Localization in MR-based Indoor Navigation System using point cloud registration

Siddharth Singh Chauhan [1], T.V. Jayakumar [1], Deepak Mishra [1], Anandakumar M. Ramiya [2]

[1] Indian Institute of Space Science and Technology, Dept. of Avionics, Thiruvananthapuram, India - siddharthsinghchauhan342001@gmail.com, jayakumartv.23@res.iist.ac.in, deepak.mishra@iist.ac.in

[2] Indian Institute of Space Science and Technology, Dept. of Earth and Space Science, Thiruvananthapuram, India - ramiya@iist.ac.in

Keywords: Localization, Mixed Reality, Registration, Point Cloud

Abstract

This paper addresses the pervasive challenge of localization within indoor navigation systems. Many existing systems grapple with this issue, relying on marker-based methods, Bluetooth beacon localization, and image processing. However, these solutions often necessitate the placement of predetermined markers within the environment, presenting limitations such as marker deterioration over time, spatial constraints, and associated costs. In response, we propose a novel point cloud registration-based approach. This method involves the creation of a comprehensive global point cloud representing the entire indoor environment, effectively serving as a three-dimensional virtual map. Concurrently, a mixed-reality device is utilized to generate a local point cloud specific to the device's current location within the building. By aligning and registering the local cloud with the global counterpart, the device's precise location within the virtual map can be determined. This approach enables seamless navigation within indoor environments without the reliance on physical markers, offering a versatile and cost-effective solution to the challenge of localization in indoor navigation systems. Finally, an error analysis was also done to test localization accuracy for different illumination conditions, demonstrating the effectiveness of this method.

1. Introduction

Today, a greater proportion of people spend their time indoors, and the significance of navigational systems designed specifically for use within buildings is growing. In a variety of locations, like shopping malls, airports, hospitals, museums, university campuses, and factories such systems can be extremely helpful. Many indoor navigation systems are being developed and each one uses different methods for final implementation, but almost all systems use Augmented Reality(AR) as a Human-Machine Interface. They offer an exciting and innovative approach to indoor navigation that has the potential to revolutionize the way, we navigate complex indoor environments, hence a Mixed-reality-based device was chosen for this application that goes one step ahead not only it has the capability of overlaying instructions on the real-world but also various in-built sensors to sense its surroundings. To build any navigation system the first requirement is to find the initial position of the user, this paper tries a point-cloud-based approach to do the same. This study introduces a method for locating mixed-reality (MR) devices within indoor environments, by generating a global point cloud of the entire indoor space using a Terrestrial Laser Scanner just like a virtual map. For localization of the user, a local point cloud is created with depth and RGB cameras on the MR device, and registration techniques are used to align the local cloud with the global to determine the device's location in the virtual map. An Error analysis is done that validates the effectiveness of this approach, showcasing its potential to enhance navigation using MR devices beyond traditional human-machine interfaces.

2. Related Work

Many AR-based navigation systems are being developed, in (Huang et al., 2020) they developed an indoor positioning system called ARBIN, the system has an AR interface on the user level, localization is done using Bluetooth beacon-based waypoints, where constant Bluetooth Low Energy (BLE) messages are received and position is tracked. This approach involves placing BLE beacons throughout the indoor environment, which can be detected by a user's device to determine their location. In (Lövdahl, 2020) indoor navigation system was developed for the MagicLeap headset, the system uses $A*$ algorithm for pathfinding, and it uses spatial mesh created by magic leap itself to locate its position by using a world anchor placed in the digital twin of the environment. The AR interface created uses minimaps like in video games for the user to locate their position in real world. In (Rehman and Cao, 2017) the indoor navigation system where pre palced trackables with location information is palced in real world which can be overlayed on AR device to aid the navigation process. In (Koch et al., 2014) they developed a system that can locate the user based on the natural markers inside the building like the signage. In (Dong et al., 2019), ViNav was created, which is a more cost-effective method that creates a 3D model of the environment from photos and locates users depending on the Wifi-signal and image requested by the user.In most the cases for indoor navigation use some sort of artificial markers to get the initial location is prominent, though being effective there are some common problems with such approaches like wear and tear of markers, the initial cost of placing beacons and scanning the environments with headsets to place the anchors, one divergence from such methods is LORAX(Elbaz et al., 2017) a deep learning-based point cloud registration, to align two point clouds in huge geographical settings taken from the same initial point cloud. This paper tries to register two point-clouds taken from different sources. One will be the huge prescanned cloud of the building, the other a small local cloud generated by mixed reality device with its position as the origin. Proper registration of the two-point data will result in the position of the mixed reality device in the prescanned virtual map.

3. Point cloud localization

The indoor navigation system will have a pre-scanned virtual 3D map of the building. Point clouds are chosen for the same, as they are raw 3D data and can further be processed. A Terrestrial Laser Scanner was used to generate the pre-scanned cloud of two floors of an indoor environment shown in figure 1. Point clouds from such a device are more accurate and highly dense, it is used similarly as people use maps. The aim is to localize the HoloLens2 device in this virtual map. The HoloLens 2 incorporates both a traditional RGB camera and a Time-of-Flight (ToF) depth camera. The data streams originating from this device are collected and subsequently utilized for voxelization. This information undergoes processing via the Truncated Signed Distance Function (TSDF)(Curless and Levoy, 2023) technique to generate a three-dimensional (3D) scene. The resultant scene is then sampled to derive a point cloud representing the immediate surroundings of the device, as illustrated in Figure 2, with the device’s location serving as the origin. The next step involves computing a transformation matrix between two clouds such that the origin of the pre-scanned cloud gives the location of the HoloLens2 device. This is done by aligning the two clouds and the process is called registration. Henceforth, within the context of this paper, all point clouds generated by the HoloLens 2 device shall be denoted as ”local clouds,” while the term ”global cloud” shall exclusively refer to point clouds acquired by Terrestrial Laser Scanning (TLS) technology.

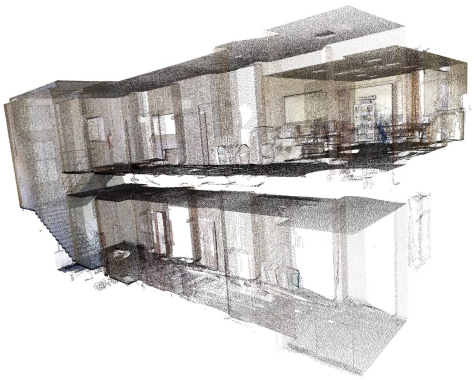

Figure 1. Pre-scanned TLS-based point cloud acting as known 3D map of the building

Figure 2. The HoloLens 2 local cloud is defined relative to the spatial coordinates of the device, with the location of the device serving as the origin.

4. Registration

Point cloud registration is the process of aligning two or more point clouds to share a common coordinate system. The goal of point cloud registration is to find the rigid transformation (translation and rotation) that best aligns the points in one cloud with those in another. There are many methods for the registration process, but the following methods are used here

4.1 ICP

The most basic and reliable form of registration is where an initial correspondence between the point cloud is given manually at the start. Iterative Closest point (ICP)(Besl and McKay, 1992) is the most widely used algorithm for point cloud registration, it requires initial correspondence for convergence. There are many variations to ICP (Rusinkiewicz and Levoy, 2001), but the basic vanilla algorithm is chosen. In adhering to the fundamental principles of the Iterative Closest Point (ICP) algorithm, a basic approach is employed wherein the algorithm iteratively minimizes the sum of squared distances between two point clouds. This entails selecting the nearest point in one cloud as a correspondence for each point in the other cloud, and iteratively refining these correspondences until convergence is achieved. To facilitate convergence, it is essential to establish initial correspondences effectively, a task typically addressed through computer vision techniques that primarily use feature extraction and matching.

4.2 Feature Extraction

To ensure real-time functionality in navigation systems, localization processes necessitate faster and simpler features, that offer reasonable reliability, particularly as only initial correspondences are required for the Iterative Closest Point (ICP) algorithm to converge. In this context, Fast Point Feature Histograms (FPFH)(Rusu et al., 2009) have been selected due to their expedited computation and robust geometric characteristics. FPFH features are known for their ability to incorporate local neighborhood information efficiently, thus enhancing their reliability and suitability for near real-time applications in navigation systems.

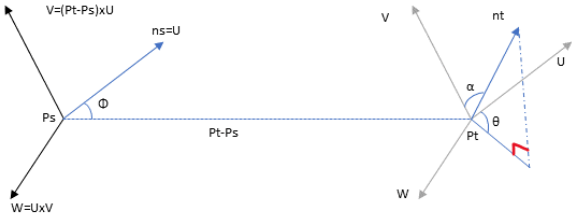

Figure 3. Creating the point features frame from two points for computing the PFH

The FPFH is a variant of Point Feature Histograms(PFH)(Rusu et al., 2008), where (α, ϕ, θ) is computed as shown in figure 3 between each query point (p_q) and its neighbours and binned called Simplified Point Feature Histograms$(SPFH)$.

In a subsequent stage, the ***k*** neighbours of each point are once again identified as shown in figure 4, and the neighbouring SPFH

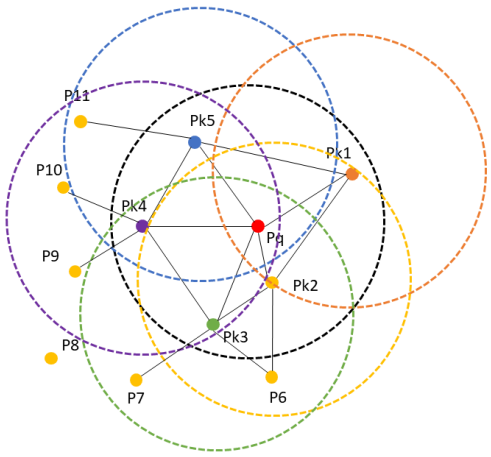

Figure 4. FPFH feature estimation for pq, computing PFH for each point in neighbor- hood of query point and using it weight PFH of query point

values are then utilised to weight the final histogram of p_q (referred to as the FPFH) as follows:

$$FPFH(p_q) = SPFH(p_q) + \frac{1}{k}\sum_{i=1}^{k}\frac{1}{\omega_i}\cdot SPFH(p_i) \quad (1)$$

4.3 RANSAC with ICP

Feature extraction is conducted on each point of two clouds, using Fast Point Features, followed by the creation of histogram descriptors for matching purposes. Following the computation of features, a series of subsequent steps are undertaken for registration using the RANSAC algorithm (Fischler and Bolles, 1987). first subsets are randomly sampled from each cloud, and transformations are computed to map similar features. These transformations are then applied to all the points. Error is computed using a distance metric between the two clouds, typically Euclidean distance and inliers are computed using a threshold. The process continues for a fixed number of iterations, and the transformation matrix with most inliers is selected achieving optimal registration between the source and target point clouds. Consider two point-clouds P_s and P_t. We seek a rigid transform T that aligns P_s and P_t. Let $F(P_s) = \{F(p_s) : p_s \in P_s\}$ be the set of features(FPFH) for point cloud P_s, where $F(p_s)$ is some extracted feature for point p. Analogously, define $F(P_t) = \{F(p_t) : p_t \in P_t\}$ for point cloud P_t. To align these two, a generic optimization problem can be formulated as:

$$\epsilon(T) = \sum_{p_s \in P_s, p_t \in P_t} \| p_s - Tp_t \|^2 \quad (2)$$

Here, T is the transformation that must be estimated so as to minimise the sum of squared Euclidean distances between the corresponding points within both sets P_s and P_t. The algorithm is described as shown at 1.

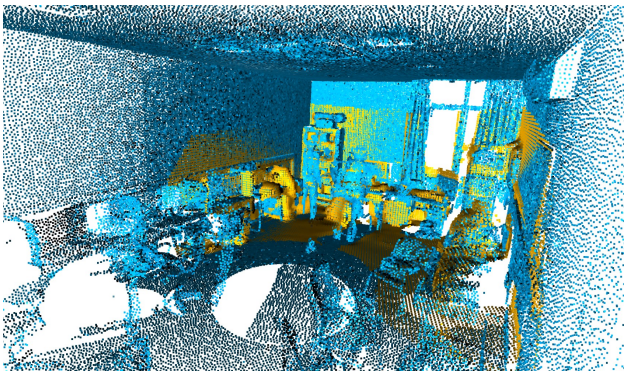

Figure 5. The final registered cloud from the RANSAC-based registration, yellow points from local cloud, and blue is the TLS-based map.

Algorithm 1: RANSAC-based Point Cloud Registration

Input : Source point cloud P_s, target point cloud P_t, maximum iterations N, sample size k, distance threshold d
Output: Transformation T that maps P_s to P_t

Compute features $F(P_s)$ and $F(P_t)$;
for $i = 1$ *to* N **do**
  $S \leftarrow$ random sample of $k \geq 3$ points from P_s;
  Find correspondences of these points in P_t by nearest neighbour search in $F(P_s)$ and $F(P_t)$;
  Estimate T from correspondences
  Apply T to P_t
  Search for inliers between P_s and transformed P_t by spatial nearest neighbour search, with max distance threshold given
  if *ratio of inliers is too low* **then**
    Skip the iteration
  end
  Reestimate T with the found inlier correspondences.
  Compute the error ϵ from equation 2 using estimated T and inliers.
  $\epsilon^(best)$ is the previous lowest error
  if $\epsilon < \epsilon^(best)$ **then**
    $T^{best} = T$
    $\epsilon^{best} = \epsilon$
  end
end
return T^{best}

5. Error Analysis

Automatic registration, facilitated by feature extraction, matching, and outlier rejection has been achieved. However, a critical inquiry arises regarding its accuracy when compared with manual registration, which is precise. Furthermore, there is a need to gauge the performance of this localization method across diverse lighting conditions. To address these inquiries systematically, an application-specific error computation method is proposed. This method will meticulously quantify the disparities between automated and manual registration outcomes and assess the robustness of the localization approach under varying lighting conditions. Through this comprehensive evaluation, insights into the effectiveness and reliability of automated registration techniques in practical scenarios will be discerned.

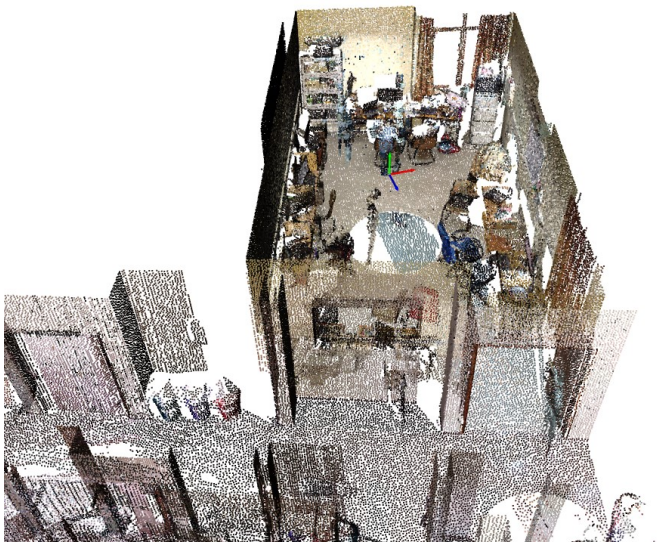

Figure 6. The origin is marked on the TLS cloud after computed transformations are applied, it gives location of the HoloLens2

5.1 Ground Truth Estimation

The first step is to compute the ground truth data which will contain a transformation matrix such that when applied to the global point cloud ($TLS - based$), changing the reference and its origin and three axes give the location and orientation of the HoloLens 2 device. This is accomplished in three steps. The first is to manually register TLS and the HoloLens2 clouds using point selection giving us the transformation matrix, next step is to transform the TLS cloud and uniquely mark the origin and the axes using 4 points, one at the origin and the other three at the individual axis. This is done by marking the points with unique colors as shown in figure 7. The final step is to apply the inverse of manual transformation computed with registration to the TLS cloud so it gets back to the original coordinate system. Now we have TLS cloud with the exact location of the HoloLens2 device embedded in it with the original reference frame. This will be the ground truth for that particular local cloud, this process is repeated multiple times for different local clouds and corresponding global clouds are stored.

5.2 Error Calculation

The ground truth data was created manually by registering and marking the location and orientation of the device in a pre-scanned map as discussed. This map is used again to register with local clouds but using the automatic method, now we have the location and orientation of the device in the global cloud with ground truth marked as shown in figure 7, and the following four errors are calculated.

$$Distance = \| O_c - O_{gt} \| \tag{3}$$

$$\alpha = \arccos\left(\frac{\vec{X_c}.\vec{X_{gt}}}{\| X_c \| \| X_{gt} \|}\right) \tag{4}$$

$$\beta = \arccos\left(\frac{\vec{Yc}.\vec{Y_{gt}}}{\| Y_c \| \| Y_{gt} \|}\right) \tag{5}$$

Figure 7. Transformed Ground truth TLS cloud by RANSAC-based registration with the location of HoloLens2 marked and ground truth points marked for error analysis, shown in blue box. local cloud shown in figure 2 is used for registration

$$\gamma = \arccos\left(\frac{\vec{Z_c}.\vec{Z_{gt}}}{\| Z_c \| \| Z_{gt} \|}\right) \tag{6}$$

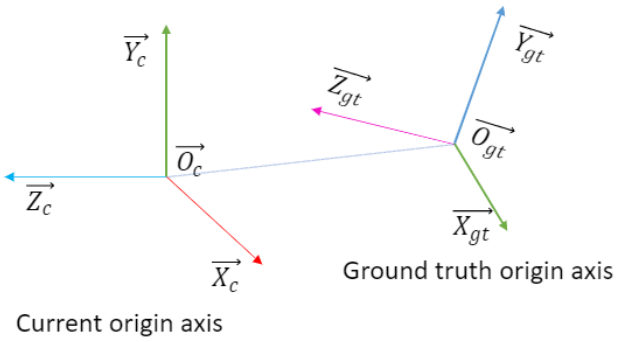

Figure 8. Error Computation between ground truth and current references

O_c stands for the origin of the current estimation and O_gt is the ground truth origin, location-error is simply the Euclidean distance between two points. The angles are calculated using the dot product between the axes of both the references as shown in figure 8. The following tables present error values associated with five distinct local cloud instances observed under two distinct lighting conditions, the distances are in meters and angles are in degrees

	Good Light			
	Distance (m)	**Alpha (°)**	**Beta (°)**	**Gamma (°)**
PCD1	0.16	0.09	0.09	0.03
PCD2	0.14	0.01157241	0.01	0.01
PCD3	0.31	0.09	0.08	0.06
PCD4	0.57	0.034	0.01	0.03
PCD5	0.19	0.03	0.03	0.02

Table 1. Table showing error values for different point clouds under good lighting conditions

The first three clouds include the geometries of the rooms only, which tend to have more unique features; hence, registration

	Low light			
	Distance (m)	**Alpha (°)**	**Beta (°)**	**Gamma (°)**
PCD1	0.35	0.10	0.11	0.01
PCD2	1.11	0.16	0.01	0.16
PCD3	3.92	0.11	0.04	0.11
PCD4	3.83	2.92	0.012	2.92
PCD5	0.32	0.04	0.05	0.04

Table 2. Table showing error values for different point clouds under poor lighting conditions

is so precise in this case, with errors in centimeters with only one exception of PCD3 in table 2, which is in low-light cases. The other two are point clouds of corridors; also, their size is small, hence there are less unique features, in low illumination the algorithm tend to perform poorly. This analysis showcases the fact that this approach to localization is a viable option for future navigation systems for indoor environments.

6. Conclusion

In conclusion, this study focuses on advancing localization in indoor environments through registration methods utilizing data captured by the Microsoft HoloLens 2. By harnessing the device's precise spatial mapping capabilities via depth sensing, we successfully generated a comprehensive point cloud representation of the surrounding environment employing the Truncated Signed Distance Function (TSDF) integration technique. Our primary achievement lies in the effective alignment of the HoloLens 2-generated cloud with data acquired from Terrestrial Laser Scanners (TLS), facilitated by an optimized Iterative Closest Point (ICP) algorithm augmented with a Robust Random Sample Consensus (RANSAC) based feature matching technique. This refinement significantly enhances the accuracy and efficiency of the registration process. Furthermore, this methodology exhibits promising results in localizing local clouds within the pre-scanned map of the environment. A straightforward yet robust error analysis method is also implemented that is capable of evaluating localization accuracy across varying lighting conditions. This study holds profound implications for a multitude of fields, including robotics, autonomous vehicles, and industrial automation. Despite constituting a modest contribution, the transformative potential of mixed reality applications is undeniable, promising significant advancements across diverse domains.

References

Besl, P. J., McKay, N. D., 1992. Method for registration of 3-d shapes. *Sensor fusion IV: control paradigms and data structures*, 1611, Spie, 586–606.

Curless, B., Levoy, M., 2023. *A Volumetric Method for Building Complex Models from Range Images*. 1 edn, Association for Computing Machinery, New York, NY, USA.

Dong, J., Noreikis, M., Xiao, Y., Ylä-Jääski, A., 2019. ViNav: A Vision-Based Indoor Navigation System for Smartphones. *IEEE Transactions on Mobile Computing*, 18(6), 1461-1475.

Elbaz, G., Avraham, T., Fischer, A., 2017. 3d point cloud registration for localization using a deep neural network auto-encoder. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2472–2481.

Fischler, M. A., Bolles, R. C., 1987. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. M. A. Fischler, O. Firschein (eds), *Readings in Computer Vision*, Morgan Kaufmann, San Francisco (CA), 726–740.

Huang, B.-C., Hsu, J., Chu, E. T.-H., Wu, H.-M., 2020. ARBIN: Augmented Reality Based Indoor Navigation System. *Sensors*, 20(20). https://www.mdpi.com/1424-8220/20/20/5890.

Koch, C., Neges, M., König, M., Abramovici, M., 2014. Natural markers for augmented reality based indoor navigation and facility maintenance. *Automation in Construction*, 48, 18–30. https://doi.org/10.1016/j.autcon.2014.08.009.

Lövdahl, N., 2020. Navigation and assistance using augmented reality. Master's thesis, Lund University.

Rehman, U., Cao, S., 2017. Augmented-Reality-Based Indoor Navigation: A Comparative Analysis of Handheld Devices Versus Google Glass. *IEEE Transactions on Human-Machine Systems*, 47(1), 140-151.

Rusinkiewicz, S., Levoy, M., 2001. Efficient variants of the icp algorithm. *Proceedings third international conference on 3-D digital imaging and modeling*, IEEE, 145–152.

Rusu, R. B., Blodow, N., Beetz, M., 2009. Fast point feature histograms (fpfh) for 3d registration. *2009 IEEE International Conference on Robotics and Automation*, 3212–3217.

Rusu, R. B., Blodow, N., Marton, Z. C., Beetz, M., 2008. Aligning point cloud views using persistent feature histograms. *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3384–3391.

7. Appendix

Some additional results from registration are shown

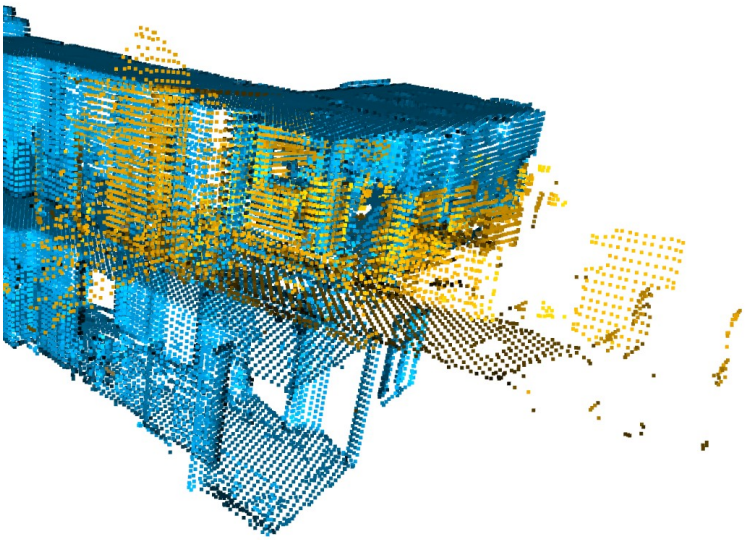

Figure 9. Registration result for another local cloud

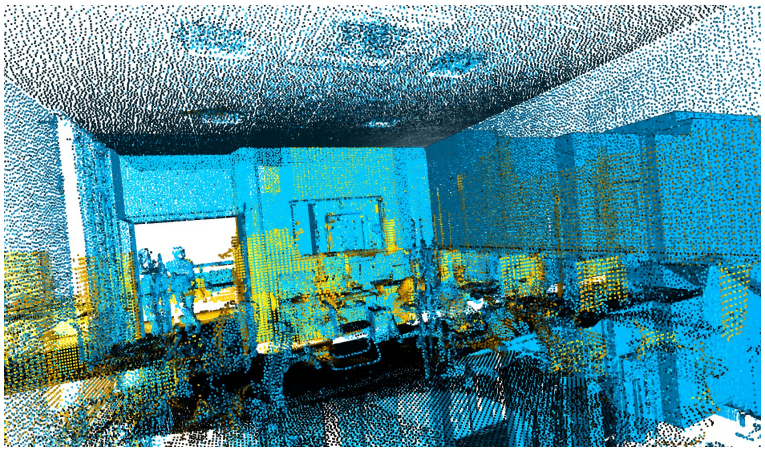

Figure 10. Registration result for another local cloud

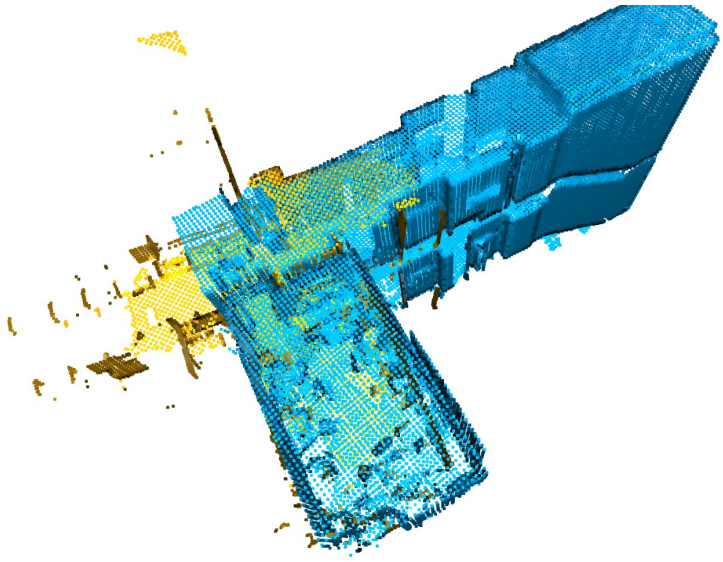

Figure 11. Registration result for another local cloud