Interactive Semantic Segmentation of 3D Medical Images

Comparative Analysis of Discrete and Gradient Descent Based Batch Query Retrieval Methods in Active Learning

Arjun Vilakathara Supervisor(s): Klaus Hildebrandt, Nicolas Chaves de Plaza
EEMCS, Delft University of Technology, The Netherlands

June 25, 2023

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 25, 2023

Abstract

Accurate segmentation of anatomical structures and abnormalities in medical images is crucial, but manual segmentation is time-consuming and automated approaches lack clinical accuracy. In recent years, active learning approaches that aim to combine automatic segmentation with manual input have gained attention in the field, aiming to reduce the annotation effort required for training segmentation models. Batch query retrieval is a key component of active learning as it is a technique that allows for the simultaneous selection of multiple regions/points for annotation. This study investigates the effectiveness of discrete batch query retrieval methods compared to the traditional approach using gradient descent in the context of 3D medical image segmentation. Our experiments show that the active learning paradigm with batch query retrieval provides a few advantages over the gradient descent approach. Additionally, we analyze the impact of discretizing the query retrieval strategy on system performance. Our findings suggest that discretization can lead to slight performance degradation in terms of segmentation quality but offer computational advantages and faster convergence. We also discuss open issues, such as the interpretability of active learning methods, and recommend further research on combining active learning with other segmentation techniques. Overall, our study contributes to the understanding of active learning in medical image segmentation and provides insights for developing more efficient and accurate interactive segmentation models.

1 Introduction

Three-dimensional image segmentation is a crucial task in medical image analysis, allowing for accurate segmentation of anatomical structures and abnormalities in medical images. However, the process of segmentation is challenging, and there are many approaches to it. While manual segmentation, involving a professional annotating slice by slice, provides accurate results, it is a tedious and expensive process that can be subject to user bias. On the other hand, fully automated segmentation approaches [10] are prone to errors and may not meet the necessary level of accuracy and robustness required for clinical practice. Thus, there is a growing interest in developing highly automated segmentation methods that can reduce the need for manual intervention while improving accuracy and thus further questions regarding the techniques and approaches used in them.

The focus of this research paper is on Active Learning (AL) [9], an approach that combines human expertise with machine learning algorithms to enhance segmentation accuracy. Specifically, the journal "Active Learning for Interactive 3D Image Segmentation" [11] presents an AL-based solution for 3D image segmentation. This solution enables user interaction with 3D data in a 2D format and employs a query strategy to identify regions where additional labeled training data can improve segmentation.

To provide a comprehensive analysis, it is important to acknowledge the relevance of other studies in the field. For instance, the paper "A Continual Learning Framework for Uncertainty-Aware Interactive Image Segmentation" [13] introduces a continual learning framework that effectively learns from user annotations, aiming to improve segmentation not only for the current image but also for future tasks. Although the specific framework is not directly incorporated in this research paper, the concept of leveraging user annotations for improved segmentation is in line with the goal of this study.

Similarly, the paper "A Survey on Active Learning and Human-in-the-Loop Deep Learning for Medical Image Analysis" [3] explores the role of humans in deep learning-enabled systems and highlights techniques such as active learning for optimal model performance

and interaction with model outputs for meaningful interpretation and response. While not directly influencing the research in this paper, the insights from this survey reinforce the significance of active learning approaches in medical image analysis and their potential to enhance segmentation accuracy.

Considering the relevance of these papers, this research paper aims to fill a gap in the field by exploring the impact of discretizing the query retrieval strategy on system performance. The investigation specifically focuses on the strategy used for selecting regions presented to the user for labeling. By modifying the gradient descent algorithm and cost functions and imposing constraints, such as limiting the orientation of evaluated planes, this study aims to improve the performance of the AL-based solution proposed by [11].

The motivation behind this investigation is to determine whether these modifications would lead to improved performance in terms of the number of iterations required to achieve acceptable segmentation results, while also considering the accessibility of the system. The accessibility aspect focuses on the preference for slices of canonical anatomic axes, which are easier to annotate compared to slices of arbitrary axes.

Overall, this research paper builds upon the foundation laid by previous studies on active learning and interactive image segmentation, such as the work by [11], by addressing a specific research gap related to the query retrieval strategy. By exploring the impact of discretization and imposing constraints on the AL-based solution, this study aims to contribute to the advancement of interactive segmentation methods for improved accuracy and efficiency in the domain of medical image analysis.

2 Interactive Segmentation using Active Learning

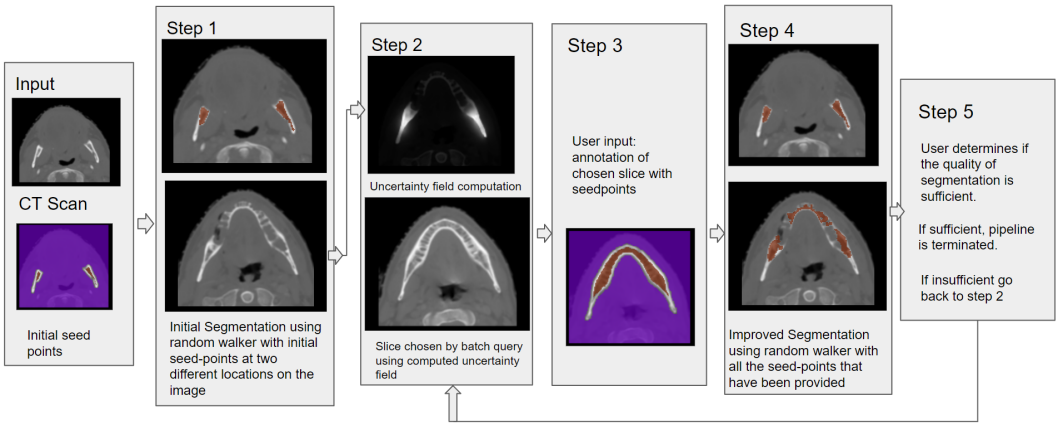

Figure 1: Pipeline of the system

To gain a better understanding of the research question and the interactive segmentation process, it is important to keep the entire pipeline in mind. The system can be described by the illustrative pipeline shown in Figure 1. Initially, in step 1, the image (e.g., a CT scan) is segmented using a random walker method, starting with initial seed points. The pipeline proceeds to step 2. In step 2, an uncertainty field is computed, which serves as the basis

for an active learning (AL) batch query. The AL process, described in detail in subsection 3.1, identifies the region of maximum uncertainty, as depicted in the chosen slice in Figure 1 in the 2 box, and presents it to the user as shown in step 3. The user then annotates this particular slice, which is then fed back into the system to generate an improved segmentation as shown in step 4. At this stage, the user once again evaluates the new segmentation as shown in step 5, and the pipeline continues to iterate steps 2 and 5 until the user is satisfied with the segmentation at which point the pipeline is terminated.

3 Methodology

The following section will discuss what will be done in order to answer the question of how discretizing the query retrieval strategy works in comparison to the continuous alternative and further how it would affect the performance of the system. To do so, first, we detail the batch query mechanism based on gradient descent in subsection 3.1 and then present a discrete process to find an optimal plane in subsection 3.2. To ensure clarity, the following notations are introduced: the spatial image domain is denoted as $\Omega \subset R^3$, the classifier labeling is represented by $y : \Omega \to 0, 1$, the image intensity function is denoted as $I : \Omega \to R$, and the classification uncertainty field is represented by $U : \Omega \to R$. These notations will be referenced in the subsequent equations and discussions.

3.1 Batch Query Active Learning Based on Planar Slices

As mentioned earlier, in cases where the segmentation is insufficient, the user is prompted to heuristically provide additional labels (which in our implementation of the random walker are seed points) in the most uncertain regions to improve the segmentation. Although one approach could be to instruct the user to label the most uncertain points identified by the uncertainty field point by point, this method would be time-consuming. Hence, a more effective strategy involves querying the user for multiple labels simultaneously, presenting them with a plane of uncertain points that they can label simultaneously, known as a batch query [4].

To determine the plane of maximal uncertainty $U_{\mathcal{P}}$, we follow the methodology described in [11]. The plane is parameterized using a normal n and a point P, also described as $(P_1\ P_2\ P_3)$. Thus, the plane p can be expressed as $f_{\mathcal{P}}(u, v) = x + ua + vb$, where a and b are orthonormal vectors perpendicular to n, and p is the point on the plane. The absolute value of the cross product of the partial derivatives of $f_{\mathcal{P}}$ with respect to u and v gives 1. Therefore, the equation for $U_{\mathcal{P}}$ is defined as follows:

$$U_{\mathcal{P}} = \int_{\mathcal{P}} \int_{\mathcal{P}} U(x) \mathrm{d}A = \int_{\infty} \int_{\infty} U\left(f_{\mathcal{P}}(u, v)\right) \left| \frac{\partial f_{\mathcal{P}}}{\partial u} \times \frac{\partial f_{\mathcal{P}}}{\partial v} \right| \mathrm{d}u\ \mathrm{d}v. \tag{1}$$

To update the normal n, we compute the vector $\nabla_{n_{\mathcal{P}}} U_{\mathcal{P}}$, denoted as T, using the following equation:

$$T = \int_{\infty} \int_{\infty} \left(u J^T_{a,n_{\mathcal{P}}} + v J^T_{b,n_{\mathcal{P}}}\right) \nabla_x U\left(f_{\mathcal{P}}(u, v)\right) \mathrm{d}u\ \mathrm{d}v \tag{2}$$

where $\nabla_x U\left(f_{\mathcal{P}}(u, v)\right)$ represents the gradient of the uncertainty field at the position defined by u and v according to the plane equation. The result of equation (2) is a vector $\mathbf{T} = (T_1\ T_2)$, which is used to update the normal n' as follows:

$$\mathbf{n}' = \mathbf{n} + T_1\mathbf{a} + T_2\mathbf{b}. \tag{3}$$

Similarly, the point on the plane is updated using the following equation:

$$\nabla_{p\mathcal{P}} U\mathcal{P} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \nabla_x U\left(f_{\mathcal{P}}(u,v)\right), \mathrm{d}u, \mathrm{d}v, \tag{4}$$

yielding a vector $\mathbf{M} = (M_1\ M_2\ M_3)$, which is added to the point following:

$$P = (P_1 + M_1\ P_2 + M_2\ P_3 + M_3). \tag{5}$$

The baseline implementation performs this optimization for 40 unique random initializations of normals and points, where for each initialization it updates the normal and point using equations (2) and (4) 250 times. The vectors returned from equations (2) and (4) are scaled using a scaling factor that becomes smaller over the 250 iterations.

3.2 Discrete Batch Query

To address the research question at hand, a discrete batch query method based on planar slices is introduced and compared to the continuous baseline method (gradient descent). The discrete batch query method involves adding constraints on the orientation of planes that are considered when finding the plane with maximal uncertainty $U_{\mathcal{P}}$. This section explains the process of discretizing the volume and finding the plane with the highest uncertainty through it.

3.2.1 Discretizing the Uncertainty Field and Image

In this research, the constraints are based on the axes of the plane symmetries of a cube, as illustrated in Figure 2. The x, y, and z axes correspond to the axial plane (z-plane), sagittal plane (x-plane), and coronal plane (y-plane) in medicine [2]. These axes were chosen due to their familiarity among professionals working with medical images and their relevance in pathology analysis [5, 6], which are the target users of the evaluated system. Additionally, Figure 2 shows the other six axes corresponding to the cube diagonals.

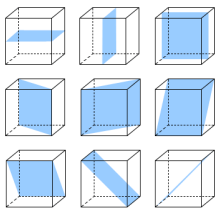

Figure 2: Drawings of the plane symmetries of a cube [1]

By defining the axes of iteration, the uncertainty field, represented by a 3D array in code, can be traversed. For the x, y, and z axes, the iteration takes place through the planes in the uncertainty field, starting from the baseline formed by the (y, z), (x, z), and (x, y) axes, respectively. Likewise, for each diagonal axis, the uncertainty field is rotated to align the diagonal axes parallel to the x-axis, allowing iteration through the planes formed by the (y, z) axes. The same iterative process can be applied to the image, enabling the presentation of any desired slice to the user.

3.2.2 Finding the Optimal Plane

To find the most uncertain plane, all nine iterations through the uncertainty field, as described above, are executed. The iteration involves calculating the sum of uncertainties for each plane. By considering all iterations, the plane with the largest sum of uncertainty is determined. Subsequently, the corresponding slice from the original image is sent to the user for annotation. This approach ensures that the most uncertain plane, as determined by the highest sum of uncertainties, is selected for user interaction.

4 Experiments and Results

The following subsection presents a description of the experiments conducted to evaluate the discrete batch query retrieval in comparison to the baseline gradient-descent-based approach. The subsection covers the experimental methodology, presents the obtained results, and provides a brief discussion of the findings. The solutions are tested on both synthetic, discussed under subsection 4.1, and real data discussed under subsection 4.2.

4.1 Experiment 1: Synthetic Data

This experiment aims to evaluate the performance of each method by creating two synthetic uncertainty fields, inspired by the approach in [8]. The first uncertainty field will be a 20x20x20 grid filled with zeros, except for one slice that will be filled with ones (as shown in Figure 3). The second uncertainty field will have its highest value in the center slice, gradually decreasing towards the outer slices (visualized in Figure 3). To ensure a comprehensive evaluation, these fields will be rotated to align their most uncertain slice with each discrete axis. For instance, when considering the single plane of maximal uncertainty, the resulting nine uncertainty fields would resemble the plane symmetries of a cube (as shown in Figure 2). Furthermore, the fields will be rotated randomly to different axes, and the evaluation process will be repeated accordingly. The goal is to see how the discrete method functions against the baseline gradient descent.

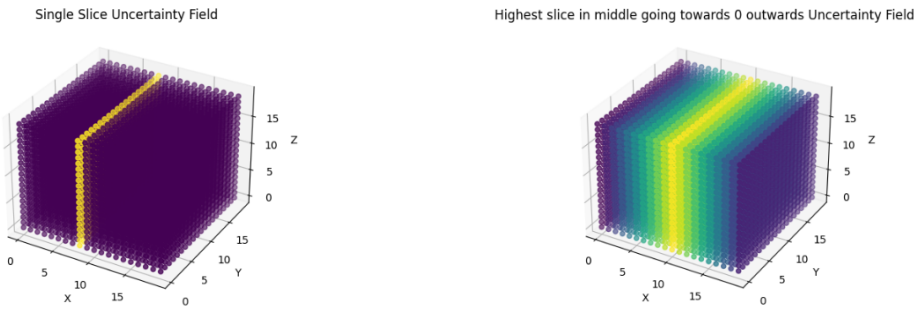

Figure 3: The two synthetic uncertainty fields, yellow is 1 and purple is 0

4.1.1 Synthetic Data Setup

The generation of uncertainty fields involves creating NumPy arrays of zeroes with dimensions 20x20x20. For the first set of uncertainty fields, a plane containing ones is inserted at a random point within the array, using specific indexing for the x, y, and z axes to achieve maximal planes along these axes. To obtain uncertainty fields with maximal planes along the diagonal axes, a custom Python function utilizes Scipy's affine transform to rotate the uncertainty field initially aligned with the x-axis. In total, ten additional uncertainty fields are generated with random orientations. Similarly, for the second set of uncertainty fields, values are assigned to the array using a loop, where the intensity varies based on the position such that the plane in the center has the highest value and the planes outwards go toward zero. The maximal planes on the x, y, and z axes are assigned using specific indexing, while the uncertainty field with the maximal plane on the x-axis is rotated to align with the diagonal axes. Ten more rotated uncertainty fields are created with random orientations.

Both discrete and baseline batch approaches are tested on all the uncertainty fields generated above and whether they find the maximal plane is recorded, true if it was found, and false otherwise.

4.1.2 Synthetic Data Results

Table 1: Single slice test

Orientation of most uncertain slice	x	y	z	d1	d2	d3	d4	d5	d6
Testing with slice on same axis as discrete	T	T	T	T	T	T	T	T	T
Gradient descent	F	F	F	F	F	F	F	F	F

Table 2: Slice with middle highest and going to 0

Orientation of most uncertain slice	x	y	z	d1	d2	d3	d4	d5	d6
Testing with slice on same axis as discrete	T	T	T	T	T	T	T	T	T
Gradient descent	T	T	T	T	T	T	T	T	T

Table 3: Type of Uncertainty Field

Type of Uncertainty Field	Slice with middle highest and going to 0	Single slice of 1's
Orientation of most uncertain slice	Unaligned	Unaligned
Testing with a slice on the same axis as discrete	F	F
Gradient descent	T	F

4.1.3 Synthetic Data Results Discussion

The performance of the discrete and baseline methods on an uncertainty field consisting of a single slice of 1's among 0's is presented in Table 1. The discrete method successfully identifies the optimal slice when it parallels one of the planes of symmetries. However, when the optimal slice is unaligned, the discrete method chooses a slice with an uncertainty of 20,

which is inefficient and requires multiple iterations to cover the entire optimal plane. On the other hand, the baseline method consistently fails to identify the optimal slice, producing slices with a total uncertainty of 0.

In the tests using the uncertainty field with a slice starting with maximal uncertainty in the middle and decreasing towards the edges, the discrete method performs just as well as in the single slice test. The additional values do not affect its evaluation of the most uncertain slice, and it consistently identifies the optimal slice. However, unlike before, the baseline method performs well in this scenario, consistently identifying the most optimal slice. It's worth noting that the slices chosen by the baseline method have the highest possible uncertainty value but may not be intuitive from a human perspective, for example, it found a slice with normal $(0.59511033, 0.21037498, 0.77561979)$ at point 10.

When examining synthetic data where the maximal slices are not parallel to one of the planes of symmetries, only the gradient descent method is able to find the maximal plane, as indicated in Table 9.

4.2 Experiment 2: Real Data

The experiments aim to investigate the benefits of batch query approaches in real medical image segmentation, specifically comparing the gradient-descent-based approach to the discrete approach. The primary focus is to evaluate the performance in terms of segmentation quality against the iterations required which is defined as the number of times the system selects a slice for user annotation.

The data used for this report and test is sourced from the Public Domain Database for Computational Anatomy (PDDCA), a publicly available dataset maintained by Dr. Gregory C Sharp and his group at Harvard Medical School – MGH in Boston. The dataset [7], version 1.4.1, was originally utilized for the 'Head and Neck Auto Segmentation MICCAI Challenge (2015)', and it can be accessed at the following website. PDDCA version 1.4.1 comprises 48 patient CT images obtained from the Radiation Therapy Oncology Group (RTOG) 0522 study, a multi-institutional clinical trial led by Dr. Kian Ang. The dataset includes manual segmentations of various anatomical structures, including the left and right parotid glands, brainstem, optic chiasm, optic nerves (both left and right), mandible, and submandibular glands (both left and right), as well as manual identification of bony landmarks. As this dataset is openly available for research purposes, we have chosen to incorporate it into our study.

This experiment involves running the baseline implementation defined in section 2 on five random images from the dataset, specifically focusing on segmenting the mandible, the brainstem, and the left parotid gland. The choice of these three regions of interest was due to their volume in the images we had, the mandible and brainstem are both features that are longer in some axes than others and the parotid was an organ in the dataset that was more spherical i.e. occupies the three canonical axes equally. These choices serve as a reference point for comparing and evaluating the results.

4.2.1 Real Data Setup

This subsection provides an overview of the experimental setup, including the data collection process and the metrics used to evaluate the performance of different methods.

This subsection describes how the experiment is set up. Data for this experiment is collected by running the pipeline for all the chosen images with the discrete method multiple times to cover all the restrictions described and then running it using the baseline.

Before describing the pipeline, it is important to decide on a metric that is easy to understand and compare to then evaluate an arbritary segmentation against the ground truth. The evaluation metric chosen is the Sørensen–Dice coefficient, or Dice coefficient defined as

$$Dicecoefficient = \frac{2 \cdot |A \cap B|}{|A| + |B|}$$

where A is the segmentation retrieved from the system and B is the ground truth. Having defined the metric of comparison, the pipeline used can be shown as the pseudocode in Algorithm 1.

Algorithm 1: Pipeline for simulating Interactive Segmentation

```
Input  : segmentation: chosen CT image
threshold ← 0.95;
dice ← [0];
prev_slices ← [];
segmentation ← chosen_CT_image;
chosen_slice ← middle_slice(segmentation);
seedpoints ← [];
seedpoints ← simulate_user_input(seedpoints, chosen_slice);
segmentation ← segment(segmentation, seedpoints);
while dice [-1] < threshold do
    uncertainty_field ← calculate_uncertainty_fields(segmentation);
    chosen_slice ← batch_query(uncertainty_field);
    seedpoints ← simulate_user_input(seedpoints, chosen_slice);
    segmentation ← segment(segmentation, seedpoints);
    dice.append(dice_coefficient(segmentation, ground_truth));
    if chosen_slice in prev_slices then
        break;
    prev_slices.append(chosen_slice);
Output: segmentation: final segmented image
```

Algorithm 1 illustrates the pipeline employed for processing the input CT images. In this pipeline, the function `simulate_user_input` is utilized to simulate the perfect user interaction by perfectly annotating the selected slice based on the ground truth, specifically determining which organ is being segmented. The function `batch_query` can be implemented using either the baseline gradient-descent-based approach or the discrete batch query method, as outlined in Section 3.

Each iteration of the while loop corresponds to one step in the pipeline, where a slice is selected and presented to the user for annotation. During each iteration, the dice coefficient between the segmentation and its corresponding ground truth is recorded. The while loop continues until either the segmentation achieves a dice coefficient of at least 0.95 or when a previously perfectly annotated slice is encountered again for annotation.

This pipeline is repeated twice, once with the discrete batch query and once with the gradient-descent-based batch query, for each of the three selected features across all five chosen images. The results of these experiments are presented in Subsection 4.2.2.

4.2.2 Real Data Results

This subsection will present the data from the experiment with real data. There was an issue with the `simulate_user_input` function for slices retrieved using the gradient-descent-based batch query which resulted in false segmentations due to improper seed points, thus the data for this will be presented as a direct comparison to discrete batch query. The pipeline performance of the discrete method was recorded and will be shown.

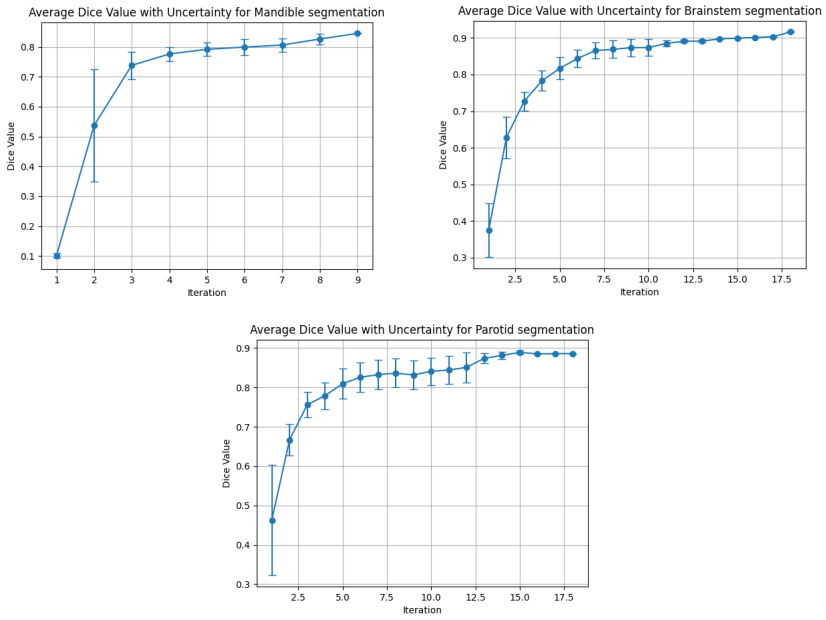

Figure 4: Results of experiment 1

Table 4: Frequency of chosen axis for each feature

Axis	Mandible	Brainstem	Parotid
$[1, 0, 0]$	41	61	63
$[0, 0.707, -0.707]$	0	6	4

The following table shows what baseline selects at each iteration of the pipeline on the image of 522c0002 while running using discrete batch selection.

Table 5: Baseline Gradient Descent Plane selection for mandible segmentation for CT scan of 522c0002

Iteration	Baseline Plane
1	[-0.07963096 0.14342765 -0.98645194] and point [5 78 44] = 852.3120528832836
2	[0.13184563 -0.59376647 0.793762] and point [5 62 109] = 1507.7850677041665
3	[-0.18278358 -0.57783129 0.79542514] and point [19 104 92] = 1473.1654093402556
4	[-0.32462209 -0.8953792 -0.30482224] and point [26 77 99] = 657.5410695195642
5	[0.23416316 -0.58087513 0.77958431] and point [27 45 41] = 1374.924708313807
6	[-0.28626193 0.67847399 -0.67655536] and point [21 32 74] = 1103.4101223308817
7	[0.00111328 0.82433538 0.56610065] and point [22 102 94] = 1103.7735635651325
8	[0.33323026 0.69267829 0.63965176] and point [13 104 34] = 1099.6196725034704
9	[-0.46800686 -0.27051476 -0.84130336] and point [27 97 126] = 796.1460434980742

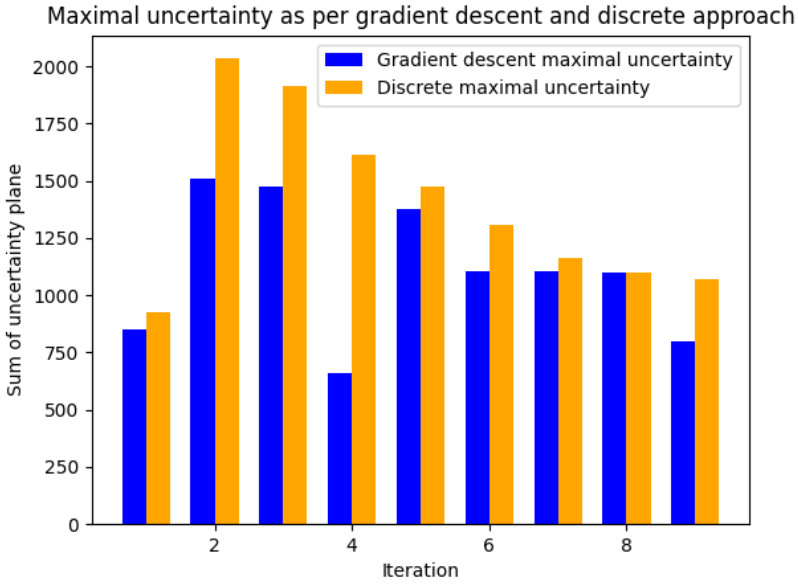

Figure 5: Baseline vs discrete maximal uncertainty at each iteration of pipeline

4.2.3 Real Data Discussion

Figure 4 consists of three graphs corresponding to the chosen feature, depicting the average value of the dice coefficient of the segmentation against the pipeline iteration (the number of annotated slices) for five CT images using the discrete batch query. Detailed data for these images can be found in Appendix Tables 7 to 21.

An interesting observation is that for all three features, the dice coefficient never exceeds 0.95, with the highest value being 0.91. Consequently, the termination criterion for these pipelines occurred when a previously encountered slice was presented to the user for annotation.

Table 4 displays the frequency of axes chosen by the discrete batch query. Notably, the

transverse plane (corresponding to the longitudinal axis, x-axis in the code) is chosen much more frequently than any other axis. The only other axis chosen is one of the diagonal axes defined by the normal vector (0, 0.707, -0.707), which occurs a total of 10 times. Specifically, for the mandible, only the x-axis is chosen. For the brainstem, the mentioned diagonal axis is chosen 6 times, while the x-axis is chosen otherwise. Finally, for the parotid, the diagonal axis is chosen 4 times, with the x-axis being chosen otherwise.

Figure 5 presents a direct comparison between the sum of the uncertainty of the plane selected by the baseline gradient descent and the plane chosen by the discrete approach. The pipeline from algorithm 1 utilized the slice selection of the discrete approach, while the baseline slice selection was also applied to the uncertainty field (computed from the segmentation using the annotations on the chosen slice from the discrete approach) to observe its choices, which are presented in Table 5.

Figure 5 demonstrates that the sum of uncertainty (SOU) of the plane chosen by the gradient-descent-based approach is at best equal to, and in most cases worse than, the SOU of the discrete approach. In some instances, for example, at iteration 4, the difference is significant. As mentioned earlier, the pipeline terminated when the discrete approach selected a previously annotated plane. However, in the case of mandible segmentation in patient 522c0002's image, the discrete approach chose the same plane on iteration 9 as it did on iteration 4. On the other hand, using the same uncertainty fields as the discrete approach, the gradient-descent-based approach selected two unique planes on iterations 4 and 9, as seen in Table 5.

5 Responsible Research

In this section, we will reflect on the ethical aspects of our research and discuss the reproducibility of our methods. Ethical considerations and reproducibility are important elements of responsible research, and it is crucial to address them in order to maintain transparency and integrity in our work.

5.1 Ethical Considerations

To ethically use the data from the PDDCA (Public Domain Database for Computational Anatomy) [7], it is important to adhere to the guidelines set by the database provider. This includes respecting data usage restrictions, maintaining patient confidentiality, acknowledging potential errors and their origin in the data, and properly citing the database as per their instructions. By following these guidelines explained by them on the website, we can utilize the data responsibly and contribute to scientific advancements while respecting privacy and intellectual property rights.

5.2 Reproducibility

Reproducibility is very important scientific research. It allows for the validation and verification of findings and if needed can be used for further extensions as well. In our study, we recognize the importance of reproducibility and have made efforts to ensure that our methods and experiments are reproducible.

To ensure reproducibility, we have provided detailed descriptions of our methodology, including the steps involved in comparing the query retrieval strategies and evaluating system performance in sections 2 through 5. We have documented the specific datasets used, the

preprocessing steps applied, and the implementation details of the algorithms or techniques employed.

Furthermore, we have made our code available in a publicly accessible repository on GitHub. By sharing our code, datasets, and experimental configurations, we hope to enable other researchers to replicate our experiments and validate our findings.

6 Discussion

This section aims to discuss the relevance of the obtained results from the experiments, highlighting the functionality of the methods and their significance in the context of active learning.

The experimental results yielded several key findings. Firstly, the performance of the discrete method was evaluated using different uncertainty fields. When the optimal slice aligned with one of the planes of symmetry, the discrete method successfully identified it. However, if the optimal slice was unaligned, the discrete method selected a sub-optimal slice with high uncertainty, necessitating multiple iterations to cover the entire optimal plane. In contrast, the baseline method consistently failed to identify the optimal slice and yielded slices with low uncertainty. Notably, the slices chosen by the baseline method had the highest uncertainty values, but their selection may not be intuitive from a human perspective, as discussed in the synthetic data analysis. The choice of these non-intuitive slices has both advantages and disadvantages. On one hand, it eliminates user control over slice presentation, which is essential for analyzing medical images as emphasized in previous works [5, 6, 12]. For instance, in shoulder scans, positioning each slice approximately parallel to the scapula while covering the complete shoulder anatomy is crucial for professional preference [12]. On the other hand, this method, when perfectly implemented, can find the slice with maximum uncertainty, reducing the iterations required for better segmentation. Overall, the discrete method outperformed the baseline method in most cases, effectively identifying the optimal slice.

Although synthetic data provided insights into the performance of the approaches, it is important to note that it is not definitive and further examination using real data is necessary to strengthen the arguments. Additionally, it is essential to mention that the baseline method did not function as intended in the pipeline due to an error in transferring seed points from the selected slice back into the segmentation algorithm. Hence, comparing the results of the discrete method with the faulty baseline method for experiment one was considered invalid, leading to a comparison with random slice selection as done in a similar study [11].

Several observations were made regarding the discrete approach's performance. Firstly, the dice coefficient values obtained from the segmentation pipeline, representing the accuracy of the segmentation, did not exceed 0.95 for any of the three features. The highest achieved value was 0.91, which triggered the termination of the pipeline upon encountering a previously annotated slice. This finding is significant as the goal of interactive segmentation is to achieve clinical standards. If a dice index of 0.8 meets the clinical standard, the discrete approach is valid. However, if a dice coefficient of 0.95 or higher is required, the current implementation of the pipeline using the discrete approach does not fulfill the criteria.

Furthermore, a frequency analysis of axes chosen by the discrete batch query in Table 4 revealed that the transverse plane (corresponding to the longitudinal axis) was by far the most frequently selected. A single diagonal axis was chosen in a few instances, while the x-axis was predominantly chosen for specific features. The only chosen diagonal axis was

the one corresponding to the normal $(0, 0.707, -0.707)$ from the center of the cube. This suggests that even with the ability to choose from many different axes, the choices are not needed as in all cases the most uncertain slices shared predominately the same orientation in the perspective of the discrete approach. This consistency and predictability in selecting slices, compared to the gradient-descent-based approach, offer significant benefits. Since the discrete approach allows users to define constraints, it enables the selection of slices of a certain orientation, as supported by the data in Table 6, where mandible segmentation achieved dice coefficients higher than 0.8 using slices from a single axis.

Table 6: Results for 0522c0002 Mandible

Iteration	Dice	plane
1	0.1106302708	[1 0 0] and point 8 = 928.3539969573221
2	0.7127848476	[1 0 0] and point 5 = 2034.704441525146
3	0.7832123919	[1 0 0] and point 10 = 1916.383405068218
4	0.7868319734	[1 0 0] and point 9 = 1614.7149295183817
5	0.813808325	[1 0 0] and point 3 = 1475.8687321730931
6	0.8370350368	[1 0 0] and point 7 = 1306.043667584752
7	0.8377065767	[1 0 0] and point 6 = 1160.6521058672924
8	0.8433694522	[1 0 0] and point 2 = 1099.6196725034704
9	0.8450185737	[1 0 0] and point 9 = 1068.5385625865815

Analyzing the data presented in Figure 4, it is evident that, on average, around five iterations (user annotations) with the discrete approach resulted in a segmentation with a dice coefficient of approximately 0.8. This performance is significantly better than random slice selection, which required 19 iterations before achieving a similar increase in the dice coefficient. However, the graphs also demonstrate that further iterations only yield slight improvements in the dice coefficient. Similar data exists for the gradient descent-based approach, and although not directly comparable due to their testing being done on different images, a similar pattern is observed in the graph presented by [11], where the dice coefficient reaches 0.95.

A comparison between the baseline gradient descent and discrete approach was conducted in terms of the sum of uncertainty (SOU) of the chosen planes. The discrete approach consistently performed on par with or better than the baseline, as illustrated in Figure 5. The gradient-descent-based approach often had a higher SOU compared to the discrete approach, indicating inferior plane selection. Interestingly, the discrete approach sometimes selected the same plane twice, while the baseline method found two unique planes using the same uncertainty fields. These findings imply that the baseline implementation of the pipeline if functioning properly, has the potential to achieve a higher dice coefficient by identifying unique planes, as observed in [11]. Furthermore, it is important to mention that, unlike the discrete approach where the chosen slices mostly had the same orientation the baseline approach chose slices whose normals were all different and of different orientations as seen in Table 5.

The gradient descent-based approach selecting worse planes only means that it would take at the very least one more iteration to reach a dice coefficient that the discrete approach would have reached. But the fact that it has the potential to reach a higher dice coefficient is a tradeoff that is worth considering.

Finally, it is important to highlight a few properties of the two approaches. Firstly, the discrete method is significantly faster than the baseline, with an average time of 11 seconds

to find a plane in real data experiments compared to the baseline's average of 600 seconds (based on [11]'s implementation, which found the most uncertain plane within 40 seconds on average, albeit on a different dataset). Secondly, the discrete approach appears to be deterministic, selecting the same slices in the same order assuming perfect user input. This characteristic enables additional features such as saving previous user input for redoing a previous annotation or pipeline reloading in case of crashes. In contrast, the gradient descent-based approaches yield different planes each time.

Overall, the experiments demonstrate the effectiveness of the discrete method in interactive segmentation. It consistently identifies slices with relatively high dice coefficients within a few iterations, although it struggles to improve beyond certain thresholds. Conversely, the baseline gradient-descent method shows worse performance in terms of SOU of its chosen plane, implying a lower improvement in the dice coefficient per iteration compared to the discrete approach, but with the potential to reach higher dice coefficients. These findings emphasize the advantages of the discrete method, such as speed, deterministic behavior, and the selection of slices from predefined axes, making it more user-friendly. However, it is crucial to stress that the discrete approach can only be considered valid if the required segmentation quality standard is approximately a dice coefficient of 0.85. If higher standards are necessary, the discrete approach may not be suitable.

7 Conclusions and Future Work

The main focus of this research study was to compare the effectiveness of discretizing the query retrieval strategy to its continuous alternative and investigate the impact of this discretization on system performance. By addressing these research questions, valuable insights have been gained into different approaches to query retrieval.

Based on the analysis and experiments conducted, several conclusions can be drawn. Firstly, discretizing the query retrieval strategy offers advantages over the continuous alternative. It provides a structured and manageable framework for organizing and processing queries, leading to improved efficiency and performance in terms of pipeline runtime and user experience. The discretization allows for better control and optimization of the retrieval process, enabling users to explicitly choose the orientation of slices they wish to annotate.

Furthermore, the experiments demonstrate that discretizing the query retrieval strategy enhances the overall performance of the system. The structured nature of the discretized approach enables more accurate and targeted retrieval of relevant information, resulting in higher precision and recall metrics compared to the continuous alternative.

However, it is important to note that the discrete approach has a major limitation. The current implementation of the discrete approach is unable to achieve segmentations with dice coefficients greater than 0.85 on average. Depending on the required precision, this limitation may render the discrete approach unsuitable for the intended application of the system.

n contrast, the gradient descent-based approach, despite lacking some of the advantages of the discrete version, proves to be superior in terms of the accuracy of the final segmentation and thus the intended application of the system.

The contributions of this work are twofold. Firstly, it provides a comparative analysis of discretized and continuous query retrieval strategies, adding to the existing literature. This analysis sheds light on the benefits and trade-offs associated with each approach, assisting researchers in making informed decisions when designing and implementing retrieval systems.

Secondly, the findings highlight the need for further research in this area. While the experiments show promising results for the discretized query retrieval strategy, there are still open issues and opportunities for improvement. It is crucial to investigate why the discrete approach failed to choose an unannotated slice properly and explore potential fixes. Additionally, further investigation is needed to ensure the correct implementation of the gradient descent-based approach, as there may be room for improvement or alternative interpretations of the algorithm.

In light of the findings, it is recommended to explore other research avenues that aim to refine and optimize the discretization process. Investigating different discretization techniques, hybrid approaches that combine the advantages of discrete and gradient descent-based methods, evaluating their impact on a wider range of queries and datasets, and developing adaptive approaches that dynamically adjust the discretization based on query characteristics are all promising directions for future investigation.

To conclude, this research demonstrates that discretizing the query retrieval strategy offers certain advantages over the continuous alternative, leading to improved system performance. However, it is important to note that the discrete approach exhibits significantly less precision in segmentations compared to the gradient descent approach, particularly in the context of clinical applications of the system. While there are still open issues and opportunities for further research, the findings of this study contribute to the understanding of query retrieval strategies and provide valuable insights for researchers and practitioners in the field of information retrieval.

References

[1] Plane symmetry. `https://www.math.net/plane-symmetry`. Accessed: 2023-05-20.

[2] Planes in anatomy. `https://teachmeanatomy.info/the-basics/anatomical-terminology/planes/`. Accessed: 2023-05-20.

[3] Samuel Budd, Emma C Robinson, and Bernhard Kainz. A survey on active learning and human-in-the-loop deep learning for medical image analysis. *Medical Image Analysis*, 71:102062, 2021.

[4] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. Ieee, 2016.

[5] PD Molyneux, PS Tofts, A Fletcher, B Gunn, P Robinson, H Gallagher, IF Moseley, GJ Barker, and DH Miller. Precision and reliability for measurement of change in mri lesion volume in multiple sclerosis: a comparison of two computer assisted techniques. *Journal of Neurology, Neurosurgery & Psychiatry*, 65(1):42–47, 1998.

[6] S Ostlere. Imaging the knee. *Imaging*, 19(3):249–268, 2007.

[7] Patrick F Raudaschl, Paolo Zaffino, Gregory C Sharp, Maria Francesca Spadea, Antong Chen, Benoit M Dawant, and Florian Jung. Evaluation of segmentation methods on head and neck ct: Auto-segmentation challenge 2015. *Medical Physics*, 44(5):2020–2036, 2017.

[8] Ahmed Saad, Torsten Möller, and Ghassan Hamarneh. Probexplorer: Uncertainty-guided exploration and editing of probabilistic medical image segmentation. In *Computer Graphics Forum*, volume 29, pages 1113–1122. Wiley Online Library, 2010.

[9] B Settles. Active learning literature survey univ. wisconsin, madison, comput. Technical report, Sci. Tech. Rep. 1648, 2010.

[10] Neeraj Sharma and Lalit M Aggarwal. Automated medical image segmentation techniques. *Journal of medical physics/Association of Medical Physicists of India*, 35(1):3, 2010.

[11] Andrew Top, Ghassan Hamarneh, and Rafeef Abugharbieh. Active learning for interactive 3d image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2011: 14th International Conference, Toronto, Canada, September 18-22, 2011, Proceedings, Part III 14*, pages 603–610. Springer, 2011.

[12] Yu Zhao, Ke Zeng, Yiyuan Zhao, Parmeet Bhatia, Mahesh Ranganath, Muhammed Labeeb Kozhikkavil, Chen Li, and Gerardo Hermosillo. Deep learning solution for medical image localization and orientation detection. *Medical Image Analysis*, 81:102529, 2022.

[13] Ervine Zheng, Qi Yu, Rui Li, Pengcheng Shi, and Anne Haake. A continual learning framework for uncertainty-aware interactive image segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6030–6038, 2021.

Appendix

Results on experiments on real data

Table 7: Results for 0522c0002 Mandible

Iteration	Dice	plane
1	0.1106302708	[1 0 0] and point 8 = 928.3539969573221
2	0.7127848476	[1 0 0] and point 5 = 2034.704441525146
3	0.7832123919	[1 0 0] and point 10 = 1916.383405068218
4	0.7868319734	[1 0 0] and point 9 = 1614.7149295183817
5	0.813808325	[1 0 0] and point 3 = 1475.8687321730931
6	0.8370350368	[1 0 0] and point 7 = 1306.043667584752
7	0.8377065767	[1 0 0] and point 6 = 1160.6521058672924
8	0.8433694522	[1 0 0] and point 2 = 1099.6196725034704
9	0.8450185737	[1 0 0] and point 9 = 1068.5385625865815

Table 8: Results for 0522c0002 Brainstem

Iteration	Dice	Plane
1	0.36549214520280465	[0. 0.707 -0.707] and point 41 = 949.0212866444924
2	0.6579589493325678	[1 0 0] and point 19 = 1138.8256415408896
3	0.7187264118575448	[1 0 0] and point 21 = 1017.8421307065672
4	0.7641652892561983	[1 0 0] and point 7 = 922.3217440396679
5	0.8058842759071592	[1 0 0] and point 22 = 883.3925931803518
6	0.8291025641025641	[1 0 0] and point 15 = 824.4214171405287
7	0.8733501135735773	[1 0 0] and point 14 = 780.1378752408523
8	0.8795613767895217	[1 0 0] and point 20 = 761.6912575501244
9	0.8832949168991872	[1 0 0] and point 18 = 746.2782929764014
10	0.8814841938239961	[1 0 0] and point 10 = 726.0172679625459
11	0.8844928956393925	[1 0 0] and point 13 = 719.7683038016435
12	0.8854765691912394	[1 0 0] and point 16 = 685.9864248070793
13	0.8873695134607167	[1 0 0] and point 20 = 614.468134901411

Table 9: Results for 0522c0002 Parotid

Iteration	Dice	Plane
1	0.550942573825869	[0. 0.707 -0.707] and point 36 = 940.4482828366497
2	0.7286045433472416	[1 0 0] and point 7 = 925.4680675689374
3	0.7697893972403776	[1 0 0] and point 16 = 829.514434633506
4	0.7976694915254238	[1 0 0] and point 12 = 750.4069951688805
5	0.8070267435762979	[1 0 0] and point 5 = 665.1128832900754
6	0.841224665804226	[1 0 0] and point 17 = 656.5589289769657
7	0.8484584567148581	[1 0 0] and point 15 = 615.3960706856908
8	0.8532754880694143	[1 0 0] and point 9 = 588.9498516087287
9	0.8588935876413222	[1 0 0] and point 13 = 588.8120047721409
10	0.859658208182289	[1 0 0] and point 11 = 569.4473257424775
11	0.8616975441619992	[1 0 0] and point 3 = 515.46595275138
12	0.8807089297886844	[1 0 0] and point 8 = 510.65672915447186
13	0.8835116318560027	[1 0 0] and point 18 = 500.39505260514
14	0.8920203735144312	[1 0 0] and point 6 = 471.28210227880317
15	0.8929814139013833	[0. 0.707 -0.707] and point 36 = 466.7628927830432

Table 10: Results for 0522c0017 Mandible

Iteration	Dice	plane
1	0.1096925013	[1 0 0] and point 12 = 1039.5580796987533
2	0.4203210161	[1 0 0] and point 10 = 1006.1269110452615
3	0.7820378151	[1 0 0] and point 7 = 1367.5008360770746
4	0.8005018821	[1 0 0] and point 11 = 1236.2293856881818
5	0.8107925548	[1 0 0] and point 9 = 964.0036810934739
6	0.8120689655	[1 0 0] and point 4 = 935.9778252775765
7	0.8206157965	[1 0 0] and point 14 = 856.7638170024911
8	0.8443700171	[1 0 0] and point 10 = 761.3117172783057

Table 11: Results for 0522c0017 Brainstem

Iteration	Dice	Plane
1	0.3984156570363467	[0. 0.707 -0.707] and point 40 = 1082.1614281368388
2	0.6956089683870567	[1 0 0] and point 19 = 1381.7820729576983
3	0.7778616644404437	[1 0 0] and point 21 = 1220.1127627520768
4	0.832047938744937	[1 0 0] and point 22 = 1116.8684064909135
5	0.8619499568593615	[1 0 0] and point 8 = 1024.7078333197642
6	0.8878943111350769	[1 0 0] and point 14 = 874.5418332955803
7	0.8933082504703037	[1 0 0] and point 18 = 856.9490985436113
8	0.8954694469823185	[1 0 0] and point 20 = 799.4048887591204
9	0.8968113136527397	[1 0 0] and point 17 = 738.8871546556288
10	0.8959827446751146	[1 0 0] and point 10 = 735.0546638989433
11	0.8926145079991233	[1 0 0] and point 15 = 685.8561121788218
12	0.8945410786566022	[1 0 0] and point 21 = 681.4715678445002

Table 12: Results for 0522c0017 Parotid

Iteration	Dice	Plane
1	0.5747981028073323	[1 0 0] and point 15 = 976.7539627821906
2	0.6536212862426918	[1 0 0] and point 8 = 914.884971565958
3	0.6925415010521393	[1 0 0] and point 13 = 772.8561702581675
4	0.7138751580641453	[1 0 0] and point 17 = 745.3937742789053
5	0.739404869251578	[1 0 0] and point 5 = 643.0292461353353
6	0.7553132302214309	[1 0 0] and point 10 = 665.0090230180144
7	0.7625994694960212	[1 0 0] and point 14 = 657.9366952476609
8	0.7665382498624106	[1 0 0] and point 12 = 652.9403089905813
9	0.7674214112991866	[1 0 0] and point 18 = 644.2696226558794
10	0.7817473567056205	[1 0 0] and point 16 = 639.1957529477589
11	0.7834913552704964	[1 0 0] and point 9 = 566.5979006900928
12	0.7867975022301517	[1 0 0] and point 15 = 508.6177746044813

Table 13: Results for 0522c0081 Mandible

Iteration	Dice	plane
1	0.1034009874	[1 0 0] and point 10 = 814.610643923965
2	0.7276654571	[1 0 0] and point 12 = 2100.955982305558
3	0.7475352113	[1 0 0] and point 6 = 2026.755291064384
4	0.7972908411	[1 0 0] and point 11 = 1586.3129913754585
5	0.7974716276	[1 0 0] and point 8 = 1459.285196164866
6	0.805999943	[1 0 0] and point 4 = 1351.7229024872563
7	0.8095562752	[1 0 0] and point 9 = 1260.711387035436
8	0.8126593755	[1 0 0] and point 10 = 1168.436432689367

Table 14: Results for 0522c0081 Brainstem

Iteration	Dice	Plane
1	0.502548778344173	[0. 0.707 -0.707] and point 37 = 1777.565412147556’
2	0.5257708550928659	[1 0 0] and point 19 = 1670.0890657309901
3	0.720584193789015	[1 0 0] and point 22 = 1570.2010212067476
4	0.7955291600575455	[1 0 0] and point 24 = 1420.6692185462039
5	0.8356850897458681	[1 0 0] and point 25 = 1337.378856920921
6	0.8480376655774569	[1 0 0] and point 26 = 1319.220836537815
7	0.8720734348678104	[1 0 0] and point 12 = 1213.1321220288255
8	0.8749570358149447	[1 0 0] and point 17 = 1105.3141008217244
9	0.8877731035418646	[1 0 0] and point 21 = 1013.0869981068186
10	0.8916765553483641	[1 0 0] and point 23 = 926.1624478354081
11	0.8914246901001868	[1 0 0] and point 15 = 922.6338097204034
12	0.8936213503340229	[1 0 0] and point 20 = 897.4729491099342
13	0.8950223789502237	[1 0 0] and point 13 = 868.5196200793118
14	0.8975962676222996	[1 0 0] and point 11 = 872.9472352570094
15	0.8992410895785337	[1 0 0] and point 18 = 852.2297634089603
16	0.9014541765302672	[1 0 0] and point 16 = 803.639489585325
17	0.903495748414091	[1 0 0] and point 6 = 798.3976919097819
18	0.9165220872814627	[1 0 0] and point 25 = 783.786619077585

Table 15: Results for 0522c0081 Parotid

Iteration	Dice	Plane
1	0.1923603713285649	[0. 0.707 -0.707] and point 50 = 1317.7686125141176
2	0.672031532271309	[1 0 0] and point 8 = 1624.416053972649
3	0.782218717658075	[1 0 0] and point 16 = 1367.9894642974734
4	0.8123822861834162	[1 0 0] and point 5 = 1300.9647337579731
5	0.8512166312308056	[1 0 0] and point 11 = 1167.7471405284002
6	0.8630258786092002	[1 0 0] and point 18 = 1111.6665785818952
7	0.8739690482809749	[1 0 0] and point 25 = 1134.694104572331
8	0.8739690482809749	[1 0 0] and point 25 = 1134.694104572331

Table 16: Results for 0522c0195 Mandible

Iteration	Dice	plane
1	0.08989136744	[1 0 0] and point 11 = 1706.029381099714
2	0.2313821637	[1 0 0] and point 7 = 1721.952074390366
3	0.6597178232	[1 0 0] and point 9 = 1794.7227526708557
4	0.7508636031	[1 0 0] and point 5 = 1550.3702642643548
5	0.784387563	[1 0 0] and point 10 = 1489.089541077894
6	0.7834608532	[1 0 0] and point 8 = 1379.615939336623
7	0.7904578363	[1 0 0] and point 3 = 1238.798069585089
8	0.8044973817	[1 0 0] and point 9 = 1218.6376945167553

Table 17: Results for 0522c0195 Brainstem

Iteration	Dice	Plane
1	0.3028009084027252	[0. 0.707 -0.707] and point 32 = 567.7348858659625
2	0.6226703330278033	[1 0 0] and point 17 = 696.1096927833858
3	0.7061523017352646	[1 0 0] and point 19 = 628.1258723857459
4	0.7604395604395604	[1 0 0] and point 8 = 566.8481515175365
5	0.7721639656816015	[1 0 0] and point 20 = 565.7631520292698
6	0.821479651540762	[1 0 0] and point 13 = 529.5951488229407
7	0.8245248633168446	[1 0 0] and point 18 = 479.2141287989443
8	0.8247557003257329	[1 0 0] and point 16 = 471.2204091617753
9	0.8288756167229291	[1 0 0] and point 14 = 419.31571688493887
10	0.8309804429478047	[1 0 0] and point 20 = 412.18430423354454

Table 18: Results for 0522c0195 Parotid

Iteration	Dice	Plane
1	0.45990650845019776	[1 0 0] and point 6 = 663.2816123491405
2	0.6042003231017771	[1 0 0] and point 12 = 645.5708534504085
3	0.7731326644370122	[1 0 0] and point 14 = 672.8542878842743
4	0.7901065864990434	[1 0 0] and point 4 = 604.5813985751467
5	0.8338839698691688	[1 0 0] and point 10 = 548.1618153870062
6	0.837884184144572	[1 0 0] and point 15 = 527.9403443644603
7	0.8402240597492665	[1 0 0] and point 3 = 523.2155888859262
8	0.848476821192053	[1 0 0] and point 8 = 511.76702944527676
9	0.8477366255144033	[1 0 0] and point 11 = 469.98961418151316
10	0.8516949152542372	[1 0 0] and point 5 = 465.9259885014278
11	0.8556728232189974	[1 0 0] and point 7 = 452.56605573312333
12	0.8583431797295523	[1 0 0] and point 2 = 447.9594677216323
13	0.8556008950901671	[1 0 0] and point 13 = 425.8305972425442
14	0.8684107770402186	[1 0 0] and point 12 = 402.3284243467042

Table 19: Results for 0522c0708 Mandible

Iteration	Dice	plane
1	0.09304286718	[1 0 0] and point 10 = 498.21718117331056
2	0.5915233415	[1 0 0] and point 7 = 1234.7908975331366
3	0.718943305	[1 0 0] and point 12 = 1055.184872049801
4	0.7446700273	[1 0 0] and point 9 = 922.3568692181763
5	0.7528027936	[1 0 0] and point 5 = 862.8497023549445
6	0.7577685232	[1 0 0] and point 11 = 850.8283550124384
7	0.7725546917	[1 0 0] and point 10 = 778.8028495707302

Table 20: Results for 0522c0708 Brainstem

Iteration	Dice	Plane
1	0.303840877914952	[1 0 0] and point 18 = 811.1077189318976
2	0.6408014571948998	[1 0 0] and point 21 = 928.9900904125998
3	0.7129565816678153	[1 0 0] and point 23 = 792.2661650368739
4	0.763822412939429	[1 0 0] and point 24 = 752.8119353330981
5	0.8090599508988675	[1 0 0] and point 16 = 742.3699978965182
6	0.83232885180034	[1 0 0] and point 9 = 734.4427929958435
7	0.8651469798142605	[1 0 0] and point 19 = 622.7478242102347
8	0.8695918055343219	[1 0 0] and point 15 = 574.1884798924236
9	0.8704536789935189	[1 0 0] and point 22 = 575.4430744986141
10	0.8687859057832248	[1 0 0] and point 20 = 567.9022438514637
11	0.8715596330275229	[0.707 -0.707 0.] and point 28 = 557.0458504282745
12	0.8244458721490524	[1 0 0] and point 17 = 517.3962571177124
13	0.8277021617293835	[0.707 -0.707 0.] and point 33 = 502.0961655323375
14	0.8297021517283213	[1 0 0] and point 19 = 510.6714451684536

Table 21: Results for 0522c0708 Parotid

Iteration	Dice	Plane
1	0.5359477124183006	[0. 0.707 -0.707] and point 37 = 1037.3191417042567
2	0.6728894762927474	[1 0 0] and point 10 = 993.8774403650411
3	0.7615165730985157	[1 0 0] and point 21 = 877.3368955630574
4	0.7796323359741558	[1 0 0] and point 7 = 777.3114153653911
5	0.8152295632698768	[1 0 0] and point 16 = 700.8806405538435
6	0.8315681082671373	[1 0 0] and point 12 = 659.1756495838703
7	0.8358033749082905	[1 0 0] and point 19 = 633.2888862149109
8	0.8384683098591549	[1 0 0] and point 5 = 614.1692963147702
9	0.8520958517338739	[1 0 0] and point 23 = 579.8073930458182
10	0.870550631991864	[1 0 0] and point 17 = 572.4514913696855
11	0.8755060728744939	[1 0 0] and point 15 = 567.8748248916406
12	0.8774714966084572	[1 0 0] and point 9 = 542.7984549676776
13	0.8825346648466126	[1 0 0] and point 20 = 530.6945652972627
14	0.8826655177366077	[1 0 0] and point 13 = 525.8893885034688
15	0.8848606434047431	[1 0 0] and point 11 = 491.0385598049983
16	0.8849824208940231	[1 0 0] and point 8 = 485.0737996845172
17	0.8856692009754698	[1 0 0] and point 3 = 473.9330450191751
18	0.8857142857142857	[1 0 0] and point 16 = 464.7644319444437

Table 22: Baseline results for 0522c0002 Mandible

Iteration	Dice	plane
1	0.1106	[-0.28179 -0.955 -0.086] and point 72 = 932
2	0	[-0.0459 0.9442 -0.326] and point 70 = 1544
3	0.00129	[0.7121 0.1873 -0.676] and point 153 = 943
4	0	[0.194 -0.797 0.571] and point 153 = 547

Table 23: Random slice select for 0522c0002 Mandible

Iteration	Dice	Plane
1	0.1106302708	[0, 0, 1] and point 68 = 110.6699604358203
19	0.6656254881	[-0.0459 0.9442 -0.326] and point 70 = 1544
34	0.6656254881	[0, 0, 1] and point 128 = 32.247902200398826