OpenStreetSLAM: Global Vehicle Localization Using OpenStreetMaps

Georgios Floros, Benito van der Zander, and Bastian Leibe

***Abstract*— In this paper we propose an approach for global vehicle localization that combines visual odometry with map information from OpenStreetMaps to provide robust and accurate estimates for the vehicle's position. The main contribution of this work comes from the incorporation of the map data as an additional cue into the observation model of a Monte Carlo Localization framework. The resulting approach is able to compensate for the drift that visual odometry accumulates over time, significantly improving localization quality. As our results indicate, the proposed approach outperforms current state-of-the-art visual odometry approaches, indicating in parallel the potential that map data can bring to the global localization task.**

I. Introduction

The development of autonomously driving vehicles has become a very active research area and many systems have been created towards the goal of self-driving cars over the last few years (*e.g.*, [1], [2]). Localization is an important component regarding the stability of such systems and thus its precision and robustness is of high importance. Current localization methods rely mostly on GPS information. However, GPS signals are not always available (*e.g.*, inside narrow streets) and are usually only accurate up to a range of several meters. To overcome this problem, several authors have proposed to use image information from camera systems mounted on top of the vehicles.

These vision-based methods can be classified into two main categories. One line of interesting research has emerged towards performing image based localization using large geotagged image corpora acquired from specially equipped platforms [3] (*e.g.*, Google Street View data). Their estimation of the vehicle location is based on matching the image acquired from the vehicle's cameras to an image database and finding the best matches, which will determine also the vehicle's position. Although localization results are promising, the large image databases required for these systems make them expensive to build up and maintain for real-world applications. A second category of approaches is based on *visual Simultaneous Localization and Mapping* (vSLAM, *e.g.*, [4]). One of the main problems of the vSLAM framework is the accumulation of drift that results in poor localization, especially for long distances. As a remedy, vehicles are forced to drive in loops and perform loop closures in order to compensate for possible localization errors, which restricts the vehicle's behavior.

In this paper we propose a localization framework which uses simple internal map data to localize the vehicle, without the restriction of driving loops or the requirement for any other complicated infrastructure. The basic assumption is that *visual odometry (VO)* can be made sufficiently accurate over short distances. Based on this assumption, which holds for the majority of the existing VO algorithms, we build a localization system that takes advantage of publicly available map data to provide accurate localization over time. Our framework takes as input a very rough initial GPS position of the vehicle (the initial GPS can be several hundred meters off), downloads the corresponding map of the surrounding area and localizes the vehicle after it has driven a certain minimum distance. A Monte Carlo Localization framework (MCL) is then initiated and tracks the vehicle's position using VO as input to the motion model and chamfer matching to evaluate the alignment of the different hypotheses to the map. As our experimental results indicate, we achieve state-of-the-art localization performance, overcoming traditional problems of VO systems, such as drift accumulation.

The paper is structured as follows. The following section discusses related work. Sections III, IV, V and VI present a detailed analysis of the individual components, namely VO estimation, OSM data processing, Chamfer matching and Monte Carlo Localization. Section VII lays out the combination of the individual components into a unified framework and describes the localization algorithm in detail. Finally, Section VIII demonstrates experimental results.

II. Related Work

Global localization is an essential part of every autonomous driving system and its robust performance is crucial for the stability and robustness of the system. To achieve this goal, several VO approaches have emerged in recent years, offering high quality localization performance even at

All authors are with Computer Vision Group, RWTH Aachen University. Email: {floros,leibe}@vision.rwth-aachen.de

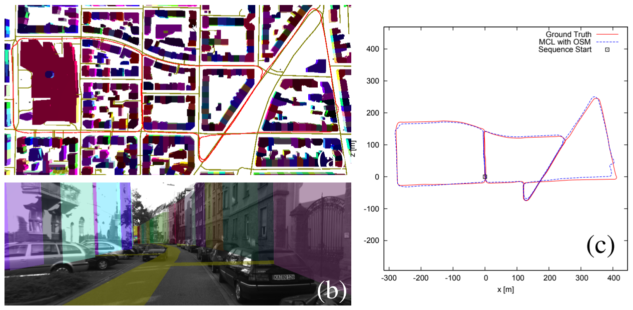


Fig. 1. Our approach incorporates input from a visual odometry component together with map information from OpenStreetMap (a) to improve the global localization accuracy of a vehicle. Chamfer matching is used to align the travelled path to the map of the environment over time. The localization accuracy greatly improves (c) using the proposed framework and allows the flow of semantic information from the OpenStreetMap to the vehicle (b).

real-time execution speeds. Most notably, the approaches by Kitt *et al.* [5] and more recently by Alcantarilla *et al.* [6] are the latest achievements in a long line of research. The basic pipeline of these methods includes a feature extraction stage, followed by feature matching, 3D reconstruction of a sparse point cloud, and finally estimation of the pose of the vehicle. A windowed bundle adjustment is typically run at the end. The main drawback of this category of methods is that drift is accumulated over time. As a result, after the vehicle has travelled a certain distance, the localization error becomes significant, making the localization result unusable over longer distances.

In parallel to the VO approaches, there are a number of methods that perform vSLAM [4], [7]. The vSLAM pipeline is similar to the VO one, with the difference that in the former the reconstructed 3D points are used as landmarks and their position is optimized together with the camera pose. To alleviate the problem of drift accumulation, the vehicles are either forced to drive in loops, or another vehicle needs to have mapped the same street before, so that the algorithm can recognize the same landmarks and thus compensate for errors in localization. However, driving in loops severely limits the vehicle's freedom of movement, and a pre-computed map of landmark image features requires extensive storage.

On the other hand, there has been research towards using map information that is already available in the form of 2D (*e.g.*, Google Maps, OpenStreetMaps) or 3D road maps (*e.g.*, Google Maps 3D) for localization. Hentschel *et al.* [8] propose the use of OpenStreetMap maps to facilitate robot localization and navigation. A cadastral map with the building footprints is extracted and the GPS position of the robot is used in an MCL framework to provide localization for the robot. Recently, Senlet *et al.* [9], [10] have presented a system that uses Google Static Maps to perform global localization. In particular, they perform a local reconstruction of the scene, using stereo depth maps, which they later on compare with the 2D map. Both of the aforementioned approaches show the potential that the use of map information can bring to the localization task. However, their systems require a manual initial localization and have only been tested in suburban areas where the environment is relatively simple and the paths they have travelled are short in length. In addition, these systems become overly complex by performing stereo depth estimation and local 3D reconstructions, thus increasing their computational complexity.

In this work, we propose a novel global localization approach that brings together the advantages of VO algorithms with useful map information in the form of street graphs. Our system requires only a very rough initial GPS estimate to automatically localize the initial position of the vehicle in a map with a radius of ~ 1 km. An MCL stage then performs the global localization task using Chamfer Matching to align the different location hypotheses to the street map.

Our approach extends the classical VO pipeline by incorporating additional map information from OpenStreetMaps. The system is structured as an MCL framework, where VO estimates the motion of the vehicle on a frame-to-frame basis. The output of VO is then fed into the motion model of the particle filtering algorithm, which produces a number of hypotheses (particles). Each particle, representing a different possible location of the vehicle, is then evaluated according to its alignment to the map. In the next chapters, we will present the individual components and their combination to form a unified framework and we will provide implementation details.

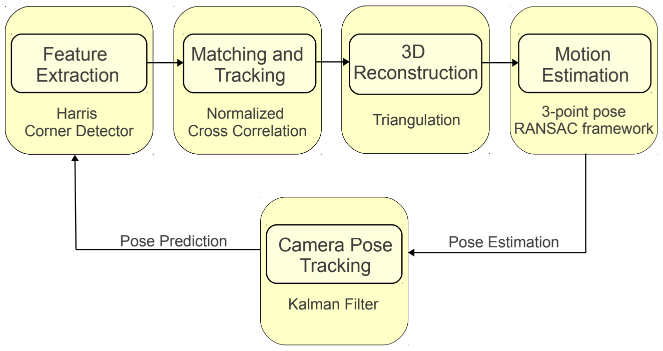


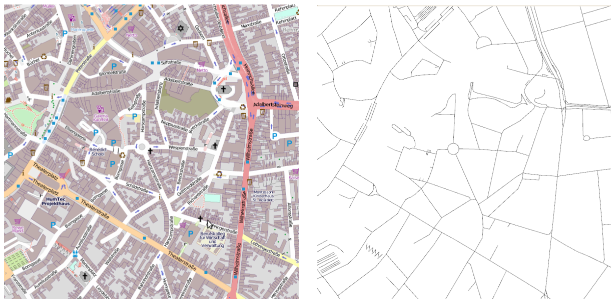
Fig. 2. **Visual odometry pipeline.** Features are extracted from the images and are then matched and triangulated to form a sparse 3D reconstruction. The camera pose is then estimated using a RANSAC framework and is tracked over time using a Kalman filter.

III. Visual Odometry (VO)

In the front-end of the VO component, feature extraction is run on the input images. Harris corners [11] are extracted from both the left and the right camera images. We chose this specific type of feature for three reasons. Firstly, Harris corners are reported to have high response stability and repeatability [12] under different image transformations. Secondly, they can be implemented very efficiently and they are therefore suitable for real-time applications. Finally, since no loop-closure is involved in our system, there is no need for more elaborate descriptors to account for scale invariance. We follow the approach of Nister *et al.* [13] and subdivide the images using a 10×10 grid, applying relative thresholds to each of the 100 resulting buckets, which imposes uniform feature coverage over the entire image.

For each of the detected Harris corners, an 11×11 pixel patch descriptor is computed. The descriptors of the left and the right image are then matched with each other and with the corresponding left and right images of the previous stereo frame using normalized cross-correlation. A match is considered as valid only if the features are mutual nearest neighbors [13]. The valid matches are then triangulated and the corresponding 3D points are generated.

In the final stage, the pose of the camera system is estimated from the triplets of reconstructed points [14]. A hypothesis for the camera pose is created from each of the triplets and is then fed into a RANSAC framework, which evaluates them based on the back-projection error measure, and the best hypothesis is selected. Since our VO component is feature-based, it is strongly affected by the presence of other moving objects in the scene. To alleviate the influence of these effects, a constant-velocity Kalman filter tracks the



(a) OSM Map (b) Street Graph

Fig. 3. **OpenStreetMap representation.** OSM map is shown at the left side, as it appears in the corresponding website with all the semantic information that it contains. The street graph extracted from the OSM map is shown at the right side.

camera position, extrapolating the camera path whenever the pose estimation algorithm output is inconsistent [15].

IV. OpenStreetMap Data Representation

OpenStreetMap (OSM) data can be accessed via the corresponding website, and the user can download the map of an area of interest by specifying a bounding box $\mathbf{b}$ in terms of longitude and latitude, $\mathbf{b} = (lat_{min}, lon_{min}, lat_{max}, lon_{max})$. The map is given in XML format and is structured using three basic entities: *nodes*, *ways* and *relations* [16].

The node $\mathbf{n}$ represents a point element in the map and is defined by its GPS coordinates, longitude and latitude $\mathbf{n} = (lat, lon)$. Linear and area elements are represented by ways $\mathbf{w}$, which are formed as a collection of nodes $\mathbf{w} = \{\mathbf{n}_i\}_{i=1\dots k}$. The relations $\mathbf{r}$ are used to represent relationships between the aforementioned geometric entities to form more complicated structures.

The most interesting map structures such as streets and buildings are modeled as ways and can be extracted from the XML data in order to form the street graph of the area. This representation is very convenient for later processing and in particular for the shape matching component that we introduce in the next section.

V. Chamfer Matching

Chamfer matching (CM) [17] is a popular algorithm to match a query edge map $Q = \{\mathbf{q}_i\}$ to a template edge map $T = \{\mathbf{t}_i\}$ by finding the homogeneous transformation $\mathbf{H} = (\theta, t_x, t_y)$, $\mathbf{H} \in SE(2)$ which aligns the two edge maps. The objective can be formalized as the following minimization problem:

$$\hat{\mathbf{H}} = \arg \min_{\mathbf{H} \in SE(2)} d_{CM}(\mathbf{W}(Q;\mathbf{H}), T), \tag{1}$$

where $\mathbf{W}(Q;\mathbf{H})$ and $d_{CM}(Q,T)$ are defined as

$$\mathbf{W}(Q;\mathbf{H}) = \mathbf{H} \cdot Q, \tag{2}$$

$$d_{CM}(Q,T) = \frac{1}{|Q|} \sum_{\mathbf{q}_i \in Q} \min_{\mathbf{t}_j \in T} |\mathbf{q}_i - \mathbf{t}_j|. \tag{3}$$

The CM algorithm has two basic advantages. Firstly, it is robust to small misalignments and deformations by finding the best matching alignment even if this does not exactly fit the query to the template. Secondly, the matching cost can be computed very efficiently with the use of distance transform images. In particular, the computation of the distance transform requires two passes over the image [18] and makes the evaluation of (3) feasible in linear complexity $O(n)$. Several approaches [19], [20] try to increase the robustness of Chamfer matching by incorporating information from edge orientations. To realize that, they either quantize the edges into several orientation planes, evaluated separately, or they augment the matching cost function to include an additional orientation term. Although these approaches improve the results, this improvement comes with additional computational cost.

Recently, Liu *et al.* introduced Fast Directional Chamfer Matching (FDCM) [21], an algorithm which produces state-of-the-art results by incorporating edge orientation information, being at the same time $45\times$ faster than the other Chamfer matching alternatives. The power of this algorithm comes from the fact that the edge maps are converted from a set of 2D points to a collection of line segments. This conversion reduces the cardinality of the edge set from n points to m line segments, where $m \ll n$. The edge orientation information is added to the matching cost function and a 3D distance transform is computed to make the evaluation of the matching cost functions faster.

FDCM has a number of properties that render it suitable for our application. Firstly, FDCM takes as input a set of line segments, which exactly matches the street graph representation of the OSM data. Thus, no extra representation conversion is required. Secondly, FDCM is very efficient, allowing for matching local VO paths to OSM maps of a large area (which correspond to edge maps of $\sim$ 5MPixel).

VI. Standard Monte Carlo Localization

Monte Carlo Localization (MCL) [22] or the *particle filter* algorithm estimates the vehicle's pose $\mathbf{x}_t$ at time t recursively from a set of M particles $\mathbf{X}_t = \{\mathbf{x}_t^{[1]}, \mathbf{x}_t^{[2]}, \dots, \mathbf{x}_t^{[M]}\}$. The algorithm samples from a motion model, having the current particles as initial points. Each of the sampled particles is then evaluated under the observation model and an importance weight is assigned to it. The last step includes a re-sampling of the particles according to their weights and a weight normalization. A single location hypothesis is finally extracted by taking a weighted average of the particles. In this specific application, the vehicle pose $\mathbf{x} = (x, y, \theta)$ consists of the position and the orientation of the vehicle in the 2D map. The standard odometry model [22] is used as a motion model, whereas the measurement model is given by an exponential distribution over the Chamfer matching cost function as follows:

$$w^{[m]} = \lambda \exp^{-\lambda \cdot d_{CM}^{[m]}}, \quad m \in 1\dots M. \tag{4}$$

The distribution parameter λ is determined experimentally.

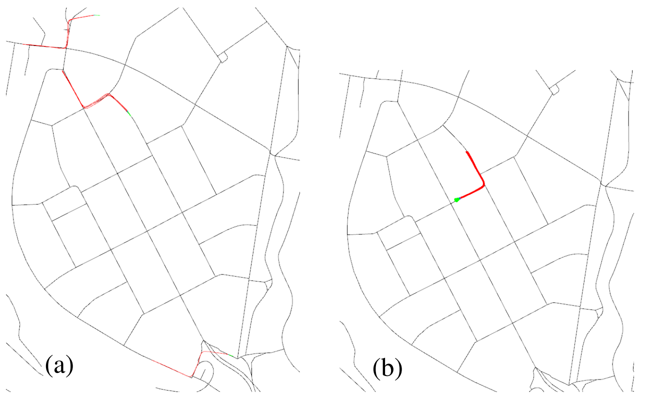


Fig. 4. (a) Visualization of the initial particle distribution. Chamfer matching is exhaustively run over the entire street graph and the particles are positioned at the most prominent matches. (b) Illustration of the MCL propagation stage. Each particle (green) holds a local history of the previous VO estimates (red).

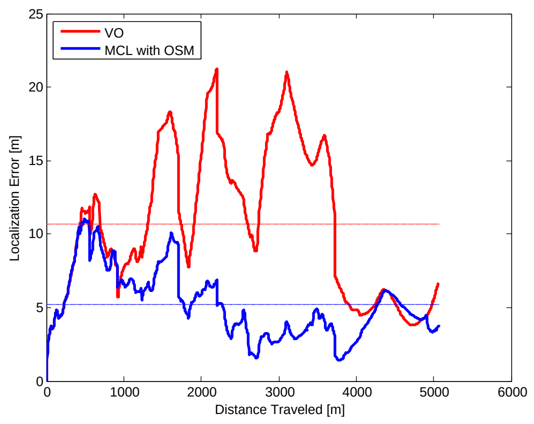


Fig. 5. **Average localization error** over all 11 test sequences. The red curve shows the error of the standard VO and the blue curve show the localization error of the proposed localization algorithm. The horizontal lines illustrate the average error over the travelled distance for each of the approaches.

VII. Localization Using Map Information

While VO approaches localize a mobile platform with high accuracy over short distances, they typically accumulate drift over time, which can result in very large localization errors after a couple of kilometers have been travelled by the vehicle. The proposed approach takes as input a rough GPS position of the vehicle and downloads a map from the OSM website which is centered at that position (see Fig. 3(a)). Once the map has been downloaded, the street graph is extracted and is converted to a set of line segments, as illustrated in Fig. 3(b). This set of line segments will form the template edge map for the Chamfer matching that will take place in the later steps. For this reason, the line segments are quantized into 60 different orientation channels and the distance transforms are pre-computed prior to the initialization of the vehicle pose tracking. This computation is done only once in the beginning of the algorithm; later on, we use these distance transforms to evaluate the matching functions for the different path queries.

The algorithm starts by running VO to track the pose of the vehicle over a specific travelled distance of k meters, so that an initial path is created. After the vehicle has travelled at least k meters and has also accumulated a minimum turning angle θ_{min} along its path, a first localization of the path is performed on the map. For this, the travelled path is transformed into a set of line segments, which connect the subsequent vehicle positions to form the query edge map. The query edge map is then matched exhaustively over all possible positions and orientations against the template edge map using FCDM, and a set of possible localization hypotheses is computed (see Fig. 4(a)). These hypotheses are then used to initialize the MCL module by spreading the particles according to the hypothesis weights. The different particles correspond to different possible locations of the vehicle and each of them keeps a path history of the d previous positions of the vehicle. This local path is used to help the evaluation of each of the particles under the observation model. At this point, MCL starts its iterations and samples a new position for each particle based on the local VO estimate and the particle's current position. The particles move to their new positions and each of them is evaluated based on its chamfer matching cost to the map, using the measurement model of (4). This way, particles that are not well aligned to the map are weighted with low scores and particles which conform to the map are weighted higher. The re-sampling step then moves the mode of the particle distribution towards the most prominent particle positions and the scheme is iterated (see Fig. 4(b)).

VIII. Experimental Results

A. Dataset

We evaluate our method on the KITTI vision benchmark suite [23]. The dataset consists of 22 stereo sequences, captured at 10Hz with a resolution of 1392×512 pixels over a driving distance of 39.2 km. Ground truth poses are given from a GPS/IMU localization unit for the first 11 sequences. We have augmented the dataset by downloading the corresponding OSM maps which include the area of ~ 1 km radius around the starting position of the vehicle. These maps are used for extracting the street graphs used by our algorithm.

B. Comparison to Standard VO

In a first round of quantitative experiments, we evaluate the performance improvement that our proposed system yields in comparison to a (manually initialized) standard VO algorithm. In particular, we compare a baseline system which implements the VO pipeline as described in Sec. III to the MCL system of Sec. VII, where map information is used to compensate for possible drifts of the VO algorithm. Except where noted otherwise, we initialize the particle filter algorithm after the vehicle has travelled a minimum distance of $k = 400$ m and has accumulated a turning angle $\theta_{min} = \pi$. The temporal window that each of the particles keeps as a local history has a length of $d = 250$ frames and we sample 500 particles in the MCL framework. Finally, the parameter λ is set to 0.01.

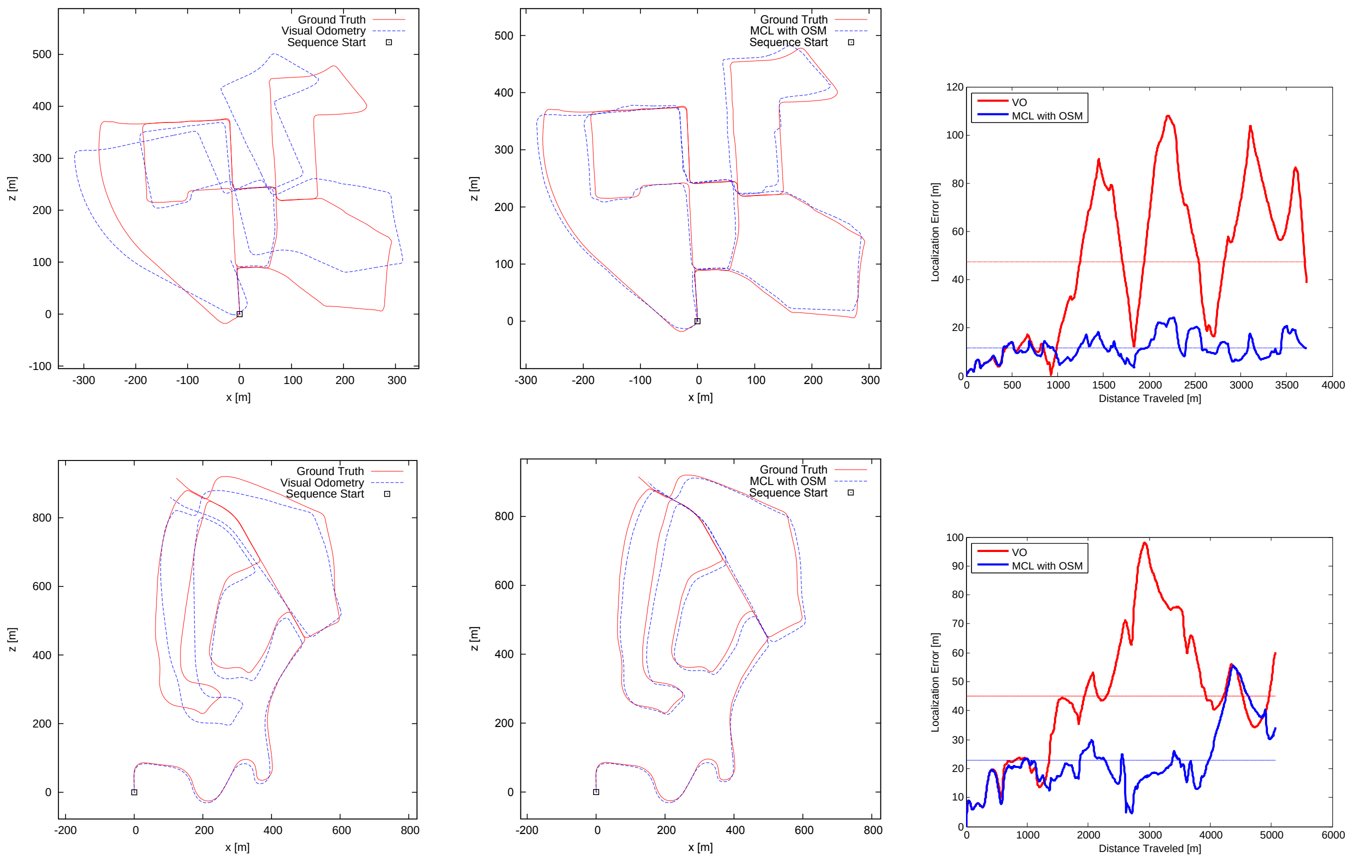


Fig. 6. **Quantitative and qualitative results for Sequence 00 and 02 of KITTI.** The first row shows the results for Sequence 00. (From left to right) The estimated path of the standard VO, the estimated path of our algorithm and the localization error comparison between the two methods. In the second row the same plots for Sequence 02 are illustrated.

As can be seen in Fig. 5, the proposed framework outperforms the standard VO algorithm, producing a consistently lower localization error. In particular, the proposed MCL framework yields an average error of 5.19 m in comparison to 10.65 m of the standard VO algorithm. This demonstrates that localization accuracy can increase significantly with the use of map information and its incorporation into the MCL system. In order to further underline the improvement that our system brings to localization performance, we illustrate in Fig. 6 quantitative and qualitative results from two longer sequences of the dataset, where the vehicle has driven 3.7 km and 5.1 km, respectively. The path visualizations show that our system is capable of compensating for the drift that the visual odometry locally accumulates by assessing the quality of the fit that the travelled path has with the underlying map. In addition, it should be noted that our initial assumption also holds that standard VO can produce high-quality results for short-range distances. Over longer distances, however, even though standard VO is able to reconstruct the topology of the traversed path, it often gives significant localization errors due to the accumulated drift. Our algorithm, on the other hand, is able to keep the localization error consistently in a low range, making the whole system more robust and stable. It is interesting to note that the proposed method can run on top of any visual odometry algorithm, providing further localization accuracy improvements. Another advantage of the method is that the uncertainty of the particle filter can be used as a confidence metric in cases where the street graph includes self-similarities (*e.g.*, Manhattan-like streets, highways, *etc*.). In such cases, a global re-localization can be afforded to run in parallel at a lower frequency to recover from spurious local minima during the localization procedure.

C. *Comparison to Other Localization Methods*

Although the KITTI benchmark provides an evaluation procedure for VO algorithms, the evaluation metrics used in that benchmark are not suitable for evaluating the global localization task. In particular, evaluation is performed in a windowed fashion, measuring rotation and translation errors at a fixed distance. Therefore, this evaluation places more emphasis on whether the topology of the actual path is recovered, rather than on the actual global position of the vehicle at a specific point in time. However, in order to demonstrate the improvement that our method brings in comparison even to the best performing algorithms of this benchmark, we present a qualitative comparison of one of the leading algorithms [24] to our method on a challenging sequence from the KITTI test set. Fig. 7 illustrates that our method can improve upon the state-of-the-art regarding

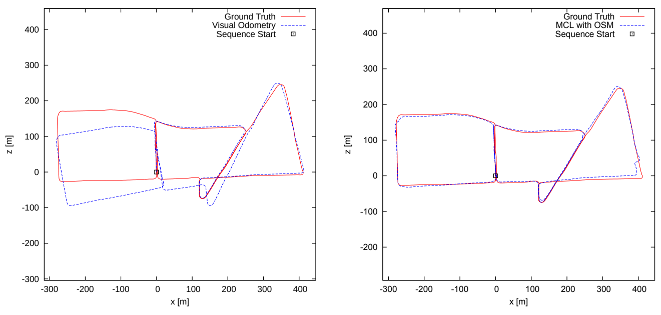


Fig. 7. **Qualitative comparison to [24].** The left plot shows the result of [24] for the Sequence 13 of KITTI. Our result for the same sequence is shown on the right.

localization accuracy and that the use of additional map information significantly reduces drift.

D. Run-time Efficiency

Our system is composed of several components, each of them contributing to the run-time of the whole pipeline. In detail, the VO framework of Sec. III takes 0.24 s per frame on average to give out a new pose estimate. The computation of the distance transforms, which is done once when downloading the map (in parallel to the time the vehicle needs to cover the required initial distance), takes 11.5 s on average. The initial vehicle localization (also done only once) takes 15.6 s on average. The evaluation of the particles inside the MCL framework takes only 0.02 s, adding only a minimal cost on top of the VO execution time. The approach has been implemented in C++ and runs on a single processor core. With careful optimization and parallelization of the implementation, we are confident it can be made real-time capable.

IX. Conclusions and Future Research

We have presented a novel approach for globally localizing a vehicle's position. Our approach combines a classical visual odometry pipeline with map data from OpenStreetMaps in a unified framework, achieving improved performance on a set of challenging sequences. We have described the algorithmic and implementation details of our method and we have thoroughly evaluated it using data from a demanding benchmark. Our results show that the proposed system is able to provide fully automatic global localization at a low infrastructural cost (only street graphs are necessary) and that it is able to stabilize odometry against drift over long travel distances.

In the future we plan to further investigate the use of maps to facilitate several other tasks, such as understanding of the environment and benefiting from the large amount of additional information that is stored in such maps.

Acknowledgments: This research has been funded, in parts, by ERC Starting Grant CV-SUPER (ERC-2012-StG-307432) and by the cluster of excellence UMIC (DFG EXC 89).

References

[1] S. Thrun, "Toward robotic cars," *Communications of the ACM*, vol. 53, no. 4, pp. 99–106, 2010.

[2] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Kolter, D. Langer, O. Pink, V. Pratt, *et al.*, "Towards fully autonomous driving: Systems and algorithms," in *IEEE Intelligent Vehicles Symposium*, 2011.

[3] A. Zamir and M. Shah, "Accurate image localization based on google maps street view," in *European Conference on Computer Vision*, 2010.

[4] H. Lategahn, A. Geiger, and B. Kitt, "Visual SLAM for Autonomous Ground Vehicles," in *IEEE International Conference on Robotics and Automation*, 2011.

[5] B. Kitt, A. Geiger, and H. Lategahn, "Visual Odometry based on Stereo Image Sequences with RANSAC-based Outlier Rejection Scheme," in *IEEE Intelligent Vehicles Symposium*, 2010.

[6] P. Alcantarilla, J. Yebes, J. Almazán, and L. Bergasa, "On combining visual SLAM and dense scene flow to increase the robustness of localization and mapping in dynamic environments," in *IEEE International Conference on Robotics and Automation*, 2012.

[7] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid, "RSLAM: A system for large-scale mapping in constant-time using stereo," *International Journal of Computer Vision*, vol. 94, no. 2, pp. 198–214, 2011.

[8] M. Hentschel and B. Wagner, "Autonomous Robot Navigation Based on OpenStreetMap Geodata," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2010.

[9] T. Senlet and A. Elgammal, "A framework for global vehicle localization using stereo images and satellite and road maps," in *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011.

[10] ——, "Satellite Image Based Precise Robot Localization on Sidewalks," in *IEEE International Conference on Robotics and Automation*, 2012.

[11] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey Vision Conference (AVC'88)*, 1988.

[12] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.

[13] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *Journal of Field Robotics*, vol. 23, no. 1, pp. 3–20, 2006.

[14] D. Nistér and H. Stewénius, "A minimal solution to the generalised 3-point pose problem," *Journal of Mathematical Imaging and Vision*, vol. 27, no. 1, pp. 67–79, 2007.

[15] A. Ess, B. Leibe, K. Schindler, and L. Van Gool, "Robust Multi-Person Tracking from a Mobile Platform," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 10, pp. 1831–1846, 2009.

[16] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, 2008.

[17] H. Barrow, J. Tenenbaum, R. Bolles, and H. Wolf, "Parametric Correspondence and chamfer matching: Two new techniques for image matching," in *Int. Joint Conference on Artificial Intelligence (IJCAI'77)*, 1977.

[18] P. F. Felzenszwalb and D. P. Huttenlocher, "Distance Transforms of Sampled Functions," *Theory of Computing*, vol. 8, no. 1, pp. 415–428, 2012.

[19] D. Gavrila, "Pedestrian detection from a moving vehicle," in *European Conference on Computer Vision (ECCV'00)*, 2000.

[20] J. Shotton, A. Blake, and R. Cipolla, "Multiscale Categorical Object Recognition Using Contour Fragments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 7, pp. 1270–1281, 2008.

[21] M. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa, "Fast directional chamfer matching," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*, 2010.

[22] W. Burgard, D. Fox, and S. Thrun, *Probabilistic robotics*. MIT Press, 2005.

[23] A. Geiger, P. Lenz, and U. R., "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[24] C. Beall, B. Lawrence, V. Ila, F. Dellaert, *et al.*, "3D reconstruction of underwater structures," in *Intelligent Robots and Systems*, 2010.