Analysis of mixed reality cross-device localization algorithms based on point cloud registration

Alexander Osipov
Center for Technologies in Robotics and Mechatronics Components
Innopolis University
Innopolis, Russia
a.osipov@innopolis.university

Mikhail Ostanin
Center for Technologies in Robotics and Mechatronics Components
Innopolis University
Innopolis, Russia
m.ostanin@innopolis.ru

Alexandr Klimchik
Center for Technologies in Robotics and Mechatronics Components
Innopolis University
Innopolis, Russia
a.klimchik@innopolis.ru

***Abstract*—State of art approaches for localization and mapping is based on local features in images. Along with this, modern augmented and mixed reality devices allow building a mesh of the surrounding space. Using this mesh map, we can solve the problem of cross-device localization. This approach is independent of the type of feature descriptors and SLAM used onboard the AR / MR device. The mesh could be reduced to the points cloud, which takes only vertices. We propose the approach for co-localization of AR / MR devices using point clouds which do not depend on algorithms onboard the device. We analyzed various algorithms for registering point clouds and discuss of it limitation for the co-localization problem.**

***Index Terms*—Indoor collaborative localization, augmented and mixed reality devices, point cloud registration.**

I. Introduction

Localization and mapping are the main technical capabilities of modern mixed reality systems and robotics. This function allows connecting the real and digital world into a single reality. For people to collaborate in mixed reality or for people and robots to work together, they need to be localized in a single coordinate space. In addition, localization of devices is also required for a content position on a pre-built map.

Nowadays, devices solve the problems of localization and mapping on board. However, there is a trend towards cloud computing and moving content localization to the cloud. For example, there are systems Microsoft Azure Spatial Anchors [1], Niantic [2], Google Cloud Anchors [3], etc. These systems send key points and descriptors to the cloud for content localization.

Modern systems have several limitations of collaborative localization. Co-localization with devices using different features is impossible. [4] In addition, various systems use various simultaneous localization and mapping (SLAM) algorithms and have hardware acceleration for them. For example, various features can be used, such as SIFT [5] or SOSNet [6]. Moreover, existing devices will lag behind the new SLAM algorithms. This makes the systems impossible to co-localize with each other. In addition to the above, it is difficult or impossible for a human to perceive a map consisting of features. This imposes a restriction on the remote installation of content on the map.

There is a common functionality for constructing a mesh map of the real environment in mixed reality systems. The mesh map should geometrically represent the real environment. Respectively, on different systems, regardless of the algorithms, these maps will be similar to each other in the same place. However, the mesh itself has a lot of information and is difficult to transfer and save, so we propose to reduce the spatial map to a point cloud consisting only of the mesh vertices. As a result, we will get a sparse cloud of points that are easier to save and transfer. At the same time, this point cloud will be readable for human perception, which could help to use it for content location.

In this paper we analysis of point cloud registration methods with modifications. Our general goal is to develop an approach capable of solving the problem of indoor mixed reality cross-device localization. We collected a new dataset with pairs of reconstructed point clouds for one environment.

Our first stage is experimental efficiency comparison of four global registration methods on real point clouds of indoor environments to know which methods of registering point clouds are suitable for solving the problem of MR devices co-localization and to understand which parameters of the algorithm determine more high robustness and probability of point cloud registration success.

Our second stage is a more detailed analysis of the effectiveness for more perspective registration methods capable of solving the problem of collaborative localization for different parameters and methods modifications. As a modification, we added and used a local feature descriptor Weighted Height Image (WHI) [7] in addition to the default and often used one Fast Point Feature Histogram (FPFH) [8]. Our third stage is the efficiency analysis of the hybrid approach: feature correspondence-based methods + ICP which are proposed to solve the problem of co-localization of mixed reality devices.

In our final stage, we discuss the weak points of the investigation approach and obtained results, which can be useful for fine-tuning of approach for collaborative localization in real scenarios.

II. Related Work

In this section, we consider the fundamental foundations of the algorithms under study and their capabilities. In the registration problem, we are given two 3D point clouds $\mathcal{A} = \{\mathbf{a}_i\}_{i=1}^N$ source and $\mathcal{B} = \{\mathbf{b}_i\}_{i=1}^M$ target point clouds with $\mathbf{a}_i, \mathbf{b}_i \in \mathbb{R}^3$. Many algorithms exist to solve this problem. One of the most often used approaches for point cloud registration is Iterative Close Point (ICP) [9].

A. Standard-ICP

This algorithm solves the L_2-norm registration problem to estimate rigid motion such as rotation $\mathbf{R} \in SO(3)$ and translation $\mathbf{t} \in \mathbb{R}^3$ between source $\mathcal{A}$ and target $\mathcal{B}$ point clouds, which minimizes the objective L_2-error function:

$$E(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{N} \|\mathbf{R}\mathbf{a}_i + \mathbf{t} - \mathbf{b}_{k^*}\|^2, \tag{1}$$

where $\mathbf{a}_i$ is i point of source cloud and $\mathbf{b}_{k^*}$ is the closest point of target cloud to transformed $\mathbf{a}_i$-point. Defined L_2-error is non-convex, because there are non-convex constraints. Standard ICP algorithm solves this problem iteratively that alternate between estimating the transformation by (1), and finding closest-point correspondences. Although ICP characterized by its simplicity and effectiveness, it is guarantees convergence to a local minimum only [9]. Hence, ICP needs a good initial pose between point clouds for global convergence. Some variants of ICP [10], [11], [12], [13] have suggested to use robust cost functions to improve convergence, but they are still local methods and don't provide guarantee global convergence.

B. ICP-based point cloud registration methods

As a rule in real scenarios, point clouds from several devices have large relative displacement, and rotation error can be near 360 degrees for one of Euler angles. Therefore to solve the matching problem of point clouds, it is required to use global methods of registration that do not depend on good initialization.

1) Go-ICP: Globally Optimal ICP (Go-ICP) [14] is the point cloud registration method providing an optimal global solution without initial guess. To find global solution authors of Go-ICP method use BnB (Branch-and-Bound) [15] algorithm extending to search 3D motion $SE(3) = SO(3) \times \mathbb{R}^3$. Authors apply domain parameterization which compactly represents 3D rotation search space as a solid radius-π ball in $\mathbb{R}^3$ and for translation part domain is represented as a bounded cube $[-\xi, \xi]$ where ξ can be easily set. Also they bounds L_2-norm error function [14]. Domain parameterization and bounding functions allow to apply BnB search to a problem (1). To sum up, the Go-ICP method presents the integration of two main processes: global BnB search and the local ICP search that helps each other to reach the global minimum of the objective function.

2) Bayesian-ICP: While Go-ICP is deterministic approach to registration point clouds, there is also a probabilistic approach such as algorithm Bayesian-ICP [16]. Bayesian-ICP combines ideas from stochastic gradient descent-ICP (SGD-ICP) [17] and stochastic gradient Langevin dynamics (SGLD) [18] that allows estimating a pose distribution. For each iteration of SGD-ICP, small mini-batches $\mathcal{M}_k$ from the source cloud instead of the full point cloud are formed and associated with closest points in the target cloud as in Standart ICP to solve the optimization problem (1). SGLD idea is to add the right amount of noise to SGD optimization which allows converging to the posterior distribution. Therefore general SGD-ICP update rule is modified by adding Gaussian noise $\eta_k \sim \mathcal{N}(0, A\alpha)$ and prior $p(\vartheta)$ over the transformation parameters θ, so general update rule for Bayesian-ICP becomes following:

$$\vartheta_{k+1} = \vartheta k - \frac{\alpha}{2} A\Big(-\nabla \log p(\vartheta_k) + N\vec{g}(\vartheta_k, \mathcal{M}_k)\Big) + \eta_k, \tag{2}$$

where N is size of point cloud, and $\nabla \log p(\vartheta_k)$ is gradient for prior distribution, α is learning rate, $A \in \mathbb{R}^{6\times 6}$ acts as a pre-conditioner and $\vec{g}$ is gradient of objective function (1).

C. Correspondence-based point cloud registration methods

Correspondence-based registration method is another approach to solve registration problem providing a global solution. The most perspective algorithms of correspondences-based registration methods are Fast Global Registration (FGR) [19] and Teaser++ [20]. They use invariant to rotations and translations feature descriptors to build correspondences between point clouds. Feature descriptors continue to be improved (robustness to noise, to occlusion, to clutter and etc., compactness, descriptiveness), also including using neural networks to this day.

1) Feature descriptors: **Fast Point Feature Histograms.** Fast Point Feature Histograms [8] is a 33-dimensional local feature descriptor that describes the local geometry of space around a point in a 3D point cloud. This descriptor represent a simplified version of Point Feature Histograms (PFH), but it keeps the discriminative power of the PFH and can be calculated in milliseconds time [21], due to the computational complexity of the algorithm is $O(k)$ compare with $O(k^2)$ for PFH.

Weighted Height Image descriptor. Weighted Height Image descriptor [7] (WHI) is a compact 3D local feature descriptor for describing the 3D local shape in the point cloud. When FPFH is classified as an algorithm based on Rotation-Invariant Metrics (RIM), WHI feature descriptor is based on Local Reference Frame (LRF). LRF based descriptors have clear advantages compared with Rotation-Invariant Metrics. Firstly, descriptors estimate a rotation-invariant local frame (LRF), which is more repeatable and robust to occlusions and clutter. Secondly, using LRF simplifies the process of coding information, because rotation invariance is not needed consideration, and allows to save the original information about point cloud. In addition, WHI feature descriptor can have a maximum compactness - dimension 16, without much loss in precision.

2) Fast Global Registration: Fast Global Registration is correspondence-based registration method that consists of the following main submodules: Advanced Matching and transformation estimation [19]. Firstly, for each point in source $\mathcal{A}$ and target $\mathcal{B}$ point clouds feature points are extracted. Then $\mathbf{F}(\mathcal{B})$ is a set of source feature points, and $\mathbf{F}(\mathcal{A})$ is set of target feature points. Secondly, Advanced Matching algorithm is used to build correspondences by using feature points and also to prune partially wrong pairs of them. It consists of 3 steps. First step: build set $\mathcal{K}_I$ pairs points by computing nearest neighbors between feature points $\mathbf{F}(\mathcal{B})$ and $\mathbf{F}(\mathcal{A})$. Second and third steps: apply reciprocity test on $\mathcal{K}_I$ to get $\mathcal{K}_{II}$ and tuple test on $\mathcal{K}_{II}$ to get $\mathcal{K}_{III}$ set to prune correspondences [19]. And finally, FGR use building and pruning correspondences to find a transformation matrix $\mathbf{T}$ that aligns two points cloud, it is solved the optimization problem with the following objective function:

$$E(\mathbf{T}) = \sum_{(\mathbf{b},\mathbf{a})\in\mathcal{K}_{III}} \rho(\|\mathbf{b} - \mathbf{T}\mathbf{a}\|), \tag{3}$$

where ρ is the penalty term. This penalty function is very important because a well-chosen penalty allows to control the shape of the objective function (3) by rapidly doing validation and pruning of bad correspondences to remove them from consideration. Also, it allows to solve optimization for one pass, without recomputation during the optimization as in the Standard-ICP algorithm. Authors use Geman-McClure estimator as penalty function [22]. The optimization problem (3) can not be solved directly. Therefore authors use Black-Rangarajian duality [23] that allows to optimize the objective function (3) very fast.

3) Teaser++: Teaser++ registration method can be divided conditionally into the following main submodules: Advanced Matching, Maximal Clique Inlier Selection (MCIS) and transformation estimation [20]. Authors of Teaser++ also in FGR use Advanced Matching [19] to build correspondences $(\mathbf{a}_i, \mathbf{b}_i)$ by using extracted feature points for each point of source $\mathcal{A}$ and target $\mathcal{B}$ point clouds. Next, they use MCIS algorithm to prune a significant amount of outliers. Then Truncated Least Squares (TLS) optimization problem is formulated to estimate the unknown transformation based on pruned correspondences:

$$\min_{s>0,\mathbf{R}\in SO(3)t\in\mathbb{R}^3} \sum_{i=1}^{N} \Big(\frac{1}{\beta_i^2}\|\mathbf{b}_i - s\mathbf{R}\mathbf{a}_i - \mathbf{t}\|^2, \bar{c}^2\Big), \tag{4}$$

where $s > 0$ is estimated scale, $\mathbf{R} \in SO(3)$, $\mathbf{t}$ are estimated rotation and translation, β_i is given bound of Gaussian noise for i correspondence, $\bar{c}^2$ is threshold to penalty the correspondences with big residual. This formulation allows to take into account that the set of correspondences has an extreme amount of outliers and that inliers have unknown Gaussian noise. For solving the optimization problem (4) the authors introduced two invariant measures: the Translation Invariant Measurement (TIM), and the Translation and Rotation Invariant Measurement (TRIM) [20]. It created the possibility to decouple transformation estimation on three separate estimations: scale, rotation, and translation. The scale and translation estimation problems are solved in polynomial time using an adaptive voting algorithm [24]. The TLS rotation estimation is relaxed to tight semidefinite relaxation problem and solved fast by using graduated non-convexity [25].

III. Datasets

In our work, we compare point cloud registration algorithms for the co-localization problem of mixed reality devices. The algorithm takes point clouds of the reconstructed environment from one and the other device. The reconstructed point cloud only approximates the geometric parameters of the user's real environment but does not describe it with high accuracy. So data set is required to consist of reconstructed point clouds pairs for different locations, and each pair of point clouds should have different point distributions from each other.

Dataset A. KTH Longterm and ICL-NUIM datasets. The dataset A consists of two sub-datasets: KTH Longterm[1] and ICL-NUIM[2]. KTH Longterm was collected autonomously by a Scitos G5 robot with an RGB-D camera on a pan-tilt. It contains data from 8 different areas of the KTH office environment. Half of the areas are rooms and the others are corridors. ICL-NUIM sub-dataset was collected by RGB-D camera sensor and contains data of two rooms: living and office rooms. As a result, the first dataset contains 11 pairs of KTH Longterm point clouds and 2 pairs of ICL-NUIM point clouds. In both cases, the data is not collected from mixed reality devices.

Dataset B. Indoor HoloLens dataset. We collected dataset B using two mixed reality devices: Microsoft HoloLens 1st and 2nd gen. Each device builds a mesh map of the environment. We have explored one space from two devices and used Windows Device Portal to download the Spatial mapping. Spatial mapping is the mesh, we take only vertexes as a point cloud. Dataset B contains sparse point clouds of 4 areas: three different rooms and a corridor. We obtained 20 pairs of point clouds, where each pair of point clouds is obtained from different devices.

To evaluate the accuracy, we created a synthetic dataset of point clouds based on real point clouds from dataset B. We selected large samples of point clouds and took different size parts from each large cloud. Each part has the ground truth transformation matrix is set for each pair randomly in the range of $[-90, 90]$ degree for the rotation and $[-50, 50]$ cm for the translation. The basic idea of creating a synthetic dataset is that we know the actual position of a pair of point clouds relative to each other before registration. Hence, we can estimate the accuracy of the point cloud pair alignment.

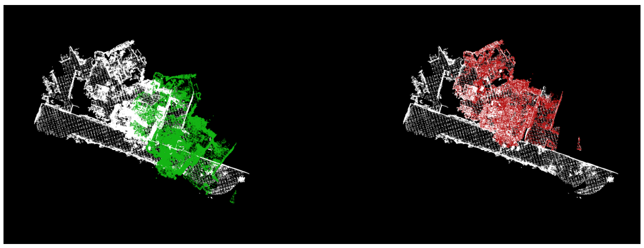
IV. Methodology

A. Efficiency evaluation of registration algorithms

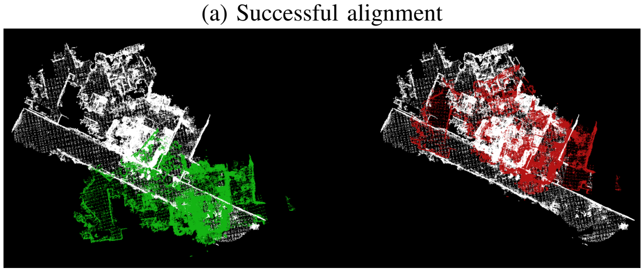
At the first stage, we evaluated the effectiveness of four registration algorithms: Go-ICP, Bayesian-ICP, FGR, and

[1]https://strands.readthedocs.io/en/latest/datasets/kth_lt.html
[2]http://redwood-data.org/indoor/dataset.html

Teaser++. FGR and Teaser++ were also evaluated depending on Advanced Matching usage modes and different feature descriptors namely FPFH, WHI16, WHI36. The used feature radius did not exceed 150 cm with the ratio close to the recommended, which is described in the next subsection. The effectiveness assessment consisted of the average registration time and the rate of successful alignments. There is no ground truth information about the actual transformation between pairs of point cloud origins. Thus, the alignment success for each pair was evaluated visually, as shown in the example (Fig. 1). All algorithms were tested on datasets A and B.



(a) Successful alignment



(b) Non-successful alignment

Fig. 1: Examples of visually successful and non-successful registrations. The left half of the sub-figures shows state before registration and the right half shows a state after registration

B. Accuracy and runtime analysis of registration methods: FGR and Teaser++ for different local feature descriptors

At the second stage, we studied the efficiency of the algorithms depending on the different radius of the features FPFH, WHI16, and WHI36. We use synthetic data set for accuracy and runtime evaluation. To find rotation error, we use roll, pitch, and yaw angles calculated for transformation matrix T_a obtained by the algorithm and for ground truth transformation matrix T_g:

$$\begin{gathered}\phi = \operatorname{atan2}(r_{32}, r_{33}), \theta = \arcsin(-r_{30}),\\ \psi = \operatorname{atan2}(r_{21}, r_{11}),\end{gathered} \tag{5}$$

where r_{ij} - is ij element of the rotation part in transformation matrix. Rotation error R_{error} is defined as summary error of roll (ϕ_{error}), pitch (θ_{error}), yaw (ψ_{error}) angles:

$$\begin{gathered}\phi_{error} = |\phi_g - \phi_a|, \theta_{error} = |\theta_g - \theta_a|,\\ \psi_{error} = |\psi_g - \psi_a|,\\ R_{error} = \phi_{error} + \theta_{error} + \psi_{error}.\end{gathered} \tag{6}$$

Translation error t_{error} was calculated following:

$$t_{error} = \sqrt{(x_g - x_a)^2 + (y_g - y_a)^2 + (z_g - z_a)^2}. \tag{7}$$

We also evaluated the success rate of registrations for FGR and Teaser++. For FPFH, WHI16, WHI36 local feature descriptors we calculated metrics for different feature radius, but we kept optimal ratio downsampling (r_d)/normal (r_n)/feature (r_f) radius equal $1:2:5$ for FPFH and optimal ratio downsampling (r_d)/feature (r_f) radius equal $1:5$ for WHI types feature descriptors. FPFH ratio is recommended by authors of FGR and WHI ratio is selected by us for a more reliable comparison of correspondence-based registration methods with different feature descriptors. Successful alignment of two synthetic point clouds we considered the fulfillment of accuracy condition:

$$R_{error} \leq 0.03rad; \quad t_{error} \leq 1.0cm. \tag{8}$$

Such accuracy requirements we considered as satisfactory for the co-localization of mixed reality devices. Regarding the time of registration, it is enough not more than 5 seconds for successful synchronization of the devices, because it is enough one registration with further update and refinement. Thus, the success in registration we considered the fulfillment of the following conditions: 100% cases satisfy accuracy condition (8) and runtime bellow 5 s.

C. Accuracy and runtime analysis of hybrid approaches

In the third stage, we evaluated the registration efficiency of the hybrid approach (correspondence-based method [FGR or Teaser++] as coarse + ICP as local refinement). The hybrid approach allows us to use the advantages of the two techniques. The first one does not require good initialization for the point clouds registration, but the usage of downsampling limits the accuracy of the method. The second has high convergence accuracy (moves to a local minimum, converges globally only when close to the global minimum), but it requires having the appropriate initial position of point clouds relative to each other. So the first method allows to exclude the disadvantages of the second, and the second method allows to exclude the disadvantages of the first.

We used different numbers of iterations of ICP to evaluate the effectiveness of the hybrid approach and to discover the working range of the radius of features. We used previous metrics and we add a new criterion for a more flexible evaluation: 90% of cases satisfy accuracy condition (8) and runtime 10 s. Since accuracy and runtime can be improved by using the hybrid approach sub-modules improvements and more powerful hardware.

All experiments were done by using Point Cloud Library (PCL) [26] on a laptop with CPU AMD Ryzen 7 4800HS.

V. Results

A. Efficiency evaluation of registration algorithms on real datasets

Tables I and II show estimated registration time and the rate of success alignments of point clouds pairs. The algorithms based on ICP: Go-ICP and Bayesian-ICP have shown a very low success rate of alignments with high registration time.

TABLE I: Results for FGR and Teaser++ on dataset A.

Method	Feature	Advanced Matching	Average runtime (ms)	Alignment success (%)
Go-ICP	-	-	24427	8
Bayesian-ICP	-	-	1647	54
FGR	FPFH	On	390	100
	WHI16		371	100
	WHI36		752	100
	FPFH	Off	442	100
	WHI16		441	100
	WHI36		823	92
Teaser++	FPFH	On	409	100
	WHI16		428	100
	WHI36		823	100
	FPFH	Off	1847	100
	WHI16		1209	100
	WHI36		1897	100

TABLE II: Results for FGR and Teaser++ on dataset B.

Method	Feature	Advanced Matching	Average runtime (ms)	Alignment success (%)
Go-ICP	-	-	24158	0
Bayesian-ICP	-	-	1564	5
FGR	FPFH	On	219	68
	WHI16		223	53
	WHI36		419	79
	FPFH	Off	259	42
	WHI16		262	26
	WHI36		458	42
Teaser++	FPFH	On	219	63
	WHI16		213	58
	WHI36		446	63
	FPFH	Off	365	79
	WHI16		382	100
	WHI36		641	100

Features-based algorithms (FGR and Teaser++) significantly outperform ICP-based ones in terms of both successful alignments and execution time.

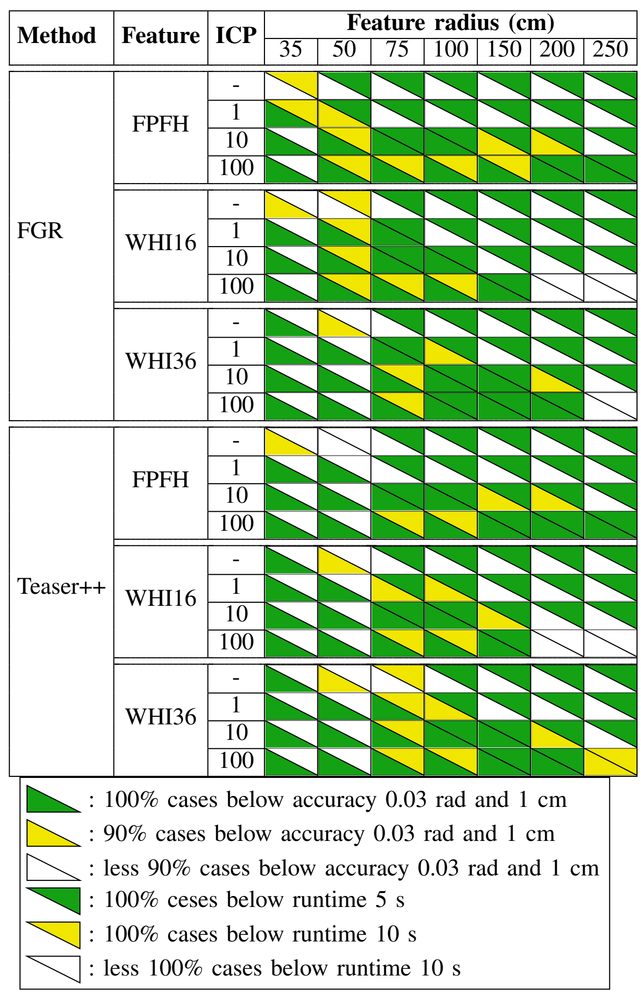
The rate of successful alignments by algorithms on dataset B is lower than on dataset A. This indicates that dataset B has cloud pairs that are more difficult for registration. In dataset B, more than half of the point cloud pairs have a small overlap fraction: less than 50%, and more different points distribution. The registration success of a point cloud pair depends on the overlap fraction as well as the initial degree of point cloud sparsity. The overlap fraction determines how many pairs of point clouds have common geometric parts. In other words, the greater the overlap area of the point clouds, the easier it is for the registration algorithm to find correspondences between them and match them to each other. As for the sparsity degree of the point clouds, it affects the degree of dissimilarity in point cloud distribution. Sparse clouds may have more dissimilar point distributions than dense clouds.

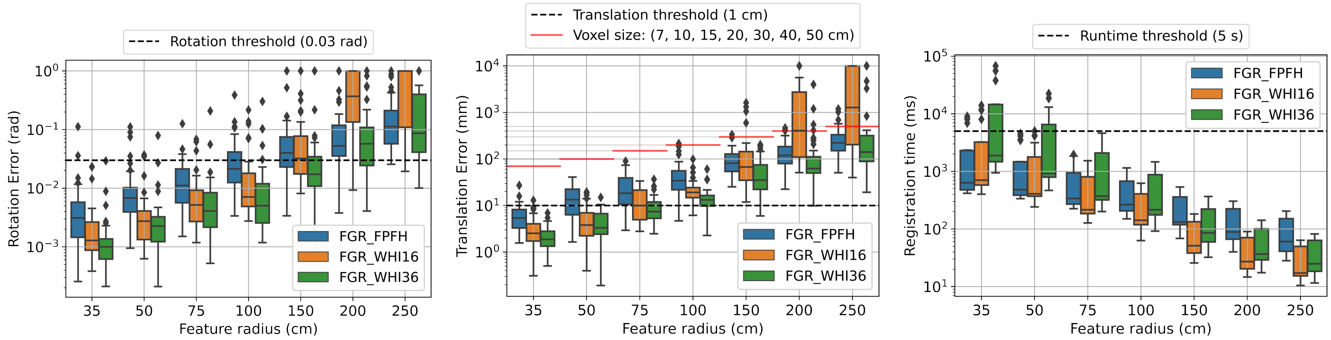
When the Advanced Matching for FGR algorithm is turned off, the probability of successful point cloud alignment tends to decrease. At the same time for the Teaser++, when Advanced Matching is turned off, the percentage of successful point cloud alignments on the contrary increases and reached 100% with local feature descriptors WHI16 and WHI36. It happens because probably the pruning correspondences part of Advanced Matching rejects not only wrong correspondences but also part of good ones, and MCIS submodule of Teaser++ selects inliers effectively.

B. Accuracy and runtime analysis of FGR and Teaser++ for different feature descriptors

Figures 2 and 3 show accuracy and runtime results of FGR, Teaser++ alignments with different local feature descriptors. The rotation and translation accuracy of the methods with local feature descriptors WHI16, WHI36 exceeds the accuracy of the methods with FPFH, while the runtime of methods with feature WHI16 is the shortest for feature radius more than 50 cm. In figures 2 and 3, we can notice that point cloud registration translation accuracy is less than the downsampling level for feature radius 150 cm or less. Voxel Grid Downsampling allows saving the surface structure since a centroid point is calculated for each voxel. Hence, on the one hand, the small voxel size allows to slightly reduce the number of calculations without losing information about the surface geometry. On the other hand, with a large feature radius, we significantly speed up calculations. Therefore, there is a compromise between

TABLE III: Results for FGR and Teaser++ on synthetic dataset



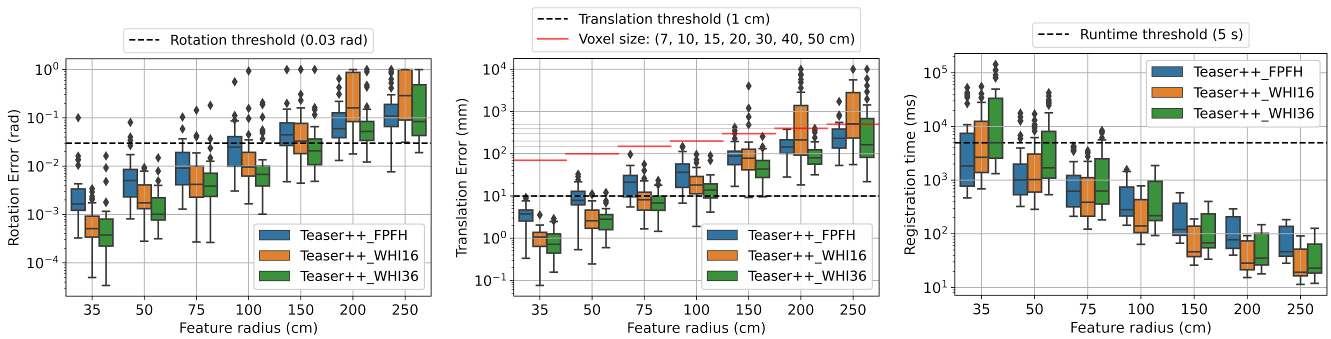


(a) Rotation estimation

(b) Translation estimation

(c) Runtime estimation

Fig. 2: FGR_FPFH vs FGR_WHI16 vs FGR_WHI36.



(a) Rotation estimation

(b) Translation estimation

(c) Runtime estimation

Fig. 3: Teaser++_FPFH vs Teaser++_WHI16 vs Teaser++_WHI36.

accuracy and runtime, that is the optimal range of the feature radius satisfying some accuracy and runtime requirements of real applications.

C. Accuracy and runtime analysis of hybrid approaches: FGR and Teaser++ with ICP

We used FGR and Teaser++ with ICP as a hybrid approach. Table III shows the summary evaluation of FGR and Teaser++ for different types and parameters of local feature descriptors and ICP iterations. The table shows the radius ranges of the feature descriptors that correspond to the defined accuracy and execution time criteria. The table III can be useful in determining the optimal feature radius range needed to solve the collaborative localization problem. After our tests, we can notice an increase in registration accuracy for all features radius, not more than 150 cm. There is no increase in accuracy for the WHI16 feature radius of more than 150 cm. It appears because the FGR and Teaser++ methods have poor accuracy for the WHI16 feature radius greater than 150 cm. This creates faulty initialization for ICP and as a result, ICP falls to a local minimum. It is significant that the hybrid approach for FPFH feature radius of more than 150 cm lets ICP converge to a global minimum. The accuracy of FGR and Teaser++ for FPFH feature the radius of more than 150 cm is greater than for WHI feature type. It is possible that WHI for a large feature radius partially loses its descriptiveness, unlike the FPFH feature. Based on the results with the synthetic dataset, it cannot be reliably determined feature descriptor is best used in a real application. But the results on real data show the superiority of WHI feature over FPFH. Therefore we recommend using WHI feature in the feature radius range before 150 cm, and for more feature radius to use FPFH.

VI. Conclusion

In this paper, we proposed the approach to mixed reality cross-devices localization and show its performance and limitations. We estimated the efficiency of four point cloud registration methods: Go-ICP, Bayesian-ICP, FGR, Teaser++ on real point clouds of rooms obtained by Microsoft HoloLens (1st and 2nd gen) MR devices. Feature correspondences-based methods: FGR and Teaser++ showed milliseconds runtime efficiency and a high probability of successful alignments compared with ICP based methods: Go-ICP, Bayesian-ICP. We tested a new WHI feature descriptor for the point cloud registration method. On synthetic data, we tested a hybrid approach and provided the table with different algorithm parameters and performance for co-localization MR devices. For co-localization MR devices in a real scenario, we recommend using WHI feature descriptor with feature radius before 150 cm, as it is more robust to interference and descriptive on real data compared to FPFH.

In future works, we would like to extend the approach for co-localization of the multi-robot system and multi mixed reality device in one space. We will investigate the methods for localization of MR devices in large pre-built and labeled maps.

REFERENCES

[1] Microsoft, "Azure spatial anchors," https://azure.microsoft.com/en-us/services/spatial-anchors/, 2019 (accessed November 11, 2021).

[2] Niantic, "Niantic lightship," https://lightship.dev/, 2015 (accessed November 11, 2021).

[3] G. Developers, "Share ar experiences with cloud anchors — arcore," https:// developers.google.com/ar/develop/java/cloud-anchors/overview-android, 2018 (accessed November 11, 2021).

[4] M. Dusmanu, O. Miksik, J. L. Schönberger, and M. Pollefeys, "Cross-descriptor visual localization and mapping," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6058–6067.

[5] P. C. Ng and S. Henikoff, "Sift: Predicting amino acid changes that affect protein function," *Nucleic acids research*, vol. 31, no. 13, pp. 3812–3814, 2003.

[6] Y. Tian, X. Yu, B. Fan, F. Wu, H. Heijnen, and V. Balntas, "Sosnet: Second order similarity regularization for local descriptor learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 016–11 025.

[7] T. Sun, G. Liu, S. Liu, F. Meng, L. Zeng, and R. Li, "An efficient and compact 3d local descriptor based on the weighted height image," *Information Sciences*, vol. 520, pp. 209–231, 2020.

[8] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 3212–3217.

[9] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. International Society for Optics and Photonics, 1992, pp. 586–606.

[10] S. Granger and X. Pennec, "Multi-scale em-icp: A fast and robust approach for surface registration," in *European Conference on Computer Vision*. Springer, 2002, pp. 418–432.

[11] L. Maier-Hein, A. M. Franz, T. R. Dos Santos, M. Schmidt, M. Fangerau, H.-P. Meinzer, and J. M. Fitzpatrick, "Convergent iterative closest-point algorithm to accomodate anisotropic and inhomogenous localization error," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 8, pp. 1520–1532, 2011.

[12] D. Chetverikov, D. Stepanov, and P. Krsek, "Robust euclidean alignment of 3d point sets: the trimmed iterative closest point algorithm," *Image and vision computing*, vol. 23, no. 3, pp. 299–309, 2005.

[13] S. Kaneko, T. Kondo, and A. Miyamoto, "Robust matching of 3d contours using iterative closest point algorithm improved by m-estimation," *Pattern Recognition*, vol. 36, no. 9, pp. 2041–2047, 2003.

[14] 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*, vol. 38, no. 11, pp. 2241–2254, 2015.

[15] D. R. Morrison, S. H. Jacobson, J. J. Sauppe, and E. C. Sewell, "Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning," *Discrete Optimization*, vol. 19, pp. 79–102, 2016.

[16] F. A. Maken, F. Ramos, and L. Ott, "Estimating motion uncertainty with bayesian icp," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 8602–8608.

[17] ——, "Speeding up iterative closest point using stochastic gradient descent," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6395–6401.

[18] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient langevin dynamics," in *Proceedings of the 28th international conference on machine learning (ICML-11)*. Citeseer, 2011, pp. 681–688.

[19] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *European Conference on Computer Vision*. Springer, 2016, pp. 766–782.

[20] H. Yang, J. Shi, and L. Carlone, "Teaser: Fast and certifiable point cloud registration," *IEEE Transactions on Robotics*, 2020.

[21] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. M. Kwok, "A comprehensive performance evaluation of 3d local feature descriptors," *International Journal of Computer Vision*, vol. 116, no. 1, pp. 66–89, 2016.

[22] D. Geman and C. Yang, "Nonlinear image recovery with half-quadratic regularization," *IEEE transactions on Image Processing*, vol. 4, no. 7, pp. 932–946, 1995.

[23] M. J. Black and A. Rangarajan, "On the unification of line processes, outlier rejection, and robust statistics with applications in early vision," *International journal of computer vision*, vol. 19, no. 1, pp. 57–91, 1996.

[24] H. Yang and L. Carlone, "A polynomial-time solution for robust registration with extreme outlier rates," *arXiv preprint arXiv:1903.08588*, 2019.

[25] H. Yang, P. Antonante, V. Tzoumas, and L. Carlone, "Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1127–1134, 2020.

[26] "Point cloud library (pcl)," *https://pointclouds.org/*, 2020.