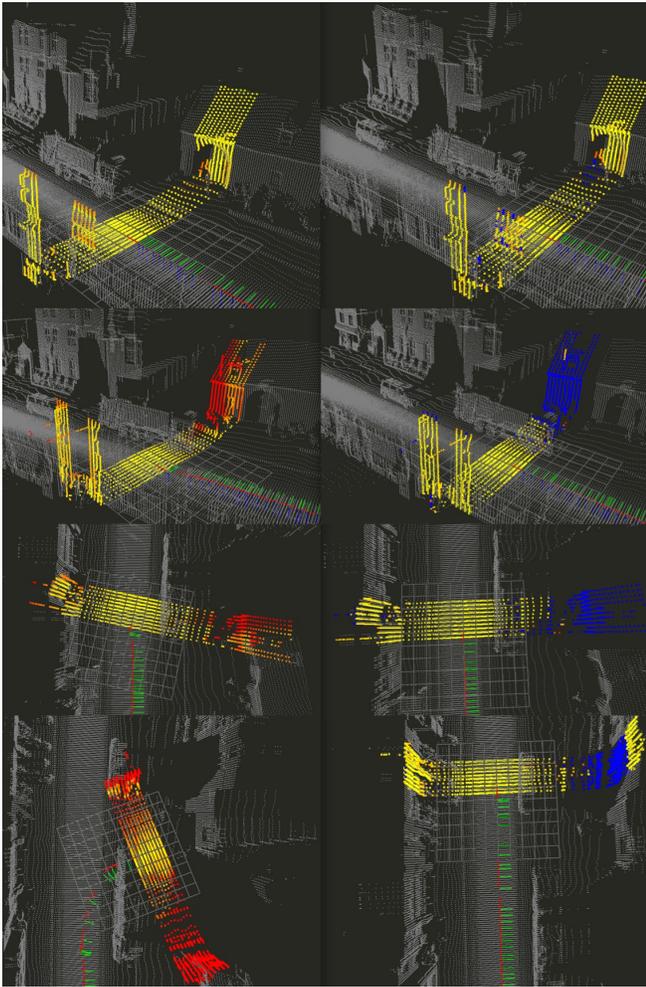


Fig. 7: Matrix of stills from localisation. Grey points show the map pointcloud. Swathe points are yellow-orange-red ranging from low to high cost as calculated in the objective function. Blue points (right column only) are swathe points that were rejected prior to the optimisation due to their correspondence to high error map points but are shown for illustrative purposes.

VII. CONCLUSION

Laser based localisation with a single 2D laser is a well mined area of research. But when one tries to apply it to large (km) scale and long duration deployments new problems arise. Largest amongst these is dealing with local yet drastic spatial change. Gross changes (e.g occlusions from trucks) are typically handled by solving an objective function with robust statistics to implicitly model outliers. We have shown that this is not always sufficient for robust localisation and have proposed a spatially varying model which captures the value/trustworthiness of any point when involved in pose optimisation. We have shown the proposed system exhibits superior performance to a baseline localiser as we leverage more and more error data from our experiences.

REFERENCES

- [1] A. Stewart and P. Newman, "LAPS - localisation using appearance of prior structure: 6-dof monocular camera localisation using prior pointclouds," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Minnesota, USA, May 2012.
- [2] W. Maddern, G. Pascoe, and P. Newman, "Leveraging Experience for Large-Scale LIDAR Localisation in Changing Cities," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA, May 2015.
- [3] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992. [Online]. Available: <http://dx.doi.org/10.1109/34.121791>
- [4] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *IEEE International Conference on Robotics and Automation, ICRA 2010, Anchorage, Alaska, USA, 3-7 May 2010*, 2010, pp. 4372–4378. [Online]. Available: <http://dx.doi.org/10.1109/ROBOT.2010.5509700>
- [5] W. Burgard, O. Brock, and C. Stachniss, *Map-Based Precision Vehicle Localization in Urban Environments*. MIT Press, 2008, pp. 352–. [Online]. Available: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6280129>
- [6] I. Baldwin and P. Newman, "Road vehicle localization with 2d push-broom lidar and 3d priors," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Minnesota, USA, May 2012.
- [7] —, "Laser-only road-vehicle localization with dual 2d push-broom lidars and 3d priors," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Vilamoura, Algarve, Portugal, October 2012.
- [8] 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)*, May 2015, pp. 2814–2821.
- [9] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Third International Conference on 3D Digital Imaging and Modeling (3DIM)*, Jun. 2001.
- [10] G. Turk and M. Levoy, "Zippered polygon meshes from range images," in *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '94. New York, NY, USA: ACM, 1994, pp. 311–318. [Online]. Available: <http://doi.acm.org/10.1145/192161.192241>
- [11] T. Masuda, K. Sakaue, and N. Yokoya, "Registration and integration of multiple range images for 3-d model construction," in *Proceedings of the 1996 International Conference on Pattern Recognition (ICPR '96) Volume 1 - Volume 7270*, ser. ICPR '96. Washington, DC, USA: IEEE Computer Society, 1996, pp. 879–. [Online]. Available: <http://dl.acm.org/citation.cfm?id=844381.845150>
- [12] S. Weik, "Registration of 3-d partial surface models using luminance and depth information," in *3-D Digital Imaging and Modeling, 1997. Proceedings., International Conference on Recent Advances in*, May 1997, pp. 93–100.
- [13] N. Sünderhauf and P. Protzel, "Switchable constraints for robust pose graph slam," in *In Proc. of IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2012.
- [14] P. Biber, "Dynamic maps for long-term operation of mobile service robots," in *In Proc. of Robotics: Science and Systems (RSS)*, 2005.
- [15] C. McManus, W. Churchill, A. Napier, B. Davis, , and P. Newman, "Distraction suppression for vision-based pose estimation at city scales," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, May 2013.
- [16] S. Agarwal, K. Mierle, and Others, "Ceres solver," <http://ceres-solver.org>.
- [17] P. J. Huber, "Robust estimation of a location parameter," *Ann. Math. Statist.*, vol. 35, no. 1, pp. 73–101, 03 1964. [Online]. Available: <http://dx.doi.org/10.1214/aoms/1177703732>
- [18] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.
- [19] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017. [Online]. Available: <http://dx.doi.org/10.1177/0278364916679498>