Matching Geometry for Long-term Monocular Camera Localization

Tim Caselitz Bastian Steder Michael Ruhnke Wolfram Burgard

***Abstract*— Localizing a camera with respect to a given map is a fundamental problem in vision-based navigation. Real-world applications require methods that are capable of long-term operation because recordings of available maps often date back considerably compared to the time of localization. Unfortunately, the photometric appearance of the environment can change tremendously even over short periods, making image matching a difficult problem. Since geometric properties of the environment are typically more stable than its photometric characteristics, we propose to rely on matching geometry in order to achieve long-term camera localization. Thus, our approach is agnostic to changes in photometric appearance. We present real-world experiments which demonstrate that our method accurately tracks the 6-DoF pose of a camera over long trajectories and under varying conditions. We evaluate our method using publicly available and own datasets and visualize the successful tracking by post-colorizing the given geometric map.**

I. Introduction

Accurate localization is an important prerequisite for a large variety of navigation tasks. For example, an accurate pose estimate enables an autonomous mobile robot or a pedestrian using its smartphone to plan a path to a given goal location. While GPS can provide accurate pose estimates at a global scale, it suffers from substantial errors due to multipath effects in urban canyons and does not work indoors. This is a major drawback, since it limits navigation capabilities in areas that are highly populated and therefore have a high demand for localization services, e.g., city centers and shopping malls. A popular approach to robotic localization is to match sensor observations against a previously acquired map. In most approaches, sensors used for mapping and localization are identical to allow for a direct comparison of their data. While LiDARs provide highly accurate metric measurements, they are expensive and heavy. In contrast, cameras are low-priced and light-weight but do not directly provide metric information. In this paper, we present an approach to localize a camera within a geometric map that was acquired by a LiDAR. In this way, we can rely on its accuracy but still exploit the advantages of cameras during localization.

Our method employs visual odometry [10] to track the camera motion and to reconstruct a sparse set of 3D points via bundle adjustment [12]. For this purpose we rely on components of ORB-SLAM [7], a state-of-the-art solution for monocular SLAM (simultaneous localization and mapping) that was presented by Mur-Artal *et al.* and is available open-source. It stands in a line of research with PTAM [5] that was presented by Klein *et al.* and made bundle adjustment applicable to real-time visual SLAM by parallelizing camera tracking and map building. In contrast to approaches based on stereo cameras, *monocular* visual odometry suffers from the unobservability of scale. This means the whole reconstruction is only correct up to scale and scale is prone to drift over time (see Strasdat *et al.* [11]).

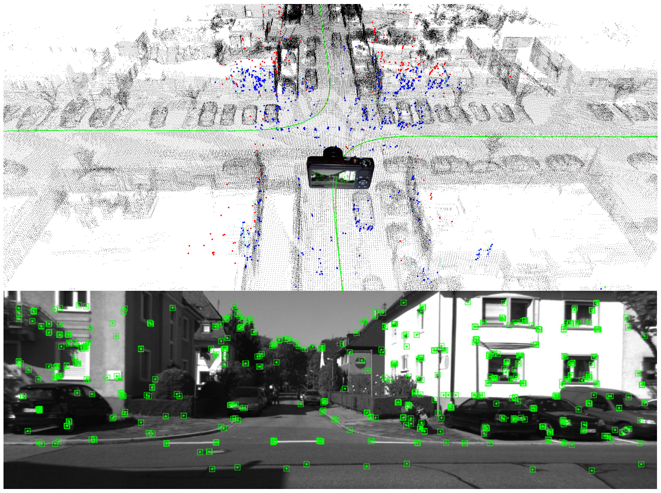

Fig. 1. Our method tracks the 6-DoF camera pose by reconstructing 3D points (blue, red) from image features (bottom, green) and matching them against a prior geometric map (black) built from LiDAR data. Blue points are estimated to lie inside the map, while red points have no correspondence. The green line represents the camera trajectory.

The majority of research concerning visual localization has focused on matching photometric characteristics of the environment. This is either done by comparing image feature descriptors like SIFT or SURF or directly operating on image intensity values. Yet, one of the main issues in visual localization is that the environments' photometric appearance changes substantially over time, especially across seasons. Churchill *et al.* [2] approach this problem by storing multiple image sequences for the same place from different times. Naseer *et al.* [8] tackle the problem by matching trajectories using a similarity matrix.

In contrast to methods based on matching photometry, approaches for camera localization in geometric maps built from LiDAR data are less present in the literature. Wolcott *et al.* [13] proposed a method to localize an autonomous vehicle in urban environments. Using LiDAR intensity values, they render a synthetic view of the mapped ground plain and match it against the camera image by maximizing normalized mutual information. While this approach only provides the 3-DoF pose, the method presented by Pascoe *et al.* [9]

All authors are with the Department of Computer Science, University of Freiburg, Germany.

estimates the full 6-DoF camera pose. Their appearance prior (map) combines geometric and photometric data and is used to render a view that is then matched against the live image by minimizing the normalized information distance. Both approaches perform matching in 2D space and therefore require expensive image rendering supported by GPU hardware. Furthermore, their prior comprises LiDAR intensities or visual texture respectively. In contrast, our method relies on geometric information only. By directly matching 3D geometry, we also avoid the need for computations on a GPU. However, we achieve comparable results in terms of accuracy and framerate.

The approach we present in this paper tracks the 6-DoF pose of a monocular camera by reconstructing a sparse set of 3D points from image features via bundle adjustment and subsequently matching it against a map previously built from LiDAR data (see Fig. 1). The main contribution of this work is to align the reconstructed point set with the prior geometric map by continuously applying an estimated similarity transformation. We formulate the estimation problem as least-squares error minimization and solve it with g2o [6] in an iterative closest point (ICP) [1] procedure. Since geometry is a characteristic of the environment that typically remains reasonably stable over time, we think it is favorable to rely on geometric information for long-term localization. In contrast, photometric appearance of the environment is highly dependent on time of day, weather, and season.

II. Proposed Method

The objective of our method is to localize a monocular camera. The inputs are an image stream and a prior geometric map represented as point cloud. Since our approach is intended for tracking, a coarse estimate to initially localize the camera within the prior map is required. The output of our method is a 6-DoF camera pose estimate at frame-rate.

Our approach builds on a visual odometry system that uses local bundle adjustment to reconstruct camera poses and 3D points from image features. Given the camera poses relative to these points, we indirectly localize the camera by aligning the reconstructed points with the prior map. In this way, we continuously eliminate the drift accumulated by visual odometry. Since we use a *monocular* camera, drift occurs in translation, rotation, and scale. We therefore realize the alignment by applying a 7-DoF similarity transformation whenever a new keyframe is selected.

In the following subsections we first define the local reconstruction consisting of camera poses and 3D points. Subsequently, we describe the data association between reconstructed points and points of the prior map. Finally, we discuss the estimation of the similarity transformation used for alignment.

A. Local Reconstruction

The local reconstruction is a set of points $\mathbf{d}_i \in \mathbb{R}^3$ observed as features in images captured from keyframe poses $\mathbf{T}_i \in SE(3)$. We include all points and keyframes in our local reconstruction that are part of the current visual odometries' local bundle adjustment problem. Adhering to this choice has two reasons. First, it ensures that the local reconstruction is consistent because the bundle adjustment optimization is performed right before our localization. Second, the alignment we perform does not influence and potentially corrupt the visual odometry, since we *uniformly* transform its local bundle adjustment problem.

B. Data Association

To determine the alignment of local reconstructions with the prior map, we search correspondences between reconstructed points and prior map points. We do this within an ICP scheme, i.e., iteratively update the data associations based on the current similarity transformation estimate. For each reconstructed point, we find its nearest neighbor in the prior map, which is stored in a k-d tree to allow for fast look-up. If the nearest neighbor is close enough, the pair is added to the correspondence set. We reduce the distance threshold linearly with the number of iterations: as the algorithm converges, point-to-point distances decrease, and we can choose the threshold more strictly in order to retain only high-quality associations.

A major problem of finding correspondences exclusively based on nearest neighbors is that the set of reconstructed points typically overlaps *only partially* with the prior map. Therefore points might have no meaningful correspondence even though their position is perfectly estimated. The partial overlap is often caused by an incomplete mapping due to the LiDARs' limited vertical field of view. Reconstructed points originating from unmapped geometric structure would be associated with boundary points of the prior map, thus leading to biased transformation estimates. Since these can have severe consequences in terms of accuracy but also convergence, we refine the correspondence set by rejecting associations to points that lie outside the prior map according to its local point distribution.

In order to analyze its local point distribution, we voxelize the prior map and perform a principal component analysis (PCA) on the covariance matrix of all points inside a voxel. Since the prior map is computed offline, this does not affect the online performance of our method. We reject correspondences if the amount of points is insufficient or if reconstructed points lie outside of a multiple of the standard deviation along the voxels' principle component axes. The refined correspondence set is used to determine the alignment of reconstructed points and prior map.

C. Alignment

Given a set of correspondences, we estimate a similarity transformation $\mathbf{S} \in Sim(3)$ that aligns the local reconstruction with the prior map. The set of correspondences $\mathcal{C}_k$ is updated iteratively. In each iteration k we estimate

$$\mathbf{S}_k^* = \underset{\mathbf{S} \in Sim(3)}{\operatorname{argmin}} \; F_{Data}(\mathbf{S}, \mathcal{C}_k) \tag{1}$$

by minimizing a non-linear least squares objective function, which we solve with the Levenberg-Marquardt algorithm.

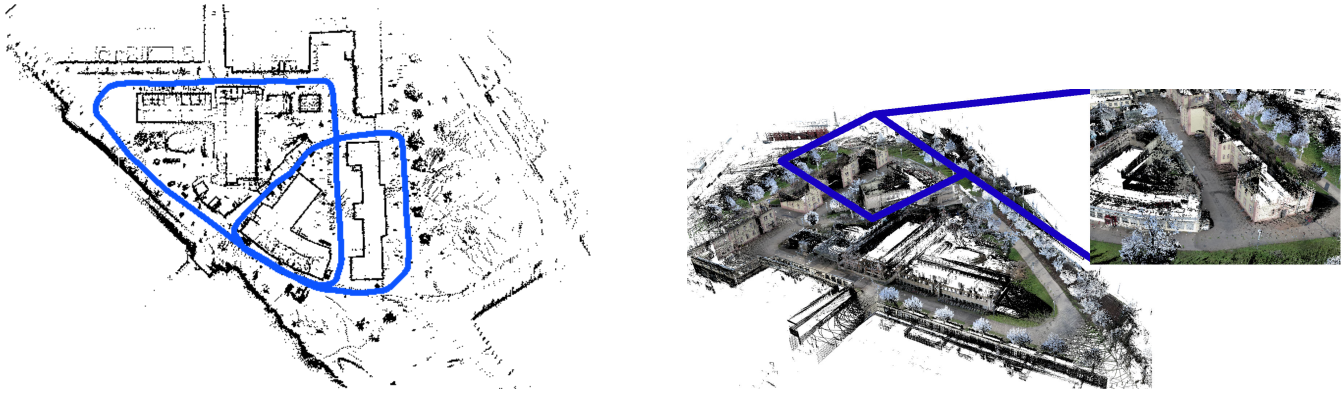

Fig. 2. Left: Map of the Freiburg campus built from LiDAR data (top-view, ground removed for visualization). The blue trajectory was used during the map acquisition and in reverse direction for the localization runs. Right: The same LiDAR point cloud post-colorized based on the pose estimates of our localization method. The images were captured with a hand-held compact camera.

We estimate $\mathbf{S}_k^*$ in the reference frame of the current keyframe. This is advantageous compared to a parametrization in the reference frame of the prior map because the optimization variables better correspond to the dimensions of drift which we want to compensate with $\mathbf{S}_k^*$. Our error function F_{Data} is the sum of squared Euclidean distances between points $\mathbf{d}_i$ and corresponding points $\mathbf{m}_j \in \mathbb{R}^3$ of the prior map:

$$F_{Data}(\mathbf{S}, \mathcal{C}_k) = \sum_{\mathcal{C}_k} \rho\left(\mathbf{e}_{Data}^\top \mathbf{e}_{Data}\right), \tag{2}$$

$$\mathbf{e}_{Data}(\mathbf{S}, \mathbf{d}_i, \mathbf{m}_j) = s\mathbf{R}\mathbf{d}_i + \mathbf{t} - \mathbf{m}_j, \tag{3}$$

where $s \in \mathbb{R}$, $\mathbf{R} \in SO(3)$, and $\mathbf{t} \in \mathbb{R}^3$ form $\mathbf{S} \in Sim(3)$. We use a Huber cost function $\rho(\cdot)$ to be more robust against outliers in the data association. As shown by Fritzgibbon [3] this leads to better convergence properties of the objective function. We choose the kernel size such that we retain the quadratic error range over all iterations.

Once we have estimated $\mathbf{S}_k^*$ for all iterations, we compute a joint similarity transformation

$$\mathbf{S}^* = \prod_{k=0}^{K-1} \mathbf{S}_{K-k}^*. \tag{4}$$

To align the local reconstruction with the prior map, we transform all point positions $\mathbf{d}_i$ and keyframe poses $\mathbf{T}_i$ (as defined in II-A) with the estimated transformation $\mathbf{S}^*$. Since we estimate $\mathbf{S}^*$ in the reference frame of the current keyframe, it has to be transformed back into the frame of the prior map before it is applied.

III. Experimental Evaluation

A. Evaluation of Accuracy

In order to evaluate the accuracy of our method and to allow for easy comparison, we tested it on sequence 00 of the publicly available KITTI odometry dataset [4]. We used a LiDAR-based SLAM system to build a consistent geometric map from the provided LiDAR data (see Fig. 1). This map was then employed to track the pose of camera 0 on the vehicle. Since the relative transformation between camera and LiDAR is known, the ground truth camera trajectory is given as an output of the SLAM system. We can therefore compute 6-DoF localization errors.

The camera was tracked successfully along the whole trajectory in all 10 localization runs. Averaging over all runs and the entire trajectory yielded an translational error of 0.30 ± 0.11 m and an rotational error of $1.65 \pm 0.91\,^\circ$. Since our method runs at approximately 10 fps on an i7 CPU using three threads, we achieved online tracking in the KITTI experiment.

B. Evaluation under Varying Conditions

To evaluate our approach under varying conditions, we captured a dataset with one of our robots, which is equipped with a Velodyne HDL-32E LiDAR. We built the geometric map of the Freiburg campus shown in Fig. 2 (left). We also collected images from two different hand-held cameras (a Canon S100 compact camera and an Apple iPhone 5S) at two different days with varying weather conditions and tracked the poses of the two cameras relative to this map. Fig. 3 (top) shows example images for both settings. To highlight the viewpoint invariance of our method, the localization runs were performed in the opposite direction compared to the map acquisition. For visual localization approaches relying on non-omnidirectional cameras, this can be extremely difficult as the photometric appearance of the environment can tremendously depend on the viewing direction.

For both camera trajectories we performed 10 successful localization runs. The standard deviation was 0.06 m / 0.46 ° for the Canon camera trajectory and 0.12 m / 0.35 ° for the iPhone trajectory. Since we do not have ground truth trajectories for these camera poses, we cannot provide errors. Yet, given the camera pose with respect to the map, we can post-colorize the map points to visualize the quality of the estimates. Fig. 2 (right) and Fig. 3 (bottom) show the post-colorized Freiburg campus map. The varying conditions under which these images were captured support the hypothesis that matching geometry is agnostic to changes in the photometric appearance of the environment.

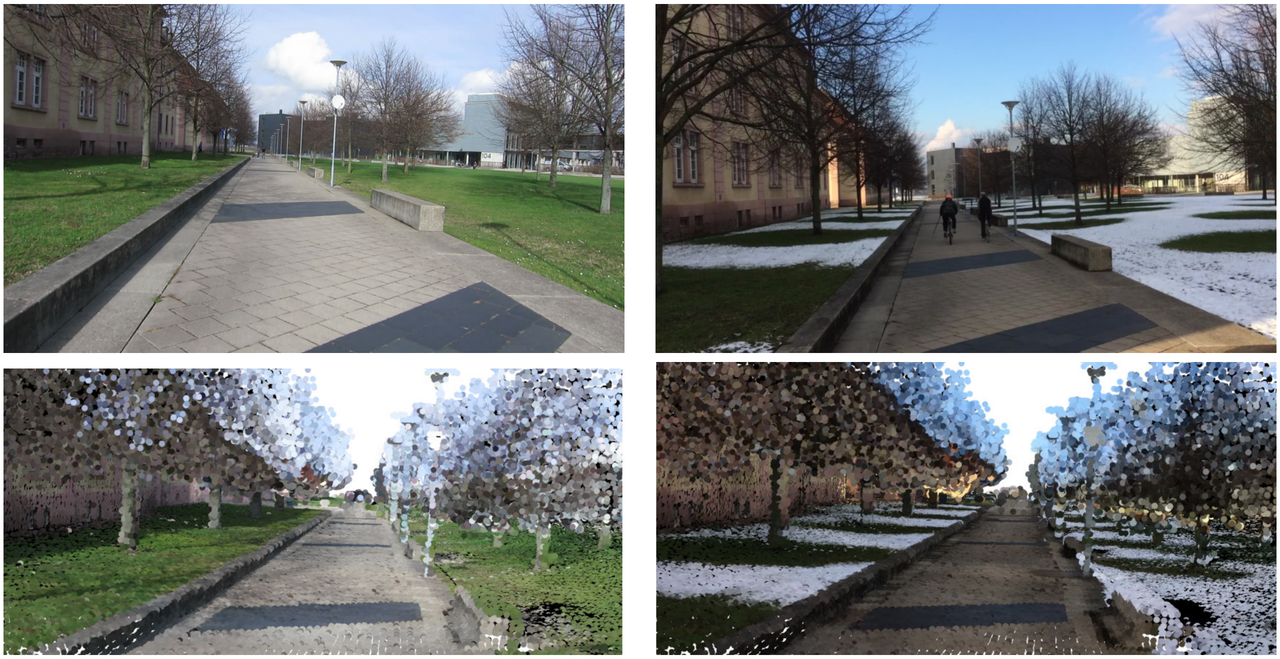

Fig. 3. Top: Example camera images from the Freiburg campus datset. Left: A day without snow captured with a Canon S100 compact camera. Right: A snowy day captured with the camera of an Apple iPhone 5S. Bottom: Post-colorized point cloud from a similar point of view. As can be seen, very fine structures like thin tree branches tend to be colorized incorrectly (with sky texture). This is related to the localization accuracy but also caused by inaccuracies of the 3D map.

IV. Conclusions

In this paper, we presented a novel approach to localize a monocular camera with respect to a given geometric map. Our method employs visual odometry to track the 6-DoF camera pose and to reconstruct a sparse set of 3D points via bundle adjustment. We align the reconstructed points with the prior map by continuously applying an estimated similarity transformation to indirectly localize the camera. Matching geometry turns out to be advantageous regarding long-term localization as geometric properties of the environment are typically more stable than its photometric appearance. Experiments carried out using a publicly available large-scale dataset demonstrate that the accuracy and framerate of our method are comparable to state-of-the-art approaches even though it does not rely on any additional information or GPU hardware support. Additional experiments carried out under varying conditions indicate that approaching visual localization in this way yields the benefit of being agnostic to changes in photometric appearance inevitable in long-term localization scenarios. They furthermore suggest that localizing a camera in panoramic maps built from LiDAR data provides excellent view-point invariance.

Acknowledgment

The authors would like to thank Mur-Artal *et al.* for providing the ORB-SLAM source code and Geiger *et al.* for providing the KITTI dataset.

References

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

[2] W. Churchill and P. Newman, “Experience-based navigation for long-term localisation,” *International Journal of Robotics Research*, vol. 32, no. 14, pp. 1645–1661, 2013.

[3] A. W. Fitzgibbon, “Robust registration of 2d and 3d point sets,” *Image and Vision Computing*, vol. 21, no. 13, pp. 1145–1153, 2003.

[4] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[5] G. Klein and D. Murray, “Parallel tracking and mapping for small ar workspaces,” in *Proc. of the IEEE/ACM Int. Symposium on Mixed and Augmented Reality (ISMAR)*, 2007.

[6] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, “g2o: A general framework for graph optimization,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

[7] R. Mur-Artal, J. Montiel, and J. D. Tardos, “Orb-slam: a versatile and accurate monocular slam system,” *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[8] T. Naseer, M. Ruhnke, C. Stachniss, L. Spinello, and W. Burgard, “Robust visual slam across seasons,” in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2015.

[9] G. Pascoe, W. Maddern, and P. Newman, “Direct visual localisation and calibration for road vehicles in changing city environments,” in *Proc. of the IEEE Int. Conf. on Computer Vision (ICCV) Workshops*, 2015.

[10] D. Scaramuzza and F. Fraundorfer, “Visual odometry [tutorial],” *IEEE Robotics Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[11] H. Strasdat, J. Montiel, and A. J. Davison, “Scale drift-aware large scale monocular slam,” in *Proc. of Robotics: Science and Systems (RSS)*, 2010.

[12] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, “Bundle adjustment — a modern synthesis,” in *Vision Algorithms: Theory and Practice*. Springer, 1999, pp. 298–372.

[13] R. W. Wolcott and R. M. Eustice, “Visual localization within lidar maps for automated urban driving,” in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2014.