Expert Systems with Applications 39 (2012) 4965–4971

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Dimensionality reduction for computer facial animation

Flora S. Tsai [*]

School of Electrical & Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore

ARTICLE INFO

Keywords:
Facial animation
Dimensionality reduction
Principal Components Analysis
Expectation–Maximization algorithm for PCA
Multidimensional Scaling
Locally Linear Embedding

ABSTRACT

This paper describes the usage of dimensionality reduction techniques for computer facial animation. Techniques such as Principal Components Analysis (PCA), Expectation–Maximization (EM) algorithm for PCA, Multidimensional Scaling (MDS), and Locally Linear Embedding (LLE) are compared for the purpose of facial animation of different emotions. The experimental results on our facial animation data demonstrate the usefulness of dimensionality reduction techniques for both space and time reduction. In particular, the EMPCA algorithm performed especially well in our dataset, with negligible error of only 1–2%.

1. Introduction

Facial animation is one alternative for enabling natural human computer interaction. Computer facial animation has applications in many fields, such as realistic virtual humans with different facial expressions used in the entertainment industry. In communication applications, interactive talking faces make the interaction between users and machines better, and also provide a friendly interface which help to attract users. Humanlike expression is critical among the issues concerning the realism of synthesized facial animation. Computer facial animation still remains a very challenging topic for the computer graphics community because of the deformation of a moving face is complex and humans have an inherent sensitivity to the subtleties of facial motion. Furthermore, depiction of human emotion is an extremely difficult interdisciplinary research topic studied by researchers in computer graphics, artificial intelligence, communication, psychology and others. In this paper, a realistic and expressive computer facial animation system is presented by automated learning from Vicon Nexus facial motion capture data. Ultimately, the emotions data are mapped to a 3D animated face using Autodesk Motionbuilder.

Vicon works by tracking infrared reflective markers using MX cameras that emit infrared light. A component called the MX Ultranet processes each camera image in hardware essentially. After that, multiple camera views are computed into a list of 3D marker coordinates in the host computer. The data used for this paper are the facial markers motion data collected using Vicon Nexus. Dimensionality reduction techniques are used to process the data collected. The emotions data are then mapped to a 3D animated face using Autodesk Motionbuilder. Because our approach used data captured from a real speaker, more natural and lifelike facial animations can be created.

Computer facial animation is important for many applications, such as video games, virtual reporters and other interactive human–computer interfaces. Computer facial animation also covers applications of different technologies including image processing, text-to-speech synthesis, and graphic visualization, and programming skills. This paper will focus on developing 3D animated facial animation using machine learning techniques.

This paper is organized as follows. Section 2 reviews related work on facial animation and dimensionality reduction techniques of Principal Components Analysis (PCA), Expectation–Maximization (EM) algorithm for PCA, Multidimensional Scaling (MDS), and Locally Linear Embedding (LLE). Section 3 describes the design, methodology, and algorithms for the recording, modeling, and animation stages. Section 4 presents the results of the experiments. Finally, Section 5 concludes the paper.

2. Literature review

In this paper, facial animation techniques and dimensionality reduction techniques were studied. Dimensionality reduction techniques are implemented to process the facial motion data collected.

2.1. Facial animation

In this section, related facial animation work are reviewed. In a previous study (Deng et al., 2006), the facial animation technique was composed of four stages which were recording, modeling, synthesis and animation. In the recording stage, expressive facial motion and its accompanying audio are recorded simultaneously and preprocessed. Vicon Nexus is used for the facial markers motion capturing. In the modeling stage, an approach is presented to

* Tel.: +65 6790 6369; fax: +65 6793 3318.
E-mail address: fst1@columbia.edu

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.10.018

learn speech coarticulation models from facial motion capture data, and a Phoneme-Independent Expression Eigenspace (PIEES) is constructed. In the synthesis stage, based on the learned speech coarticulation models and the PIEES from the modeling stage, the corresponding expressive facial animation is synthesized according to the given input speech/texts and expression. There are two subsystems in the synthesis system which are neutral speech motion synthesis and dynamic expression synthesis. The speech motion synthesis subsystem learns explicit but compact speech coarticulation models from recorded facial motion capture data, based on a weight-decomposition method. Given a new phoneme sequence, this system synthesizes corresponding neutral visual speech motion by concatenating the learned coarticulation models. In the dynamic expression synthesis subsystem, PIEES is first constructed by a phoneme-based time warping and subtraction, then novel dynamic expression sequences are generated from the constructed PIEES by texture-synthesis approaches. Finally, the synthesized expression signals are weight-blended with the synthesized neutral speech motion to generate expressive facial animation. The compact size of the learned speech coarticulation models and the PIEES make it possible for the system to be used for on-the-fly facial animation synthesis. Finally, in the animation stage, the captured motion markers are mapped to the 3D face model (Cao, Tien, Faloutsos, & Pighin, 2005).

2.2. Principal Components Analysis

Principal Components Analysis (PCA) is a useful statistical and widely-used technique for finding patterns in data of high dimensions. It is useful in reducing dimensionality and finding new, more informative, uncorrelated features (Tsai, 2010). There are some mathematical concepts that are used in PCA which covers standard deviation, covariance, eigenvectors and eigenvalues. PCA is a way of identifying patterns in data and highlight their similarities and differences. While the luxury of graphical representation is not available, patterns can be hard to find in data of high dimensions. Therefore, PCA is a powerful tool for analyzing data of high dimensions. The other main advantage of PCA is that once we have found these patterns in the data, then we could compress the data, reducing the number of dimensions without much loss of information. There are six steps to perform PCA on a set of data which are to get data, subtract the mean, calculate the covariance matrix, calculate the eigenvectors and eigenvalues of the covariance matrix and then choose components for forming a feature vector (Smith, 2002).

2.3. EM algorithms for PCA

A few eigenvectors and eigenvalues are allowed to be extracted from the large amount of high dimensional data with Expectation–Maximization (EM) algorithm for Principal Component Analysis. Missing information are accommodated naturally, resulting in high computational efficiency in both space and time. Principal Component Analysis can be viewed as a limiting case of a particular class of linear-Gaussian models (Roweis, 1998). The covariance structure of an observed p-dimensional variable can be captured using fewer than the $p(p+1)/2$ free parameters which are required in a full covariance matrix. Linear-Gaussian models assume that it was produced as a linear transformation of some k-dimensional latent variable x plus additive Gaussian noise (Roweis, 1998). Denoting the transformation by the $p \times k$ matrix C, and the p-dimensional noise by v (with covariance matrix R), the generative model can be expressed as

$$y = Cx + v \quad x \sim N(0, I) \quad v \sim N(0, R) \tag{1}$$

According to a unit variance spherical Gaussian, the latent or cause variables x are assumed to be independent and identically distributed. V is also independent and normal distributed. It is assumed to be independent of x. The model can be reduced to a single Gaussian model which can be expressed as:

$$y \sim N\left(0, CC^T + R\right) \tag{2}$$

It is necessary to have the condition $k < p$ to restrict the covariance structure of the Gaussian noise v by constraining the matrix R and to save parameters over the direct covariance representation in p-space as well.

2.3.1. Inference and learning

When working with the linear-Gaussian models, there are two problems that arise. The first problem is related to state inference or compression. Given fixed model parameters C, R, and also some observation y, we are interested in the posterior probability $P(x|y)$ over a single hidden state given the corresponding single observation. Linear matrix projection is used for computation and the resulting density is itself Gaussian:

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)} = \frac{N(Cx, R)|_y N(0, I)|_x}{N(0, CC^T + R)|_y} \tag{3}$$

$$P(x|y) = N(\beta y, I - \beta C)|x, \beta = C^T(CC^T + R)^{-1} \tag{4}$$

Not only is the expected value βy of the unknown state obtained but also an estimate of the uncertainty in this value in the form of the covariance $I - \beta C$. Computing y from x (reconstruction) is also straightforward: $P(y|x) = N(Cx, R)|_y$. Finally, the likelihood of any data point y is computed using Eq. (1).

2.3.2. Zero noise limit

Principal Component Analysis is a limiting case of the linear-Gaussian model because of covariance of the noise v becomes infinitesimally small and equal in all directions. PCA is obtained by taking the limit $R = \lim_{\epsilon \to 0} \varepsilon I$ mathematically. This makes the likelihood of a point y dominated solely by the squared distance between it and its reconstruction Cx. The directions of the columns of C which minimize this error are known as the principal components. Inference now reduces to simple least squares projection:

$$P(x|y) = N(\beta y, I - \beta C)|x, \beta = \lim C^T(CC^T + \varepsilon I)^{-1} \tag{5}$$

$$P(x|y) = N((C^TC)^{-1}C^Ty, 0)|_x = \delta(x - (C^TC)^{-1}C^Ty) \tag{6}$$

The posterior over states collapses to a single point and the covariance becomes zero since the noise has become infinitesimal.

2.3.3. EM algorithm

Even though the principal components can be computed explicitly, there is still an EM algorithm for learning them. It can be easily derived as the zero noise limit of the standard algorithms by replacing the usual e-step with the projection. The algorithm is:

$$\textit{e-step}: \quad X = (C^TC)^{-1}C^TY \tag{7}$$

$$\textit{m-step}: \quad C^{new} = YX^T(XX^T)^{-1} \tag{8}$$

where Y is a $p \times n$ matrix of all the observed data and X is a $k \times n$ matrix of the unknown states. The columns of C will span the space of the first k principal components. The algorithm can be performed online using only a single data point at a time and so its storage requirements are only $O(kp) + O(k^2)$. The intuition behind the algorithm is as follows: guess an orientation for the principal subspace. Fix the estimated subspace and project the data y into it to give the values of the hidden states x. Finally, fix the values of the hidden states and choose the subspace orientation which minimizes the squared reconstruction errors of the data points.