Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

Bonn-Aachen
International Center for
Information Technology

Fraunhofer
IAIS

R&D Project

Registering and visualizing point cloud data with existing 3D CityGML Models

Alan Orlando Gomez Torres

Submitted to Hochschule Bonn-Rhein-Sieg,
Department of Computer Science
in partial fullfilment of the requirements for the degree
of Master of Science in Autonomous Systems

Supervised by

Prof. Dr Paul G. Plöger
Dr. Stefan Rilling
Alex Mitrevski

September 2020

I, the undersigned below, declare that this work has not previously been submitted to this or any other university and that it is, unless otherwise stated, entirely my own work.

Date	Alan Orlando Gomez Torres

Abstract

- INSERT ABSTRACT

Acknowledgements

Thanks to ....

Contents

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Challenges and Difficulties . . . 1
1.3 Problem Statement . . . 2

2 State of the Art **5**
2.1 Point-to-point registration . . . 6
2.1.1 Global methods . . . 6
2.1.2 Local methods . . . 7
2.1.3 Hybrid methods . . . 8
2.2 Point-to-model registration . . . 8
2.2.1 Sampling methods . . . 8
2.2.2 Local methods . . . 9
2.3 Others . . . 10
2.4 Limitations of previous work . . . 10

3 Methodology **13**
3.1 Setup . . . 13
3.2 Experimental Design . . . 13

4 Solution **15**
4.1 Proposed algorithm . . . 15
4.2 Implementation details . . . 15

5 Evaluation **17**

6 Results **19**
6.1 Use case 1 . . . 19
6.2 Use case 2 . . . 19
6.3 Use case 3 . . . 19

7 Conclusions **21**
7.1 Contributions . . . 21
7.2 Lessons learned . . . 21
7.3 Future work . . . 21

Appendix A Design Details **23**

Appendix B Parameters **25**

References **27**

List of Figures

1.1 A-DRZ architecture system. 3
1.2 System diagram. 4

2.1 Classification of the State of the Art. 6

List of Tables

1

Introduction

1.1 Motivation

Every day people are exposed to have some kind of accident and the worst kind of accidents are those that put their lives at risk. Fortunately, for those kinds of accidents, we have rescue services like firefighters. But firefighters are humans and put their lives at risk trying to save other lives. According to the Deutscher Feuerwehrverband (German Firefighters Association), the number of firefighters missions in Germany in 2017 ascends to more than 3 million, and the number of deaths in fires in 2016 was approximately 350 [1].

What if the firefighters are supported by rescue robots? With the use of rescue robots that explore the accident areas, we can help firefighters do their job more safely. Rescue robots can explore buildings to find entrances, exits, and other key areas that help the firefighters to rescue people in a faster and efficient way. But robots cannot find their exact position inside a building with a single GPS, because this kind of technology is unreliable under certain conditions (e.g. inside buildings).

In this work, we try to enable rescue robots to explore buildings with the use of LiDAR systems and 3D models of the buildings. With the use of LiDAR systems, we can obtain representations of the actual status of buildings at risk without endangering any lives. These kinds of representations are known as point clouds. With the use of CityGML models [14], we have access to the 3D representation of many buildings in Germany. Using modern registration techniques, one can find a match between the point cloud given by the LiDAR system and the 3D model of the building. The registration result can be then used to find the exact position of the robot with respect to the building. The registration result can also help to locate relevant parts of the building like windows, doors, stairs, or corridors. This information can be presented to the firefighters in a way that enables them to elaborate on a rescue plan, with the only objective of supporting firefighters in their mission to rescue lives and safeguard theirs.

1.2 Challenges and Difficulties

Most of the registration algorithms are focused on working for two point clouds. Just a few registration algorithms address the registration of 3D models with point clouds. And between those registration algorithms, only a couple are created to work with CityGML Models, as described in chapter 2.

The registration process is normally done manually, due to the fact that it is a simple task for a human since we all learn to match patterns when we are just kids. The automation of registration is complicated because for a computer it is difficult to find features and match them between two different types of 3D data.

An additional challenge is the data itself, the point clouds available are partial and they do not represent the whole 3D model, normally they just capture part of the walls of the buildings and the surface outside the buildings. And vice versa, the 3D models do not have information of the surface outside the buildings. The only information available in both, the 3D models and the point cloud, are some of the walls of the buildings.

At the time this work was started, there was no GPS information of the point clouds used in the registration, just of the CityGML Model. Therefore, it was not possible to perform a coarse registration with GPS information and the coarse registration need to be done by other means. Moreover, there was just one test set available for the evaluation of the solution. All this together made deciding on a specific approach difficult.

1.3 Problem Statement

This work intends to implement an automated registration method for a point cloud with a CityGML model. This automated registration method will then be used as part of the A-DRZ (Aufbau des Deutschen Rettungsrobotik-Zentrums) project of the Fraunhofer IAIS. Figure 1.1 shows the system architecture of the A-DRZ project. The final objective of the automated registration in the A-DRZ project is to be able to look inside the CityGML models, in order to find the location and status of relevant parts of the buildings (e.g. doors, windows, stairs, and corridors) that can be used for rescue task.

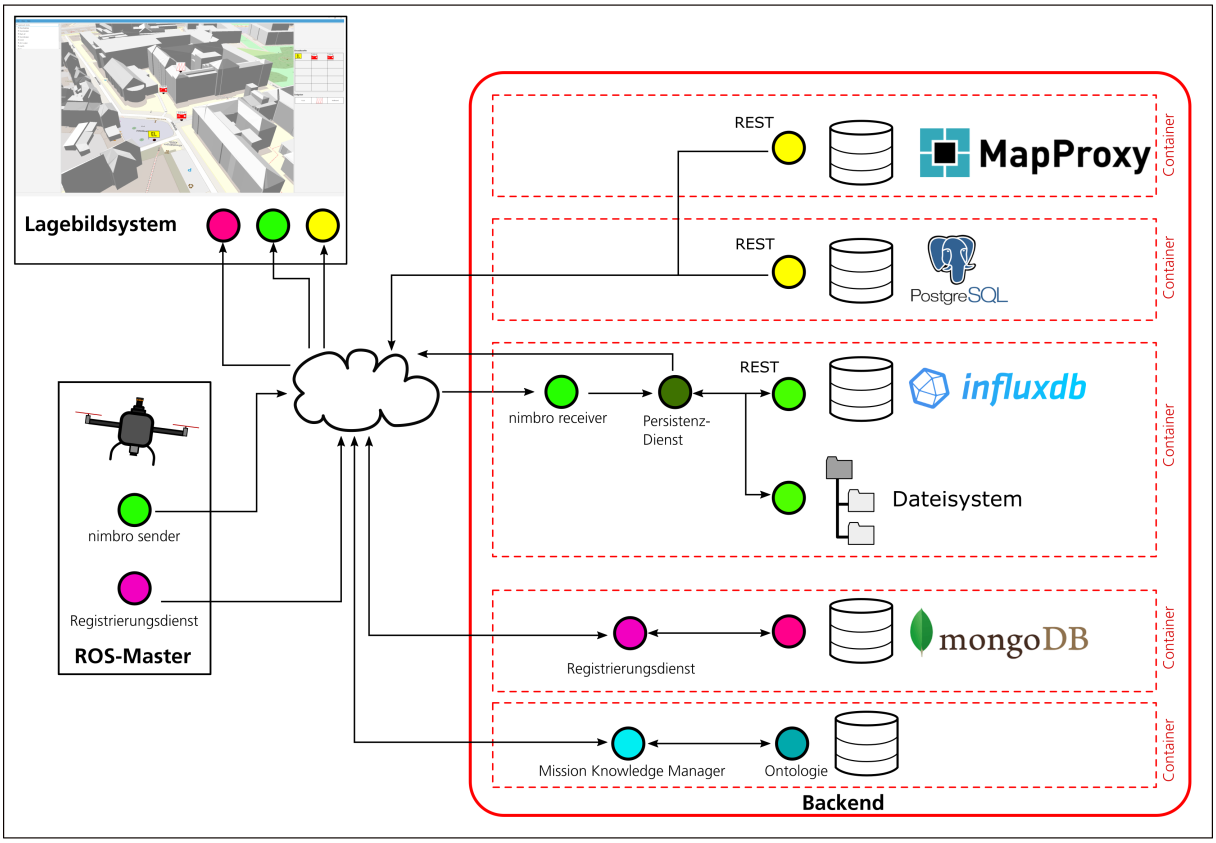

Figure 1.1: A-DRZ architecture system.

The current automated approaches are mainly focused on point-to-point registration. Therefore, the exploration of point-to-model automated registration methods is necessary to implement a solution that aims to work in real-time. Moreover, there is no automated process implemented in A-DRZ project yet. However, as an assistance system, the A-DRZ project should avoid human interaction for performing the registration.

Figure 1.2 shows the general workflow of the needed method. The input of the method is a point cloud together with the corresponding CityGML model of a building (the process to obtain this is outside of the scope of this work). The provided input is then preprocessed and used for the registration algorithm. The output of the registration algorithm is a transformation that maps the point cloud to the CityGML model. Finally, the output transformation and the input data are used for visualization of the point cloud on the CityGML model.

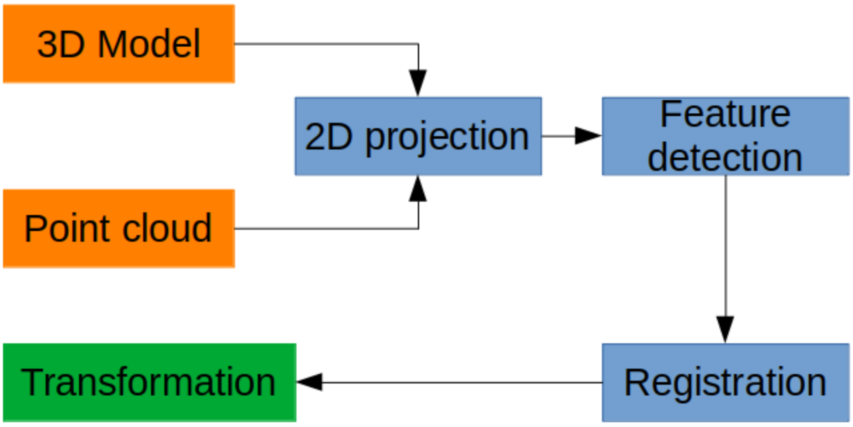

Figure 1.2: System diagram.

Firstly, the implementation will be tested individually. Secondly, the implementation will be tested with its integration on a visualization software of the Fraunhofer IAIS. This visualization software is a simple 3D rendering of CityGML models limited to a certain geographical area. Thirdly, the implementation will be tested with its integration on the A-DRZ project.

2

State of the Art

Point cloud registration algorithms are normally classified into two major categories: global (coarse) registration and local (fine) registration [25, 16]. Coarse registration algorithms attempt to find a transformation that approximately maps the source point cloud to the target point cloud. Fine registration algorithms attempt to refine an initial transformation to map the source point cloud to the target point cloud. There are some approaches that combine both of these methods, they first apply a global method and then refine the results with the help of a local method. We can classify these approaches into a 3rd category of hybrid methods.

For the purpose of this work we can classify point cloud registration algorithms into two major categories: point-to-point and point-to-model. Point-to-point registration algorithms attempt to align two point clouds, while point-to-model registration algorithms attempt to align a point cloud with a 3D model.

The approaches that do not work with point cloud data or 3D models, but are still related to the registration problem, are classified into the category of others.

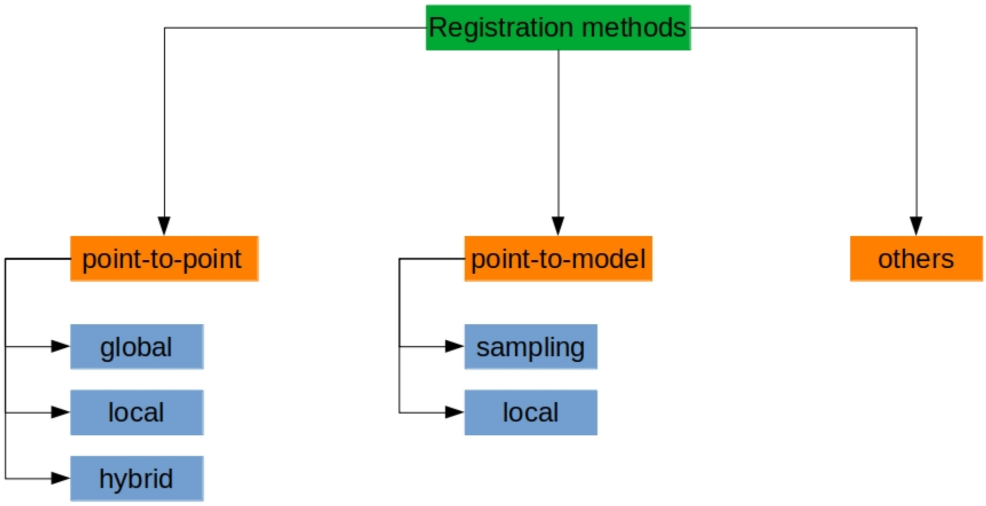

Figure 2.1: Classification of the State of the Art.

Figure 2.1 shows an overview of the papers reviewed.

2.1 Point-to-point registration

2.1.1 Global methods

RANSAC [9] can also be used to perform the registration task but is time-consuming due to the randomness of the RANSAC algorithm. Therefore, Pankaj et al. [22] present a guided sampling improvement to RANSAC, to reduce the registration time, sorting the correspondences obtained from a previous 3D feature matching according to its quality.

Following the same line, S. Quan and J. Yang [25] propose a method that helps the RANSAC algorithm sampling the best point correspondences. Their approach first detects keypoints using Harris3D, then it matches the keypoints by using the Local Voxelized Structure (LoVS) descriptor. After that, the set of correspondences is reduced using the Nearest Neighbor Similarity Ration (NNSR) and the compatibility score of the remaining correspondences with the rigidity and DSP constraint. Finally, the correspondences with the best compatibility scores are using to generate transformations and the best one (the one with the maximum number of inliers) is the final result. Unfortunately, the RANSAC-based approaches are sensitive to outliers.

Another less common approach for coarse registration is proposed by Sakakubara et al. [26]. This coarse registration approach uses Mixed Integer Linear Programming to find pairs of corresponding points that are then used to find a coarse transformation that aligns two different point clouds. This approach finds a global optimal result without using the values of the invariant features, it adjusts the error tolerance depending on the accuracy of the given data and it finds the best correspondences between the points. Nevertheless, Mixed Integer Linear Programming problems are NP-hard. Therefore, this approach is excessively time-consuming.

Others have proposed the use of neural networks to attempt to solve the global registration problem. This is the case of Wang and Solomon [29]. They propose a learning-based method called Deep Closest Point (DCP). The approach proposed first uses PointNet [23] or DGCNN [30] to embed the point clouds. Then, it uses an attention-based module to encode the output of the first part. Finally, it estimate the transformation using a differentiable SVD layer.

Sarode et. al [27] present a registration framework that also use the PointNet to extract feature from the point clouds. The PointNet features are then aligned using another network. The process is then repeated a given number of times to improve the registration result.

Ding and Feng [7] present another deep learning method called DeepMapping, whose main structure is constituted of two deep neural networks. The first network is a localization network, whose output is the pose estimation for the input point clouds. And the second network is a map network that estimates the occupancy status of the input locations to compute a loss function that reveals the registration quality.

More recently, Huang et al. [15] propose a learning method that can be trained in a semi-supervised or unsupervised way. The proposed framework is composed of an encoder, that extracts features from the input point clouds, and a decoder that decodes the extracted features to compute a projection error that serves as input for an algorithm that estimates a transformation increment. This increment is then employed to update the parameters of the transformation and transform one of the point clouds. The whole process is repeated until the loss computed is less than some desired value.

2.1.2 Local methods

Besl and N. D. McKay [3] propose the Iterative Closest Point (ICP) algorithm to perform registration between point sets, line segments, implicit curves, parametric curves, triangles, implicit surfaces, and parametric surfaces. ICP iteratively finds and updates correspondences between the desired sets based on an initial approximation to the solution and the spatial distance.

According to Segal et al. [28], the ICP algorithm can be summarized in two main steps: the computation of correspondences between the two scans and the computation of a transformation that minimizes the distance between corresponding points. They present a generalized version of the ICP algorithm by attaching a probabilistic model to the second step. This generalization makes the algorithm more robust to outliers without modifying the simplicity. However, it does not improve the speed of the original ICP.

ICP and its variants are still the standard algorithms for point cloud registration because of their simplicity.

2.1.3 Hybrid methods

The results of the global methods can be refined with the use of a local method, most of the time a variation of the popular ICP. In this category, some of these methods can be found.

Go-ICP is a branch-and-bound (BnB) based approach proposed by Yang et al. [32] to address the susceptibility of ICP to local minima. The main idea of Go-ICP is that BnB and ICP work iteratively until a global minimum is reached: BnB finds a solution and this is refined by ICP. Go-ICP guarantees global optimality regardless of the initialization and it is recommended for scenarios where real-time performance is not critical.

The approach proposed by Jun Lu et al. [19] obtains a consistent four-point set by using Super4PCS [21], then it creates a neighborhood ball with each point as the center and obtains the overlapping regions as the intersections of these neighborhood balls. Finally, ICP is applied to the overlapping regions. The main drawback of this method is that there should exist overlapping regions in order to perform the registration task.

Because of the increasing success of Deep Learning in 2D applications, there have been attempts to use deep neural networks to solve the complete point cloud registration problem (global and local registration). DeepICP [20] is the first end-to-end learning-based point cloud registration framework. DeepICP makes use of other networks to build a complete framework. The first step of DeepICP is the Deep Feature Extraction (PointNet++ [24]), which extracts features from the point clouds. In the second step, the Pont Weighting (3DFeatNet [33]) selects the relevant points. In the third step, the selected points of the target point cloud are sampled using the initial transformations. In the fourth step, Deep Feature Embedding is used to obtain more specific descriptors of the points. In the final step, the corresponding pair of points are generated with a 3D Convolutional Neural Network to compute an improvement of the initial transformation.

Another approaches does not use a neural network to solve the complete point cloud registration problem but a part of it. LORAX is an algorithm proposed in [8] that selects some sets of points called super-points and uses a low-dimensional descriptor generated by an autoencoder to describe the structure of each super-point. The sets of superpoints of both pointclouds are then matched using the euclidean distance between the descriptors. Finally, a coarse registration is computed using a RANSAC procedure, followed by a fine registration using ICP. The main contribution of LORAX is the use of an autoencoder to extract features of 3D point clouds. Nevertheless, the implementation is not optimized for real-time performance and it is not proved to be fast for big point clouds.

2.2 Point-to-model registration

2.2.1 Sampling methods

In order to register a 3D model with a point cloud, the simpliest idea is to sample points from the 3D model to generate a second point cloud and then perform one of the point-to-point registration methods.

Kim et al. [16] register a 3D CAD model of a construction site with a point cloud by sampling points from the 3D model and then applying a point-to-point registration algorithm. The registration consists of three main steps. First, the point clouds are re-sampled to the same resolution. Second, PCA is used to identify the principal axes of the data to obtain a coarse registration based on these axes. Third, fine registration is performed with the Levenberg-Marquardt iterative closest point (LM-ICP) algorithm [10].

Kim et al. [17] is an extension of their previous work [16] In this approach, they just add a noise filter step to the pre-processing of the point cloud in order to improve the registration results. The rest of the steps remain the same. However, using PCA for coarse registration will not assure a good performance when the form of the building is close to a cube.

2.2.2 Local methods

Other approaches work directly with the information provided in the 3D models to perform the registration task.

Li and Song [18] propose two extensions to the most popular local registration method, the ICP algorithm, to be able to use it with an STL-file (3D model) and a point cloud. The first of these extensions is called ICP-STL, and its main modification consists of finding the corresponding triangle mesh of the STL file of the points in the point cloud, and then the closest point in the triangle mesh to each point of the point cloud. The rest of the ICP algorithm remains the same. The second modification is called ICP-DAF (ICP dynamic adjustment factor), and it aims to reduce the time of the ICP-STL method. ICP-DAF adds a dynamic adjustment factor to adjust the transformation parameters dinamically. The disadvantage of this method is the same disadvantage present in the ICP algorithm, to perform well it requires a good initial approximation of the solution.

In the specific case of CityGML, Goebbels et al. [11] propose an extension of the well-known ICP to perform point-to-model registration based on the point-to-plane registration. This approach only considers points on the perimeter of the buildings. The main contribution of this approach is the speed. Point-to-model ICP is faster than the point-to-plane ICP. Nevertheless, CityGML models used do not contain terrain information. Therefore, the ground plane is the lowest point of the building and some walls might intersect with the real terrain, increasing the error of the algorithm. An additional disadvantage of this method is that it expects the point cloud to be coarsely register with the model.

Another approach by Goebbels et al. [12] explores the possibility to find features that help to perform the registration. The approach first rotates the point cloud to align the walls to the z-axis. Then it projects all the points onto the xy-plane to obtain a 2D image of the point cloud viewed from above. After that, it detects the corners or lines in the 2D image, and having the corners or lines of the CityGML model it applies a Mixed Integer Linear Program to perform the registration. Nonetheless, the point cloud has to be already coarsely registered in order to obtain a successful and fast result.

An extension of the work in [12] has been made by Goebbels et al. [13], where the steps to perform the registration are the same. However, in the last step, a Linear Program is used to fine-tune the coarse result. Unfortunately, the results of this method depend on the value of an error bound, which affects the generalization of the algorithm. Moreover, it has the same disadvantage as the previous approach: the point cloud has to be already coarsely registered in order to obtain a successful and fast result.

2.3 Others

There are other approaches that attempt to solve different matching problems with or without correspondences. These approaches cannot be classified in the previous categories because their input data is not composed of two point clouds or one point cloud and a 3D model. Some of these approaches are collected into this category.

Breuel [4] proposes the use of matchlist-based BnB techniques to solve geometric matching problems without correspondences. For example, point and line segment matching in 2D and 3D.

Bazin et. al [2] present an approach that combines Linear Programming and BnB procedures to achieve global optimality. The proposed algorithm receives two different sets of features extracted from two images and outputs the correspondences between the features verifying the appearance similarity and geometric constraints. In the end, it not only computes the feature correspondences but also computes an estimation of a geometric transformation between the features and identifies the inliers/outliers.

Brown et. al [5] extend the idea of the two previous approaches and present a 2D-3D registration method that does not need correspondences, by using a BnB algorithm. They also propose a deterministic annealing algorithm to reduce computation time. The proposed method works with points or lines, but not with a combination of both. Therefore, Brown et. al [6] propose a framework to solve the 2D-3D registration problem without feature correspondences that is able to use points, lines, or a combination of both.

Windheuser et al. [31] propose the use of Integer Linear Programming to find correspondences between non-rigid 3D shapes. They use an Integer Linear Program to minimize the elastic thin-shell energy requires to deform one shape into the other one. Their method can be improved by a relaxation of the Integer Linear Program and by including feature descriptors.

2.4 Limitations of previous work

ICP and its variants provide simple and easily-implemented iterative methods, but these algorithms can converge to false local optima [29]. Furthermore, they do not scale well with the number of points and they are not differentiable, therefore they cannot be integrated into end-to-end deep learning pipelines [27]. And the principal disadvantage of these methods is that they require a good initial approximation to the solution to perform well.

RANSAC and its variations cannot guarantee any global optimality in their final results [2]. According to [25], RANSAC has two main issues. The first one is its computational complexity, which makes it time-consuming. The second one is its randomness during sampling that can lead to inaccurate results. These two disadvantages worsen with the number of outliers.

The globally optimal methods such as Linear programming and BnB based methods are very time-consuming. Therefore, their application in real-time tasks is limited [27].

In the last years, there has been a boom in the use of neural networks and deep neural networks to solve 2D problems. Therefore, their use in 3D tasks has been explored too. Unfortunately, methods based on deep learning generally suffer from issues regarding the generalization ability and the requirement of massive training data [25].

In general, one thing that can be noticed is that none of this methods offers a direct solution for the coarse registration of a model with a point cloud. The only solution available for this is to sample a point cloud from the model and apply a global method for two point clouds. Unfortunately, this solution does not leverage the complete information contained in the model.

3 Methodology

How you are planning to test/compare/evaluate your research. Criteria used.

3.1 Setup

- The data used to evaluate the approach was collected and provided by different Universities (The University of Bonn, Technical University of Darmstadt).

- Unfortunately, there is no ground truth for the data available. Therefore, the results are evaluated manually by a person who decides whether the registration was successful or not.

- The input to the developed system is a CityGML model together with its corresponding point cloud.
- The output is the visualization of the CityGML model aligned with its corresponding point cloud.

3.2 Experimental Design

4

Solution

Due to the different sources of data and the time by which this was provided, we propose two similar but different solutions to solve this problem. The core part of the proposed solutions is a Linear Programming method and they differ in the preprocessing of the data.

The solution to the problem is to implement the algorithm presented by [12]

- Due to poor data availability, a (deep) learning method is unthinkable to solve the problem: there is not enough data to train a model.

4.1 Proposed algorithm

- The main steps of the algorithm are listed as follows:
- Project the point cloud into the XY plane
- Project the corners of the CityGML Model onto the XY plane
- Identify the corners in the projection of the point cloud onto the XY-plane
- Use Linear Programming to find a transformation that best aligns both projections onto the XY-plane

4.2 Implementation details

- Math - Code

5 Evaluation

Implementation and measurements.

6

Results

6.1 Use case 1

Describe results and analyse them

6.2 Use case 2

6.3 Use case 3

7
Conclusions

7.1 Contributions

- The solution presented in this work can be used and improved for the registration of a 3D Model with a point cloud using different kinds of features

7.2 Lessons learned

- How to make a research in a specific field.
- How to work in a project that already has many specifications defined.
- How to work with third-party libraries in order to ease the implementation of a project.
- How to adapt yourself to the changes in the requirements of a project.

7.3 Future work

- There are two possible options to improve the performance of the system (reduce the time of registration)
- One can work with 3D features and find directly a 3D transformation that aligns the CityGML-Model and the point cloud.
- a Branch-and-Bounding [2, 4, 5, 6] approach could be applied to find a transformation that aligns the projections of the CityGML model and the point cloud onto the XY-plane or to find such a transformation directly with 3D features such as lines or corners.

A

Design Details

Your first appendix

B

Parameters

Your second chapter appendix

References

[1] Deutscher Feuerwehrverband: Feuerwehr-Statistik. URL `https://www.feuerwehrverband.de/presse/statistik/`.

[2] Jean Charles Bazin, Hongdong Li, In So Kweon, Cédric Demonceaux, Pascal Vasseur, and Katsushi Ikeuchi. A branch-and-bound approach to correspondence and grouping problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1565–1576, 2013. ISSN 01628828. doi: 10.1109/TPAMI.2012.264.

[3] P J Besl and N D McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, feb 1992. ISSN 1939-3539. doi: 10.1109/34.121791.

[4] Thomas M. Breuel. Implementation techniques for geometric branch-and-bound matching methods. *Computer Vision and Image Understanding*, 90(3):258–294, jun 2003. ISSN 10773142. doi: 10.1016/S1077-3142(03)00026-2.

[5] Mark Brown, David Windridge, and Jean Yves Guillemaut. Globally optimal 2D-3D registration from points or lines without correspondences. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2015 Inter, pages 2111–2119, 2015. ISBN 9781467383912. doi: 10.1109/ICCV.2015.244.

[6] Mark Brown, David Windridge, and Jean Yves Guillemaut. A family of globally optimal branch-and-bound algorithms for 2D–3D correspondence-free registration. *Pattern Recognition*, 93:36–54, sep 2019. ISSN 00313203. doi: 10.1016/j.patcog.2019.04.002. URL `https://doi.org/10.1016/j.patcog.2019.04.002`.

[7] Li Ding and Chen Feng. DeepMapping: Unsupervised Map Estimation From Multiple Point Clouds. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[8] Gil Elbaz, Tamar Avraham, and Anath Fischer. 3D Point Cloud Registration for Localization Using a Deep Neural Network Auto-Encoder. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[9] Martin A Fischler and Robert C Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, 1981. ISSN 0001-0782. doi: 10.1145/358669.358692. URL `https://doi.org/10.1145/358669.358692`.

[10] Andrew W Fitzgibbon. Robust registration of 2D and 3D point sets. *Image and Vision Computing*, 21(13):1145–1153, 2003. ISSN 0262-8856. doi: https://doi.org/10.1016/j.imavis.2003.09.004. URL `http://www.sciencedirect.com/science/article/pii/S0262885603001835`.

[11] S Goebbels, R Pohle-Fröhlich, and P Pricken. Iterative Closest Point Algorithm for Accurate Registration of Coarsely Registered Point Clouds with CityGML Models. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 201–208, 2019.

[12] Steffen Goebbels and Regina Pohle-Froehlich. Line-based Registration of Photogrammetric Point Clouds with 3D City Models by Means of Mixed Integer Linear Programming. In *Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)*, 2018.

[13] Steffen Goebbels, Regina Pohle-Fröhlich, and Philipp Kant. A Linear Program for Matching Photogrammetric Point Clouds with CityGML Building Models. In Natalia Kliewer, Jan Fabian Ehmke, and Ralf Borndörfer, editors, *Operations Research Proceedings 2017*, pages 129–134, Cham, 2018. Springer International Publishing. ISBN 978-3-319-89920-6.

[14] Gerhard Gröger, Thomas H Kolbe, Claus Nagel, and Karl-Heinz Häfele. *OGC City Geography Markup Language (CityGML) Encoding Standard.* Open Geospatial Consortium, 2.0.0 edition, 2012.

[15] Xiaoshui Huang, Guofeng Mei, and Jian Zhang. Feature-metric Registration: A Fast Semi-supervised Approach for Robust Point Cloud Registration without Correspondences. *arXiv preprint arXiv:2005.01014*, 2020.

[16] Changmin Kim, Joohyuk Lee, Minwoo Cho, and Changwan Kim. Fully automated registration of 3D CAD model with point cloud from construction site. In *28th International Symposium on Automation and Robotics in Construction, Seoul, Korea*, pages 917–922, 2011.

[17] Changmin Kim, Hyojoo Son, and Changwan Kim. Fully automated registration of 3D data to a 3D CAD model for project progress monitoring. *Automation in Construction*, 35:587–594, 2013. ISSN 0926-5805. doi: https://doi.org/10.1016/j.autcon.2013.01.005. URL `http://www.sciencedirect.com/science/article/pii/S0926580513000071`.

[18] Weimin Li and Pengfei Song. A modified ICP algorithm based on dynamic adjustment factor for registration of point cloud and CAD model. *Pattern Recognition Letters*, 65:88–94, 2015. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2015.07.019. URL `http://www.sciencedirect.com/science/article/pii/S0167865515002287`.

[19] J Lu, W Wang, H Shao, and L Su. Point Cloud Registration Algorithm Fusing of Super 4PCS and ICP Based on the Key Points. In *2019 Chinese Control Conference (CCC)*, pages 4439–4444, 2019. doi: 10.23919/ChiCC.2019.8866059.

[20] Weixin Lu, Guowei Wan, Yao Zhou, Xiangyu Fu, Pengfei Yuan, and Shiyu Song. DeepICP: An End-to-End Deep Neural Network for 3D Point Cloud Registration. *CoRR*, abs/1905.0, 2019. URL `http://arxiv.org/abs/1905.04153`.

[21] Nicolas Mellado, Dror Aiger, and Niloy J Mitra. SUPER 4PCS Fast Global Pointcloud Registration via Smart Indexing. *Computer Graphics Forum*, 33(5):205–215, 2014. doi: 10.1111/cgf.12446.

[22] Dhanya S. Pankaj and Rama Rao Nidamanuri. A robust estimation technique for 3D point cloud registration. *Image Analysis and Stereology*, 35(1):15–28, 2015. ISSN 18545165. doi: 10.5566/ias.1378.

[23] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. Technical report, 2017.

[24] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In I Guyon, U V Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, and R Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5099–5108. Curran Associates, Inc., 2017. URL `http://papers.nips.cc/paper/7095-pointnet-deep-hierarchical-feature-learning-on-point-sets-in-a-metric-space.pdf`.

[25] S Quan and J Yang. Compatibility-Guided Sampling Consensus for 3-D Point Cloud Registration. *IEEE Transactions on Geoscience and Remote Sensing (2020)*, pages 1–13, 2020. ISSN 1558-0644. doi: 10.1109/TGRS.2020.2982221.

[26] Shizu Sakakubara, Yuusuke Kounoike, Yuji Shinano, and Ikuko Shimizu. Automatic range image registration using mixed integer linear programming. In *Asian Conference on Computer Vision*, pages 424–434. Springer, 2007.

[27] Vinit Sarode, Xueqian Li, Hunter Goforth, Yasuhiro Aoki, Animesh Dhagat, Rangaprasad Arun Srivatsan, Simon Lucey, and Howie Choset. One Framework to Register Them All: PointNet Encoding for Point Cloud Alignment. *arXiv preprint arXiv:1912.05766*, 2019.

[28] Aleksandr V Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-ICP. *Robotics: science and systems*, 2:435, 2009.

[29] Yue Wang and Justin M Solomon. Deep Closest Point: Learning Representations for Point Cloud Registration. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[30] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics*, 38(5):Article 146, 2019. ISSN 15577368.

[31] Thomas Windheuser, Ulrich Schlickwei, Frank R Schimdt, and Daniel Cremers. Large-Scale Integer Linear Programming for Orientation Preserving 3D Shape Matching. *Computer Graphics Forum*, 30 (5):1471–1480, 2011. doi: 10.1111/j.1467-8659.2011.02021.x. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2011.02021.x`.

[32] J Yang, H Li, D Campbell, and Y Jia. Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2241–2254, nov 2016. ISSN 1939-3539. doi: 10.1109/TPAMI.2015.2513405.

[33] Zi Jian Yew and Gim Hee Lee. 3DFeat-Net: Weakly Supervised Local 3D Features for Point Cloud Registration. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *European Conference on Computer Vision (ECCV) 2018*, pages 630–646, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01267-0.