Automatic 3D Model Registration for Global Localization based on Publicly Available Georeferenced CityGML Data

Zhenyu Liu*, Christoph Blut, Jörg Blankenbach

Geodetic Institute and Chair for Computing in Civil Engineering & GIS, RWTH Aachen University, Germany - (zhenyu.liu, christoph.blut, blankenbach)@gia.rwth-aachen.de

Keywords: Localization, Registration, CityGML, Point Cloud, Feature Matching.

Abstract

Nowadays, there are many publicly available georeferenced data, like 3D CityGML models, that can be used as prior knowledge to perform accurate global localization. Iterative Closest Point (ICP) is a promising method for achieving this task, but it requires two point clouds that need to be partially overlapping in the initial state for better registration performance. Therefore, we investigated different detection and matching methods to automatically pre-register two non-overlapping point clouds based on a 2D overhead view and evaluated the registration results produced by an ICP algorithm. We used public data from the city of Aachen, Germany. A georeferenced point cloud was derived from the LOD2 CityGML model and a local point cloud was reconstructed from an image sequence using Structure from Motion (SFM). The evaluation results show that georeferenced LOD2 CityGML models can successfully be used for city-scale sub-meter global localization.

1. Introduction

Extreme weather is more common today as a result of climate change. Among the myriad consequences, urban flooding due to heavy rainfall is increasingly frequent and poses a significant risk to the safety of cities, especially those situated near rivers or other water bodies. The safety of residents and the normal functioning of cities are seriously threatened. Therefore, the development and improvement of response techniques to urban flooding has become a high priority at all levels of governance (Bosseler et al., 2021).

The rapidly developing mixed reality (MR) technology is favored by many disaster management projects due to its advantages in spatial visualization, environmental interactivity, immersive experience, and ability of team collaboration (Helzel et al., 2021). The active project “TeleTHW” aims at connecting frontline emergency responders and the disaster command center with MR devices to react to the urban flooding problem. Flood control experts from the command center will be able to access real-time information about the flooded area through the MR device, integrate it with the city emergency response system to get the best solution, and then guide emergency responders without professional backgrounds to execute the solution through the interactive and collaborative MR environment. For example, the disaster command center will be able to view the inundation extent as well as the destruction of houses and trees based on real-time environmental information of the flooded area captured by the MR devices, and then can design a construction plan for temporary flood control facilities such as sandbags or walls. The command center can then intuitively deploy the plan to the frontline emergency responders via MR, and guide them in construction. Frontline responders will also be able to see the construction progress in the MR environment or be able to report real-time disaster information to help the command center revise plans.

One of the key challenges in realizing the TeleTHW project and the MR environment global location-based localization and pose (position and orientation) tracking system. MR devices typically operate in their own local coordinate system, in which digital information is placed relative to the user. For TeleTHW, a global reference system is mandatory, so that the helpers on-site and the digital objects, such as sandbags or flood protection walls, can be coordinated correctly by the command center. While the pose of the digital objects is already provided by plans, the pose of the MR device needs to be determined every time on startup via a global localization. For outdoor spaces, a commonly used method for global localization is a combination of global navigation satellite systems (GNSS) and inertial measurement units (IMU) (Zangenehnejad & Gao, 2021). However, for MR GNSS are often not accurate enough and in many cases MR devices do not support GNSS. Due to the increasing quality of mobile cameras, as for example in smartphones, visual camera-based methods are a promising and recently much investigated method for (global) localization. One way to achieve visual localization with a simple camera is to capture the surrounding environment with it, create a 3D model from the images with structure from motion (SFM) (Ullman & Brenner, 1997) and register this local 3D model to a georeferenced 3D model with an iterative closest point (ICP) algorithm (Chetverikov et al., 2002). Nowadays, there are many openly accessible city models with global spatial references, such as CityGML models (Gröger & Plümer, 2012), that can be used as prior knowledge (reference model), but to apply an ICP algorithm, the two point clouds need to be roughly aligned already to avoid local optima.

In this paper, we evaluate the feasibility of planer feature matching algorithms with overhead views of local reconstructed point clouds as well as georeferenced CityGML city models for use as initial input for an ICP-based fine registration to obtain a global georeferenced pose.

* Corresponding author

2. Related work

2.1 3D model-based global localization

3D model-based map matching is one of the main approaches to achieve global localization of mobile devices in urban environments. A typical representative is the 3D point cloud model generated by laser scanners utilizing Light Detection and Ranging (LiDAR) technique, depth sensor, or multiple view geometry (e.g., SFM), which is widely used in 3D map matching using methods such as normal distribution transform (NDT) (Kan et al., 2021) as well as many Simultaneous Localization and Mapping (SLAM) based applications (Yin et al., 2020; Xu et al., 2022). Point cloud models perform spatial reconstruction with high accuracy and rich details, but their representation of space is discrete and requires a large amount of storage space as well as computational cost, especially in city-scale environments.

To address these drawbacks of point cloud models, Javanmardi et al. (2019) extracted 3D vector models from the original point cloud as a prior base map for localization to reduce the data size. Bureick et al. (2019) and Lucks et al. (2021) used more standardized 3D city models, such as CityGML, as prior base maps for global localization, which increased the generalizability of their localization methods. However, the above methods using 3D city models were mainly concerned with the correction of low-accuracy GNSS trajectories and the solution to the urban canyon problem. It means that they still relied on the initial global location provided by the GNSS data in order to solve the rough registration between the mobile device and the prior 3D model. Therefore, these methods cannot be directly applied to MR devices that do not support GNSS (Blut et al., 2019; Blut & Blankenbach, 2021).

Several recent studies also used 3D building models to achieve localization with purely vision-based methods. Kadosh et al. (2021) trained a convolutional neural network (CNN) model with textureless projection images of a 3D city model to estimate the camera pose. Such approaches need to address the cross-domain gap problem between model-rendered images and real camera images (Chen et al., 2022). Panek et al. (2023) evaluated the localization capabilities of imperfect 3D building models from the Internet, such as 3D game model resources. Though, they only focused on single buildings and did not expand further into larger scale areas.

Overall, there is still a large potential for exploration of 3D model-based global localization methods without GNSS assistance. Georeferenced CityGML models contain all the information necessary for global large-scale localization and, thus, can be used as a prior base data. CityGML is a more compact implementation of spatial representations than point clouds and high-resolution remote sensing images. As a standardized 3D city model, CityGML enables easier design of localization methods with stronger generalization capabilities than other 3D models. A possible solution for getting global spatial reference from CityGML to local environment is first sampling a point cloud from the prior CityGML model and one from the local environment. The registration of these two point clouds can be mostly done with methods such as ICP or recent deep learning-based methods (Aoki et al., 2019; Lu et al., 2019). However, there is an important prerequisite to achieve good performance using the above-mentioned fine registration methods, which is a good initialization, i.e., the rough alignment between the two point clouds, otherwise the registration is prone to be trapped in a local optima.

2.2 Feature-based image matching

Image matching aims to geometrically distort a target image into the common spatial reference of a source image and align their common parts on pixel-level, i.e., image registration (Jiang et al., 2021). Feature-based image registration is widely used in such tasks due to its robustness as well as generalizability. Features are specific semantic structures of an image, like corners, blobs, lines, edges, and morphological regions (Zitová & Flusser, 2003). Learnable features have received more and more attention, including classical learning-based detectors like FAST detector (Trajković & Hedley, 1998) and recent deep learning-based detectors (Joshi & Patel, 2020).

Traditional image feature matching can be divided into two types: The first one is direct matching, which directly establishes a mapping between two feature sets through spatial geometric relationships, including graph matching and point set registration. In comparison, indirect matching first establishes an initial matching relationship through the similarity of feature points or feature descriptors and distance judging from the measuring space, and then removes false matches using local/global geometric constraints (Ma et al., 2021). Data-driven learning-based approaches also impact the feature matching task, which usually use CNNs for information extraction, similarity measurement and geometric relationship estimation tasks (Kuppala et al., 2020).

3. Methodology

This section presents our approach to roughly register local environment point clouds to CityGML city models using overhead view feature point matching. Since point clouds are discrete representations of space, it is challenging to directly register a local point cloud or vector 3D model to a georeferenced point cloud for global localization through 3D feature extraction and matching. In contrast, research related to feature extraction and matching of images is more mature. Therefore, the CityGML model is first sampled as a point cloud and then projected as a 2D overhead view image together with the local point cloud. An image-based feature point detection and matching method is used to obtain the transform matrix of the two overhead view images, finally achieving the alignment of the local point cloud with the CityGML point cloud in the horizontal (*xy*-) plane. The result of our proposed method can provide the initial information for fine registration methods like ICP. The whole workflow is demonstrated in Figure 1.

3.1 Project 3D models to overhead raster images

For both the prior and local models, we focus only on their façades. There are two main reasons: The first one is that, for the low Level of Detail (LOD) CityGML model, the structure and shape of its façade is closer to the real situation compared to other components such as the roof. The second reason is that it is difficult for most handheld and head-mounted mobile device cameras to completely capture the roofs of buildings and reconstruct them in the point cloud.

For the prior city model, all its façade components are first extracted and sampled as a point cloud P_1. Next, the z-dimension of all points in P_1 is removed to project P_1 into a 2D point cloud. The minimum bounding rectangle (MBR) of this 2D point cloud is split into a 2D grid I_1 with a resolution of

0.5m, which is the overhead view of the point cloud P_1. Grid cells of I_1 that contain 2D points are considered occupied and painted white, and vice versa in black. However, the current binary image I_1 includes not only the projections of outer façades (outer edges) but also the projections of some inner façades (inner edges), such as partition walls between two neighboring houses. The inner edges need to be removed as they are not observable from the local model. The flood fill algorithm finds the connected region for a given seed point. It is used to extract the largest parcel in the image, i.e., the ground, whose contour is the expected outer edges of all buildings.

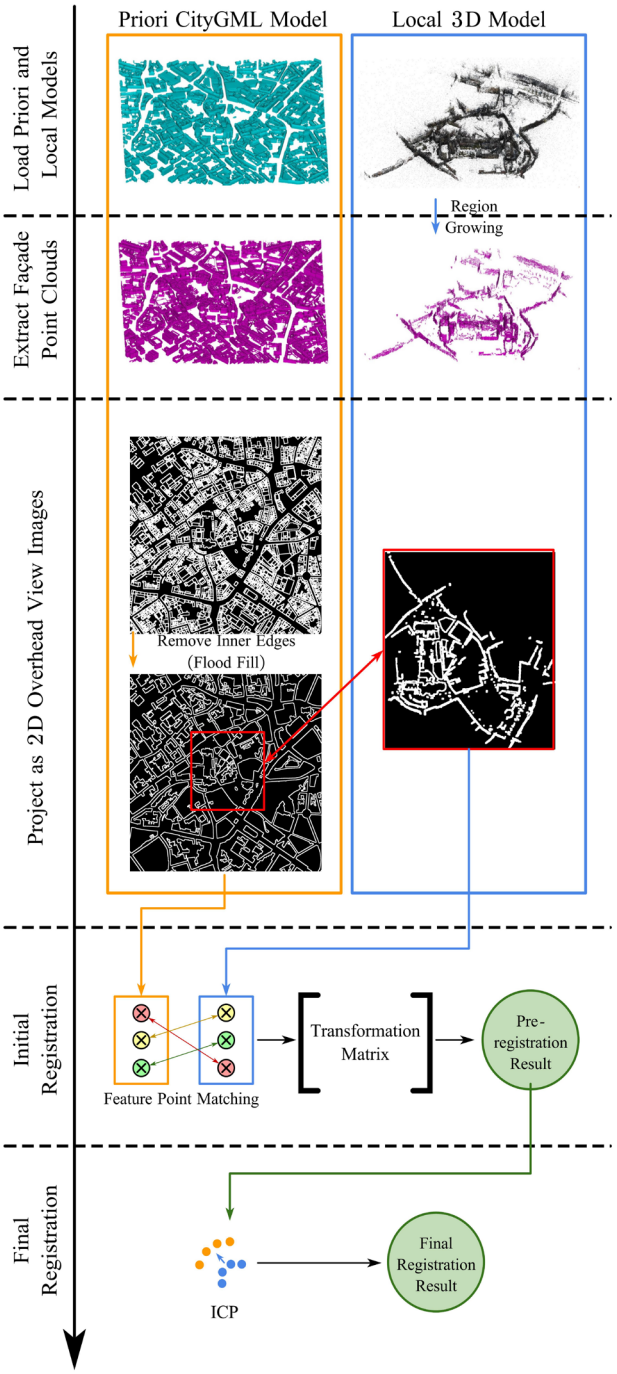

Figure 1. Workflow.

For local 3D models P_2, first noise reduction is performed, then planes are detected using the region growing algorithm, which can automatically segment a point cloud into several planes based on features like normal vector and curvature. Next, all planes with normal vectors parallel to the horizontal plane are extracted and labelled as façades. Finally, these façades are also projected as a 2D overhead raster image I_2.

3.2 Overhead view feature matching

Many state-of-the-art image feature detection and matching methods are now available to handle abstract images such as paintings (further introduced in Section 4.2). They are used to match feature points detected in the two overhead view images I_1 and I_2. The 2D affine transformation matrix T from the image I_2 to the image I_1 can be solved from the obtained feature point pairs using the Random Sample Consensus (RANSAC) optimized least squares. As shown in Eq.1, the 2D transformation matrix T can be represented as a 3×3 square matrix. a-d denote, in order, the scaling factor in the x-direction, the shear factor in the x-direction, the shear factor in the y-direction, and the scaling factor in the y-direction. t_x and t_y denote the translations along the x-direction and y-direction, respectively. The plane scaling factor s_{xy} is defined as the average of the x- and y-direction scaling factors.

$$T = \begin{bmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{bmatrix} \qquad s_{xy} = \frac{a+d}{2} \tag{1}$$

3.3 Global registration for 3D models

Each point p_{1i} of the point cloud P_1 can be transformed to a new position p_{3i} by matrix T. Assuming that the scaling factor in the z-direction is the same as in the xy-plane, thus s_{xy} is applied to the z-value of p_{3i} (Eq.2). The point cloud P_3 formed by all transformed p_{3i} is the initial registration result, which can provide the initial information for fine registration using ICP.

$$\begin{cases} \begin{bmatrix} p_{3i}.x \\ p_{3i}.y \end{bmatrix} = T \cdot \begin{bmatrix} p_{2i}.x \\ p_{2i}.y \end{bmatrix} \\ p_{3i}.z = s_{xy} \cdot p_{2i}.z \end{cases} \tag{2}$$

4. Evaluation

4.1 Data preparation

From the CityGML repository of the OpenGeodata.NRW database (IT.NRW, 2017), we extracted data for the city center of Aachen, Germany and set it as the prior city model. OpenGeodata.NRW provides CityGML models in two LODs, LOD1 and LOD2, which differ in the roof parts. However, even roofs in LOD2 model are quite different from the real roofs. Considering that our evaluation method does not rely on any other components of buildings other than the outer façade, we only used the LOD2 model during the evaluation phase, while our evaluation conclusions are also applicable to the LOD1 CityGML product of OpenGeodata.NRW.

The Aachen Day-Night dataset (Sattler et al., 2018), which includes 14,607 images of Aachen city center taken by mobile devices, was used to reconstruct a local 3D model using SFM via COLMAP (Schönberger et al., 2016; Schonberger & Frahm,

2016). Next, we manually registered the local 3D model to the prior city model to obtain the ground truth data. The façade point clouds were extracted and projected as overhead views with a pixel resolution of 0.5m.

4.2 Feature detection and matching methods

Four different methods were used to detect and match the feature points between the overhead view images of the prior city model and the local 3D model during the evaluation phase. The first one uses SuperGlue to match features extracted with SuperPoint and then filters outliners with a learned matching strategy (DeTone et al., 2018; Sarlin et al., 2020). ASLFeat uses a combination of deformable convolutional networks, the inherent feature hierarchy, and a peakiness measurement to improve the shape-awareness of feature points as well as the localization accuracy of keypoints (Luo et al., 2020). Patch2Pix first obtains patch-level matching proposal and then refines them to pixel-level matching results (Zhou et al., 2021). ncnet uses a twin structure of convolutions to obtain initial correspondences between images, and then optimizes the matching results by the neighborhood consensus scoring (Rocco et al., 2018).

The rotational invariance of some methods, such as SuperPoint + SuperGlue, is only maintained between approximately 0-45°. For extending the rotational invariance of these methods to 360°, i.e., making it possible for the overhead view image of the local model (I_2) to match the overhead view image I_1 of the prior model in any initial orientation, I_2 is rotated clockwise seven (7) times in steps of 45°. Seven rotated images and one original image are matched with I_1 respectively, and the one with the most pairs of matching feature points is marked as the best-matched image. Its feature points are rotated counterclockwise in the opposite direction back to its original orientation, and then solved with the feature points of I_1 to derive the transformation matrix T. The whole process is shown in Figure 2.

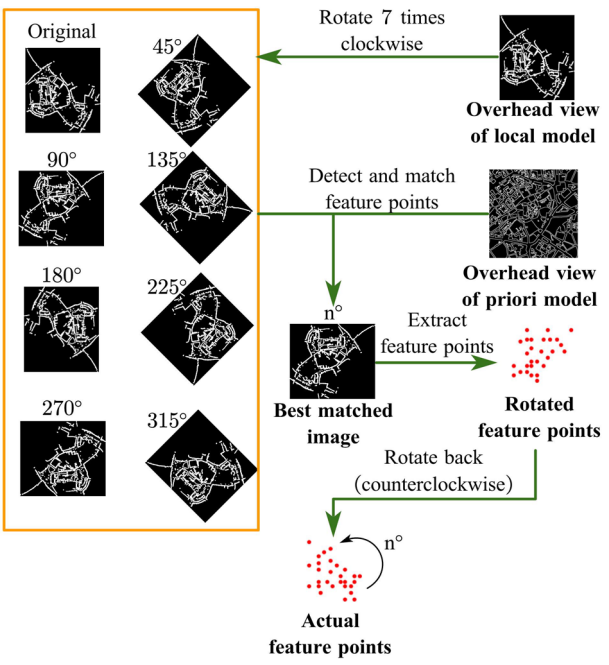

Figure 2. Extend rotational invariance.

4.3 Software requirements

The point cloud preprocessing, overhead view projection, transformation matrix solving, initial point cloud registration, and error calculation were implemented in a C++ environment with the OpenCV (Bradski, 2000) and Eigen (Eigen Development Team, 2021) libraries. The software CloudCompare (Cloudcompare Development Team, 2024) provided point cloud visualization, point sampling for the CityGML model, ground truth data alignment, and final point cloud (fine) registration (ICP).

4.4 Evaluation method

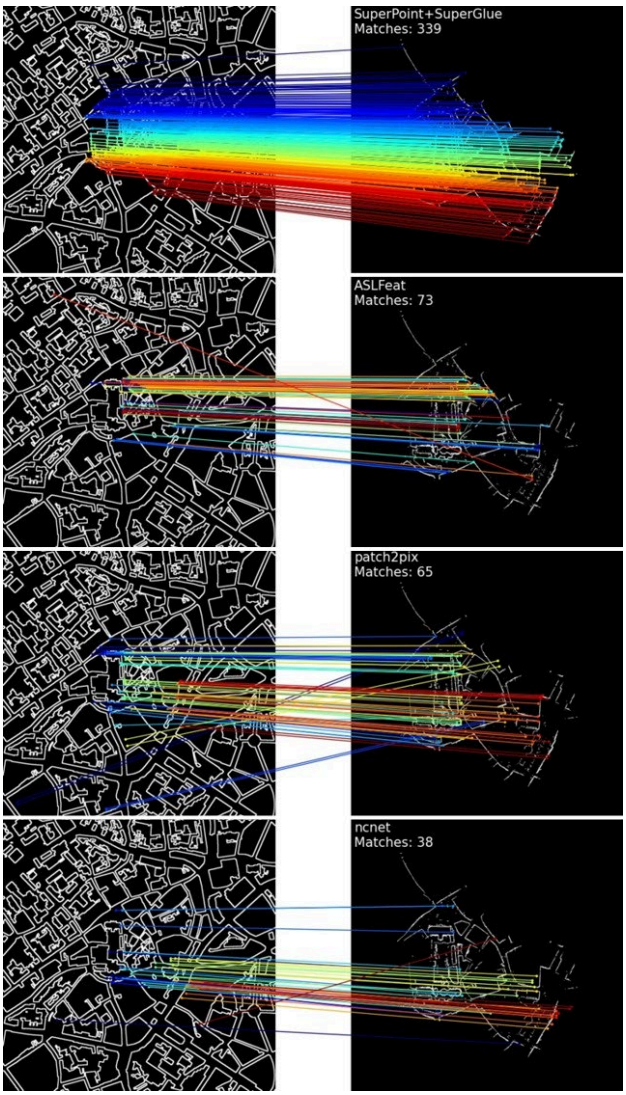

Figure 3. Feature point matching results.

In the first step, the four methods mentioned in Section 4.2 were used to detect and match feature points. The results are shown in Figure 3 and Table 1. After obtaining the transformation matrix T from the matched feature point pairs, the initial registration of the local 3D model was performed. Then the 2D error in the xy-plane between the initial registration result and the ground truth was calculated (Δ_{2D}). The initial registration result was moved along the z-axis to place its center at the same height as the center of the prior model point cloud. This step achieved the rough overlapping between the prior and local

models. Next, the moved initial registration result and the prior city model were fine registered using ICP with 50,000 iterations, was tested to ensure full convergence of final registration accuracy in our case. Since the scaling operation had already been performed in the initial registration, only the translation and rotation were performed in the ICP. The 3D error of the finial registration result was marked Δ_{3D}. The examples of the final registration results are illustrated in Figure 4. Neither affine transformation nor ICP changed the order in which the points are stored in the point cloud. Thus, Δ_{2D} and Δ_{3D} can be calculated by Eq. 3, where p_{3i}, p_{4i}, and p_{gti} are the i-th point ($i \in [1, n]$) in initial registration result, final registration result, and ground truth data, respectively. The accuracy is listed in Table 1.

$$\Delta_{2D} = \frac{1}{n}\sum_{i=1}^{n}\sqrt{(p_{3i}.x - p_{gti}.x)^2 + (p_{3i}.y - p_{gti}.y)^2} \quad (3)$$

$$\Delta_{3D} = \frac{1}{n}\sum_{i=1}^{n}\sqrt{(p_{4i}.x - p_{gti}.x)^2 + (p_{4i}.y - p_{gti}.y)^2 + (p_{4i}.z - p_{gti}.z)^2}$$

Method	Matched Point Pairs	Δ_{2D} (m)	Δ_{3D} (m)
SuperPoint + SuperGlue	339	0.578	0.706
ASLFeat	73	1.805	1.486
Patch2Pix	65	1.255	1.266
ncnet	38	2.811	2.186

Table 1. Evaluation results.

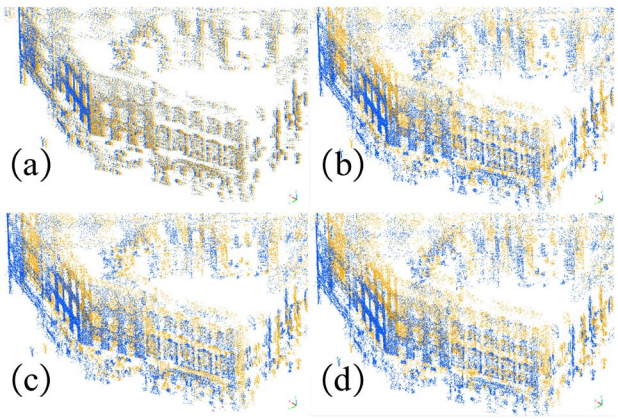

Figure 4. Partial illustration of the final registration results (blue points) and ground truth data (yellow points) using SuperPoint + SuperGlue (a), ASLFeat (b), Patch2Pix (c) and ncnet (d).

5. Discussion

All four evaluated methods could successfully detect and match the required amount of feature point pairs from the overhead view images of the prior model and the local model, for solving the transformation matrix for initial registration. ncnet matched only 38 pairs of feature points and still achieved a 2D planar accuracy Δ_{2D} of 2.811 m in the end. As the number of matched feature points increases, the 2D registration accuracy improves. SuperPoint + SuperGlue matched a total of 339 pairs of feature points and obtained a 2D registration accuracy of 0.578 m, which is very close to the pixel resolution (0.5 m) of the overhead view image during the evaluation. In addition, with our improvements shown in Figure 2, these methods demonstrate good rotational invariance, which significantly reduces the effect of the initial orientation of the overhead view images on the matching results (Figure 5). Methods that have more accurate initial registration results also perform better with the ICP algorithm. Among the four evaluated methods, SuperPoint + SuperGlue performed the best, with a 3D accuracy Δ_{3D} of 0.706 m. While ncnet is relatively underperforming, resulting in a 3D accuracy Δ_{3D} of 2.186 m.

Overall, evaluation results show that the overhead view features of the LOD2 CityGML model can provide sub-meter registration capabilities for local 3D models with unknown spatial references in city-scale, which can also be a good initial relative pose for other accurate registration methods.

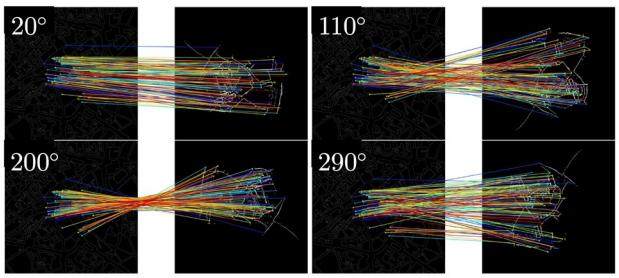

Figure 5. Feature point matching results with different initial image orientations using SuperPoint + SuperGlue (for illustration purpose, the number of matching point pairs is limited to better view the matching lines).

6. Conclusions

TeleTHW is an active research project aimed at using MR devices to connect frontline emergency responders with the disaster command center in response to increasingly frequent urban flooding. One of the key tasks is to achieve global localization of MR devices. A promising solution is to register the local environment information captured by the MR device to a georeferenced prior 3D CityGML model to obtain a global localization. Therefore, in this paper, we evaluated the 2D registration capabilities between local reconstructed point clouds and prior LOD2 CityGML format city models using feature point matching of overhead view projection images. The results show that current state-of-the-art image feature point detection and matching methods are proved to be successfully applied to overhead images. With our improvements, these evaluated methods demonstrate good rotational invariance. The 2D registration results can provide initial information for methods such as ICP for fine registration. In conclusion, the georeferenced LOD2 CityGML models can successfully be used for city-scale sub-meter global localization as prior 3D base map.

Acknowledgements

The authors gratefully acknowledge the financial support of the German Federal Ministry for Economic Affairs and Climate Action in the project ‘‘Entwicklung eines AR-Systems zur effizienten Koordinierung der Logistik- und Aufbauprozesse von Hochwasserschutzmaßnahmen (TeleTHW)” (reference number 16KN112821).

References

Aoki, Y., Goforth, H., Srivatsan, R. A., & Lucey, S. (2019). PointNetLK: Robust & Efficient Point Cloud Registration Using PointNet. 7163–7172. https://openaccess.thecvf.com/content_CVPR_2019/html/Aoki_

PointNetLK_Robust__Efficient_Point_Cloud_Registration_Using_PointNet_CVPR_2019_paper.html

Blut, C., & Blankenbach, J. (2021). Three-dimensional CityGML building models in mobile augmented reality: A smartphone-based pose tracking system. International Journal of Digital Earth, 14(1), 32–51. https://doi.org/10.1080/17538947.2020.1733680

Blut, C., Blut, T., & Blankenbach, J. (2019). CityGML goes mobile: Application of large 3D CityGML models on smartphones. International Journal of Digital Earth, 12(1), 25–42. https://doi.org/10.1080/17538947.2017.1404150

Bosseler, B., Salomon, M., Schlüter, M., & Rubinato, M. (2021). Living with Urban Flooding: A Continuous Learning Process for Local Municipalities and Lessons Learnt from the 2021 Events in Germany. Water, 13(19), Article 19. https://doi.org/10.3390/w13192769

Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools. http://www.drdobbs.com/open-source/the-opencv-library/184404319 (accessed July 27, 2023)

Chen, J., Li, S., Liu, D., & Lu, W. (2022). Indoor camera pose estimation via style-transfer 3D models. Computer-Aided Civil and Infrastructure Engineering, 37(3), 335–353. https://doi.org/10.1111/mice.12714

Chetverikov, D., Svirko, D., Stepanov, D., & Krsek, P. (2002). The Trimmed Iterative Closest Point algorithm. 2002 International Conference on Pattern Recognition, 3, 545–548 vol.3. https://doi.org/10.1109/ICPR.2002.1047997

Cloudcompare Development Team. (2024). CloudCompare—Open Source project (2.13.1) [Computer software]. Cloudcompare.org. https://www.danielgm.net/cc/

DeTone, D., Malisiewicz, T., & Rabinovich, A. (2018). SuperPoint: Self-Supervised Interest Point Detection and Description. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 337–33712. https://doi.org/10.1109/CVPRW.2018.00060

Eigen Development Team. (2021). Eigen (3.4.0) [C++]. https://eigen.tuxfamily.org/index.php?title=Main_Page

Gröger, G., & Plümer, L. (2012). CityGML – Interoperable semantic 3D city models. ISPRS Journal of Photogrammetry and Remote Sensing, 71, 12–33. https://doi.org/10.1016/j.isprsjprs.2012.04.004

Helzel, K. P., Klaus, A., & Jahnke, M. (2021). Mixed Reality Maps to help convey disaster information. Advances in Cartography and GIScience of the ICA, 3, 1–7. https://doi.org/10.5194/ica-adv-3-6-2021

IT.NRW. (2017). OpenGeodata.NRW. OpenGeodata.NRW. https://www.opengeodata.nrw.de/produkte/

Javanmardi, E., Gu, Y., Javanmardi, M., & Kamijo, S. (2019). Autonomous vehicle self-localization based on abstract map and multi-channel LiDAR in urban area. IATSS Research, 43(1), 1–13. https://doi.org/10.1016/j.iatssr.2018.05.001

Jiang, X., Ma, J., Xiao, G., Shao, Z., & Guo, X. (2021). A review of multimodal image matching: Methods and applications. Information Fusion, 73, 22–71. https://doi.org/10.1016/j.inffus.2021.02.012

Joshi, K., & Patel, M. I. (2020). Recent advances in local feature detector and descriptor: A literature survey. International Journal of Multimedia Information Retrieval, 9(4), 231–247. https://doi.org/10.1007/s13735-020-00200-3

Kadosh, M., Moses, Y., & Shamir, A. (2021). On the role of geometry in geo-localization. Computational Visual Media, 7(1), 103–113. https://doi.org/10.1007/s41095-020-0196-2

Kan, Y.-C., Hsu, L.-T., & Chung, E. (2021). Performance Evaluation on Map-Based NDT Scan Matching Localization Using Simulated Occlusion Datasets. IEEE Sensors Letters, 5(3), 1–4. https://doi.org/10.1109/LSENS.2021.3060097

Kuppala, K., Banda, S., & Barige, T. R. (2020). An overview of deep learning methods for image registration with focus on feature-based approaches. International Journal of Image and Data Fusion, 11(2), 113–135. https://doi.org/10.1080/19479832.2019.1707720

Lu, W., Wan, G., Zhou, Y., Fu, X., Yuan, P., & Song, S. (2019). DeepVCP: An End-to-End Deep Neural Network for Point Cloud Registration. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 12–21. https://doi.org/10.1109/ICCV.2019.00010

Luo, Z., Zhou, L., Bai, X., Chen, H., Zhang, J., Yao, Y., Li, S., Fang, T., & Quan, L. (2020). ASLFeat: Learning Local Features of Accurate Shape and Localization. 6588–6597. https://doi.org/10.1109/CVPR42600.2020.00662

Ma, J., Jiang, X., Fan, A., Jiang, J., & Yan, J. (2021). Image Matching from Handcrafted to Deep Features: A Survey. International Journal of Computer Vision, 129(1), 23–79. https://doi.org/10.1007/s11263-020-01359-2

Panek, V., Kukelova, Z., & Sattler, T. (2023). Visual Localization using Imperfect 3D Models from the Internet. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 13175–13186. https://doi.org/10.1109/CVPR52729.2023.01266

Rocco, I., Cimpoi, M., Arandjelović, R., Torii, A., Pajdla, T., & Sivic, J. (2018). Neighbourhood consensus networks. Proceedings of the 32nd International Conference on Neural Information Processing Systems, 31, 1658–1669. https://doi.org/10.48550/arXiv.1810.10510

Sarlin, P.-E., DeTone, D., Malisiewicz, T., & Rabinovich, A. (2020). SuperGlue: Learning Feature Matching With Graph Neural Networks. 4937–4946. https://doi.org/10.1109/CVPR42600.2020.00499

Sattler, T., Maddern, W., Toft, C., Torii, A., Hammarstrand, L., Stenborg, E., Safari, D., Okutomi, M., Pollefeys, M., Sivic, J., Kahl, F., & Pajdla, T. (2018). Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8601–8610. https://doi.org/10.1109/CVPR.2018.00897

Schonberger, J. L., & Frahm, J.-M. (2016). Structure-from-Motion Revisited. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4104–4113. https://doi.org/10.1109/CVPR.2016.445

Schönberger, J. L., Zheng, E., Frahm, J.-M., & Pollefeys, M. (2016). Pixelwise View Selection for Unstructured Multi-View Stereo. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), Computer Vision – ECCV 2016 (pp. 501–518). Springer International Publishing. https://doi.org/10.1007/978-3-319-46487-9_31

Trajković, M., & Hedley, M. (1998). Fast corner detection. Image and Vision Computing, 16(2), 75–87. https://doi.org/10.1016/S0262-8856(97)00056-5

Ullman, S., & Brenner, S. (1997). The interpretation of structure from motion. Proceedings of the Royal Society of London. Series B. Biological Sciences, 203(1153), 405–426. https://doi.org/10.1098/rspb.1979.0006

Xu, X., Zhang, L., Yang, J., Cao, C., Wang, W., Ran, Y., Tan, Z., & Luo, M. (2022). A Review of Multi-Sensor Fusion SLAM Systems Based on 3D LIDAR. Remote Sensing, 14(12), Article 12. https://doi.org/10.3390/rs14122835

Yin, H., Wang, Y., Ding, X., Tang, L., Huang, S., & Xiong, R. (2020). 3D LiDAR-Based Global Localization Using Siamese Neural Network. IEEE Transactions on Intelligent Transportation Systems, 21(4), 1380–1392. https://doi.org/10.1109/TITS.2019.2905046

Zangenehnejad, F., & Gao, Y. (2021). GNSS smartphones positioning: Advances, challenges, opportunities, and future perspectives. Satellite Navigation, 2(1), 24. https://doi.org/10.1186/s43020-021-00054-y

Zhou, Q., Sattler, T., & Leal-Taixé, L. (2021). Patch2Pix: Epipolar-Guided Pixel-Level Correspondences. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 4667–4676. https://doi.org/10.1109/CVPR46437.2021.00464

Zitová, B., & Flusser, J. (2003). Image registration methods: A survey. Image and Vision Computing, 21(11), 977–1000. https://doi.org/10.1016/S0262-8856(03)00137-9