Assignment: Shape Interpolation

CSE 4280/5280

February 3, 2020

1 Objective

In this assignment, you will implement a method for interpolating between 2-D shapes. You will write programs to solve the following problems:

1. Create the average shape of a given set of shapes (e.g., the average female face).
2. Create an animation showing the warping of a shape into another.
3. Transfer the triangulation (i.e., edges) from the average shape to another shape.

2 Overview of shape interpolation

Shape interpolation allows us to combine multiple shapes to create new shapes. It is also a key component of morphing algorithms [Wolberg, 1999]. Figure 1 shows an example of interpolating between two 3-D shapes. Interpolating shapes is the high-dimensional analogue of interpolating numbers. When interpolating between two numbers, we calculate new intermediate values that follow an *interpolation function*, which can be linear (e.g., linear interpolation) or non-linear (e.g., cubic interpolation). The interpolation of high-dimensional data such as 2-D shapes is analogous to the interpolation of scalar values.

Next, we discuss a simple interpolation method that uses *convex combination*. It assumes that, if we "walk" a straight-line path between two points of a convex set, we will not "leave" the set (i.e., convex set) [Boyd and Vandenberghe, 2004]. This interpolation method works equally for numbers and for high-dimensional objects such as shapes.

Figure 1: Interpolating 3D shapes (Figure from [Eisenberger et al., 2019])

2.1 Interpolation of one-dimensional data

We can interpolate between two real numbers, s_1 and s_2, by using the following equation:

$$s_j = (1 - \alpha)s_1 + \alpha s_2, \tag{1}$$

for $\alpha \in [0, 1]$. This equation is a simple weighted average and is called a *convex combination*. It is one of many ways to perform interpolation. For example, we can use Equation 1, to generate intermediate numbers between $s_1 = 21$ and $s_2 = 31$, i.e.:

$$\begin{aligned}
\alpha = 0.0 &\implies s_1 = 1.0 \times 21 + 0.0 \times 31 = 21, \\
\alpha = 0.2 &\implies s_2 = 0.8 \times 21 + 0.2 \times 31 = 23, \\
\alpha = 0.5 &\implies s_3 = 0.5 \times 21 + 0.5 \times 31 = 26, \\
\alpha = 0.8 &\implies s_4 = 0.2 \times 21 + 0.8 \times 31 = 29, \\
\alpha = 1.0 &\implies s_5 = 0.0 \times 21 + 1.0 \times 31 = 31. \qquad (2)
\end{aligned}$$

In the above sequence, each value of α yields a number that combines the two input numbers, 21 and 31. As α varies from 0 to 1, the “influence” of s_1 decreases while that of s_2 increases. We can think of this "influence" as a type of “similarity”, e.g., for $alpha = 0.2$, the new number is 80% “similar” (or close) to s_1 and 20% “similar” to s_2.

2.2 Interpolation of multi-dimensional data

When solving problems in science and engineering, we often work in multiple dimensions. Multi-dimensional quantities and objects can represent many real-word "things" (e.g., geometrical shapes, feelings, motions, sounds), and are naturally related to vectors. In this assignment, we will use vectors to represent 2-D shapes, and think of them as points living in a multi-dimensional space. If we assume that this (shape) space is convex then we can interpolate between shapes by using a vector form (i.e., multi-dimensional) of the interpolation method in Equation 1, which is given by:

$$\mathbf{s}_j = (1-\alpha)\mathbf{s}_1 + \alpha\mathbf{s}_2. \tag{3}$$

This equation combines vectors $\mathbf{s}_1$ and $\mathbf{s}_2$, to create a new vector, $\mathbf{s}_j$, which is a mixture of proportions of $\mathbf{s}_1$ and $\mathbf{s}_2$. Figure 2 illustrates the interpolation of two triangle shapes represented as vectors. Here, a triangle is a point (or vector) in a 6-D space (i.e., 3 vertices, each represented by two coordinates).

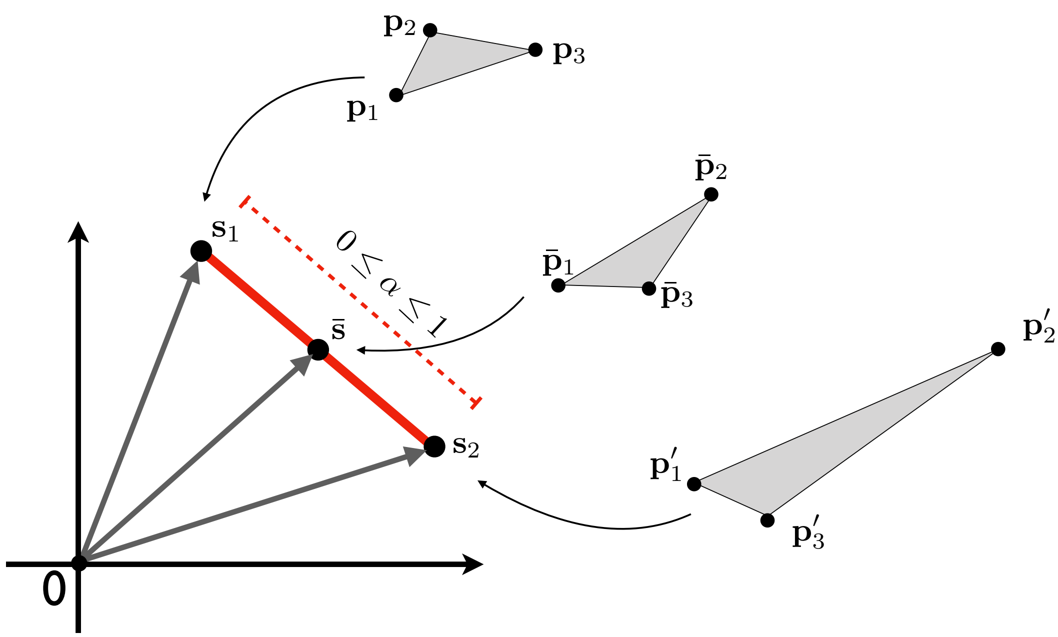

Figure 2: Interpolating between triangles using Equation 1. Triangles $\mathbf{s}_1$ and $\mathbf{s}_2$ are combined to create an intermediate triangle $\bar{\mathbf{s}}$. Here, the new triangle, $\bar{\mathbf{s}}$, is also the average of the two input triangles (i.e., $\alpha = 0.5$). By using Equation 1 and varying the value of α in the interval $[0, 1]$, we can create multiple interpolated shapes.

The intuition that we used for triangles applies to general shapes of many vertices. Consider for instance a set of shapes of faces, described by a set of ordered 2-D landmarks $\mathcal{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_N\}$, where $\mathbf{s}_j = \{(x_1, y_1)^\mathsf{T}, \ldots, (x_M, y_M)^\mathsf{T}\}$ is the j^{th} face.

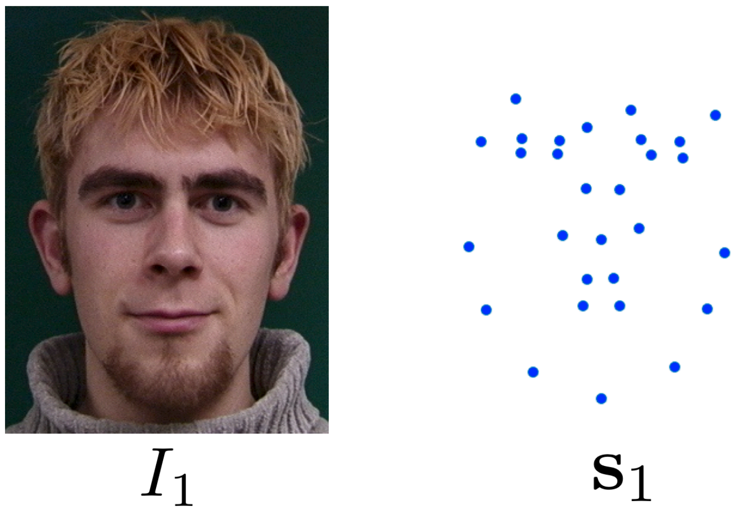

Figure 3: A image of a face and its corresponding shape. The shape consists of a set of spatial landmarks (x_i, y_i).

Given two different faces, $\mathbf{s}_1 \in \mathbb{R}^{2\times M}$ and $\mathbf{s}_2 \in \mathbb{R}^{2\times M}$ (Figure 4.left), we can use Equation 3 to create interpolated face shapes that show a "warping sequence" between $\mathbf{s}_1$ and $\mathbf{s}_2$.

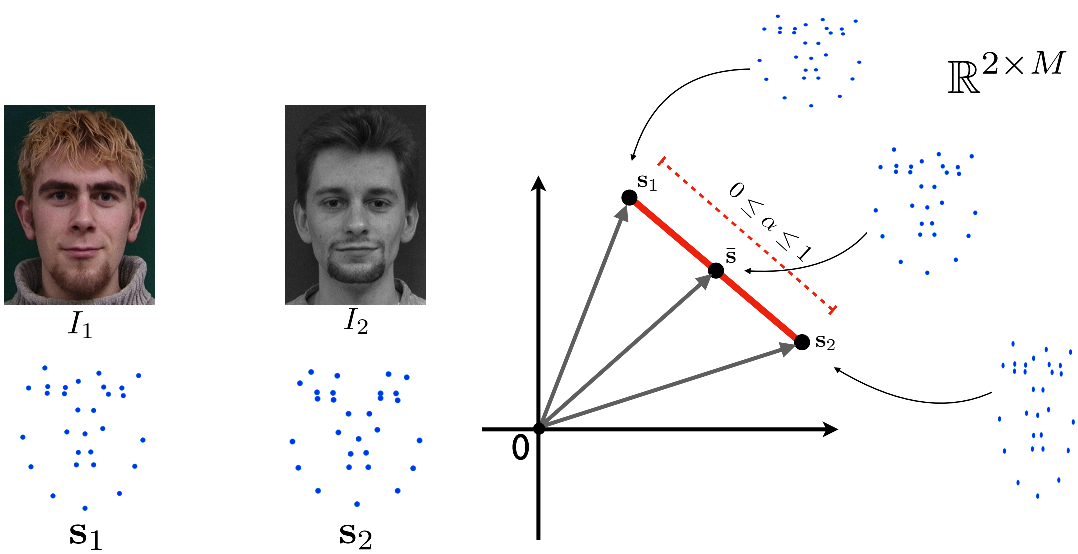

Figure 4: Left: Two different faces and their shapes. Right: Interpolation.

In addition to interpolating between two face shapes, we can create a new face shape that combines all faces in the dataset. We can combine all shapes by using a simple average calculation, which is in fact another form of the weighted average in Equation 3. Given the set of shapes $\mathcal{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_N\}$, the average shape is given by:

$$\bar{\mathbf{s}} = \frac{1}{N}\sum_{j=1}^{N}\mathbf{s}_j, \tag{4}$$

which is the set of M (averaged) landmarks, i.e., $\bar{\mathbf{s}} = \left\{(\bar{x}_1, \bar{y}_1)^\mathsf{T}, \ldots, (\bar{x}_M, \bar{y}_M)^\mathsf{T}\right\}$. In this case, all shapes contribute equally to the resulting shape (i.e., $\alpha = 1/N$).

3 Using triangulation to visualize shapes

We can visualize a shape by plotting a triangulation of its vertices. Here, the vertices are connected into triangles by means of a triangulation method such as the *Delaunay Triangulation* (Figure 5). Computationally, this triangulation is represented by a graph $\mathcal{T} = (V, E)$ consisting of a set of vertices V (i.e., landmarks) and a set of edges E (i.e., connections).

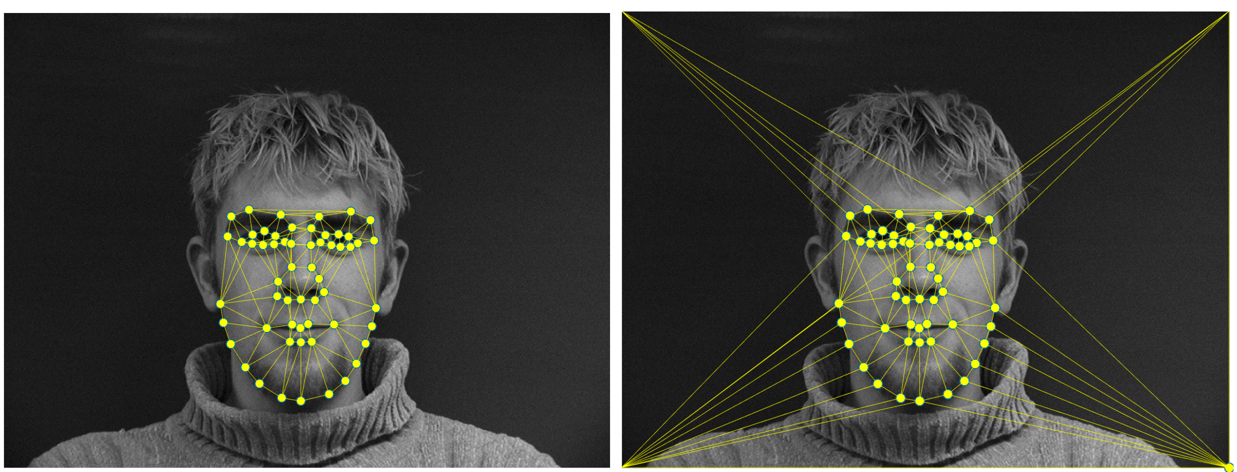

Figure 5: Two triangulations of the face landmarks. Left: Just the landmarks. Right: Including the image corners.

We can "transfer" a triangulation from one shape to the triangulation of another shape. The transfer of triangulation is as follows. Consider a shape $\mathbf{s}_j \in \mathcal{S}$ with triangulation $\mathcal{T}_j = (V_j, E_j)$, and another shape $\mathbf{s}_k \in \mathcal{S}$ with triangulation $\mathcal{T}_k = (V_k, E_k)$. To transfer the triangulation from $\mathbf{s}_k$ to $\mathbf{s}_j$, we keep the original vertices of $\mathcal{T}_j$, and replace its edges with the edges from $\mathcal{T}_k$. Thus, the new triangulation of shape $\mathbf{s}_j$ is:

$$\mathcal{T}_j' = (V_j, E_k)\,, \tag{5}$$

where V_j are the original vertices of shape $\mathbf{s}_j$ and E_k are the edges (i.e., connectivity) of the triangulation of $\mathbf{s}_k$. Equation 5 means that only the connections are transferred from $\mathcal{T}_k$ to $\mathcal{T}_j$ while the vertices (i.e., nodes) remain the same.

4 Implementation details

4.1 Datasets

To complete this assignment you will need a dataset shapes with annotated landmarks. The following are some annotated datasets that you can use to complete this assignment.

- MIT's Style Transfer for Headshot Portraits dataset (`https://people.csail.mit.edu/yichangshih/portrait_web/`).
- Stegmann's The IMM Face Database (240 images) for statistical models of shape (`http://www2.imm.dtu.dk/~aam/datasets/datasets.html`).
- FEI Face Database (`https://fei.edu.br/~cet/facedatabase.html`).

Each dataset comes with a *readme* or instruction file that describes the dataset's format and content. You need the landmarks (i.e., xy-coordinates) for each face. While you will not need the actual face images to complete the assignment, they can be helpful for visualizing the triangulation results superimposed on the images.

4.2 The average face shape

To create the average face shape, select a subset of shapes from the dataset (e.g., males, females) and store the coordinates of the landmarks in arrays. For example, you can represent all female faces by first storing the x and y coordinates of the landmarks into two separate sub matrices:

$$X_{\text{female}} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1M} \\ x_{21} & x_{22} & \dots & x_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{NM} \end{bmatrix}, \quad Y_{\text{female}} = \begin{bmatrix} y_{11} & y_{12} & \dots & y_{1M} \\ y_{21} & y_{22} & \dots & y_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ y_{N1} & y_{N2} & \dots & y_{NM} \end{bmatrix}. \tag{6}$$

Then, concatenate the sub matrices to form a single matrix of female shapes as follows:

$$S_{\text{female}} = \left[\frac{X_{\text{female}}}{Y_{\text{female}}} \right] = \left[\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_N \right]. \tag{7}$$

By using sub-matrices and concatenation, your programming instructions will mostly follow the mathematical notation. Finally, use Equation 4 (or an equivalent function, e.g., `mean`) to calculate $\bar{\mathbf{s}}_{\text{female}}$. To visualize the result, calculate the Delaunay triangulation of the resulting average shape, and display it as shown in Figure 6.

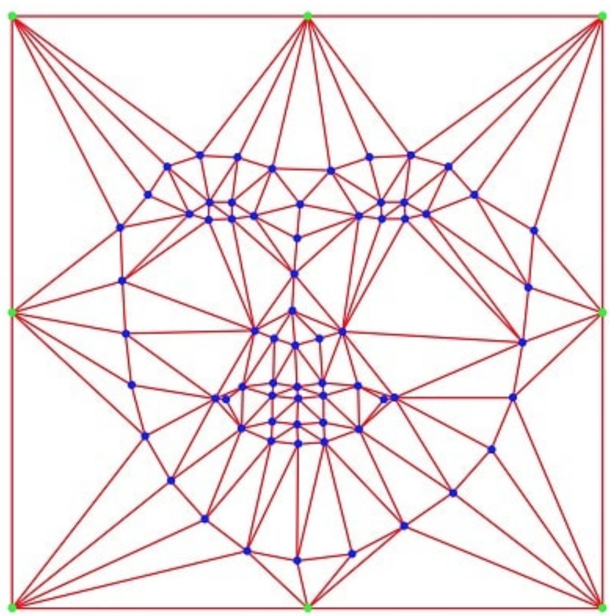

Figure 6: Triangulation example. To include the image corners in the triangulation, calculate the locations of the corners and middle points along the border lines, and then add these locations to the set of landmarks to be triangulated by the Delaunay-triangulation function.

4.3 Shape warping animation

Select two different faces from the dataset, e.g., $\mathbf{s}_1$ and $\mathbf{s}_2$. Then, vary the value of α within the interval $[0, 1]$. For each α, evaluate Equation 3 to create a new shape. Triangulate the new shape save the triangulation (e.g., gif image). Once all shapes are created, create an animation (i.e., video, animated gif) with the sequence of triangulations.

4.4 Transfer the average triangulation

Triangulations are graph data structures comprising two components: vertices and edges. To transfer a triangulation, we take an existing triangulation and replace its edges (e.g., connections) with the edges from another triangulation. The vertices of the triangulation must remain intact. To transfer the average triangulation, first calculate the triangulation of the average face shape, $\bar{\mathcal{T}} = \left(\bar{V}, \bar{E}\right)$. Then, choose a triangulation of any shape, $\mathcal{T}_j = (V_j, E_j)$. The new triangulation that uses the edges from the average face will be:

$$\mathcal{T}_j' = \left(V_j, \bar{E}\right), \tag{8}$$

where V_j are the vertices of $\mathcal{T}_j$ and $\bar{E}$ are the edges (i.e., connectivity) of the average face.

References

[Boyd and Vandenberghe, 2004] Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, USA.

[Eisenberger et al., 2019] Eisenberger, M., Lahner, Z., and Cremers, D. (2019). Divergence-free shape correspondence by deformation. In *Computer Graphics Forum*.

[Wolberg, 1999] Wolberg, G. (1999). Image morphing: A survey. *The Visual Computer*, 14.