The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLVIII-2/W8-2024
8th International ISPRS Workshop LowCost 3D - Sensors, Algorithms, Applications, 12–13 December 2024, Brescia, Italy

Visual localization in urban environments employing 3D city models

Yasmin Loeper[1], Markus Gerke[1], Ahmed Alamouri[1], Alexander Kern[2], Mohammad Shafi Bajauri [1], Phillipp Fanta-Jende[3]

[1] Institute of Geodesy and Photogrammetry, Technische Universität Braunschweig,
Brunswick, Germany - (y.loeper, m.gerke, m.bajauri, a.alamouri)@tu-braunschweig.de
[2] Institute of Flight Guidance, Technische Universität Braunschweig, Brunswick, Germany - a.kern@tu-braunschweig.de
[3] Unit Assistive and Autonomous Systems, Center for Vision, Automation and Control,
AIT Austrian Institute of Technology, Vienna, Austria - phillipp.fanta-jende@ait.ac.at

Keywords: Visual Localization, 3D City Models, CityGML, Indirect Pose Estimation, Line Features, Bayesian Optimization

Abstract

Reliable pose information is essential for many applications, such as for navigation or surveying tasks. Though GNSS is a well-established technique to retrieve that information, it often fails in urban environments due to signal occlusion or multi-path effects. In addition, GNSS might be subject to jamming or spoofing, which requires an alternative, complementary positioning method. We introduce a visual localization method which employs building models according to the CityGML standard. In contrast to the most commonly used sources for scene representation in visual localization, such as structure-from-motion (SfM) points clouds, CityGML models are already freely available for many cites worldwide, do not require a large amount of memory and the scene representation database does not have to be generated from images. Yet, 3D models are rarely used because they usually lack properties such as texture or only contain general geometric structures. Our approach utilizes the boundary representation (BREP) of the CityGML models in Level of Detail (LOD) 2 and the geometry of the query image scene from extracted straight line segments. We investigate how we can use an energy function to determine the quality of the correspondence between the line segments of the query image and the projected line segments of the CityGML model based on a specific camera pose. This is then optimized to estimate the camera pose of the query image. We show that a rough estimation of the camera pose is possible purely via the distribution of the line segments and without prior calculation of features and their descriptors. Furthermore, many possibilities and approaches for improvements remain open. However, if these approaches are taken into account, we expect CityGML models to be a promising option for scene representation in visual localization.

1. Introduction

Providing an absolute pose in a given coordinate frame is vital for applications such as mobile robotics, navigation via mobile phones, the navigation of drones or autonomous vehicles or augmented reality (Castle et al., 2008, Middelberg et al., 2014, Arth et al., 2009, Heng et al., 2018, Lim et al., 2012, Naseer et al., 2018, Couturier and Akhloufi, 2021, Couturier and Akhloufi, 2024). Absolute localization is usually done using GNSS positioning. GNSS provides the spatial position, heading or rotation information is derived from IMU sensors or compasses, even in low-cost smartphones. However, in urban environments, GNSS is very often affected by occlusion and multi-path effects, leading to reduced reliability and accuracy. Moreover, GNSS is prone to spoofing and jamming, which renders GNSS alone insufficiently reliable, especially in safety-critical setups such as autonomous vehicle navigation. One possibility to overcome those limitations is to use visual localization, in which the pose is estimated on the basis of image observations.

The aim of visual localization is to determine the six degrees of freedom (DoF) of the camera pose from which a specific query image was taken. This is done by comparing the similarity between the query image and a scene from a database. The methods of visual localization can be categorized into explicit and implicit approaches based on their scene representation. Explicit methods use 3D models, meshes, SfM point clouds, or georeferenced images (Sarlin et al., 2021, Sattler et al., 2012a, Svarm et al., 2017, Zhou et al., 2020), often referred to as map information. They can be further divided into direct and indirect approaches. In direct methods, the camera pose is estimated by matching 2D image features with 3D points (Schönberger et al., 2018, Snavely et al., 2008, Irschara et al., 2009), typically using a minimal solver in a RANSAC framework (Barath et al., 2019a, Barath et al., 2019b). However, these methods face scalability issues. Indirect methods first conduct an image retrieval step (Arandjelović et al., 2015, Gordo et al., 2016) to select suitable images, then estimate the camera pose by matching 2D features with visible 3D points in the retrieved images (Humenberger et al., 2020, Sarlin et al., 2018, Sattler et al., 2012b). These methods can utilize dense representations like meshes or laser point clouds (Brejcha et al., 2020, Panek et al., 2022, Sibbing et al., 2013, Zhang et al., 2020), providing precise pose estimates but challenging in terms of reconstruction and privacy.

Implicit methods leverage neural networks and can be classified into scene regressors, absolute pose regressors and relative pose regressors (Brachmann and Rother, 2017, Cavallari et al., 2019, Cavallari et al., 2020, Kendall et al., 2015, Laskar et al., 2017). While these methods yield accurate estimations, they are limited by scene specificity but can adapt in real-time to new scenes.

Less common are the use of CAD/building information model (BIM) and city models for the scene representation. Rendered synthetic images from the 3D models can be used to train networks to regress the camera poses (Acharya et al., 2019, Acharya et al., 2022, Acharya et al., 2023) or the 3D models can be used explicitly for 2D feature matching (Panek et al., 2022, Sibbing et al., 2013, Panek et al., 2023). Aligning query images with 3D models can improve accuracy, though these methods

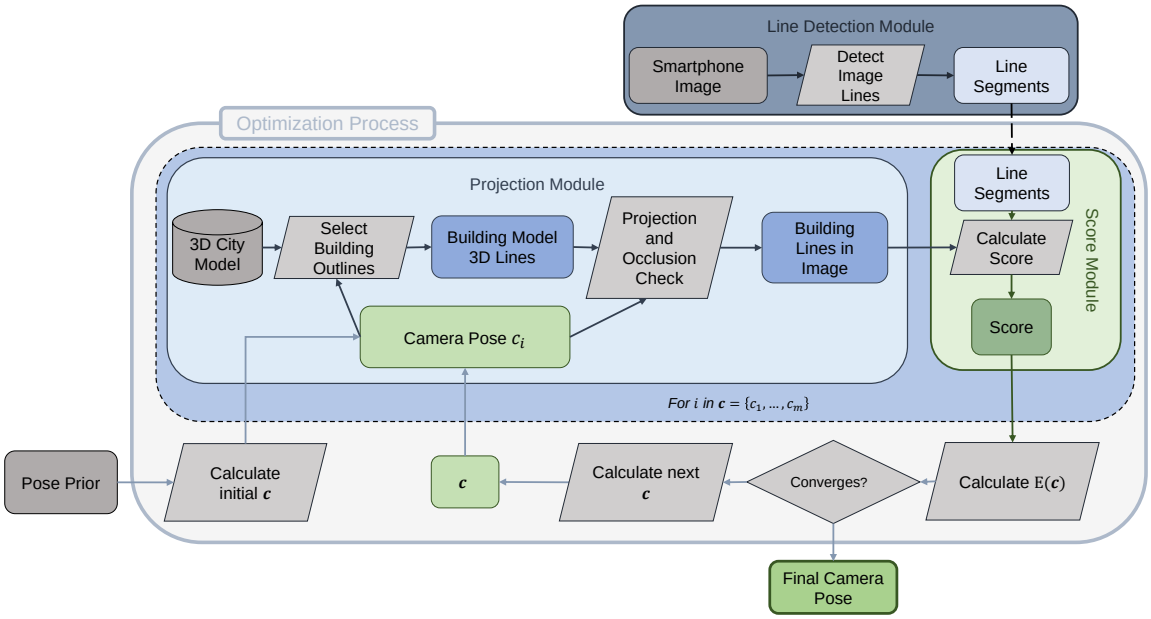

Figure 1. Overall workflow of the object-based visual localization approach, where **c** represents the configuration of possible camera poses.

require substantial training time and are sensitive to texture and lighting conditions. The availability of suitable models also remains a challenge.

We propose a method that employs CityGML models and uses a sampling approach in the solution space, i.e., the coordinate frame, with an score function which is interpreted as an energy in an optimization framework to estimate the camera pose of the query image. If such models are used, no distinct point features usually exist, but lines or surfaces can be employed. The use of line features has advantages over the use of point features, especially in urban environments. Lines, in particular, bounding edges, are mostly detectable in images, even in poor lighting conditions, because of the high contrast to the background. They provide a geometric indication of the structure of a scene (Pautrat et al., 2021). In addition, boundary representation (BREP), i.e. point-line-topology, is a simple and accepted means to represent the core geometric elements of buildings and contains more structural information than points alone.
However, one disadvantage of using those representations is that lines need to be well distributed (in terms of coverage, but also angular directions) to guarantee that the pose of a single image can be determined. The accuracy of pose estimation depends on those parameters and the accuracy of model representation and image resolution.
The CityGML standard[1] establishes a conceptual model and exchange format designed for the portrayal, storage, and interchange of virtual 3D city models. The CityGML conceptual model supports modelling of a variety of city objects such as buildings, bridges, tunnels, water bodies, transportation, vegetation, land use, etc. The Open Geospatial Consortium, OGC, standardizes CityGML. Buildings are represented in different Levels of Detail (LOD). For visual navigation, the LOD2 is important: here the general roof structure is retained, but details such as dormers are not added. The outer facade geometry and main roof lines are interesting since those will be used to support localization from terrestrial images (facades) and airborne images (ridges, outer roof edges). Compared to other 3D reference data, 3D city models following the CityGML scheme are already available free of charge for many countries.

In our approach, we use the similarity between a query image and a projected scene of the CityGML model based on a assumed camera pose. To evaluate the similarity, we extract straight-line features of the query image and the projection of building models into the proposed camera pose. We assume that if the lines of the query image match well with the lines of the building model, the underlying pose used for this projection is correct. Based on their match, we calculate the energy. The best pose is found when the resulting optimization problem of the energy converges.

2. Related work

In principle, 3D models can be used either implicitly or explicitly to represent the scene. When using the 3D models explicitly, the indirect approach with an upstream image retrieval is usually used (Panek et al., 2022, Panek et al., 2023). If the 3D models are used in the form of the implicit method, synthetic images are rendered using the models, which are then used to train the neural network (Acharya et al., 2022, Acharya et al., 2023).
Less common are approaches that explicitly and directly use the BREP of the 3D model to search for the corresponding image lines in the image. Co-registration between vector-based city representations and airborne images is presented in (Sun et al., 2019), while (Fanta-Jende et al., 2019, Chen et al., 2021) co-register airborne images and mobile or terrestrial mapping data. To co-register generalized vector data and images is presented in (Jarzabek-Rychard and Maas, 2017). Once the matches between datasets are established, a joint adjustment of all data can be performed (Gerke, 2011, Sun et al., 2019).
Another approach is indirect and employs sampling in the solution space, in our case the unkown 3D pose of the camera. Each pose realized in this way is evaluated w.r.t. its match with the

[1] https://www.ogc.org/standard/citygml/

model, leading to a score. The overall objective is to minimize the score and thus optimize the parameters.

For our approach, we decided to optimize the energy function or the parameter estimation of the six DoF in the form of Bayesian optimization. Bayesian optimization has the advantage that it is particularly effective for continuous domains with less than 20 dimension (Frazier, 2018). With Bayesian optimization, the function to be optimized can be treated like a black box, as the analytical form does not have to be known and yet the behavior of the function can be estimated. This is because a surrogate model or an acquisition function of the target function is set up using the regression of the Gaussian process. Acquisition functions can quantify the uncertainties in the predictions so that they can help in deciding the next sampling. There are different types of acquisition functions such as expected improvement, knowledge gradient, entropy search or predictive entropy search (Frazier, 2018).
As far as straight line segment extraction is concerned, different types of line detectors and extractors can be used. The types can be broadly grouped into three different approaches based on their detection mechanism: handcrafted- and Hough-based (Akinlar and Topal, 2011, Grompone von Gioi et al., 2012, Suárez et al., 2022, Fernandes and Oliveira, 2008), learning-based (Huang et al., 2018, Zhou et al., 2019, Dai et al., 2021, Pautrat et al., 2021) and hybrid-based (Pautrat et al., 2022, Teplyakov et al., 2022) methods. Handcrafted methods yield accurate results but lack repeatability and adaptability (Pautrat et al., 2021, Pautrat et al., 2022). Deep learning techniques, introduced for edge detection and wireframe parsing, have emerged from the wireframe dataset but face bias issues (Dai et al., 2021, Pautrat et al., 2021, Pautrat et al., 2022). Hybrid methods try to combine traditional techniques with deep networks to leverage the strength of both handcrafted and learning-based approaches (Pautrat et al., 2022).

3. Methodology

Since the assumed pose prior of the sensor, obtained by the possible degraded GNSS observations might be wrong by several meters and degrees, respectively, a direct solution where objects are searched in images, is very error-prone and unreliable. Therefore, we implement a Bayesian optimization where we sample in the space of possible camera poses and assess, how well the straight-line features of the reprojection of the 3D building model fit to the straight-line features of the query image. We assume that the scene from the query image is sufficiently well represented by the building lines visible in the image. Consequently, if the building lines of the 3D model are projected into the image using the correct camera pose, they overlap with the building lines from the query image.

3.1 Object-based Visual Localization Workflow

The workflow shown in the figure 1 was developed to simulate the problem presented. The optimization process is the main module to simulate minimizing the energy function. In this process, the camera poses are initialized, and their energy is calculated and optimized using a solver. To calculate the energies of the individual camera poses the building lines must first be projected into the image plane dependent on the camera pose and the lines in the query image. This is done in the Projection and Line Detection Module. The energy of the individual camera poses is finally calculated in the Score Module. The individual modules of the object-based visual localization approach are described in more detail below.

3.1.1 Projection Module The initialized camera poses are input to the Projection Module. The steps of the Projection Module are shown in Fig. 1. Depending on the pose with it's six DoF a bounding box with a specific radius is created. Within this bounding box all surrounding buildings of the 3D city model are requested. The corner points of each building surface are returned. Using the transformation matrix, consisting of the six DoF, this vector representation is transformed into the camera coordinate system. Clipping also takes places during the transformation process. This means that corner points that lie outside the field of view (in image space) are corrected to the intersection point of the visible area. The result is the vector representation of the building edges.

An example of the result can be seen in Fig. 2(a). This is followed by the occlusion check in form of the calculation of a z-buffer. The vector representation of the building edges is rasterized for each building surface using the depth values of the corner points resulting from the transformation. The z-buffer is the result of the minimum calculation of the depth values. An exemplary z-buffer image is shown in Fig. 2(b). The edges are extracted from the z-buffer using the Canny edge extractor and the line segments, i.e. their start and end points, are calculated from this using the Hough transformation. These can be seen in Fig. 2(c). The resulting line segments are used as input for the score module.

3.1.2 Line Detection Module To calculate the second input in the score module, the lines of the query image are detected and extracted using the SOLD2 (Pautrat et al., 2021) line extractor. We use the pre-trained model of SOLD2, which is suitable for man-made environments.

3.1.3 Score Module Finally, the score reflects the alignment of the lines detected in the image to the projected building edges. The angles and the orthogonal distance of the two line types are included in the score. We deliberately refrained from comparing coverage and line length due to sources of error such as the occlusion of building edges in the query image by obstacles such as cars, people or vegetation.
In the score module, the angles of the line segments from the the building lines are calculated first. Each line segment of the query image is checked for each building line, i.e. the extent to which the angles of the query image lines deviate from the building line angle is checked. If these are within a certain threshold, they are regarded as potentially corresponding line segments. A visual representation of the potentially corresponding line segments for the query image and the building lines is shown in Fig. 3(c). The orthogonal distance is then calculated for the potential corresponding line segments. If the orthogonal distance is within a threshold, a distance score is calculated. The distance score becomes one if the orthogonal distance is zero. Based on the number of corresponding line segments of the query image an average distance score for each building line is calculated. The sum of these ultimately results in the final score. It should be noted that due to the minimization problem of the energy presented, the score must finally be inverted so that the best energy is zero and the worst is one.

3.1.4 Optimization For the optimization of the energy, we decided to consider the problem as a black box function. We have chosen to implement a method based on the search algorithm Tree-Structured Parzen Estimators (TPE). This search algorithm models the distribution of the parameters, i.e. the six DoF, within a defined search space in order to estimate the performance of different combinations.

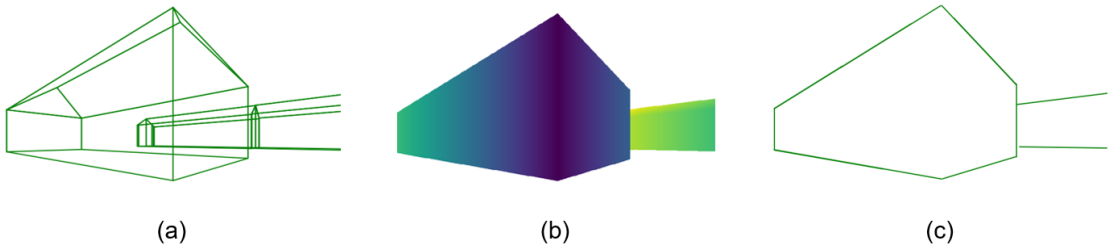

Figure 2. Visualization of the steps of the projection module: The projected building lines of the 3D model, taking into account the camera pose, can be seen in part (a). The result of the occlusion check in the form of the z-buffer is shown in (b). (c) shows the lines extracted from the z-buffer by handcrafted line extractors.

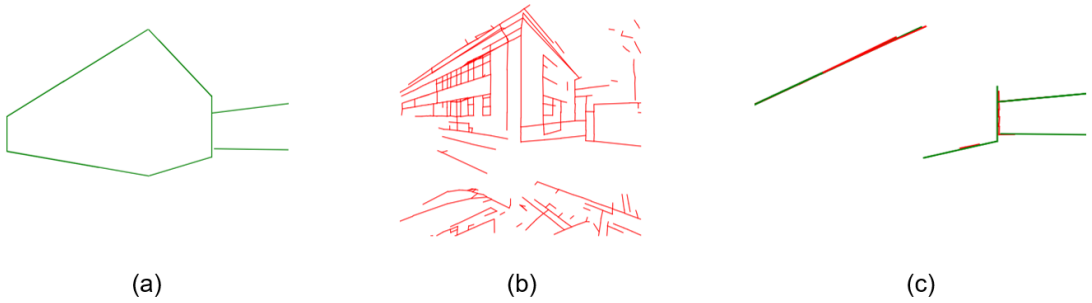

Figure 3. Visualization of the steps of the score module: (a) shows the line segments from the 3D models as part of the input to the Score Module. (b) shows the line segments from the query image as the second part of the input to the Score Module. The matched line segments are shown in (c).

4. Experiments

We are testing our approach using a selected test area in Braunschweig, a city in Lower Saxony in Germany. The test area is located near the city centre of Braunschweig and covers approx. $370 \times 280\mathrm{m}^2$. The area contains a variety of buildings and building types as well as sections with urban canyons.

4.1 Reference dataset

The Braunschweig 3D city model in LoD2 is available as 3D reference data for the area (Landesamt für Geoinformation und Landesvermessung Niedersachsen, 2024) and is stored in CityGML format on a server. The 3D city model was created using building outlines from cadastral data, digital terrain model (DTM) in $5\,\mathrm{m}$ resolution and 3D data from laser scan or matching point cloud. This means that the positional accuracy depends on the cadastral data and the height accuracy on the matching point cloud (Landesamt für Geoinformation und Landesvermessung Niedersachsen, 2024).

4.2 Query dataset

To create a query image dataset we captured images from a handheld smartphone rigidly attached to a tactical grade inertial navigation system (INS), c.f. Fig. 4. The images got resampled to a resolution 960×1280, resulting in a average ground sampling distance (GSD) of $4\,\mathrm{cm}$. By using post-processed kinematics (PPK) we estimate the trajectory of the smartphone down to a few centimetres standard deviation in good conditions. Since the test area is partly in narrow urban areas, we added a post-processing step: The dataset underwent

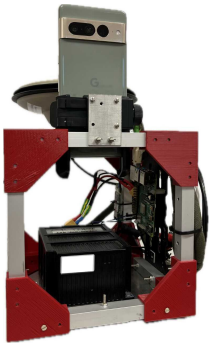

Figure 4. Smartphone rigidly mounted to INS system to capture query data.

a structure-from-motion and bundle adjustment pipeline, where also Ground Control Points (GCP) and Check Points (CP) were involved. The points were measured in the field using a surveying grade GNSS-receiver with RTK (GCP) at distinct, well-observable positions in terms of GNSS outages. The adjustment resulted in an RMSE at GCP of $(6; 7; 2)\ \mathrm{cm}$ in X, Y, Z, respectively and mean errors at CPs of $(7; 4; 1)\ \mathrm{cm}$. The exterior orientation (EO) parameters of images were obtained with a mean error of $1.5\,\mathrm{cm}$ in X, Y and $0.7\,\mathrm{cm}$ in Z and standard deviations of 6 and $2\,\mathrm{mm}$, respectively. The rotation components have a mean error of $0.02°$ at $\sigma = 0.04°$.

4.3 Setup of experiments

It is obvious that considering a small number of line segments on the side of the building lines can lead to an ambiguity of

the score and finally effects the best pose. Therefore we have to make assumptions for our experiments. We assume that we have a pose prior and that the true pose is within a certain radius with respect to the pose prior. Therefore, we limit the search space in the X and Y directions. We also assume that the edge device is approximately levelled during the query image acquisition. This means that we can set the rotation angles omega and phi to 90° and 0° respectively. We can also assume that we have an initial assessment for κ. We can then also limit the search area for κ. To rule out the possibility of proposed poses being inside buildings, we created a grid from the digital terrain model (DTM) and the building outline layer for the test area. The grid is set to None-values in places where a building is present and corresponds to the values of the DTM in all other places. This also allows us to specify that the Z coordinate corresponds to the value of the grid including an offset at point X, Y. We want to investigate two questions with our experiments. Firstly, the question of whether the score can represent the true pose well enough. Secondly, we want to investigate the accuracy of our approach. Our experiments are designed accordingly. To test the sensitivity of the score, we vary the size of the search space in the X and Y directions as well as the angle search space. We test the combinations of $40 \times 40; 20 \times 20\mathrm{m}$ search space for X, Y, an angle search space of $\pm 10^\circ$ $\pm 20^\circ$ and $50, 100, 200$ iterations during the optimization. In the following, however, we will only consider the combinations with a search space for X, Y of $20 \times 20\mathrm{m}$ and an angle search space of $\pm 20^\circ$ around the true value for κ.

5. Results

In the following, we will present and then analyze the results in relation to the questions from section 4.3. First, we check the results of the analysis to see whether the score can reliably represent the true pose and then we look at the accuracy of our approach.

5.1 Measuring reliability of the score

To analyze the score, we tested our approach for 10 query images with the scenarios mentioned above. The results of the translation errors in metres in the X and Y directions and their score are shown for 50, 100 and 200 iterations in Fig.5. Depending on the query image, we have summarized the results of the scores of individual iteration steps in classes. A low score should express a very high correspondence of the line segments from the z-buffer with those of the query image and vice versa. Clusters of low scores around the minimal deviation are therefore to be expected. These clusters cannot clearly be recognized at 50, 100 or even 200 iterations. A poor score of one occurs frequently in the vicinity of a minimal translation error. The comparison of the different number of iterations shows that, regardless of this, a low score does not necessarily cluster around a translation error of zero. The distribution of the score as a function of the rotation error for κ for the different iterations is similar to the translation error, it is noticeable with the rotation error that a score of one does not only occur with strong deviations of κ from the ground truth value. There is also no direct influence of the number of iterations on the representation of a low score of a deviation around zero for κ.

5.2 Measuring absolute errors

To investigate the accuracy of our approach, we performed the optimization of the camera pose parameters for 30 images. The results of the optimization are listed in table 1. The average of the translation error of the tested images is $8.9\,\mathrm{m}$ and that of the rotation error is $-2\,23^\circ$. For 3.85 % of the images tested, the translation error is less than or equal to $2\,\mathrm{m}$ and the rotation error is less than or equal to 2°. While for 23.08 % of the images there is a translation error of $5\,\mathrm{m}$ and a rotation error of 5°. For 34.62 % of the images, there is a translation error of maximum $9\,\mathrm{m}$ and a rotation error up to 10°. The mean absolute deviation of the translation error is $4.88\,\mathrm{m}$ and that of the rotation error is 14.02°.

Table 1. Localization results for the 30 query images. We report the % of query images localized within the given deviation of the ground truth pose.

$2\,\mathrm{m}, 2^\circ$	$5\,\mathrm{m}, 5^\circ$	$9\,\mathrm{m}, 10^\circ$	MAD
3.85	23.08	34.62	4.88 / 14.02

6. Discussion

By analyzing the deviations for translation and rotation as a function of the score, it is observable that the score does not reflect the true pose reliably enough, even with the assumptions made. By analyzing various test scenarios, it can be ruled out that the reason for this is the iteration or the search spaces. It can be seen that even with values of the translation and rotation errors around zero, the score is one. The results of the optimization with 30 test images show that the accuracy is not yet sufficient. This is the consequence of the observation of the unreliability of the score.
The final proposed camera pose is based on the reliability of the score. The score in turn is based on the fact that it is possible that the line configuration of the building edges, resulting from the pose, can match the line configuration of the query image very well. However, there may be cases in which these correlations are not given. Examples of this are strongly occluded building edges in the query image or too few true building lines. This ultimately leads to the score not reflecting the true pose.
In addition, we must take into account that there are also cases in which the number of lines may be too limited.

7. Conclusion

We present a low-cost visual localization approach using 3D city models. The use of 3D city models has the advantage that they are now available free of charge in many cities worldwide. In addition, the 3D city models, which are divided into tiles, are available in a format that requires little storage space. This makes it possible to use our approach even under offline conditions.
However, it is important to bear in mind that the quality of 3D city models is not universal and depends heavily on the modelling process. The quality of the model in turn influences the accuracy of the proposed camera poses. Further difficulties arise if only an insufficient number of building edges are visible in the query image.
To highlight the uniqueness of the different poses in such cases and in general, there are several options that we will implement. We will test how the areal information of the buildings from the z-buffer and the query image can be integrated into the score. The idea is to use image segmentation to extract the building surfaces in the query image and compare them with the visible building surfaces derived from the z-buffer. This can be used

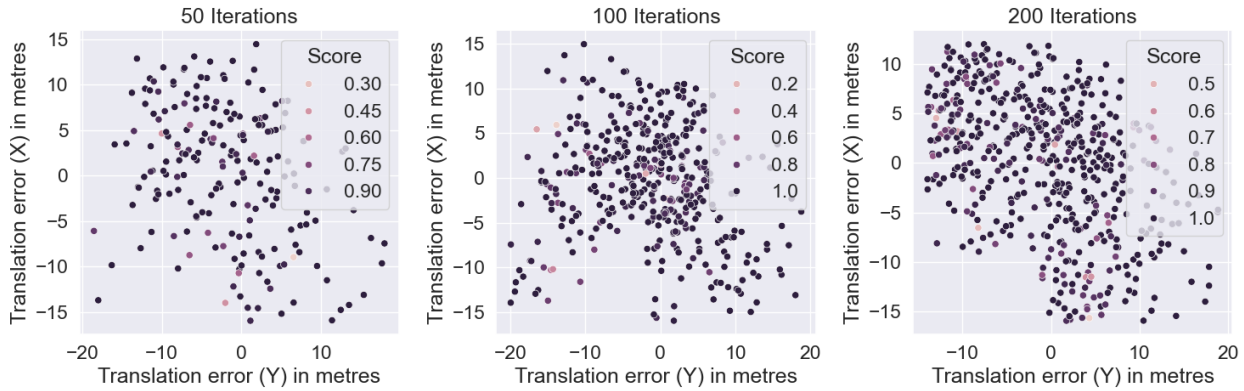

Figure 5. Scatter plot of the translation errors with different iteration steps and their score. From left to right: 50, 100 and 200 iterations. 10 images were tested.

to develop further improvement options, such as differentiating between roof surfaces and wall surfaces and comparison of these. In addition we strive to minimize the uncertainty of the score in further work and ensure that the score is stabilized by other factors, ambiguities are reduced, and the actual pose is better represented. This also concerns the use of straight line extractors: are there others which are more suitable? We will also employ quite simple image gradient computation orthogonal to the projected building edge to derive a score making directly use of low level image features.

ACKNOWLEDGEMENTS

Funded by
the European Union

This work is part of the EU-Horizon project *egeniouss*[2], which received funding under the call HORIZON-EUSPA-2021-Space with the project number 101082128.

References

Acharya, D., Ramezani, M., Khoshelham, K., Winter, S., 2019. BIM-Tracker: A model-based visual tracking approach for indoor localisation using a 3D building model. *ISPRS Journal of Photogrammetry and Remote Sensing*, 150, 157-171. 10.1016/j.isprsjprs.2019.02.014.

Acharya, D., Tatli, C. J., Khoshelham, K., 2023. Synthetic-real image domain adaptation for indoor camera pose regression using a 3D model. *ISPRS Journal of Photogrammetry and Remote Sensing*, 202, 405–421.

Acharya, D., Tennakoon, R., Muthu, S., Khoshelham, K., Hoseinnezhad, R., Bab-Hadiashar, A., 2022. Single-image localisation using 3D models: Combining hierarchical edge maps and semantic segmentation for domain adaptation. *Automation in Construction*, 136, 104152.

Akinlar, C., Topal, C., 2011. EDLines: A real-time line segment detector with a false detection control. *Pattern Recognition Letters*, 32(13), 1633–1642.

[2] https://www.egeniouss.eu/

Arandjelović, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J., 2015. NetVLAD: CNN architecture for weakly supervised place recognition. *10.48550/arXiv.1511.07247*.

Arth, C., Wagner, D., Klopschitz, M., Irschara, A., Schmalstieg, D., 2009. Wide area localization on mobile phones. *2009 8th IEEE International Symposium on Mixed and Augmented Reality*, IEEE, 73–82.

Barath, D., Ivashechkin, M., Matas, J., 2019a. Progressive NAPSAC: sampling from gradually growing neighborhoods. *10.48550/arXiv.1906.02295*.

Barath, D., Noskova, J., Ivashechkin, M., Matas, J., 2019b. MAGSAC++, a fast, reliable and accurate robust estimator. *10.48550/arXiv.1912.05909*.

Brachmann, E., Rother, C., 2017. Learning Less is More - 6D Camera Localization via 3D Surface Regression. *10.48550/arXiv.1711.10228*.

Brejcha, J., Lukáč, M., Hold-Geoffroy, Y., Wang, O., Čadík, M., 2020. Landscapear: Large scale outdoor augmented reality by matching photographs with terrain models using learned descriptors. A. Vedaldi, H. Bischof, T. Brox, J.-M. Frahm (eds), *Computer Vision – ECCV 2020*, Lecture Notes in Computer Science, 12374, Springer International Publishing, Cham, 295–312.

Castle, R., Klein, G., Murray, D. W., 2008. Video-rate localization in multiple maps for wearable augmented reality. *2008 12th IEEE International Symposium on Wearable Computers*, IEEE, 15–22.

Cavallari, T., Bertinetto, L., Mukhoti, J., Torr, P., Golodetz, S., 2019. Let's Take This Online: Adapting Scene Coordinate Regression Network Predictions for Online RGB-D Camera Relocalisation. *10.48550/arXiv.1906.08744*.

Cavallari, T., Golodetz, S., Lord, N. A., Valentin, J., Prisacariu, V. A., Di Stefano, L., Torr, P. H. S., 2020. Real-Time RGB-D Camera Pose Estimation in Novel Scenes Using a Relocalisation Cascade. *IEEE transactions on pattern analysis and machine intelligence*, 42(10), 2465–2477.

Chen, M., Fang, T., Zhu, Q., Ge, X., Zhang, Z., Zhang, X., 2021. Feature-Point Matching for Aerial and Ground Images by Exploiting Line Segment-Based Local-Region Constraints. *Photogrammetric Engineering & Remote Sensing*, 87, 767-780. 10.14358/PERS.21-00022R2.

Couturier, A., Akhloufi, M. A., 2021. A review on absolute visual localization for UAV. *Robotics and Autonomous Systems*, 135, 103666.

Couturier, A., Akhloufi, M. A., 2024. A Review on Deep Learning for UAV Absolute Visual Localization. *Drones*, 8(11), 622.

Dai, X., Gong, H., Wu, S., Yuan, X., Ma, Y., 2021. Fully Convolutional Line Parsing. *10.48550/arXiv.2104.11207*.

Fanta-Jende, P., Nex, F., Vosselman, G., Gerke, M., 2019. Co-registration of panoramic mobile mapping images and oblique aerial images. *The Photogrammetric Record*, 34, 148-173. 10.1111/phor.12276.

Fernandes, L. A., Oliveira, M. M., 2008. Real-time line detection through an improved Hough transform voting scheme. *Pattern Recognition*, 41(1), 299–314.

Frazier, P. I., 2018. A Tutorial on Bayesian Optimization. *10.48550/arXiv.1807.02811*.

Gerke, M., 2011. Using horizontal and vertical building structure to constrain indirect sensor orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66. 10.1016/j.isprsjprs.2010.11.002.

Gordo, A., Almazan, J., Revaud, J., Larlus, D., 2016. End-to-end Learning of Deep Visual Representations for Image Retrieval. *10.48550/arXiv.1610.07940*.

Grompone von Gioi, R., Jakubowicz, J., Morel, J.-M., Randall, G., 2012. LSD: a Line Segment Detector. *Image Processing On Line*, 2, 35–55.

Heng, L., Choi, B., Cui, Z., Geppert, M., Hu, S., Kuan, B., Liu, P., Nguyen, R., Yeo, Y. C., Geiger, A., Lee, G. H., Pollefeys, M., Sattler, T., 2018. Project AutoVision: Localization and 3D Scene Perception for an Autonomous Vehicle with a Multi-Camera System.

Huang, K., Wang, Y., Zhou, Z., Ding, T., Gao, S., Ma, Y., 2018. Learning to Parse Wireframes in Images of Man-Made Environments. http://arxiv.org/pdf/2007.07527v1.

Humenberger, M., Cabon, Y., Guerin, N., Morat, J., Leroy, V., Revaud, J., Rerole, P., Pion, N., de Souza, C., Csurka, G., 2020. Robust Image Retrieval-based Visual Localization using Kapture. *10.48550/arXiv.2007.13867*.

Irschara, A., Zach, C., Frahm, J.-M., Bischof, H., 2009. From structure-from-motion point clouds to fast location recognition. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2599–2606.

Jarzabek-Rychard, M., Maas, H.-G., 2017. Geometric Refinement of ALS-Data Derived Building Models Using Monoscopic Aerial Images. *Remote Sensing*, 9, 282. http://www.mdpi.com/2072-4292/9/3/282.

Kendall, A., Grimes, M., Cipolla, R., 2015. Posenet: A convolutional network for real-time 6-dof camera relocalization. *2015 IEEE International Conference on Computer Vision (ICCV)*, IEEE, 2938–2946.

Landesamt für Geoinformation und Landesvermessung Niedersachsen, 2024. Opengeodata niedersachsen. https://ni-lgln-opengeodata.hub.arcgis.com/apps/lgln-opengeodata (last visited September 24, 2024).

Laskar, Z., Melekhov, I., Kalia, S., Kannala, J., 2017. Camera Relocalization by Computing Pairwise Relative Poses Using Convolutional Neural Network. *10.48550/arXiv.1707.09733*.

Lim, H., Sinha, S. N., Cohen, M. F., Uyttendaele, M., 2012. Real-time image-based 6-dof localization in large-scale environments. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 1043–1050.

Middelberg, S., Sattler, T., Untzelmann, O., Kobbelt, L., 2014. Scalable 6-dof localization on mobile devices. D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (eds), *Computer Vision – ECCV 2014*, Lecture Notes in Computer Science, 8690, Springer International Publishing, Cham, 268–283.

Naseer, T., Burgard, W., Stachniss, C., 2018. Robust Visual Localization Across Seasons. *IEEE Transactions on Robotics*, 34(2), 289–302. 10.1109/TRO.2017.2788045.

Panek, V., Kukelova, Z., Sattler, T., 2022. MeshLoc: Mesh-Based Visual Localization. *10.48550/arXiv.2207.10762*.

Panek, V., Kukelova, Z., Sattler, T., 2023. Visual Localization using Imperfect 3D Models from the Internet. *10.48550/arXiv.2304.05947*.

Pautrat, R., Barath, D., Larsson, V., Oswald, M. R., Pollefeys, M., 2022. DeepLSD: Line Segment Detection and Refinement with Deep Image Gradients. *arXiv; 10.48550/arXiv.2212.07766*.

Pautrat, R., Lin, J.-T., Larsson, V., Oswald, M. R., Pollefeys, M., 2021. SOLD2: Self-supervised Occlusion-aware Line Description and Detection. *10.48550/arXiv.2104.03362*.

Sarlin, P.-E., Debraine, F., Dymczyk, M., Siegwart, R., Cadena, C., 2018. Leveraging Deep Visual Descriptors for Hierarchical Efficient Localization. *10.3929/ETHZ-B-000318818*.

Sarlin, P.-E., Unagar, A., Larsson, M., Germain, H., Toft, C., Larsson, V., Pollefeys, M., Lepetit, V., Hammarstrand, L., Kahl, F., Sattler, T., 2021. Back to the Feature: Learning Robust Camera Localization from Pixels to Pose. *10.48550/arXiv.2103.09213*.

Sattler, T., Leibe, B., Kobbelt, L., 2012a. Improving image-based localization by active correspondence search. D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, C. Schmid (eds), *Computer Vision – ECCV 2012*, Lecture Notes in Computer Science, 7572, Springer Berlin Heidelberg, Berlin, Heidelberg, 752–765.

Sattler, T., Weyand, T., Leibe, B., Kobbelt, L., 2012b. Image retrieval for image-based localization revisited. R. Bowden, J. Collomosse, K. Mikolajczyk (eds), *Procedings of the British Machine Vision Conference 2012*, British Machine Vision Association, 76.1–76.12.

Schönberger, J. L., Pollefeys, M., Geiger, A., Sattler, T., 2018. Semantic visual localization. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, 6896–6906.

Sibbing, D., Sattler, T., Leibe, B., Kobbelt, L., 2013. Sift-realistic rendering. *2013 International Conference on 3D Vision*, IEEE, 56–63.

Snavely, N., Seitz, S. M., Szeliski, R., 2008. Modeling the World from Internet Photo Collections. *International Journal of Computer Vision*, 80(2), 189–210.

Suárez, I., Buenaposada, J. M., Baumela, L., 2022. ELSED: Enhanced line SEgment drawing. *Pattern Recognition*, 127, 108619.

Sun, Y., Robson, S., Scott, D., Boehm, J., Wang, Q., 2019. Automatic sensor orientation using horizontal and vertical line feature constraints. *ISPRS Journal of Photogrammetry and Remote Sensing*, 150, 172-184. 10.1016/j.isprsjprs.2019.02.011.

Svarm, L., Enqvist, O., Kahl, F., Oskarsson, M., 2017. City-Scale Localization for Cameras with Known Vertical Direction. *IEEE transactions on pattern analysis and machine intelligence*, 39(7), 1455–1461.

Teplyakov, L., Erlygin, L., Shvets, E., 2022. LSDNet: Trainable Modification of LSD Algorithm for Real-Time Line Segment Detection. *IEEE Access*, 10, 45256–45265.

Zhang, Z., Sattler, T., Scaramuzza, D., 2020. Reference Pose Generation for Long-term Visual Localization via Learned Features and View Synthesis. *10.48550/arXiv.2005.05179*.

Zhou, Q., Sattler, T., Pollefeys, M., Leal-Taixe, L., 2020. To learn or not to learn: Visual localization from essential matrices. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 3319–3326.

Zhou, Y., Qi, H., Ma, Y., 2019. End-to-End Wireframe Parsing. *10.48550/arXiv.1905.03246*.