Generating a 3D Facial Model From a Single Image using Principal Components Analysis

Héctor Cuevas, DGSCA-UNAM, Isaac Rudomín, ITESM

***Abstract*--This paper describes a method for modeling a human face from a single image by using a data set consisting of facial images and 3D-models. This allows us to apply principal components analysis (PCA) to create a set of related 2D and 3D eigenfaces. We then apply PCA to the image and use the coefficients thus obtained to generate the 3D model as a linear combination of the pertinent 3D eigenfaces.**

***Index terms*—eigenfaces, PCA, IBM**

I. Introduction

The goal of this paper is describing a method that allows the automatic construction of facial 3D models from single images. Other researches have attempted to obtain facial models from images, by using different methods. To derive a three-dimensional model of a human face, it is possible to use either a single image or one or more images, either taken simultaneously from different points of view (stereo) or over a period of time (video). Single-image techniques rely mostly on shading information, while multiple-image techniques tend to use corresponding points and the parameters of the camera projection to calculate depth information. However, most systems using a single image never use explicit 3D information to assist the modeling.

A notable exception is described by Blanz and Vetter in [3]. Their system works by deforming an average 3D facial model to approximate the desired face, guiding this process by comparing the resulting rendered images –in fact, both model and texture are synthesized at the same time. While this system certainly works and its results are of high quality, it requires a lot of computing power and data. Perhaps too much data: all human faces are roughly alike - two eyes, one nose, one mouth, all in relatively the same position. What we propose in this paper is to apply principal component analysis to a data set of faces to reduce the number of faces we need to know in order to synthesize a face. By being able to deconstruct a 2D face, we can reconstruct that same face, this time in 3D space. The following sections shall explain how this is done.

II. Method

The eigenface technique was proposed by Pentland in 1991 for face coding and recognition tasks and has been widely used; some applications are described in [1,2]. While the choice of the term eigenface may be debatable, we shall use it in this work. Put simply, an eigenface is one of the principal components in a given face space. The name comes from its being the result of solving the eigenvalue problem from a matrix built by a data set of facial images.

A. *Principal Components Analysis*

Principal Components Analysis is a method for expressing the information contained in our data set in a more suitable way from the statistical point of view. It is completely mathematical in nature, and does not depend on a model of the world –it just restates our data by defining a new axis system. Each new axis is defined by having the minimum sum of squared distances to the points in our set, while being perpendicular to any previously defined axes. An equivalent condition is maximizing the explained variation in our data set by each axis. An algebraic definition follows:

Suppose that we have values for N observations of M variables each. For convenience we shall assume our variables have zero mean. Our first principal component is a variable U_1 satisfying the following conditions:

$$U_1 = w_{1,1} X_1 + w_{1,2} X_2 + \cdots + w_{1,m} X_m$$

where X_i represents the *i*-th variable, and the $w_{1,j}$ are a set of unknown constants whose sum captures the total variance to be explained. In case of normalized variables, this implies

$$w_{1,1}{}^2 + w_{1,2}{}^2 + \cdots + w_{1,m}{}^2 = 1$$

Héctor Cuevas Vázquez del Mercado works in the Visualization Department at DGSCA-UNAM. His research interests are as follows: Computer Vision, Visualization and Virtual Reality.

For all possible set of $w_{i,j}$, evaluate U_1 for all observations, and find the variance of the obtained values. The one that maximizes this variance is our first principal component. This is equivalent to the geometric condition that the sum of squared distances be minimum.

The second principal component, U_2, is defined similarly from a set of $w_{2,j}$, adding the condition that U_2 is perpendicular to U_1:

$$w_{1,1}w_{2,1} + w_{1,2}w_{2,2} + \cdots + w_{1,m}w_{2,m} = 0$$

We follow this procedure to find up to M principal components (if one of our variables is a linear combination of the others we shall find one less principal component). One decision affects the principal components obtained: whether all variables are considered equally important or some are more important than others. We shall assume some are intrinsically more important than others.

The first component explains more information than the following components. This is true for any component and those following it. For data reduction, we can drop one or more of the less explaining components, effectively reducing the dimension of our data by 1. Dropping one of our original variables wasn´t advisable, because the information loss would probably be too high, but being a principal component, we know this loss is minimum and measurable, so we can decide if our application can afford it.

B. 2D Eigenfaces

An Eigenface is just a principal component derived from a set of face images. These images and their derived eigenfaces both define what we call facial space. The more representative our face base, the closer to true facial space. Facial space is contained in image space, which is inherently bigger.
Image space is too big and arbitrary for faces –background is either redundant, irrelevant or distracting, and faces are more or less equal: same number of features of roughly the same shape located in the same relative position, and very symmetric. The canonical base for image space simply neglects this:

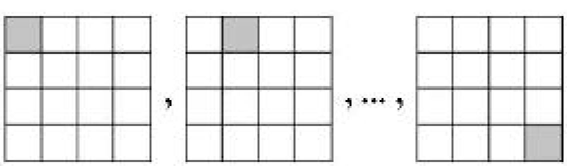


Figure 1 Canonical Base

If we take images of *m* rows by *n* columns, the dimension of image space is $m*n$. If we take a subset of K faces, we can provide a basis of K elements for it –each face is a trivial linear combination of it. This trivial basis is useless for faces outside our face base, but points to the possibility of reducing the size of our problem. The K eigenfaces provided by PCA not only generate our face set, but are capable of generating other faces, provided our face set was representative enough. One thing that should be noted is that our eigenfaces are images of $m*n$ pixels. Each face on face space is generated by a linear combination of our eigenfaces. The coefficients of this combination are called the face´s *encoding*.

Let *E* be the matrix whose *K* columns are our principal components, or eigenfaces. Let *x* be a face on image space and *y* be that same face on facial space. So to encode a face, we use the following transformation

$$y = E^T * x$$

where *y* is a vector of dimension *K*, while *x* is a vector of dimension $m*n$. *E* is a matrix of *K* columns and $m*n$ rows.
Since facial space is contained in image space, this is a many to one transformation. We can attempt to reconstruct *x* by doing

$$x' = P * y$$

where x´ = x plus some error. This error increases as we drop principal components. It also increases if our original faces are heterogeneous: differences in size, position, rotation and background variance greatly affect PCA.

As can be seen, PCA is a kind of Auto Associative Memory.
Following is the partial deconstruction of a face image from outside the training set under the first four eigenfaces; the others aren´t shown here.

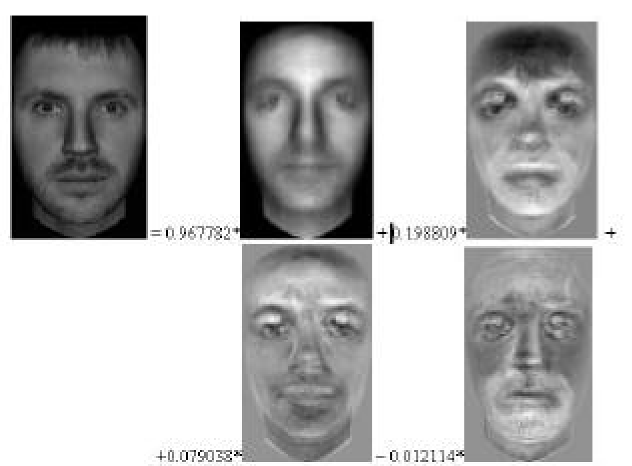


Figure 2 Reconstruction using 2D Eigenfaces

C. Obtaining 3D Faces

By using PCA, we can encode a new face as the coefficients of its linear deconstruction by the eigenfaces, provided the face base is representative enough, and from this encoding we can reconstruct the face as a linear combination of the eigenfaces weighted by the coefficients. Let us concentrate on the second part of this process –to reconstruct a face given its encoding. We can go as far as to generate a more or less arbitrary encoding to see what kind of face we obtain!
Although this is mostly a game, there is an interesting idea behind this: to create a face using an encoding not obtained from a previous deconstruction. Going a little bit further, we can ask ourselves if it is possible to create a 3D face using 3D eigenfaces and some encoding. It certainly is, if we have proper 3D eigenfaces.

Neglecting the fact that so far we don´t have any 3D eigenfaces, the next question is: If we apply an encoding derived from a 2D face to a set of 3D eigenfaces, do we obtain the model of that person´s face? Not necessarily, as we can easily convince ourselves. Even if we train our system with faces taken from the same people, there is no way to guarantee coefficients are interchangeable. We face the problem of correspondence between subsets (not even subspaces in the mathematical sense) of different dimension spaces. Worse, PCA creates its own model of the world –the researcher can do little more than choose an "adequate" base and hope PCA gives the "natural/right" answer.

Not wanting to make this more complicated than it needs to be, we take a different approach -if we want interchangeable coefficients, why not build our 3D eigenfaces to comply?
If we have corresponding faces in 2D and 3D and a set of 2D eigenfaces, we can easily construct 3D eigenfaces by assuming interchangeable encoding, just by solving a determined linear system.

D. Constructing 3D Eigenfaces

Our face base consists of an equal number of 2D and 3D faces, each pair belonging to the same person. We initialize our system by first obtaining 2D eigenfaces from the 2D faces in the normal way. Then we deconstruct each 2D face to obtain their encoding. By construction, the following holds:

$$\text{2D face} = \text{2D encoding} * \text{2D eigenfaces}$$

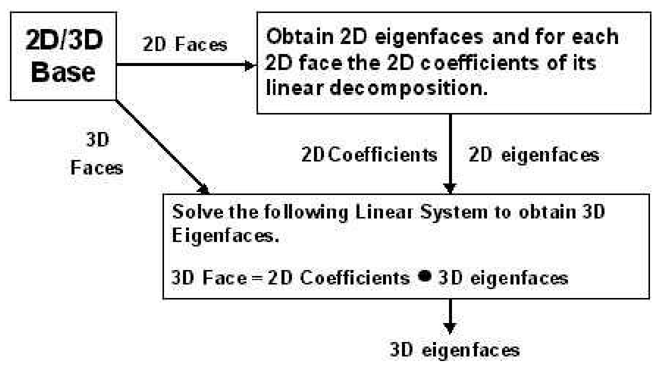
As we said earlier, a 2D face is a vector of $m*n$ dimension, its encoding has K elements, and each of our K eigenfaces is an image (somewhat resembling a face) of $m*n$ pixels. Compare this to our goal:

$$\text{3D face} = \text{2D encoding} * \text{3D eigenfaces}$$

We already know the corresponding 3D faces: by solving this equation for the eigenfaces we provide the interchangeable encoding property. Since the number of 2D faces and 3D faces is the same, and no data reduction was made (our encoding has as many coefficients as faces of each dimension in our base) this is just a determined linear system of equations which has a theoretical exact and unique solution, one that can be easily approximated numerically.
Clearly our system has the interchangeable encoding property, at least when going from 2D to 3D in our training set. If our training set is representative of 2D facial space (perhaps restricted to certain ethnic or age group), we shall have no problems in deconstructing new 2D faces within this group. The representativeness of the 3D training set and the behavior of the 3D eigenfaces on novel encodings isn´t so clear, so we shall test it in the following section.

The following diagrams illustrate the initialization and use of our system:

Figure 3 Obtaining 3D eigenfaces



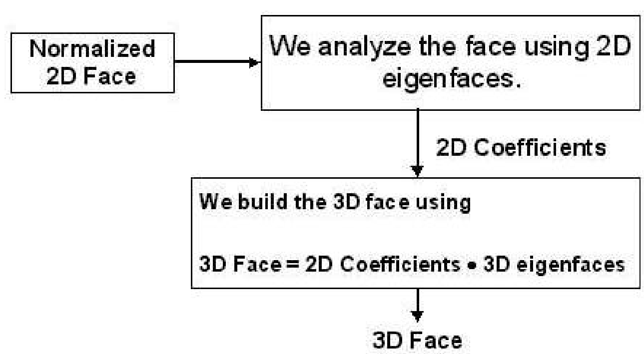


Figure 4. Reconstructing a 3D Face

III. TESTS

For testing our system, we shall use 14 meshes and their corresponding images, normalized in terms of scale and position.[1]
The people in this base does not represent an specific sex, ethnic or age group, so it will be a hard test. Each 3D face consists of 360 vertices forming 660 triangular faces while our 2D faces are 240x400 grayscale images.

Some sample scaled images of our face base follow:

Figure 5 Four faces in the 3D Face Base

A. 3D Model Generation Test

To evaluate the resulting 3D mesh, we measured the Euclidean Distance between corresponding points in the generated and “true” mesh. Each face is contained within a cube of approximately 520x849x510 units.
The errors are very small and don´t affect the overall result: the maximum was 3.029146 units, the minimum 0.000015 and the average distance between corresponding points was 1.169825. Since the values of our coordinates go as high as 800 units, the results are very good, since the test face comes from outside the training set.
The face on the left is the original, the one on the right is the generated by our system.

Figure 6 3D Reconstruction Results

IV. Conclusions and Future Work

Results are encouraging, but our work is far from over. Currently we are processing a 2D/3D face base to be suitable for use in our system . This face base consists of about 100 hundred people, and it will allow us to make a better demonstration of the technique, to explore automatic texture placement and more elaborate lighting conditions, which so far are assumed to be quite simple, as well as to make detection on a given image, instead of having the user manually editing it. 2D eigenface systems are resistant to small changes in rotation and lighting, and we expect these aspects to apply to our system as well. Another estimate we want to verify is the number of required faces in the face base: 2D systems can perform well with just a few hundreds. Other analysis are possible, like parts-based approaches, or the use of eigenfeatures, that is, the local application of PCA. A more natural parameterization based on physical characteristics of the face instead of unnatural principal components would be desirable. A possibility yet to be explored is the use of depth maps instead of meshes, to facilitate the use of existing models which otherwise would need semiautomatic processing to be used in our system, and the effects of dimensionality reduction in modeling.

V. Acknoledgments

Héctor Cuevas wishes to thank José Luis Villarreal from the Visualization Department at DGSCA-UNAM for his comments and suggestions.

VI. References

[1] A. Pentland, B. Moghaddam and T. Starner. View-Based and Modular Eigenspaces for Face Recognition. M.I.T. Media Laboratory Perceptual Computing Section Technical Report No. 245. Also appeared in *IEEE Conference on Computer Vision & Pattern Recognition*, 1994.

[2] B. Moghaddam, A. Pentland. An Automatic System for Model Based Coding of Faces. M.I.T. Media Laboratory Perceptual Computing Section Technical Report No. 317. Also appeared in *IEEE Data Compression Conference, 1995*.

[3] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D Faces. In *SIGGRAPH ’99 Conference Proceedings*, ACM, 1999

[4] S. Romdhani. Ph.D. Thesis: Face Recognition Using Principal Component Analysis.

[5] B. Bund Jackson. *Multivariate Data Analysis*. Richard D. Irwin, Inc. 1983

[1] The models in the face base were obtained from JOE (Join Our Experience) at http://www.cselt.it/ufv/joe/joe.html