

IMPROVING GPS TRAJECTORIES USING 3D CITY MODELS AND KINEMATIC POINT CLOUDS

Y. Dehbi^{1,*}, L. Lucks³, J. Behmann², L. Klingbeil¹, L. Plümer⁴

¹ Institute of Geodesy and Geoinformation, University of Bonn, Germany, (dehbi, klingbeil)@igg.uni-bonn.de

² Institute of Crop Sciences and Resource Protection, University of Bonn, Germany, behmann@uni-bonn.de

³ Fraunhofer (IOSB) Institute of Optronics, System Technologies and Image Exploitation, Germany, lukas.lucks@iosb.fraunhofer.de

⁴ Faculty of Geosciences and Environmental Engineering, Southwest Jiaotong University, Chengdu, China, pluemer@swjtu.edu.cn

KEY WORDS: Mobile mapping system, ICP, City model, Classification, SVM, Point cloud, Trajectory

ABSTRACT:

Accurate and robust positioning of vehicles in urban environments is of high importance for many applications (e.g. autonomous driving or mobile mapping). In the case of mobile mapping systems, a simultaneous mapping of the environment using laser scanning and an accurate positioning using GNSS is targeted. This requirement is often not guaranteed in shadowed cities where GNSS signals are usually disturbed, weak or even unavailable. Both, the generated point clouds and the derived trajectory are consequently imprecise. We propose a novel approach which incorporates prior knowledge, i.e. 3D building model of the environment, and improves the point cloud and the trajectory. The key idea is to benefit from the complementarity of both GNSS and 3D building models. The point cloud is matched to the city model using a point-to-plane ICP. An informed sampling of appropriate matching points is enabled by a pre-classification step. Support vector machines (SVMs) are used to discriminate between facade and remaining points. Local inconsistencies are tackled by a segment-wise partitioning of the point cloud where an interpolation guarantees a seamless transition between the segments. The full processing chain is implemented from the detection of facades in the point clouds, the matching between them and the building models and the update of the trajectory estimate. The general applicability of the implemented method is demonstrated on an inner city data set recorded with a mobile mapping system.

1. INTRODUCTION

The accurate determination of the trajectory of a vehicle in an urban environment is a key requirement in many applications. In the example of autonomous driving, it is necessary to localize a vehicle in real-time within a map in order to navigate it to the desired location. In the example of mobile mapping, laser scanners and cameras mounted on a vehicle are used to generate high precision 3D information about the surrounding of the moving car. Here the trajectory, which may be derived in a postprocessing step, is used to register all sensor data within a global coordinate system.

In both examples, Global Navigation Satellite Systems (GNSS) such as GPS, Glonass, Galileo or Beidou are the key to the absolute accuracy of the trajectory, while the integration of other sensors, such as Inertial Measurement Units (IMU), odometers, cameras or scanners increase the relative accuracy and smoothness of the trajectory or help to bridge GNSS outages. Regardless of the type of the used GNSS processing method (carrier phase based differential GNSS usually used in mobile mapping or code based GNSS usually used in autonomous driving), the urban environment imposes strong systematic errors on the observations due to multipath effects and non-line of sight signal reception. Even in differential GNSS processing these errors can be in the order of several meters. It is also possible, that a GNSS position cannot be processed at all over a longer period of time and that the other relative sensors cannot prevent the trajectory estimation from drifting unboundedly. Figure 1 (left) illustrates an example of such errors visualizing a trajectory which has been recorded in our university campus. It can be stated that the trajectory inter-

sects with building elements which do not match the true driving path.

One option to increase the absolute accuracy in weak GNSS conditions is to integrate prior knowledge about the environment into the estimation process. In map matching, prior knowledge is represented by a street map. The initially estimated trajectory is matched against this map on the basis of coordinates or curvature profiles. The transformation of this match is then used to improve the initial estimate.

In this contribution we show, that it is similarly possible to use a 3D model of the environment, in the form of LOD1 city models, as prior knowledge. By comparing the 3D information based on some initially estimated trajectory and the corresponding laser scanner observations with the georeferenced 3D model from a data base, we can update the estimate and improve its absolute accuracy. The complementarity of both GNSS and 3D building models is a key issue here. The building models contribute to a high accuracy in shadowed cities, exactly where GNSS signals are usually weak or disturbed. We implemented the full processing chain from the detection of facades in the point clouds, the matching between them and the building models and the update of the trajectory estimate. We show the general applicability on a data set, which has been recorded with a mobile mapping system. It can be shown that the inaccurately false estimated trajectory from the left side of Figure 1 has been updated and corrected integrating GNSS information and the building model of the surrounding area on the right side.

Although this is demonstrated here as a postprocessing step, it is also possible to integrate this in an online fashion, as it would be necessary for the example of autonomous driving. In this case the transformation needs to be calculated based on the very last

*Corresponding author

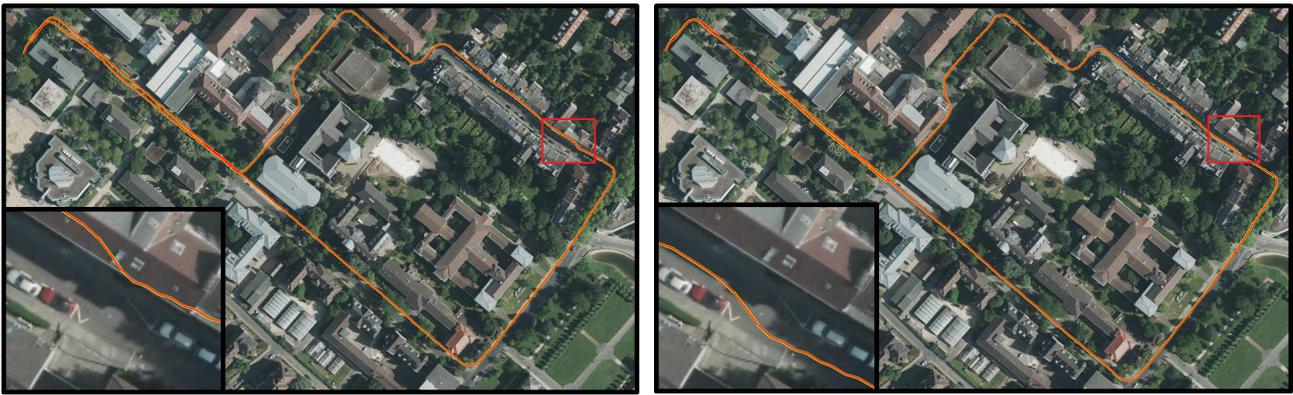


Figure 1. Trajectory recorded with a mobile mapping system in the university campus Poppelsodrf in Bonn/ Germany. The trajectory intersects with buildings (Left). The true trajectory corrected incorporating 3D city model of the surrounding area (Right).

point cloud section, generated using the trajectory output of an online estimator, such as the Kalman Filter. The transformation would be treated as an absolute measurement and integrated into the filtering algorithm accordingly.

The remainder of this paper is structured as follows: Section 2 gives a review on the most related research. Section 3 presents the theoretical background of the used methods, whereas section 4 introduces our performed experiments and discusses the achieved results. The paper is summarized and concluded in section 5.

2. RELATED WORK

For our approach three fields are relevant: Trajectory estimation in mobile mapping systems, 3D point cloud interpretation and map matching.

Trajectory estimation in mobile mapping systems. In mobile mapping systems the trajectory is usually estimated by a combination of GNSS and inertial measurement unit. The GNSS processing method (differential or absolute, code or carrier phase based) and the local GNSS conditions determine the absolute position accuracy of the trajectory and the quality of the IMU determines the accuracy of the orientation component and also the length of a timespan in which the trajectory can still be estimated, even though GNSS is interrupted or severely disturbed. Because the laser scans are usually transformed to the global coordinate system in an open loop manner using the trajectory parameters, any error in the trajectory estimation directly effects the quality of the resulting point cloud (Glennie, C., 2007). Because of this, mobile mapping systems mostly use high grade IMUs with relative carrier phase based GNSS processing. A good overview about mobile mapping technology can be found in Puente, I. et al. (2013).

One way of overcoming the problem of systematically wrong or missing absolute position data is to incorporate observations from the object space, such as the scan points or images into the estimation procedure. It is quite common to use control points along the path to adjust and evaluate the trajectory and therefore, the resulting point cloud (Gräfe, G., 2007; Clancy, 2011). Eckels and Nolan (2013) use multiple passes with the system along the same trajectory to detect and adjust errors resulting from GNSS insufficiencies. Another option to improve the trajectory accuracy is to estimate it simultaneously with the 3D map by integrating the scanner data and the navigation sensor information in a single adjustment (SLAM - Simultaneous Localization and Mapping).

This is a common technique in robotic applications (Thrun et al. (2005); Stachniss (2009)) and has also been applied to 3D laser scanning in urban environments (Nüchter et al. (2007)).

The presented work can be seen in the context of the aforementioned methods with the building models serving as 'control planes' with known absolute parameters. An estimation of the map, as it is part of SLAM approaches, is not necessary.

3D point cloud interpretation. The interpretation of LiDAR point cloud for building modeling has been intensively investigated. Niemeyer et al. (2012) applied Conditional Random Fields (CRF) for the classification of airborne LiDAR point clouds. CRFs enable the incorporation of contextual information for the discrimination between five classes: building, low vegetation, tree, natural ground, and asphalt ground. To this aim, intensity information as well as geometric features have been exploited. The latter describes the local context of a single point by its position, its normal and the distribution of positions and normals of its parametrized neighborhood. A data-driven approach is followed by Zhou and Neumann (2008) developing features for the reconstruction of buildings in an efficient and robust way. In this context, trees are identified and eliminated in a first step, then ground points and roof patches are extracted using a distance-based region growing. Finally, simple and correct polygonal mesh models are generated based on the roof boundaries in a bottom-up manner. Chehata et al. (2009) proposed a feature selection method for urban classification using random forests. They separated the designed features into five groups: height-based, waveform echo-based, eigenvalue-based, local-plane-based and full-waveform LiDAR features. Rusu et al. (2009) introduced the fast point feature histogram (FPFH) as discriminative descriptor for the detection of objects in point clouds. Rouhani et al. (2017) proposed a method for segmenting textured meshes extracted from multi-view stereo based on Markov Random Field (MRF).

Map matching. The integration of street networks and preliminary trajectories turns out to be a good tool for the improvement of the recorded positions. In this context, Haurert and Budig (2012) developed a method for matching GNSS trajectories with road data dealing with missing roads. They applied a hidden Markov model (HMM) together with a discrete set of candidate matches and off-road candidates for each point of the trajectory. Osogami and Raymond (2013) applied inverse reinforcement learning for the estimation of the importance of the number of turns compared to the travel distance. The estimated importance is then incorporated in the transition probability of an HMM such as the case of Haurert and Budig (2012). Also based

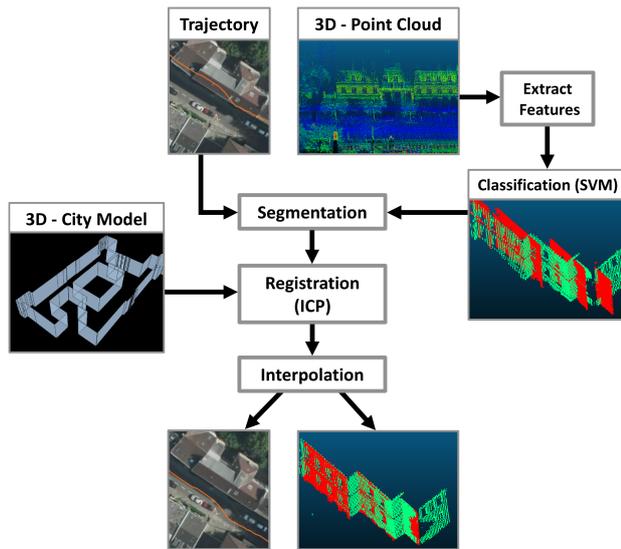


Figure 2. Workflow.

on HMMs, Viel et al. (2018) proposed a map-matching algorithm which is able to deal with noisy and sparse cellular fingerprint observations instead of GNSS positions.

3. METHODOLOGY

In this section we present our approach from a methodological point of view and give insights into the different components of our method. The work-flow is depicted in Figure 2. The input data consists of a 3D kinematic point cloud together with a GNSS track of a region of interest. Besides, a 3D city model of the surveyed region is derived on-demand via a web feature service (WFS). More details on the data and the experimental settings will be provided in section 4.1. In a first step, the point cloud is interpreted point-wise using supervised machine learning methods. In this sense, a discrimination between facade points and remaining points is achieved. This step is elaborated in section 3.1. To this aim, pre-designed features are extracted from the point cloud beforehand. It can be seen, that the point cloud is characterized by a drift which is obvious comparing the outward (green) and the return (red) journey in Figure 2. In order to overcome this deficiency, a piecewise segmentation of the point cloud based on the GNSS trajectory is performed, enabling a consecutive registration of each segment with the city model using the iterative closest point method (ICP) (Chen and Medioni, 1992; Besl and McKay, 1992). The main idea of our approach is to combine the inaccurate GNSS track and the point cloud, which has been initially georeferenced using that track, with the more accurate information from the 3D city model. This leads to an improvement of the georeferencing of the captured scene and the trajectory as well. To this end, the precise 3D city model is incorporated in ICP. This model represents an important background knowledge for finding appropriate matching points based on a learned classifier which is the topic of section 3.1. The matching step is elaborated in section 3.2. For each segment, we acquire a local transformation between the point cloud and the corresponding building model. For a seamless transition between underlying consecutive segments, we performed an interpolation step smoothing the drift and improving the trajectory simultaneously which will be explained in section 3.3.

3.1 Point-based facade classification

The mobile mapping system provides a dense 3D point cloud (mean point distance in the order of cm) as depicted on the top of Figure 6. A more detailed data description is given in subsection 4.1. In order to enable an informed sampling of appropriate points for the registration later on, a pre-classification of the input 3D point cloud is of great importance. This step leads to an efficient selection of matching points in the sense of ICP. To this aim, an active sampling of points belonging to facades is a key issue. In this context, we address this task as a classification problem where a point-based discrimination is enabled leading to a semantic segmentation of the point cloud. In this manner, we are able to differentiate between building points and remaining points using Support Vector Machines (SVM) (Vapnik, 1998; Schölkopf and Smola, 2002) as a robust classifier in a supervised machine learning task.

For each point in the 3D point cloud, we derived a set of geometric features for a binary classification. We applied features which describe the local context of a single point by its position, its normal and the distribution of positions and normals of its k -neighborhood. In this context, the deviation between a surflet (Wahl et al., 2003), i.e. 3D point together with the normal vector derived from a plane approximation of its neighborhood, and the azimuth is considered. In order to assess the planarity of the neighborhood of a given point, we used the curvature κ (Pauly et al., 2002) as measure which is calculated from the eigenvalues λ_i of the principal component analysis of the according neighborhood via this equation:

$$\kappa = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2}. \quad (1)$$

We used the point cloud library¹ to derive the radiometric features. Five neighborhood regions from 0.2m to 1.0m have been considered. Furthermore, the height variation of the individual points to their neighboring points is taken into account. To this aim, the height differences between the considered point and its neighboring points is calculated. The mean square of these differences is considered as a measure and feature reflecting the height variance. We determine the neighborhood in two different ways and for two different sizes (0.5m and 1.0m). Once we determine them using only the x and y coordinates of the points and once we use all the 3D information, so finally we get four different features. As additional feature, we used the point density in the xy plane for each point, again for two neighborhood sizes of 0.5 and 1.0m.

In contrast to ground points, facade points are characterized by a larger height variability. This property is reflected by the calculation of the ratio of the points with equal heights and all points in the neighborhood. Two points are defined to have the same height, if the difference is lower than 0.1m. In order to distinguish between facade points and those belonging to smaller objects (e.g. cars) the point heights relative to the ground are of great benefit. Since our study area is rather flat, we approximated the ground level (street surface) based on the mean of the lowest points in our training data. In the more general case the information on the ground surface (e.g. a digital terrain model) can be incorporated if available. Otherwise, the height of the ground has to be determined from the point cloud. Therefore, a feature for the approximation of height difference to the ground surface, as proposed by Chehata et al. (2009) can be calculated.

¹<http://pointclouds.org>

Name of feature	number of values
Deviation angle	5
Height difference to approx. ground	1
Height variance in local neighborhood	4
Point density	2
Mean curvature in local neighborhood	5
Intensity	1
Ratio of equally high points	2

Table 1. The used features for the point-based classification.

Beside the geometric features, we make use of intensity information measured by the mobile mapping system. This is in particular important to discriminate between vegetation and building points based on the surface properties. All in all 20 features have been used to classify building points. Table 1 shows an excerpt of the most relevant features together with the number of their values according to the used neighborhoods. As mentioned before, the classification results lead to appropriate sample points for the subsequent registration using the iterative closest point approach.

3.2 Segmentation and registration of the point cloud

The next step is to perform a segmentation of the point cloud. As the underlying point cloud is not consistent, a registration based on the whole recorded scene would improve the georeferencing but would not deal with the drifts within the point cloud. Thus, the idea is to take the temporal information into account and to partition the point cloud into several segments oriented by the trajectory. In this manner, a better local matching with the according 3D building model can be achieved since each segment is registered separately.

The registration itself is performed following the ICP algorithm of Besl and McKay (1992). The basic idea consists in matching two sets of points S and D based on the corresponding points and the iterative calculation and revision of a transformation between the two point sets. The transformation is depicted in equation 2 where p_S and p_D represent the vector of points in the according coordinate systems. μ_S and μ_D are the centroids of the point sets S and D respectively whereas R stands for a 3D rotation matrix (Förstner and Wrobel, 2016).

$$p_D = R(p_S - \mu_S) + \mu_D. \quad (2)$$

We used rotation matrices based on quaternions, which have significant advantages such as reducing the rotation axes and omitting the trigonometric terms. The determination of the rotation quaternions has been performed by singular value decomposition (SVD) (Förstner and Wrobel, 2016).

The ICP algorithm builds upon two given point sets which have to be matched. In our case, however, we have to deal with one point set which has to be directed according to a 3D building model. Thus, we make use of the so called point-to-plane ICP instead (Low, 2004). Figure 3 depicts the main idea of this method. In this context, building facades are modeled as planes. A matching point (red) is a point lying on the plane with a minimal distance to the according point (green) from the point cloud. Further, each matching point has to be contained in the polygon defined by the facade boundary (light blue). In our context, relevant facade planes are selected based on a bounding box query around points of interest from the point cloud. As mentioned in section 3.1, appropriate points for ICP are pre-filtered based on a classification step. We used trimmed ICP as a robust version of ICP which do

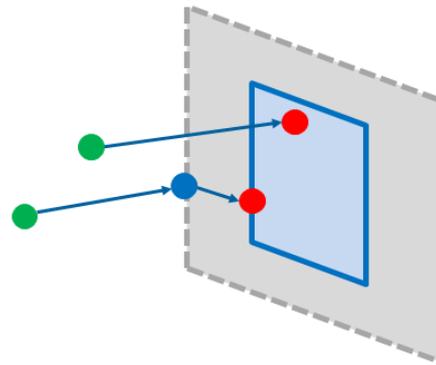


Figure 3. Finding matching points using point-to-plane ICP. Source points (green) from a point cloud are associated to matching points (red) on the facade plane with minimal distance. Matching points are contained in the facade polygon (light blue). Despite minimal distance, the blue point is not a matching point.

not take matching points with a high distance to source points into account (Chetverikov et al., 2002).

The point cloud is characterized by an accuracy of several meters. In this context, a mismatching between points from the point cloud and the according building facade has to be avoided and, hence, a false transformation can be prevented. This is in particular important after the partitioning of the point cloud into small segments which possibly lead to a spatial information loss. A lack of structured spatial information is especially noticed by straight ahead driving where mostly facades parallel to the driving direction are captured. Together with the inaccurate georeferencing of the point cloud, there is the risk that the approach will not be able to sufficiently transform such segments. The ICP algorithm cannot find a unique assignment between point cloud and model and may converge to an inaccurate solution. In order to overcome this problem, the points of each segment are pre-transformed based on the computed transformation of the previous segments in a sequential manner. We assume that the first segment contains highly structured spatial information and can thus be transformed with sufficient precision. The resulting points have a smaller offset to the building models so that the approach can robustly find the necessary ICP correspondences for each segment. The impact of this modification will be explained in section 4.2 where outliers in the sense of ICP correspondence can robustly deal with.

3.3 Interpolation after the piecewise segmentation

The registration provides a set of transformation parameters for each segment of the point cloud. Unfortunately, the ICP solutions may slightly differ for each segment. If the points of each segment are transformed using only the local transformation, overlapping areas or gaps in the point cloud may occur between the segments. Figure 4 on the left illustrates overlapped segments in their transition area. In order to overcome these artifacts, we performed an interpolation between two subsequent segments i and j . Each point p_S is transformed twice according to the transformations T_i and T_j from the corresponding ICP solution. For p_S , two different solutions (p_{T_i} and p_{T_j}) are obtained, which are then weight-averaged for the final solution using the following equation:

$$p_T = p_{T_i} w_i + p_{T_j} w_j, \quad (3)$$

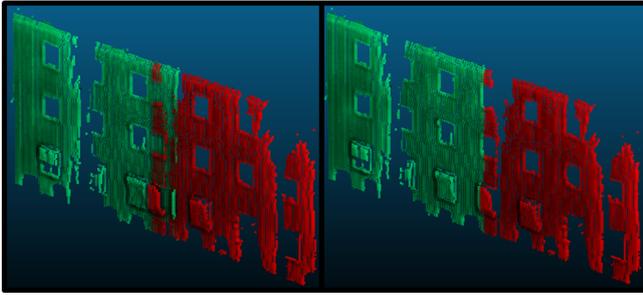


Figure 4. Excerpt of the transformed point cloud in the transition between two segments without interpolation (left) and with interpolation (right).

the weights w_i and w_j are calculated as follows:

$$w_i = \frac{d_i}{d_i + d_j} \text{ and } w_j = \frac{d_j}{d_i + d_j}, \quad (4)$$

where d_i and d_j represent the distance between the point p_T and the according centroid of the point sets from segments i and j respectively.

4. EXPERIMENTS

This section gives insight into the data used in the different steps of our approach and presents our conducted experiments and experimental results in detail facilitating replication.

4.1 Data settings

3D Point Cloud. The 3D point cloud has been recorded using a mobile mapping system (Figure 5), which consists of a high precision laser scanner (Leica HDS6100) a navigation grade inertial measurement unit (IMAR inav-FJI) and an odometer (Corrsys Datron CorreVit L-400).

The data set has been captured during a 5 minute drive through the Campus Poppelsdorf of the University of Bonn / Germany. The trajectory can be seen in Figure 1. It is important to note, that the system provides raw sensor measurements (IMU and odometer readings, GNSS observations and range & bearing measurements from the scanner) in contrast to the final georeferenced point cloud. The latter is created by custom algorithms, fusing



Figure 5. The used Mobile Mapping System with laser scanner, odometer, GNSS receiver and inertial measurement unit.

GNSS, IMU and odometer data to a position and orientation information, which is then used to transform each laser measurement into the global coordinate system, leading to the point cloud shown in Figure 6 (top). Only due to this procedure we are able to feed the transformation gathered by the proposed method back to the trajectory and then to improve the quality of the final point cloud.

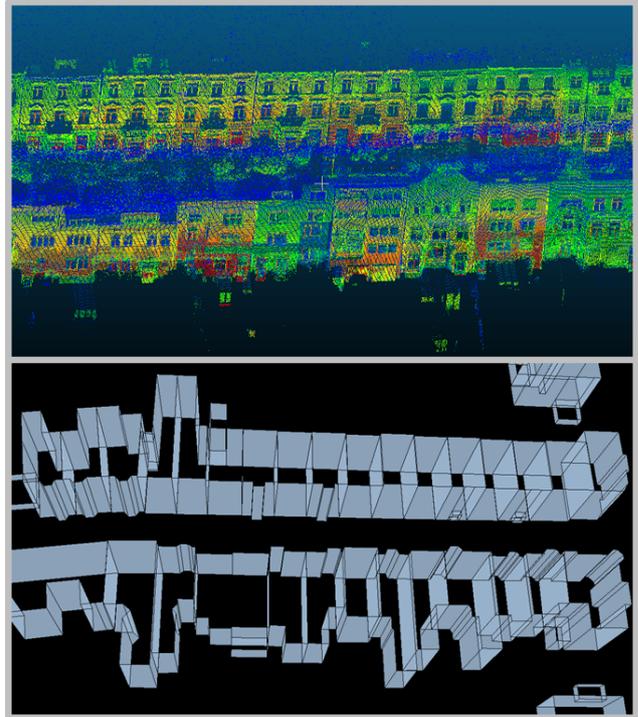


Figure 6. Input data: kinematic 3D point cloud (top) and the corresponding 3D city model acquired from a WFS-Request (bottom).

The absolute accuracy of the point cloud is determined by the absolute accuracy of the trajectory estimation, which is limited by the accuracy of the GNSS observations. In this data set, a code based navigation solution has been used, leading to an accuracy of several meters. This uncertainty can be easily seen later when comparing the point cloud generated from driving past the same buildings from both directions. It also provides a perfect example, where the proposed method can improve the results.

3D City Model. A further important input for our approach are building models in the surrounding of our driving area. In this context, we used building models of the environment in the level of detail 1 (LoD1) according to the standardized CityGML format (Gröger et al., 2012). This level of detail includes models derived from footprints with extruded heights. Figure 6 (bottom) visualizes an excerpt of the city model of our test area. The buildings are acquired performing a Web Feature Service (WFS) request specifying a bounding box of the region of interest. The WFS is hosted on a GeoServer² providing CityGML models from open authoritative data sources from OpenGeodata.NRW³ of the state of North Rhine-Westphalia in Germany⁴. The two dimensional position accuracy amounts few centimeters which enables to improve the accuracy of the according point clouds and in turn the

²<http://geoserver.org/>

³<https://www.opengeodata.nrw.de/produkte/>

⁴<https://www.bkg.bund.de/EN/Home/home.html>

Class	no-Building	Building	precision (%)
pred. no-Building	1276804	7220	99.44
pred. Building	7732	222403	96.64
recall (%)	99.40	96.86	

Table 2. Results of the point-based classification of building and no-building points.

accuracy of the GNSS trajectory. The height, however, is characterized by a lower accuracy (around 5m).

Coordinate systems. Both the 3D point cloud and the 3D city model are represented in an Universal Transverse Mercator (UTM) coordinate system. The heights of the 3D points are however ellipsoidal, as GNSS systems provide these per default, whereas those of the city model are represented as physical heights with respect to the German main height network. In this context, an adjustment step is needed in order to map both heights. To this end, the geoid undulation has to be determined. We make use of a precise network of measured heights provided by the German federal agency for cartography.

4.2 Experimental results

Point-based classification. For the supervised learning, we labeled points from the 3D point cloud as “building” and “no-building” points manually for the training phase. For the training dataset, a support vector machine has been learned based on the labels and the corresponding derived features (cf. Table 1). As kernel function we used the RBF kernel (Schölkopf and Smola, 2002) in order to perform a non-linear classification with the parameters $\gamma = 0.25$ and the penalty parameter $c = 304.43$. In order to determine the optimal values of these two a-priori not known parameters, we performed a grid parameter optimization (Hsu et al., 2003). For building the SVM-Model (training), we choose the parameter set with the lowest error rate performing in a 10-fold cross-validation. An overall accuracy of 99.01% has been achieved.

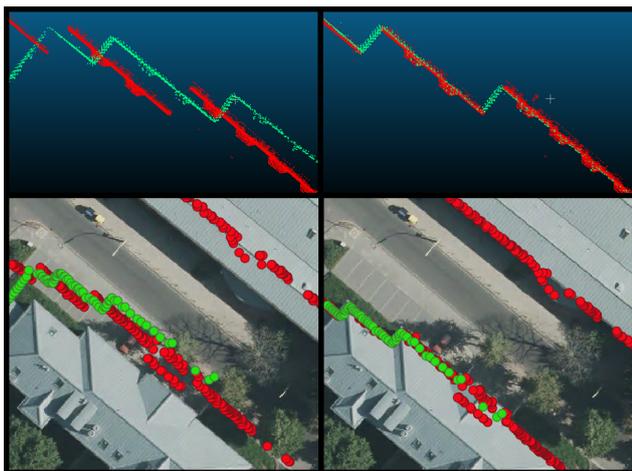


Figure 7. A code-based solution with an accuracy of several meters before the application of our method (left). An offset is noticed driving past the same building during the outward (green) and return (red) journey. The incorporation of 3D city models in our approach improves the point cloud accuracy (right).

For the evaluation of our learned model, we tested on an independent dataset achieving an overall accuracy of 98.01%. The

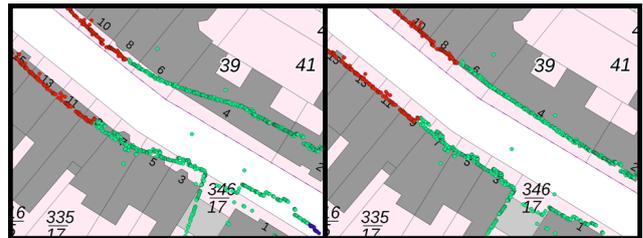


Figure 8. Excerpt of the point cloud with two segments (red) and (green). Small segments are influencing the result. Results without a pre-transformation of the point cloud (left) and with a pre-transformation (right) are depicted.

detailed confusion matrix is depicted in Table 2. For our experiments, the libSVM toolbox has been used (Chang and Lin, 2011). Our points of interest, i.e. the class building, are predicted with a recall of about 97%. Hence, the error rate of 3% does not turn out to be critical to sample appropriate facade points. Further, the use of trimmed ICP reduces as mentioned the impact of misclassified points in this context.

Segmentation and registration of the point cloud. The result of our developed approach is exemplary illustrated in Figure 7 where an extract of the point cloud is shown before (left) and after (right) the application of our method. The pre-classified point cloud has been partitioned in 12 different segments of 100 meters as depicted in Figure 10. The segments have been subsequently registered using ICP and an interpolation for a seamless transition has been applied according to the steps described in section 3.3. The green points in Figure 7 stem from the outward journey whereas the red points belong to scans from the return journey. A deviation between the two original point clouds on the left can be stated. On the right side, this deviation cannot be observed after the application of our approach.

The same point cloud is illustrated in the bottom of the figure projected on an orthophoto of the driving area. Beside the consistency, an improvement of the absolute position is achieved. The deviation to the building on the left has been corrected enabling points locations which even reflect the building positions.

The visual analysis of the results shows that the procedure improves the point cloud significantly. However, since the data set does not have any fixed reference points, it is difficult to quantify the results. In order to obtain a rough estimate of the magnitude of the changes, we analyze the distances of the points to the building models before and after the registration step which can be shown in Table 3. We consider the median of the distances instead of the mean which is highly affected by outliers (e.g. misclassified



Figure 9. Impact of our approach on the recorded trajectory (left). The corrected trajectory (right) using our method. The underlying point cloud (blue).

segment id	1	2	3	4	5	6	7	8	9	10	11	12
origin points	5.3	5.1	5.2	5.3	5.2	2.9	4.0	6.1	5.7	3.7	5.3	2.4
registered points	0.4	0.2	0.3	0.3	0.2	0.3	0.3	0.3	0.1	0.4	0.2	0.3

Table 3. Overview of the median distance [m] between the point cloud and the according building model per segment, after and before the registration.

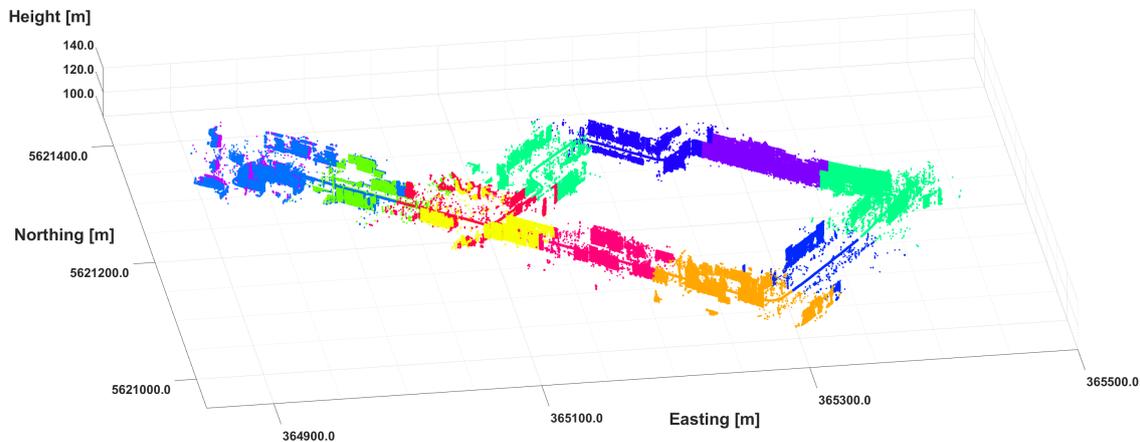


Figure 10. The segmentation of the point cloud in 12 segments (different colors). Overlapped colored areas represent segments from outward and return journeys.

points). Using the original point cloud, the median of the differences amounts from 2.4 to 6.1 meters, whereas its value after the registration is in a range of few decimeters (0.1 to 0.4). Hence, an improvement of the point cloud position between 2 and 6 meters has been achieved.

The improvement of the point cloud position is reflected in the trajectory as already described in Figure 1. There is no drift more to notice at the start and the end point of the trajectory. Besides, the offset to the road has been corrected as well.

Interpolation after the piecewise segmentation. The noticed overlapping between two consecutive segments in the partitioned point cloud from Figure 4 is attributed to the fact that these segments belong to areas from the point cloud dominated by facades which are parallel to the driving direction. In this context, the ICP does not yield a unique solution due to the sparsely structured facade planes and the resulting transformation has to be considered as an outlier. In these cases, the interpolation between two segments may lead to a distortion of the cloud, as visible on the left side of Figure 8. Here, some points which belong to the green segment have been assigned to the rear of the building rather than to the front facade. Together with the subsequent interpolation, the point clouds become distorted. As mentioned in section 3.2, we avoid these effects by providing the ICP algorithm with an improved start solution, given by the resulting transformation of the previous segment. The right part of the figure shows the result of this method.

We tested our method with different segment sizes between 50 m and 200 m. As mentioned, too small segments do not provide enough spatial structure to obtain a unique solution of the ICP, whereas too big segments increase inaccuracies as well as the influence of sensor drifts. For point clouds segments around 100 m, we achieved very good and stable results leading to a significant improvement of the point cloud accuracy and the trajectory as well. Figure 9 shows the improvement achieved using our approach leading to an update of the trajectory estimate. Nev-

ertheless, the segment length as measure is not the optimal measure. The integration of further parameters such as the number of curves or the spatial distribution of the recorded points will be addressed in future work.

5. CONCLUSION AND OUTLOOK

The paper presents a novel approach for improving trajectories using 3D city models and kinematic point clouds for highly accurate positioning. The method has been demonstrated on a data set captured by a mobile mapping system. The latter relies on the combination between relative and absolute positioning sensors to generate their trajectories from a captured 3D point cloud. A combination of 3D building models and mobile mapping systems enabled to bridge weak GNSS signals and drifts characterizing drives within shadowed cities. We consecutively registered segments of the kinematic point cloud using the building model of the surrounding area using a point-to-plane trimmed iterative closest point (ICP). An informed sampling of candidate matching points is possible performing a classification task which pre-filters facade points using support vector machines (SVM). The resulting transformation has been applied to update the trajectory using an interpolation between segments and the new trajectory have been used to update the point cloud. The accuracy of the classification of facade points showed an accuracy of 99%. The general functionality of the method and the impact on the point cloud and the trajectory have been demonstrated using a real data set taken with a mobile mapping system. The results show that 3D building models turn out to be suitable to improve the positioning of vehicles in shadowed cities.

By now, we showed, that it is possible to derive correcting transformations to the trajectory by matching point cloud segments to a city model and that these transformations are able to improve the trajectory. However, the next step would be to formally integrate these transformations into the actual trajectory filtering process in the form of a measurement. Then the method could

be applied in real-time to obtain better positioning information. In this context, it has to be investigated, which segment sizes are suitable and how often the correction has to be applied. It would also be possible to integrate this transformation in the form of constraints to the trajectory estimation process. A rigorous quantitative evaluation of the accuracy improvement using a high accuracy reference trajectory and a larger data set are also subjects of current research.

References

- Besl, P. J. and McKay, N. D., 1992. Method for registration of 3-d shapes. In: *Sensor Fusion IV: Control Paradigms and Data Structures*, Vol. 1611, International Society for Optics and Photonics, pp. 586–607.
- Chang, C.-C. and Lin, C.-J., 2011. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* 2(3), pp. 27:1–27:27.
- Chehata, N., Guo, L. and Mallet, C., 2009. Airborne lidar feature selection for urban classification using random forests. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 38 (3/W8), pp. 207–212.
- Chen, Y. and Medioni, G., 1992. Object modelling by registration of multiple range images. *Image and vision computing* 10(3), pp. 145–155.
- Chetverikov, D., Svirko, D., Stepanov, D. and Krsek, P., 2002. The trimmed iterative closest point algorithm. In: *Object recognition supported by user interaction for service robots*, Vol. 3, IEEE, pp. 545–548.
- Clancy, S., 2011. The importance of applied control. *LiDAR Magazine*.
- Eckels, R. and Nolan, J., 2013. Mobile laser scanning: Field methodology for achieving the highest accuracy at traffic speed. In: *Proceedings of the 18th Association of Public Authority Surveyors Conference (APAS2013)*.
- Förstner, W. and Wrobel, B., 2016. *Photogrammetric Computer Vision: Geometry, Orientation and Reconstruction*. Geometry and Computing, Springer International Publishing.
- Glennie, C., 2007. Rigorous 3D error analysis of kinematic scanning LIDAR systems. *Journal of Applied Geodesy* 1(3), pp. 147–157.
- Gräfe, G., 2007. Quality management in kinematic laser scanning applications. *Proceedings of the 5th International Symposium on Mobile Mapping Technology (MMT2007)*.
- Gröger, G., Kolbe, T. H., Nagel, C. and Häfele, K. H. (eds), 2012. OGC City Geography Markup Language (CityGML) Encoding Standard, Version 2.0.0., Open Geospatial Consortium: OGC Doc. No. 12-019.
- Hauert, J.-H. and Budig, B., 2012. An algorithm for map matching given incomplete road data. In: *Proc. 20th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS '12)*, pp. 510–513.
- Hsu, C.-W., Chang, C.-C. and Lin, C.-J., 2003. A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University.
- Low, K.-L., 2004. Linear least-squares optimization for point-to-plane icp surface registration. *Chapel Hill, University of North Carolina*.
- Niemeyer, J., Rottensteiner, F. and Sörgel, U., 2012. Conditional random fields for lidar point cloud classification in complex urban areas. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* I-3, pp. 263–268.
- Nüchter, A., Lingemann, K., Hertzberg, J. and Surmann, H., 2007. 6d slam - 3D mapping outdoor environments. *Journal of Field Robotics (JFR)* 24(8-9), pp. 699 – 722.
- Osogami, T. and Raymond, R., 2013. Map matching with inverse reinforcement learning. In: *Twenty-Third International Joint Conference on Artificial Intelligence*.
- Pauly, M., Gross, M. and Kobbelt, L. P., 2002. Efficient simplification of point-sampled surfaces. In: *Proceedings of the conference on Visualization'02*, IEEE Computer Society, pp. 163–170.
- Puente, I., González-Jorge, H., Martínez-Sánchez, J. and Arias, P., 2013. Review of mobile mapping and surveying technologies. *Measurement* 46(7), pp. 2127–2145.
- Rouhani, M., Lafarge, F. and Alliez, P., 2017. Semantic segmentation of 3d textured meshes for urban scene analysis. *ISPRS journal of photogrammetry and remote sensing* 123, pp. 124–139.
- Rusu, R., Blodow, N. and Beetz, M., 2009. Fast point feature histograms (fpfh) for 3d registration. In: *IEEE International Conference on Robotics and Automation, 2009. ICRA '09.*, pp. 3212–3217.
- Schölkopf, B. and Smola, A. J., 2002. *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT Press, Cambridge, USA.
- Stachniss, C., 2009. *Robotic Mapping and Exploration*. Springer Tracts in Advanced Robotics, Springer-Verlag GmbH, Berlin.
- Thrun, S., Burgard, W. and Fox, D., 2005. *Probabilistic Robotics*. The MIT Press, Cambridge, USA.
- Vapnik, V. N., 1998. *Statistical learning theory*. A Wiley-Interscience publication, Wiley, New York, USA.
- Viel, A., Gubiani, D., Gallo, P., Montanari, A., Dalla Torre, A., Pittino, F. and Marshall, C., 2018. Map matching with sparse cellular fingerprint observations. In: *2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)*, IEEE, pp. 1–10.
- Wahl, E., Hillenbrand, U. and Hirzinger, G., 2003. Surflet-pair-relation histograms: a statistical 3d-shape representation for rapid classification. In: *Proceedings of Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003*, pp. 474–481.
- Zhou, Q.-Y. and Neumann, U., 2008. Fast and extensible building modeling from airborne lidar data. In: *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '08*, ACM, New York, NY, USA, pp. 7:1–7:8.