THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF MECHANICAL ENGINEERING

Predicting 2D Geometrical Shapes Using Partial Contour Data and Principal Component Analysis

ALEXANDRIA ZIEGLER
SPRING 2024

A thesis
submitted in partial fulfillment
of the requirements
for baccalaureate degrees
in Biomedical and Mechanical Engineering
with honors in Mechanical Engineering

Reviewed and approved* by the following:

Daniel Cortes
Professor of Mechanical Engineering
Thesis Supervisor and Honors Advisor

Jason Moore
Professor of Mechanical Engineering
Faculty Reader

* Electronic approvals are on file.

ABSTRACT

The inaccuracy of spinal needle placement drastically diminishes the treatment of patients with chronic, debilitating back pain. With that, it is imperative to formulate a method to create a more accurate representation of spinal anatomy from limited ultrasound data. The primary goal of this study is to determine if a statistical shape model is an accurate predictor for the shape of an unknown image. Utilizing airplanes, numerous images were inputted to create the shape model which includes an average image associated with numerous principal component modes. From there, a separate image not included in the training set was inputted to a prediction function utilizing the average image and mode shapes from the statistical shape model. To understand the accuracy of the model with partial data, the airplane outline was manipulated to have data removed.

From the study, it was found that about 1/3 of the outline points can be removed while still preserving an accurate prediction. For vertical slicing, up to 65% of the image could be removed while still preserving a highly accurate prediction. As for top-down horizontal slicing, the prediction was accurate through up to 70% of the airplane being removed. The bottom-up slicing was much less accurate, being accurate only through 35% of the airplane being removed. This is likely due to the heavy dependence on the shape of the tail within the prediction algorithm.

The findings of this study support the extension into the 3D prediction of spinal anatomy from ultrasound imaging data. With the extension, it is important to identify key landmark features on spinal anatomy, as well as to include a significant number of images in the training set to ensure high accuracy of the predictions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to start out by thanking Dr. Cortes who has supported me over the last two years in completing my thesis document. I greatly appreciate all the assistance in troubleshooting hundreds of lines of code to determine the small error causing catastrophic issues. I also want to extend my gratitude for the assistance provided as my Schreyer Honors Advisor in helping me figure out and satisfy all my requirements as a double major. Lastly, I want to extend my full appreciation that you chose me to work on such an important and valuable project that will hopefully improve the medical industry one day.

I would also like to extend my appreciation to Dr. Moore who was quick to accept being my thesis reader. I want to thank Dr. Moore for taking the time to read my thesis and for the valuable feedback that was provided to make this document the best it could be.

Lastly, I want to highlight my extreme gratitude to my friends and family who supported me throughout my college journey. Writing a thesis is quite a daunting task but having the support of friends and family made it a whole lot easier. Specifically, to my parents, I will forever be thankful for the love and support you have offered me through all the obstacles that I have faced. You always say I am ‘Supergirl’, and I will always strive to make that true. Thank you for always pushing me to be the best version of myself in everything that I do. I know for a fact that I would not have gotten to where I am today without your support and guidance, so thank you!

Chapter 1

Introduction

Back pain is a growing concern with an estimated 75% of the adult population experiencing lower back pain at least once in their lives. Lower back pain is the top reason for decreased activity in those 45 and younger and is the third most common surgical procedure performed [1]. In addition, nearly a quarter of the population reports chronic back pain. Chronic back pain is debilitating, interfering with daily life and activities that one enjoys. Thus, it is important to continue to improve and develop treatments to heal those afflicted with chronic back pain so that they can return to the things that they love. Standard treatment options such as over the counter painkillers, anti-inflammatory medications, and physical therapy are the first treatment options. While these treatments may work for minor pains from an awkward night of sleep or a strained muscle, they often do not treat the root cause or location of the pain [2].

When conservative treatments are unsuccessful, doctors will recommend more extreme measures including epidural corticosteroid injections, radiofrequency ablation, non-fusion stabilization, and other surgical interventions. The goal of the surgical intervention is to reduce pain, improve mobility, and enhance overall happiness. While the surgery may not eliminate all pain, it reduces it significantly to improve the quality of life substantially. It is important to note that certain surgical interventions are quite dangerous, thus, they should be reserved for extreme, debilitating cases. The importance of finding and improving less invasive approaches is imperative. Less invasive options include the epidural corticosteroid injection and radiofrequency ablation procedure which offer relief with less risk to the patient [1].

1.1 Literature Review on X-Rays and Ultrasounds

With less invasive approaches, an X-ray scan or ultrasound image are used for guidance of needle placement. X-ray scans pose unnecessary amounts of radiation which can be dangerous for the patient. High exposure to radiation from X-ray imaging in youth has been linked to higher occurrences of breast cancer, prostate cancer, and leukemia [3,4]. In addition, X-rays are a much more expensive option for a procedure. Using ultrasound as guidance during injections proves to be a much safer and cheaper option for needle placement. However, the placement accuracy is very variable and often inaccurate. In a study performed by Ashmore *et al*, there was a significant error in injection placement during ultrasound guided injections when confirmed by an X-ray. The study found that 11% of ultrasound-guided injections for MBB had incorrect needle positioning. For FJI ultrasound-guided injections, 13% had inaccurate needle placement. Each injection must be confirmed via a CT or X-ray scan, still exposing the patient to radiation [5].

In a study performed by Ungi *et al*, some of the shortfalls of current ultrasound imaging techniques are highlighted. The study investigated using sonography rather than radiography to determine the curvature angle of the spine. The experiment was striving to achieve a more accurate and safer methodology to determine curvature angles in patients afflicted with scoliosis. The results determined the average error to be 1.27 +/- 0.84 degrees in adults and 0.96 +/- 0.87 degrees in children. The maximum error during experimentation was 3.4 degrees. Depending on the application, an error of up to 3 degrees can be quite significant and dangerous [3].

The study by Ungi *et al* highlights several limitations involved in sonography technology. First, the model being used was rigid, which is not fully descriptive of a patient who is moving. The movement of the body alters the curvature of the spine, which is unaccounted for with ultrasound technology. In other words, ultrasound captures a live image, independent of the orientation of the patient. Movement by the patient will cause different orientations which can be hard to account for within the model. Second, the study utilized manual marking of landmarks for the program to determine the curvature angle. Not only does this include inherent bias, but it also is very time-consuming for the operator [3].

Further limitations of ultrasound imaging are described in a study performed by Brudfors *et al*. The study was focused on improving the current procedure for administering anesthesia into the spine. The study highlights that ultrasound imaging provides a 2D image that is of relatively poor quality and resolution. Due to the 2D nature of the scan, it is extremely difficult to obtain visualization of deep, intricate structures in the body. This is especially emphasized by the fact that an ultrasound can only obtain an image of what it is scanning. For example, when an ultrasound is used on the back, it only obtains the posterior view. With these limitations, the study focuses on incorporating image enhancement, 3D imaging, and continuous anatomy localization. With the improvements made, the study was able to successfully administer the epidural 92% of the time. The epidural had a 3mm error whereas the facet joint had a 7mm error which falls outside the acceptable range. The study suggests that the error is due to limitation in the sample size and subjectivity to rotation error. This highlights that ultrasound is extremely sensitive and limited by the rotation of the subject [6]. Even with a 92% success rate, the 8% margin of error is far too high when it comes to injections into the spine. That means 8% of people under this methodology would experience an unsuccessful attempt at needle placement. This produces unnecessary pain, discomfort and aggravation for the patient who is already in significant pain.

The limitations of current ultrasound technology provide the motivation to improve the ultrasound imaging process to produce a more accurate, comprehensive image of the spine. This would enable more accurate needle placement and inherently improve the success rate of the procedure. As a result, the patients will be provided with a more pleasant and successful experience during what is typically a stressful situation. In addition, by increasing the use of ultrasound, the need for X-ray or CT imaging will be eliminated, thus decreasing cost, and improving patient safety.

1.2 Statistical Shape Analysis Overview

A statistical shape model is taking a set of images to create an average image with variation principal component modes. The model average is taken as the mean of all the points within the training set images. The variation modes are calculated via a complex mathematical algorithm including the signed distance function. The signed distance function determines the distance between a point on an image to a specified boundary. From the algorithm, several modes are outputted as eigenvalues and eigenvectors. These modes describe different aspects of the image. Taking specific weights and multiplying them by the modes will produce different airplanes from the mean airplane.

The use of a statistical shape model is not a new concept. In a study performed by Peloquin *et al*, the mechanical implications of degeneration were investigated. The current methodology utilizes finite element analysis (FEA). The largest limitation of this current methodology is the use of morphometry, which specifies the distances between two points. From theses distances, a rough shape is constructed. While this is relatively accurate in 2D, the 3D extension requires significant approximations and assumptions to fill in voids. This results in an inaccurate representation of the structure. To combat this issue, the study took a sample of the vertebrae and created a mean model. The mean model included shape variability using principal components. The principal components describe different changes within the actual structure. For example, PC1 in the study represented the overall size (diameter) of the vertebra. PC2-4 described different combinations of height, width, and other dimensional variations [7]. The study highlights the possibility and functionality of using the statistical shape model approach to create a working model. It provides supporting evidence to utilize this methodology and extend it to predicting spinal anatomy from samples not included in the training set. One limitation of this study is that the data used for the training set was fully known. In other words, several views of the spine were readily available, thus the entire anatomy and structure of the vertebra was known. From an ultrasound image, the full anatomy is not available, which may pose some issues.

Another study performed by Abler *et al,* investigated the use of a statistical shape model in predicting the shape of the premorbid glenoid cavity. A statistical shape model was created from 64 healthy scapulae. Then, scapula not in the training set were sent into the model and the glenoid cavity was predicted. The methodology had a surface distance accuracy of 1.0+/-0.2mm. In addition, the prediction error for glenoid version and inclination was 2.3+/- 1.8 degrees and 2.1+/-2.0 degrees, respectively [8]. The error is quite small and within the acceptable tolerance ranges. From this study, a proof of concept in using a statistical shape model approach is provided. This provides support in using the mode but with partial data inputs. This can be extended to other parts of the body, such as spinal vertebra. In addition, the study highlights ways to check, verify, and confirm that the results are within the tolerances. For example, a root mean squared value was taken to compare the surface distance accuracies of the predicted cavity to the original cavity.

With proof that a statistical shape model is a viable approach to predict spinal anatomy, it is necessary to understand how to construct the model. In a study performed by Tsai *et al*, the 2D case of a model was investigated and created. The paper specifies the methodology, equations, and mathematics involved in creating the statistical shape model. The study took a series of 12 airplanes to produce a mean airplane. With the average airplane and several modes, an airplane not in the training set can be predicted by a combination of the modes and weights added to the mean airplane. For example, one mode may describe the length of the wings. The airplane to be predicted may have slightly longer wings, thus this mode will be multiplied by a greater value to produce longer wings. This paper provides the foundation to be referenced when creating the statistical shape model [9].

One limitation to note is that the study performed the statistical shape model and prediction in 2D. Spinal anatomy is in 3D, which is an important factor to consider. With the goal of creating a model to be used in needle injections, the accuracy in 3D must be high. In addition, the information provided to the model will be limited to the orientation that the ultrasound can capture. With that, the model must be able to predict the structure accurately from limited information. The study provides significant guidance and instruction to create a successful statistical shape model.

1.3 Statistical Shape Model Mathematical Constructs

Much of the initial research followed the steps outlined in Tsai *et al* to formulate an average image with the corresponding modes and weights. This will briefly highlight the important principles and mathematics for general understanding of the analysis. For a more in-depth explanation, refer to the research performed by Tsai *et al* [9].

First, the signed distance function is an important mathematical construct used in calculating a statistical shape model. In general terms, the function determines how far a point is from the boundary. For example, imagine a white circle surrounded by black. Locate a point from the center of the circle. This falls in the middle of the white section of the circle. The signed distance function will calculate how far that point is to the nearest black (exterior) point. The signed distance function is an integral part of calculating the mean shape amongst a group of images.

Next, the idea of eigenvalues and eigenvectors must be discussed. The eigenvectors for each calculated eigenvalue represent principal modes of an image. They describe certain aspects of a system, or in this case, an image. Each eigenvalue-eigenvector pair represents a different shape mode. For example, the first eigenvalue may represent the width of an image. The second eigenvalue may represent the height of an image, and so on. The more eigenvalues and eigenvectors that are included, the more shape modes there are. This can be beneficial to capture all the different variations that may occur within a group of images. However, including too many eigenvalues can lead to the inclusion of noise. The noise will produce a final product with significant distortion which is not ideal. It is imperative to utilize a process to determine enough eigenvalues to capture all the variation and shape modes, without including too many that there is significant noise distortion.

Lastly, the idea of cross-correlation plays a vital role in understanding how similar two things are. Visually, two things may appear to be similar; however, there is no numerical value that is obtained that can be used for comparison. The cross-correlation takes the first input and slides it across the second input to find the maximum position of similarity. This maximum value creates a numerical representation as to how similar two things are.

1.4 Objective

The goal of this study is to extend upon previous research to apply a statistical shape model and predicting algorithm to predict new images not in the training set. Once a strong prediction algorithm is generated, it is important to see how resilient and versatile the model and algorithm are in their predictions. First, the airplane to be predicted will be tested using less points in the airplane's outline each time. Second, the airplane will be cut vertically and horizontally to see if the model can predict the shape accurately with only partial information. This is an imperative step because an ultrasound image provides very limited information. It is critical to ensure that the model is successful even with a limited knowledge base. This study will determine proof of concept that may be extended to 3D if the methodology is confirmed.

Chapter 2

Methodology and Materials

This research will utilize MATLAB R2023a (academic use) to create the appropriate algorithms. MATLAB R2023a was selected due to its powerful computational ability for matrix mathematics.

2.1 Initial Statistical Shape Model

The first step of this study was to solidify a code that can predict a shape from an outline in 2D. To perform this, images from Tsai *et al* were utilized (Figure 2-1). The airplane images were approved for use in this thesis by the IEEE database [9] (Appendix A). The following description of processes can be found in more detail outlined in the paper by Tsai *et* al. The initial portion included preprocessing of the images. This required importing the images into a Word Document and cropping them to relatively the same size. This was performed on 11 airplanes.

With the images the same size, they were imported into MATLAB utilizing a user selection script. The user can pick any number of airplanes to develop the average airplane and varying shape modes. By structuring the code this way, it allows for easy selection of the airplanes for inclusion in the statistical shape model.

The initially selected airplanes are sent to a function to perform the statistical shape analysis. Since the images being entered are not guaranteed to be centered or properly oriented, the code performs a centering and rotation process. Using the functions *imrotate* and *circshift*, the images were appropriately oriented and centered.

With the images appropriately oriented and centered, the signed distance function was employed on each image. The signed distance function calculates how far a point on an individual airplane falls from the image border. A negative signed distance function value represents a point within the airplane, designated in white. A positive signed distance function value represents a point on the outside of the airplane, designated in black.

Next, the mean was taken of all signed distance functions calculated. This will be used to determine the shape variabilities. The mean signed distance was subtracted from each independent signed distance function.

$$\Phi = \frac{1}{n}\sum_{i=1}^{n} \psi_i$$

$$Where\ \Phi\ is\ the\ average\ signed\ distance\ function, n\ is\ the\ number\ of\ images\ selected$$

$$by\ the\ user\ and\ \psi\ is\ the\ signed\ distance\ function\ for\ each\ image$$ [9]

$$\tilde{\psi} = \ \psi_i - \Phi$$

$$Where\ \tilde{\psi}\ is\ the\ mean - offset$$ [9]

For visualization of the average airplane from the selected images, the average signed distance function was also used. If this average value was greater than 0, it was determined to be on the outside of the averaged airplane and assigned a value of 0. If the averaged value was less than 0, it was determined to be on the inside of the averaged airplane, thus assigned a value of 1.

Next, the shape variability matrix, *S*, could be formed with the $\tilde{\psi}_i$ as a column matrix. With eigenvalue decomposition and reshaping of the vectors, the eigenvectors and eigenvalues could be taken from W.

$$W = \frac{1}{n} S^T S$$

Next, implementing the following equation, the corresponding weights for each eigenvector can be determined. The paper does not define how many eigenvectors or eigenvalues (k) to include. This is one aspect that this research tries to quantify and optimize. It is important to use enough to capture the major mode shapes, yet not too many that noise or intricate detail impacts the final predicted shape.

$$\bar{\Phi}[w] = \bar{\Phi} + \sum_{i=1}^{k} w_i \Phi_i$$

This is where this thesis diverges to advance, progress, and improve upon the research performed by Tsai *et al*.

2.2 Prediction Algorithm

To eliminate some of the errors due to a limited training group size, additional airplanes were added to provide more training for the model. These new airplanes were created as extensions of the already existing airplanes. The original 11 airplanes can be seen in Figure 2-1. 10 additional airplanes were created by adding black or white to the existing 11 airplane sample (Figure 2-2).

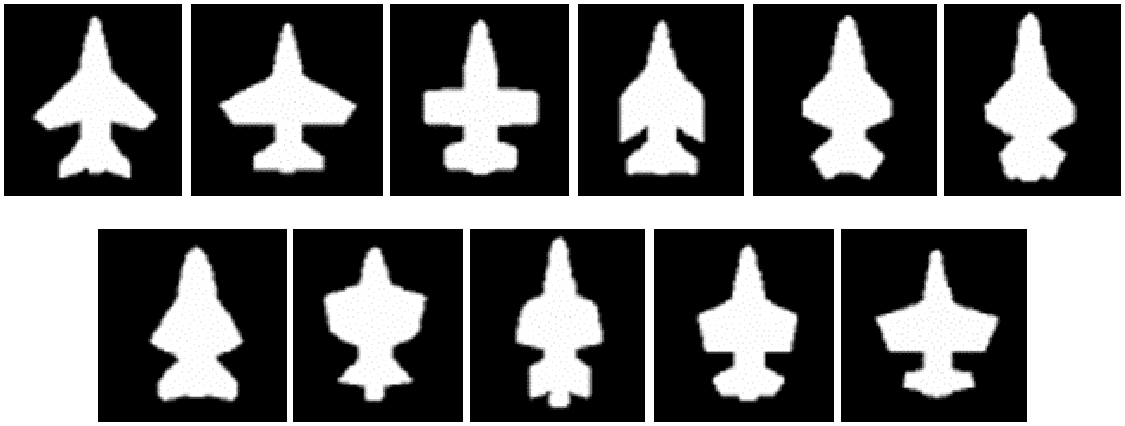


Figure 2-1. 11 Airplanes Utilized in the Shape Model and Predictions © [2003] IEEE

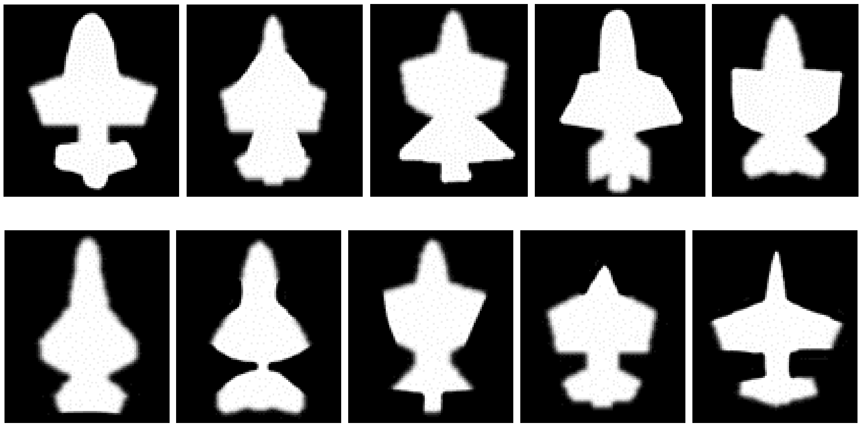
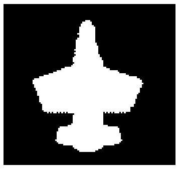
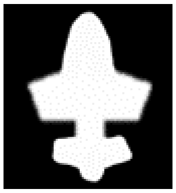
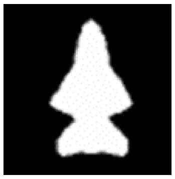
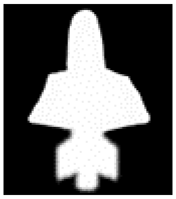
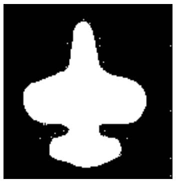


Figure 2-2. 10 Airplanes Modified from Original Images Utilized in the Shape Model and Predictions

For clarity, the airplanes referred to as Airplane 1, Airplane 2, Airplane 3, Airplane 4, and Airplane 5 are summarized in Table 2-1.

Table 2-1. Airplane Identification

Airplane 1	
Airplane 2	
Airplane 3	
Airplane 4	
Airplane 5	

The next step is to be able to predict an airplane not included in the training set. This begins by allowing the user to select any airplane within the folder. It is imperative to select an airplane that is not part of the initial training set. The goal is to optimize the function's ability to predict an entirely unknown, new airplane. Like the statistical shape model code, the airplane must be appropriately sized, centered and oriented. The program includes appropriate functions and steps to align the airplane correctly for analysis.

Again, the signed distance function will be calculated for the new airplane. The inside of the airplane is still defined by a negative value and the outside of the airplane is defined by a positive value. Next, the outline of the image is extracted. This is an important step because this is what will be used to optimize and predict. This is to simulate limited information provided by an ultrasound image. The image outline was calculated by locating where the signed distance function equaled 1. These locations were set to be 1 on the image and everywhere else was set to be 0. This makes all but the outline of the image to be black.

As was talked about previously, the weights of each mode must be calculated for this new airplane. Utilizing an optimization function, *fminsearch*, these weights were calculated to minimize a specific parameter. An external function was created, labeled the objective function. From the script, the average image, new image outline, weight guesses, and eigenvectors were sent to the function.

Within the objective function, the weights are multiplied by the eigenvectors. These weights are what the function is trying to optimize. This multiplication will be added to the average airplane to construct a predicted airplane. To determine how similar or different the prediction is from the actual airplane, an objective function value is calculated. The objective function takes the predicted airplane and multiplies it by the true image's outline (which was calculated previously). Then, to standardize the value regardless of the size of the input airplane, this value is divided by the number of points in the airplane. The goal of *fminsearch* is to change the weights (w) to minimize the value of this objective function value.

With the minimized weight values, the predicted airplane can be constructed using the eigenvectors and average airplane calculated during the preliminary statistical shape model steps. Now that an airplane has been predicted, it is important to see how accurate it is to the originally selected airplane.

To compare the similarities between the actual image and predicted image, several tests were utilized. First, a cross-correlation was calculated. This takes the predicted airplane and slides it across the actual airplane. The maximum of the cross-correlation represents the maximum similarity between the two images.

Another parameter utilized to measure similarity was the dot product. If the images are completely identical, their dot product will be equal to 1. If the images deviate from each other, the dot product will decrease. Thus, a dot product closer to 1 represents high similarity between images.

In addition, three built-in MATLAB functions were incorporated to calculate various aspects of image similarity. The *immse* function calculates the error between images. The *ssim* and *psnr* functions calculate different aspects of similarity between two images. The wide variety of tests were calculated because it was unsure upfront which would be most accurate and descriptive of the image similarity. All seemed like plausible options; thus, each was calculated for later analysis and determination of the best methodology and parameter to measure similarity.

2.3 Determining the Number of Eigenvectors

From this analysis of image similarity, the number of eigenvectors to use could be determined for the remaining analysis. As discussed previously, the research performed by Tsai *et* *al* did not specify what number of eigenvectors was suitable. It is known that too few eigenvectors will not appropriately capture the main modes of variations. It is also known that too many eigenvectors will include intricate detail or noise that will distort the image. However, it is difficult to understand what number of eigenvectors is truly optimal. In this case, image similarity analysis comes into play.

An initial number of eigenvectors was set to 2, which is known to be too few. However, this will provide a baseline for comparison of the similarity values. The number of eigenvectors was increased by one until it equaled the number of airplanes selected. Again, it is known that the number of eigenvectors should be less than the number of airplanes selected, but this will provide a top bound for comparison.

At each set of eigenvectors, the cross-correlation, dot product, immse, ssim, and psnr was calculated. This was then repeated, but a different airplane was left out of the training set to be the predicted airplane. By performing this analysis on several airplanes, an optimal number of eigenvectors can be determined.

This methodology was performed on four different airplanes. An example airplane eigenvector analysis can be seen in Table 2-2. Based on the similarity values obtained (Table 2-3), the ideal number of eigenvectors for this airplane was 4-9. The same analysis performed on the other three airplanes can be seen in Appendix B. From this, it was determined that 6 eigenvectors would be optimal since it fell within the ideal range for all airplanes tests. Thus, for the remaining research, 6 eigenvectors were utilized.

Table 2-2. Determining the Sufficient Number of Eigenvectors for Airplane 1

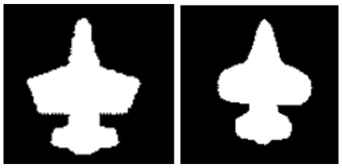
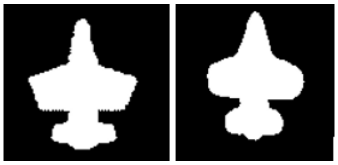
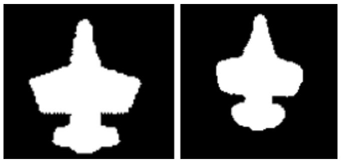
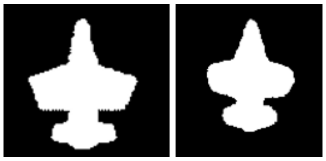
Number of Eigenvectors	Results (Initial Airplane, Prediction)
5	
6	
7	
8	

Table 2-3. Similarity Values for Each Eigenvector of Airplane 1

Eigs	xcorr	sim	Err	psnr	Dot
2	0.9261	0.8661	0.0243	16.1378	0.9314
3	0.9307	0.8716	0.0229	16.4017	0.9352
4	0.9306	0.8738	0.0222	16.5344	0.9373
5	0.9295	0.8701	0.0231	16.3692	0.9351
6	0.9349	0.8718	0.0224	16.5008	0.9370
7	0.9309	0.8778	0.0201	16.7530	0.9416
8	0.9287	0.8733	0.0220	16.5795	0.9391
9	0.9315	0.8785	0.0206	16.8605	0.9434
10	0.9124	0.8655	0.0258	15.8890	0.9277
11	0.9247	0.8747	0.0225	16.4786	0.9378
12	0.8948	0.8457	0.0323	14.9064	0.9092
13	0.7701	0.7652	0.0771	11.1275	0.8091

2.4 Removal of Outline Points

As stated, and explained extensively in the introduction, an ultrasound image is very limited in information. To simulate the limitation, dots from the outline were excluded to reduce the available information. This was performed by using a *for loop* to take a certain portion of the total outline. Initially, every other point was removed which will create an outline with half the total number of points. The loop was edited to remove a third, a quarter, a fifth, and so on until the prediction was outside acceptable ranges.

With each outline with less points, the predicted image was compared to the initial input image. Using the same functions as were used to evaluate the eigenvectors, the image similarity was calculated. This included the dot product, cross-correlation, similarity value, psnr value, and error between images. By comparing these similarity values, the minimum number of outline points to create an accurate prediction could be determined.

2.5 Slicing Airplane

To simulate the limited information provided by the ultrasound, the airplane was sliced in various ways. Typically, the image will only capture a certain angle of the vertebra, thus having very limited information. It is important to see if the model can accurately predict the rest of the shape of the vertebra with extremely limited information. In addition, it is critical to determine the minimum information needed to be sent into the model to ensure an accurate prediction. The image to be sliced is the initial, user-selected input image that is used for creating an outline. The outline of the input airplane is calculated, then sliced appropriately.

To perform this test, the airplane image was first sliced vertically. This was done by setting a certain number of columns of the airplane to zero. First, only 5% of the airplane was removed. Then, the prediction was run with corresponding similarity calculations. As stated earlier, the similarity values were evaluated to determine if they fell within an acceptable range. The final similarity comparison was made between the model's predicted airplane and the initial user-selected input airplane without slicing (prior to data removal).

The airplane was then sliced to eliminate 10% of the data. The data in the airplane input image was removed in 5% increments, with a corresponding similarity analysis to follow. This was performed to determine the minimum amount of airplane in the vertical direction required to create an accurate final airplane prediction.

Following the vertical slicing, a horizontal slicing algorithm was performed. This is to simulate that the information from the ultrasound may be limited in multiple directions. The process for horizontal slicing is the same as that of vertical slicing. 5% of the airplane from the bottom was removed, followed by a similarity analysis. This increased in 5% increments until the model failed to accurately predict the airplane. The horizontal slicing was then performed in a top-down algorithm. Since the airplane is not symmetrical across the x-axis, understanding how limited data in each direction impacts the prediction is important.

Chapter 3

Results

The results will be broken into several sections to analyze 1) missing outline points, 2) vertical slicing, 3) and horizontal slicing.

3.1 Removal of Outline Points

Points from the outline of three different airplanes were removed to understand the impact missing points would have on the airplane prediction effectively simulating missing data from an ultrasound image. The first airplane to be predicted, shown in Figure 3-1 was selected due to its strong similarity to the average airplane.

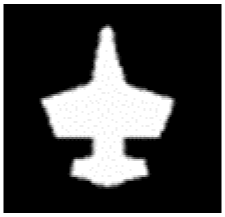
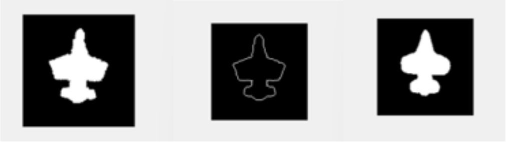
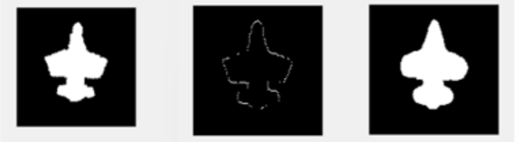
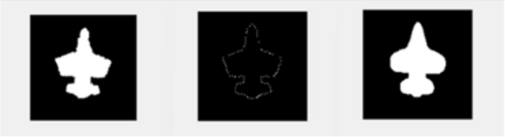
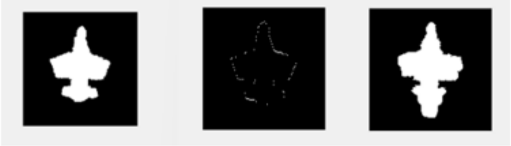
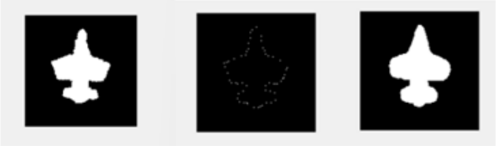
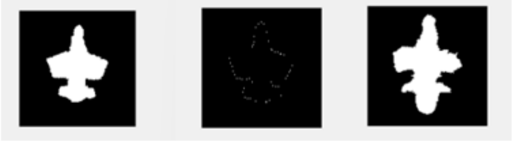


Figure 3-1. Airplane 1 Original

A baseline similarity value was obtained for comparison which utilized the entire airplane outline. Then, 1/2 the points were removed, 1/3 the points, 1/4 the points, and so on. The visual representation was captured comparing the original airplane, the outline with removed points, and the final airplane (Table 3-1). See Appendix C for all images of the airplane with removed points.

Table 3-1. Airplane 1 with Outlined Removed

Full Outline	1/2 Outline
	
1/3 Outline	**1/4 Outline**
	
1/5 Outline	**1/6 Outline**
	

The cross-correlation, similarity value, error value, psnr value, and dot product were recorded for each test (Table 3-2).

Table 3-2. Similarity Values for Removal of Outline Points for Airplane 1

Portion of Outline	xcorr	sim	Err	psnr	Dot
Full	0.9327	0.8759	0.0214	16.6944	0.9394
½ Points	0.9255	0.8652	0.0242	16.1584	0.9319
1/3 points	0.9304	0.8775	0.0213	16.7060	0.9394
¼ Points	0.8767	0.8475	0.0386	14.1313	0.8932
1/5 Points	0.9302	0.8710	0.0224	16.4897	0.9367
1/6 Points	0.8136	0.7749	0.0711	11.4843	0.8208
1/7 Points	0.9134	0.8637	0.0274	15.6263	0.9246
1/8 Points	0.7955	0.7777	0.0680	11.6780	0.8179
1/9 Points	0.9135	0.8605	0.0269	15.6998	0.9247
1/10 Points	0.8110	0.7727	0.0717	11.4424	0.8244
1/11 Points	0.6835	0.7046	0.1256	9.0090	0.7297
1/12 Points	0.8483	0.8017	0.0537	12.7034	0.8625
1/13 Points	0.6248	0.6757	0.1418	8.4828	0.6978
1/14 Points	0.7175	0.7331	0.1050	9.7893	0.7631

Two other airplanes were analyzed with the same methodology. The second airplane was selected due to its large variation from the average airplane. The third airplane was selected due to its large initial prediction accuracy. See Appendix C for the tables and figures associated with airplanes 2 and 3 for the removed point analysis.

With the cross-correlations calculated for each of the airplanes, the values can be plotted for analysis (Figure 3-2). This can help determine the cutoff for points removed when the prediction is no longer accurate.

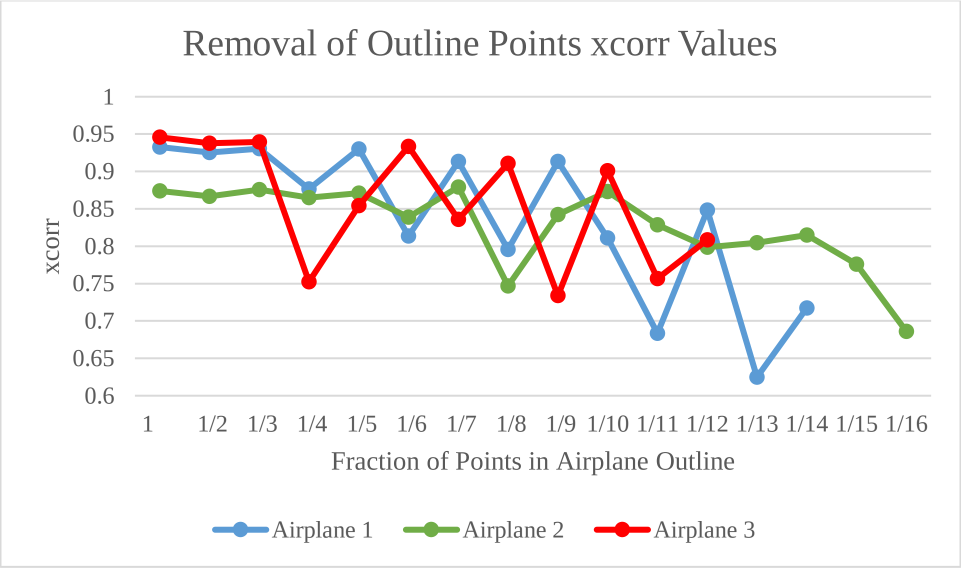
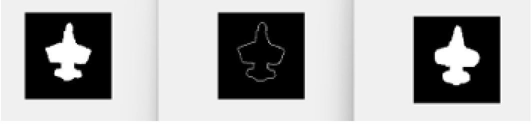
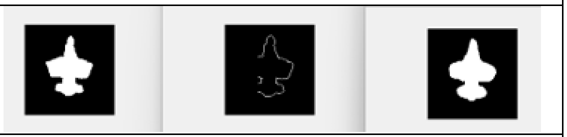
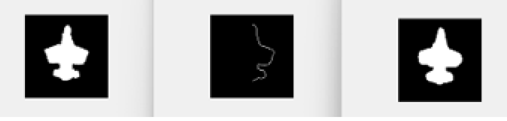
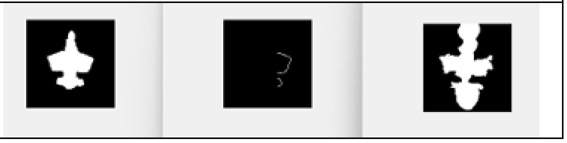


Figure 3-2. Removal of Outline Points xcorr Values

3.2 Vertical Slicing

Airplane 1 was used to perform vertical slicing removal of the airplane's outline. Vertical slicing means removing columns of the airplane's outline. A baseline similarity value was obtained for the entire airplane outline. Then, 5% of the airplane was removed at a time and similarity values were recorded. Representative images for this process were obtained for airplane 1 (Table 3-3), and the rest of the airplane 1 images can be found in Appendix D.

Table 3-3. Airplane 1 Vertical Slicing

100% Outline	75% Outline
	
50% Outline	**30% Outline**
	

The cross-correlation, similarity, error, psnr and dot product were recorded for each 5% sliced interval of airplane 1 (Table 3-4). The images and data tables for airplanes 2 and 3 can be found in Appendix D.

Table 3-4. Similarity Values for Vertical Slicing of Airplane 1

Percent (%)	xcorr	sim	Err	psnr	Dot
100	0.9349	0.8718	0.0224	16.5008	0.9370
95	0.9312	0.8700	0.0230	16.3909	0.9356
90	0.9318	0.8690	0.0231	16.3584	0.9351
85	0.9328	0.8705	0.0227	16.4455	0.9362
80	0.9341	0.8723	0.0223	16.5120	0.9369
75	0.9330	0.8706	0.0228	16.4236	0.9357
70	0.9310	0.8693	0.0235	16.2942	0.9342
65	0.9309	0.8679	0.0235	16.2836	0.9341
60	0.9310	0.8688	0.0235	16.2836	0.9341
55	0.9317	0.8704	0.0231	16.3584	0.9351
50	0.9277	0.8645	0.0247	16.0768	0.9311
45	0.9351	0.8707	0.0228	16.4236	0.9355
40	0.9317	0.8732	0.0224	16.5008	0.9346
35	0.9247	0.8697	0.0240	16.1893	0.9290
30	0.6631	0.7029	0.1266	8.9754	0.7259
25	0.9094	0.8560	0.0293	15.3273	0.9121

With the cross-correlations calculated for each of the airplanes through vertical slicing, the values can be plotted for analysis (Figure 3-3).

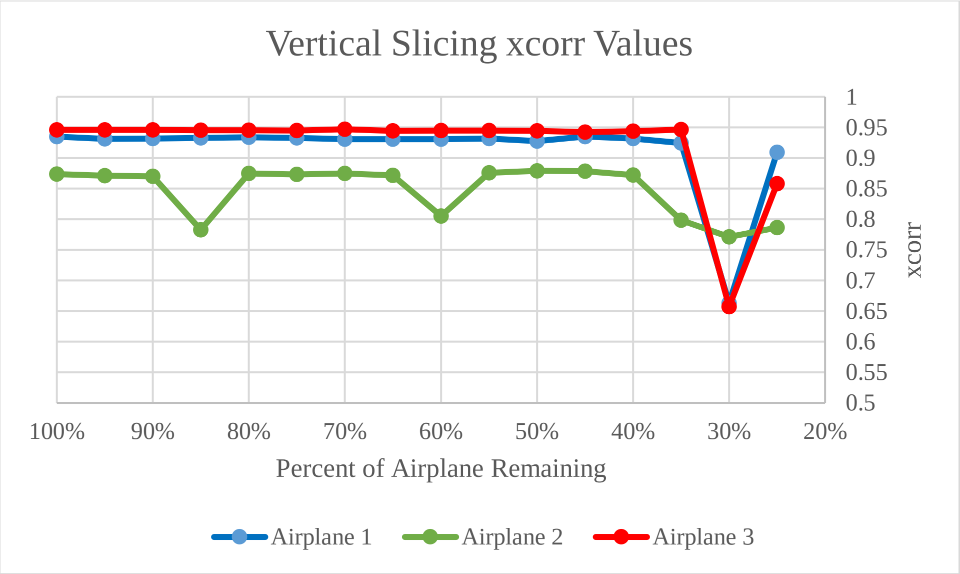
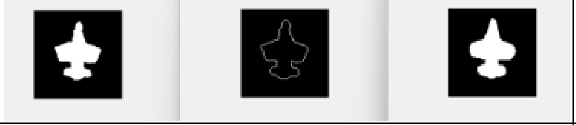
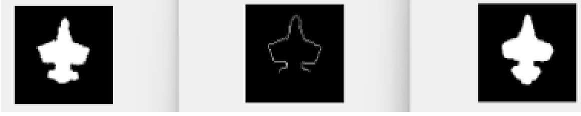
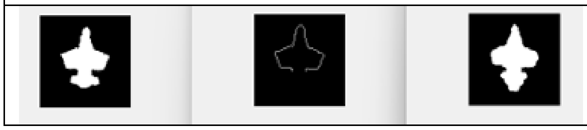
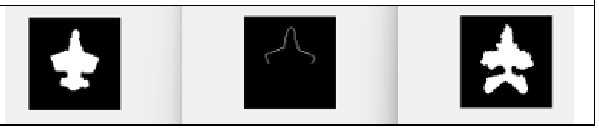


Figure 3-3. Vertical Slicing of Airplanes xcorr Values

3.3 Horizontal Slicing: Bottom-Up

Airplane 1 was then tested via the horizontal slicing bottom-up. Bottom-up means that rows were removed from the bottom, tail-end, of the airplane moving upwards. The baseline values were recorded, then 5% of the airplane was removed at a time, with values being recorded at each increment. The images from each increment were captured, with representative images displayed in Table 3-5. All the airplane 1 image captures can be seen in Appendix E.

Table 3-5. Airplane 1 Horizontal Slicing: Bottom-Up

100% Outline	85% Outline
	
75% Outline	**60% Outline**
	

The cross-correlation, similarity, error, psnr, and dot product values for all 5% increment slices were recorded for airplane 1 (Table 3-6). The same images and tables were obtained for airplanes 2 and 3, which can be referenced in Appendix E.

Table 3-6. Similarity Values for Horizontal Slicing Bottom-Up of Airplane 1

Percent (%)	xcorr	sim	Err	psnr	Dot
100	0.9349	0.8718	0.0224	16.5008	0.9370
95	0.9347	0.8756	0.0218	16.6136	0.9385
90	0.9371	0.8767	0.0211	16.7648	0.9404
85	0.9232	0.8668	0.0249	16.0366	0.9293
80	0.9060	0.8588	0.0294	15.3103	0.9168
75	0.9029	0.8566	0.0304	15.1769	0.9142
70	0.9052	0.8612	0.0289	15.3872	0.9184
65	0.9358	0.8737	0.0215	16.6712	0.9390
60	0.7731	0.7600	0.0771	11.1275	0.7960
55	0.8944	0.8547	0.0327	14.8527	0.9090
50	0.8315	0.7970	0.0590	12.2915	0.8411
45	0.9342	0.8741	0.0219	16.6022	0.9384
40	0.9289	0.8741	0.0230	16.3909	0.9330
35	0.8621	0.8417	0.0393	14.0546	0.8809
30	0.8344	0.8166	0.0512	12.9078	0.8503
25	0.5549	0.6059	0.1922	7.1623	0.6397

For the three airplanes, the cross-correlations were obtained and plotted for easy comparison (Figure 3-4).

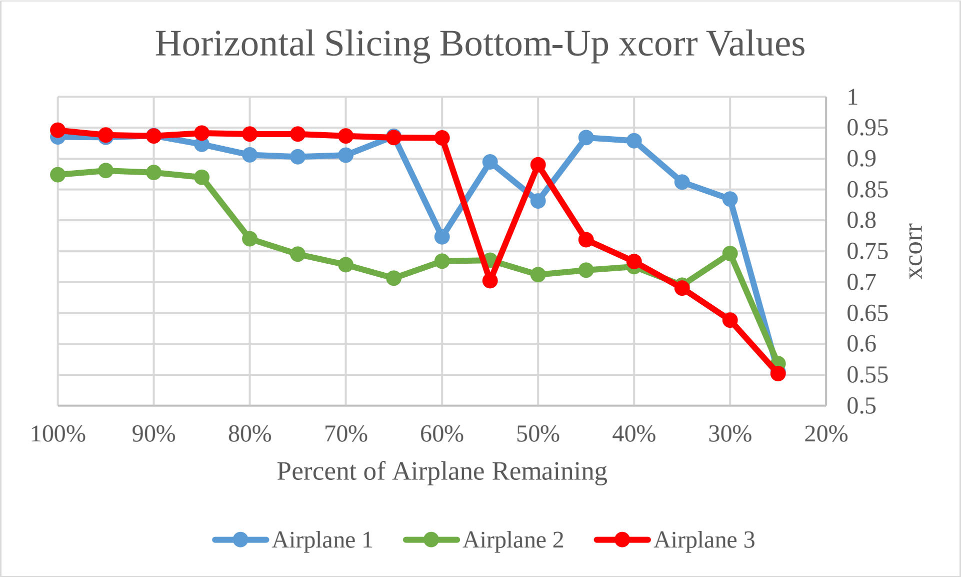
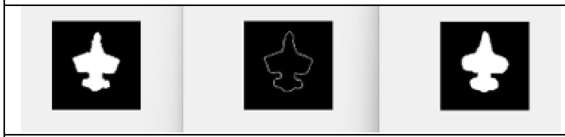
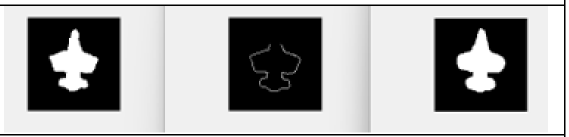
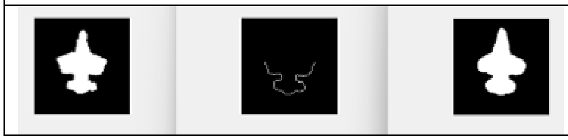
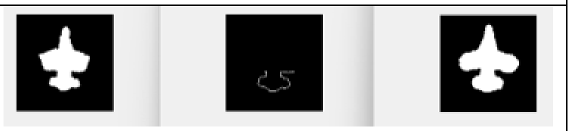


Figure 3-4. Horizontal Slicing Bottom-Up of Airplanes xcorr Values

3.4 Horizontal Slicing: Top Down

Airplane 1 was then tested via the horizontal slicing top-down. Top-down means that rows were removed from the top, head-end, of the airplane moving downwards. The baseline values were recorded, then 5% of the airplane was removed at a time, with values being recorded at each increment. The images from each increment were captured, with representative images displayed in Table 3-7. All the airplane 1 image captures can be seen in Appendix F.

Table 3-7. Airplane 1 Horizontal Slicing: Top-Down

100% Outline	75% Outline
	
50% Outline	**30% Outline**
	

The cross-correlation, similarity, error, psnr, and dot product values for all 5% increment slices were recorded for airplane 1 (Table 3-8). The same images and tables were obtained for airplanes 2 and 3, which can be referenced in Appendix F.

Table 3-8. Similarity Values for Horizontal Slicing Top-Down of Airplane 1

Percent (%)	xcorr	sim	Err	psnr	Dot
100	0.9349	0.8718	0.0224	16.5008	0.9370
95	0.9340	0.8723	0.0223	16.5232	0.9372
90	0.9348	0.8720	0.0224	16.5008	0.9369
85	0.9350	0.8721	0.0223	16.5120	0.9371
80	0.9351	0.8726	0.0222	16.5344	0.9375
75	0.9355	0.8724	0.0222	16.5344	0.9375
70	0.9335	0.8711	0.0226	16.4676	0.9365
65	0.9339	0.8718	0.0223	16.5120	0.9371
60	0.9333	0.8704	0.0228	16.4236	0.9360
55	0.9261	0.8689	0.0233	16.3262	0.9344
50	0.9111	0.8626	0.0259	15.8601	0.9265
45	0.8961	0.8530	0.0302	15.1934	0.9141
40	0.9013	0.8542	0.0289	15.3872	0.9184
35	0.8980	0.8541	0.0308	15.1117	0.9159
30	0.8712	0.8331	0.0413	13.8442	0.8913
25	0.8667	0.8234	0.0449	13.4794	0.8824

The cross-correlation values for all three airplanes through horizontal slicing top-down were obtained and plotted for easy analysis (Figure 3-5).

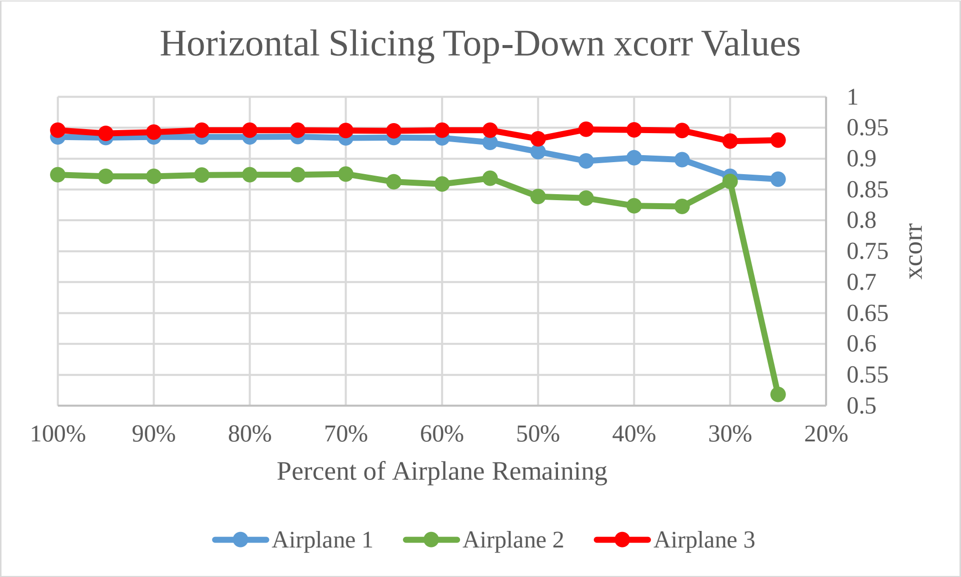


Figure 3-5. Horizontal Slicing Top-Down of Airplanes xcorr Values

Chapter 4

Discussion

This study aimed to provide a proof of concept in using a statistical shape model and principal component analysis to predict shapes with partial data. Through the results, a compelling case for the use of this model for more complex 3D structures has been made.

The primary statistic to determine the accuracy of the predictions was the cross-correlation value. This was chosen over the other similarity statistics because cross-correlation is widely used in research. The cross-correlation is widely understood, thus making it the best similarity statistic. Still, other similarity statistics were incorporated to ensure that the results obtained were accurate. In addition, the other similarity statistics enable slightly different comparison and conclusions to be drawn. The psnr value is the peak signal to noise ratio and is typically used as a quality measurement between an original image and some reconstruction of the image. The psnr value can identify the amount of noise that the model generates by finding the error between the prediction and the original image. The error is the mean squared error between the two images. The ssim similarity calculation represents the structural similarity index and is primarily used for measuring quality of images. The final similarity measurement was the dot product. The dot product calculates how similar the two image vectors are to being in the same direction.

In the removal of outline points, the prediction was consistent and accurate through 1/3 of the data points in the outline. Beyond 1/3 of the data points, the prediction abruptly dropped, becoming much less accurate. In Figure 3-2, the general trend can be seen for the three airplanes under testing. The cross-correlation value for two of the airplanes drops significantly when ¼ of the airplane outline is removed. As for airplane 2, the cross-correlation value is rather steady through 1/5 of the data being removed. An interesting phenomenon to note is that when more data was removed, to 1/5 of the outline, the cross-correlation values actually increased for the 3 airplanes. This pattern continues for the three airplanes where the cross-correlation value alternates from increasing to decreasing for each increment of the outline being removed. One explanation for this phenomenon is the distribution of outline points depending on the fraction of data removed. In Appendix C, each increment of outline removal can be seen for Airplane 1. At 1/6 of the outline, the points are not uniformly distributed. Rather, the points are clustered, providing the model less definition of the true airplane outline. When moving to 1/7 of the outline, the points become much more uniformly distributed. This provides the model with a more accurate outline definition, leading to less error in the prediction. This trend continues for each of the airplanes, which is likely the cause of the alternating cross-correlation phenomenon.

Next, the vertical slicing of the airplanes was analyzed to understand the ability of the model to accurately predict the shape. In Figure 3-3, the cross-correlation values for the three airplanes tested can be observed. Airplanes 1 and 3 were extremely consistent and accurate when just 35% of the airplane was present. Beyond that, the airplane prediction dropped significantly. As for Airplane 2, the prediction failed when 85% of the airplane still remained. The value jumped back to a consistent value when 80% of the airplane was present. This was consistent until another drop was recorded at 60% of the airplane remaining. This abnormal behavior from the second airplane is important to note and investigate further. One possible cause would be the asymmetric structure of the airplane. Airplanes 1 and 3 were perfectly symmetric, whereas Airplane 2 was not perfectly symmetric. When vertical slicing, the model relies more heavily on the symmetry from the data remaining from the unaffected portion of the airplane. Since Airplane 2 does not have perfect symmetry, the model likely struggled to accurately form the prediction.

The horizontal slicing was first performed in the bottom-up form, meaning data was removed from the tail-end of the airplane. As seen in Figure 3-4, Airplanes 1 and 3 were consistent and accurate through just 65% and 60% of the airplane remaining, respectively. Again, Airplane 2 failed much earlier at just 85% of the airplane remaining. Overall, the three airplanes had much poorer predictions with bottom-up slicing, indicating that the tail is a defining feature for the model. Without the tail, the model struggled to accurately predict the airplane.

The horizontal slicing was then performed top-down, meaning data was removed from the head of the airplane. As seen in Figure 3-5, the three airplanes all had accurate and relatively consistent predictions through 30% of the airplane remaining. When looking at the portion remaining, as seen in Appendix F, the tail was all that was left. This tail of the airplane provided enough information for the prediction to remain very consistent and accurate.

With the results better understood, it is critical to draw any parallels to work that has already been performed. The idea of using landmarks to create a shape model is used in a variety of applications. First, the identification of a face relies on key landmarks. In one study, 64 landmarks were identified on a human face to fully define all key features for future use [10]. This is like the removal of points from the outline of each airplane image. If too much data is removed, the key landmark features are eliminated. Key features for the airplane include the tips of the wings, curves and points on the tail, and the top point of the head. When these key features are removed from the outline, the prediction no longer remains accurate. This is the same with facial identification. If less than 64 landmarks are used, key information will be lost, and the facial recognition will fail to be accurate.

In some statistical shape model approaches, landmarks are used for alignment and more accurate predictions. In a study by Clogenson *et al*, a statistical shape model for the second cervical vertebra was created. A key aspect of the study was the identification of seven landmarks that can assist in the proper alignment of images to create a more accurate model. Based on the key landmarks, the rotation and centering process was performed. Since the orientation process was based upon the same seven landmarks for all training set images, the statistical shape model was inherently more accurate [11]. As was seen in this study, certain points within the outline are key defining features. As the wing or tail points were removed, the prediction failed. As was found in the study, it is important to ensure that the outline includes the key landmark points to ensure the prediction remains accurate, even with extremely limited data. From the study by Clogenson *et al*, it would be interesting to investigate how the use of landmarks for the alignment of images would impact the statistical shape model and overall results of the study. In addition, it would be valuable to investigate the prediction accuracy when removing points from the outline, while ensuring the key identified landmark points are still in the outline. These landmarks may include the head point of the airplane, tip of the wings, sharp corners or rounded edge points, and tail defining points.

Before identifying the limitations of this study, the key applications of this research should be highlighted. This study will primarily be implemented into the prediction of spinal anatomy from data obtained on an ultrasound image. An ultrasound image has limited data due to the 2D nature of the scan. In addition, ultrasound only provides a posterior view of the vertebrae, offering limited information on the anterior view. With this, it is important to understand how accurate the proposed model is with limited information. With an accurate representation of vertebral anatomy, ultrasound guided injections will be significantly improved, bettering the outcome for patients suffering from chronic, debilitating back pain. This study provides a strong proof of concept that parallels quite well with the application to 3D predictions. In this study, points were used to assist in the prediction of a 2D shape. In future research, a surface will be used to predict a 3D anatomy.

With the results understood, there are several limitations to the study that must be addressed. First, when a fraction of the outline is removed, the actual distribution of the points is the defining factor in the accuracy of the prediction. If the outline points, although may be limited, are uniformly distributed, the prediction will be more accurate. In addition, if the outline points highlight the abrupt curves or points on the airplane, the prediction will be more accurate. In future studies this will not be a limitation if it is ensured that the outline includes the key landmark features.

Second, the actual shape of the airplane drastically impacted the results of the prediction, as seen by the differences in Airplanes 1, 2, and 3. When the shape is symmetric, the prediction was extremely accurate through less data being present. However, when the shape was not symmetric, as was the case for Airplane 2, the model struggled to create an accurate prediction when some data was removed. This should not be a significant limitation when implemented into the prediction of spinal anatomy. Spinal vertebrae typically exhibit bilateral symmetry; thus, the prediction should be more accurate.

Another limitation of the model is its difficulty predicting sharp corners or small crevices of the airplane. The prediction for all three airplanes failed to capture the sharp corners present in the wings and body of the airplane. While this is a limitation of the study, it will not be a problem when the model is used on the spinal vertebrae. The vertebrae of the spine are relatively smooth and have less abrupt points or corners. By including the landmark points on the shape to be predicted, the model should be able to represent the shape more accurately.

The study is also limited by the number of airplanes in the training set to create the statistical shape model. With the airplanes used, there are significant variations in the tail shape, wing type, and head length. This large variation makes it more difficult to accurately predict an unknown airplane. The training set may not include an airplane that possesses similar characteristics to the unknown airplane, resulting in a less accurate prediction. This will not be an issue for the application of the prediction of spinal vertebrae. First, spinal vertebrae have less variation from person to person. The vertebrae all hold the same general shape and only have smaller variations. Second, the training set for the spinal anatomy will be significantly larger than that used for this study. With a larger training set, the variations that could be present in the vertebrae will be appropriately captured and represented in the statistical shape model.

Chapter 5

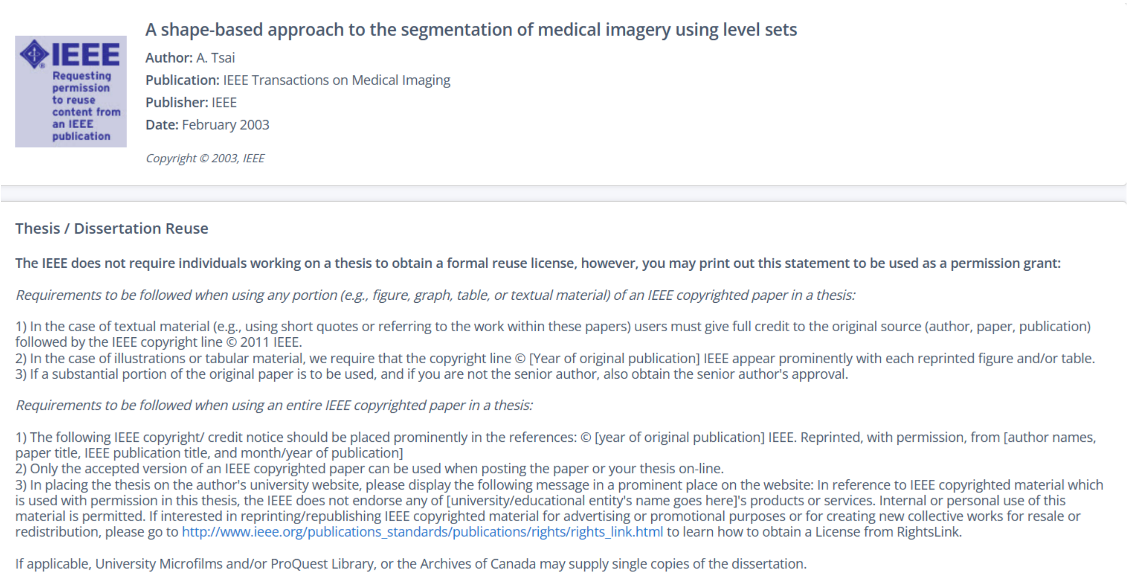
Conclusion

From the study performed, it can be concluded that a statistical shape model and principal components can be used to accurately predict shapes with partial data. Furthermore, the results support extending the research into 3D to predict spinal anatomy from ultrasound imaging data. The study found predictions to be accurate up to 70% of the data being removed. The partial data obtained with ultrasound imaging is missing the anterior view of the vertebra. This is about 50% of the data removed. Since the methodology in this research was generally accurate through 50% of data removed, the model should provide accurate predictions with the ultrasound's partial data.

It is highly recommended to extend this methodology to 3D applications keeping the following limitations in mind. The partial data provided must include key defining features for the structure to be predicted. The key features of the airplanes including tips of the airplane, sharp corners, and curves played a dominant role in the prediction. Based on this, for the vertebra, the key features are hypothesized to be the spinous process, transverse process, bottom point of the body, defining perimeter of the vertebral foramen, and facets. These seem to be the key defining features that will greatly influence the accuracy of the prediction. However, it is important to investigate all key features to determine the true landmarks within the statistical shape model. Even with the other limitations identified in this study, they should not be limiting in the extrapolation of the methodology to the human spinal vertebra. The spinal anatomy is generally smooth and possesses less variations, thus being an ideal application of the statistical shape model approach.

Appendix A

Rights to Use Images



IEEE
Requesting permission to reuse content from an IEEE publication

A shape-based approach to the segmentation of medical imagery using level sets

Author: A. Tsai

Publication: IEEE Transactions on Medical Imaging

Publisher: IEEE

Date: February 2003

Copyright © 2003, IEEE

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

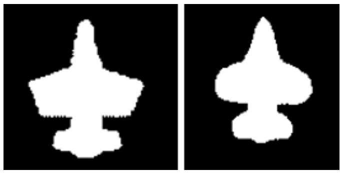
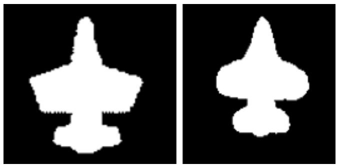
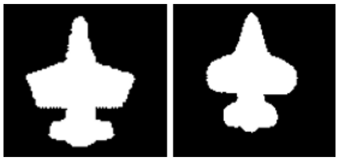
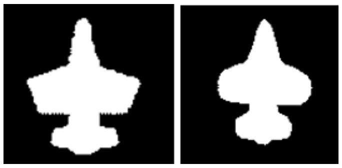
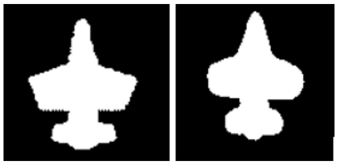
Figure A-1. IEEE Rights to Use Images

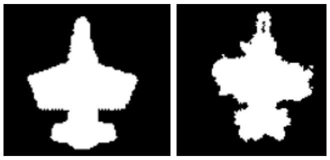
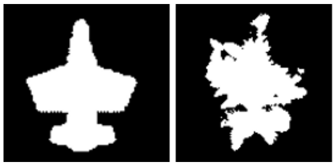
Appendix B

Determining the Number of Eigenvectors

Airplane 1

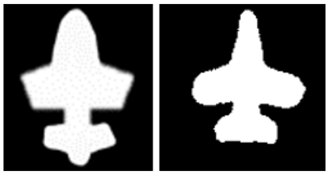
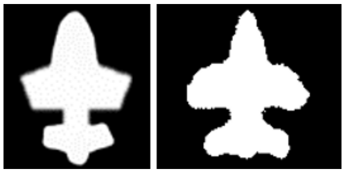
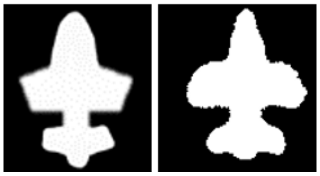
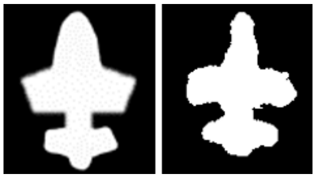
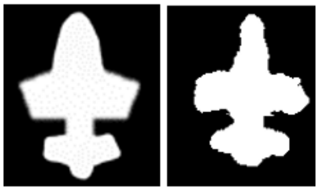
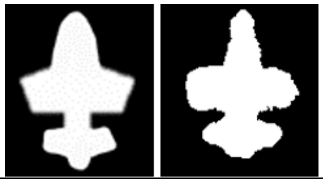
Table B-1. Determining the Sufficient Number of Eigenvectors for Airplane 1 (all data)

Number of Eigenvectors	Results (Initial Airplane, Prediction)
2	
3	
4	
5	
6	
7	

8	
9	
10	
11	
12	
13	

Airplane 2

Table B-2. Determining the Sufficient Number of Eigenvectors for Airplane 2 (all data)

Number of Eigenvectors	Results (Initial Airplane, Prediction)
2	
3	
4	
5	
6	
7	

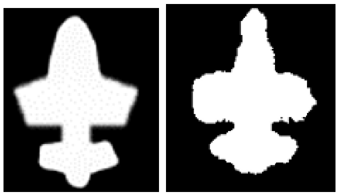
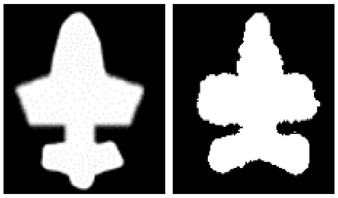
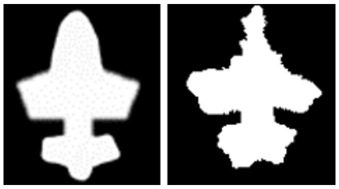
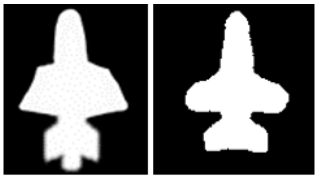
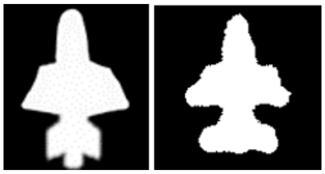
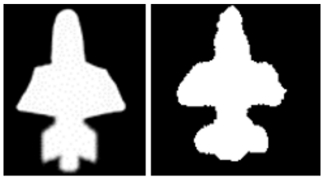
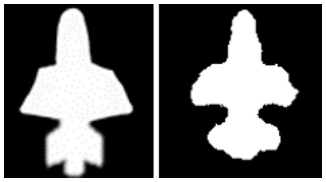
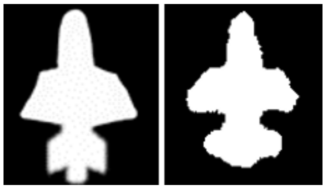
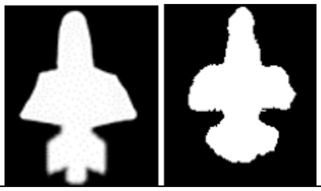
8	
9	
10	

Table B-3. Similarity Values for Each Eigenvector of Airplane 2

Eigs	xcorr	sim	Err	psnr	Dot
2	0.8359	0.7899	0.0592	12.1747	0.8708
3	0.8558	0.7907	0.0534	12.7220	0.8877
4	0.8718	0.8024	0.0473	13.2522	0.9005
5	0.8736	0.8026	0.0479	13.1946	0.9007
6	0.8738	0.8025	0.0478	13.2102	0.9011
7	0.8912	0.8160	0.0410	13.8684	0.9165
8	0.8866	0.8122	0.0426	13.7074	0.9139
9	0.8195	0.7698	0.0716	11.4528	0.8635
10	0.8413	0.7777	0.0639	11.9465	0.8736

Airplane 4

Table B-4. Determining the Sufficient Number of Eigenvectors for Airplane 4 (all data)

Number of Eigenvectors	Results (Initial Airplane, Prediction)
2	
3	
4	
5	
6	
7	

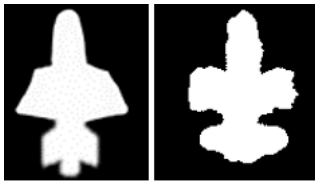
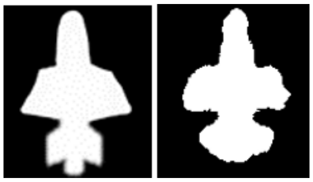
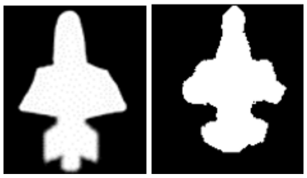
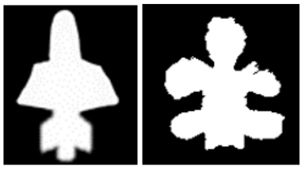
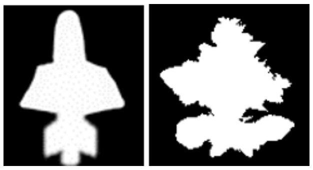
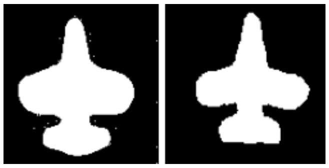
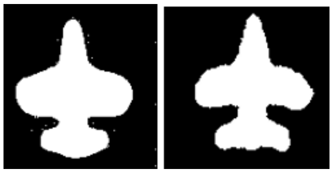
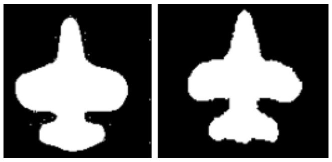
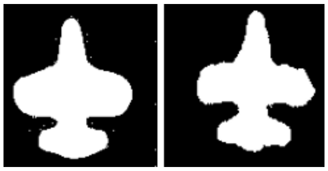
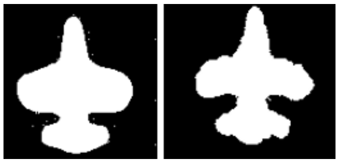
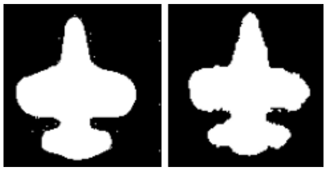
8	
9	
10	
11	
12	

Table B-5. Similarity Values for Each Eigenvector of Airplane 4

Eigs	xcorr	sim	Err	psnr	Dot
2	0.8668	0.8184	0.0471	13.2680	0.8950
3	0.8836	0.8235	0.0423	13.7368	0.9113
4	0.9061	0.8355	0.0350	14.5582	0.9258
5	0.8973	0.7269	0.0371	14.3092	0.9216
6	0.9034	0.8378	0.0351	14.5440	0.9264
7	0.9075	0.8373	0.0336	14.7325	0.9295
8	0.8649	0.8085	0.0509	12.9322	0.8978
9	0.9016	0.8317	0.0362	14.4181	0.9252
10	0.9065	0.8348	0.0340	14.6883	0.9284
11	0.7146	0.6926	0.1165	9.3365	0.7855
12	0.7465	0.6900	0.1177	9.2918	0.7911

Airplane 5

Table B-6. Determining the Sufficient Number of Eigenvectors for Airplane 5 (all data)

Number of Eigenvectors	Results (Initial Airplane, Prediction)
2	
3	
4	
5	
6	
7	
8	

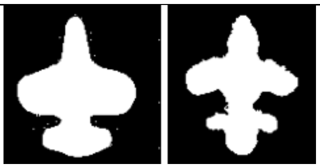
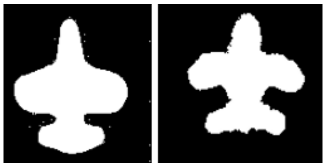
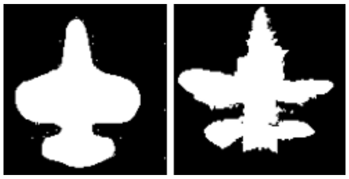
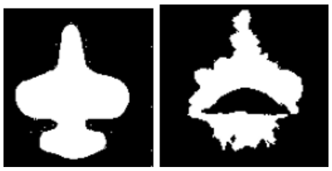
	
9	
10	
11	

Table B-7. Similarity Values for Each Eigenvector of Airplane 5

Eigs	xcorr	sim	Err	psnr	Dot
2	0.9091	0.8332	0.0330	14.8148	0.9212
3	0.9079	0.8320	0.0345	14.6227	0.9215
4	0.9130	0.8400	0.0313	15.0475	0.9289
5	0.9249	0.8402	0.0301	15.2182	0.9309
6	0.9113	0.8377	0.0329	14.8224	0.9250
7	0.9197	0.8504	0.0281	15.5095	0.9375
8	0.8324	0.7822	0.0596	12.2495	0.8702
9	0.8609	0.7985	0.0495	13.0514	0.8857
10	0.7528	0.7258	0.0895	10.4802	0.8095
11	0.75754	0.7524	0.0843	10.7413	0.8016
12	0.6804	0.7037	0.1139	9.4361	0.7468

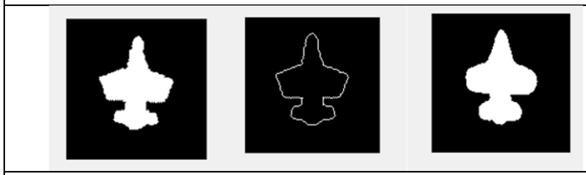
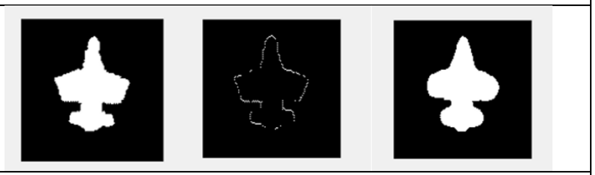
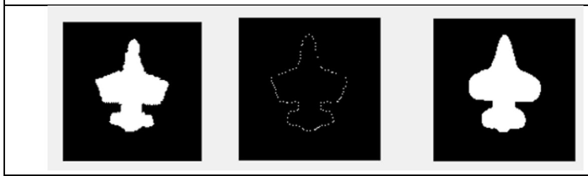
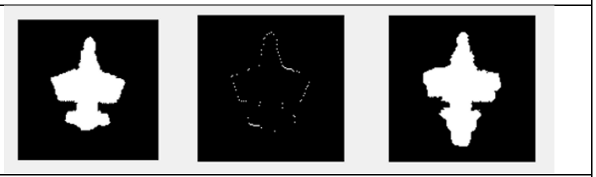
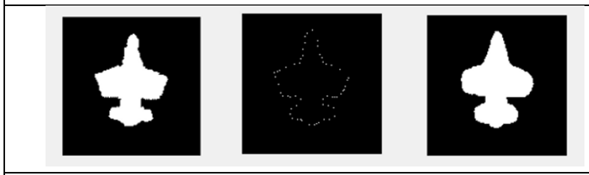
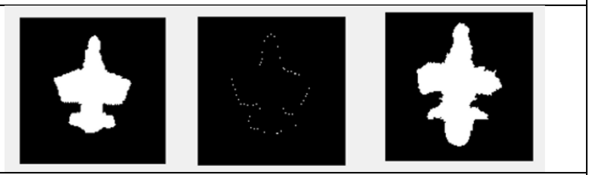
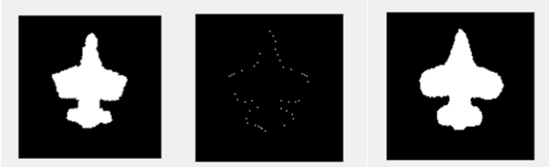
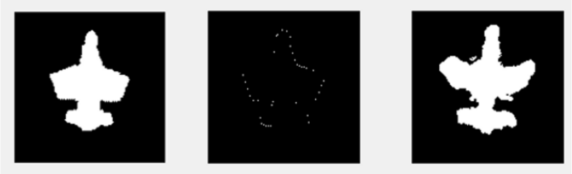
Appendix C

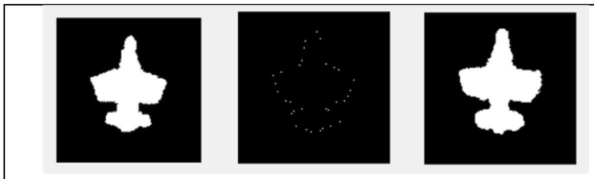
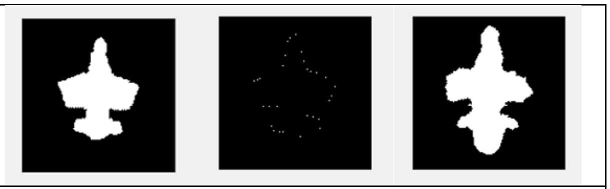
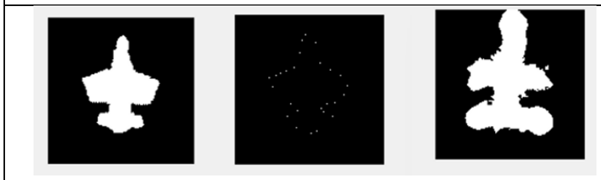
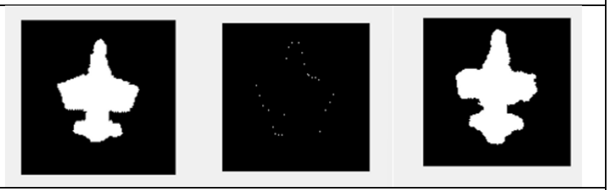
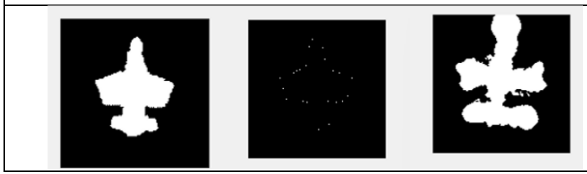
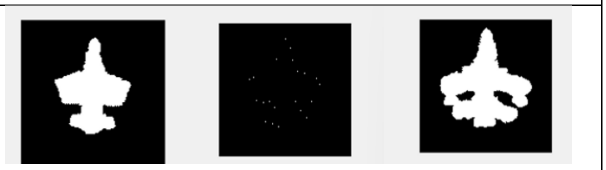
Removal of Outline Points

Airplane 1

In order from left to right, the original airplane, the airplane outline, and the predicted airplane.

Table C-1. Airplane 1: Removal of Outline Points

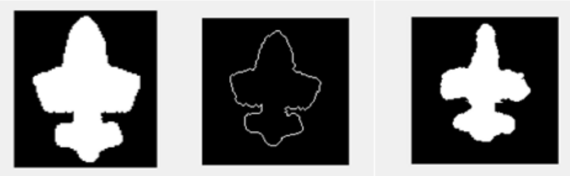
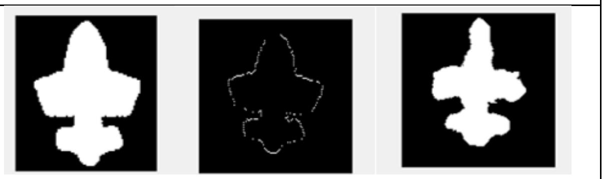
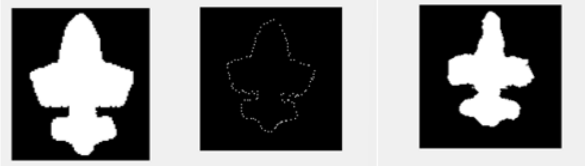
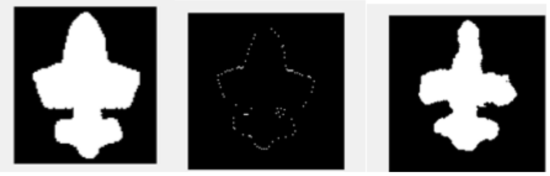
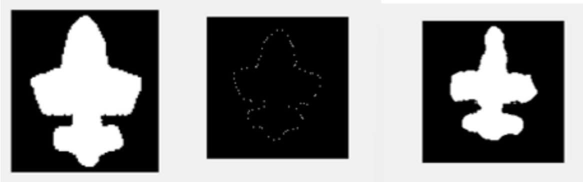
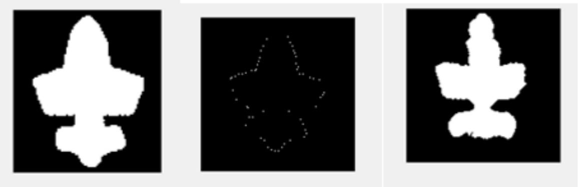
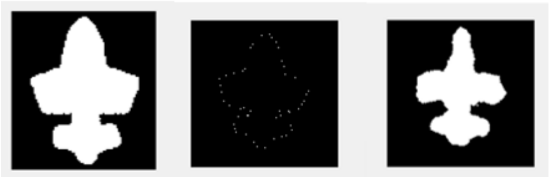
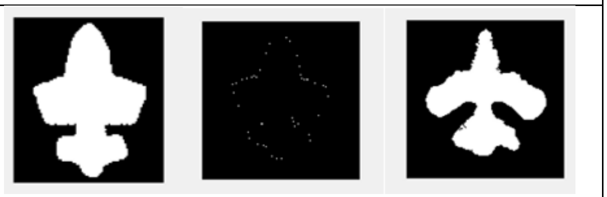
Full Outline	**1/2 Outline**
	
1/3 Outline	**1/4 Outline**
	
1/5 Outline	**1/6 Outline**
	
1/7 Outline	**1/8 Outline**
	
1/9 Outline	**1/10 Outline**

	
1/11 Outline	**1/12 Outline**
	
1/13 Outline	**1/14 Outline**
	

Airplane 2

In order from left to right, the original airplane, the airplane outline, and the predicted airplane.

Table C-2. Airplane 2: Removal of Outline Points

Full Outline	**1/2 Outline**
	
1/3 Outline	**1/4 Outline**
	
1/5 Outline	**1/6 Outline**
	
1/7 Outline	**1/8 Outline**
	
1/9 Outline	**1/10 Outline**

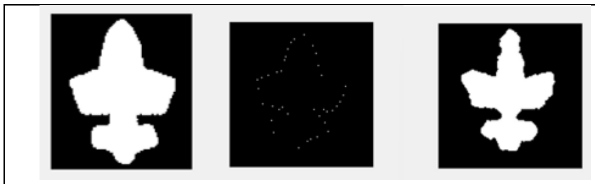
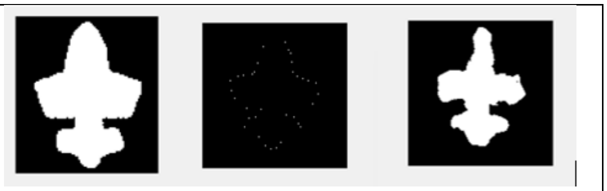
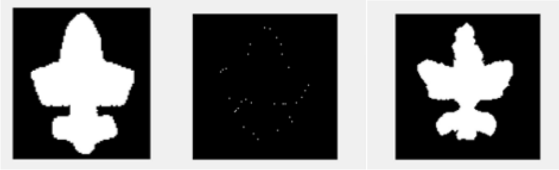
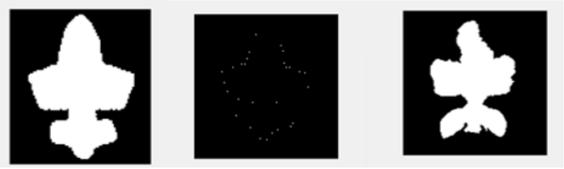
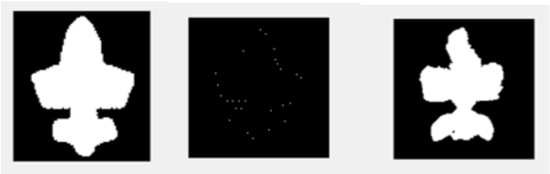
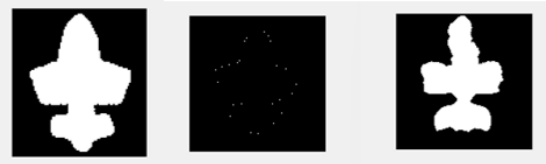
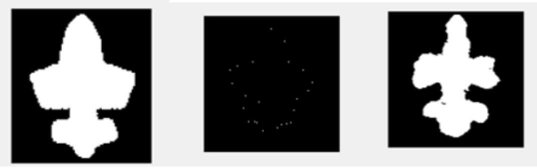
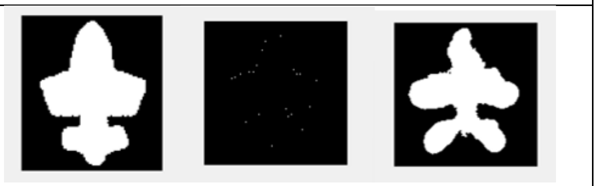
	
1/11 Outline	**1/12 Outline**
	
1/13 Outline	**1/14 Outline**
	
1/15 Outline	**1/16 Outline**
	

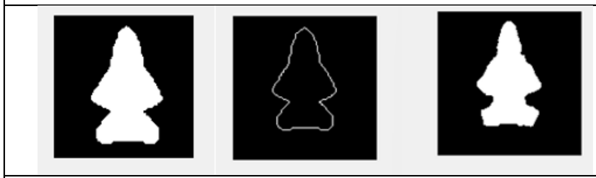
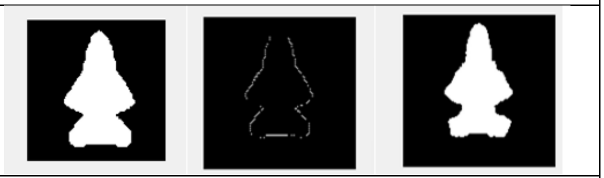
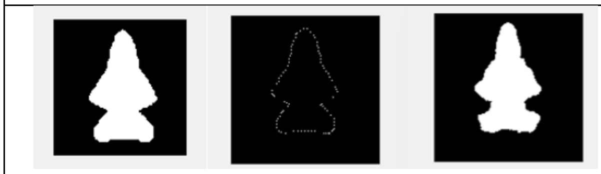
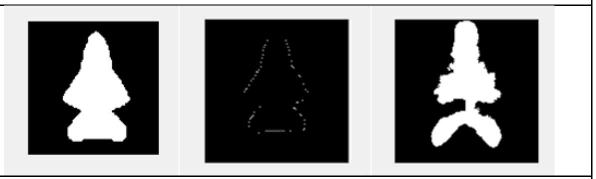
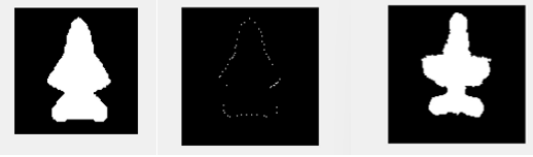
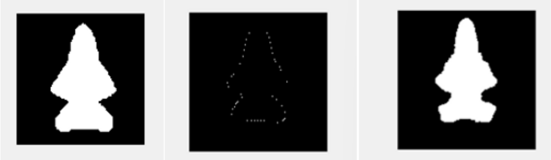
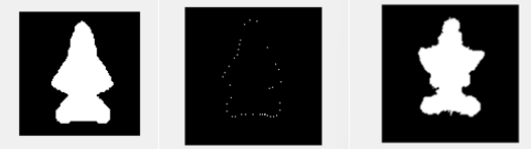
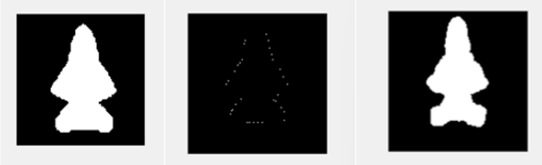
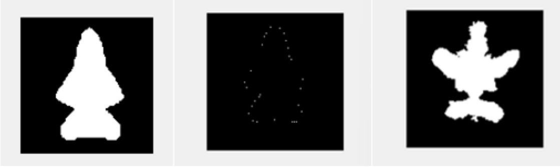
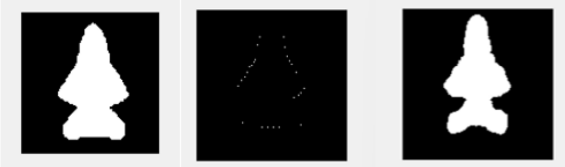
Table C-3. Similarity Values for Removal of Outline Points for Airplane 2

Portion of Outline	xcorr	sim	Err	psnr	Dot
Full	0.8738	0.8025	0.0478	13.2102	0.9011
1/2	0.8666	0.8007	0.0504	12.9716	0.8957
1/3	0.8757	0.8027	0.0471	13.2733	0.9026
1/4	0.8650	0.7975	0.0509	12.9322	0.8952
1/5	0.8712	0.8002	0.0484	13.1481	0.9004
1/6	0.8387	0.7782	0.0617	12.0974	0.8772
1/7	0.8793	0.8079	0.0456	13.4078	0.9051
1/8	0.7467	0.7167	0.0980	10.0891	0.8087
1/9	0.8423	0.7756	0.0594	12.2621	0.8808
1/10	0.8733	0.8000	0.0508	12.9420	0.8933
1/11	0.8286	0.7675	0.0667	11.7557	0.8683
1/12	0.7985	0.7455	0.0801	10.9626	0.8440
1/13	0.8046	0.7477	0.0785	11.0538	0.8484
1/14	0.8151	0.7557	0.0745	11.2787	0.8559
1/15	0.7760	0.7263	0.0921	10.3567	0.8294
1/16	0.6862	0.6661	0.1342	8.7212	0.7701

Airplane 3

In order from left to right, the original airplane, the airplane outline, and the predicted airplane.

Table C-4. Airplane 3: Removal of Outline Points

Full Outline	1/2 Outline
	
1/3 Outline	**1/4 Outline**
	
1/5 Outline	**1/6 Outline**
	
1/7 Outline	**1/8 Outline**
	
1/9 Outline	**1/10 Outline**
	
1/11 Outline	**1/12 Outline**

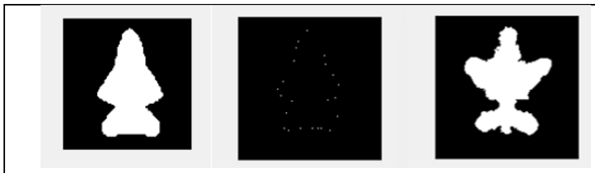
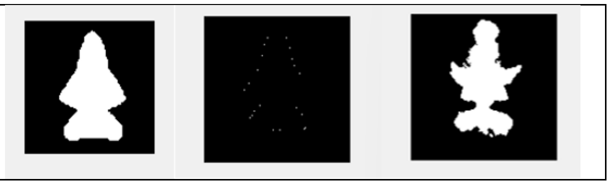
	

Table C-5. Similarity Values for Removal of Outline Points for Airplane 3

Portion of Outline	xcorr	sim	Err	psnr	Dot
Full	0.9457	0.8912	0.0172	17.6403	0.9548
1/2	0.9377	0.8841	0.0202	16.9461	0.9478
1/3	0.9396	0.8864	0.0195	17.1095	0.9496
¼	0.7524	0.7590	0.0817	10.8764	0.8018
1/5	0.8543	0.8257	0.0461	13.3643	0.8814
1/6	0.9337	0.8796	0.0218	16.6251	0.9440
1/7	0.8361	0.8195	0.0517	12.8642	0.8667
1/8	0.9107	0.8662	0.0296	15.2850	0.925
1/9	0.7338	0.7584	0.0869	10.6074	0.7861
1/10	0.9008	0.8604	0.0324	14.8910	0.9178
1/11	0.7569	0.7705	0.0814	10.8917	0.8058
1/12	0.8086	0.7901	0.0619	12.0813	0.8404

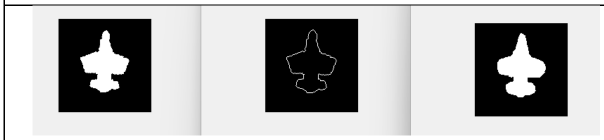
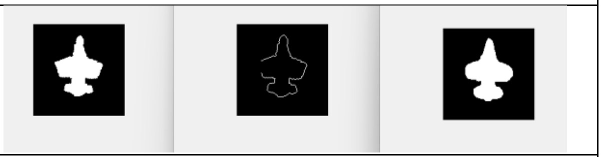
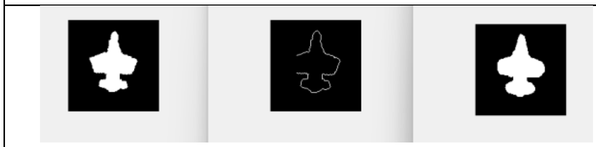
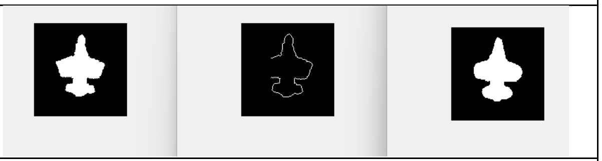
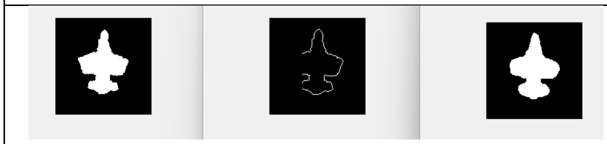
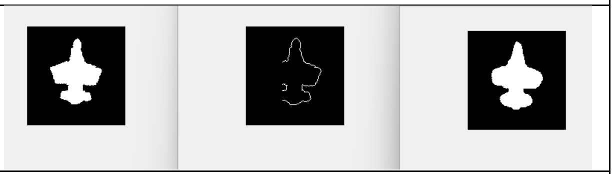
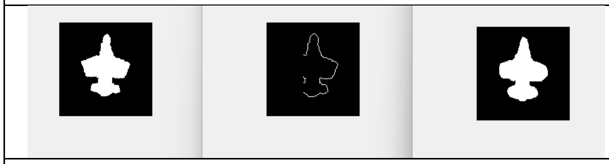
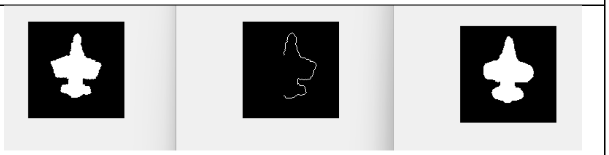
Appendix D

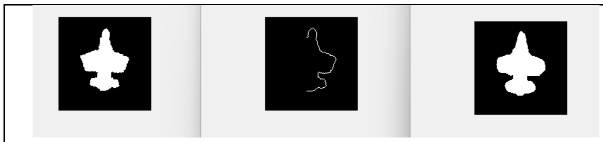
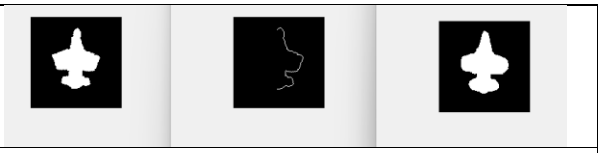
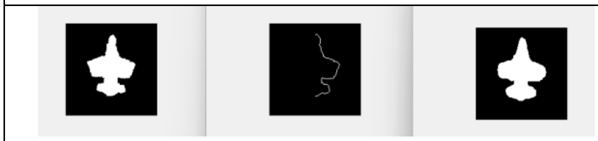
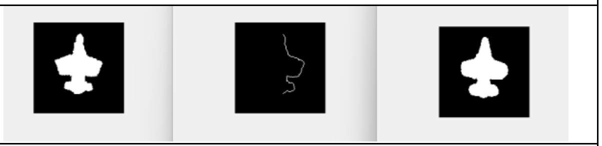
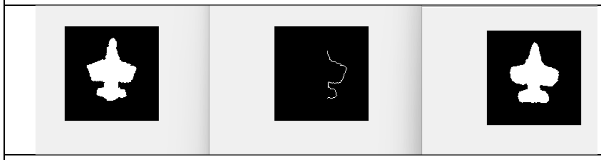
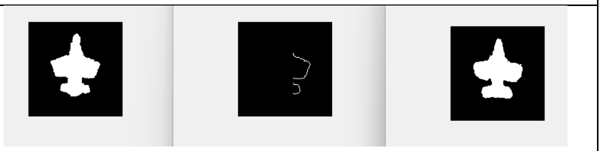
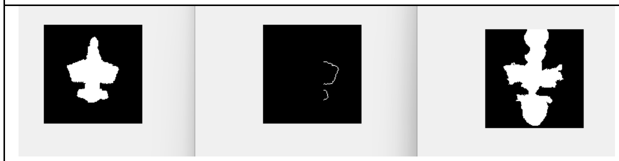
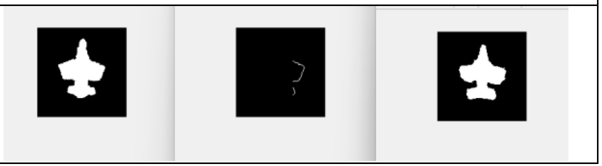
Vertical Slicing

Airplane 1

In order from left to right, the original airplane, the airplane outline, and the predicted airplane.

Table D-1. Airplane 1: Vertical Slicing (all data)

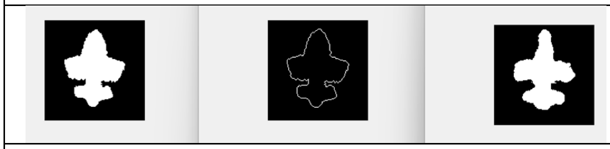
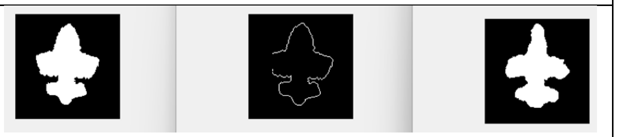
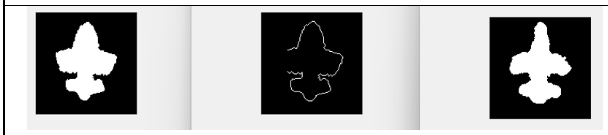
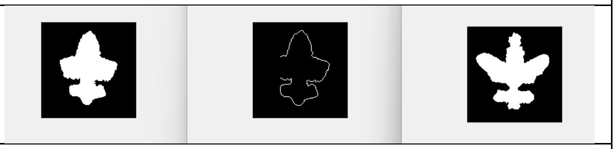
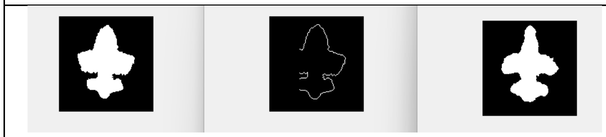
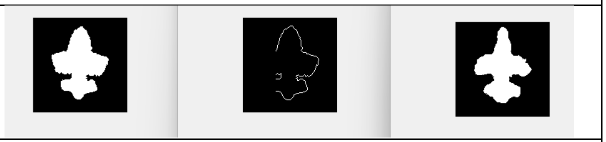
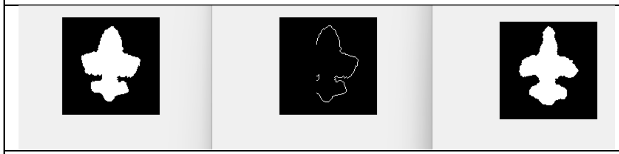
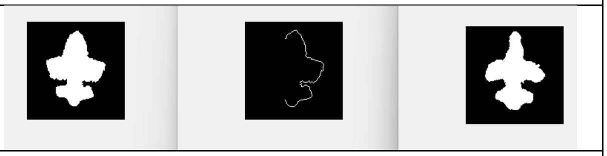
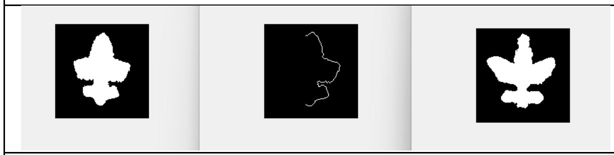
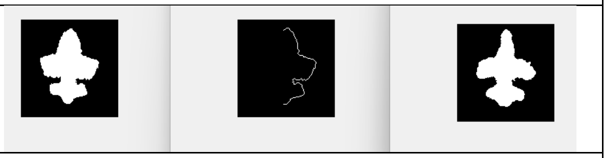
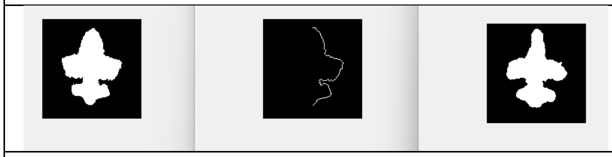
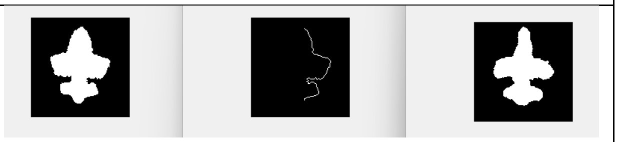
Full Outline	**95% Outline**
	
90% Outline	**85% Outline**
	
80% Outline	**75% Outline**
	
70% Outline	**65% Outline**
	
60% Outline	**55% Outline**

	
50% Outline	**45% Outline**
	
40% Outline	**35% Outline**
	
30% Outline	**25% Outline**
	

Airplane 2

In order from left to right, the original airplane, the airplane outline, and the predicted airplane.

Table D-2. Airplane 2: Vertical Slicing (all data)

Full Outline	**95% Outline**
	
90% Outline	**85% Outline**
	
80% Outline	**75% Outline**
	
70% Outline	**65% Outline**
	
60% Outline	**55% Outline**
	
50% Outline	**45% Outline**
	

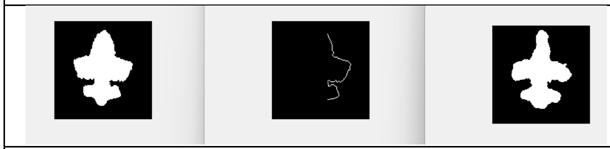
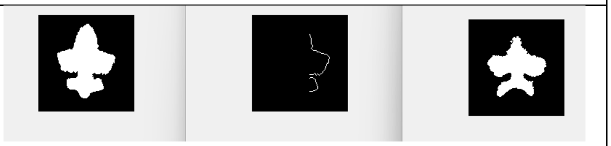
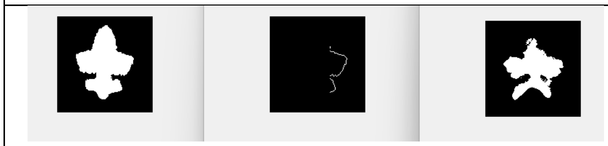
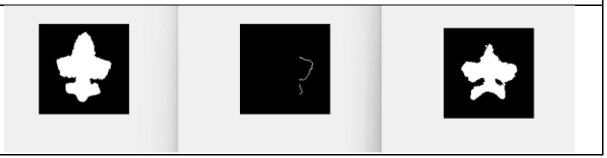
40% Outline	35% Outline
	
30% Outline	25% Outline
	

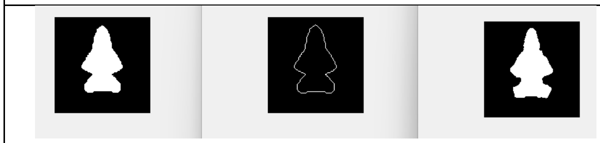
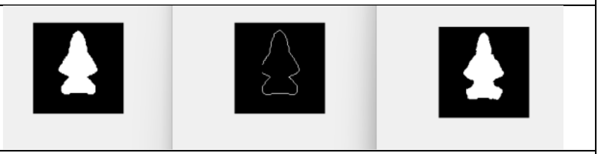
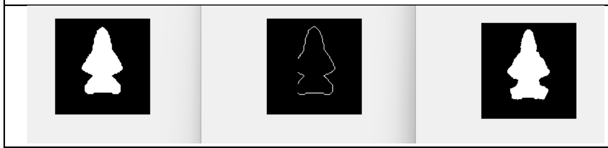
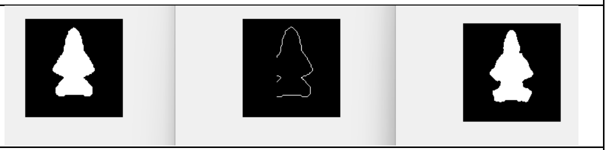
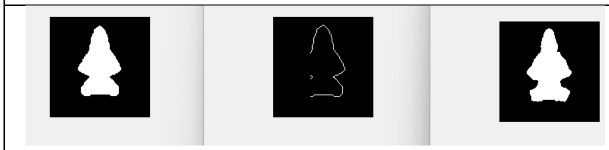
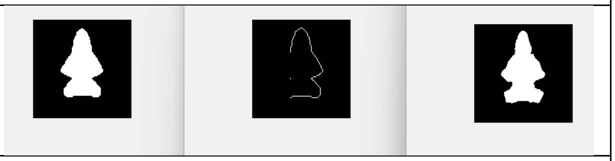
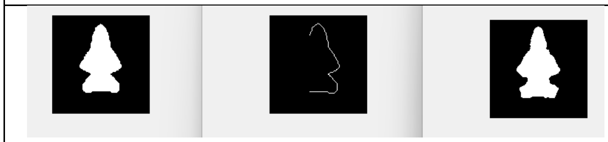
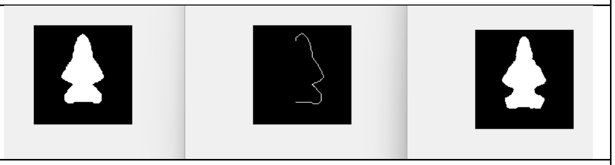
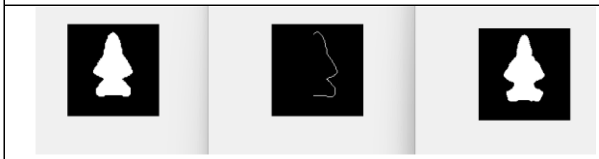
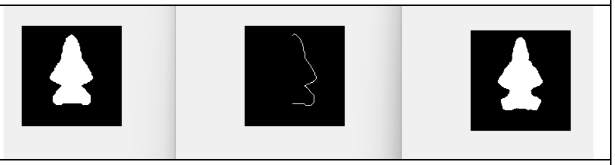
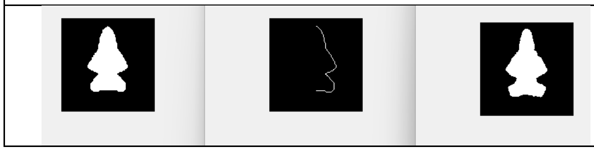
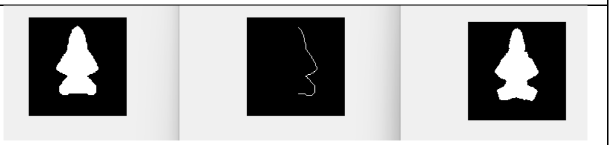
Table D-3. Similarity Values for Vertical Slicing of Airplane 2

Percent (%)	xcorr	sim	Err	psnr	Dot
100	0.8738	0.8025	0.0478	13.2102	0.9011
95	0.8712	0.8038	0.0476	13.2207	0.9009
90	0.8700	0.8056	0.0476	13.2259	0.9008
85	0.7827	0.7378	0.0839	10.7620	0.8382
80	0.8747	0.8086	0.0460	13.3697	0.9033
75	0.8732	0.8055	0.0468	13.2999	0.9016
70	0.8750	0.8074	0.0459	13.3806	0.9036
65	0.8714	0.8051	0.0481	13.1790	0.8987
60	0.8054	0.7491	0.0754	11.2255	0.8530
55	0.8757	0.8103	0.0471	13.2733	0.9005
50	0.8791	0.8165	0.0452	13.4517	0.9048
45	0.8785	0.8178	0.0447	13.4961	0.9058
40	0.8721	0.8082	0.0480	13.1842	0.8993
35	0.7984	0.7767	0.0750	11.2487	0.8336
30	0.7711	0.7554	0.0853	10.6884	0.8097

Airplane 3

In order from left to right, the original airplane, the airplane outline, and the predicted airplane.

Table D-4. Airplane 3: Vertical Slicing (all data)

Full Outline	**95% Outline**
	
90% Outline	**85% Outline**
	
80% Outline	**75% Outline**
	
70% Outline	**65% Outline**
	
60% Outline	**55% Outline**
	
50% Outline	**45% Outline**
	

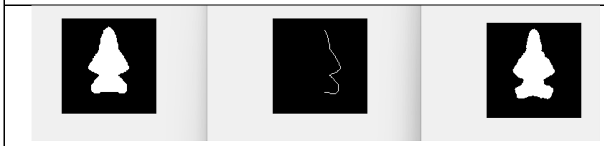
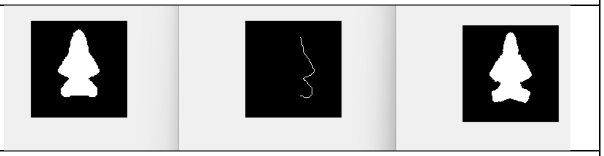
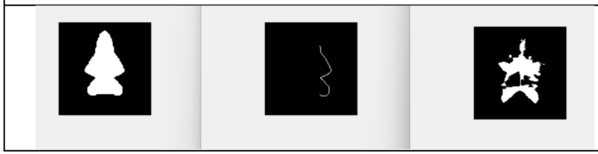
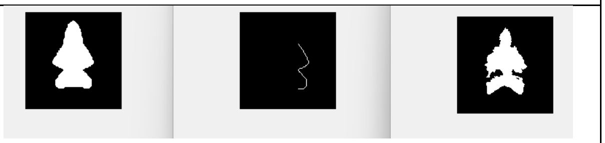
40% Outline	35% Outline
	
30% Outline	**25% Outline**
	

Table D-5. Similarity Values for Vertical Slicing of Airplane 3

Percent (%)	xcorr	sim	Err	psnr	Dot
100	0.9457	0.8912	0.0172	17.6403	0.9548
95	0.9459	0.8925	0.0170	17.6839	0.9552
90	0.9461	0.8923	0.0170	17.6986	0.9553
85	0.9453	0.8926	0.0171	17.6693	0.9550
80	0.9455	0.8909	0.0172	17.6548	0.9548
75	0.9448	0.8901	0.0176	17.5401	0.9537
70	0.9470	0.8917	0.0169	17.7280	0.9555
65	0.9445	0.8907	0.0173	17.6114	0.9544
60	0.9448	0.8915	0.0170	17.6986	0.9552
55	0.9450	0.8925	0.0169	17.7133	0.9554
50	0.9443	0.8929	0.0172	17.6548	0.9548
45	0.9421	0.8898	0.0178	17.4839	0.9530
40	0.9436	0.8925	0.0173	17.6114	0.9542
35	0.9467	0.8960	0.0162	17.8936	0.9566
30	0.6573	0.7520	0.0961	10.1739	0.7069
25	0.8582	0.8317	0.0424	13.7250	0.8807

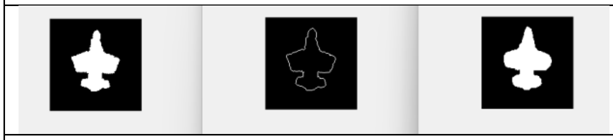
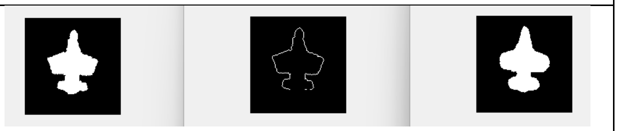
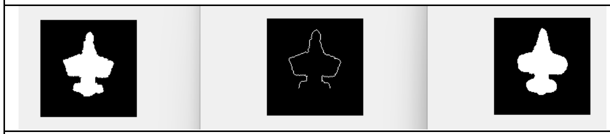
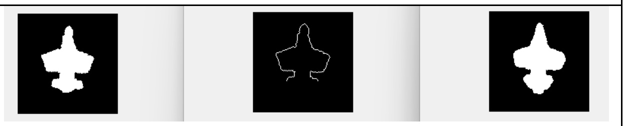
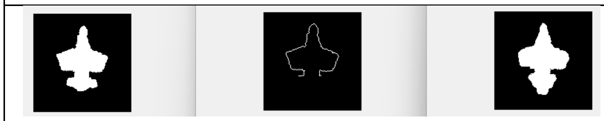
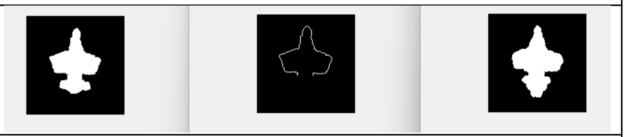
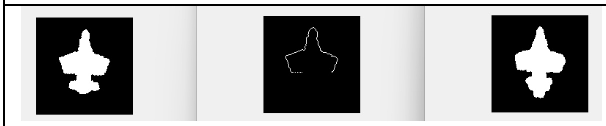
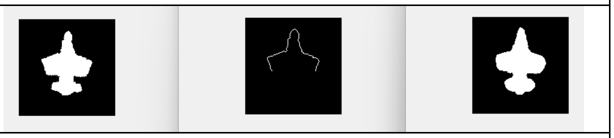
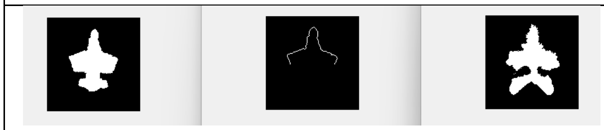
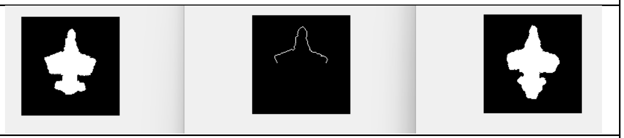
Appendix E

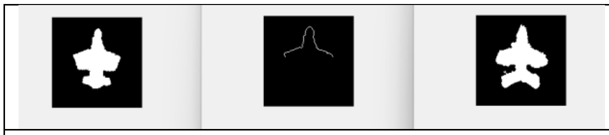
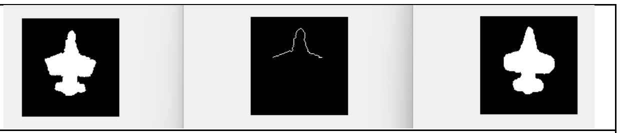
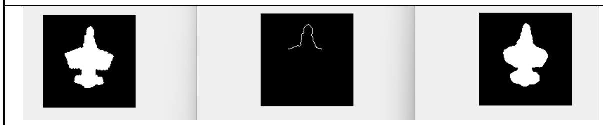
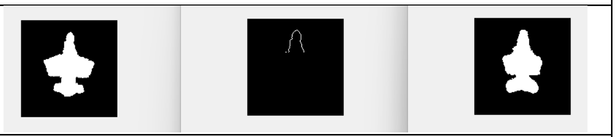
Horizontal Slicing Bottom-Up

Airplane 1

In order from left to right, the original airplane, the airplane outline, and the predicted airplane.

Table E-1. Airplane 1: Horizontal Slicing Bottom-Up (all data)

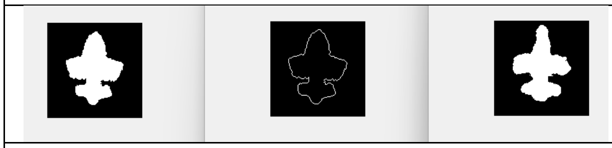
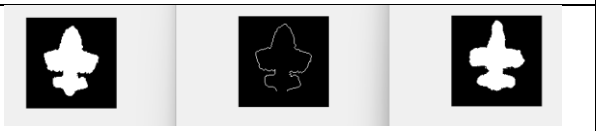
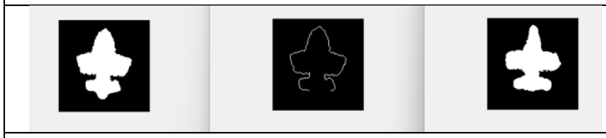
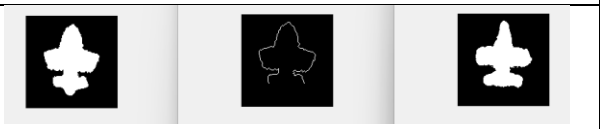
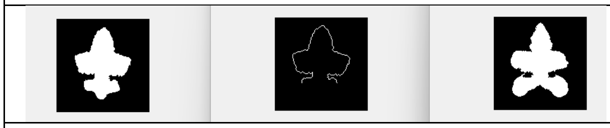
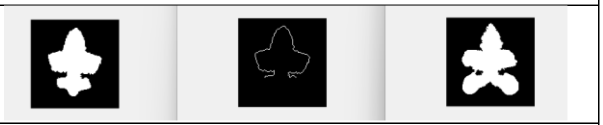
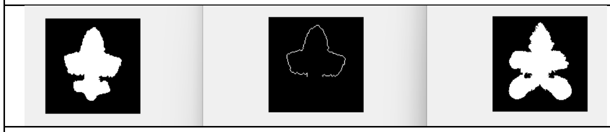
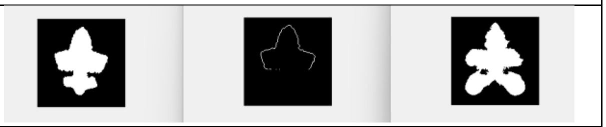
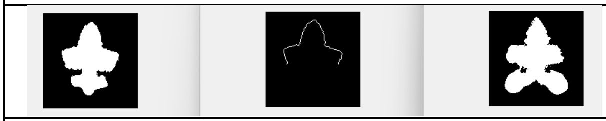
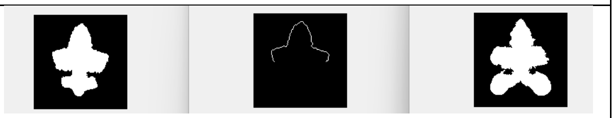
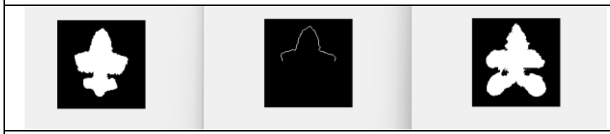
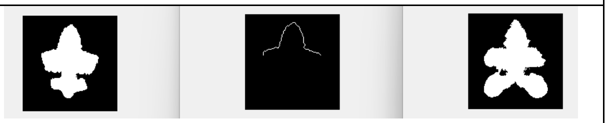
Full Outline	**95% Outline**
	
90% Outline	**85% Outline**
	
80% Outline	**75% Outline**
	
70% Outline	**65% Outline**
	
60% Outline	**55% Outline**
	
50% Outline	**45% Outline**

	
40% Outline	**35% Outline**
	
30% Outline	**25% Outline**

Airplane 2

In order from left to right, the original airplane, the airplane outline, and the predicted airplane.

Table E-2. Airplane 2: Horizontal Slicing Bottom-Up (all data)

Full Outline	**95% Outline**
	
90% Outline	**85% Outline**
	
80% Outline	**75% Outline**
	
70% Outline	**65% Outline**
	
60% Outline	**55% Outline**
	
50% Outline	**45% Outline**
	
40% Outline	**35% Outline**

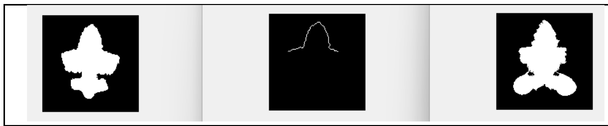
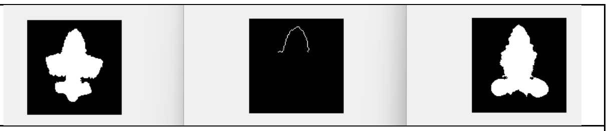
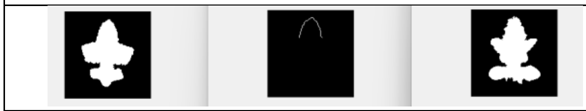
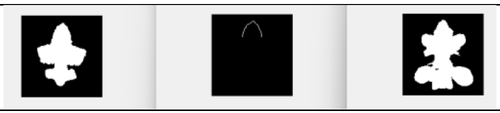
	
30% Outline	**25% Outline**
	

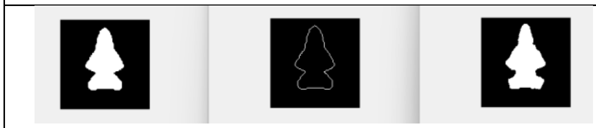
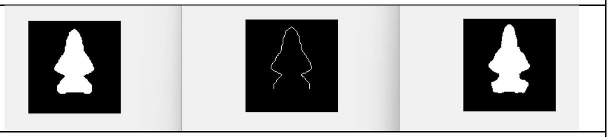
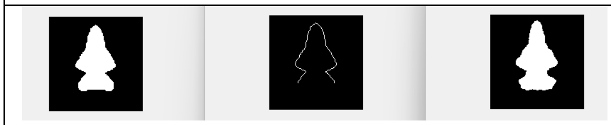
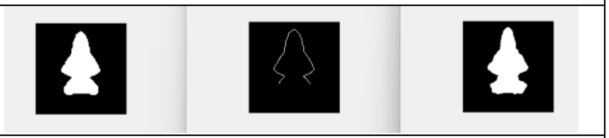
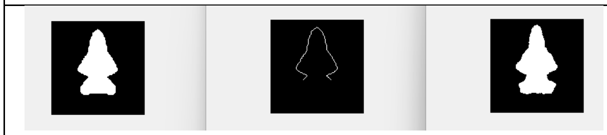
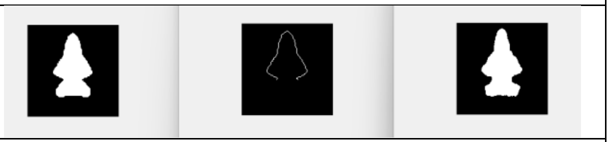
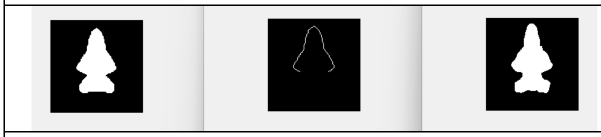
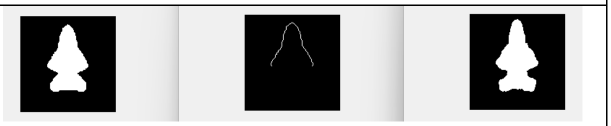
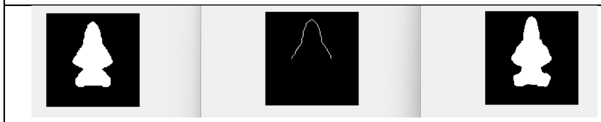
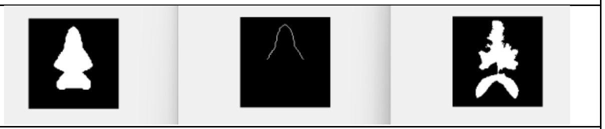
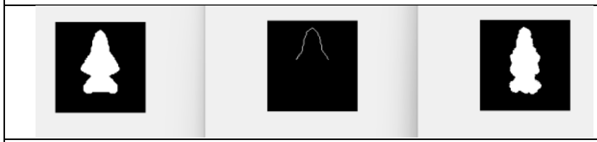
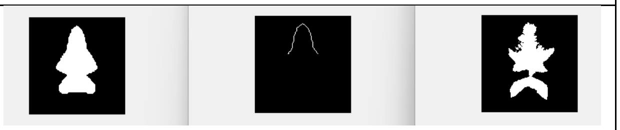
Table E-3. Similarity Values for Horizontal Slicing Bottom-Up of Airplane 2

Percent (%)	xcorr	sim	Err	psnr	Dot
100	0.8738	0.8025	0.0478	13.2102	0.9011
95	0.8804	0.8106	0.0453	13.4352	0.9053
90	0.8773	0.8074	0.0479	13.1946	0.8983
85	0.8696	0.8001	0.0506	12.9568	0.8929
80	0.7703	0.7502	0.0911	10.4057	0.8271
75	0.7453	0.7301	0.1046	9.8060	0.8076
70	0.7280	0.7179	0.1127	9.4801	0.7955
65	0.7061	0.6961	0.1248	9.0369	0.7809
60	0.7337	0.7176	0.1124	9.4934	0.7959
55	0.7353	0.7187	0.1098	9.5921	0.8013
50	0.7120	0.7000	0.1233	9.0891	0.7835
45	0.7193	0.7085	0.1194	9.2308	0.7892
40	0.7252	0.7252	0.1115	9.5268	0.7852
35	0.6950	0.7292	0.1154	9.3774	0.7643
30	0.7461	0.7392	0.0987	10.0562	0.8018
25	0.5677	0.6130	0.1928	7.1481	0.6779

Airplane 3

In order from left to right, the original airplane, the airplane outline, and the predicted airplane.

Table E-4. Airplane 3: Horizontal Slicing Bottom-Up (all data)

Full Outline	**95% Outline**
	
90% Outline	**85% Outline**
	
80% Outline	**75% Outline**
	
70% Outline	**65% Outline**
	
60% Outline	**55% Outline**
	
50% Outline	**45% Outline**
	
40% Outline	**35% Outline**

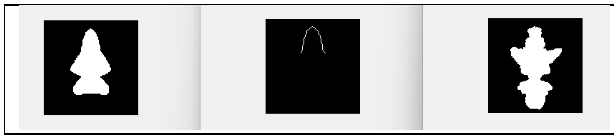
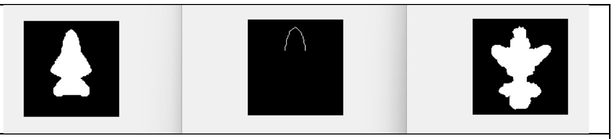
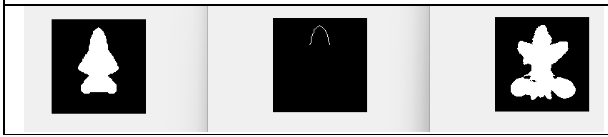
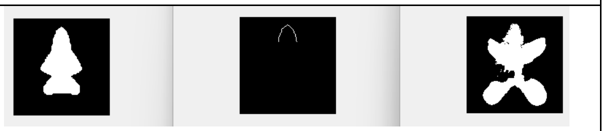
	
30% Outline	**25% Outline**
	

Table E-5. Similarity Values for Horizontal Slicing Bottom-Up of Airplane 3

Percent (%)	xcorr	sim	Err	psnr	Dot
100	0.9457	0.8912	0.0172	17.6403	0.9548
95	0.9379	0.8858	0.0197	17.0459	0.9488
90	0.9364	0.8872	0.0197	17.0459	0.9484
85	0.9410	0.8879	0.0184	17.3464	0.9519
80	0.9398	0.8866	0.0192	17.1740	0.9500
75	0.9396	0.8876	0.0196	17.0839	0.9489
70	0.9367	0.8855	0.0212	16.7295	0.9448
65	0.9338	0.8839	0.0214	16.6944	0.9446
60	0.9334	0.8829	0.0214	16.6944	0.9446
55	0.7023	0.7368	0.0917	10.3757	0.7557
50	0.8899	0.8550	0.0341	14.6663	0.9069
45	0.7686	0.7709	0.0698	11.5621	0.8089
40	0.7331	0.7647	0.0914	10.3920	0.7860
35	0.6902	0.7298	0.1154	9.3774	0.7554
30	0.6383	0.6832	0.1454	8.3734	0.7148
25	0.5519	0.6321	0.1740	7.5942	0.6411

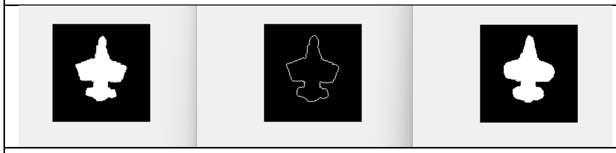
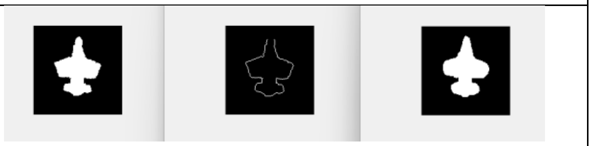
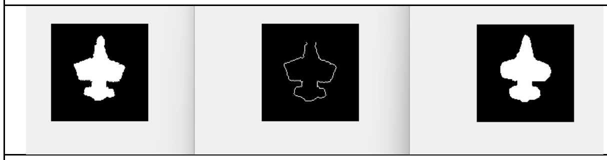
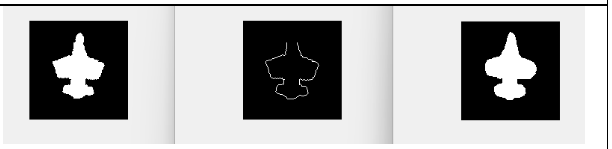
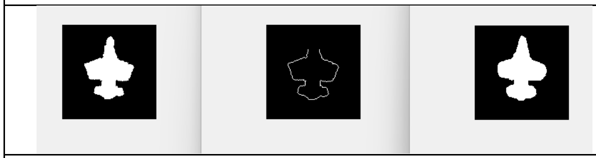
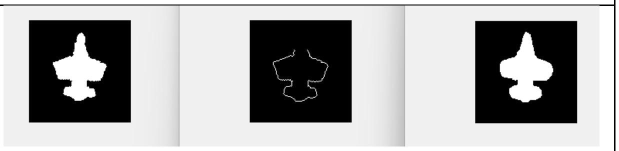
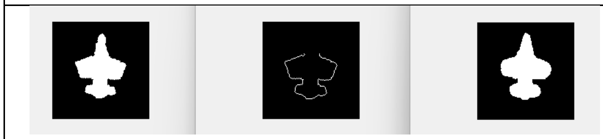
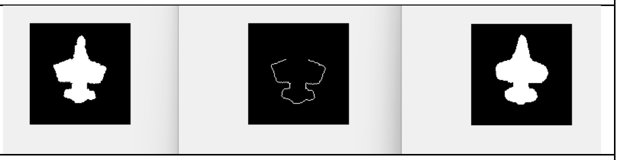
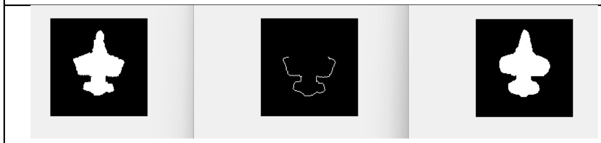
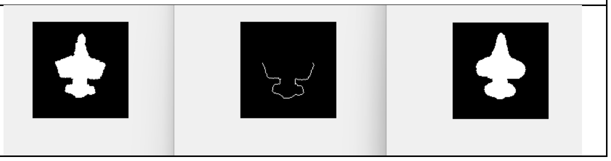
Appendix F

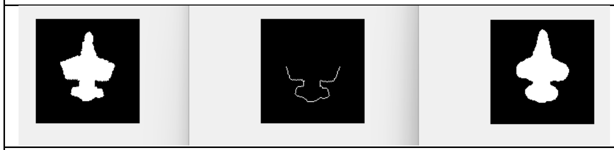
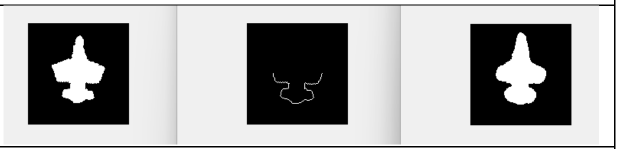
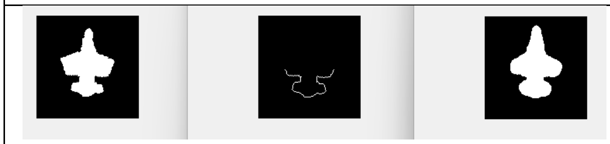
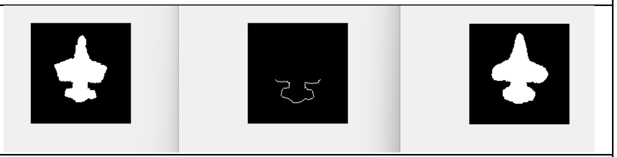
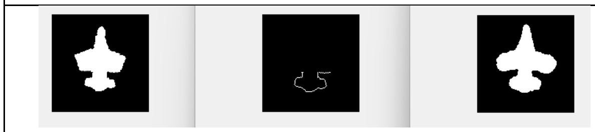
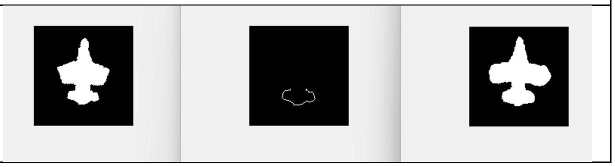
Horizontal Slicing Top-Down

Airplane 1

In order from left to right, the original airplane, the airplane outline, and the predicted airplane.

Table F-1. Airplane 1: Horizontal Slicing Top-Down (all data)

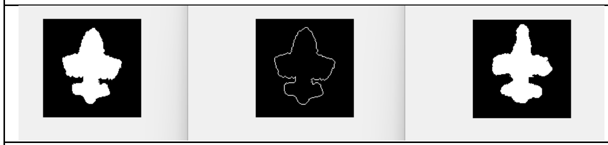
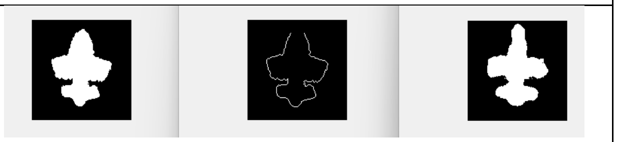
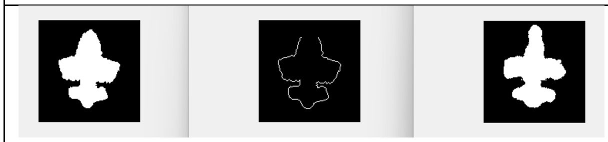
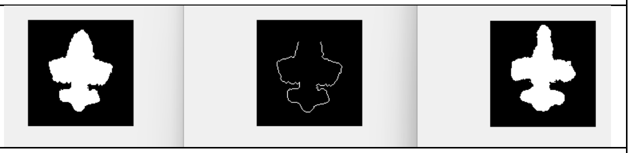
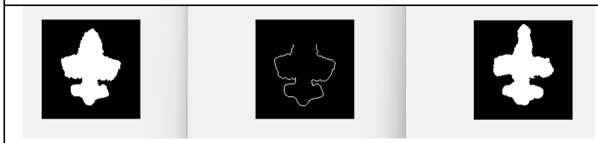
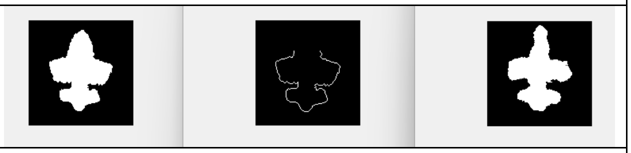
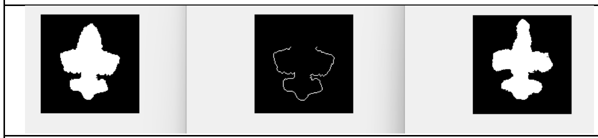
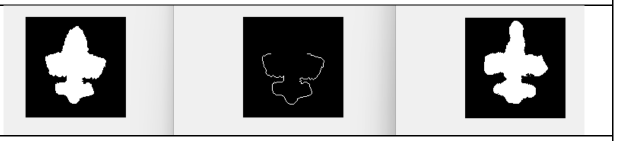
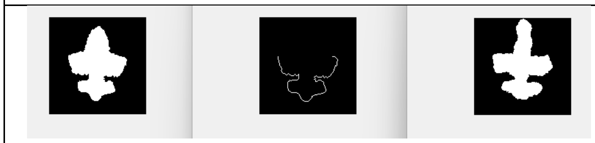
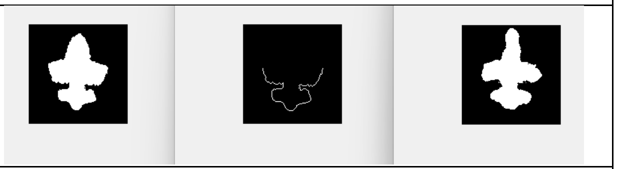
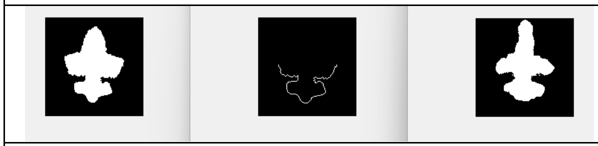
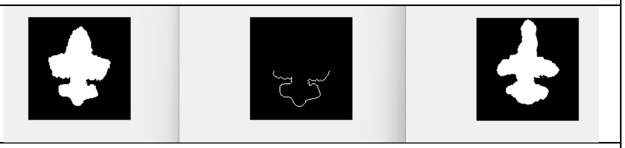
Full Outline	**95% Outline**
	
90% Outline	**85% Outline**
	
80% Outline	**75% Outline**
	
70% Outline	**65% Outline**
	
60% Outline	**55% Outline**
	

50% Outline	**45% Outline**
	
40% Outline	**35% Outline**
	
30% Outline	**25% Outline**
	

Airplane 2

In order from left to right, the original airplane, the airplane outline, and the predicted airplane.

Table F-2. Airplane 2: Horizontal Slicing Top-Down (all data)

Full Outline	**95% Outline**
	
90% Outline	**85% Outline**
	
80% Outline	**75% Outline**
	
70% Outline	**65% Outline**
	
60% Outline	**55% Outline**
	
50% Outline	**45% Outline**
	

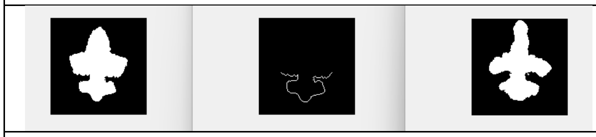
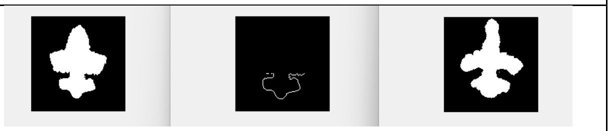
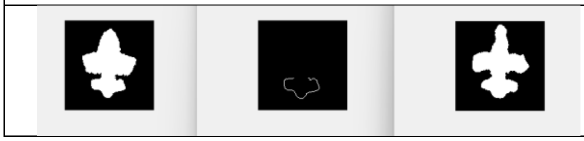
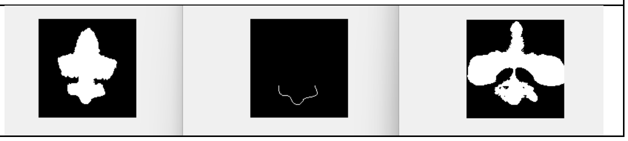
40% Outline	35% Outline
	
30% Outline	25% Outline
	

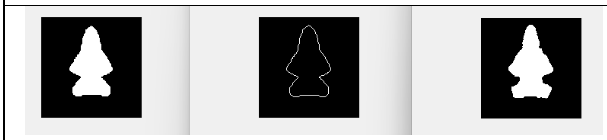
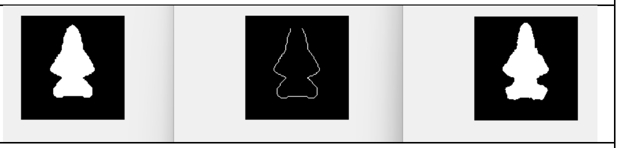
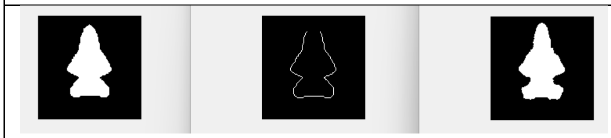
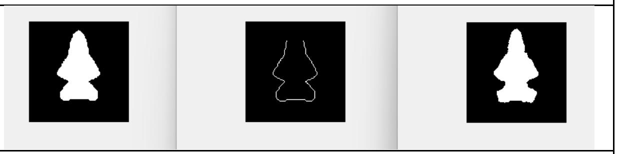
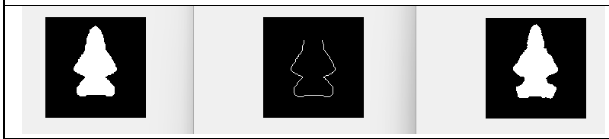
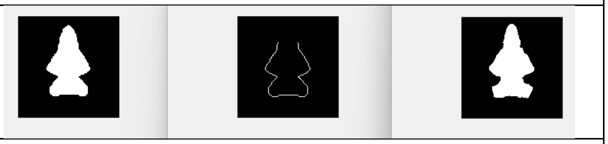
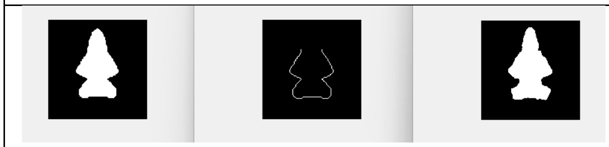
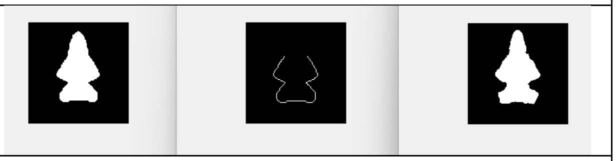
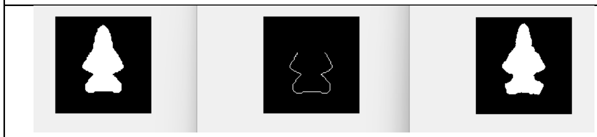
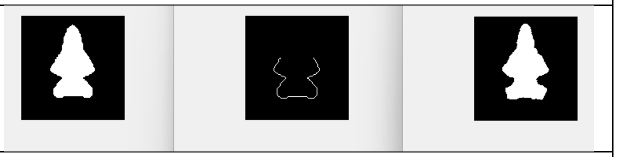
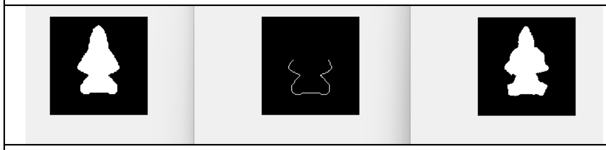
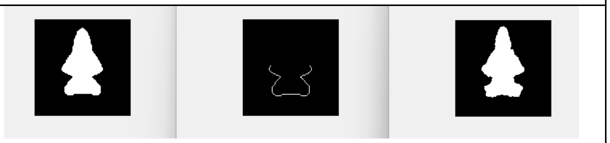
Table F-3. Similarity Values for Horizontal Slicing Top-Down of Airplane 2

Percent (%)	xcorr	sim	Err	psnr	Dot
100	0.8738	0.8025	0.0478	13.2102	0.9011
95	0.8710	0.8015	0.0488	13.1173	0.8998
90	0.8712	0.8009	0.0487	13.1224	0.8996
85	0.8734	0.8039	0.0482	13.1739	0.9005
80	0.8738	0.8043	0.0483	13.1635	0.8998
75	0.8739	0.8039	0.0482	13.1739	0.8998
70	0.8750	0.8053	0.0476	13.2207	0.9009
65	0.8624	0.7939	0.0526	12.7925	0.8929
60	0.8588	0.7968	0.0519	12.8450	0.8931
55	0.8679	0.8064	0.0482	13.1739	0.8993
50	0.8384	0.7857	0.0587	12.3127	0.8766
45	0.8359	0.7832	0.0596	12.2495	0.8746
40	0.8235	0.7710	0.0647	11.8884	0.8658
35	0.8227	0.7705	0.0653	11.8539	0.8651
30	0.8627	0.7969	0.0519	12.8450	0.8932
25	0.5179	0.5790	0.2129	6.7189	0.6548

Airplane 3

In order from left to right, the original airplane, the airplane outline, and the predicted airplane.

Table F-4. Airplane 3: Horizontal Slicing Top-Down (all data)

Full Outline	**95% Outline**
	
90% Outline	**85% Outline**
	
80% Outline	**75% Outline**
	
70% Outline	**65% Outline**
	
60% Outline	**55% Outline**
	
50% Outline	**45% Outline**
	

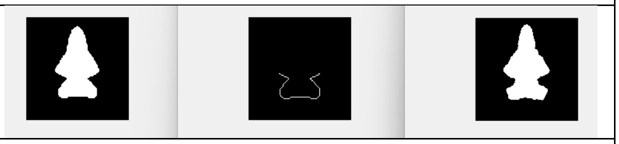
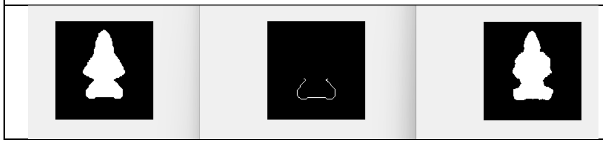
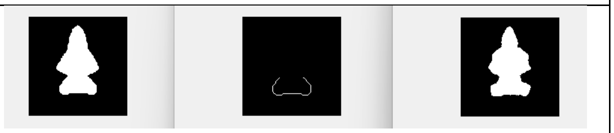
40% Outline	35% Outline
	
30% Outline	25% Outline
	

Table F-5. Similarity Values for Horizontal Slicing Top-Down of Airplane 3

Percent (%)	xcorr	sim	Err	psnr	Dot
100	0.9457	0.8912	0.0172	17.6403	0.9548
95	0.9408	0.8870	0.0190	17.2132	0.9505
90	0.9427	0.8892	0.0183	17.3872	0.9523
85	0.9458	0.8915	0.0172	17.6403	0.9548
80	0.9459	0.8918	0.0172	17.6548	0.9549
75	0.9459	0.8921	0.0170	17.6893	0.9552
70	0.9453	0.8907	0.0172	17.6403	0.9547
65	0.9451	0.8908	0.0173	17.6258	0.9546
60	0.9459	0.8914	0.0172	17.6548	0.9549
55	0.9460	0.8914	0.0171	17.6693	0.9550
50	0.9319	0.8792	0.0213	16.7060	0.9435
45	0.9472	0.8976	0.0165	17.8327	0.9566
40	0.9466	0.8911	0.0173	17.6114	0.9546
35	0.9452	0.8932	0.0170	17.6839	0.9552
30	0.9281	0.8739	0.0231	16.3584	0.9385
25	0.9297	0.8785	0.0223	16.5232	0.9412

Appendix G

MATLAB Code

```
%% Predicting 2D Geometrical Shapes Using Partial Contour Data
% and Principal Component Analysis

%% Create the statistical Shape Model
% this includes the average image and the shape modes with corresponding
% weights

clc,clear,clf

% Load in the images to create the average airplane and shape modes (model)

[files,path] = uigetfile({'*.jpg;*.png;*.tif;*.jpeg;*.tiff',['image files' ...
    ' (*.jpg,*.png,*.tif,*.jpeg,*.tiff)']},'MultiSelect','on');

% Send the images to the function that calculates the average plane and the
% shape modes. Get back the weights, eigenvectors for each shape mode and
% the average image.

[w, d_S, images_avg] = ssm_clean(files,path);

% Plot the average image

figure(1)
a = double(images_avg>=0.5);
imagesc(a)

%% Prediction

% Select an image to be predicted
[airplane12,path1] = uigetfile({'*.jpg;*.png;*.tif;*.jpeg;*.tiff', ...
    'image files (*.jpg,*.png,*.tif,*.jpeg,*.tiff)'},'MultiSelect','on');
nfiles = length(airplane12);

% Convert image to a binary image
[im_temp,map] = imread(airplane12);
   if size(im_temp,3)~=1
       im_temp = ind2gray(im_temp,map);
   end
images(1:size(im_temp,1),1:size(im_temp,2),1) = im_temp;
airplane12_bin = double(images>=10);

% Display original image
figure(23)
imshow(airplane12_bin)

% Pad array to ensure size is the same and to allow space for centering and
% rotating processes
```

```
airplane12_bin = padarray(airplane12_bin, [10 10], 0,'both');
airplane12_bin = imresize(airplane12_bin, [size(images_avg,1),size(images_avg,2)]);
airplane12_bin = double(airplane12_bin>=0.5);

% Center and Rotate Process

% Center image
s = regionprops(airplane12_bin,'centroid');
airplane12_bw = circshift(airplane12_bin, round([size(airplane12_bin,2) ...
    /2-s.Centroid(2),size(airplane12_bin,1)/2-s.Centroid(1),0]));

%rotate the image
images_binary_rotate = airplane12_bw*0;
ang = regionprops(airplane12_bw,'Orientation');
     if ang.Orientation(1) < 0
         true_ang = 180 + ang.Orientation(1);
     else
         true_ang = ang.Orientation(1);
     end
     final_ang = 90 - true_ang;
     airplane12 = imrotate(airplane12_bw, final_ang, 'Crop');

% Center the image
images_binary_centered = airplane12_bw*0;
s = regionprops(airplane12,'centroid');
airplane12 = circshift(airplane12, round([size(airplane12,2)/2-
s.Centroid(2),size(airplane12,1)/2-s.Centroid(1),0]));

% Calculate signed distance function (sdf)
images_sdf = airplane12_bw*0;
images_sdf= bwdist(airplane12)-bwdist(1-airplane12);

% Create image outline
one_points = find(images_sdf(:,:,1)==1);
image_outline = zeros(size(airplane12,1), size(airplane12,2));
for i = 1:size(one_points)
    image_outline(one_points(i)) = 1;
end

% Objective function (what we want to minimize)
% Obj_fun = sum(sum(abs(images_sdf).*image_outline));
% inputs to the function will be the weights
Obj_fun_value = objfun(w, d_S, images_avg, image_outline);

% Find min weights (w) for the objective function
w_guess = ones(1,length(w));
f = @(w_guess) objfun(w_guess, d_S, images_avg, image_outline);
[w_min, fval_fmin] = fminsearch(f, w_guess);
fval_initial = f(w);

% Using new minimized weights, create the predicted image
airp_pred = images_avg;
    for ee = 1:size(d_S,2)
    sum_part = w_min(ee).*d_S(:,ee);
    wd_S = reshape(sum_part, size(images_avg,1),size(images_avg,2));
    airp_pred = airp_pred + wd_S;
    end

% Show the new airplane prediction
airp_pred = imcomplement(airp_pred);
airp_pred = double(airp_pred>=0.5);
figure(2)
imshow(airp_pred)

% Determine Number of Eigenvectors to Use

% Cross-correlation
vector_airp = reshape(airp_pred, [1, length(airp_pred)*length(airp_pred)]);
vector_airp12 = reshape(airplane12, [1, length(airplane12)*length(airplane12)]);
x_corr = normxcorr2(vector_airp, vector_airp12);
x_corr_max = max(x_corr);
x_corr2 = normxcorr2(airp_pred,airplane12);
max_corr = max(max(x_corr2));

% Similarity, error, and psnr
sim = ssim(airp_pred, airplane12);
err = immse(airp_pred, airplane12);
psnr_value = psnr(airp_pred,airplane12);

% Dot product
A = norm(vector_airp);
B = norm(vector_airp12);
norm_vec_airp = vector_airp/A;
norm_vec_airp12 = vector_airp12/B;
dot_prod = dot(norm_vec_airp, norm_vec_airp12);

%% Partial Airplane sent in (Missing outline points)
% MUST RUN PREDICTION CODE PRIOR TO EXECUTING

% Now create image outline
one_points = find(images_sdf(:,:,1)==1);
image_outline = zeros(size(airplane12_bw,1), size(airplane12_bw,2));
inc = 12; % Increment for number of points
for i = 1:inc:size(one_points)
    image_outline(one_points(i)) = 1;
end
% Display image outline
figure(77)
imshow(image_outline)

% Objective function (what we want to minimize)
% Obj_fun = sum(sum(abs(images_sdf).*image_outline));
% inputs to the function will be the weights
Obj_fun_value = objfun(w, d_S, images_avg, image_outline);

% Find min weights (w) for the objective function
```

```
w_guess = ones(1,length(w));
f = @(w_guess) objfun(w_guess, d_S, images_avg, image_outline);
[w_min, fval_fmin] = fminsearch(f, w_guess);
fval_initial = f(w);

% Using new minimized weights, create the predicted image
airp_pred = images_avg;
    for ee = 1:size(d_S,2)
    sum_part = w_min(ee).*d_S(:,ee);
    wd_S = reshape(sum_part, size(images_avg,1),size(images_avg,2));
    airp_pred = airp_pred + wd_S;
    end

% Display image prediction
airp_pred = imcomplement(airp_pred);
airp_pred = double(airp_pred>=0.5);
figure(2)
imshow(airp_pred)

% Cross-correlation
vector_airp = reshape(airp_pred, [1, length(airp_pred)*length(airp_pred)]);
vector_airp12 = reshape(airplane12, [1, length(airplane12)*length(airplane12)]);
x_corr = normxcorr2(vector_airp, vector_airp12);
x_corr_max = max(x_corr);
x_corr2 = normxcorr2(airp_pred,airplane12);
max_corr = max(max(x_corr2))

% Similarity, error, adn psnr
sim = ssim(airp_pred, airplane12)
err = immse(airp_pred, airplane12)
psnr_value = psnr(airp_pred,airplane12)

% Dot product
A = norm(vector_airp);
B = norm(vector_airp12);
norm_vec_airp = vector_airp/A;
norm_vec_airp12 = vector_airp12/B;
dot_prod = dot(norm_vec_airp, norm_vec_airp12)


%% Slicing
clf
% HORIZONTAL SLICING

% Select an image to be predicted
[airplane12,path1] = uigetfile({'*.jpg;*.png;*.tif;*.jpeg;*.tiff', ...
    'image files (*.jpg,*.png,*.tif,*.jpeg,*.tiff)'},'MultiSelect','on');
nfiles = length(airplane12);

% Convert image to a binary image
[im_temp,map] = imread(airplane12);
   if size(im_temp,3)~=1
       im_temp = ind2gray(im_temp,map);
   end
%im_temp = im_temp./(max(max(im_temp)));
images(1:size(im_temp,1),1:size(im_temp,2),1) = im_temp;
% images = double(images);
airplane12_bin = double(images>=10);

% Pad array to ensure size is the same and to allow space for centering and
% rotating process
airplane12_bin = padarray(airplane12_bin, [10 10], 0,'both');
airplane12_bin = imresize(airplane12_bin, [size(images_avg,1),size(images_avg,2)]);
airplane12_bin = double(airplane12_bin>=0.5);

% Center and Rotate Process

% Center image
s = regionprops(airplane12_bin,'centroid');
airplane12_bw = circshift(airplane12_bin, round([size(airplane12_bin,2) ...
    /2-s.Centroid(2),size(airplane12_bin,1)/2-s.Centroid(1),0]));

%rotate the image
images_binary_rotate = airplane12_bw*0;
ang = regionprops(airplane12_bw,'Orientation');
     if ang.Orientation(1) < 0
         true_ang = 180 + ang.Orientation(1);
     else
         true_ang = ang.Orientation(1);
     end
     final_ang = 90 - true_ang;
     airplane12 = imrotate(airplane12_bw, final_ang, 'Crop');

% Center the image
images_binary_centered = airplane12_bw*0;
s = regionprops(airplane12,'centroid');
airplane12 = circshift(airplane12, round([size(airplane12,2) ...
    /2-s.Centroid(2),size(airplane12,1)/2-s.Centroid(1),0]));

% Display original Airplane
airplane_og = airplane12;
figure(15)
imshow(airplane_og)

% Calculate sdf
images_sdf = airplane12*0;
images_sdf= bwdist(airplane12)-bwdist(1-airplane12);

% Now create image outline
one_points = find(images_sdf(:,:,1)==1);
image_outline = zeros(size(airplane12,1), size(airplane12,2));
for i = 1:size(one_points)
    image_outline(one_points(i)) = 1;
end

% Determine the first and last rows/columns with nonzero elements
```

```
[R,C] = find(image_outline == 1);
first_col = min(C);
last_col = max(C);
first_row = min(R);
last_row = max(R);
P = 0.25; % percent of plane to set to zero

% Vertical sicing code:

% i = round((last_col - first_col)*P);
% image_outline(:, first_col:last_col-i) = 0;

% Horizontal slicing code (top-down):

% i = round((last_row - first_row)*P);
% image_outline(first_row:last_row-i,:) = 0;

% Horizontal slicing code (bottom-up):

i = round((last_row-first_row)*P);
image_outline(first_row+i:last_row,:) = 0;

% Display image outline
figure(18)
imshow(image_outline)

% Objective function (what we want to minimize)
% Obj_fun = sum(sum(abs(images_sdf).*image_outline));
% inputs to the function will be the weights
Obj_fun_value = objfun(w, d_S, images_avg, image_outline);

% Find min weights (w) for the objective function
w_guess = ones(1,length(w));
f = @(w_guess) objfun(w_guess, d_S, images_avg, image_outline);
[w_min, fval_fmin] = fminsearch(f, w_guess);
fval_initial = f(w);

% Using new minimized weights, create the predicted image
airp_pred = images_avg;
    for ee = 1:size(d_S,2)
    sum_part = w_min(ee).*d_S(:,ee);
    wd_S = reshape(sum_part, size(images_avg,1),size(images_avg,2));
    airp_pred = airp_pred + wd_S;
    end

% Display predicted image
airp_pred = imcomplement(airp_pred);
airp_pred = double(airp_pred>=0.5);
figure(2)
imshow(airp_pred)

% Similarity value calculations
% Cross-correlation
vector_airp = reshape(airp_pred, [1, length(airp_pred)*length(airp_pred)]);
vector_airp12 = reshape(airplane_og, [1, length(airplane_og)*length(airplane_og)]);
x_corr = normxcorr2(vector_airp, vector_airp12);
x_corr_max = max(x_corr);
x_corr2 = normxcorr2(airp_pred,airplane_og);
max_corr = max(max(x_corr2));

% Similarity, error, adn psnr
sim = ssim(airp_pred, airplane_og);
err = immse(airp_pred, airplane_og);
psnr_value = psnr(airp_pred,airplane_og);

% Dot product
A = norm(vector_airp);
B = norm(vector_airp12);
norm_vec_airp = vector_airp/A;
norm_vec_airp12 = vector_airp12/B;
dot_prod = dot(norm_vec_airp, norm_vec_airp12);


% SSM Function
function [w,d_S,images_avg] = ssm_clean(files,path)

% Number of files selected
nfiles = length(files);

% Read in each file & convert to binary
for ii = 1:nfiles
   [im_temp,map] = imread(files{ii},'png');
   if size(im_temp,3)~=1
       im_temp = ind2gray(im_temp,map);
   end
   images(1:size(im_temp,1),1:size(im_temp,2),ii) = im_temp;
end
images_binary = double(images>=0.5);

% Pad the image to ensure same size and room to shift for centering and
% rotation
pad_image = padarray(images_binary, [10 10], 0,'both');

% Prepare a variable for the rotated&centered image
images_binary_centered = pad_image*0;

% Center and Rotation Process

% Center image
for ii = 1:nfiles
    s = regionprops(pad_image(:,:,ii),'centroid');
    images_binary_centered(:,:,ii) = circshift(pad_image(:,:,ii), ...
        round([size(pad_image,2)/2-s.Centroid(2), ...
        size(pad_image,1)/2-s.Centroid(1),0]));
end
```

```
% Rotate the image
 images_binary_rotate = pad_image*0;
 for ii = 1:nfiles
     ang = regionprops(images_binary_centered(:,:,ii),'Orientation');
     if ang.Orientation(1) < 0
         true_ang = 180 + ang.Orientation(1);
     else
         true_ang = ang.Orientation(1);
     end
     final_ang = 90 - true_ang;
     images_binary_rotate(:,:,ii) = imrotate(images_binary_centered(:,:,ii), ...
         final_ang, 'bilinear', 'Crop');
 end

% Center the image
for ii = 1:nfiles
    s = regionprops(images_binary_rotate(:,:,ii),'centroid');
    images_binary_centered(:,:,ii) = circshift(images_binary_rotate(:,:,ii), ...
        round([size(images_binary_rotate,2)/2-s.Centroid(2), ...
        size(images_binary_rotate,1)/2-s.Centroid(1),0]));
end

% Calculate signed distance function
images_sdf = images_binary_centered*0;
for ii = 1:nfiles
    images_sdf(:,:,ii) = bwdist(images_binary_centered(:,:,ii) ...
        )-bwdist(1-images_binary_centered(:,:,ii));
end

% Calculate the average sdf
images_avg = mean(images_sdf,3);

% Calculate Average Binary Image
avg_image = images_avg*0;
avg_image(images_avg>=0) = 0;
avg_image(images_avg<=0) = 1;

% Find the difference between the average image and each of the independent
% images... this will help in determing the eigenvectors for the different
% shape modes

image_dif = pad_image*0;
for ii = 1:nfiles
    image_dif(:,:,ii) = images_sdf(:,:,ii) - images_avg;
end

% Reshape into a column vector --> S (Each image difference will be a
% column on the matrix S) (bring it back to 2D)

S = reshape(image_dif,[size(image_dif,1)*size(image_dif,2),nfiles]);
S_transpose = S';
```

```
% Calculate W, which will help to find the eigenvectors and eigenvalues
W = (1/nfiles)*S_transpose*S;

% Now need eigenvectors (v) of W. The number of eigenvectors taken will
% be determined later
[v,d] = eigs(W,6);

% Need eigenvectors of S*S_Transpose which is just S*v
d_S = S(:,:)*v(:,:);

% Last, need to calculate the weights associated with each shape mode
% (eigenvector d_S)
w = [];
for cc = 1:size(d_S,2)
% Note: d_S is the eigenvectors of S*S'
    dphi = images_sdf(:,:,1) - images_avg;
    dphi_col = reshape(dphi, [size(dphi,1)*size(dphi,2),1]);
    LHS = dot(dphi_col,d_S(:,cc));
    mag_phi1 = dot(d_S(:,cc),d_S(:,cc));
    w(cc) = LHS/mag_phi1;
end

end

% Objective function to minimize. Minimize the difference between the
% predicted image and the image outline.

function [obj_fun_value] = objfun(w, d_S, images_avg, image_outline)
    airplane12_fun = images_avg;
    % Alter w
    for ee = 1:size(d_S,2)
    sum_part = w(ee).*d_S(:,ee);
    wd_S = reshape(sum_part, size(images_avg,1),size(images_avg,2));
    airplane12_fun = airplane12_fun + wd_S;
    end
    % Minimize difference between airplane prediction and the actual image
    % outline
    number_of_pts = sum(sum(image_outline));
    obj_fun_value = (sum(sum(abs(airplane12_fun).*image_outline)))/number_of_pts;
end
```

BIBLIOGRAPHY

[1] Baliga, Santosh, et al. “Low back pain: Current surgical approaches,” *Asian Spine Journal*, vol. 9, no. 4, Jul. 2015. doi:10.4184/asj.2015.9.4.645

[2] Casiano, Vincent E., et al. *Back Pain.* [Updated Dec. 2023]. In: StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing; 2024 Jan-. Available from: https://www.ncbi.nlm.nih.gov/books/NBK538173/

[3] Ungi, Tamas *et al.*, “Spinal curvature measurement by tracked ultrasound snapshots,” *Ultrasound in Medicine & Biology*, vol. 40, no. 2, Feb. 2014. doi:10.1016/j.ultrasmedbio.2013.09.021

[4] “Ultrasound,” *National Institute of Biomedical Imaging and Bioengineering*, U.S. Department of Health and Human Services, https://www.nibib.nih.gov/science-education/science-topics/ultrasound#:~:text=When%20used%20in%20an%20ultrasound,tissue%20or%20tissue%20and%20bone). (accessed Mar. 2024).

[5] Ashmore, Zachary M., *et al.*, “Ultrasound-guided lumbar medial branch blocks and intra-articular facet joint injections: A systematic review and meta-analysis,” *PAIN Reports*, vol. 7, no. 3, May 2022. doi:10.1097/pr9.0000000000001008

[6] Brudfors, Mikael *et al.*, “Towards real-time, tracker-less 3D ultrasound guidance for Spine Anaesthesia,” *International Journal of Computer Assisted Radiology and Surgery*, vol. 10, no. 6, Apr. 2015. doi:10.1007/s11548-015-1206-1

[7] Peloquin, John M. *et al.*, “Human L3L4 intervertebral disc mean 3D shape, modes of variation, and their relationship to degeneration,” *Journal of Biomechanics*, vol. 47, no. 10, Jul. 2014. doi:10.1016/j.jbiomech.2014.04.014

[8] Abler, Daniel *et al.*, “A statistical shape model to predict the premorbid glenoid cavity,” *Journal of Shoulder and Elbow Surgery*, vol. 27, no. 10, Oct. 2018. doi:10.1016/j.jse.2018.04.023

[9] Tsai, Andy *et al.*, “A shape-based approach to the segmentation of medical imagery using level sets,” *IEEE Transactions on Medical Imaging*, vol. 22, no. 2, Feb. 2003. doi:10.1109/tmi.2002.808355

[10] Çeliktutan, Oya *et al.*, “A comparative study of face landmarking techniques,” *EURASIP Journal on Image and Video Processing*, vol. 2013, no. 1, Mar. 2013. doi:10.1186/1687-5281-2013-13

[11] Clogenson, Marine *et al.*, “A statistical shape model of the human second cervical vertebra,” *International Journal of Computer Assisted Radiology and Surgery*, vol. 10, no. 7, Oct. 2014. doi:10.1007/s11548-014-1121-x