Performing patient alignment utilizing point-cloud surface registration techniques in HoloNav

Maarten Weyns
Supervisor(s): Abdullah Thabit, Ricardo Marroquim
EEMCS, Delft University of Technology, The Netherlands
20-6-2022

A Dissertation Submitted to EEMCS faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

Abstract

In order to be able to use the Microsoft HoloLens for surgical navigation purposes, performing good patient alignment is of utmost importance. This paper will discuss how this patient alignment can be done using different point cloud registration algorithms.

A lot of research is being conducted on point cloud registration algorithms. However, most research assumes that the point clouds to be aligned are almost identical, while patient alignment aims at aligning a very detailed pre-operative scan with a very sparse point cloud obtained by the surgeon using an optical marker.

In order to get around this problem, the HoloLens' depth camera is used to obtain a detailed point cloud so that registration algorithms can be used. Then the performance of different point cloud registration algorithms is tested on this depth sensor data to see whether using the HoloLens' depth sensor is a viable option for patient alignment.

From the results, it appears that algorithmic approaches for performing patient alignment are feasible, but the performance of these algorithms is very dependent on the quality of the input data.

1 Introduction

Surgical navigation has become indispensable from the operating room in the past years. It is a technology that aims to help surgeons perform surgery accurately. For example, the system will help a brain surgeon locate the tumor and aid the surgeon to navigate to it. Currently, these systems use sensors and trackers to locate the patient and the tools of the surgeon, and then display the output on a screen.

Using modern technologies, such as a Microsoft HoloLens, would make this technology even more useful. Using augmented reality to display the navigational information right on the patient itself would get rid of the need for the surgeon to look on a screen, meaning the surgeon can be fully focused on the patient alone. This is exactly what the HoloNav project is aiming to achieve.

However, using the HoloLens brings some challenges. One of the most important elements in this type of technology is patient alignment. This means that the pre-operative model, which is acquired during a scan, should perfectly align to the patient, in order to guarantee accurate navigation.

One of the ways this alignment can be achieved is by using fiducial markers. These are physical markers placed on the patient itself, which are then used for aligning the pre-operative model to the patient. The downside of this alignment method is that is might be inaccurate and very dependent on the accuracy of the marker placement. As an alternative, point cloud registration can be used to perform patient alignment.

The goal of point cloud registration is to estimate a transformation that transforms a source point cloud so that it aligns as closely as possible to a target point cloud. The main way of solving this problem is by using the Iterative Closest Point (ICP) algorithm. This algorithm works by finding the closest point in the target point cloud to each point in the source point cloud. It then performs a transformation to minimize these distances. This is performed iteratively, until the algorithm converges or a set maximum amount of iterations is met. The problem with ICP is that it only relies on the geometric properties of the point clouds. This results in the algorithm easily converging to a local minimum, resulting in inaccurate point cloud registration [6].

That is why point cloud registration is usually split into two parts: rough registration, and fine registration. The goal of rough registration is to estimate a transformation that roughly aligns the two point clouds together. This transformation is taken as a starting point for the fine alignment, which will improve the alignment accuracy.

There is a lot of research available on improving point cloud registration algorithms, each presenting their own improvements to make it more accurate and less computationally intensive. For example, some research suggests the use of local point-pair features [9] to perform rough alignment while other research suggests the use of mass-centers and main axes of point clouds to perform this rough alignment [2; 6].

However, most of this research assumes that the point clouds to be aligned are similar to each other, which is not the case in this application. The source point cloud, which is acquired by the surgeon using an optical tracker, consists of roughly 50 points, while the target point cloud (the pre-operative model), which is acquired during a scan, can consist of roughly 5000 points.

This paper will cover how the HoloLens, together with its depth sensor, can be used to provide accurate patient alignment.

2 Related work

In the field of Computer Graphics, point cloud registration is a well-known problem which is being researched very actively. Many algorithms have been created, each improving on the algorithms already known.

Rusu et al. [4; 5] introduces the usage of Point Feature Histograms (PFH) for performing rough alignment of two point clouds. PFHs are used to describe the features of each point in the point cloud, based on its relation to other points in its k-neighborhood. Using these features increases the probability of finding a correct correspondence between two point clouds, and the obtained transformation can then be used as a starting point for the regular ICP algorithm. Tests conducted in their research show that using FPH for rough initial alignment significantly improves the accuracy of the ICP algorithm.

However, the computational complexity of calculating PFHs grows quadratically with the number of neighbors analyzed. This is why Rusu et al. [3] introduced Fast Point Feature Histograms (FPFH). This improved way of obtaining the PFH reduces the computational complexity significantly, resulting in a complexity that only grows linearly with the number of neighbors analyzed.

Alternatively, Makovetskii et al. [2] and Xin et al. [8] show a new algorithm for rough point cloud alignment based on the mass centers and main axes of given point clouds. The rough alignment is performed by translating the source point cloud so that its center of gravity matches that of the target point cloud. Then, the source point cloud gets rotated so that its main axis matches that of the target point cloud. This method allows for relatively fast rough alignment, as calculating these properties of the point clouds is less computationally intensive. The downside of this approach is that the algorithm does not consider the features of the point cloud. This means that the algorithm might be less accurate in situations where the point clouds to be aligned are not very similar to each other.

3 Research methodology

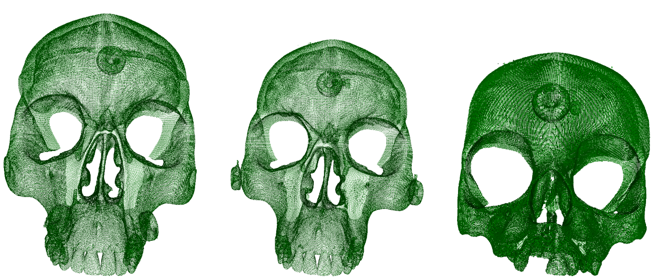
In order to find a good algorithm to perform patient alignment, different point cloud registration algorithms were tested. This section will cover which algorithms were tested, the data that were used to test them, and how the performance of the algorithms was evaluated.

3.1 Input data

To be able to properly evaluate the performance of the algorithms for this specific application, representative data should be supplied to the algorithms. In total, data from three different skulls were available for evaluation of the algorithms. For each of the skulls, the following data were available:

- A pre-operative model obtained with a scanner.
- A corresponding target point clouds obtained with an optical pointer, already aligned with the pre-operative model.

In Figure 1, the pre-operative models from the three skulls can be seen.



(a) Source point cloud of the first skull

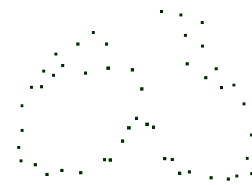
(b) Source point cloud of the second skull

(c) Source point cloud of the third skull

Figure 1: Different available skulls

In Figure 2, two corresponding point clouds can be seen. In this figure, it can also be seen that the target point cloud, obtained by the surgeon using an optical marker, is very sparse. Many point cloud registration algorithms would fail under these conditions, as 50 points is not enough to be able to extract features from it. In order to solve this issue, the depth sensor of the HoloLens is used to create a detailed point cloud.

(a) Source point cloud obtained using a scan, around 400.000 points



(b) Target point cloud obtained by a surgeon using an optical marker, around 50 points

Figure 2: Source and target point clouds to be aligned using point cloud registration algorithms

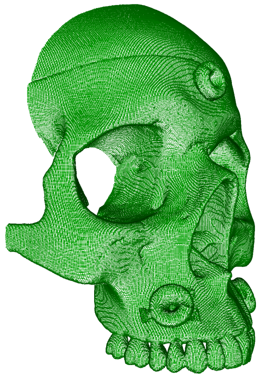
Simulated HoloLens depth sensor data

Unfortunately, during most of the project, a point cloud obtained by the HoloLens depth sensor was not available for testing. Therefore, these data were simulated by taking the pre-operative scan and adding noise, occlusion, and sparsity to it:

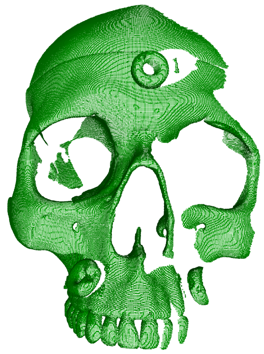
- In order to add noise to the point cloud, the amount of noise that should be added needs to be defined. Then, the maximum translation on each axis gets calculated by using the Pythagorean formula, with a being the maximum translation on each axis, and n being the amount of noise that should be added:

$$\begin{gathered} a^2 + a^2 + a^2 = n^2 \\ \Updownarrow \\ 3a^2 = n^2 \\ \Updownarrow \\ a^2 = \frac{n^2}{3} \\ \Updownarrow \\ a = \sqrt{\frac{n^2}{3}} \end{gathered}$$

- To add occlusion to the point cloud, the point cloud was transformed in a way that simulates looking at the face from an angle. Then, only the visible points were selected, the selection was inverted, and the selection was deleted. The points remaining in the point cloud are those visible from the viewing angle. In Figure 3, a point cloud with occlusion can be seen.
- In order to add sparsity to the simulated depth sensor data, a voxel-based downsampling method is used. A voxel grid ensures that the data are uniformly



(a) Source point cloud looked at from an angle



(b) Source point cloud with invisible points removed

Figure 3: Source point cloud with occlusion applied at 45 degrees

downsampled. In Figure 4, a downsampled point cloud can be seen.

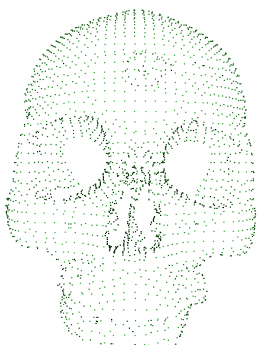


Figure 4: Downsampled point cloud using a voxel size of 4

Research conducted on the first generation Microsoft HoloLens concluded that the depth sensor has around 4.3mm of noise when capturing a 3D scene with the 'near' capturing mode [1]. Unfortunately, no similar research on the second generation Microsoft HoloLens' depth sensor was found. However, from reading different articles online, it seems like the depth sensor hardware between the two different versions has not changed.

In Figure 5, a simulation of a point cloud obtained by the HoloLens' depth sensor is shown.

Actual HoloLens depth sensor data

During the last phase of the project, actual HoloLens depth sensor data were made available for research. In Figure 6, raw depth-sensor data from the HoloLens can be seen. These data were obtained by using the HoloLens "short-throw" depth sensor mode.

It very quickly becomes clear that using these raw depth sensor data will require heavy preprocessing in order to be able to isolate the skull from the rest of the scene. This is however outside of the scope of this research. Tests using the real HoloLens depth sensor data will be conducted by manually extracting the skull from the 3D scene.

Figure 5: Simulated front-view depth sensor point cloud with random noise added

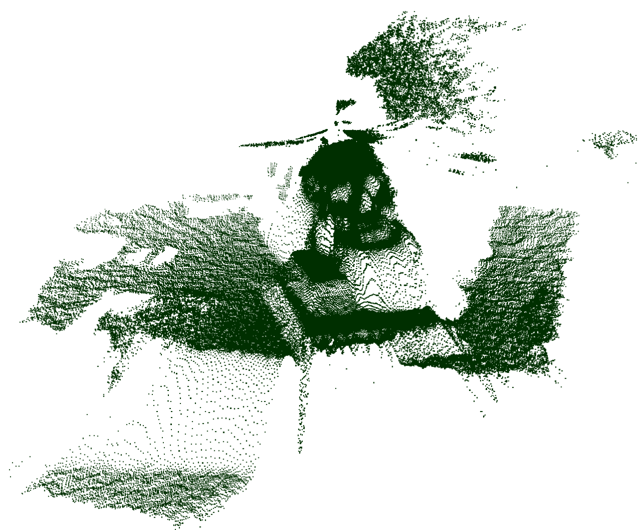


Figure 6: Raw HoloLens depth sensor data

3.2 Registration algorithms

Point cloud registration consists of two steps. These steps can be defined as follows:

1. **Rough registration**
   The goal of this first step is to find a rough transformation that roughly transforms the source point cloud to the target point cloud.
2. **Precise registration**
   In the second step, an exact transformation should be found to precisely align the source point cloud with the target point cloud.

The first step plays a critical role in the alignment process, as a good rough estimation can prevent the ICP algorithm from getting stuck in a local minimum.

Rough registration

Two main methods for rough registration were tested during this research.

The first algorithm tested is based on the fast point feature histograms (FPFH) [3]. As mentioned in Section 2, point feature histograms are used to describe the features of a certain point in the point cloud by looking at the properties of the points in its neighborhood. Then, the rough registration can be obtained from finding matching features between the two point clouds.

The second algorithm is based on center of gravity and principal component analysis (PCA) [6]. In this algorithm, the center of gravity of both point clouds is computed. Additionally, the algorithm calculates a quaternion used for rotating the source point cloud so that its main direction matches the main direction of the target point cloud. This quaternion is derived from the eigenvector corresponding to the highest eigenvalue of a matrix obtained form the covariance matrices of the two point clouds.

A third approach to rough point cloud registration is performing manual point selection and registration. In this scenario, the surgeon will link certain points of the pre-operative model with points on the patient obtained by an optical marker. The downside of this approach is that the accuracy is highly dependent on the accuracy of the surgeon, and the accuracy of the optical trackers in the HoloLens, which track the optical marker in the 3D space around the patient.

Precise registration

For local registration, the iterative closest point algorithm (ICP) is used [7]. As mentioned in Section 1, this algorithm works by finding the closest point to the source point cloud in the target point cloud, and form a point pair. It then performs a rotation and translation to minimize the distances between the points in the point pairs. This is performed iteratively, until the algorithm converges or a set maximum amount of iterations is met.

3.3 Initial testing setup

To create a test scenario, the point clouds must first be misaligned. A starting position was defined randomly, where a random translation and rotation were applied to the source point cloud. This transformation is now used as a starting point for all tests conducted, so that the initial position does not influence the performance of the algorithms across different tests. In Figure 7, the initial position can be seen.

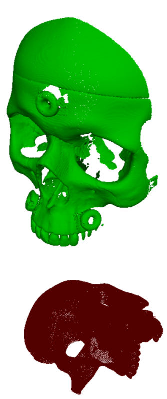


Figure 7: Initial misalignment of the source and target point clouds. The green point cloud is the target, and the red point cloud is the source.

3.4 Performance analysis

The most important performance metric for these algorithms is alignment accuracy. This accuracy can be evaluated by calculating the mean square error (MSE) of the distances between points in the source point cloud and their corresponding points in the target point cloud.

First, reference points should be created between the source and target point clouds. This can be achieved by taking all the points from the smallest point cloud (in this specific case, the point cloud obtained by the surgeon using an optical marker) and adding them to the larger point cloud (in this case the pre-operative model). This is done to ensure that the MSE will be 0 when no transformation is applied to the source point cloud.

After alignment, distances between point clouds are measured by, for each point in the smallest point cloud, finding the distance to its corresponding point in the largest point cloud.

The MSE is calculated by taking the previously calculated distances, adding their squares and then dividing the result by the amount of points in the smallest point cloud.

The data provided are scaled in such a way that the MSE is expressed in millimeters. This makes it very easy to give an MSE a contextual meaning.

4 Results

In this section, the results of the tests will be shown. Each result shows the alignment accuracy of the rough alignment algorithm alone, as well as in combination with the ICP algorithm used for fine alignment.

4.1 Reference results

In order to be able to discuss the results, some reference results should be obtained first. In Table 1, the results can be seen from an alignment where both the source and the target data are the same point clouds, in this case the pre-operative model. This means that no noise, sparsity, or occlusion has been added to the data. Figure 8 visualizes this alignment.

FPFH	FPFH and ICP	PCA	PCA and ICP
0.0	0.0	5.577723	0.0

Table 1: Error in mm after alignment with the different algorithms.

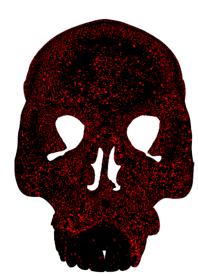


Figure 8: Visualization of a perfect alignment. Red dots are part of the source point cloud, black dots are the target point cloud.

It can be seen that the rough alignment performed by the PCA-based algorithm performs worse than the rough alignment performed by the FPFH-based algorithm, which manages to have a perfect alignment. However, after refining the rough alignment with the ICP algorithm, both registration techniques manage to perform a perfect registration.

4.2 Noise tests

In Table 2, the alignment accuracy of the different algorithms can be seen. The amount of noise used to simulate depth sensor data, as described in Section 3.1, can also be seen.

Noise	FPFH	FPFH and ICP	PCA	PCA and ICP
0.0	0.71432	1.3434	5.57772	1.3434
0.5	0.98729	1.3402	5.56782	1.34949
1.0	0.59815	1.47238	5.59073	1.32688
1.5	5.83117	1.37534	5.57377	1.29048
2.0	54.86322	1.1218	5.53751	1.26212
2.5	112.78644	1.3851	5.54046	1.22339
3.0	47.90795	0.87403	5.61039	1.20077
3.5	144.31412	1.81052	5.53009	1.16468
4.0	6143.1341	6054.8097	5.68467	1.1466
4.5	56.90995	0.88746	5.56207	1.13068
5.0	440.68531	322.11505	5.52201	1.12898
5.5	2724.69633	2505.25479	5.59876	1.1059
6.0	126.87457	5.74996	5.51097	1.1059

Table 2: Error in mm after alignment with noisy data. Results indicated in red represent failed alignments.

It can be seen that both the FPFH-based algorithm and PCA-based algorithm are quite resilient to noisy data when combined with the ICP fine alignment. However, it appears that the FPFH-based rough alignment has quite unreliable results in comparison to the PCA-based rough registration. Additionally, at high noise levels (4mm and higher) the FPFH-based algorithm struggles to deliver consistent results. However, in all other cases, the FPFH-based registration outperforms the PCA-based registration.

4.3 Occlusion tests

In Table 3, the alignment accuracy of the different algorithms can be seen. The viewing angle used to generate the occluded data, as described in Section 3.1, can also be seen.

Angle	FPFH	FPFH and ICP	PCA	PCA and ICP
0°	3.53468	1.10841	40.02235	1.13809
5°	3.30004	1.1129	2554.49362	2460.32122
10°	0.55443	1.08763	58.22161	1.15165
15°	3.83333	1.06005	2538.79867	2488.61779
20°	4.95372	1.03576	89.39094	1.10652
25°	10.17274	1.00146	2691.75543	2387.4065
30°	4.62448	1.00659	84.56393	1.07644
35°	7.44704	1.04795	706.87402	405.61437
45°	2.89342	1.10348	687.89514	318.33838

Table 3: Error in mm after alignment with different occlusion amounts. Results indicated in red represent failed alignments.

It can be seen that the PCA rough registration algorithm is very sensitive to occlusion, up to the point where the ICP algorithm is unable to find the correct alignment. In comparison, the FPFH-based rough registration remains very consistent in its alignments, allowing the ICP algorithm to perform very accurate alignments.

4.4 Sparsity tests

In Table 4, the alignment accuracy of the different algorithms can be seen. The voxel size is also given, which is used to downsample the simulated depth sensor data as described in Section 3.1. Additionally, the amounts of points remaining in the depth sensor data are shown.

Voxel size	Number of points	FPFH	FPFH and ICP	PCA	PCA and ICP
1	29613	3.19233	1.11036	2577.85716	2672.05739
2	8263	1.51687	1.12713	2640.62596	2439.06767
3	3873	2.03372	1.18044	370.2956	3.22495
4	2283	519.77833	382.83185	2633.17455	2664.65124
5	1449	150.91408	1.00298	2626.40857	2646.13936
6	1035	186.95435	1.83306	3144.23206	2467.10342
7	784	1.09607	1.23063	2782.62606	2507.81229
8	600	7845.36542	7887.99673	3924.82102	3815.58808
9	472	109.42272	1.09453	2188.59803	2087.25121
10	384	2071.38652	2109.04632	2748.54431	2809.09537

Table 4: Error in mm after alignment with different sparsities in the simulated depth sensor data. Results indicated in red represent failed alignments.

From these results, it can be seen that the PCA algorithm is very sensitive to sparse data. The FPFH-based alignment mostly remains quite accurate after the ICP algorithm has been run, but the rough alignment becomes quite unreliable. The ICP algorithm also starts to have trouble when the performance of the FPFH-based rough aligment degrades.

4.5 Manual point selection

The manual point selection (MPS) algorithm has also been tested, but has not been included in the previous tests as it works quite differently. The rough alignment in this point cloud registration approach is done manually. Two different tests were conducted on this algorithm:

- **Error test**
  This test evaluates how resilient the MPS algorithm is to inaccuracy of the surgeon performing the manual point selection.
- **Sparsity test**
  This test performs MPS with a varying number of manually linked points in order to see how many points are needed for reliable registration.

In Table 5, the accuracy of the manual point selection algorithm can be seen with different levels of inaccuracy when linking the points.

At low inaccuracy levels, it can be seen that the ICP algorithm actually worsens the accuracy of the manual point selection algorithm. However, it also ensures that the results stay reliable when the manual point selection performs a less accurate rough alignment, as can be seen at higher error levels.

In Table 6, the accuracy of the manual point selection algorithm can be seen with different amounts of points that are manually linked.

It can be seen that the MPS algorithm perfectly aligns the source point cloud with the target point cloud when there are

Error	MPS	MPS and ICP
0.0	0.0	1.1072
0.5	0.01736	1.1072
1.0	0.04668	1.1072
1.5	0.15065	1.1072
2.0	0.22021	1.1072
2.5	76.01306	1.11403
3.0	0.65925	1.10254
3.5	0.39172	1.1072
4.0	0.85834	1.1072
4.5	0.60797	1.1072
5.0	3.03763	1.10911
5.5	1.34152	1.10841
6.0	2.52741	1.1072

Table 5: Error in mm after alignment with different error levels (in mm) in the manually linked points.

Points	MPS	MPS and ICP
1	45046.86625	45046.86625
2	6603.43749	5412.88978
3	7509.28353	8600.74887
4	0.0	1.1072
5	0.0	1.1072
6	0.0	1.1072
7	0.0	1.1072
8	0.0	1.1072
9	0.0	1.1072
10	0.0	1.1072

Table 6: Error in mm after alignment with different amounts of linked points. Results indicated in red represent failed alignments.

4 or more points manually linked. It is however important to keep in mind that this test assumes perfect accuracy of the surgeon selecting and matching these points. It can also be seen that the ICP algorithm is again worsening the results of the MPS algorithm, always bringing the accuracy to around 1mm.

4.6 Actual depth sensor data

Because of the limitations of the depth sensor data available, only a visual evaluation can be made. An exact error from perfect alignment cannot be calculated, because a ground truth state is not available. Additionally, the depth sensor data used for these tests are manually cleaned from the raw depth sensor data, as described in Section 3.1.

In Figure 9, a FPFH-based rough alignment of the pre-operative model and the depth sensor data can be seen.

From visual inspection, it looks like the algorithm has been able to perform a very rough point cloud registration between the two point clouds. However, the FPFH-based algorithm is not consistent in performing this alignment. Many different attempts need to be made in order to get an alignment this accurate.

The PCA-based rough alignment was also tested on the actual depth sensor data. In Figure 10, the rough alignment of the PCA-based rough registration algorithm can be seen.

Figure 9: FPFH rough alignment with downsampled source and depth sensor data. The black point cloud is the depth sensor data, the red point cloud is the pre-operative model.

Figure 10: PCA rough alignment on real depth sensor data. The black point cloud is the depth sensor data, the red point cloud is the pre-operative model.

It can be concluded from visual inspection of this result that the alignment is only partially correct. Only the center of gravity is matched correctly, but the main axis of the point clouds has not been aligned properly.

5 Discussion

When only considering the rough point cloud registration algorithms (FPFH and PCA, without ICP), it can be seen that the FPFH-based rough registration method outperforms the PCA-based rough registration method in almost every test scenario. It is only in the noise test in Section 4.2 that the PCA-based rough registration had a more consistent alignment, albeit sometimes less accurate.

When looking at the results of the rough registration algorithms followed by the ICP fine registration algorithm, it is interesting to see how well ICP can handle a relatively bad rough initial alignment. In every case where the rough registration algorithm has an accuracy of 200 mm or less, the ICP algorithm was able to convert this rough alignment into very precise alignments. However, when the rough registration is less accurate than, the ICP algorithm fails to improve the alignment. This shows the importance of having a robust rough point cloud alignment algorithm before applying the ICP algorithm.

The poor performance of the PCA rough registration in the occlusion and sparsity tests could be due to the fact that the algorithm does not look at the features of the point clouds. It only considers their center of gravity and their main direction, which are both variables that can easily change when the layout of the point cloud changes.

Considering the manual point selection algorithm, it can be seen that this algorithm is by far the most consistent. Depending on the accuracy of the linked points, it can even be beneficial to use this registration method without the ICP algorithm. Doing this would even make using a detailed point cloud obtained with the HoloLens' depth sensor unnecessary.

When looking at the attempted alignment with actual HoloLens depth sensor data, it appears that the FPFH-based algorithm has made a good attempt at roughly aligning the datasets. The PCA-based algorithm on the other hand struggles to align the data properly. This is probably due to the very different structure of the two point clouds, resulting in the main axes of these point clouds to be different, or in this case even opposite.

6 Responsible research

In order to fulfil the concept of responsible research, the findings in this report should be critically analysed. This section will cover the efforts done to ensure that this research is as transparent as possible.

Firstly, it is important to discuss all obtained results. During this research, no obtained results were left out or dismissed from the report.

Another point of possible concern is the source of the point cloud data. All the data used during this research are fake, phantom data. The pre-operative models, from which all other point clouds are derived, do not belong to any known patient.

In order to make the experiments reproducible so that the obtained results can be verified, the source code of the experiments was made available in a public GitHub repository [1]. This gives anyone interested in this research the opportunity to easily run the tests themselves, and tinker with the parameters of the algorithms to see how they behave. The data used to perform the tests werealso included in this repository.

Lastly, the accuracy of the results could be questioned. The input data are partially generated using randomness. For example, the noise that gets added to the input data are randomly generated. This also makes that the data being used for testing are always slightly different, causing a result that is not fully deterministic. In order to provide the results as accurately as possible, the tests have been run multiple times to ensure that, for example, a bad outcome is not a fluke.

7 Conclusion and future work

This paper discussed performing patient alignment when using the Microsoft HoloLens for surgical navigation. Three algorithmic approaches were researched: rough alignment using fast point feature histograms followed by precise alignment using the ICP algorithm, rough alignment using principal component analysis followed by the ICP algorithm, and finally a manual registration approach.

As can be derived from the results described in Section 4, algorithmic approaches can quite accurately align a pre-operative model with a patient. The MSE of successful alignments generally lays in the neighborhood of 1mm. However, the performance is highly dependent on the quality of the input data. For example, more occlusion means less accuracy, so the cleaner the input data the better.

It can be concluded that the FPFH-based registration algorithm generally outperforms the PCA-based algorithm. Especially in scenarios where the quality of the input data starts to degrade (noise, sparsity, occlusion).

The best accuracy however was obtained by performing manual point selection. In these cases, the ICP algorithm even worsened the results obtained by the manual point selection algorithm. The downside of this approach is that it requires the surgeon to manually align points in order to perform patient alignment. This takes time, and the accuracy of the algorithm is directly dependent on the accuracy of the surgeon performing the point selection.

Recommendations for future research include researching algorithmic approaches for separating the patient from the rest of the 3D scene when using the HoloLens' depth camera. Next to that, research should be conducted on proper HoloLens depth sensor data, instead of on simulations of them, to ensure representative results.

It is however nice to see that algorithmic approaches for performing patient alignment using the Microsoft HoloLens are possible, bringing the HoloLens one step closer to being used in the operating room.

References

[1] Patrick Hübner, Kate Clintworth, Qingyi Liu, Martin Weinmann, and Sven Wursthorn. Evaluation of HoloLens Tracking and Depth Sensing for Indoor Mapping Applications. *Sensors (Basel, Switzerland)*, 20(4):1021, February 2020.

[2] Artyom Makovetskii, Sergei Voronin, Vitaly Kober, and Alexei Voronin. An algorithm for rough alignment of point clouds in three-dimensional space. In *2020 International Conference on Information Technology and Nanotechnology (ITNT)*, pages 1–4, May 2020.

[3] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast Point Feature Histograms (FPFH) for 3D registration. In *2009 IEEE International Conference on Robotics and Automation*, pages 3212–3217, May 2009. ISSN: 1050-4729.

[4] Radu Bogdan Rusu, Nico Blodow, Zoltan Csaba Marton, and Michael Beetz. Aligning point cloud views using persistent feature histograms. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3384–3391, September 2008. ISSN: 2153-0866.

[5] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, and Michael Beetz. Learning informative point classes

[1] https://github.com/maartenweyns/holonav-patient-alignment

for the acquisition of object model maps. In *Robotics and Vision 2008 10th International Conference on Control, Automation*, pages 643–650, December 2008.

[6] Heqiang Tian, Xiaoqing Dang, Jihu Wang, and Dongmei Wu. Registration method for three-dimensional point cloud in rough and fine registrations based on principal component analysis and iterative closest point algorithm. *Traitement du Signal*, 2017.

[7] Fang Wang and Zijian Zhao. A survey of iterative closest point algorithm. In *2017 Chinese Automation Congress (CAC)*, pages 4395–4399, October 2017.

[8] Wei Xin and Jiexin Pu. An Improved ICP Algorithm for Point Cloud Registration. In *2010 International Conference on Computational and Information Sciences*, pages 565–568, December 2010.

[9] Xiaofeng Yue, Zeyuan Liu, Juan Zhu, Xueliang Gao, Baojin Yang, and Yunsheng Tian. Coarse-fine point cloud registration based on local point-pair features and the iterative closest point algorithm. *Applied Intelligence*, February 2022.