3D Expression Retargeting

An approach through spatiotemporal model

Muhammad Ahmed Riaz(12100035), Tahir Cheema(12100092)

Aim and Background

Animation industry is one of the leading industries in the field of Computer Science. And one of the very hard tasks is to program and render the animation frame by frame. If only there is a way that without programming the whole scene, we could take some data from real life and project it on any character we want to. Hence, retargeting can be a key solution to this problem as it can perform this service in real time once we perform some initial manual setup.

Approach

A paper published by Ijaz Akhter (CV lab) on bilinear spatiotemporal model, provided the ground work for our approach. If we are somehow able to compute the shape basis of a face then we can represent any expression as a linear combination of these bases. All we need to do is to find the right coefficients of every basis. This service was provided by the concept of PCA bases which can be computed if a Mo-cap data is available for some expressions.

Various steps involved in the process are as follows

Computing PCA Basis

PCA basis can be computed by using the SVD technique, applied on covariance matrix of the mocap. Let's call the input data as S_{Fx3P}, where we have F frames, P points, and each point having 3 coordinates. Then the covariance matrix is given as $\frac{1}{N-1}(S-\mu)(S-\mu)'$. Applying SVD to the covariance matrix gives us PCA basis for the given shape. The eigen vector are the basis where as square root of the eigen values are the basis coefficients. The coefficients for each frame can be obtained by projecting the original data on the basis matrix.

$$C = Sd \times B; \qquad \text{_____ (1)}$$

$$S_{recon} = C \times B' + \mu \qquad \text{Where } Sd = S - \mu$$

These C are the coefficients for each basis and each frame. This method is used to check the validity of the basis formation.

Matching Correspondences

Given the mocap data, and a 3D character on which the data is to be retargeted, we need to find the point to point correspondences. These correspondences help in finding the shape basis of the character we are given.

PCA Basis Visualization

The PCA basis we obtain, if added to in the mean posture of the mocap, can form the expressions that they represent. This can be expressed by the eq(2) :

$$S^{expression} = \mu + 3\sigma B^{expression}$$

Where S is a specific expression

B is the corresponding basis which represents that expression.

σ is the square root of the eigen value of B. It is the standard deviation.

We use the term 3 σ assuming that most of the data lies within 3 standard deviations of a normal distribution.

Transformation between mocap and basis

Once we have extreme expressions obtained using (equation 2), we compute transformations for each triangulation between extreme expression and neutral pose. Each of the transformation is applied to respective set of points of 3D character on which retargeting is required. In this step, the correspondences mapped by the user are used to find corresponding points of character. Ideally if the transformations and correspondences are perfectly done and the mocap structure resembles the character shape, the character is also transformed into its extreme pose.

Computation of Character Basis

Once we have an extreme pose of the character, we can compute its basis using the same (equation 2) stated above. The σ for character is taken to be the same as that of mocap and basis turn out to be

$$B^{expression} = \frac{S^{expression} - \mu}{3\sigma}$$

Retargeting of a given sequence

To retarget any given sequence of mocap, we simply compute the projection of that sequence onto its own basis, using the (equation 1). These coeffiencts are multiplied with the character basis to obtain the retargeted animation. This only involves taking an inner product (matrix multiplication), which is quite cheap computationally. Hence with little processing, one can retarget the data, if we have set up the basis initially.

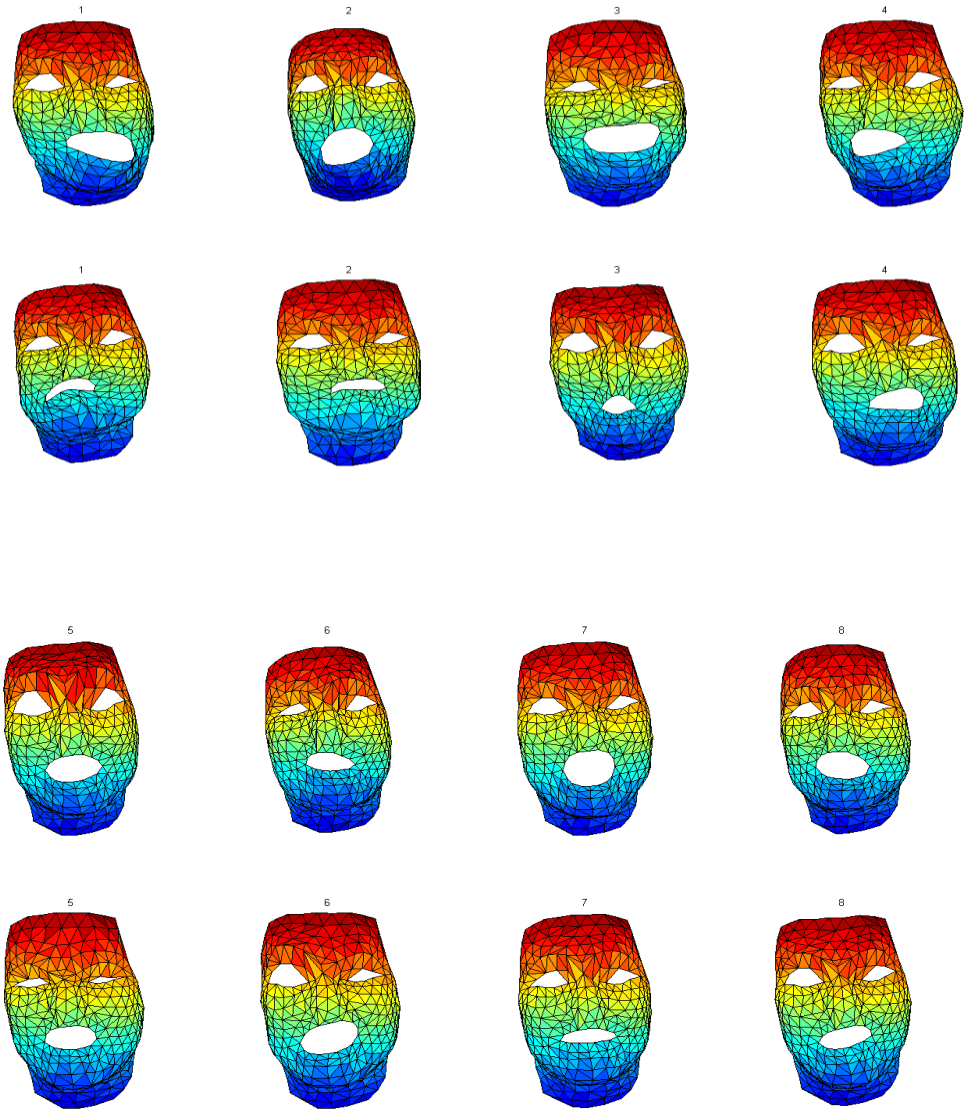
Implementation Details

Model cleanup

The model we got had a lot of processing and cleanup to do before it could be used for retargeting. It had a full round head and multilayers like hair, eyes, teeth, skin. Extra points were removed and a recursive script file was written *del_recur.m* which made the removal of eye balls and teeth relatively less tedious. After the removal of extra points, the TRIangulation matrix had to be adjusted appropriately. For the purpose the script file *cleanface.m* was written, which spots missing points and changes the order of points in TRI.

PCA Analysis

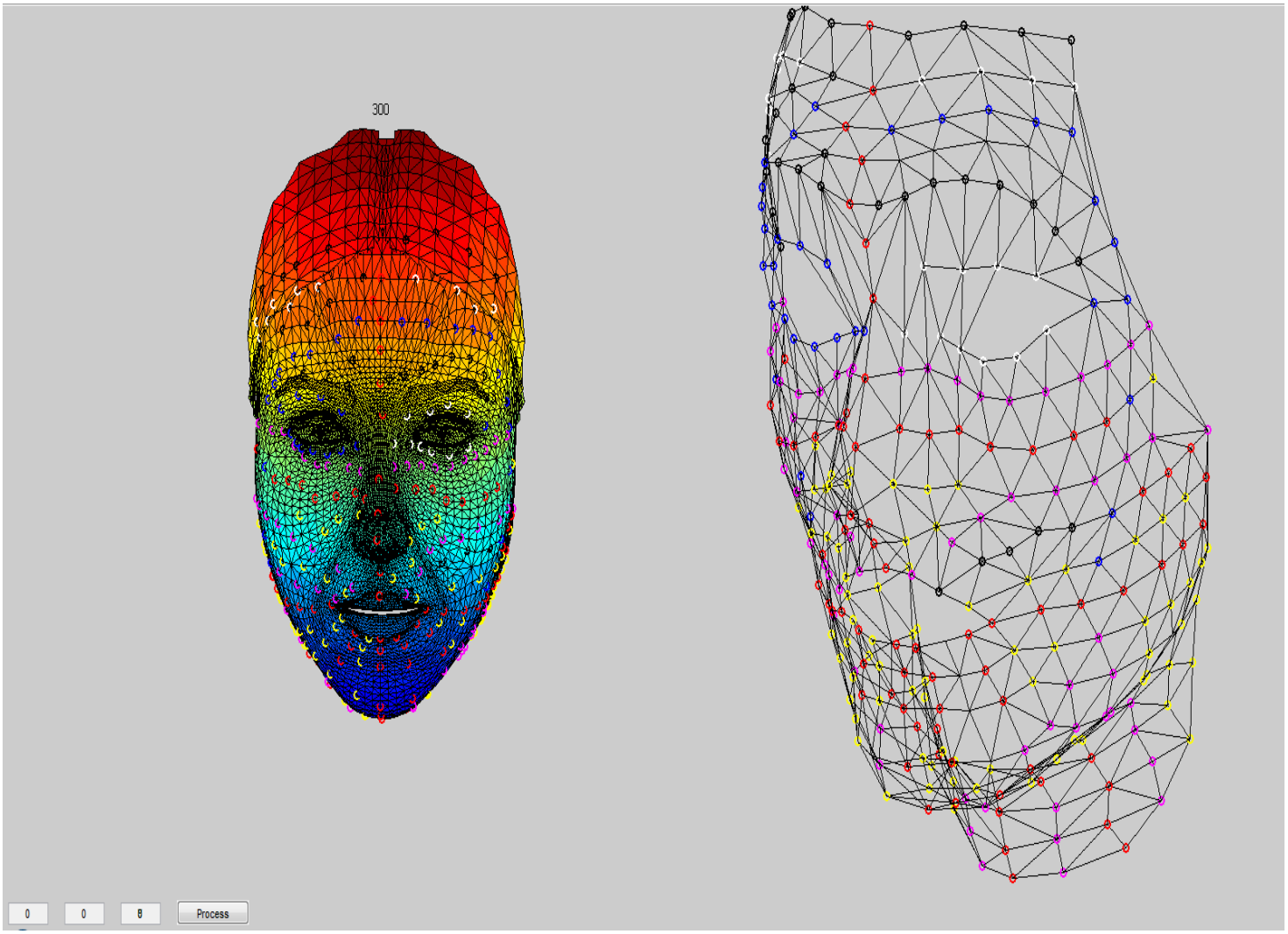
PCA analysis was done using the standard technique of SVD. The function used for the purpose in our case is `findpca()`.



Matching correspondences

For correspondence matching, we developed our own custom tool, which provides the following capabilities:

- Neutral pose selected from all the given frames of moacp using a slider control.
- Correspondence matching, between the two models, by simple mouse click.
- Custom marker color for each correspondence point for easy division of face into segments e.g. the line cutting the nose horizontally is colored green, whereas the horizontal line through eyes is blue. This helps a lot in reducing confusion during excessive matching.
- Changing of view easily using keyboard shortcuts i.e. 'Arrow Keys' are used for rotation of camera and '+ -'keys for zooming in and out.
- Undo feature for both models separately, using the shortcut keys 'z' and 'x'.
- Changing active figure for camera control using the shortcut keys 1,2.



Transformation between neutral and extreme mocap pose

The following steps are followed to find the required transformation:

- Each triangle(set of 3 points) is selected from neutral pose, lets call them *tri*
- To compute 3D affine, we need more points, we search over the neighbors of these points to get 3 extra points.
- From obtained 6 points, mean of *tri* is subtracted to shift the surface to origin. It is rotated to align it with Z plane, its first point is shifted exactly on origin and it's rotated about Z axis to align its 2nd point with x axis. Thus the transform results in a standard plane, aligned with Z axis ans having two of its points on x axis. This helps in a standard transform. The same set of transformation is applied on the extreme pose corresponding triangle, to obtain the shifted points in the standard coordinates (same as those of neutral pose).
- A transformation T between the two pairs of points is found by applying pseudo inverse on the equation

$$P^{extreme}_{3x5} = T_{3x4} \, x P^{neutral}_{4x5}$$

- For the 3D character, using the user defined correspondences, corresponding three points are obtained.
- As the character is high density, all the points that lie inside the given three points are also found out using method described in the next section.
- All these obtained points are also shifted to the coordinated system, where these are aligned with Z axis, have their first points on origin and one edge of the triangle on x axis.
- Once in the proper system, the same transform T is applied, and the result if hoped to the points of extreme pose of character.

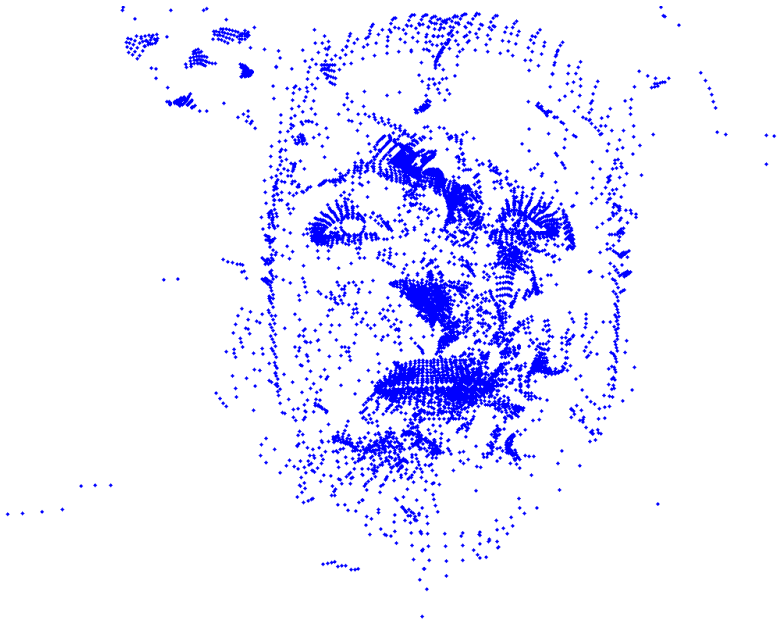
After visiting all the triangles of the character, the shape obtained is the extreme pose of character and from which we should get the corresponding basis. If there are minor problems in the shape, an editing tool name *Editor.m* can be used to edit individual points. Details of the editor are provided in coming section. The same process is repeated for each mocap basis to get corresponding basis.

Problem encountered

On application of transformation T on the character data, the result we get has a lot of noise and garbage values. Thus the process of character basis extraction could not be completed successfully. The issue lied somewhere in the application of T. We tried to investigate its cause, changed our approach a number of times, but still could not improve the result in given timeframe. Our approach evolved over time as follows:

- Found transformation between neutral and extreme pose directly
- Converted them to zero mean and found transform
- Aligned the plane with Z plane
- Shifted one of the points to origin and aligned its neighbor with x axis

But no method bore better results than the previous one.



Editor tool

The editor tool has the following features:

- Moving a point in the plane perpendicular to optical axis
- Moving a point parallel to normal vector of the surface
- The change in a point propagated to other parts of the face by a normal distribution
- Provision for entering custom standard deviation for the Gaussian applied

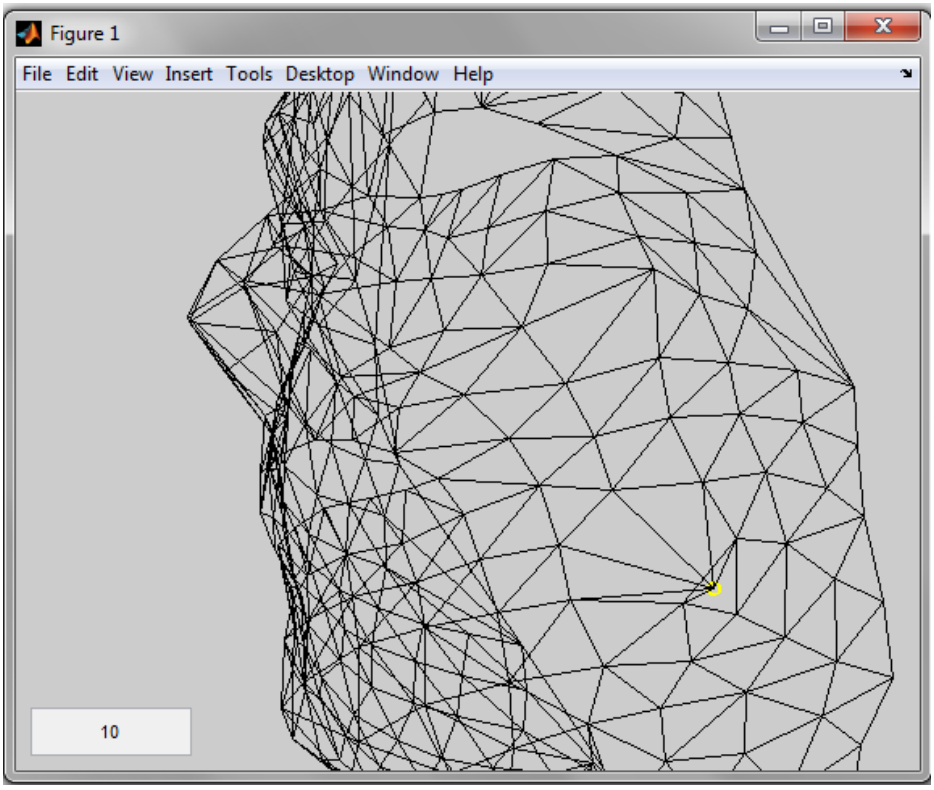
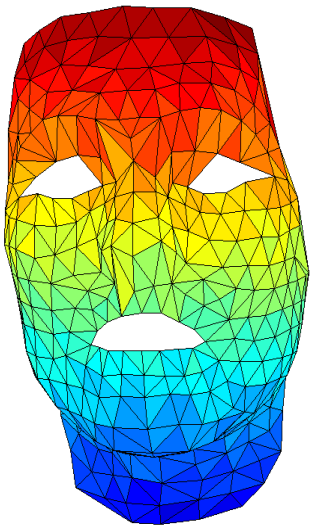
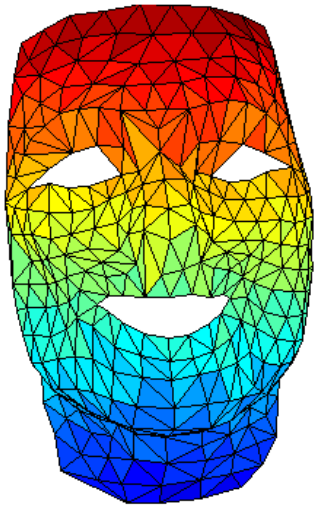


Figure 1
File Edit View Insert Tools Desktop Window Help
10

Editor result



Original structure



After editing

Finding high density points between selected 3 points

As from mocap we obtain low density data. The three points of a triangle we get a far apart and we have to select all the points that lie inside the triangle they form, the following algorithm is used:

- A plane is fitted in the three points
- All the points of the face are projected onto this plane
- A small sphere is fitted around the triangle
- All the points whose projected lies inside the triangle and which themselves lie inside the sphere fitted, are called INSIDE the given points