

# 3D Face Landmarks Project Final Report

Student: Richard Bi

Mentor: Dr. Nic Herndon

July 2022

## 1 Introduction

Face landmarks have a wide range of applications within the medical field. For instance, they have been used to analyze patients with sleep apnea [7] and to diagnose fetal alcohol syndrome [15]. Outside of the previously mentioned examples, one important application of face landmarks to the medical field is the use of face landmarks in assisting the design and visualization of dentures [2]. Currently, 2D face landmarks are primarily used for the design and visualization of dentures; however, there are inherent limitations of landmarking 2D images of edentulous patients. Specifically, landmarking 2D face images of edentulous patients can be inaccurate and pose sensitive. With an eye on mitigating the inherent limitations of landmarking 2D face images of edentulous patients, we propose the use of 3D face landmarking of edentulous patients. 3D face scans mitigate the issues associated with 2D face images, because the extra geometric information available in 3D face scans can help improve the accuracy of landmark placement and because 3D face scans are inherently not pose sensitive. In this project, we develop a process for landmarking 3D face scans that could be applied to edentulous patients within the context of improving the design and visualization of dentures.

## 2 Related Works

Over the past 10 years, myriad methods have been developed for the detection of landmarks on 3D face scans. Broadly speaking, these techniques can be divided into the following two categories: those whose criteria for landmark detection are manually specified by researchers and those whose criteria for landmark detection are learned from a set of training data.

Within the realm of 3D face landmark detection schemes that require researcher-specified landmark detection criteria, much of the work conducted in the past 10 years has been focused on harnessing various geometric descriptors derived from 3D face scans. In [21], Vezzetti and Marcolin proposed a method for detecting landmarks on 3D face scans that utilized measurements such as the first, second, and mixed derivatives; the coefficients of the first and second fundamental

forms; the maximum and minimum principal curvatures; the Gaussian and mean curvatures; and the Shape and Curvedness Indices. With these descriptors in hand, Vezzetti and Marcolin developed a two-stage process for identifying face landmarks. First, the above descriptors were used to select a subset of the points within a given 3D face scan that were candidates for the target landmark. Then, among these candidate points, one of the features, typically a coefficient of a fundamental form, was maximized or minimized to finalize the location of the target landmark. In [21], nine landmarks located in the vicinity of the nose and eyes were detected; however, [21] did not specify a process for finding landmarks located near the mouth or brow. This omission was rectified by another paper by Vezzetti and Marcolin [22]. In this paper, Vezzetti and Marcolin employed a method similar to the method utilized in their previous paper to detect landmarks around the mouth and brow on 3D face scans. Although the core of their technique remained unchanged, one addition Vezzetti and Marcolin applied to their method in [22] was the inclusion of a point's position relative to previously located landmarks when evaluating that point's candidacy for the target landmark. In [13], Liu et al. devised a strategy that fully integrated global and local constraints to locate landmarks on 3D face scans. In particular, for a given target landmark, Liu et al. specified the position of the target relative to previously located landmarks as the global constraints and utilized thresholds on the same geometric descriptors used in [21] as the local constraints. The method described in [22] differed from those described by Vezzetti and Marcolin as it relied on a more varied array of descriptors when conducting the optimization necessary for finalizing landmark locations. In [6], Galvánek et. al explicated another method for finding 3D face landmarks that combined both global and local constraints. Similar to previously mentioned papers, [6] based its search for landmarks off of the positions of theses landmarks relative to the pronasal. However, unlike the methods from the previously mentioned papers, the method from [6] employed only the maximum and minimum principal curvatures as the local constraints and independently calculated these curvatures for each landmark it located. Moreover, in [6], landmarks located in the center of the face (e.g. labrum superior, nasion, etc.) are extracted by determining the locations of local minima and maxima along the symmetry profile of the face. While many of the above methods utilized the same or very similar geometric descriptors, it is an open question whether there exists more powerful descriptors that have yet to be wielded in the pursuit of locating landmarks on 3D face scans. To provide insight into this question, in [14], Marcolin and Vezzetti investigated the utility of new descriptors generated from the aforementioned base geometric properties (derivatives, coefficients of the fundamental forms, curvatures, etc.). Specifically, new descriptors were derived from the base properties through the application of a function like mean, median, natural logarithm, sine, cosine, tangent, arcsine, arccosine, or arctangent, or new descriptors were composed by taking linear combinations, fractions, products, etc. of the base properties. Marcolin and Vezzetti found that some of the new descriptors were more powerful than the original base properties because these new descriptors either highlighted the dividing line between different regions of the face or varied

their behavior across the face in a manner that allowed for the easy identification of different regions of the face.

Outside of methods for landmark detection on 3D face scans that require researcher-inputted criteria, there are also methods that learn the criteria for landmark detection from a set of training data. These methods can be divided into two categories: those that utilize convolutional neural networks (CNNs) for landmark localization and those that do not.

Many 3D face landmarking techniques that rely on CNNs utilize 2D representations of 3D face scans. This is because a volumetric CNN that could operate on detailed 3D face scans would consume too much memory to be practical [17]. Thus, the methods explicated in [17, 20, 23] all first created 2D representations of the 3D face scans before detecting landmarks. There are multiple approaches to creating these 2D representations of 3D faces. In [17], Paulsen et. al. extracted 2D representations of 3D face scans by using a multiple-view approach, where images of a particular face scan were captured from different perspectives. On the other hand, a single-view approach was adopted in [20, 23]. Specifically, in [23], Zhang et. al. first determined the orientation of the 3D face scan before rotating the face into a standard position and projecting the face onto a 2D surface. In all three papers, after 2D representations of the 3D face scans were obtained, landmarking occurred on the 2D representations using CNN. These landmarks were then projected back into 3D. In [20, 23], this projection of the detected 2D landmarks into 3D was a relatively straightforward inversion of the 3D to 2D projection. However, in [17], this inverse projection was complicated by the need to correspond the 3D landmark location for a given landmark with the 2D landmark locations for that landmark from multiple views. To accomplish this task, Paulsen et. al. converted each 2D landmark detected across the multiple views into a line in 3D. Then, to determine the location of a given 3D landmark, the lines derived from the 2D locations of this landmark were intersected.

Similar to the above CNN-based methods, the Gabor wavelet-based approach described in [4] also used 2D representations of 3D face scans. In that approach, the face was first registered by fitting it to an ellipsoid. Then, a Mercator projection was utilized to project the 3D face scan onto the ellipsoid, and five different 2D maps were extracted. These maps corresponded to texture, height of the scan above the ellipsoid (relief map), derivatives with respect to the x and y axes, and the Laplacian of Gaussian. After these maps were extracted, an EBGM algorithm for 2D face landmark detection was employed to detect the locations of the landmarks in 2D. Then, the locations of these 2D landmarks were projected back into 3D using the height above the ellipsoid stored in the relief map. As with the approach described in [4], the three-stage process described in [19] also utilized 2D representations of face scans during a portion of its landmark localization. The approach described in [19] first used curvature to pinpoint the location of the tip of the nose and the inner corners of the eyes. Then, maps describing the x, y, and z components of the normal vectors were generated, and an Active Normal Model was trained to locate the face landmarks. As a final step, local binary features and cascaded regression were

combined to tune the landmark locations predicted by the Active Normal Model. Another approach that makes use of local binary patterns is the approach described in [5]. In [5], the authors built a constrained local model (CLM), which used patch experts to determine the probability that all landmarks were placed correctly by analyzing features in the local region that surrounded the predicted location of each landmark. In particular, the patch experts used included three histogram-based descriptors: mesh local binary patterns (mesh-LBP), mesh-SIFT, and mesh histogram of gradients (mesh-HOG). Mesh-LBP operated by determining whether a given scalar value (such as curvature) exceeded a designated threshold for a set of facets arranged in a ring around the target landmark. Mesh-SIFT built histograms using the descriptors of shape index and slant angle for a set of circular regions around the target landmark. And mesh-HOG built histograms by measuring the gradient of a scalar function such as curvature in a circular region around the target landmark. The approach described in [3] is another approach that employed a CLM. However, the authors of [3] emphasized the use of features derived from the face scans' normal vectors as patch experts. Specifically, the authors created a histogram of normal vectors to describe the normal vectors of the cells near each landmark and utilized local normal binary patterns, which calculated histograms for the landmarks based on how closely the normal vectors for a ring of cells around the landmark aligned with the normal vector at the landmark. Unlike the previous two approaches, the approaches outlined in [7] and [1] did not build CLMs. Instead, the approach in [7] sampled points from both high and low curvature regions of the face and built a mean face by corresponding these sampled points across its training faces through the minimization of bending energy. Then, to determine the locations of landmarks on a target 3D face scan, the mean face was reshaped through rigid transformations and non-rigid deformations. The final landmark locations were determined by using a nearest-neighbor search from the morphed mean face to the target face. In a similar vein, the approach outline in [1] generated a template mesh by placing and corresponding 16 anatomical landmarks and 484 randomly and uniformly distributed semi-landmarks on each training face. Then, target faces were landmarked by fitting the template mesh to the target mesh through six iterations of bending energy minimization, which ultimately allowed for the extraction of landmarks.

### 3 Method

An overview of the process for locating landmarks on 3D face scans is described in Figure 1. Briefly, the first step for placing landmarks is preprocessing. During this step, the target face scan is oriented into a standard position, and the relevant geometric descriptors for the points within the scan are calculated. After preprocessing, the location of each landmark is determined. To do so, points of interest are first located for each landmark. Then, a combination of a regression model and clustering are employed to predict the final location of each landmark.

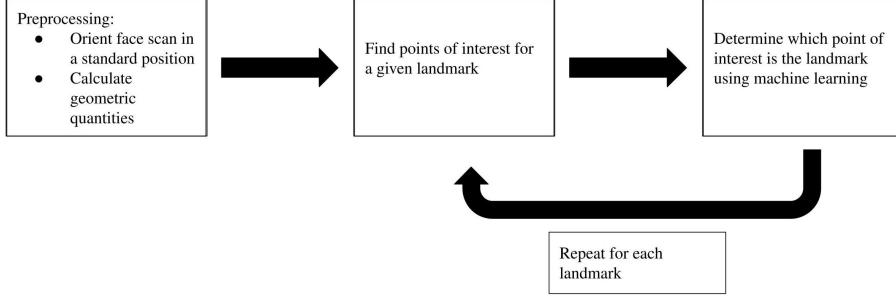


Figure 1: An overview of the process for locating landmarks on 3D face scans

### 3.1 Orientation

The standard orientation for face scans within this project is described in Figure 2. As seen in that figure, the  $x$ -axis is horizontal with respect to the face, and the  $y$ -axis is vertical with respect to the face. The  $z$ -axis is the axis orthogonal to both the  $x$  and  $y$  axes that comes out of the sheet.

To orient the face into the standard position, we first assume the target face scan is a PyVista PolyData object [18] without occlusions whose points can be described by the  $3 \times n$  matrix  $F_1$ , where  $n$  is the number of points in the target face scan. This target face scan may potentially be rotated into an arbitrary orientation. Let the  $3 \times n$  matrix  $F_0$  represent the points of the face scan after it has been rotated into the standard position. Then, if we let  $T$  be the linear transformation that rotates the points of  $F_0$  into the points of  $F_1$ , we have  $F_1 = [T(\vec{e}_1) \ T(\vec{e}_2) \ T(\vec{e}_3)] F_0$  or

$$F_0 = [T(\vec{e}_1) \ T(\vec{e}_2) \ T(\vec{e}_3)]^{-1} F_1,$$

where  $\vec{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ , the vector that points in the  $+x$ -direction;  $\vec{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ , the vector that points in the  $+y$ -direction; and  $\vec{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ , the vector that points in the  $+z$ -direction. Thus, to determine  $F_0$ , we determine  $T(\vec{e}_1)$ ,  $T(\vec{e}_2)$ , and  $T(\vec{e}_3)$ .

To determine  $T(\vec{e}_1)$ , we note that when  $T(\vec{e}_1) = \vec{e}_1$ , the components of the face mesh's unit normal vectors along  $\vec{e}_1$  generally form the bowl-shaped distribution seen in Figure 3. This is due to the symmetry of a face scan in the standard orientation about the  $yz$ -plane and the large, flat sides of the face. Thus, to find  $T(\vec{e}_1)$ , we create an objective function whose input is a vector  $\vec{v} \in \mathbb{R}^3$ . This objective function should be minimized when  $\frac{\vec{v}}{\|\vec{v}\|} = T(\vec{e}_1)$ . For this project, we use an objective function that calculates the sum of squared residuals between a histogram generated using the components of the



Figure 2:  $x$  and  $y$  axes

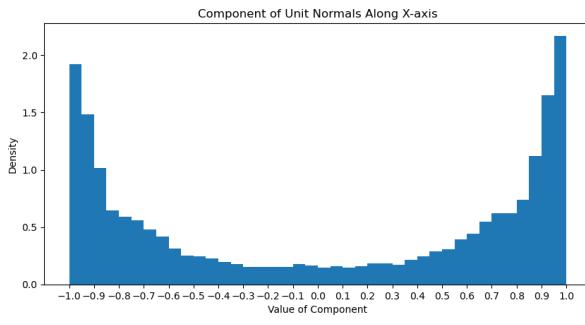


Figure 3: Unit normal components along  $T(\vec{e}_1)$

unit normal vectors of the target face scan along  $\frac{\vec{v}}{|\vec{v}|}$  and an idealized distribution of a vertically and horizontally scaled secant function. Then, we employ SciPy’s optimize module to determine the vector  $\vec{v} \in \mathbb{R}^3$  that minimizes the objective function, and let  $T(\vec{e}_1) = \frac{\vec{v}}{|\vec{v}|}$ .

A similar process is undertaken to determine  $T(\vec{e}_3)$ . In particular, we note that when  $T(\vec{e}_3) = \vec{e}_3$  and when the mesh’s normal vectors are oriented such that they point away from the mesh’s center of mass, the vast majority of the components of the normal vectors along  $T(\vec{e}_3)$  are positive. Thus, we first guarantee that the normal vectors of the face mesh point away from the mesh’s center of mass. Then, we define our objective function with input  $\vec{v} \in \mathbb{R}^3$  to be the negative mean of the components of the face mesh’s unit normal vectors along  $\frac{\vec{v}}{|\vec{v}|}$ . This function should be minimized when  $\frac{\vec{v}}{|\vec{v}|} = T(\vec{e}_3)$ . Once again, we use SciPy’s optimize module to determine the vector  $\vec{v} \in \mathbb{R}^3$  that minimizes our objective function. Additionally, when determining  $T(\vec{e}_3)$ , we also enforce the fact that  $[T(\vec{e}_1) \ T(\vec{e}_3)]$  is a set of orthonormal vectors. To do so, we let  $u = v - \text{comp}_{T(\vec{e}_1)}v$  and let  $T(\vec{e}_3) = \frac{\vec{u}}{|\vec{u}|}$ .

Finally, since  $[T(\vec{e}_1) \ T(\vec{e}_2) \ T(\vec{e}_3)]$  forms an orthonormal set of vectors, we know that  $T(\vec{e}_2) = T(\vec{e}_1) \times T(\vec{e}_3)$  or  $T(\vec{e}_2) = T(\vec{e}_3) \times T(\vec{e}_1)$ . To complete the construction of  $[T(\vec{e}_1) \ T(\vec{e}_2) \ T(\vec{e}_3)]^{-1}$ , we initially choose  $T(\vec{e}_2) = T(\vec{e}_1) \times T(\vec{e}_3)$ .

After determining  $[T(\vec{e}_1) \ T(\vec{e}_2) \ T(\vec{e}_3)]^{-1}$ , we find  $F_0$ . The results for this rotation from  $F_1$  back to  $F_0$  are not always satisfactory after a single iteration. Accordingly, we repeat the process for finding  $[T(\vec{e}_1) \ T(\vec{e}_2) \ T(\vec{e}_3)]^{-1}$  outlined above until we reach an iteration where each of  $T(\vec{e}_1)$ ,  $T(\vec{e}_2)$ , and  $T(\vec{e}_3)$  lie on approximately the same lines as the  $T(\vec{e}_1)$ ,  $T(\vec{e}_2)$ , and  $T(\vec{e}_3)$  found in the previous iteration. More concretely, we iterate the above process until the norm of the element-wise absolute value of  $[T(\vec{e}_1) \ T(\vec{e}_2) \ T(\vec{e}_3)]^{-1} - I_{3,3}$  is within a certain tolerance of 0.

Finally, to complete the process of orienting the target face scan into a standard position, we correct for the arbitrary choice of  $T(\vec{e}_2)$  we made earlier. In particular, we first locate the tip of the nose or pronasal as the point with the largest  $z$ -value from all the points within the face scan. Then, we examine the set of points in the neighborhood of the pronasal, dividing this set of points into two subsets: the set of points with a larger  $y$ -value than the pronasal and the set of points with a smaller  $y$ -value than the pronasal. We then approximate the derivative of  $z$  with respect to  $y$  ( $z_y$ ) at each of these points using the unit normal

vectors. In particular, if the unit normal vector at a given point  $\vec{n} = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix}$ , we

find that  $z_y = -\frac{n_y}{n_z}$ . Using  $z_y$  calculated at each point, we calculate the mean magnitude of  $z_y$ ,  $|z_y|$ , for both the points with larger  $y$ -values than the pronasal and the points with smaller  $y$ -values than the pronasal. If our choice for  $T(\vec{e}_2)$  was correct, we should find that  $|z_y|$  for the points below the pronasal is larger than  $|\bar{z}_y|$  for the points above the pronasal. When  $|\bar{z}_y|$  for the points above the pronasal is larger than  $|z_y|$  for the points below the pronasal, we flip the face mesh we obtained after finding  $[T(\vec{e}_1) \ T(\vec{e}_2) \ T(\vec{e}_3)]^{-1}$  about the  $xz$ -plane.

### 3.2 Geometric Descriptors

After a target face scan has been rotated into the standard orientation, geometric descriptors of the target face that are needed for locating landmarks can be calculated. Specifically, the following geometric descriptors are calculated: the first derivatives of  $z$ ; the second derivatives of  $z$ ; the coefficients of the first fundamental form,  $E$ ,  $F$ , and  $G$ ; the coefficients of the second fundamental form,  $L$ ,  $M$ ,  $N$ ; the first derivatives of the coefficients of the first and second fundamental forms; the second derivatives of the coefficients of the first and second fundamental forms; the Gaussian curvature,  $K$ ; the mean curvature,  $H$ ; the maximum and minimum principal curvatures; the curvedness index,  $C$ ; and the shape index,  $S$ . A detailed description of the geometric descriptors excluding  $C$  and  $S$  can be found in [16]. The descriptors  $C$  and  $S$  are detailed in [12]. Of the descriptors that may be unfamiliar, the descriptor  $S$  deserves special attention. The descriptor  $S$  can vary within the range  $[-1, 1]$ , and different values within this range specify different shapes of the surface. In particular, if  $S \approx -1$  for a point, the shape of the surface at that point is a cup or a depression. On the other hand, if  $S \approx 1$  at a point, the surface at that point is a dome. If  $S \approx 0$  at a point, the surface at that point is flat, like a plane.

To calculate the first and second derivatives, we first store the  $z$ -values of the points within the face scan as a data array of PyVista PolyData object [18] that represents the face scan. Then, we project the mesh onto the  $xy$ -plane and apply PyVista's compute\_derivative function [18] on our stored  $z$ -values to calculate the first derivatives of  $z$ . To compute the second derivatives of  $z$ , we simply apply the compute\_derivative function to our calculated first derivatives. All the remaining geometric descriptors can be calculated by simply applying their

respective formulas after the first and second derivatives of  $z$  are calculated.

Finally, after all the geometric descriptors have been calculated, a second PyVista PolyData object [18] is created as a deep copy of the face scan. For this PolyData object, the values for each geometric descriptor are scaled such that a given value is replaced by that value's z-score. This is the "scaled face mesh," and the geometric descriptors for this mesh are the "scaled geometric descriptors." These terms will be useful for section 3.4 of the paper.

### 3.3 Landmark Points of Interest

After the geometric descriptors are calculated for the target face mesh in standard orientation, the points of interest for each landmark can be located. Given a landmark, that landmark's points of interest are the set of points whose geometric descriptors are highly similar to the geometric descriptors of the landmark. To search for points of interest, we outline a set of criteria involving the geometric descriptors introduced in the previous subsection, and we select points that fulfil all criteria as points of interest.

#### 3.3.1 Pronasal

In this project, the pronasal is unique in that it has only one point of interest. We specify that the point in the oriented target face scan with the largest  $z$ -value is the point of interest for the pronasal. Because there is only one point of interest, this point is automatically the predicted location of the pronasal landmark. Accordingly, the refinement procedure outlined in section 3.4 can be skipped for the pronasal landmark.

#### 3.3.2 Endocanthions

To find the points of interest for the endocanthions or the inner eye corners, we first filter the points within the face scan using the following two requirements: First, the shape indices,  $S$ , for points of interest must be within the range [-1, -0.375]; second, the points of interest for the endocanthions must be above the pronasal, with a distance from the pronasal between 7.5% and 20% of the range of  $y$ -values for the points within the mesh. After points have been filtered by these initial two requirements, we calculate  $z_x^2 + z_y^2$  for every remaining point and keep only the 10% of the points that passed through the initial filter with the smallest values for  $z_x^2 + z_y^2$ . Then, we partition the remaining points into two sets: points with  $x$ -values smaller than that of the pronasal (i.e. points to the left of the pronasal) and points with  $x$ -values larger than that of the pronasal (i.e. points to the right of the pronasal). Finally, for each these two sets of points, we select the  $\frac{1}{1000}n$  points with the largest Gaussian curvatures,  $K$ , where  $n$  is the number of points in the mesh (which has possibly been clipped during the calculation of geometric descriptors). Finally, we return these two sets of selected points as the points of interest for the endocanthions. An example of the points of interest (POIs) for the endocanthions is shown in Figure 4.

### 3.3.3 Cheilions

To find the points of interest for the cheilions or the corners of the mouth, we first filter the points within the face on the basis of their position relative to the pronasal. In particular, we only consider points whose distances along the  $x$ -axis from the pronasal exceed 10% of the range of the  $x$ -values in the face mesh. Additionally, we require that points of interest have smaller  $y$ -values than the pronasal and that points of interest are at least a distance of 10% of the range of the  $y$ -values in the face mesh from the pronasal in the  $y$ -direction. We also require that the distance between points of interest and the pronasal in the  $z$ -direction is no more than 25% of the range of the  $z$ -values in the face mesh. Next, of the points whose positions relative to the pronasal are valid, we select only the points whose shape indices are in the range  $(-1, 0)$ . We then partition the remaining points based on whether they lie to the left (smaller  $x$ -values than the pronasal) or to the right (larger  $x$ -values than the pronasal) of the pronasal. Finally, we select the 50% of points within each of the two sets with the largest curvedness indices,  $C$ , and return these points as points of interest for the cheilions. An example of the points of interest (POIs) for the cheilions is shown in Figure 5.

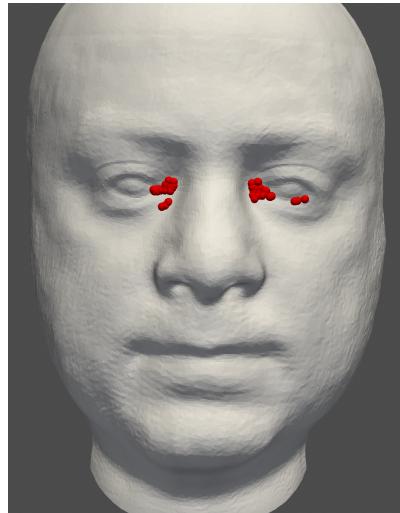


Figure 4: Endocanthion POIs

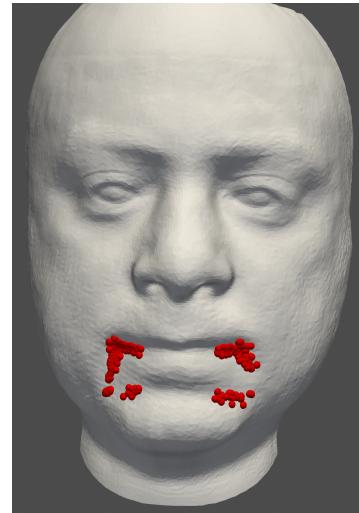


Figure 5: Cheilion POIs

### 3.4 Landmark Refinement

After finding the points of interest for a given landmark, it's generally necessary to refine these points of interest to obtain the final predicted location of the landmark. The first step in refining a given landmark's location is the use of a regression model that predicts each point of interest's distance from the

actual location of landmark. After this regression is complete, the results of the regression model and the results of an application of clustering to the points of interest are combined to deduce the final predicated location of the landmark.

### 3.4.1 Regression Model Training

To conduct landmark refinement as described above, it's necessary to train a regression model for each landmark that can predict the Euclidean distance between the actual location of the landmark and a given point of interest for this landmark.

The first step in creating these regression models for landmark refinement is the acquisition of a dataset with labelled landmarks. For this project, the Texas 3D Face Recognition Database (Texas3DFRD) is used [10,11]. In particular, the 2D representations of 3D face scans from the Texas3DFRD are converted to 3D face scans using 3DDFA\_V2 [8,9]. Then, the face landmarks for the faces in the database are extracted by projecting the landmarks generated by 3DDFA\_V2 onto the surface of the face meshes.

After the acquisition of a dataset of annotated 3D face scans, the training of the regression models for each landmark can commence. First, the points of interest for a target landmark are found on each face within the database of annotated 3D face scans. Then, for each point of interest, a feature vector is generated. This feature vector contains the scaled geometric descriptors associated with this point of interest, the Euclidean distance from this point of interest to the target landmark's location on the face from which this point of interest was extracted, and information about the point of interest's coordinates. For the endocanthions, the feature vector includes the scaled  $x$ ,  $y$ , and  $z$  coordinates of each point of interest. For the cheilions, the feature vector only includes the scaled  $x$  and  $z$  coordinates. Finally, after a feature vector has been generated for every point of interest for a given landmark on every face scan in the annotated face database, an AdaBoost regressor is trained to regress the target variable, which is the Euclidean distance between a point of interest and the actual landmark location. This AdaBoost regressor uses a decision tree regressor of maximum depth 20 as its base regressor and contains an ensemble of 50 estimators.

### 3.4.2 Landmark Refinement on a Query Face

Having determined the points of interests for a given landmark on a query face and trained a regression model for that landmark, we can now refine those points of interests into a final predicated landmark location. We do so by first predicting the Euclidean distance from each point of interest to the actual landmark location using the regression model. Then, we cluster the points of interest based on their  $x$ ,  $y$ , and  $z$  coordinates using the OPTICS algorithm. The minimum cluster size we use with OPTICS is  $\frac{1}{2}m$ , where  $m$  is the number of points of interest. Next, for each cluster, we determine the median predicted Euclidean distance between the points of interest within the cluster and the

actual landmark. We then select the cluster with the minimum median predicted Euclidean distance between the points of interest within the cluster and the actual landmark. Finally, the predicted location of the landmark on the query face is determined by calculating the median  $x$ ,  $y$ , and  $z$  coordinates for the points of interest in the selected cluster and then projecting the point composed of the median coordinates onto the query face using a nearest-neighbor search.

## 4 Results

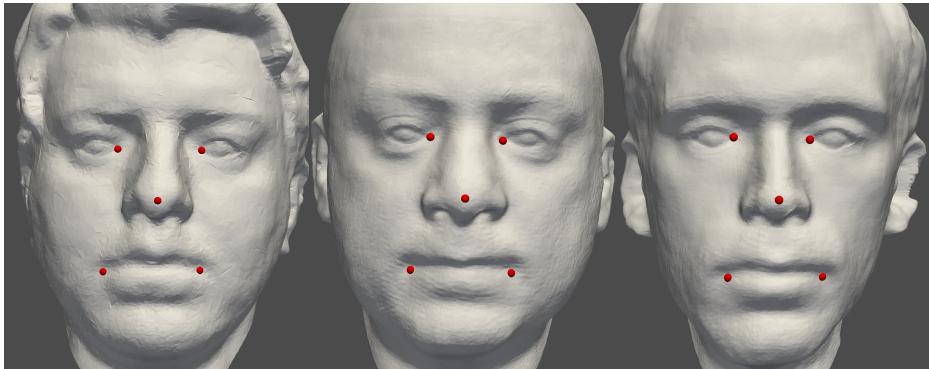


Figure 6: Results for neutral facial expressions

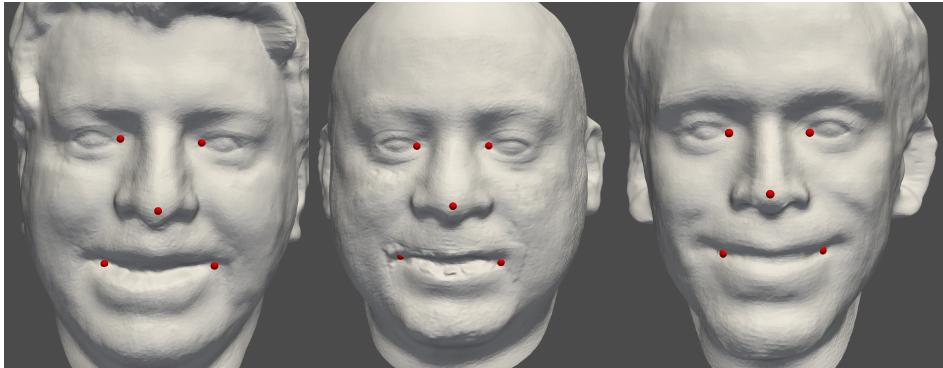


Figure 7: Results for smiling facial expressions

Above, we show the results of our landmark placement process for six 3D face scans. Figure 6 shows the results of the landmark placement for three faces in neutral expressions. Figure 7 shows the results of the landmark placement for three smiling faces.

## 5 Discussion

From the results, it's clear that the pronasal and endocanthions are placed more accurately than the cheilions. There may be multiple reasons for this discrepancy. First, the regression models for the cheilions were trained without including any information about the  $y$ -coordinates of the points of interest. The absence of  $y$ -coordinate information was motivated by the fact that the collapsed mouths of edentulous patients distort the  $y$ -coordinate information of points near their mouths, so training with  $y$ -coordinate information using a training set of people without collapsed mouths could lead to overfitting. Second, it was difficult to pinpoint the points of interest for the cheilions. In particular, in order to obtain all points in the region around the corners of the mouth, the criteria for determining points of interest had to be loosened, which caused points too close to the center of the face and points below the lower lip to be included as candidates for the cheilions. For an example of this phenomenon, see Figure 5.

A couple avenues of future research could help remedy the issue of cheilion inaccuracy. For instance, a more selective set of criteria for determining the cheilion points of interest could be developed. Alternatively, feature selection could be applied before the feature vectors for candidate points are fed into the regression model to improve the performance of the regression model.

Outside of correcting the inaccuracy of the cheilions, there exists a plethora of further research that could sprout from this work. With regards to the process used for orienting face scans into the standard position, the current process sometimes requires many iterations to successfully orient the face scan. It would be interesting to explore whether improved objective functions or a more adroit application of numerical optimization could help decrease the time spent orienting the face scan. Alternatively, it may be possible to completely rework the strategy for orienting the face scan. Perhaps the use of a template face mesh and an iterative closest point algorithm could provide an optimal solution.

Another facet of this project that could be examined further is the possibility of reusing a set of points of interest to predict the locations of multiple landmarks. Currently, each set of points of interest is used to predict the location of only one landmark. Accordingly, the detection of a large number of landmarks is prohibitively time-consuming because researchers need to specify a new set of point of interest criteria for each new landmark detected. If the work in [14] concerning the use of geometric descriptors to locate different regions of the face could be extended and combined with a refinement procedure capable of discerning multiple different landmarks from the same set of points of interest, the procedures outlined within this paper could be used to detect a much larger number of face landmarks without a commensurate increase in time and effort invested by researchers.

## References

- [1] O. Agbolade, A. Nazri, R. Yaakob, A. A. Ghani, and Y. Kqueen Cheah. Homologous multi-points warping: An algorithm for automatic 3d facial landmark. In *2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, pages 79–84, 2019.
- [2] A. Amirkhanov, M. Bernhard, A. Karimov, S. Stiller, A. Geier, M. E. Gröller, and G. Mistelbauer. Visual analytics in dental aesthetics. In *Computer Graphics Forum*, volume 39, pages 635–646. Wiley Online Library, 2020.
- [3] S. Cheng, S. Zafeiriou, A. Asthana, and M. Pantic. 3d facial geometric features for constrained local model. *2014 IEEE International Conference on Image Processing (ICIP)*, 2014.
- [4] M. A. de Jong, A. Wollstein, C. Ruff, D. Dunaway, P. Hysi, T. Spector, F. Liu, W. Niessen, M. J. Koudstaal, M. Kayser, E. B. Wolvius, and S. Börhringer. An automatic 3d facial landmarking algorithm using 2d gabor wavelets. *IEEE Transactions on Image Processing*, 25(2):580–588, 2016.
- [5] M. C. El Rai, C. Tortorici, H. Al-Muhairi, H. Al Safar, and N. Werghi. Landmarks detection on 3d face scans using local histogram descriptors. In *2016 18th Mediterranean Electrotechnical Conference (MELECON)*, pages 1–5, 2016.
- [6] M. Galvánek, K. Furmanová, I. Chalás, and J. Sochor. Automated facial landmark detection, comparison and visualization. In *Proceedings of the 31st Spring Conference on Computer Graphics, SCCG ’15*, page 7–14, New York, NY, USA, 2015. Association for Computing Machinery.
- [7] S. Z. Gilani, F. Shafait, and A. Mian. Shape-based automatic detection of a large number of 3d facial landmarks. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [8] J. Guo, X. Zhu, and Z. Lei. 3ddfa. <https://github.com/cleardusk/3DDFA>, 2018.
- [9] J. Guo, X. Zhu, Y. Yang, F. Yang, Z. Lei, and S. Z. Li. Towards fast, accurate and stable 3d dense face alignment. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.
- [10] S. Gupta, K. R. Castleman, M. K. Markey, and A. C. Bovik. Texas 3d face recognition database. In *2010 IEEE Southwest Symposium on Image Analysis & Interpretation (SSIAI)*, pages 97–100. IEEE, 2010.
- [11] S. Gupta, M. K. Markey, and A. C. Bovik. Anthropometric 3d face recognition. *International journal of computer vision*, 90(3):331–349, 2010.

- [12] J. J. Koenderink and A. J. van Doorn. Surface shape and curvature scales. *Image Vis. Comput.*, 10:557–564, 1992.
- [13] J. Liu, Q. Zhang, and C. Tang. Automatic landmark detection for high resolution non-rigid 3d faces based on geometric information. *2015 IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 2015.
- [14] F. Marcolin and E. Vezzetti. Novel descriptors for geometrical 3d face analysis. *Multimedia Tools and Applications*, 76(12):13805–13834, 2016.
- [15] T. Mutsvangwa, J. Smit, H. Hoyme, W. Kalberg, D. Viljoen, E. Meintjes, and T. Douglas. Design, construction, and testing of a stereophotogrammetric tool for the diagnosis of fetal alcohol syndrome in infants. *IEEE Transactions on Medical Imaging*, 28(9):1448–1458, 2009.
- [16] N. M. Patrikalakis and T. Maekawa. Shape interrogation for computer aided design and manufacturing. 2010.
- [17] R. R. Paulsen, K. A. Juhl, T. M. Haspang, T. F. Hansen, M. Ganz, and G. Einarsson. Multi-view consensus CNN for 3d facial landmark placement. *CoRR*, abs/1910.06007, 2019.
- [18] C. B. Sullivan and A. Kaszynski. PyVista: 3d plotting and mesh analysis through a streamlined interface for the visualization toolkit (VTK). *Journal of Open Source Software*, 4(37):1450, may 2019.
- [19] J. Sun, D. Huang, Y. Wang, and L. Chen. Expression robust 3d facial landmarking via progressive coarse-to-fine tuning. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 15(1):1–23, 2019.
- [20] T. Terada, Y.-W. Chen, and R. Kimura. 3d facial landmark detection using deep convolutional neural networks. *2018 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, 2018.
- [21] E. Vezzetti and F. Marcolin. Geometry-based 3d face morphology analysis: Soft-tissue landmark formalization. *Multimedia Tools and Applications*, 68(3):895–929, 2012.
- [22] E. Vezzetti and F. Marcolin. 3d landmarking in multiexpression face analysis: A preliminary study on eyebrows and mouth. *Aesthetic Plastic Surgery*, 38(4):796–811, 2014.
- [23] J. Zhang, K. Gao, Q. Zhao, and D. Wang. Pose invariant 3d facial landmark detection via pose normalization and deep regression. *2020 2nd International Conference on Image Processing and Machine Vision*, 2020.