# FlexCloud: Direct, Modular Georeferencing and Drift-Correction of **Point Cloud Maps**

Maximilian Leitenstern<sup>®a</sup>, Marko Alten, Christian Bolea-Schaser, Dominik Kulmer<sup>®b</sup>, Marcel Weinmann<sup>©c</sup> and Markus Lienkamp<sup>©d</sup>

Institute of Automotive Technology, Technical University of Munich, Boltzmannstraße 15, 85748 Garching, Germany

Mapping, Georeferencing, Sensor Fusion, Point Clouds, Autonomous Driving. Keywords:

Current software stacks for real-world applications of autonomous driving leverage map information to ensure Abstract: reliable localization, path planning, and motion prediction. An important field of research is the generation of point cloud maps, referring to the topic of simultaneous localization and mapping (SLAM). As most recent developments do not include global position data, the resulting point cloud maps suffer from internal distortion and missing georeferencing, preventing their use for map-based localization approaches. Therefore, we propose FlexCloud for an automatic georeferencing of point cloud maps created from SLAM. Our approach is designed to work modularly with different SLAM methods, utilizing only the generated local point cloud map and its odometry. Using the corresponding GNSS positions enables direct georeferencing without additional control points. By leveraging a 3D rubber-sheet transformation, we can correct distortions within the map caused by long-term drift while maintaining its structure. Our approach enables the creation of consistent, globally referenced point cloud maps from data collected by a mobile mapping system (MMS). The source code of our work is available at https://github.com/TUMFTM/FlexCloud.

#### **INTRODUCTION** 1

So-called High-Definition (HD) maps are considered a key technology to enable real-world autonomous driving (Seif and Hu, 2016) as they provide essential prior information on static surroundings of the vehicle, including temporally occluded objects (Bao et al., 2022; Srinara et al., 2023). Therefore, HD maps can be seen as an additional sensor facilitating localization and motion planning by combination with other sensors, such as a Global Navigation Satellite System (GNSS), an Inertial Measurement Unit (IMU), or a Light Detection and Ranging (LiDAR) system. Unlike traditional Standard-Definition (SD) maps used for navigation, HD maps feature an accuracy of 10 cm to 20 cm (Jeong et al., 2022). Depending on the application's requirements, different map representations evolved (Khoche et al., 2022). Hence, an HD map is usually split into a point cloud map (PCM) and a vector map (Jeong et al., 2022). While a PCM de-



Figure 1: Steps for HD map generation (extended from (Srinara et al., 2023)).

picts the 3D structure of objects by individual points, a vector map represents the location of relevant objects (e.g., lanes, traffic signs) by combining points, lines, and polygons (Jeong et al., 2022). Leveraging the strengths of the respective map format, the PCM is used for localization while the vector map assists during motion planning (Jeong et al., 2022).

HD map generation can be classified into offline and online approaches (Bao et al., 2022). Following the architecture of the presented approach, this work classifies as a tool to enhance the offline generation of HD maps. This process generally comprises the steps

In Proceedings of the 11th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2025), pages 157-165 ISBN: 978-989-758-745-0; ISSN: 2184-495X

<sup>&</sup>lt;sup>a</sup> https://orcid.org/0009-0008-6436-7967

<sup>&</sup>lt;sup>b</sup> https://orcid.org/0000-0001-7886-7550

<sup>&</sup>lt;sup>c</sup> https://orcid.org/0009-0008-7174-4732

<sup>&</sup>lt;sup>d</sup> https://orcid.org/0000-0002-9263-5323

Leitenstern, M., Alten, M., Bolea-Schaser, C., Kulmer, D., Weinmann, M. and Lienkamp, M FlexCloud: Direct, Modular Georeferencing and Drift-Correction of Point Cloud Maps DOI: 10.5220/0013359600003941

Paper published under CC license (CC BY-NC-ND 4.0)

Proceedings Copyright © 2025 by SCITEPRESS - Science and Technology Publications, Lda

shown in Figure 1. It starts with data collection using a MMS (Bao et al., 2023). During data preprocessing, several algorithms are applied to the collected LiDAR data to facilitate the creation of the PCM in the following step (Bao et al., 2023; Srinara et al., 2023). After that, a PCM is created by aligning the individual LiDAR scans, usually using scan registration techniques. In the last step, the vector map is created from the PCM, completing the HD map.

The generation of PCMs from sensor data is extensively studied in the literature, usually known as SLAM. However, recent developments within this field focus on approaches that only use LiDAR and, occasionally, IMU data. Although this reduces the required data sources and avoids calibration or time synchronization issues, the created local PCM inherently suffers from long-term drift and missing global reference. Hence, georeferencing and correction for the long-term drift are necessary to leverage these maps for localization, especially if global GNSS coordinates are used with a LiDAR-based approach. In a previous work of ours (Leitenstern et al., 2024), we presented an approach to overcome this issue. However, our approach still suffered from the required manual effort and the limitation to 2D. Therefore, in this paper, we present FlexCloud as a modular pipeline to overcome these limitations. Our contributions are as follows:

- We propose a novel, modular approach to automatically georeference local PCMs created from SLAM.
- We use an interpolation-based strategy to automatically determine control points (CPs) from the corresponding GNSS positions, considering their uncertainty.
- We implement a 3D rubber-sheet transformation with Delaunay triangulation to effectively correct the long-term drift of the PCM while maintaining its structure and continuity.
- We open-source our pipeline as a standalone *ROS* 2-package upon acceptance of our work.

We evaluate the results of our pipeline on selfrecorded data at the Yas Marina Circuit (YMC) in Abu Dhabi, UAE. Additionally, we present the results for sequence 00 of the KITTI vision benchmark suite (Geiger et al., 2012; Geiger et al., 2013) to demonstrate its generalizability.

## 2 RELATED WORK

This chapter classifies different methods for georeferencing of mapping data and presents related work regarding the creation of PCMs.

#### 2.1 Georeferencing of Mapping Data

Generally, georeferencing describes the problem of finding the transformation parameters of a local set of points to a common global reference system, such as WGS84 (Paffenholz, 2012; Lasse Klingbeil, 2023). Initially, a LiDAR captures point clouds in a local coordinate system. With scan registration techniques, the relative transformation between single scans can be computed, and thus, the scans can be assembled to a local PCM. The transformation of this PCM to a global frame is referred to as georeferencing and is described by the following equation (Wilkinson et al., 2010):

$$x^G = sR^G_S x^S + t^G_S. \tag{1}$$

The transformation of a point  $x^S$  in the PCM consists of the rotation  $R_S^G$ , the translation  $t_S^G$  and the scaling parameter *s*, with the indices *S* and *G* referring to the local, LiDAR frame and the global frame, respectively. According to Paffenholz (Paffenholz, 2012), georeferencing techniques can be classified as follows:

#### • Indirect Georeferencing:

These methods use well-spread ground CP, whose coordinates are known locally and globally. The coordinates may be obtained by detection approaches and surveys in local and global frames. The additional effort needed to capture the ground CP is a drawback, making the method timeconsuming.

• Direct Georeferencing:

This approach utilizes data from external sensors, usually a GNSS receiver, attached to the LiDAR during data acquisition. Thus, the transformation parameters are directly available from the data acquisition.

#### • Data-Driven Georeferencing:

These methods leverage existing georeferenced data to match the newly acquired data. However, this requires a preceding indirect or direct georeferencing to generate this data.

As our approach classifies as a direct method for georeferencing, the following section reviews existing approaches within this field.

### 2.2 Direct Point Cloud Georeferencing

In 2005, Schuhmacher and Böhm (Schuhmacher and Boehm, 2005) compared different techniques for georeferencing in architectural modeling applications. For direct georeferencing, they combined a low-cost GNSS system with a digital compass to get the translation and rotation parameters of the transformation, respectively. Wilkinson et al. (Wilkinson et al., 2010) utilize a dual-antenna GNSS system to capture the orientation of the LiDAR. As the accuracy suffers in bad GNSS conditions, they combine it with photogrammetric observations of a camera in a subsequent work (Wilkinson et al., 2015). Osada et al. (Osada et al., 2017) propose a method for direct georeferencing that leverages an external, global earth gravity model and thus only requires a minimum number of GNSS measurements. They use a least squares method to estimate the transformation parameters from a GNSS position and the vertical deflection components of the LiDARs rotational axis, determined within the gravitational force field of the earth gravity model.

However, the approaches presented so far have in common that they do not consider the mapping system in a mobile operating mode, thus neglecting uncertainties from kinematic measurement noise and geometric falsification offsets (Liu et al., 2021). To georeference LiDAR data from an MMS, Oria-Aguilera et al. (Oria-Aguilera et al., 2018) leverage a Real-Time Kinematic (RTK) GNSS system, providing centimeter-level accuracy. They use interpolation to relate the LiDAR scans with GNSS poses to account for issues from different frequencies and time synchronization. The GNSS receiver's Inertial Navigation System (INS) is used to directly determine the transformation parameters for georeferencing without using scan registration techniques. A drawback of this method is the reliance on the INS solution and, thus, good GNSS coverage. To this end, Hariz et al. (Hariz et al., 2021) present an approach for creating georeferenced PCMs using SLAM. They use the Cartographer SLAM (Hess et al., 2016), taking LiDAR point clouds, IMU measurements, and RTKcorrected GNSS positions as input to create submaps, which are later concatenated to a global map. Koide et al. (Koide et al., 2019) present a similar approach, including GNSS constraints in a graph-based SLAM algorithm to generate georeferenced PCMs.

However, many advancements in LiDAR odometry and SLAM concentrate on algorithms that rely only on LiDAR and potentially IMU data, e.g. *LOAM* (Zhang and Singh, 2014), *LeGO-LOAM* (Shan and Englot, 2018), *SuMa*++ (Chen et al., 2019), *LIO-SAM* (Shan et al., 2020), *FAST-LIO* (Xu and Zhang, 2021; Xu et al., 2021), and *KISS-ICP* (Vizzo et al., 2023). As the PCMs generated with these approaches are in a local frame and distorted due to long-term drift, we presented *FlexMap Fusion* in a previous work (Leitenstern et al., 2024). It leverages rubber sheeting, an approach for the alignment of two spatial datasets (Marvin S. White and Griffin, 1985), based on RTK-corrected GNSS positions and the SLAM trajectory. Although this work demonstrates the general concept of the approach, it is limited to 2D and requires manual effort to select CPs on the GNSS and SLAM trajectories.

# **3** METHODOLOGY

To enable georeferencing of PCMs generated from SLAM algorithms, we present FlexCloud. It is designed to work with various SLAM approaches and can transform the generated PCM to a global frame while correcting the long-term drift resulting from scan registration errors. FlexCloud is intended for use with an MMS, providing LiDAR and GNSS data, preferably from an RTK-corrected GNSS receiver. Figure 2 illustrates the usage of *FlexCloud* based on these inputs. After data recording, an odometry trajectory and a corresponding initial PCM must be generated using SLAM. The experiments in this work are conducted with trajectories generated from KISS-ICP (Vizzo et al., 2023) with additional loop closure constraints inserted using Interactive SLAM (Koide et al., 2021). This process proved robust and accurate results for PCM generation, as shown in Sauerbeck et al. (Sauerbeck et al., 2023). The resulting odometry trajectory with the corresponding PCM and the GNSS trajectory are the inputs to FlexCloud, consisting of the modules Keyframe Interpolation, Rigid Alignment, and Rubber-Sheet Transformation. The following sections describe the modules in detail.

### 3.1 Keyframe Interpolation

This module determines temporal correspondences between the GNSS and odometry trajectories, later used to select CPs for georeferencing. As a first step, the global GNSS positions are transformed into metric coordinates to facilitate mathematical operations and computations. *FlexCloud* implements a transformation to the East-North-Up (ENU) frame using *GeographicLib*<sup>1</sup>. However, other transformations, such as the Universal Transverse Mercator (UTM) projection, may be used as well (Groves, 2013).

Our *Keyframe Interpolation* is based on the use of the multidimensional spline curves implemented in  $Eigen^2$ . The shape of the B-spline, defined by a set of

<sup>&</sup>lt;sup>1</sup>https://geographiclib.sourceforge.io/

<sup>&</sup>lt;sup>2</sup>https://eigen.tuxfamily.org

VEHITS 2025 - 11th International Conference on Vehicle Technology and Intelligent Transport Systems



Figure 2: Flowchart illustrating the creation of a global, georeferenced PCM using *FlexCloud*. The computations within the single modules are conducted with the global GNSS trajectory and the local odometry trajectory. The resulting transformations from the *Rigid Alignment* and the *Rubber-Sheet Transformation* are then applied to the PCM.



Figure 3: Principle of *Keyframe Interpolation*. For interpolation of the global position corresponding to the odometry position at time  $t_o$ , four reference points on the global trajectory (blue points) are necessary. The interpolated global position  $p_{g,t_0,inter}$  is on the resulting spline (green).

GNSS keyframes, follows the equation:

$$C(u) = \sum_{i=0}^{n} N_{i,p}(u) P_i$$
 (2)

A point C(u) on the spline is described by the basis function  $N_{i,p}(u)$  of degree p, the curve parameter u, and the reference point  $P_i$ . We implement a spline of degree p = 3, requiring four reference points. The reference points are selected by determining the two temporally closest GNSS frames before and after the current odometry point. A minimum spatial distance threshold between the GNSS frames used for interpolation ensures their well-spread distribution for reliable interpolation. The timestamp of the odometry keyframe is normalized to the range [0,1], with the smallest GNSS timestamp serving as the minimum and the biggest GNSS timestamp as the maximum value. The interpolated position is calculated using the normalized value. Figure 3 illustrates the functionality of the Keyframe Interpolation graphically.

#### 3.2 Rigid Alignment

Given the interpolated trajectories, the odometry trajectory is optimally aligned to the global trajectory by applying a rigid transformation. To find the transformation, Umeyama's (Umeyama, 1991) approach is used, minimizing the squared error between two point sets  $\{x_i\}$  and  $\{y_i\}$  with *n* elements:

$$e^{2}(R,t,s) = \frac{1}{n} \sum_{i=1}^{n} ||y_{i} - (sRx_{i} + t)||^{2}$$
(3)

The transformation consists of the rotation matrix R, the translation vector t, and a scaling factor s. However, as both trajectories are the same scale, only the rotational and translational parts of the transformation are applied to the odometry trajectory and the PCM.

#### 3.3 Rubber-Sheet Transformation

The final step is a piecewise linear *Rubber-Sheet Transformation*, derived from the research area of cartography (Gillman, 1985). Given a global reference map, it allows to correct distortions within a map by dividing it into subdivisions with different transformations (Marvin S. White and Griffin, 1985). As existing approaches are designed for standard 2D maps, we elevate the approach of Leitenstern et al. (Leitenstern et al., 2024) to 3D and improve the CP selection and the triangulation method.

The input parameters for the *Rubber-Sheet Transformation* are the interpolated, global trajectory and the rigidly aligned odometry trajectory. CPs are automatically selected in the first step, leveraging the correspondence between both trajectories after the *Keyframe Interpolation*. A configurable total amount of CPs is evenly distributed over the trajectories. However, if the GNSS standard deviation of a potential CP exceeds a threshold, the CP is skipped. The set of CPs is then extended by the eight points, forming a cuboid that encloses the trajectories with a configurable offset. In the next step, a Delaunay triangulation is created from the CPs, using the implementation of CGAL (Jamin et al., 2024), guaranteeing its uniqueness. The Delaunay triangulation is a well-defined and angle-optimal triangulation on a finite set of points, and thus, ideal for the given application (Gillman, 1985). The transformation matrices of the resulting tetrahedra are calculated based on the CPs defining their corners:

$$p_{g,i} = T_j p_{o,i}$$
  $i \in \{k, l, m, q\}$  (4)

Here,  $T_i$  presents the linear transformation matrix of a single tetrahedron, defined by the four CPs  $p_{g,i}$  and  $p_{o,i}$  on the global and odometry trajectory, respectively. Leveraging this relationship for every vertex i leads to a linear system of equations with 16 variables, defining the components of  $T_j$ . To solve this system and thus compute the transformation matrices  $T_i$  for the single tetrahedra, we employ the Householder QR decomposition with full pivoting provided by Eigen. Finally, the odometry trajectory and the PCM are transformed by locating the corresponding tetrahedron for each point x, and applying the corresponding transformation matrix:

$$x' = T_j x \tag{5}$$

While the transformation within a tetrahedron remains linear, the boundaries are adjusted in a way that preserves the topological properties of the map. Figure 4 exemplarily illustrates a resulting continuous PCM. The whole pipeline is implemented as a stan-

#### RESULTS 4

Both datasets, the YMC, and the KITTI dataset, comprise RTK-corrected GNSS positions and LiDAR point clouds, used as input to FlexCloud. As georeferencing requires the GNSS positions in a global frame, raw data of the KITTI dataset is used. However, it is noted at this point that due to the urban environment in sequence 00 of the KITTI dataset, the GNSS positions feature high standard deviations for several sections. The GNSS positions recorded at the YMC are post-processed and show standard deviation in the centimeter range, except for short occlusions. The initial odometry trajectory and corresponding PCM are created as described in Section 3. The evaluation consists of two parts: First, we present quantitative results on the deviation of the final odometry to the GNSS trajectory. For this, the Euclidean norm is computed for each point on the odometry trajectory and its corresponding point on the GNSS trajectory after the Keyframe Interpolation. In contrast to the Rubber-Sheet transformation, which only uses selected CPs, the evaluation is conducted on all trajectory points without excluding points with high standard deviation. The second part of the evaluation is conducted on publicly available satellite imagery. This enables a qualitative evaluation with an external data source.

The resulting trajectories after the Keyframe Interpolation consist of 2560 and 1550 poses for the YMC and the KITTI dataset, respectively. To improve the understanding of the pipeline, intermediate results are presented for the YMC. Figure 5 shows the devia-



Figure 4: Excerpt of a final PCM with color-coded point deformations by the Rubber-Sheet Transformation at the YMC. Gray points represent the PCM after Rigid Alignment with the orange mark being a selected CP subdividing the excerpt into tetrahedra.

dalone ROS 23 package, leveraging RVIZ to provide an interactive visualization of the final result.



Figure 5: Deviation of the odometry to the GNSS trajectory (gray) after Rigid Alignment at the YMC.

tion of the odometry to the GNSS trajectory after the Rigid Alignment step. Although the two trajectories are optimally aligned with a rigid transformation at this point, the deviation exceeds 15 m in certain sections. However, as the deviation is not constant, this highlights the distortion of the trajectory and thus the need for the Rubber-Sheet Transformation.

15

<sup>&</sup>lt;sup>3</sup>https://ros.org/

Figure 6 illustrates the triangulation of the odometry trajectory in the *Rubber-Sheet Transformation*. First, the CPs are selected based on the configuration, and the maximum allowed standard deviation of the GNSS positions (Figure 6a). In the second step, the corner points of the enclosing cuboid are added, and the Delaunay triangulation is computed.

Leveraging the transformation matrices computed from the triangulation, the Rubber-Sheet Transformation is applied according to Equation 5 to the odometry trajectory and the PCM. The deviation of the final odometry to the GNSS trajectory using 200 CPs for the YMC is illustrated in Figure 7. Comparing the final deviation to the intermediate results after the Rigid Alignment step (Figure 5), the benefit of the Rubber-Sheet Transformation in rectifying the trajectory is clearly visible. Except for a short section  $(x \approx 400 \,\mathrm{m}, y \approx -800 \,\mathrm{m})$ , where no CPs are selected following high GNSS standard deviations, the deviation falls below 1 m for the entire trajectory. The results for sequence 00 of the KITTI dataset in Figure 8 show similar results. Analogous to the YMC, the deviation increases within a section in the middle of the trajectory due to high GNSS standard deviations preventing the CP selection. As this shows the influence of the amount and location of the CPs, Table 1 compares the results of different quantities of CPs statistically. For both datasets, the MAE decreases with an

Table 1: Quantitative results for the mean absolute error (MAE), the standard deviation (STDDEV), and the maximum value (MAX) of the deviation of the georeferenced odometry trajectory to the GNSS trajectory. Results are provided for different amounts of CP  $n_{cp}$  with a standard deviation threshold of 0.05 m and 0.25 m for the YMC and the KITTI dataset, respectively.

<b>Dataset</b>   <i>n<sub>cp</sub></i>		Metric in m		
		MAE	STDDEV	MAX
	10	1.71	1.59	9.11
Yas Marina	50	0.25	0.32	2.56
Circuit	100	0.13	0.16	1.43
	200	0.08	0.11	1.27
	10	1.26	0.77	4.03
	50	0.60	0.77	3.71
KII II 00	150	0.47	0.83	4.07

increasing amount of CPs, leading to an average error of 0.08 m and 0.47 m for the YMC and the KITTI dataset, respectively. This is expected as the distance between successive CPs decreases. The larger value for the KITTI dataset follows from an overall worse quality of the GNSS positions. While STDDEV and MAX also decrease with an increasing amount of CPs for the YMC, this is not observable for the KITTI dataset. This is due to the long section with insufficient GNSS accuracy in the middle of the trajectory. It is noticeable that for  $n_{cp} = 10$ , the KITTI results are superior to the YMC despite the worse quality of the GNSS trajectory. This can be explained by the shorter overall length of the KITTI dataset, causing a reduced distance between consecutive CPs.

We leverage aerial photography to evaluate the results independently of the GNSS positions, which are also used as input to FlexCloud. Although this does not enable a quantitative but a qualitative evaluation, it allows further investigation of the performance of Flex-*Cloud*, especially within the sections with high GNSS uncertainty. For the YMC, the south part is visualized over satellite imagery from ESRI (Esri, 2024) (Figure 9). A visual comparison indicates a strong overlap between the PCM and the satellite image, especially in the turn in the southeast of the image. Although Figure 7 shows an increased deviation between the trajectories in this section, no deterioration in the overlap is visible. This proves the ability of FlexCloud to enable reliable georeferencing even for short sections with insufficient GNSS coverage.

As the KITTI dataset was recorded in the state of Baden-Württemberg, Germany, public orthophotos with a resolution of 20 cm are available for evaluation (Figure 10). The shown excerpt is taken from the southwest of the trajectory ( $x \approx -60$  m,  $y \approx 70$  m) featuring good GNSS coverage. From the high resolution of the orthophoto, it is visible that the road boundaries and building structures of the PCM accurately match the orthophoto. This indicates that PCMs georeferenced with *FlexCloud* can reach the required accuracy for HD maps given a global trajectory with sufficient accuracy.

## 5 DISCUSSION

The presented results demonstrate that *FlexCloud* can accurately georeference a given local PCM based on the corresponding odometry and GNSS trajectory. Nevertheless, our approach still features limitations and areas for improvement, which are discussed in this section. By introducing the *Keyframe Interpolation*, we can directly utilize the GNSS trajectory recorded by the MMS to generate CPs for georeferencing. Although this avoids the need for additional measurements, it has the following limitations: To reach an accuracy sufficient for autonomous driving applications, the MMS needs to be equipped with an expensive RTK-corrected GNSS receiver. Furthermore, the *Keyframe Interpolation* requires proper

<sup>&</sup>lt;sup>4</sup>https://www.lgl-bw.de/



(a) Automatically selected CP (orange dots) on odometry trajectory.

(b) Tetrahedra of resulting Delaunay triangulation.

Figure 6: Exemplary triangulation of the odometry trajectory (blue) using  $n_{cp} = 10$  CP on the YMC. The size of the enclosing cuboid equals 0.1 and 10 times the expansion of the trajectory in x/y and z, respectively.



Figure 7: Final deviation of the odometry trajectory after *Rubber-Sheet Transformation* using  $n_{cp} = 200$  CP at the YMC.



Figure 8: Final deviation of the odometry trajectory after *Rubber-Sheet Transformation* using  $n_{cp} = 150$  CP on sequence 00 of the KITTI dataset (Geiger et al., 2012).

time synchronization between the GNSS receiver and the LiDAR, as the interpolation leverages the timestamp differences between the trajectory positions.

The *Rubber-Sheet Transformation* leverages several CPs, a subset of all positions of the trajectories, to compute the transformation of the PCM. By excluding GNSS positions with high uncertainty from being



Figure 9: Visualization of PCM of the YMC on satellite imagery from ESRI (Esri, 2024).



Figure 10: Visualization of PCM of KITTI sequence 00 over orthophotos of the State Office for Geoinformation and Rural Development (LGL)<sup>4</sup> of Baden-Württemberg.

selected as CP, these sections can still be georeferenced to a certain extent. However, this requires a reliable standard deviation estimation of the GNSS receiver. Furthermore, our current implementation does not utilize the actual values of the GNSS standard deviation but only excludes high values based on a fixed threshold. Hence, additional focus must be spent on post-processing the GNSS positions, e.g., using curvature correction. Another point to be investigated is the shape structure surrounding the trajectories/map during the *Rubber-Sheet Transformation*. Although the rectangular cuboid used within this work shows robust results, limitations arise, e.g., for trajectories/maps, where the surrounding cuboid features large variations in the side lengths. An alternative approach would be the replacement of the cuboid with a polygon-like shape.

## 6 CONCLUSION

FlexCloud provides a framework for direct georeferencing and drift-correction of local PCMs created from SLAM. Following the modular design, our approach can be combined with different SLAM pipelines, utilizing only the generated odometry trajectory and local PCM. By implementing the Keyframe Interpolation, the pipeline can directly leverage the GNSS positions recorded with an MMS, avoiding the need for additional measurements. The Rubber-Sheet Transformation enables rectification of the local PCM and georeferencing of short sections without accurate GNSS positions. Although our approach still features limitations, it provides the opportunity to georeference PCMs with high accuracy, allowing their usage for map-based localization algorithms. Future work on the pipeline needs to focus on the boundary shape used for the Rubber-Sheet Transformation and improved processing of the initial GNSS positions based on their standard deviation.

## ACKNOWLEDGEMENTS

As the first author, Maximilian Leitenstern initiated and designed the paper's structure. He contributed to designing and implementing the overall pipeline concept. Christian Bolea-Schaser and Marko Alten contributed to designing and implementing the Rubber-Sheet Transformation and the Keyframe Interpolation during their student research project, respectively. Dominik Kulmer, Marcel Weinmann, and Markus Lienkamp revised the paper critically for important intellectual content. Markus Lienkamp gives final approval for the version to be published and agrees to all aspects of the work. As a guarantor, he accepts responsibility for the overall integrity of the paper.

#### REFERENCES

- Bao, Z., Hossain, S., Lang, H., and Lin, X. (2022). High-Definition Map Generation Technologies For Autonomous Driving. *ArXiv*, abs/2206.05400.
- Bao, Z., Hossain, S., Lang, H., and Lin, X. (2023). A review of high-definition map creation methods for autonomous driving. *Engineering Applications of Artificial Intelligence*, 122:106125.
- Chen, X., Milioto, A., Palazzolo, E., Giguere, P., Behley, J., and Stachniss, C. (2019). SuMa++: Efficient LiDARbased Semantic SLAM. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE.
- Esri (2024). Arcgis world imagery. Available at: https://www.arcgis.com/apps/mapviewer/index. html?layers=10df2279f9684e4a9f6a7f08febac2a9 (accessed on 2024-11-29).
- Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237. \_eprint: https://doi.org/10.1177/0278364913491297.
- Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. In 2012 IEEE Conference on Computer Vision and Pattern Recognition, pages 3354–3361.
- Gillman, D. (1985). Triangulations for rubber-sheeting. In Proceedings of 7th International symposium on computer assisted cartography (AutoCarto 7), volume 199.
- Groves, P. (2013). Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Second Edition. Artech.
- Hariz, F., Souifi, H., Leblanc, R., Bouslimani, Y., Ghribi, M., Langin, E., and Mccarthy, D. (2021). Direct Georeferencing 3D Points Cloud Map Based on SLAM and Robot Operating System. In 2021 IEEE International Symposium on Robotic and Sensors Environments (ROSE), pages 1–6.
- Hess, W., Kohler, D., Rapp, H., and Andor, D. (2016). Realtime loop closure in 2d lidar slam. In 2016 IEEE International Conference on Robotics and Automation (ICRA), pages 1271–1278.
- Jamin, C., Pion, S., and Teillaud, M. (2024). 3D triangulations. In CGAL User and Reference Manual. CGAL Editorial Board, 6.0.1 edition.
- Jeong, J., Yoon, J. Y., Lee, H., Darweesh, H., and Sung, W. (2022). Tutorial on High-Definition Map Generation for Automated Driving in Urban Environments. *Sensors*, 22(18).
- Khoche, A., Wozniak, M. K., Duberg, D., and Jensfelt, P. (2022). Semantic 3D Grid Maps for Autonomous Driving. 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), pages 2681–2688.
- Koide, K., Miura, J., and Menegatti, E. (2019). A portable three-dimensional LIDAR-based system for long-term and wide-area people behavior measurement. *International Journal of Advanced*

*Robotic Systems*, 16(2):1729881419841532. \_eprint: https://doi.org/10.1177/1729881419841532.

- Koide, K., Miura, J., Yokozuka, M., Oishi, S., and Banno, A. (2021). Interactive 3D Graph SLAM for Map Correction. *IEEE Robotics and Automation Letters*, 6(1):40–47.
- Lasse Klingbeil (2023). Georeferencing of Mobile Mapping Data. PhD Thesis, Rheinische Friedrich-Wilhelms-Universität Bonn.
- Leitenstern, M., Sauerbeck, F., Kulmer, D., and Betz, J. (2024). FlexMap Fusion: Georeferencing and Automated Conflation of HD Maps with OpenStreetMap. In 35th IEEE Intelligent Vehicles Symposium, IV 2024, IEEE Intelligent Vehicles Symposium, Proceedings, pages 272–278. Institute of Electrical and Electronics Engineers Inc.
- Liu, W., Li, Z., Sun, S., Du, H., and Sotelo, M. A. (2021). Georeferencing kinematic modeling and error correction of terrestrial laser scanner for 3D scene reconstruction. *Automation in Construction*, 126:103673.
- Marvin S. White, J. and Griffin, P. (1985). Piecewise linear rubber-sheet map transformation. *The American Cartographer*, 12(2):123–131.
- Oria-Aguilera, H., Alvarez-Perez, H., and Garcia-Garcia, D. (2018). Mobile LiDAR Scanner for the Generation of 3D Georeferenced Point Clouds. In 2018 IEEE International Conference on Automation/XXIII Congress of the Chilean Association of Automatic Control (ICA-ACCA), pages 1–6.
- Osada, E., Sośnica, K., Borkowski, A., Owczarek-Wesołowska, M., and Gromczak, A. (2017). A Direct Georeferencing Method for Terrestrial Laser Scanning Using GNSS Data and the Vertical Deflection from Global Earth Gravity Models. *Sensors*, 17(7).
- Paffenholz, J.-A. (2012). Direct geo-referencing of 3D point clouds with 3D positioning sensors. PhD Thesis, Leibniz-Universität Hannover.
- Sauerbeck, F., Kulmer, D., Pielmeier, M., Leitenstern, M., Weiß, C., and Betz, J. (2023). Multi-lidar localization and mapping pipeline for urban autonomous driving. In 2023 IEEE SENSORS, pages 1–4.
- Schuhmacher, S. and Boehm, J. (2005). Georeferencing of Terrestrial Laserscanner Data for Applications in Architectural Modelling. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 36.
- Seif, H. G. and Hu, X. (2016). Autonomous Driving in the iCity—HD Maps as a Key Challenge of the Automotive Industry. *Engineering*, 2(2):159–162.
- Shan, T. and Englot, B. (2018). LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 4758–4765.
- Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., and Rus, D. (2020). LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping. \_eprint: 2007.00258.
- Srinara, S., Chiu, Y.-T., Chen, J.-A., Chiang, K.-W., Tsai, M.-L., and El-Sheimy, N. (2023). Strategy on High-

Definition Point Cloud Map Creation for Autonomous Driving in Highway Environments. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-1/W2-2023:849–854.

- Umeyama, S. (1991). Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380.
- Vizzo, I., Guadagnino, T., Mersch, B., Wiesmann, L., Behley, J., and Stachniss, C. (2023). KISS-ICP: In Defense of Point-to-Point ICP Simple, Accurate, and Robust Registration If Done the Right Way. *IEEE Robotics and Automation Letters*, 8(2):1029–1036. Publisher: Institute of Electrical and Electronics Engineers (IEEE).
- Wilkinson, B., Mohamed, A., and Dewitt, B. (2015). Dual-Antenna Terrestrial Laser Scanner Georeferencing Using Auxiliary Photogrammetric Observations. *Remote Sensing*, 7(9):11621–11638.
- Wilkinson, B. E., Mohamed, A. H., Dewitt, B. A., and Seedahmed, G. H. (2010). A Novel Approach to Terrestrial Lidar Georeferencing. *Photogrammetric En*gineering & Remote Sensing, 76(6):683–690.
- Xu, W., Cai, Y., He, D., Lin, J., and Zhang, F. (2021). FAST-LIO2: Fast Direct LiDAR-inertial Odometry. \_eprint: 2107.06829.
- Xu, W. and Zhang, F. (2021). FAST-LIO: A Fast, Robust LiDAR-inertial Odometry Package by Tightly-Coupled Iterated Kalman Filter. \_eprint: 2010.08196.
- Zhang, J. and Singh, S. (2014). LOAM : Lidar Odometry and Mapping in real-time. *Robotics: Science and Systems Conference (RSS)*, pages 109–111.