Building a Statistical Shape Model from 3D Face Scans

1 Introduction

Working with a large data set of separate shapes of a specific object type might make it difficult to draw conclusions about the object type in general. Instead it is natural to work with a statistical interpretation of the data set and from this do calculations on variance, mean, etc. A statistical shape model is a popular such method. The base object is the mean shape which can be deformed to resemble the different shapes in the set. This can be used to investigate new shapes such as in object recognition, and to synthesize shapes similar to those in the training set.

2 Method

Building a statistical shape model is conceptually easy, and involves only a few steps. Preparing the shapes for alignment and decomposition is what requires the most effort.

2.1 Acquiring Data

The data used here was acquired using a laser scanner. 15 human faces were scanned, each consisting of roughly 20000 3D points.

2.2 Registration

The scanning process differs from scan to scan, which means that the raw scan data does not contain the same number of points, and the point ordering is different. To be able to create a shape model, all faces must have the same number of points and the same point ordering. Making sure this is fulfilled is called registration.

2.2.1 The Use of a Template Shape

The basic idea of the registration method used here is choosing one shape as a template, and altering the point ordering and data extent of the other shapes

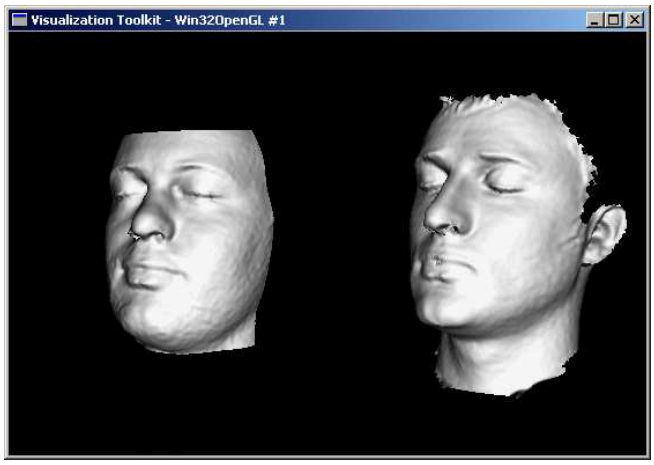

Figure 1: The template shape next to an unregistered face

according to this. The template face should be well represented and contain no statistical abnormalities. It should also be pruned, so that the corresponding extent of the template is present in all the other shapes. Figure 1 shows the pruned template next to an unregistered face.

2.2.2 Landmarking shapes

Each shape is then landmarked, i.e points of correspondence are manually marked out in each shape. Nine landmarks are used here, following [1]. These are (in this order and from the observers point of view): the chin, the left and right corner of the mouth, the tip of the nose, the left and right corner of the left eye, the nose curve minimum and the left and right corner of the right eye. Figure 2 shows a landmarked shape in the 3D landmarking software written by Rasmus Paulsen [7].

2.2.3 Registration

The registration process is then conducted for each shape as follows:

The template face is deformed [2] to roughly fit the shape to be registered using the thin-plate spline warping technique. This makes the shapes coincide at the landmarks as can be seen in figure 3. See appendix B for more information on thin-plate spline warping.

The template is now treated as a set of points, and the shape to be registered is treated as a continuous surface. For each point in the template, the closest point on the shape surface is found and registered. When this is done for all template points, the old shape points are discarded and replaced by the new registered points. This makes sure that the template and the new shape has the same number of points and the same point ordering. An effect of this is that the new shapes are also pruned according to the template.

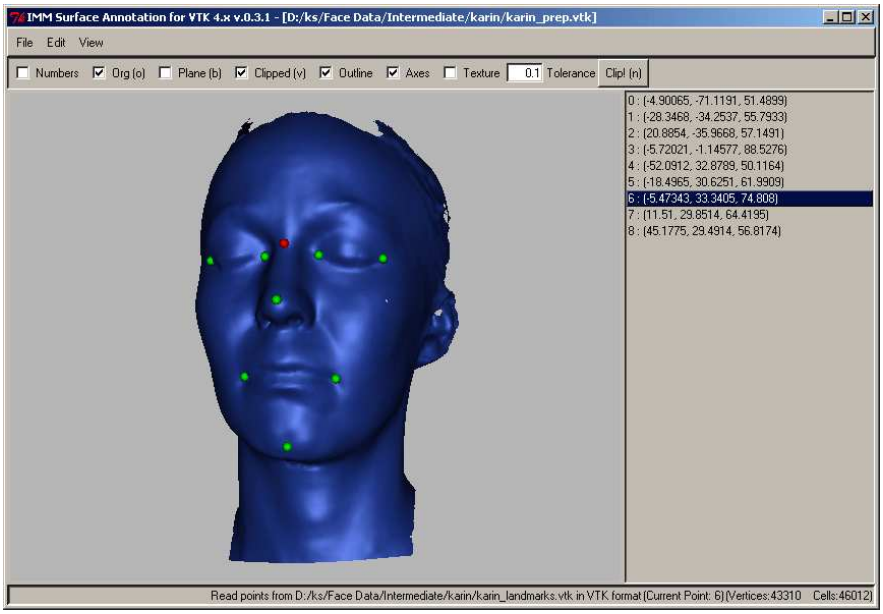

Figure 2: Face with landmarks.

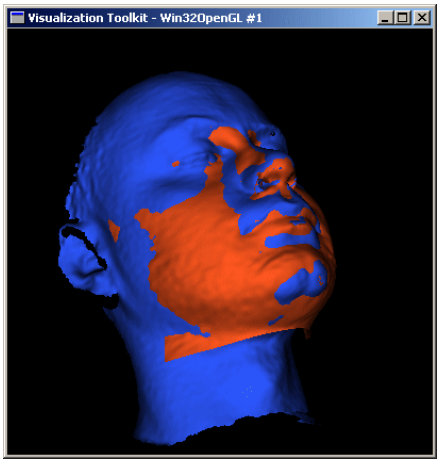

Figure 3: The warped template (red) over a face to be registered (blue)

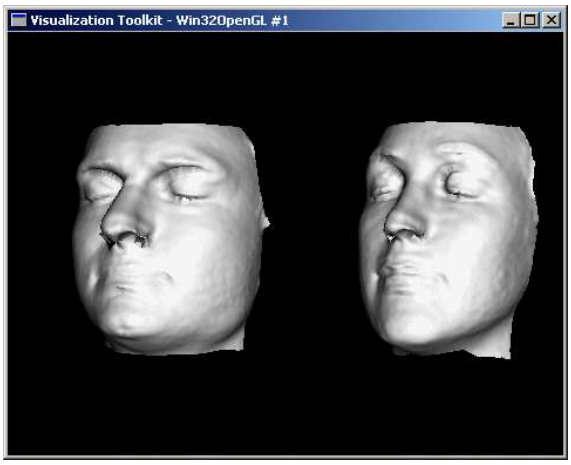

Figure 4: Two registered faces

2.3 Data Alignment

With the set of registered shapes the model can now be built. To remove any variation in between the shapes caused by different location, rotation and/or scale, i.e point variance not caused by difference in shape, a general procrustes analysis (GPA) is preformed [4]. This essentially aligns the shapes optimally using a similarity transform.

2.4 Change Of Basis and Reduction of Dimensionality

A 3D shape consisting of n points can be seen as a single vector in $\Re^{3n}$ where the vector can be constructed as

$$v = (x_0, \ldots, x_n, y_0, \ldots, x_n, z_0, \ldots, z_n)$$

The set of shapes now consists of a point cloud in $3n$-dimensional space. The current coordinate system basis makes it possible to move single points which is not very useful for statistical purposes. Through a change of basis more interesting axes can be found describing for example age, gender, etc. Here, the new basis is found through a principal component analysis (PCA), probably the most common method for shape decomposition. The new basis consists of the main axes of the point cloud. This makes it possible to represent the objects in the data set using only a few parameters. Appendix A gives an in-depth description of PCA.

3 Implementation

The implementation of the techniques described were done using the Visualization ToolKit (VTK) through Tcl/Tk and C++ [6].

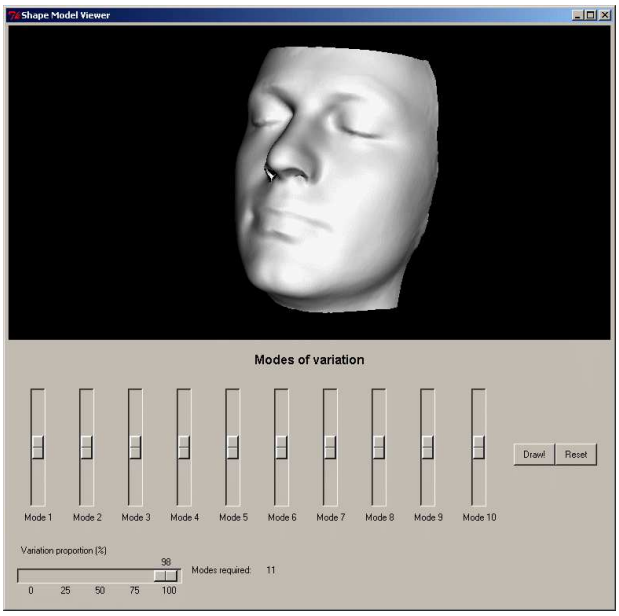

Figure 5: The shape model viewer program showing the mean face

3.1 Scanning Faces

The faces were scanned using a Minolta Vivid 900 laser scanner. One scan takes approximately 5 seconds, and three scans from different angles are necessary to get a decent representation of the face area. The scanner is not able to register hair, so a full head representation is not possible with this type of camera.

3.2 Quality of Data and Methods of Improvement

The three scans needed for each face produce three separate surfaces that need to be merged. This is done by the camera software. The result is decent but the surface becomes rough, and mesh holes are common. Therefore the shapes are post-processed to remove some of the holes and smoothed and relaxed in VTK using a windowed sinc function interpolation kernel.

The implementation of the registration algorithm is straightforward as VTK provides functions for thin-plate spline warping, finding the closest point on a plane etc.

To be able to examine the model easily, a simple Tcl/Tk program showing the model has been implemented. Figure 5 shows the program with the user interface. The face representation shown is the mean face.

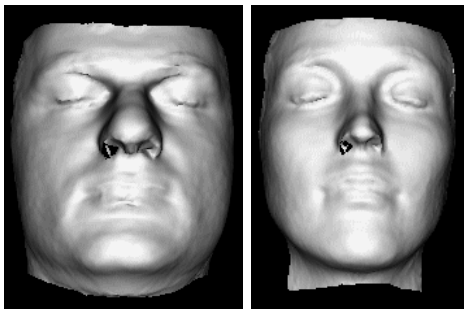

Figure 6: The first mode of variation

Figure 7: The second mode of variation

3.3 PCA Implementation

VTK also provides functionality for alignment and PCA which makes implementation trivial.

4 Interpreting the Model

Figure 6, 7 and 8 show the three major modes of variation. The interpretation of these is not clear but the first mode seems to model gender, while the two other model aspect ratio along the x and z axes.

5 Future Work

The Minolta camera used here captures both shape and texture. Currently work is done to extend the shape model to an appearance model [3]. This model will be used for object recognition of faces in varying pose.

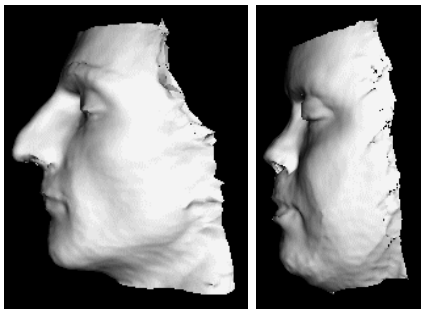

Figure 8: The third mode of variation

A Principal Components Analysis

PCA is concerned with transforming data in one coordinate system to another with the following properties:

- The first axis is chosen so that the variance of the projected data along the axis is maximised. The second axis is chosen so that the remaining variance is maximised along that axis, and so on for all axes.
- All axes must be orthogonal.
- Each new variable (axis) is a linear combination of the original variables (axes).
- The transformation must not contain scaling.

To illustrate these ideas, a simple example with two-dimensional geometrical data will be given. This can then be generalised to higher dimensions. Figure 9 shows a set of 2D points and the line $x_1 = x_2$. Table A lists the coordinates.

Observation	x_1	x_2
1	1.02	0.546
2	1.44	0.372
3	1.21	0.897
4	4.31	5.19
5	4.17	4.39
6	5.75	5.78
7	5.12	6.93
8	6.86	5.53
9	8.33	7.75
10	9.19	8.35
Variance	48.7 %	51.3 %

It is immediately seen that there is strong correlation between the points, they seem to roughly fulfill the equation $x_1 = x_2$. Let's assume that the greatest variation indeed can be found along this line and choose $x_1 = x_2$ as the first

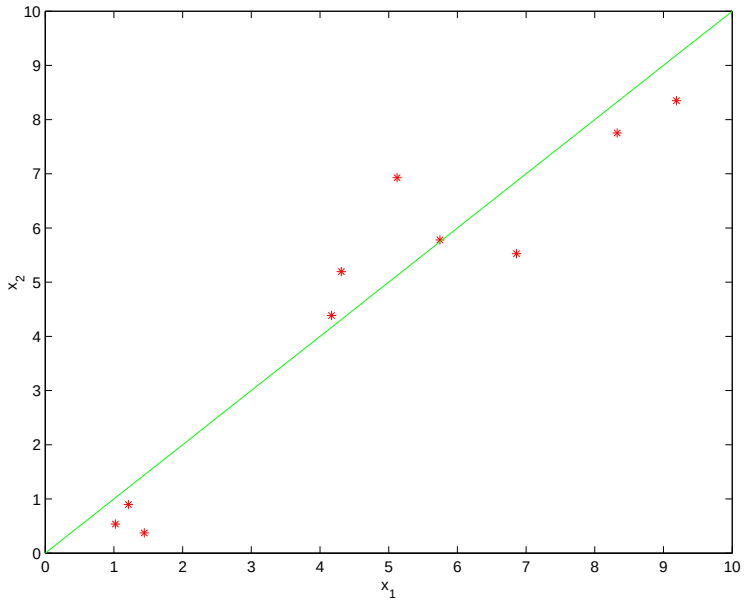

Figure 9: A set of 2D points

new axis. The second and last axis then becomes $x_1 = -x_2$ as this is the only choice for orthogonal axes. Projecting the points onto these new axes by

$$\tilde{x}_1 = cos\frac{\pi}{4}x_1 + sin\frac{\pi}{4}x_2 = \frac{1}{\sqrt{2}}x_1 + \frac{1}{\sqrt{2}}x_2$$

$$\tilde{x}_2 = -sin\frac{\pi}{4}x_1 + cos\frac{\pi}{4}x_2 = -\frac{1}{\sqrt{2}}x_1 + \frac{1}{\sqrt{2}}x_2$$

gives

Observation	$\tilde{x}_1$	$\tilde{x}_2$
1	1.09	-0.372
2	1.26	-0.790
3	1.48	-0.260
4	6.73	0.456
5	6.05	0.0015
6	8.15	-0.183
7	8.55	1.06
8	8.73	-1.16
9	11.4	-0.692
10	12.4	-0.906
Variance	97.4 %	2.6 %

Almost all variance is now along $\tilde{x}_1$, just as expected. This leads to the suspicion that there is an optimal choice of $\tilde{x}_1$, which maximises the variation. There is, and the choice is the eigenvector of the covariance matrix of $\mathbf{x}$ corresponding to the largest eigenvalue. The eigenvector corresponding to the second largest

eigenvalue gives the axis with the second most variation, and so on. Why is this? The proof given here follows the one given by S. Sharma [5].

Let $\mathbf{x}$ be a $p \times n$ matrix where p is the number of variables and n is the number of observations. In our case $p = 2$ and $n = 10$. The covariance matrix, $\mathbf{\Sigma_x}$, is given by

$$\mathbf{\Sigma_x} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x_i} - \bar{\mathbf{x}})(\mathbf{x_i} - \bar{\mathbf{x}})^T.$$

where $\bar{\mathbf{x}}$ is the $p \times 1$ vector describing the mean observation. The new decorrelated, or variance maximised, variables will be given by linear combinations of the old. Let therefore $\mathbf{C} = (\mathbf{c}_1\mathbf{c}_2 \dots \mathbf{c}_p)^T$ denote the $p \times p$ matrix in which each row $\mathbf{c}_i$ contains weights to form each new variable. In the example above

$$\mathbf{C} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}.$$

Let $\tilde{\mathbf{x}} = \mathbf{Cx}$ denote the new set of variables.

The variance of the new variable is given by

$$\mathbf{\Sigma_{\tilde{x}}} = \frac{1}{n}\sum_{i=1}^{n}(\tilde{\mathbf{x}}_i - \bar{\tilde{\mathbf{x}}})(\tilde{\mathbf{x}}_i - \bar{\tilde{\mathbf{x}}})^T =$$

$$\frac{1}{n}\sum_{i=1}^{n}(\mathbf{Cx}_i - \mathbf{C}\bar{\mathbf{x}})(\mathbf{Cx}_i - \mathbf{C}\bar{\mathbf{x}})^T = \ldots = \mathbf{C\Sigma_x C}^T.$$

This means that for $\mathbf{\Sigma_{\tilde{x}}}$ to be maximised, we must find a suitable $\mathbf{C}$ subject to the constraint $\mathbf{CC}^T = \mathbf{I}$ (i. e. no scaling). The solution can be found by optimising $\mathbf{C\Sigma_x C}^T$ using Lagrange multipliers.

Let

$$Z = \mathbf{C\Sigma_x C}^T - \lambda(\mathbf{CC}^T - \mathbf{I}).$$

The partial derivative is given by

$$\frac{\partial Z}{\partial \mathbf{C}} = 2\mathbf{\Sigma_x C}^T - 2\lambda\mathbf{C}^T.$$

Setting the above to zero yields the final solution. That is,

$$(\mathbf{\Sigma_x} - \lambda\mathbf{I})\mathbf{C}^T = \mathbf{0}$$

Rearranging we get,

$$\mathbf{\Sigma_x C}^T = \lambda\mathbf{C}^T.$$

This is recognised as the definition of eigenvectors and eigenvalues. That is, for the above to be true, the rows of $\mathbf{C}$ must hold the eigenvectors of $\mathbf{\Sigma_x}$ and the p possible choices of λ must be the corresponding eigenvalues.

In conclusion, for $\mathbf{\Sigma_{\tilde{x}}}$ to be maximised, the weight rows of $\mathbf{C}$ should be chosen as the eigenvectors of $\mathbf{\Sigma_x}$. In the example above we get eigenvalues 152.1 and 4.019 corresponding to eigenvectors $(0.6970, 0.7171)$ and $(-0.7171, 0.6970)$ respectively. The first principal component is therefore the line $x_1 = 1.029x_2 - 0.3045$, not far off the initial guess. This component accounts for 97.4% of the total variance. Therefore, if suitable, the second principal component could be omitted, thus reducing dimensionality; one of the main purposes of PCA.

B Thin-Plate Spline Warping

Imagine an image printed on an (infinitely) elastic rubber sheet. Picture piercing the sheet with a set of pins and dragging these to new positions, thus transforming the image. The image represents visualisation data and the pins represent landmarks. Dragging the pins to new positions translates to transforming the data to fit new landmark positions, such as mean landmarks. This type of transformation is called warping.

Warping m-dimensional data $\mathbf{x}$ with landmarks $\mathbf{x}_i$ to m-dimensional data $\mathbf{y}$ with landmarks $\mathbf{y}_i$ is preformed using a multivariate function $\mathbf{y} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ which preferably holds the following properties:

- continuous
- smooth
- bijective
- interpolating, i.e. $\mathbf{f}(\mathbf{x}_i) = \mathbf{y}_i \forall i$

The rubber sheet warping mentioned above can be achieved using the bivariate function $(x', y') = \mathbf{f}(x, y) = (f_1(x, y), f_2(x, y))$. Since the equations for x' and y' are independent, the rest of the discussion will focus on a single scalar valued thin-plate spline function.

Warping is essentially the same as interpolation. In interpolation, the task is to find suitable values in-between known data, while in warping, the task is to find suitable positions for data in-between known positions. In one dimension interpolation can be preformed using piecewise cubic polynomials called natural cubic splines. These lead to globally smooth functions, i.e. the second order derivatives are continuous throughout the spline. Physically, the cubic spline represents a thin metal rod which is somehow held in place at the points where data is known. The rod will take on a shape that minimises its internal bending energy. The extension of cubic splines to $n \geq 2$ dimensions are thin-plate splines where, as the name suggests, the steel rod has been replaced by a thin steel plate.

The bending energy function of the steel plate is

$$\iint_{\Re^2} (f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2) dxdy \tag{1}$$

not taking into account other physical factors such as gravity. The function $f(x, y)$ that minimises this bending energy is

$$f(x, y) = \sum_{j=1}^{n} w_j U(r) + a_0 + a_1 x + a_2 y \tag{2}$$

where

$$U(r) = r^2 \log r^2 \tag{3}$$

and

$$r = \sqrt{(x^2 + y^2)} \tag{4}$$

The coefficients w_j, a_0, a_1 and a_2 are unknown, but the constraints imposed by $\mathbf{f}(\mathbf{x}_i) = \mathbf{y}_i$ and the wish to minimise the total bending energy makes it possible to determine these by solving linear systems of equations.

For n landmarks, the demand for exact interpolation gives the following n equations:

$$x'_i = \sum_{j=1}^{n} w_j U_{ij} + a_0 + a_1 x_i + a_2 y_i \,, \;\; 1 \leq i \leq n \tag{5}$$

where $U_{ij} = U(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2})$. This can be expressed in matrix form as

$$\mathbf{x}' = E\mathbf{w} + X\mathbf{a} \tag{6}$$

where $E = [E_{ij}]$ $(n \times n)$ and $X = [1 \;\; x_i \;\; y_i]$ $(n \times 3)$. The requirement for minimised bending energy gives

$$X^T \mathbf{w} = \mathbf{0} \tag{7}$$

Solving equation 6 for $\mathbf{w}$ gives

$$\mathbf{w} = E^{-1}(\mathbf{x}' - X\mathbf{a}) \tag{8}$$

Inserting this into equation 7 and solving for $\mathbf{a}$ gives

$$\mathbf{a} = (X^T E^{-1} X)^{-1} X^T E^{-1} \mathbf{x}' \tag{9}$$

Equations 8 and 9 are the analytical expressions for all unknown parameters collected in $\mathbf{w}$ and $\mathbf{a}$. Together these form $n+3$ equations for the same number of unknowns.

References

[1] Tim J. Hutton, Bernard F. Buxton and Peter Hammond. *Dense Surface Point Distribution Models of the Human Face*

[2] Rasmus Paulsen, Rasmus Larsen, Claus Nielsen, Søren Laugesen and Bjarne Ersbøll. *Building and Testing a Statistical Shape Model of the Human Ear Canal*

[3] T.F. Cootes and C.J. Taylor. *Statistical Models of Appearance for Computer Vision*

[4] J. C. Gower. *Generalized Procrustes analysis. Psychometrika, 40:33-50, 1975.*

[5] S.C. Sharma. *Applied Multivariate Techniques*

[6] Kitware Inc. *www.vtk.org - Visualization Toolkit*

[7] Rasmus Paulsen. *ISA - 3D landmarking software. www.imm.dtu.dk/∼rrp*