Long-term 3D Localization and Pose from Semantic Labellings

Carl Toft[1], Carl Olsson[1,2], Fredrik Kahl[1,2]
[1]Chalmers University of Technology, [2]Lund University
{carl.toft, caols, fredrik.kahl}@chalmers.se

Abstract

One of the major challenges in camera pose estimation and 3D localization is identifying features that are approximately invariant across seasons and in different weather and lighting conditions. In this paper, we present a method for performing accurate and robust six degrees-of-freedom camera pose estimation based only on the pixelwise semantic labelling of a single query image. Localization is performed using a sparse 3D model consisting of semantically labelled points and curves, and an error function based on how well these project onto corresponding curves in the query image is developed. The method is evaluated on the recently released Oxford Robotcar dataset, showing that by minimizing this error function, the pose can be recovered with decimeter accuracy in many cases.

1. Introduction

In 1982 Marr's unified theory of vision [25] was published and it has been a major source of inspiration to the vision community. The theory resembles human perception and works on multiple levels; starting with local visual primitives and ending with a global understanding of the scene. Interestingly, when examining today's best performing visual mapping [16, 1, 27] and localization [32, 31] systems, the overall understanding of the scene is largely lacking. Instead they rely on the geometry of point projections and the availability of local features that are descriptive enough to be uniquely and reliably matched across images, without any semantic understanding.

The reliance on local texture descriptors makes the system sensitive to viewpoint changes, weather conditions, lighting and seasonal variations etc. that all affect local scene appearance. Additionally, without any high level understanding it is hard to determine which parts of the scene may be unreliable for localization such as cars or other moving objects. As a consequence traditional geometric localization systems are insufficiently constrained under weak local appearance information.

This paper addresses the fundamental question "Is it possible to perform image localization from high level information, such as a semantic understanding of the image content?" Such information is in contrast to local texture largely invariant to weather, lighting and seasonal changes. We leverage the recent progress in pixelwise semantic image labelling to obtain robust scene descriptions suitable for long-term localization. The basic idea is that the distribution of semantic classes in the query image should alone be sufficient to provide strong constraints on the camera pose.

Figure 1. Two examples of successfully localized pictures. In the left column, the query images are shown together with the reprojection of the 3D curves corresponding to road edges and poles. The images on the right show approximately the same location as seen in the mapping sequences. Note that our baseline method based on LIFT features failed to obtain consistent 2D-3D matches.

To solve the problem we create a scene model consisting of simple geometric primitives, such as 3D points and curves, but with a meaningful semantic label. These are projected into the query image and compared to its semantic content. Our results show that this simple approach can be used for reliable long-term localization from a single query image, see Figure 1 for two examples. As we are only using semantically labelled information in the query image, the added invariance allows us to localize images captured under completely different conditions than the model. One

may argue that we are not using all the information present in the query image, as we only rely on the semantic labels. This is of course correct, and in a practical system one should use all the available information in the query. In this work, we are pushing the limits and investigating if it is possible to achieve reliable camera pose estimation at all under these conditions. Our experimental results on long-term 3D localization in urban street scenes are quite encouraging. We show that one can in many cases achieve global, metric localization from a single image despite variations in seasons and challenging lighting conditions where localization approaches based on local features fail completely.

2. Related work

Traditionally, camera pose estimation (sometimes called "camera resectioning", or simply "localization") is performed by matching point features between the query image and the 3D model. In this case, the model simply consists of a set of points in three-dimensional space. Associated to each point is one or more descriptor vectors, describing the local appearance of the point as it was seen when the 3D model was constructed. When a picture is to be localized, feature points are extracted from the image, and each image point is matched to the most similar point in the 3D model. In this way, a set of 2D-3D correspondences are obtained, from which the full six degrees-of-freedom pose can be calculated [15]. This is in contrast to approaches working in the image domain only, solving the problem of "visual place recognition", see [23] for a survey. We will only be concerned with the 3D localization problem.

The main problem that makes long-term localization difficult is the fact that the feature descriptors used to describe the image and the 3D scene are not invariant to the changes in environment seen during different seasons, weather and time of day (such as SIFT [22], ORB [29] and SURF [5]). Valgren and Lilienthal [34] examined the suitability of SIFT and SURF for long-term localization from a single image and found that the upright U-SURF performed best for their scenario. Another way to approach the long-term localization problem is to find a new descriptor that better copes with changes of the environment. For example, Yi *et al* [35] created a new feature descriptor, called LIFT, by training a convolutional neural network on image patches corresponding to the same feature but viewed under different ambient conditions, and found that this descriptor generated more correct matches between pictures taken under very different lighting conditions compared to SIFT. We use the LIFT descriptor for baseline comparisons to our approach as LIFT outperforms many competing feature descriptors by a large margin including SIFT.

Badino *et al* [3, 4] performed cross-seasonal visual localization on a nine kilometer stretch of road in Pittsburgh. The road was traversed more than a dozen times throughout the span of a year, capturing seasonal variations and a variety of weather conditions. A map was created using one of the traversals, storing the GPS location and the SURF features visible in the camera at more or less equally spaced locations on the road. Localization could then be performed on the remainder of the datasets using a Bayesian filtering approach. The approach is hence dependent on the invariance of the SURF descriptor. To compensate for feature matches being unreliable, a sequence of consecutive images was used to perform localization. The idea of using multiple images for localization (or rather place recognition) was also pursued in [26] where up to 300 consecutive images were used to perform localization based on a image intensity correlation measure. Self-localization using only visual odometry information was investigated in [6].

There are a number of elaborate 3D localization algorithms from a single image that have been developed for handling large rates of incorrect matches, see [20, 7, 19, 9, 33, 30, 36, 18, 32, 31]. Still, if local feature matching is not working properly, such approaches are doomed to fail. In [21], a mining approach is applied to find stable local features over time. Deep learning approaches are presented in [17, 8]. In [28] an information-theoretic metric is derived to compare the query image and a rendered image without relying on individual pixels for the purpose of long-term visual localization. Though it requires a complete geometrical 3D model of the environment. We explore an alternative route to obtain cues that are reliable in the long run by using semantic information.

In [2], object recognition in indoor scenes is applied to obtain more stable matches for robot localization in a 2D map. The approach is based on particle filtering, which means that multiple observations over time are needed. Another source of inspiration for our work is on semantic 3D reconstruction [14]. Here it is shown that 3D reconstruction and multi-view stereo can be supported by using semantic labellings in the image.

3. A motivating example

The input to the localization algorithm is the pixel-wise semantic labelling of the query image. If the camera pose can be computed accurately using only this information, then 3D localization can be performed robustly under varying environmental conditions, provided that the method used for semantic segmentation outputs accurate labels under these conditions. The localization problem is thus moved over to the segmentation itself, making accurate long-term localization a natural consequence of the progress in semantic labelling.

Figure 2 shows a typical semantic labelling of an image from the Oxford Robotcar dataset [24], where the labelling has been obtained by applying the method described in [13] trained on the Cityscapes dataset [10]. The labelling con-

Figure 2. A typical example of a pixelwise semantic labelling of a picture from the Oxford Robotcar dataset.

sists of a single integer for each pixel, denoting the semantic class assigned to it. The classes commonly include road, pavement, buildings, vegetation, poles and sky, among others. Note also that the different connected components in the image are completely featureless and fully characterized by their contours.

Through inspection of the image, one might expect to be able to extract two kinds of pose information from the image. The course spatial distribution of semantic classes in the image should be able to provide rough information about where in the map the image is taken; pictures taken in parks would be dominated by vegetation, whereas pictures taken in the city center would likely contain considerably more buildings.

However, it also seems reasonable to expect to be able to extract more precise metric information as well. The road and the contour where the sky meets the distant vegetation provide information about the camera rotation, the two edges of the road provide information about the lateral position of the car on the road, and the poles on the side of the road should provide accurate information about the longitudinal position along the road. Taking all the evidence into account, it should thus be possible to calculate the full six degrees-of-freedom pose from a single labelled image. In the following section, we present a framework that handles this information in a unified manner and allows efficient pose calculation by minimization of a loss function.

4. Framework for semantic localization

4.1. Model

Our model consists of two types of primitives; 3D points and space curves. The 3D points $\{X_i\}_{i=1}^M$ are each assumed to belong to a single semantic category and therefore have an associated label. Given a candidate 3×4 camera matrix P (which encodes both orientation and position) we compute the projection PX_i and penalize $d_{L_i}(PX_i)$, where $d_{L_i}(x)$ measures the distance between x and the closest pixel in the image labelled L_i. Note that for a pixel labelled L_i the absence of a correctly labelled projection does not incur any penalty. It is only when a 3D point is projected into a semantically different segment that a penalty occurs. This is essential since our 3D models are built using standard SfM systems and are therefore far from complete. Additionally, this allows us to handle occlusion in a very simple but effective way by recording at what distances a 3D point should be seen and adding a depth threshold to the $d_{L_i}(PX_i)$ term.

Since much of the information in a semantically labelled image is stored in the curves separating different classes, our 3D model also includes a set of space curves $\{\mathcal{C}_i\}_{i=1}^N$ endowed with two semantic labels L_i^1 and L_i^2. For the 3D curves we use a penalty $\int_{P\mathcal{C}_i} \eta_{L_i^1,L_i^2}(x(s))ds$, where $\eta_{L_i^1,L_i^2}(x)$ is a function that computes the smallest truncated distance between the point x and an image curve separating regions labelled L_i^1 and L_i^2. Note that our space curves may not correspond to actual physical curves. While the curve separating road and sidewalk is real the skyline is not. We still found that using these and treating them as curves far away helps to constrain the localization. In particular they are useful for determining orientation.

Similar to the 3D points the curves in the 3D model do not need to explain the entire observed image. For example, if we wish to use the skyline where the distant vegetation meets the sky as a curve type (as we do in the experimental section), we are not penalized if the skyline curve in the 3D model is not reprojected onto the entire observed skyline in the image. Instead, we are only penalized for every point where the projection of the 3D curve representing the skyline does not coincide with the observed skyline in the query image.

Our complete loss function is of the following form:

$$E(P) = \sum_{i=1}^{N} \lambda_{L_i^1,L_i^2} \frac{1}{l_i} \int_{P\mathcal{C}_i} \eta_{L_i^1,L_i^2}(x(s))\mathrm{d}s \; + \quad (1)$$
$$+ \sum_{i=1}^{M} \gamma_{L_i} \frac{1}{M_{L_i}} d_{L_i}(PX_i),$$

where the integral is computed with arc-length parametrization in s. The numbers $\lambda_{L_i^1,L_i^2}$ and γ_{L_i} are weights for the different semantic classes, giving us the choice to give some evidence more weight than other, if desired. In the experiments performed in Sec. 5, these constants were all set to one. l_i is the length of the reprojected curve i. The value M_{L_i} is the number of points seen in the image with label L_i. The loss function (1) can be evaluated very efficiently by storing the distance functions $\eta_{L_i^1,L_i^2}$ and d_{L_i} in a look-up table, as shown in Figure 3. When we wish to evaluate

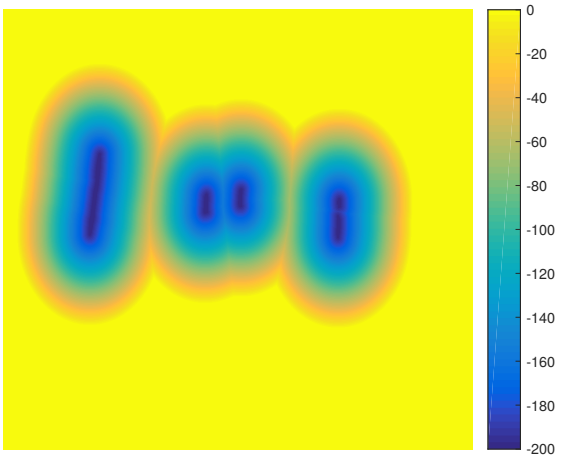

Figure 3. Error map η for the class "poles" in the image in Figure 2.

E for a given pose P, the curves and points are projected into P, and then the corresponding values for η and d are retrieved from the pre-computed table. This makes iterative minimization of (1) very fast.

In the framework presented above, we have not specified what types of curves to use for localization. In the localization experiments in Section 5, the curves $\mathcal{C}_i$ were piecewise linear curves, since these are very simple to project into the cameras and integrate over. We have also not specified any specific semantic labels for the curves yet. The only requirement for them to be useful is that it should be possible to reliably extract these curves from a semantically labelled picture.

4.2. Optimization of loss function

The loss function is a complicated, non-convex function with many local minima. In order to find a good minimum of (1), some prior knowledge about the problem structure must be utilized. Otherwise, if gradient descent is performed on an initially estimated camera pose, we run a high risk of ending up in a local minimum unless the initial pose happens to be very close to the global minimum. In the experiments presented in the next section, we used curves representing the two edges of the road and poles along the street, as well as curves representing the contours of distant trees across the sky. Exploiting this knowledge, the following procedure was performed to minimize $E(P)$.

Given an initial estimate of the pose, gradient descent is performed on (1) using only points and the road edges, the terms for the other lines set to zero. This will likely yield good estimates for the camera rotation. This is followed by gradient descent where only the terms corresponding to road edges and poles are included.

At this stage, the rotation and lateral position of the car are likely close to their optimal values. It is thus reasonable to assume that five of the six degrees-of-freedom have been fixed: three for rotation, one for the lateral direction, and one for the vertical direction. Since only one degree of freedom remains, a line search is performed along this dimension, corresponding to the longitudinal position of the car on the road. This direction is assumed to be along the principal axis of the current camera. Figure 5 shows an example of the loss function along this direction. Finally, a last round of gradient descent is performed from the minimum obtained during the line search, keeping all terms of the loss function. All the derivatives for the gradient descent method are computed numerically.

The final question that must be addressed is where to obtain the initial estimate P_0 of the camera matrix. In a practical application, such as in a real autonomous driving scenario, there is probably a quite good estimate of the car position available from GPS (and other sensors) and internal odometry that could be used as a starting point for the local optimization. However, in this paper we perform global localization from a single labelled image with no other information, and use a simple initialization method based on the spatial statistics of the semantic labels in the query image.

Specifically, the top half of the segmented image is divided into six identically sized regions (two rows and three columns). Each region is then assigned a descriptor vector by making a histogram over all pixel classes in the region (excluding cars and pedestrians), and then normalizing the vector. To this vector, the two gradient histograms are then appended which are obtained from the binary images corresponding to the building and vegetation classes seen in the region, after being normalized and scaled by a factor $1/2$. Finally, the six vectors obtained from all regions are stacked into a final descriptor vector.

During construction of the 3D map, this descriptor vector was calculated for all images in the mapping sequence. When later presented with an image to localize, the descriptor was calculated for the query image, and then matched to the closest descriptor from the mapping sequence. The found camera was then used as the initial camera matrix P_0 for local optimization.

5. Experiments

The presented framework for localization from semantically labelled images was evaluated on the Oxford Robotcar dataset [24]. Two different locations were used for the experiments. The first was a stretch of road approximately one hundred meters long and was traversed three times in slightly different weather conditions during May 6, 2014. The second sequence was approximately 70 m long and used to evaluate cross-seasonal localization. The data collected on November 28, 2014 was used to build the 3D map, and the data collected on the February 3, 2015 was used for

Figure 4. An image from the mapping sequence, together with the reprojections of the 3D curves in the model.

3D localization. Table 1 contains some more information about the individual datasets.

To generate a gold-standard localization reference, all the sequences were reconstructed using the publicly available structure from motion pipeline described in [11]. By manually adding 2D correspondences between pairs of sequences where necessary, all trajectories were reconstructed in the same coordinate system. Note that adding manual correspondences was a necessity as there were very few correspondences across the sequences. Bundle adjustment was then applied to all points and cameras simultaneously.

The first sequence of each location (i.e., datasets 1 and 4 in Table 1) was used as a reference - so called mapping sequence - from which semantic 3D maps were created, as will be explained below, and then the remaining sequences were used to evaluate the localization algorithm. Since no ground truth camera matrices are available in the dataset, the camera matrices obtained for the test sequences after bundle adjustment were used as a gold standard reference that the semantic localization could be compared against.

Piecewise linear three-dimensional curves of three different types were reconstructed from the two mapping datasets. The different curve types used were road edges, poles and distant vegetation-sky intersections. Figure 4 shows an image from the mapping datasets, where the 3D model has been projected down into the camera. Note that the semantic 3D model is sparse in the sense that it contains few elements and does not cover all the imaged semantic content. As all space curves are piecewise linear, they are represented as a discrete set of points. The poles are thus represented by their start and end points, the road edges consist of around 100 3D points each.

The vegetation-sky curve might at first seem like a strange choice to include as a space curve, but it was found that the distant skyline was extracted from the semantic segmenter with remarkable consistency. Note also that if we can successfully match it to a curve in the observed image, we have fully determined the camera rotation. The only drawback compared with the other curves used is that the curve is not valid when the camera gets close to the curve. This turned out to not be a big problem in practice, since by the time it is no longer accurate, it has vanished from the top of the image and is no longer visible.

The road edge was automatically reconstructed by extracting four points on the road-pavement intersection in the 2D mapping image (using the semantic labellings), identifying the 3D points visible within the obtained quadrilateral, and then fitted a (road) plane through the corresponding 3D points using RANSAC [12]. The four corner points identified in the picture were then added to the 3D road curve. This procedure was repeated through the mapping sequences.

The poles were automatically reconstructed by tracking the corresponding connected components of the segmented pictures in the mapping sequence. Lines were then fitted to each pole in each image using a Hough transform. A line in 3D space was then obtained for each pole by backprojecting each observed 2D line and finding the intersection between these planes. The top and bottom points on the 3D lines were extracted based on at what height the top and bottom points of the pole were seen in the images. The 3D points of the model were obtained by triangulating consistent SIFT matches in the mapping sequences, where consistent here means that the 3D points satisfy the epipolar geometry and that they project to the same semantic label in all the visible mapping images.

The vegetation-sky curves were manually extracted by selecting a piecewise linear arc in an image. The 3D points seen in the image near that region were then retrieved, and the 3D curve was placed at a depth equal to the median depth of those points.

To perform localization of a single query image, the following procedure is followed. First, the image is semantically labelled. From this labelling, the error maps $\eta_{L_i^1,L_i^2}$ and d_{L_i} are calculated (cf. Figure 8). An initial estimate for the camera matrix P is then obtained, from which local optimization of (1) is performed. All constants λ and γ in (1) were set to one.

6. Results

The localization results are shown in Figure 6. Localization was performed on every image in the test datasets, each image being treated completely independently from the others. The results for datasets 2 and 3 (cf. Table 1) are shown together, and contain a total of 367 query images. The winter sequence (dataset 5) contains 71 images in total. The top left histogram over translational errors only show errors up

Dataset	Date of collection	Purpose	Weather	Number of images
1	2014-05-06	Map building	Cloudy, diffuse lighting, few shadows	160
2	2014-05-06	Localization	Similar to above	188
3	2014-05-06	Localization	Mostly cloudy, but some sun and shadows	179
4	2014-11-28	Map building	Overcast, diffuse lighting, few shadows	46
5	2015-02-03	Localization	Winter, snow, some sun	71

Table 1. The five datasets used for evaluating the semantic localization algorithm. The top three datasets represent the same physical road, traversed three times during the same day, and the last two datasets represent a different road, traversed during two different seasons.

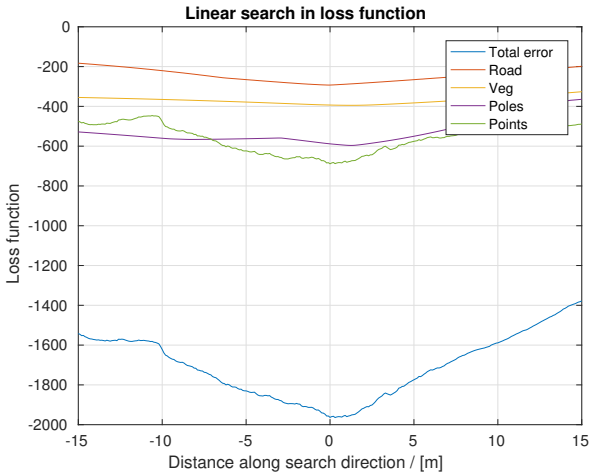

Figure 5. Example of cross-section of the loss function along the longitudinal direction. This line search is performed after the lateral direction and rotation have been established through gradient descent on the terms representing road edges in Eq. (1).

to 10 m, but there were 20 outliers with translational errors greater than this, corresponding to bad initializations by the histogram matching procedure described at the end of Section 4. The rotational error histogram in the left column show all rotational errors, while in the winter road sequence (right histogram), there were 5 outliers outside the range shown. For the translational errors in the winter road sequence, there are only three outlier images outside the range shown in the translational error histogram.

The bottom row in Figure 6 shows a comparison with a three-point RANSAC using LIFT features. For a given value on the x-axis, the y-axis shows what fraction of the test images were localized to within the given value of x.

A few remarks are in order. First, for the first two test datasets, the localization accuracy is reasonably good. The translational error is less than a meter for around 73% of the images, and it is within two meters for 89%. The rotation was recovered within 2° degrees for 89% of the images.

When an image is successfully localized by LIFT, it is in general much more precisely localized than it is by the semantic localization method presented here. LIFT often recovers the pose with centimeter accuracy, whereas a pose constrained by several clear curves in our model tends to be localized within a few decimeters or half a meter. This is not very surprising, since the semantic features are more smeared out across the image than point features, and when looking at a given semantic segmentation, there often exists a rather large ambiguity as to where, for example, the poles and road edges actually are located.

For the winter road sequence, the localization errors for both LIFT and the current method were much larger than in the other test sets. In the first sequence, all three datasets were collected during the same day, so the dataset used to create the map was similar in appearance to the two test datasets. However, for the second sequence, the mapping dataset used to create the semantic 3D map was collected in late fall during a day when there was no snow, and the test dataset was collected in February during a snowy day, so the mapping and the test datasets appear very different.

The semantic localization algorithm mostly failed due to inaccurate segmentations. Most public datasets for street-view segmentation do not contain winter scenes, making the segmenters less accurate on these scenes. For example, the snowy ground was often misclassified, typically as a car, which made the loss function inaccurate since one of its terms evaluated how well the 3D curves corresponding to the road edges project down onto the road edges as seen in the query image. Figure 7 shows an example that our algorithm failed to localize. No correct road edges were detected, and the only pole that was correctly segmented was a drain pipe on the house in the background. Since very little useful information could be extracted from the input image, the localization failed.

Overall, we have seen that when the segmentation is accurate, it is generally possible to recover the camera rotation and translation with good accuracy, confirming that the semantic labelling conveys very strong information about the camera pose. See Figures 1 and 8 for examples of successful localizations.

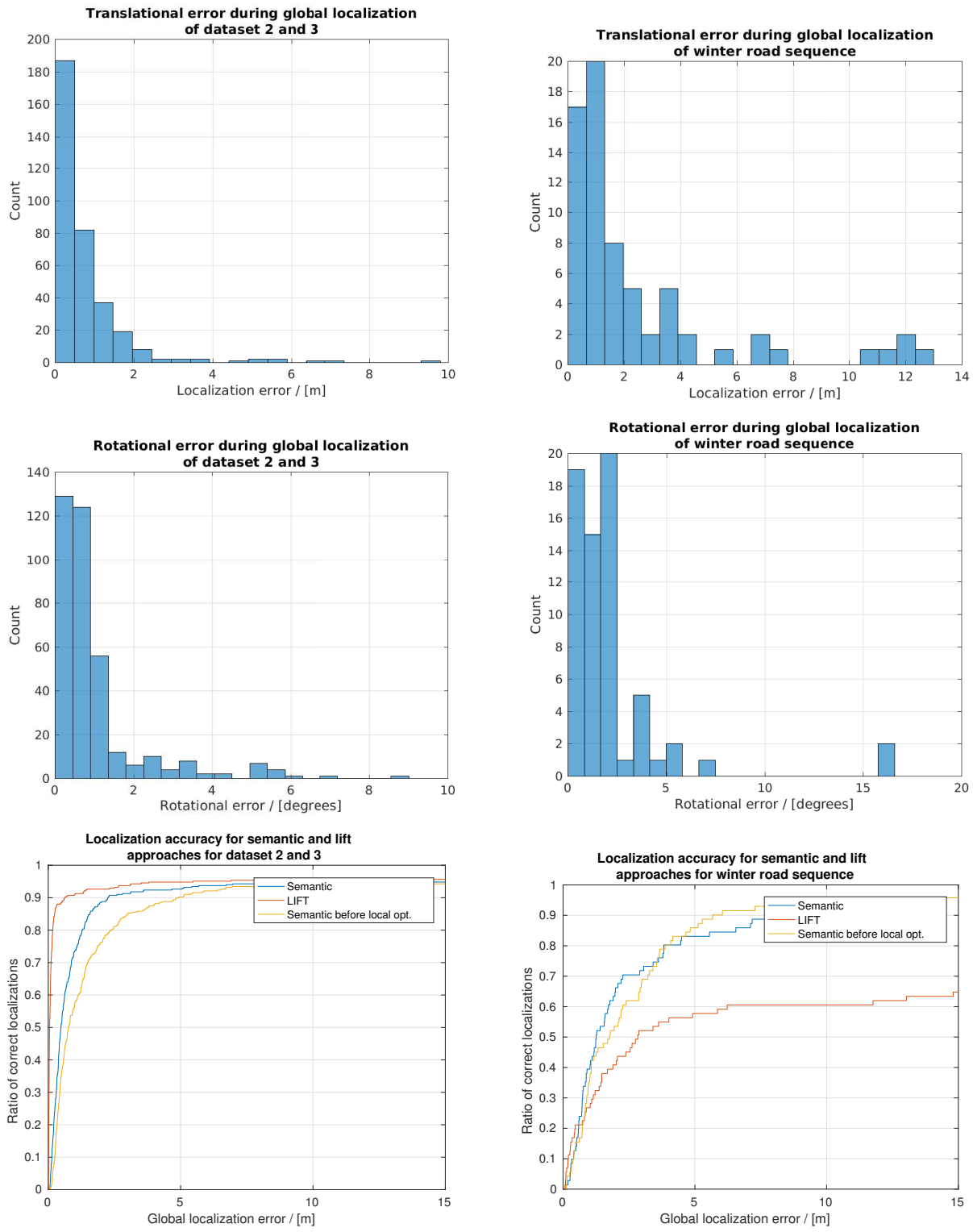

Figure 6. Localization results for the three datasets. The results for datasets 2 and 3 were similar, and have therefore been merged together. The first row shows histograms over the translational localization errors, and the second row shows the rotational errors. On the third row, a comparison is made with an approach based on LIFT point features. For a given value on the x-axis, the corresponding y-value gives the proportion of localizations with a translational error less than the x-value. For example, for datasets 2 and 3, 90% of the images were retrieved with 2.5 m or less translational error. Also shown is the translational errors before any local optimization is performed (i.e., using only the image retrieved by the semantic retrieval initialization method).

Figure 7. A failure case: No pavement was labelled as pavement, yielding no road edges that could be used for localization.

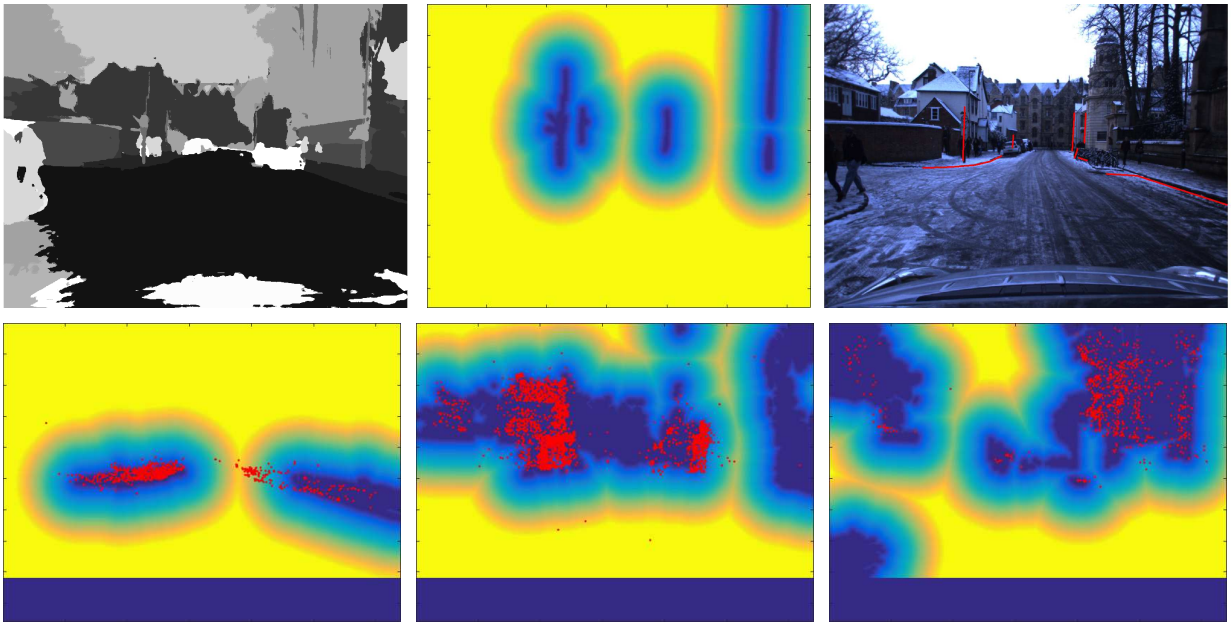

Figure 8. *Top:* An example of a successful localization from the winter sequence. The image to the top left is the semantic segmentation of the query image. The top middle image shows the error map η corresponding to the poles observed in the segmented image. The figure on the top right shows the original version of the query image before semantic segmentation, together with the 3D structure projected down into the estimated camera P found by minimizing the loss function. *Bottom:* The error maps d_{L_i} for the classes sidewalk, building and vegetation, respectively, together with the corresponding 3D points X_i projected down onto the estimated camera P.

7. Conclusion

We have considered the problem of how much pose information is stored in the semantic labels of a picture alone. We presented a method for performing full camera pose estimation based only on the pixelwise semantic labelling of the query image, and saw that in situations where the labelling is accurate, it is possible to recover the camera translation to within a few decimeters or meters accuracy, depending on the quality and location of the features observed, and the camera rotation to within a few degrees. We have thus shown that a good semantic segmentation provides very strong constraints on the camera pose.

We believe that these results are very encouraging, since it implies that the steady progress of pixelwise semantic labelling can naturally be leveraged to improve the robustness of localization algorithms that otherwise have trouble when mapping and localization occur far apart in time.

This work has been funded by the Swedish Research Council (grant no. 2016-04445), the Swedish Foundation for Strategic Research (Semantic Mapping and Visual Navigation for Smart Robots) and Vinnova / FFI (Perceptron, grant no. 2017-01942).

References

[1] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. Building rome in a day. *Commun. ACM*, 54(10):105–112, 2011. 1

[2] N. Atanasov, M. Zhu, K. Daniilidis, and G. J. Pappas. Semantic localization via the matrix permanent. In *Robotics: Science and Systems*, 2014. 2

[3] H. Badino, D. Huber, and T. Kanade. Visual topometric localization. In *Intelligent Vehicles Symposium*, 2011. 2

[4] H. Badino, D. Huber, Y. Park, and T. Kanade. Real-time topometric localization. In *International Conference on Robotics and Automation*, 2012. 2

[5] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. 110(3):346–359, 2008. 2

[6] M. Brubaker, A. Geiger, and R. Urtasun. Map-based probabilistic visual self-localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(4):652–665, 2016. 2

[7] D. M. Chen, G. Baatz, K. Köser, S. S. Tsai, R. Vedantham, T. Pylvä, K. Roimela, X. Chen, J. Bach, M. Pollefeys, et al. City-scale landmark identification on mobile devices. In *Conference Computer Vision and Pattern Recognition*, 2011. 2

[8] Z. Chen, A. Jacobson, N. Sünderhauf, B. Upcroft, L. Liu, C. Shen, I. D. Reid, and M. Milford. Deep learning features at scale for visual place recognition. *CoRR*, abs/1701.05105, 2017. 2

[9] S. Choudhary and P. Narayanan. Visibility probability structure from sfm datasets and applications. In *European Conference on Computer Vision*, 2012. 2

[10] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Conference Computer Vision and Pattern Recognition*, 2016. 2

[11] O. Enqvist, C. Olsson, and F. Kahl. Non-sequential structure from motion. In *Workshop on Omnidirectional Vision, Camera Networks and Non-Classical Cameras (OMNIVIS)*, 2011. 5

[12] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981. 5

[13] G. Ghiasi and C. C. Fowlkes. Laplacian reconstruction and refinement for semantic segmentation. In *European Conference on Computer Vision*, 2016. 2

[14] C. Häne, C. Zach, A. Cohen, and M. Pollefeys. Dense semantic 3D reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. Accepted for publication. 2

[15] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. 2

[16] F. Kahl and R. Hartley. Multiple view geometry under the L_∞-norm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(9):1603–1617, 2008. 1

[17] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *International Conference on Computer Vision*, 2015. 2

[18] V. Larsson, J. Fredriksson, C. Toft, and F. Kahl. Outlier rejection for absolute pose estimation with known orientation. In *British Machine Vision Conference*, 2016. 2

[19] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. Worldwide pose estimation using 3D point clouds. In *European Conference on Computer Vision*, 2012. 2

[20] Y. Li, N. Snavely, and D. P. Huttenlocher. Location recognition using prioritized feature matching. In *European Conference on Computer Vision*, 2010. 2

[21] C. Linegar, W. Churchill, and P. Newman. Made to measure: Bespoke landmarks for 24-hour, all-weather localisation with a camera. In *International Conference on Robotics and Automation*, 2016. 2

[22] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 2

[23] S. Lowry, N. Snderhauf, P. Newman, J. Leonard, D. Cox, P. Corke, and M. Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, 2016. 2

[24] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017. 2, 4

[25] D. Marr. *Vision*. W. H. Freeman and Company, 1982. 1

[26] M. Milford and G. Wyeth. SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights. In *International Conference on Robotics and Automation*, 2012. 2

[27] P. Moulon, P. Monasse, and R. Marlet. Global fusion of relative motions for robust, accurate and scalable structure from motion. In *International Conference on Computer Vision*, 2013. 1

[28] G. Pascoe, W. Maddern, and P. Newman. Robust direct visual localisation using normalised information distance. In *British Machine Vision Conference*, 2015. 2

[29] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571, Nov 2011. 2

[30] T. Sattler, M. Havlena, F. Radenovic, K. Schindler, and M. Pollefeys. Hyperpoints and fine vocabularies for large-scale location recognition. In *International Conference on Computer Vision*, 2015. 2

[31] T. Sattler, B. Leibe, and L. Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1744–1756, 2017. 1, 2

[32] L. Svärm, O. Enqvist, F. Kahl, and M. Oskarsson. City-scale localization for cameras with known vertical direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(7):1455–1461, 2017. 1, 2

[33] L. Svärm, O. Enqvist, M. Oskarsson, and F. Kahl. Accurate localization and pose estimation for large 3D models. In *Conference Computer Vision and Pattern Recognition*, 2014. 2

[34] C. Valgren and A. J. Lilienthal. SIFT, SURF & seasons: Appearance-based long-term localization in outdoor environments. *Robotics and Autonomous Systems*, 58(2):149 – 156, 2010. 2

[35] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. LIFT: Learned invariant feature transform. In *European Conference on Computer Vision*, 2016. 2

[36] B. Zeisl, T. Sattler, and M. Pollefeys. Camera pose voting for large-scale image-based localization. In *International Conference on Computer Vision*, 2015. 2