Utilising Deep Learning Models for the Surface Registration Problem in HoloNav

Gundogan Alp Cicimen
Supervisor(s): Prof. Ricardo Guerra Marroquim, Abdullah Thabit
EEMCS, Delft University of Technology, The Netherlands
22-6-2022

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

Surface Registration is a registration problem that handles the registration of two similar surfaces. In most research that utilises Deep Learning (DL) models to handle surface registration two theories are investigated; the first being whether surfaces sampled from the same origin can be registered together, and the second theory being whether the models can register Point Clouds with low overlapping data for utilisation in Simultaneous Localisation and Mapping (SLAM) applications. However, the surface registration to be utilised in the HoloNav Augmented Reality (AR) navigation system will utilise Point Clouds sampled from different origins with a high overlap ratio. This research, therefore, aims to determine the viability of DL methods for surface registration in HoloNav data. To determine the viability, rotation and translation errors in the match were used, with the aforementioned metrics later being evaluated manually with the utilisation of a visualiser. The results indicate that the models can generalise on the navigator data for an initial Euler angle difference of 45 degrees, but due to the difference in sampling density on the utilised point clouds can not provide accurate matches. Therefore, the utilisation of DL models can be considered to be viable if the navigator data has a sampling density similar to the pre-operative model.

1 Introduction

Surgical navigation is a technology that aids surgeons in medical procedures. This technology is utilised in order to determine the positioning of the surgical instruments with respect to the surgical site of the patient. However, traditional surgical navigation tools have certain usage challenges that hinder the hand-eye coordination of the surgeon. According to Benmahdjoub et al.[1], conventional navigation systems have two main limitations, which are "the repeated switch of attention between the intervention area and the 2D display" and "the coordination between the hands and the images presented on the screen". Therefore, a system that enables the surgeon to focus on the display data and the surgical system simultaneously would be able to improve the coordination of the surgeon and possibly simplify the surgical procedure.

In order to improve surgeon coordination, HoloNav aims to utilise the Microsoft HoloLens, an Augmented Reality (AR) system, as a surgical navigation system. Such a system would be able to map virtual data onto real-life applications and could improve the coordination of the surgeon by decreasing the need to focus on multiple locations.

However, for the system to function properly certain solutions to existing problems need to be addressed. One such problem to be addressed is the problem of registration. The registration problem is the computational problem of mapping different input data together in order to align them. The solution to the registration problem is crucial in AR systems, as the system requires information about the location to map virtual data. As a solution to the registration problem, the official documentation for HoloLens[1] states that the system utilises landmarks, or features, in the environment to locate itself in a space. However, a surgical environment may not be suitable for landmark-based registration, and therefore a solution for the registration problem without landmarks may provide more utility value for HoloNav.

To this end, surface-based registration has been considered as a viable alternative. In contrast to Landmark based registration, surface-based registration utilises matching from sampled points from a surface originating from multiple sources. In order to match the sampled points, an algorithmic approach such as the Iterative Closest Point (ICP) algorithm has been utilised in the past[2]. However, a more recent approach to the surface registration problem has been by utilising Deep Learning (DL) models, with the work of Qi and Su[3] demonstrating the capability of DL models in point cloud data segmentation. Therefore, a DL model that can discern specific features from a point cloud can be modified to match detected features robustly, providing an alternative to the ICP algorithm.

Therefore the goal of this study is to investigate the feasibility of using Deep Learning Algorithms for implementing Point-Cloud Registration and Matching solutions. In order to explore the feasibility of such systems, the question this study aims to answer is "Can Deep-Learning methods improve the patient-alignment registration for the HoloLens?". To further analyse this question, several sub-questions have been proposed:

- "What kind of Deep Learning models could be trained for usage in patient-alignment registration?"
- "How would Deep Learning models be suitable for patient-alignment registration perform in terms of alignment accuracy on a test set?"
- "How would Deep Learning models be suitable for patient-alignment registration perform in terms of time for evaluation?"
- "Why would Deep-Learning based approaches be used for patient-alignment registration as opposed to using traditional algorithmic-based approaches?"

This report has been structured accordingly to answer the provided questions. Chapter 2 provides information about past work related to this research with justification for the research outlined by this paper. Chapter 3 contains the methodology of the research, including variable analysis and explanations of the variables in detail. Chapter 4 contains instructions on how to replicate the conducted experiments, while Chapter 5 contains the results of conducted experiments. Chapter 6 contains an analysis of the research, outlining certain ethical issues related to how the experiments were conducted. To conclude, Chapters 7 and 8 will contain discussions of the results with a conclusion aiming to answer the proposed questions with recommendations for further research.

[1] https://docs.microsoft.com/en-us/hololens/hololens-environment-considerations

2 Related Work

In this section existing work related to the utilisation of DL models for utilisation in the registration problem in HoloNav will be provided and discussed. Chapter 2.1 provides existing research on Augmented Reality in surgical navigation systems, and discusses the issues with the aforementioned work that this research aims to answer. Chapter 2.2 provides an application for Deep Learning in registration problems, and explains the main difference between current applications of DL with surface registration in comparison to the considered utilisation in HoloNav.

2.1 Augmented Reality and Surgical Navigation

The utilisation of Augmented Reality (AR) for surgical navigation has been investigated in some works, and the results indicate that AR can be a viable technology for surgical environments. To give an example, the report of Incekara et al.[4] claims that surgeons reported "improved ergonomics and focus" from the utilisation of the Microsoft HoloLens, while Chen et al.[5] has reported that the accuracy of AR systems was sufficient to meet clinical requirements. This would mean that AR-based systems are able to provide similar results to display-based navigation systems while improving the performance of the surgeon. As a result, AR systems can be considered a likely candidate for utilisation as a surgical navigation technology.

However, AR-based navigation systems currently have certain drawbacks that might need to be resolved for them to be more viable. One of the drawbacks of AR systems that this research aims to find a solution for is the registration problem. Prior research in some AR systems handles the registration of the system by utilising methods that may not be viable in a surgical environment. For example, Incekara et al.[4] has reported that their registration was performed manually, and Chen et al.[5] have utilised fiducial points for landmark-based registration. While both reports employ viable methods for most surgeries, another approach may be necessary for certain operations. Incekara's method, while being viable for purposes of experimentation and non-life-threatening situations, may prove unsuitable for life-threatening situations where time is of the essence. By contrast, while Chen's method of Landmark-based registration is handled automatically, in certain surgical operations it may not be possible to utilise fiducial points as markers. In summary, to be able to improve the performance of AR-based navigation systems it would be viable to utilise a registration method that can handle registration independent of human control and one that does not utilise fiducial marking on the surgical site.

In summary, the existing work of Incekara et al. and Chen et al. demonstrates the advantage of AR systems in improving the hand-eye coordination of the surgeons during operations. However, in both aforementioned works, the utilised registration systems have certain drawbacks in several situations, indicating that a different approach to the registration problem that minimises the shortcomings of existing work could be investigated for use in HoloNav.

2.2 Utilisation of Deep Learning for Registration

While research has been conducted in determining the viability of utilising DL methods for surface-based registration in AR systems, most of them utilise solutions for Simultaneous Localisation and Mapping (SLAM). For example, Arnold et al.[6] utilise the KITTI dataset [2] to register partially overlapping point clouds, while Gao et al.[7] investigate the utilisation of DL models for solving SLAM for AR systems in particular. Both systems, by utilising partially overlapping data are able to demonstrate that DL models can successfully handle registration for the SLAM problem.

However, an important feature of the registration system of HoloNav is the fact that registration is to be handled by utilising a handheld navigator. This feature implies that in contrast to SLAM applications the inputs of HoloNav are guaranteed to have a high overlap. However, a caveat of acquired navigator data is that unlike sensor data utilised in SLAM the navigator data is sparse compared to the pre-operative model. Therefore, it is much more important for the model to generalise on input data in contrast to matching specific regions of low-overlapping data. In summary, the utilisation of DL models for HoloNav differs from past research in the utilisation of AR systems by utilising point clouds with uneven density to determine the viability of utilising navigator data for registration.

3 Methodology

In this section the utilised variables will be discussed and explained in detail. The variables will first be analysed as independent, dependent and controlled variables respectively, and subsequent chapters will analyse each mentioned variable type in detail.

3.1 Analysis of Variables

To answer the research questions outlined in the previous subchapter, the measurements will be conducted using independent, dependent and controlled variables. The system will be set up using one controlled variable first, and subsequent tests may be performed using different datasets, with each test being recorded separately. By utilising these variables, the experiment can be easily replicated since the user would be aware of which models to test, which variable to keep static and which values to measure.

Independent Variables

In the experiments to be conducted the primary independent variable will be the Deep Learning models, with the only exception of one algorithm-based approach to be compared. The DL models considered for utilisation are RPMNet by Yew and Lee[8], Overlap PREDATOR by Huang et al.[9], MS-SVConv by Horache et al.[10], and PCRNet by Sarode et al.[11]. From the considered models, RPMNet and PREDATOR have been chosen for evaluation for this research. For comparison with an ICP algorithm, the algorithms utilised in research conducted by Weyns [12] were used. The main justification behind utilising models and an ICP algorithm is

[2]http://www.cvlibs.net/datasets/kitti/

that the input dataset is sparse to be utilised as an independent variable, and that the experiment concerns itself with how Deep Learning models would perform in this situation. Therefore, the independent variables for this experiment are the models to be trained and evaluated.

Dependent Variables

The dependent variables to be measured in the experiments are primarily the evaluation metrics provided by the tested models and the duration of each model in evaluating the models. For evaluation during testing on HoloNav data the primary metrics used for the results are the rotation and translation errors provided by the models. The metrics utilised for evaluating the rotation and translation errors of the predicted matches are calculated with the equations

$$Error(Rot) = \angle R_{pred} R_{init} \tag{1}$$

$$Error(Trans) = ||T_{pred} + T_{init}||_2 \tag{2}$$

where Equation 1 calculates the absolute rotation error by calculating the resulting angle from applying the initial transform R_{init} with the predicted transform R_{pred}; and Equation 2 calculates the translation error by calculating the norm of the vector resulting from the sum of the initial transform vector T_{init} with the predicted transform vector T_{pred}. The utilised equations for the aforementioned metrics are identical to the isotropic metrics utilised by Yew and Lee[8] and are claimed to be isotropic. Therefore, these metrics calculate the error independent of the transformations on each axes and consequently provide an ”absolute” error metric.

In order to utilise the aforementioned metrics, the models require information about the metrics applied before evaluation, which is achieved by applying the rotations and translations as part of the evaluation process. Therefore, in order for the error metrics to be valid, the models require the initial transformations that were applied. However, in unregistered data said transformations are unknown. Therefore, for evaluation, the experiments utilise a secondary metric in the form of a modified Chamfer Distance metric, which has been defined by Yew and Lee[8] as:

$$\bar{CD}(\mathbf{X}, \mathbf{Y}) = \frac{1}{|\mathbf{X}|} \sum_{\mathbf{x}\in\mathbf{X}} \min_{\mathbf{y}\in\mathbf{Y}_{\text{clean}}} ||\mathbf{x}-\mathbf{y}||^2 + \frac{1}{|\mathbf{Y}|} \sum_{\mathbf{y}\in\mathbf{Y}} \min_{\mathbf{x}\in\mathbf{X}_{\text{clean}}} ||\mathbf{x}-\mathbf{y}||^2 \tag{3}$$

where X denotes the transformed source point cloud, Y denotes the reference point cloud; while X_{clean} and Y_{clean} denote the point clouds without any noise applied to them respectively. The function returns the sum of the squared distances between nearest-neighbour correspondences of the clean and modified versions of the point clouds.

Since the navigation data is contained within the subspace of the model data, the Chamfer Distance used by Yew and Lee has been modified further with the removal of the correspondence matching of the source point cloud. With this modification, the match metrics focus on the correspondences of the navigation data, leading to better matches. With the reduction, the metric is rewritten as:

$$\bar{CD}(\mathbf{X}, \mathbf{Y}) = \frac{1}{|\mathbf{Y}|} \sum_{\mathbf{y}\in\mathbf{Y}} \min_{\mathbf{x}\in\mathbf{X}} ||\mathbf{x}-\mathbf{y}||^2 \tag{4}$$

which is the metric used for the evaluation of the navigator data.

Controlled Variable

The controlled variable in the measurements is the datasets that the models are trained and tested on. The initial experiments can be executed using a modified version of the ModelNet40 dataset from Princeton University [3]. Other datasets can also be used, but some DL models such as RPMNet[8] may have to be configured to be able to evaluate such sets. For later evaluations, the models have been trained and tested on input acquired from the HoloLens Navigator. If other datasets are to be used, either for standalone experiments or to complement existing results, it would be strongly advised to keep said results separate and to mention incompatible models with said datasets. By utilising ModelNet the user can evaluate the viability of a model without requiring the modification of the utilised models. To conclude, for the experiments the datasets for training and testing are utilised as the controlled variable, with the HoloNav data being the main controlled variable. However, other datasets can be utilised for the initial evaluation of the models if desired. Therefore, the controlled variable for this experiment has been multiple different datasets, with the evaluation results being kept separate as advised.

3.2 Analysis of Existing Deep Learning Models

RPMNet

RPMNet is a Deep-Learning-based approach for Point-Cloud Registration by Yew and Lee[8]. The model implements Point Cloud Registration by utilising two Multilayer Perceptron (MLP) networks that extract matching features and predicts values for the outlier parameter α and the annealing schedule β for calculating the match matrix. For feature extraction, the model utilises the neighbouring points at a configurable distance to acquire sources of information from the model. Once the model estimates the ”soft” assignments, the model estimates the rigid transformation by generating a match matrix and applying singular value decomposition (SVD) to factorise the matrix into its constituent transformation matrices. The model can be reevaluated with the acquired transform metrics in the form of a new iteration by supplying the previous iteration results. The authors state that an accurate match is obtained for ModelNet in 5 iterations.

The fact that the model is able to evaluate parameters for outlier parameters without any major parameter tuning being required may mean that the model may be able to generalise on different datasets such as the HoloNav input. The model also demonstrates the ability to match models with non-corresponding points, which also makes it a suitable candidate for evaluation.

[3]https://modelnet.cs.princeton.edu

Overlap PREDATOR

Overlap PREDATOR is a Deep-Learning model developed by Huang et al.[9]. The model utilises Graph Neural Networks (GNN) to match sampled voxels of uniform density from the source and reference point clouds. In order to utilise a GNN, the data is encoded using a k-nearest neighbour search algorithm. The encoded blocks are evaluated using the GNN to determine the contextual information of the "superpoint" evaluated from the block. The model then utilises a MLP to establish correspondences between the superpoints. Finally, the blocks are decoded with the established correspondences to their original format, and the RANSAC registration algorithm is applied globally based on the established "hard" correspondences. The model does not perform registration in an iterative manner and excels at matching models with low overlap regions while performing well with fully overlapping datasets such as ModelNet as well.

The model being able to perform matches in models with low overlap regions by establishing neighbouring correspondences means that it can be suitable to utilise in registering regional correspondences. However, the model contains more parameters in comparison to RPMNet and may require further preprocessing and/or parameter adjustment to perform accurate registration.

3.3 Analysis of Utilised Input Data

ModelNet40

ModelNet is a 3D Point Cloud classification dataset created by Song et al.[13]. It consists of data from 40 different categories that can be utilised for classification, segmentation or registration problems. For the experiments, the dataset utilised is a modified version of the ModelNet40 by Qi et al.[3]. The modified dataset consists of normalised models contained within a unit sphere at the origin. The models are sampled equally in every region, and most models demonstrate symmetry. Due to the initial role of ModelNet being a classification dataset, the data differ between categories. Therefore, due to major differences between categories a model utilising ModelNet for registration can demonstrate its ability to generalise on data.

The models utilised in this research evaluate on ModelNet by splitting the "raw" data into source and target point clouds and applying transformations into the source cloud, with sampling and noise generation applied to both point clouds respectively. This means that the models utilise the same origin and therefore can be matched more accurately due to both the source and the target clouds containing the same feature regions.

HoloNav Scan Input and Navigator Data

The input acquired from the HoloNav consists of two types of data: the source and the target. The source data consists of three 3D scans of human skulls with fiducial markers attached. The source meshes have a high-quality version and a version with a lower amount of vertices. The models have to be converted into point clouds in order to apply registration on them. Points can be sampled by sampling uniformly from the mesh surface, or by utilising the vertices. Due to the size of the model, the source point cloud was sampled from the vertex positions in order to achieve a faster evaluation time.

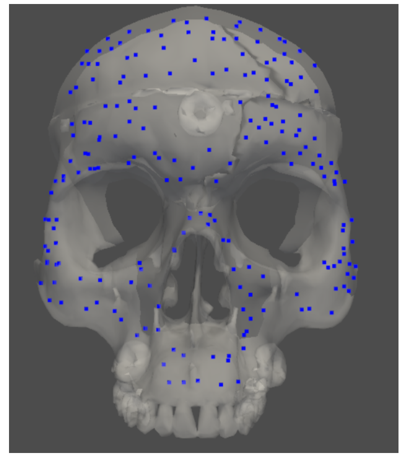

Figure 1: Visual example of a correctly registered navigator output with the pre-operative (Pre-Op) model. The Pre-Op model is Model 1, and the navigator data has the index of 5.

The target point cloud data initially consisted of three categories, and during the timeline of this research, an additional category was added. The first category, the preoperative models, consists of sampled target point clouds from the preoperative model. The second category, the raw point clouds, consists of point clouds acquired from the navigator that has not been registered. The third category is the registered navigator data with landmark registration. The final category consists of the depth sensor data that was obtained from the depth sensor data of the HoloLens. In summary, the target point cloud data consists of four different categories, with three different sampling methods applied to them. An example of correct registration of the model can be found in Figure 1, and every visual output of the ideal registration of the navigator data can be found in Appendix A.

Due to certain features of the provided data, as well as the DL models, the registered navigator data was deemed most fit for training and evaluating the DL models. The main reason for this decision lies in the fact that the models require initially registered data to create accurate evaluation metrics, and as a consequence generate accurate loss metrics to be able to fit the data properly. However, in order to compare to an ICP-based approach, the methodology outlined by Weyns [12] was used, with minor differences. For the aforementioned comparison, all pre-operative models were used, in contrast to the first pre-operative model utilised by Weyns. For the evaluation metrics, due to differences in implementation, the reduced Chamfer Metric was utilised, as outlined in Equation 4. Due to the evaluation results of the models demonstrating robustness to noise, as well as tolerance to rotational errors up to 45 degrees, the comparison metrics were utilised as differing voxel sizes in order to evaluate the generalisation abilities of the DL models.

4 Experimental Setup and Procedure

4.1 Experiment Setup

For a user to replicate the results of the experiments, hardware and software limitations have to be considered.

Hardware Requirements

The DL models utilised for evaluation require first and foremost a Graphics Processing Unit (GPU) with the main constraint of memory size. For training the models with HoloNav data an NVIDIA Tesla V100S [4] with an allocated memory of 32 gigabytes were used. However, for the experiments, a minimum GPU memory of 10 gigabytes would be satisfactory. For obtaining evaluation metrics on HoloNav data it may be possible to utilise less memory, however, the data may require downsampling on points to be evaluated on low memory. For both training and testing, it is recommended to utilise a processor with x86 architecture and 64 bits.

Software Requirements

The DL models were trained on a High-Performance Computer (HPC) utilising the Red Hat Enterprise Linux 8 operating system running on Linux kernel 4.18[14]. However, the models can be trained and tested on a Debian-based distribution such as Ubuntu 20.04, and for RPMNet and PREDATOR evaluation metrics have been obtained in a system utilising Windows 10 in conjunction with a Linux subsystem for Windows. The models were trained with Python version 3.8, and for the ML library, PyTorch was utilised, albeit with differing versions based on the requirements of the models. Therefore, due to the differing requirements of each model, it is highly recommended to utilise an environment management system such as Anaconda. For this experiment, miniconda3 was used.

4.2 Experiment Procedure

The experiment follows a simple procedure of training and evaluating the performance of a model on the utilised dataset; with the dataset being subject to change. Initially, the experiment can be conducted by training the Deep Learning models on the ModelNet40 dataset. For training, it is strongly advised to utilise a High-Performance computer if available. The model metrics are then evaluated by testing the models on the test set of ModelNet. However, for this study, models acquired from the HoloLens have been used, by preprocessing said models and initially matching identical point clouds. In later versions, the target model has been matched with a non-identical source with substantially lower acquired points in order to test the system's performance in low-detail situations. The system has finally been tested in comparison to an algorithm-based approach, in which the match performances have been evaluated. To conclude, by utilising the procedure outlined in this chapter the experiment could be replicated as desired.

[4]https://images.nvidia.com/content/technologies/volta/pdf/volta-v100-datasheet-update-us-1165301-r5.pdf

5 Experimental Results

In this section the evaluation results of the DL models will be provided. Section 5.1 contains the results of RPMNet, and section 5.2 contains the evaluation results of PREDATOR.

5.1 RPMNet

RPMNet when trained and tested on the scanned model with navigator data, has been able to demonstrate a generalisation ability on the provided source and target point clouds. The model has been trained on the source and target points sampled to an equal length of 1024 points, had random rotations and translations applied to the source model (the sensor data) and had the indices of the points shuffled to deter the model from generalising upon them. The results of the utilised models can be viewed in Table 1.

Pre-Op Model	Navigator	Isotropic Rotation Error	Isotropic Translation Error	Modified Chamfer Distance
1	**1**	9.450	231.4	38.30
1	**2**	19.63	459.6	44.74
1	**3**	3.956	108.1	41.30
1	**4**	16.09	377.5	64.10
1	**5**	4.690	65.81	49.56
2	**1**	23.01	657.6	48.22
2	**2**	4.994	84.01	46.52
2	**3**	13.89	397.0	52.65
3	**1**	14.77	480.7	35.16
3	**2**	11.90	333.2	13.32
3	**3**	3.566	112.5	20.15

Table 1: Results of the Fifth Iteration for RPMNet. The model has been trained and evaluated on initial Euler angle rotations of 45 degrees, with the skull models having a volume of 121000, 106000, and 24000 cubic millimetres respectively. The isotropic errors were calculated using equations 1 and 2, while the Chamfer Distance metric was calculated by utilising Equation 3 respectively. The rotation error is given in degrees, the translation error is given in millimetres, and the Chamfer distance is given in millimetres squared.

The model has been trained on a mean rotational difference of 45 degrees, and the maximum rotation difference has been increased to 90 degrees for later evaluations. The evaluation metrics, as well as visualised outputs, indicate that the model is able to generalise certain regions incorrectly, and therefore not suitable for large rotational differences. For example, the less accurate match in Figure 2, which was obtained with an initial Euler rotation of approximately 90 degrees, does not contain any points in the eye cavities in contrast to the more accurate match; indicating that the regions the model was able to discern were most likely the aforementioned region of eye sockets. In summary, the model is able to demonstrate general matches in rotational differences of 45 degrees but demonstrates incorrect regional generalisations in initial rotational differences of 90 degrees.

In terms of time performance, the model is able to perform with an efficiency of a mean of 1.2 seconds on average for all runs with points sampled to 1024 points. However, on non-sampled large point cloud data such as the high-quality pre-operative models, the evaluation time has been observed to exceed 60 seconds, with negligible performance difference.

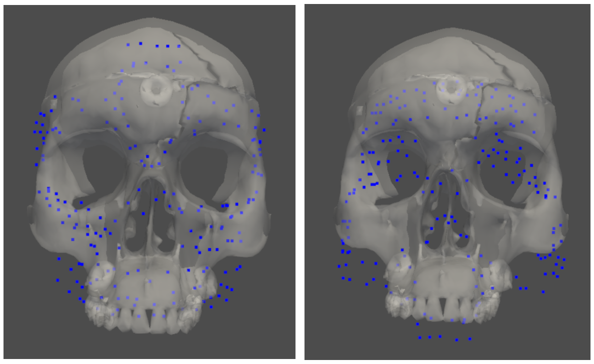

Figure 2: Comparison of two evaluation results for Preoperative model 1 matched on navigator data 5. The left model has been evaluated with an initial mean rotational error of 90 degrees for all axes, while the model on the right had an error of 45 degrees. The right model demonstrates a more accurate match result, with a rotational error of 5.23 degrees. The less accurate match does not contain any points in the eye cavity region of the skull, demonstrating a possible generalisation around the aforementioned region.

In summary, the model is able to generate transformation predictions on sampled data in under 2 seconds, without resulting in a noticeable performance loss in accuracy.

In order to compare the DL approach to the results of the ICP-based approach outlined by Weyns [12], RPMNet was evaluated on the source and reference point clouds as outlined. For this experiment, due to the model demonstrating tolerance to noise and rotations, the independent variable has been the voxel volume. The modified Chamfer distance metric outputs of the utilised models based on voxel volumes used for downsampling can be found in Table 2.

	Chamfer Metric Results			
Voxel Size	Model 1	Model 2	Model 3	Mean Evaluation Duration
4	4.75	5.06	11.4	2.53
5	16.0	52.5	34.7	1.26
6	10.6	5.43	64.8	1.13
7	11.2	7.21	8.28	1.61
8	7.37	5.27	8.96	1.89
9	6.79	9.41	7.70	2.07
10	47.6	4.93	39.4	2.04

Table 2: The evaluation results of RPMNet on voxel downsampled data for the 3 Pre-Operative model data. The voxel sizes are in cubic millimetres, the Chamfer metric results are given in square millimetres, and the evaluation duration is in seconds.

5.2 Overlap PREDATOR

PREDATOR has been able to demonstrate accurate results when utilising ModelNet40. The model, when evaluated using the parameters provided by Huang et al., is capable of achieving a rotational mean error of 1.806 degrees, with a mean translation error of 0.01895 millimetres. The model also returns a bidirectional Chamfer error of 0.0009024 millimetres squared, indicating a successful match. Therefore, based on the results of the evaluation of ModelNet the model has been deemed a viable candidate for testing on HoloNav data.

In contrast to results obtained from ModelNet, training and testing PREDATOR on HoloNav data has not resulted in a successful match. The initial approach in evaluating PREDATOR by transforming the input data to conform to ModelNet specifications has resulted in a successful match. The model then was tested by training and testing on the first Pre-Operative model without any preprocessing applied to the model. The results indicate that PREDATOR is able to register data with the same density perfectly in certain situations, but the currently utilised parameters return inconsistent results. The metrics can be found in Table 3, and the visual output of the third and fourth runs can be found in Figure 3.

Isotropic Rotation Error	Isotropic Translation Error	Chamfer Distance
52.0718	0.0341	1288235
37.3291	0.1149	1241043
31.6535	493.0022	423.4916
0.074	**0.7065**	**10.7983**
71.3116	0.0814	2308491

Table 3: The evaluation results for PREDATOR in matching source and target point clouds with random sampling. The model used for evaluation is the first preoperative model, with 1024 points randomly selected, and having applied different rotations and translations for every run. For this evaluation, The third run resulted in a semi-alignment; while the fourth run, highlighted in bold, resulted in a perfect match. Due to the source and target point clouds originating from the same origin, the Chamfer Distance proposed by Yew and Lee has been utilised.

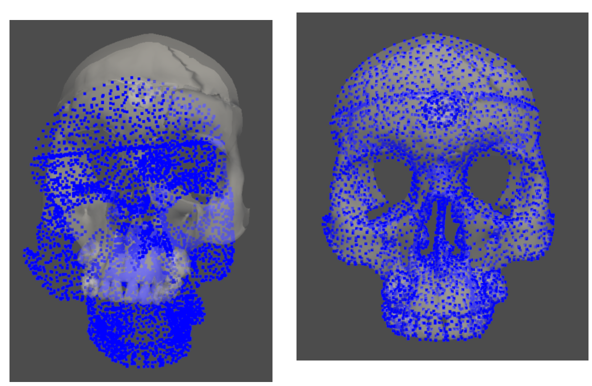

Figure 3: A visualisation of the parameters of the third and fourth run of PREDATOR. The third run demonstrates a partial overlap by the model, while the fourth run demonstrates a perfect match.

PREDATOR was also evaluated on the source and reference point clouds as outlined in subsection 5.1 for comparison with an ICP-based approach. The modified Chamfer distance metric outputs of the utilised models based on voxel volumes used for downsampling on PREDATOR can be found in Table 4.

	Chamfer Metric Results			
Voxel Size	Model 1	Model 2	Model 3	Mean Evaluation Duration
2	**658000**	**954000**	**418000**	6.32
3	**658000**	**954000**	**418000**	2.21
4	932	303	**418000**	1.27
5	809	503	740	1.06
6	724	904	**418000**	1.04
7	578	**954000**	444	1.25
8	106	179	674	1.16
9	367	**951000**	**418000**	1.13
10	300	**954000**	119	1.2

Table 4: Modified Chamfer Metric results of PREDATOR based on the work of Weyns[12]. The models were downsampled by utilising voxel downsampling instead of random point selection, with voxel size in cubic millimetres. Each pre-operative skull model was sampled on "visible" points to generate the reference point cloud, with noise applied on both source and target models to evaluate generalisation abilities. Voxel size of 1 cubic millimetre did not result in an evaluation result due to memory limitations of the system utilised for evaluation, while fields highlighted in bold have not resulted in a predicted match. The evaluation duration is in seconds, and the Chamfer metric results are given in square millimetres, approximated to 3 significant figures.

6 Responsible Research

This research has aimed to determine whether utilising DL models could be viable for Point-Cloud registration for HoloNav. Therefore, the data utilised for evaluation has been created to be similar to how the system can be used when integrated into the environment. For example, the source data utilised has been acquired by scanning a model of a skull with attached fiducial points, while the reference point cloud has been acquired by sampling points with the navigator. However, the utilised model has certain features that actual sensor outputs would not have, which can affect the match quality of the model. For example, the pre-operative model utilised as the source Point Cloud for this research is a complete scan of a 3D model that also contains surfaces that would be impossible to acquire with a depth scanner. Therefore, when the model is trained on the aforementioned input, said surfaces are considered to be features as well; which can result in lower expected performance. The surface for evaluation may also be low in detail, which would also affect feature extraction. Therefore, it should be noted that the model evaluated for this research does not reflect actual surgical environments, and therefore utilising the models as a standalone system may not be ethical when taking into consideration the fact that an incorrect match may delay a life-saving surgery.

The methods are reproducible with some adjustments to the DL models. First, the data loaders of the models need to be adjusted in order to process the data into an acceptable format. The users can acquire navigation data by utilising the HoloNav navigator; however, in cases where the navigator cannot be used, points can be randomly sampled from the surface of the source model. The evaluation metrics in the models are calculated by comparing the initial alignment of the source with the target, and therefore training the models utilising registered models is highly recommended. For evaluation, the rotation and translation errors have been obtained by utilising the registered point clouds. Therefore, the metrics will be inaccurate for raw unregistered data, and utilisation of a different metric such as the Chamfer Distance or visual comparisons may prove more useful. Finally, a public Git repository[5] will be included with this report in order to replicate the results of this experiment. To conclude, the research has been made to be reproducible, and with the outlined procedure, or by utilising the resources provided, the users can acquire similar results to the ones that were used in this research.

7 Discussion of Results

7.1 RPMNet

RPMNet is able to predict consistent but semi-accurate matches on the HoloNav data. The model was able to provide consistent semi-accurate metrics in preprocessed pre-operative data when inferred on a model trained on ModelNet. A possible explanation of this result may be the fact that the transformation generation for RPMNet is algorithmic in origin; and due to the source and reference clouds originating from the same data, the matching features can be determined easily.

However, on data that does not conform to ModelNet standards, such as the HoloNav input, the model provides less accurate results. The model relies heavily on inferred features from neighbouring points to provide accurate registration, which indicates that in data with large distances between points the model can struggle to determine features. However, due to the model utilising the Robust Point Matching algorithm it is consistent in the results it provides and has a reasonable evaluation time of approximately 1 second per 5 iterations per point cloud. To summarise, RPMNet is able to achieve robust and semi-accurate results consistently with a reasonable evaluation time, and therefore can be considered for further research for utilisation in HoloNav.

In addition to the aforementioned issue, a major weakness of RPMNet stems from the fact that the generalisation ability of the model is heavily dependent on the training data parameters. Therefore, in situations where the parameters of the test data do not conform to training parameters the model demonstrates inaccurate results. As demonstrated in Figure 2 the model, trained on an initial Euler angle difference of approximately 45 degrees, demonstrates high inaccuracy with an initial rotation of 90 degrees. A possible explanation may stem from the symmetrical nature of the models, as well as the random of the points utilised for evaluation, which may cause the model to register incorrect voxels. This is supported by the fact that the model performs better on non-symmetrical data, as evident from Figure 4. To summarise, RPMNet can be a viable candidate in environments where the initial data is aligned symmetrically, or if the navigator acquires samples non-symmetrically.

[5] https://github.com/alpcicimen/holonav-dl-registration

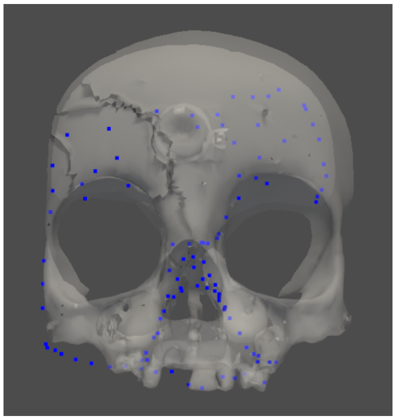

Figure 4: The visual output of Pre-Op model 3 on Navigation data 2, outlined in Table 1. The data had the lowest Chamfer Distance error and was also the data that had a noticeable asymmetry.

7.2 PREDATOR

The results of PREDATOR differ from the results obtained with RPMNet due to the architectural differences between the utilised models. RPMNet implements a robust matching algorithm by utilising nearest-neighbour clustering. In contrast, PREDATOR utilises an encoder and decoder architecture in order to establish correspondences. The encoded data is formatted as voxels of a specific volume, which is utilised in the form of ”superpoints” located in the centre of each voxel. The trained model utilises the aforementioned superpoints to extract contextual information from the source and target separately in the form of ”bottleneck points”. The model consequently establishes correspondences between the source and target point clouds based on the aforementioned bottleneck points. Therefore, for PREDATOR to perform registration, Huang et al.[9] state that the overlap region, or the bottleneck, requires a certain density on both the source and target point clouds. In summary, in contrast to RPMNet’s nearest neighbour approach, PREDATOR’s ”bottleneck matching” approach is limited in scenarios where the utilised point clouds have uneven point densities.

Therefore, the results of PREDATOR on the utilised data showcase the aforementioned shortcoming of the model reported by Huang et al. In data with even density, such as the ModelNet dataset, as well as the HoloNav pre-operative model, the model is able to demonstrate accurate registration, even on data sampled from a specific region in the source point cloud. However, the model is unable to establish accurate contextual information on the unevenly sampled point clouds of the pre-operative models and navigator outputs and therefore is unable to establish bottleneck features to generate correspondences. Therefore, the navigator data can be registered correctly if it is sampled with a similar density in comparison to the pre-operative model. In summary, due to inherent issues contained within PREDATOR in matching point clouds with uneven density, the utilisation of this DL for registration in HoloNav may not be suitable.

7.3 Discussion of the Results of the DL Models in Comparison to a Registration Algorithm

In comparison to the work of Weyns[12], both RPMNet and PREDATOR demonstrate less accurate results against the application of fast-point feature histograms (FPFH) and ICP for certain voxel volumes. For example, in Table 5 RPMNet demonstrates comparable results to FPFH with ICP for most voxel sizes, while demonstrating its generalisation capability by demonstrating consistency in contrast to an algorithmic approach. In contrast, PREDATOR was not able to achieve comparable results. However, similar to algorithmic approaches RPMNet also demonstrates a drop in accuracy in large voxel sizes due to a possible lack of information to determine correspondences.

Voxel Size	RPMNet CD	FPFH Only	FPFH (With ICP)
4	4.75	519.7783	382.8319
5	16.0	150.9141	**1.00298**
6	10.6	186.9544	**1.83306**
7	11.2	**1.09607**	**1.23063**
8	7.37	7845.365	7887.997
9	6.79	109.4227	**1.09453**
10	47.6	2071.387	2109.046

Table 5: Comparison of evaluation results of RPMNet on data sparsity results to the results of an algorithmic approach outlined by Weyns[12]. The results were obtained from the evaluation of pre-operative model 1 utilising the Chamfer Distance for RPMNet and Mean-Square error for algorithmic approaches. The results of algorithmic approaches with more accuracy in comparison to RPMNet are highlighted in bold. For voxel sizes less than 4 cubic metres RPMNet was not able to evaluate, and therefore the results of algorithmic approaches were not included.

However, due to differences in the evaluation metrics, the difference in accuracy may be less than the results outlined in Table 5. RPMNet utilises the reduced Chamfer Distance for evaluation, while Weyns applies an addition of reference points to the source cloud to ensure a ground-truth error of 0. Therefore, the metrics differ in the lack of ”correspondence points” for RPMNet. In summary, RPMNet demonstrates a similar performance to the algorithmic approaches outlined by Weyns and therefore may be a viable candidate for utilisation.

8 Conclusions and Future Work

This research has proposed the utilisation of Deep Learning models to handle the registration problem in HoloNav. To determine the viability of utilising such a system the models were evaluated based on their match accuracy and evaluation time. From the considered DL models RPMNet has demonstrated a more general but inaccurate match, while PREDATOR has demonstrated an ability to precisely align point clouds of similar density. Both models have demonstrated an ability to generalise in data with noise, while PREDATOR demonstrated further precision in matching data sampled from a specific region of the Pre-Operative model. In terms of evaluation time, PREDATOR has demonstrated a slightly faster evaluation duration, with a mean evaluation duration of 1.06 seconds in comparison to the duration of 1.87 seconds for RPMNet. In summary, both models demonstrate a quick evaluation time, as well as general robustness to noise, and therefore demonstrate that DL models can be suitable for patient-alignment registration in terms of an initial robust match solution in conjunction with a precise algorithm.

However, the DL models utilised demonstrate certain issues that, if resolved, may ensure that DL-based approaches can be viable as a standalone navigation system. The evaluation metrics of RPMNet suggest that the model, due to its feature extraction method of nearest-neighbours, does not perform well on data with points sampled randomly. In contrast, PREDATOR was not able to generalise in most cases. A possible explanation for the aforementioned issue is the fact that the models were configured for cases where the data is uniformly sampled. RPMNet is configured to utilise ModelNet, a dataset with uniformly sampled data, and the intended use of PREDATOR is in SLAM applications, where sensor data is in general uniformly sampled. Another issue is the fact that the models become inaccurate in initial rotation differences larger than 45 degrees, and therefore not being suitable for utilisation in actual surgical environments where registration may need to be performed on data with high rotational differences. In summary, the main issues in the utilisation of DL for surface registration stem from the sampling of points from the utilised data, as well as the architecture of the utilised models. Further research can be conducted by evaluating the models with more data to determine the validity of this claim, applying voxel downsampling on the source and target point clouds to obtain similarly sampled data, or by utilising a DL model that may be able to handle registration of point clouds with non-uniform sampling.

However, in comparison to algorithm-based approaches such as the Iterative-Closest-Point algorithm RPMNet was able to demonstrate similar accuracy with better consistency. In contrast, the methods utilised by Weyns have better accuracy but result in less consistency. Therefore, the results of RPMNet indicate that it may be possible to utilise a DL learning model for a more general outcome for registration. In summary, DL models demonstrate a generalisation ability in comparison to algorithmic approaches and can provide generally accurate results while demonstrating fast evaluation times. Therefore, a DL approach may be suitable for HoloNav if it can demonstrate a generalisation ability on data non-evenly sampled data.

A Visual Outputs of Utilised HoloNav Data

A.1 Pre-Operative Model 1

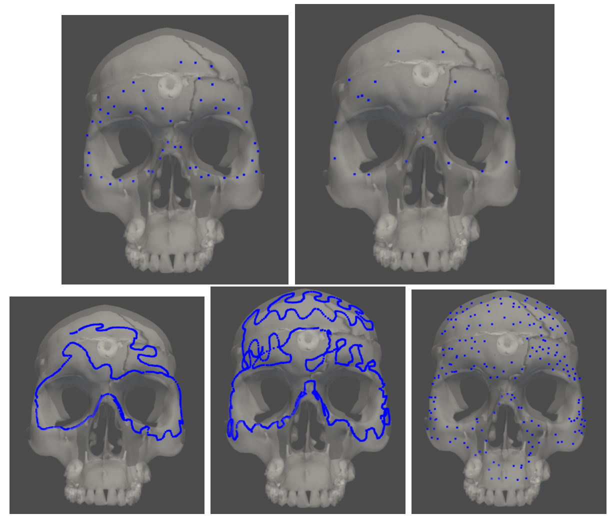

Figure 5: Various outputs of Pre-Op model 1 with registered navigator data.

A.2 Pre-Operative Model 2

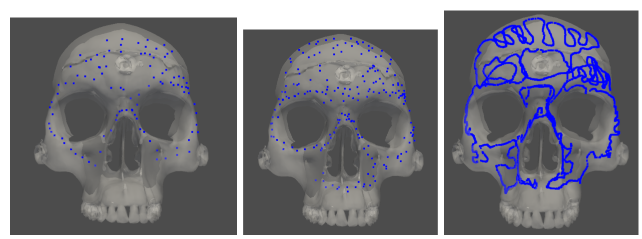

Figure 6: Various outputs of Pre-Op model 2 with registered navigator data.

A.3 Pre-Operative Model 3

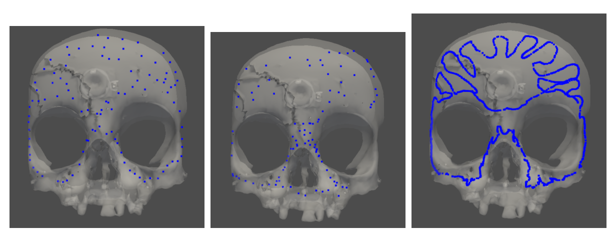

Figure 7: Various outputs of Pre-Op model 3 with registered navigator data.

References

[1] M. Benmahdjoub, T. van Walsum, P. van Twisk, and E. Wolvius, "Augmented reality in craniomaxillofacial surgery: added value and proposed recommendations

through a systematic review of the literature," *International Journal of Oral and Maxillofacial Surgery*, vol. 50, no. 7, pp. 969–978, 2021.

[2] L. Li, R. Wang, and X. Zhang, "A tutorial review on point cloud registrations: Principle, classification, comparison, and technology challenges," *Mathematical Problems in Engineering*, vol. 2021, 2021.

[3] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," *CoRR*, vol. abs/1612.00593, 2016.

[4] F. Incekara, M. Smits, C. Dirven, and A. Vincent, "Clinical feasibility of a wearable mixed-reality device in neurosurgery," *World neurosurgery*, vol. 118, pp. e422–e427, 2018.

[5] X. Chen, L. Xu, Y. Wang, H. Wang, F. Wang, X. Zeng, Q. Wang, and J. Egger, "Development of a surgical navigation system based on augmented reality using an optical see-through head-mounted display," *Journal of Biomedical Informatics*, vol. 55, pp. 124–131, 2015.

[6] E. Arnold, S. Mozaffari, and M. Dianati, "Fast and robust registration of partially overlapping point clouds," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1502–1509, 2021.

[7] Q. H. Gao, T. R. Wan, W. Tang, and L. Chen, "A stable and accurate marker-less augmented reality registration method," in *2017 International Conference on Cyberworlds (CW)*, pp. 41–47, IEEE, 2017.

[8] Z. J. Yew and G. H. Lee, "RPM-net: Robust point matching using learned features," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, jun 2020.

[9] S. Huang, Z. Gojcic, M. Usvyatsov, A. Wieser, and K. Schindler, "Predator: Registration of 3d point clouds with low overlap," 2020.

[10] S. Horache, J.-E. Deschaud, and F. Goulette, "3d point cloud registration with multi-scale architecture and unsupervised transfer learning," 2021.

[11] V. Sarode, X. Li, H. Goforth, Y. Aoki, R. A. Srivatsan, S. Lucey, and H. Choset, "Pcrnet: Point cloud registration network using pointnet encoding," *CoRR*, vol. abs/1908.07906, 2019.

[12] M. Weyns, "Improving patient alignment by leveraging point-cloud surface registration techniques," 2022.

[13] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," *arXiv preprint arXiv:1406.5670*, 2014.

[14] Delft High Performance Computing Centre (DHPC), "DelftBlue Supercomputer (Phase 1)." https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase1, 2022.