

Global Alignment of Meshes for the Microsoft HoloLens

Jessica Yi Fei Bo

ybo@student.ethz.ch

Kevin Shum

kshum@student.ethz.ch

Karen Hong

hongk@student.ethz.ch

Abstract

With devices such as the Microsoft HoloLens now able to scan and record 3D scenes as meshes, a novel way of registering scenes taken from different coordinate systems and at different times is required. We base our work off of the Guaranteed Outlier Removal Method presented by Bustos and Chin and expand it beyond point clouds to analyze 3D meshes. However, we could not conclude that GORE can perform consistently and effectively on real-world datasets generated by the HoloLens. In particular, we identified that GORE has the most difficulty when the points from two datasets do not have significant overlap, notably any values below the 80% threshold. This places severe limitations on the applications of GORE, especially on real-life alignment problems.

1. Introduction

In the field of 3D vision, registration is necessary to align two different point clouds with different reference frames. The motivation for this problem is to integrate similar features from different 3D registrations or time steps. Given a point cloud X and another Y , the goal is to find the transformation, f , that maps the reference frame of X to that of Y so as to minimise the transformation error and align the points as closely as possible. The focus of this paper is limited to rigid transformations, including rotations and translations.

The canonical approach for point cloud registration is to use a set of feature or point matches and to apply least squared analysis to estimate the transformation f . This approach becomes less robust in the presence of outliers in the sample set. The maximum consensus approach, such as RANSAC, is used in practice to find the transformation that agrees with as many of the input matches as possible up to a certain threshold. RANSAC, however, has been found to increase the runtime exponentially in relation to the outlier ratio, and does not find the optimal solution [4].

The motivation for this project is to expand the scope of geometric registration by focusing on mesh registration. Meshes generated by the Microsoft HoloLens and other

similar recording devices register on different coordinate systems with each subsequent scan. As a result, it is a non-trivial problem to accurately, efficiently, and robustly align these meshes. Other global registration algorithms, such as Branch-and-Bound, optimise an objective function, but have high computation costs with large point clouds in addition to high outlier rates [5].

Applications for mesh registration include Virtual Reality (VR) and Augmented Reality (AR), in which the stitching together of distinct meshes is required to match features taken at different points in time. One particularly notable application includes applying the algorithm to the ongoing Build2Spec project, which uses the HoloLens for overlaying meshes for construction management.

In this paper, our main objective is to implement a mesh registration pipeline based on the Guaranteed Outlier Removal (GORE) algorithm, developed by Bustos and Chin. We seek to refine GORE with the Iterative Closest Points (ICP) and RANSAC algorithms, and compare the performance of the GORE pre-processing step in relation to the subsequent registration algorithms.

2. Work Distribution

The group worked together closely throughout the project, so specific tasks were distributed as they arose. Due to this scheme, we had high flexibility with our work break-out schedule, but also overlap in tasks. The distribution is outlined in Table 1.

	Kevin	Karen	Jessica
MATLAB implementation of ICP refinement	✓		✓
MATLAB analysis of point cloud models			✓
PCL implementation of pfh and iss3d	✓	✓	
HoloLens spatial map generation		✓	
Processing and generation of spatial map data		✓	
MATLAB analysis of spatial maps	✓		✓

Table 1. Work distribution between the project members.

3. Related Works

There is a significant volume of work in the area of geometric registration [1, 2, 6]. However, there is little research concerning the registration of 3D meshes and the feasibility of running these algorithms on the Microsoft HoloLens. Iterative approaches such as RANSAC are robust but computationally expensive for datasets with large numbers of outliers where many iterations are required for precise results [4]. The Branch-and-Bound (BnB) method is able to guarantee globally-optimal solutions but is, once again, very slow to execute [5].

Several recent works aim to improve the efficiency of point cloud and shape registration through reducing computational costs or increasing the robustness of the registration [1, 2, 6]. These developments are highly relevant to our work on mesh registration, given that the main challenges of working with the HoloLens are its limited computational power and the quality of the generated meshes, which tend to be large datasets with lots of noise. Therefore, we have identified several algorithms which are promising for further work in furthering research in HoloLens mesh registration.

A common tactic to reduce the computation time of iterative algorithms is to reduce the size of the input data. The GORE method improves the runtime of RANSAC by removing outliers from the input set [2]. Bustos and Chin developed this pre-processing method using purely geometric operations which ensures that the process is fast. In addition, since the algorithm guarantees the removal of only outliers, the optimal solution can still be found by further alignment with iterative methods such as RANSAC and ICP.

An optimized method for determining the rotation between two 3D point clouds was proposed by Bustos et al. [1]. Their work improves upon the BnB algorithm by developing a new bounding function that computes tighter bounds. By using stereographic projections to pre-compute possible point matches, the new bounding function can be determined very efficiently and rapidly. Adopting this rotational transformation approach within a nested BnB algorithm results in a fast and globally optimal search method.

Another algorithm, developed by Zhou et al. [6], attempts to use a completely different approach to shape registration. The fast global registration algorithm does not rely on iterative sampling, model fitting, or local refinement. Instead, the algorithm operates on densely defined surfaces and uses candidate matches to make a tight alignment. This method was found to be capable of working on partially overlapping 3D surfaces and noisy data sets. In addition, it was also found to be faster than local refinement algorithms.

4. Methodology

4.1. The Mesh Registration Pipeline

Given a set of two HoloLens meshes, our method processes the input data through the following pipeline as seen in Figure 1.

1. Points cloud are obtained by converting the vertices of each mesh to points in a point cloud. A better alternative that is recommended for future experiments is to obtain the point cloud by sampling the surface points of the mesh.
2. Point cloud normals and keypoints are extracted, the latter using the ISS 3D algorithm to remain consistent with methods applied in the original GORE experiment.
3. A point feature histogram is computed for each point cloud using the Point Feature Histogram (PFH) algorithm from the Point Cloud Library (PCL).
4. A Matlab data structure is populated with the point cloud, keypoints, and feature histogram data.
5. The point clouds are aligned using a combination of GORE, RANSAC, and ICP. The following combinations were analysed: 1) ICP only; 2) GORE + ICP; 3) GORE + RANSAC; 4) GORE + RANSAC + ICP.
6. Analysis is conducted on the runtime, angular error, and translational error of the output alignment matrix.

4.2. The GORE Theory

The main component of the pipeline relies on the theory of GORE. Using GORE as a pre-processing step is argued to significantly reduce the size of the input set, allowing refinement algorithms to have better performance on the output of GORE. The main algorithm behind GORE is summarized below.

- Given a set of input points and correspondences, iterate over each point:
 1. Compute the improved lower bound, l , and upper bound, \hat{p}_k
 2. Reject the current point match as a true outlier if the values of the bounds are not consistent

GORE seeks to reject true outliers in a set of points H , reducing the set to H' . H' is guaranteed to be included in the globally optimal solution I^* .

The maximum consensus problem is defined as:

$$\underset{T \in SE(3), I \subset H}{\text{maximize}} |I| \text{ subject to } \|Rx_i + t - y_i\| \leq \xi, \forall i \in I$$

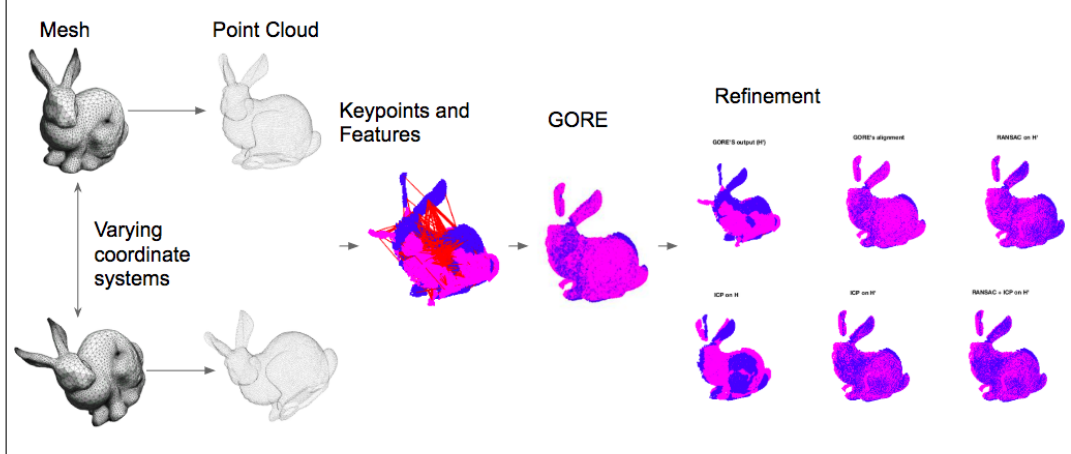


Figure 1. The inputs to the pipeline are independently generated spatial scans (meshes) created on the HoloLens with different coordinate systems. The output is a transformation matrix that aligns the two meshes into the same coordinate system. For simplicity, basic bunny meshes are used in place of HoloLens meshes to demonstrate the pipeline outlined in Section 4.1.

5. Results

5.1. Performance on Datasets with Ground Truth

We implemented the alignment pipeline on control datasets taken from online repositories and real-life datasets generated by the Microsoft HoloLens. The open source datasets were the same as the ones used by Bustos and Chin; namely, the Stanford 3D Scanning Repository [3] (bunny, armadillo, dragon, and buddha), Mians dataset (t-rex, parasauro, chef, and chicken), and the ISPRS (vaihingen-a and vaihingen-b). The datasets were pre-transformed by the authors of GORE, so we used the same modifications in our analysis to remain consistent with the original paper. Using the HoloLens, we also created spatial maps of a small, enclosed room ($\geq 10m^2$) with several distinctive and static furnishings. The spatial scans were obtained using several methods which will be discussed in Section 5.2.1. The OBJ representations of the room are found in Figure 2.

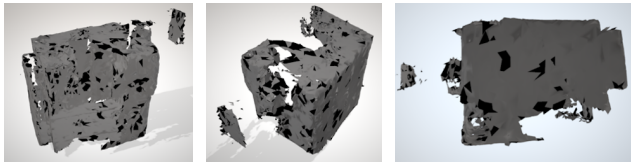


Figure 2. Example of a HoloLens spatial map generated from a small, indoor environment. The room contains a mirror, which was not mapped due to its reflection of the surroundings (see the gap in the mesh in the middle and right images). However, the environment exterior to the window was captured, hence the rectangular block projects from an opening in the adjacent wall.

Using the control datasets, we examined the runtimes, angular error, and translation error of different alignment

methods. Tests were conducted for pure rotation (from 10° to 90° per axis of rotation) and pure translation (logarithmic increments from 0.1 to 100). Due to the randomness of the point clouds, the results from all ten datasets were averaged and tabulated for each test.

We observed that the runtime for GORE is significantly higher than any other processing step. The times range from 0.29s (Stanfords dragon in rotation) to 3.96s (ISPRSs vaihingen-1 in translation). The runtimes sometimes fluctuated with increasing rotation and translation, but mostly remained steady. There is no correlation between the size of the point cloud and GOREs runtime because the number of keypoint matches is set by the user and was standardized for all trials. Thus, it could be inferred that runtime is dependent on the geometry and features of the point cloud.

Among the other processing steps, standalone ICP had computational costs at about an order of magnitude larger than GORE + ICP/RANSAC. Table 2 summarizes the average runtime values for each step in the pipeline.

Regarding the alignment errors, the trend varies from dataset to dataset, or does not exist at all. In general, using ICP as a refinement step after running GORE + RANSAC improved angular alignment errors but worsened translation errors. In Figure 3, we plotted the normalized angular and translation errors against the input transformation to showcase the relative differences between the three alignment methods. The post-alignment errors were normalized by dividing through the transformation error in order to standardize the comparison metric between multiple point clouds and mitigate bias towards datasets that produce larger errors. Hence, the observation that errors decrease with increasing rotation or translation is misleading, as those errors are divided by large initial errors. Examining individual datasets, the alignment errors usually remain steady or

	GORE	RANSAC (after GORE)	ICP	ICP (after GORE)	ICP (after GORE and RANSAC)
Rotation	1.43	0.0057	0.051	0.058	0.0183
Translation	1.38	0.0041	0.048	0.0072	0.0041
Average	1.41	0.0044	0.055	0.0079	0.0047

Table 2. Runtimes in seconds for pure rotation, pure translation, and the average for the control datasets.

increase slightly with increasing transformation; but there is no consistent pattern.

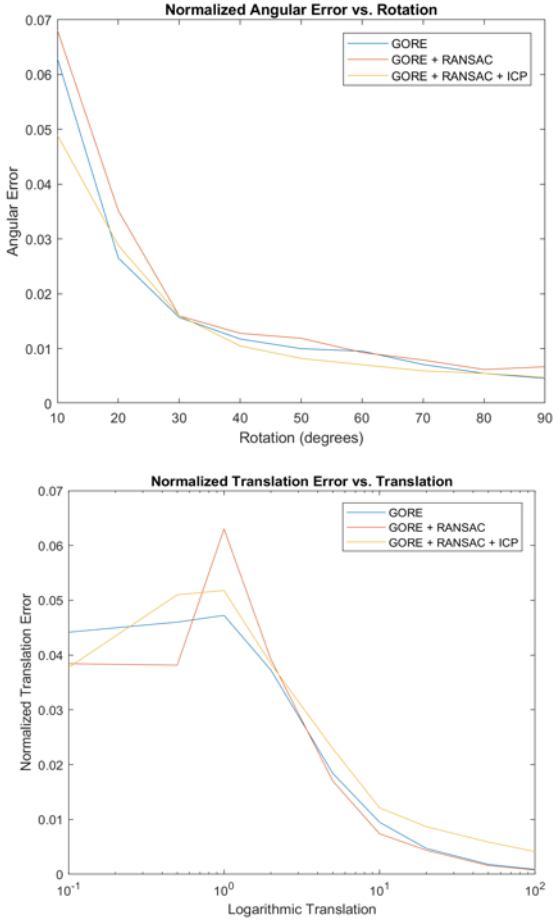


Figure 3. Graph above shows the average normalized angular error for each angle of rotation. Graph below shows the average normalized translation error for each translation increment.

5.2. Performance on HoloLens datasets

5.2.1 Capturing HoloLens Spatial Maps

We hypothesized that changing the scanning parameters of the HoloLens could drastically affect the results of alignment. In order to test the alignment pipeline on different types of input data, we used the HoloLens to generate spatial scans of 1) the same room in separate different coordinate frames; 2) separate scans of the room in the same co-

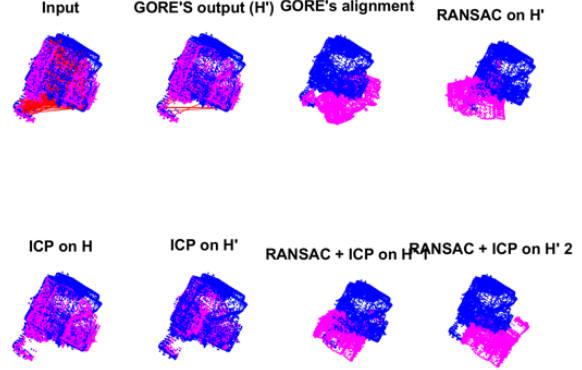


Figure 4. Failed alignment of the HoloLens mesh, scanned using Method 3.

ordinate frame; and 3) two versions of the same scan of the room in the same coordinate frame, saved time t apart (this would map new parts of the room or remap known parts).

However, all three scanning methods failed to produce accurate alignment results (refer to Figure 4). In many of the trials, GORE could not reduce the keypoint matches to eliminate outliers. This significantly increased the runtime of RANSAC. The lack of success is most likely due to the dissimilarity between the points of different scans exceeding a certain operational threshold for GORE – e.g. GORE is not able to identify that two point clouds with the same gross geometry but different vertices are still the same spatial map. To verify this theory, we then performed tests using the same point cloud divided into overlapping subsets or with introduced noise, as an attempt to quantify at which point GORE begins to fail. The input parameters to the pipeline are listed under Table 3 and kept consistent for both analyses.

5.2.2 Overlapping Subsets

Additionally, we analysed the performance of the pipeline using overlapping subsets taken from the same dataset, from 50% up to 100% in 5% increments. The control mesh subset was kept constant as the first 75% of points in the mesh. The second mesh was taken as the last 75% points. Then the constant size sliding window approached left until both subsets achieved 100% overlap. A visualization of the subsets is represented in Figure 5.

The results indicate that as we approach a complete overlap, runtime generally improves, along with a decrease in

Desired Matches	Rotation (degrees)	Translation	Normal Comput. Radius	Features Compt. Radius	Noise Magnitude
300	20	5	0.3	0.5	0.01

Table 3. User set parameters that were selected after testing and kept constant for all the experimental trials.

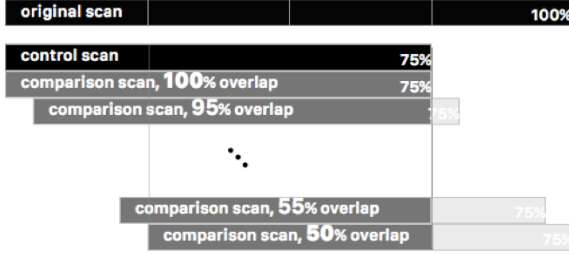


Figure 5. Diagram of the overlapping subsets approach. The control scan was taken from the first 75% of the original mesh, and a comparison scan was taken along a sliding window of the same size, starting from a 100% overlap to 50% in increments of 5%.

angular and rotational error. For all three attributes, we find the minimum at 100% overlap, as expected. From 100% decreasing to 80% overlap, we find that GORE + RANSAC performs the best, followed by GORE + RANSAC + ICP refinement. At 80% overlap, we start to see the runtime and error terms sharply increase. At this point, the scans appear to have significant dissimilarities to warrant increased computation time. This is likely a direct consequence of significantly higher outlier rates due to GOREs ineffectiveness at low overlap ratios. At this threshold, GORE has the least impact, increasing runtime computation but not decreasing the translation or rotation rates. From 80% overlap onwards, we find that GORE generally helped improve runtime and error rates, with the combination of GORE + RANSAC + ICP performing the worst, followed by GORE + RANSAC, and GORE + ICP performing the best. However, all approaches performed on a similar magnitude of runtime and error. We conclude that GORE as a pre-processing step is able to maintain the runtime and keep rotational and translational errors at a minimum as long as the scans are not too dissimilar. Once the scans start to diverge in similarity, below the 80% threshold, GORE no longer serves as an efficient and robust pre-processing step, and the registration algorithms are prone to significantly increased errors and runtime.

5.2.3 Noise Introduction

To verify the results of the overlap analysis, we re-approached the problem through introducing noise to a subset of points. We added random noise to a percentage of the points in the point cloud and aligned it against the original dataset, then increased the percentage of points by 5% each time. This way, we can guarantee a certain percent-

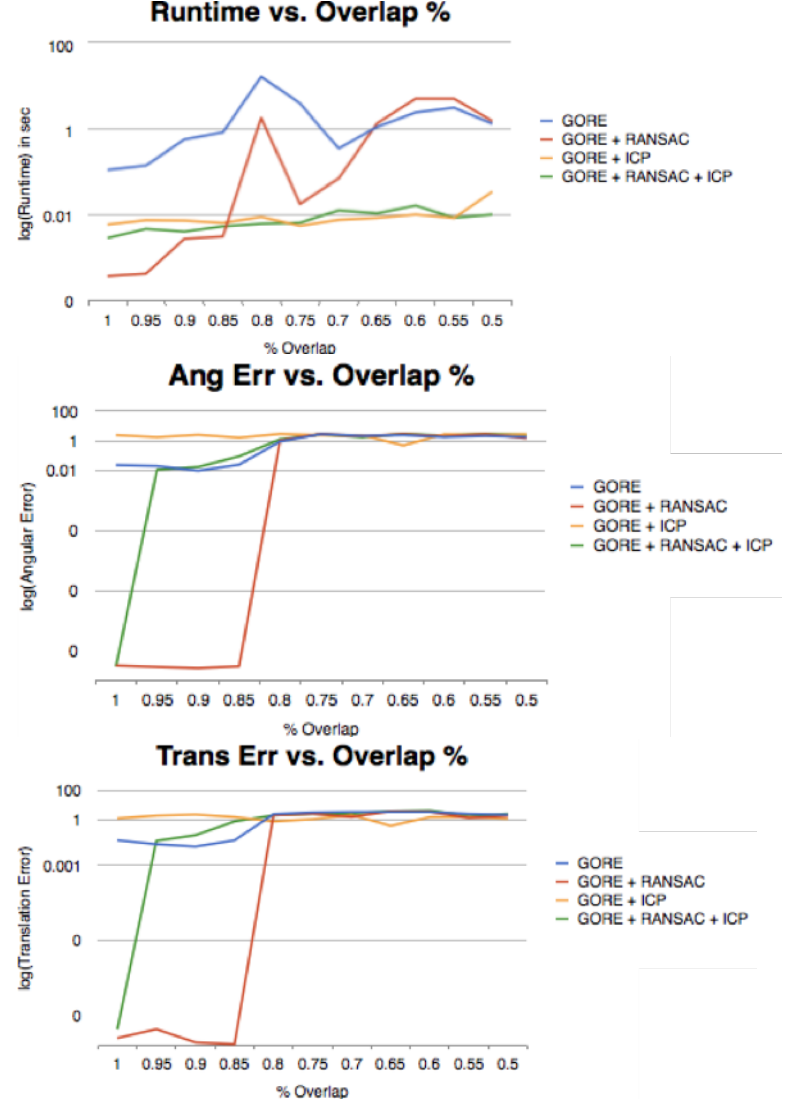


Figure 6. Logarithmic runtime, angular error and translational error were plotted against decreasing percent overlap of two subsets taken from the same scan.

age of overlap between the two models. Five trials were performed per increment to reduce the influence of outlying results. At a low noise percentage, GORE performed well for almost all the trials. At higher noise, GORE was only able to reduce outlier matches successfully half of the times, so average performance decreases. We stopped the analysis at 20% as that was found to be the approximate threshold where GORE stops performing effectively. These results

indicate that the subsets need 80% of similarity, which corroborates with the results from the overlap analysis.

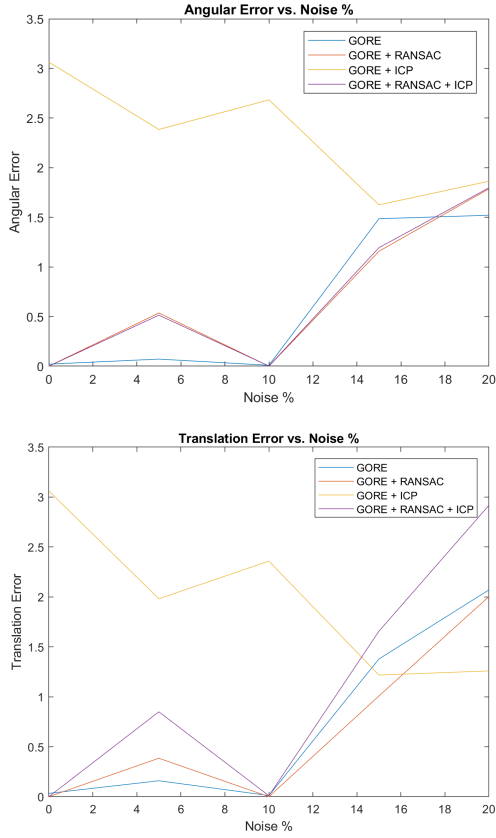


Figure 7. Graph above shows angular error as a function of percentage of points affected by noise. Graph below shows the same values but with translation error.

5.2.4 Reasons for Failure

Although it is hard to pinpoint one reason for the failure of the experiments to align point clouds past a dissimilarity threshold of 20%, we provide some suggestions as to which factors likely had a high influence. For example, the `phf()` algorithm to calculate Point Feature Histograms required user input parameters in both the normals estimation step and the histogram generation step, in which both values are related to the size of the point cloud and to each other. Basic experimentation was done to verify that the chosen parameters were within a suitable magnitude for alignment, but they were not optimized numerically. We saw very drastic differences in the histograms produced and the alignments achieved when these parameters were changed.

Another potential source of error could be due to the failure of the MATLAB implemented function `matchkps()`, which was written by the authors of GORE. Its purpose is

to generate matches between the keypoints based on their histogram descriptors, but we have no discernable method of measuring how many inliers and outliers are produced when we use the real-world datasets. It is possible that at low rates of overlap, `matchkps()` simply does not produce enough correct matches to enable the rest of the alignment pipeline to perform.

6. Conclusion

We propose additional analysis of ground truth and synthetic datasets taken from HoloLens mesh scans to determine if the HoloLens meshes suffice in terms of their quality and noise levels in practice. If not, an area for potential improvement includes improving the scan quality by averaging out noise from the HoloLens meshes using TSDF for smoothing, an approach taken by Group 26 in the 3D Vision course. Additional information can be taken advantage of as well, such as RGB-D depth maps from the HoloLens to generate more accurate depth maps and matches. RGB-D data was used by the original authors of GORE with conclusive results, but was not explored in this paper due to time limitations.

The ultimate goal for this project is to port our work on C++ and Matlab to C# and subsequently the HoloLens development platform in order to get it running on the device itself. The feasibility of running the mesh registration pipeline directly in C# on the HoloLens was explored and ultimately deemed to be too work intensive for a project of this scope. The available libraries, data structures, and matrix operations in this language are both difficult to work with and insufficient. Therefore, in order to implement the algorithm in C#, it is necessary to also reimplement many supporting functions and features. The option of creating a dynamically linked library in C++ was also explored. This remains a possibility for running the GORE algorithm directly on the HoloLens and can be investigated further in future experiments.

A final goal is to support interoperability with CAD models. One application is for the Build2Spec project, which can utilize CAD models of a space and HoloLens recordings interchangeably, and to combine both into one consistent, global view. With a robust and efficient mesh registration algorithm running on the HoloLens, we can then combine our work with the group working on the difference mapping on the HoloLens to realize a fully-functional platform to combine and analyze different 3D spaces.

7. References

References

- [1] A. J. P. Bustos, T.-J. Chin, and D. Suter. Fast rotation search with stereographic projections for 3d registration. *2014*

IEEE Conference on Computer Vision and Pattern Recognition, 2014.

- [2] A. P. Bustos and T.-J. Chin. Guaranteed outlier removal for rotation search. *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015.
- [3] B. Curless and M. Levoy. A volumetric method for building complex models from range images. *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH 96*, 1996.
- [4] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Readings in Computer Vision*, page 726740, 1987.
- [5] R. Horst and H. Tuy. Global optimization: deterministic approaches. *Springer*, 2003.
- [6] Q.-Y. Zhou, J. Park, and V. Koltun. Fast global registration. *Computer Vision ECCV 2016 Lecture Notes in Computer Science*, page 766782, 2016.