VISEME SPACE FOR REALISTIC SPEECH ANIMATION

Sumedha Kshirsagar, Nadia Magnenat-Thalmann

MIRALab – CUI, University of Geneva
{sumedha, thalmann}@miralab.unige.ch
http://www.miralab.unige.ch

ABSTRACT

For realistic speech animation, smooth viseme and expression transitions, blending and co-articulation have been so far studied and experimented widely. In this paper, we describe an approach for speech animation by smooth viseme transition. Though this method cannot form an alternative to the co-articulation phenomenon, it certainly takes us a step nearer to realistic speech animation. The approach is devised as a result of the Principal Component Analysis of facial capture data extracted using an optical tracking system. The system extracts the 3D positions of markers attached at the specific feature point locations on face to capture the facial movements of a talking person. We form a vector space representation by using the Principal Component Analysis of this data. We call this space the "viseme space". We use the *viseme space* to generate convincing speech animation and to make smooth transitions from one viseme to another. As the analysis and the resulting *viseme space* automatically consider the dynamics of and the deformation constraints on the facial movements, the resulting facial animation is very realistic.

1. INTRODUCTION

The goal of facial animation systems has always been towards obtaining a high degree of realism using optimum resolution facial mesh models and effective deformation techniques. Various muscle based facial models with appropriate parameterized animation systems have been effectively developed for facial animation [1,2,3]. The Facial Action Coding System [4] defines high-level parameters for facial animation, on which several other systems are based. Most facial animation systems typically follow the following steps

- Define an animation structure on a facial model by parameterization.
- Define "building blocks" or basic units of the animation in terms of these parameters, e.g. static expressions and *visemes* (visual counterparts of phonemes).
- Use these building blocks as key-frames and define various interpolation and blending functions on the parameters to generate words and sentences from visemes and emotions from expressions. The interpolation and blending functions contribute to the realism for a desired animation effect.
- Generate the mesh animation from the interpolated or blended key-frames.

Given the tools of parameterized face modeling and deformation, the most challenging task in facial animation is the design of realistic facial expressions and visemes. In order to produce highly realistic facial animation, advanced techniques such as opto-electronic capture and laser scanners have been used. Contrary to the previously described key-frame approach, in such methods the movements of the facial feature points are captured for every frame. The animation parameters derived from this captured data are retargeted to the facial model to obtain animation. It is not always practical to apply such motion capture data for a "performance driven facial animation", because it is often restricted by the availability of the performer and complexity of the equipment involved, which often needs tedious calibration and set-up. However, the output of such motion capture session can be used to design the above-mentioned "building blocks" of the facial animation thus ensuring an adequate degree of realism at the basic unit level.

The complexity of the key-frame based facial animation system increases when we incorporate the natural effects such as co-articulation for speech animation and blending between a variety of facial expressions during speech. The use of speech synthesis systems and the subsequent application of co-articulation to the available temporized phoneme information is a widely accepted approach [5,6]. Co-articulation is a phenomenon observed during fluent speech, in which facial movements corresponding to one phonetic or visemic segment are influenced by those corresponding to the neighboring segments. Two main approaches taken for co-articulation are by Pelachaud [7] and Cohen and Massaro [8]. Both these approaches have been based on the classification of phoneme groups and their observed interaction during speech pronunciation. Pelachaud arranged the phoneme groups according to the deformability and context dependence in order to decide the influence of the visemes on each other. Muscle contraction and relaxation times were also considered and the Facial Action Units were controlled accordingly. Cohen and Massaro defined non-linear dominance functions for the facial control parameters for each of the visemes and then used a weighted sum to calculate the control parameter trajectories for co-articulated speech animation.

Though these approaches result into satisfactory animations, we aim to show that statistical analysis of the actual facial movement data would help in improving the realism and robustness of the speech animation techniques. We consider the Principal Component Analysis (PCA) as a powerful tool for achieving this goal. Use of PCA on the facial motion capture data for facial animation has been previously tested. Kuratate *et al* [9] used PCA and Linear Estimator algorithm to drive the facial animation from opto-electronically captured facial movement data. Their main goal was focused on extracting the complete facial mesh animation from the movement of the sample tracking points on the face. They used the complete mesh vertex data obtained from laser range scanner as an input to PCA, analysis input vectors consisting of 8 static face configurations. Arsal *et al* [10] described an algorithm to extract the face point trajectories starting from a phonetically labeled speech signal. They tracked the subject's face using a multi-camera triangulation system. The principal components extracted from the tracked data were used as a compressed representation of this captured data along with the Line Spectral Frequencies (LSF) of the speech signal in a codebook. For each speech frame, a codebook search was performed to find the matching speech parameter values, and the corresponding PCs for the matched codebook entry were then subsequently used to get the face point trajectories.

Unlike these two approaches for facial animation, our motivation behind using PCA is to study the dynamics of the facial feature points during fluent speech, in addition to reducing the dimensionality of the data. We capture optical tracking data of a real person speaking a number of sentences from a database of phoneme rich sentences. The data acquisition for this purpose has been explained in Section 3 of this paper. The Principal Component Analysis of this captured data enables us to build a *viseme space*. In Section 4, we give a brief introduction to the PCA technique and explain how we use it on our captured data. As will be seen, the *viseme space* not only reduces the dimensionality of the data but also offers more insight into the dynamics of the facial movements during normal conversational speech. More importantly, it allows real-life like transitions between various components of speech animation (visemes), an important component of co-articulation. This is elaborated in Section 5. We begin by a brief description of the MPEG-4 facial animation standard we are using.

2. MPEG-4 FACIAL ANIMATION

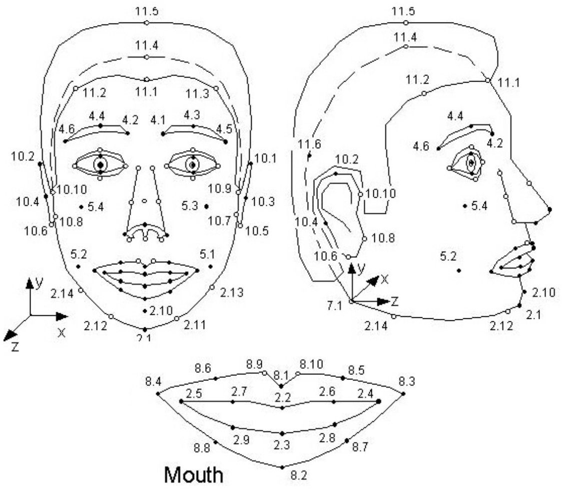

Figure 1: MPEG-4 feature points related to speech pronunciation

In this section we briefly explain the use of MPEG-4 facial animation standard [11]. The Facial Definition Parameter (FDP) set and the Facial Animation Parameter (FAP) set are designed to encode facial shape, as well as animation of faces thus reproducing expressions, emotions and speech pronunciation. The FDPs are defined by the locations of the feature points and are used to customize a given face model to a particular face. They contain 3D feature points such as mouth corners and contours, eye corners, eyebrow centers, etc. FAPs are based on the study of minimal facial actions and are closely related to muscle actions. Each FAP value is simply the displacement of a particular feature point from its neutral position expressed in terms of the Facial Animation Parameter Units (FAPU). The FAPUs correspond to fractions of distances between key facial features (e.g. the distance between the eyes). Thus, once we have the displacements of the feature points from the neutral position, it is very easy to extract the

FAPs corresponding to the given facial animation. Figure 1 shows the locations of the feature points as defined by the MPEG-4 standard. As these parameters are normalized, they specify facial animation, irrespective of the shape and size of the model. We use the MPEG-4 compatible facial animation system as described in [12].

3. DATA ACQUISITION

A commercially available optical tracking system (VICON 8) is used to capture the facial movements [13]. We use the selected MPEG-4 feature points for facial tracking as shown in the Figure 2. For the capture, we used 6 cameras and 27 markers of 3 millimeter diameter corresponding to the MPEG-4 feature point locations. We obtain the 3D trajectories for each of the marker points as the output of the tracking system. As we focus on speech animation, out of the 27 markers shown in the Figure 2, only 8 markers along the outer lip border, 2 markers on the chin, and 4 markers on the cheeks are used for the statistical analysis. These are the most important ones in order to capture the face movements during normal speech. The movements of the eyes and eyebrows can be controlled independently to the lip region movements in the facial model, and thus are not considered for analysis. The speaker is made to speak fluently 100 randomly selected sentences from the TIMIT database [14] of phoneme rich sentences. The head movement of the speaker is not restricted, and thus we need to compensate for the global head movements in order to obtain the local deformations of the markers. For tracking global head movements, 3 additional markers on a headband are used.

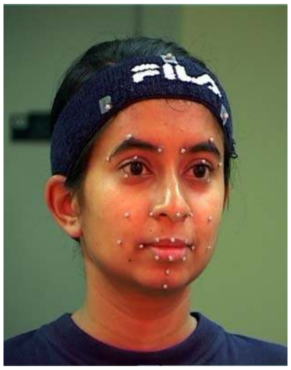

Figure 2: Placement of markers for selected feature points

We use the improved translation invariant method to extract rigid movements of the head from the tracked data [15]. Because the 3D points obtained from the motion capture system have sufficient accuracy, a linear algorithm is sufficient for this application instead of iterative algorithms based on the least square procedure. Once the global head movements are extracted, the motion trajectories of all the selected feature point markers are compensated for the global movements and the absolute local displacements for each are calculated. The recordings are also made for static mouth shapes for individual phonemes, which later serve as the basic building blocks of the speech animation. For each of the phonemes, a vector of 3D positions (compensated for global head movements) of the selected 14 markers is extracted. Their use in generating speech animation is explained in the subsequent sections.

4. DATA ANALYSIS

As explained in Section 3, for each of the 14 selected markers, 3D positions are obtained for each frame. Thus, we have a vector of 42 components for every frame (3 coordinates for each of the 14 selected markers for a frame). Thus each vector in 3D space is represented as

$$\mathbf{d}=(x_1, y_1, z_1, x_2, y_2, z_2,\ldots x_n, y_n, z_n)^{\mathbf{T}} \in R^{3n}, \qquad n=14$$

It can be easily observed that the data is highly interdependent, because of the very nature of the facial movements. For example, the displacements of the points around the lip area are highly correlated to each other, and to the jaw movements, as they cannot be physically moved independent of each other. The lower lip movements are directly linked to the global movements of the jaw. In addition, there can be local movement of the lips independent of the jaw movement. Similarly, movement of the corner lips as in lip puckering and lip sucking directly affects the movement of the cheeks. However, just observing the capture data in the form of 3D position trajectories does not enlighten us much about how these movements are inter-related. This inter-relation is the key factor for realistic animation and we employ PCA to extract this relation that occurs due to natural constraints.

4.1 Principal Component Analysis

PCA is a well-known multivariate statistical analysis technique aimed at reducing the dimensionality of a dataset, which consists of a large number of interrelated variables, while retaining as much as possible of the variation present in the dataset. This is achieved by transforming the existing dataset into a new set of variables called the principal components (PC). These are uncorrelated and are ordered so that the first few PCs retain the most of the variation present in all of the original dataset. We explain the basic concepts behind the PCA here for completeness. This description is based on [16].

Let **x** be a vector of p random variables under consideration. In order to study the correlations between the p random variables, it is not trivial to observe the data as is, unless p is very small. Hence, an alternative approach is to look at a fewer derived variables, which preserve most of the information about these variations. The first step is to look for a linear function $\boldsymbol{\alpha_1}'\boldsymbol{x}$ of the elements of $\boldsymbol{x}$ which has maximum variance, where $\boldsymbol{\alpha_1}$ is a vector of p constants; α_{11}, $\alpha_{12, \ldots\ldots,}\alpha_{1p,}$ and ′ denotes transpose, so that

$$\boldsymbol{\alpha_1}'\boldsymbol{x} = \alpha_{11}x_1 + \alpha_{12}x_2 + \ldots + \alpha_{1p}x_p = \sum\nolimits_{j=1}^{p} \alpha_{1j}x_j \qquad (1)$$

Next, look for a linear function $\boldsymbol{\alpha_2}'\boldsymbol{x}$, uncorrelated with $\boldsymbol{\alpha_1}'\boldsymbol{x}$, which has maximum variance, and so on. Thus at the kth stage, a linear function $\boldsymbol{\alpha_k}'\boldsymbol{x}$ is found which has maximum variance subject to being uncorrelated with $\boldsymbol{\alpha_1}'\boldsymbol{x}$, $\boldsymbol{\alpha_2}'\boldsymbol{x}$, …, $\boldsymbol{\alpha_{k-1}}'\boldsymbol{x}$. The kth derived variable, $\boldsymbol{\alpha_k}'\boldsymbol{x}$ is the kth Principal Component (PC). Like this, up to p PCs can be found, but it is hoped that the number of PCs found is much less than p.

To find the PC's, let us consider that the random variables **x** has a known covariance matrix, **C.** This is the matrix whose $(i,j)^{th}$ element is the covariance between the i^{th} and j^{th} elements of **x** when $i \neq j$, and the variance of the j^{th} element of **x** when $i=j$. It turns out that, for k = 1,2,…,p, the kth PC is given by

$$\mathbf{z_k} = \boldsymbol{\alpha_k}'\boldsymbol{x}$$

where $\boldsymbol{\alpha_k}$ is an eigen vector of **C** corresponding to its kth largest eigen value $\boldsymbol{\lambda_k}$. Furthermore, if $\boldsymbol{\alpha_k}$ is chosen to have unit length ($\boldsymbol{\alpha_k}\boldsymbol{\alpha_k}' = 1$), then $var(\mathbf{z_k}) = \boldsymbol{\lambda_{k,}}$ where $var(\mathbf{z_k})$ denotes the variance of $\mathbf{z_k}$. Thus, from the point of view of the implementation, finding the PCs is equivalent to finding the eigen vectors of the covariance matrix **C**. For the derivation of this result, the reader is referred to [16].

We use the entire set of motion trajectory data of the 3D positions of the selected markers as an input to the PCA analysis. Thus, each input vector is 42 dimensional. As a result of the PCA on all the frames of the captured data, the matrix **T** whose columns are the eigen vectors corresponding to the non-zero eigen values of the above mentioned covariance matrix **C**, forms the transformation matrix between the 3D vector space and the transformed *viseme space.* Thus, each 42 dimensional vector $\boldsymbol{d}$ can be mapped onto a unique vector $\boldsymbol{e}$ in this space.

$$\mathbf{e=Td} \qquad (2)$$

The inverse transformation is appropriately given by

$$\mathbf{d=T^Te} \qquad (3)$$

where $\mathbf{T^T}$ denotes the transpose of the matrix **T**. Each distinct viseme is represented as a point in this transformed multidimensional space. The very nature of the dynamic speech input data makes it sure that the transitions between these points in the newly formulated *viseme space* correspond to the real-life like transitions of the markers in 3D position space. We exploit this for smooth and realistic speech animation. The next subsection explains what these "abstract" principal components represent in real life.

4.2 Contribution of principal components

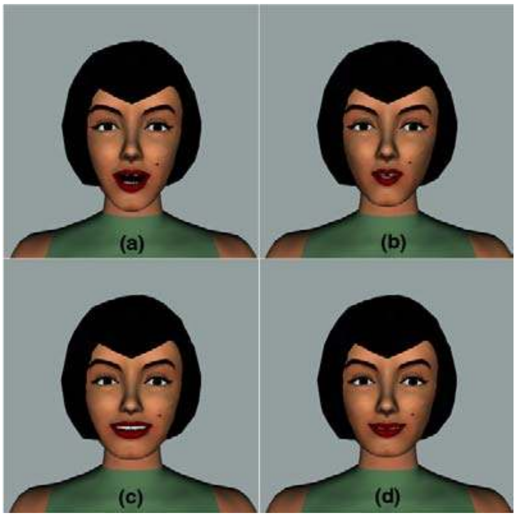

Figure 3: Influence of the first four principal components

Once we obtain the principal components, we can easily establish their role in generating facial animation. We notice that the last three eigen values of the covariance matrix of the input data are zero, and thus the dimensionality of the space is directly reduced to 39. In addition, 99% of the variation has been accommodated in only the first 7 principal components. In general for any dataset, the principal components may or may not represent any real-life parameters. We notice however that, for the facial capture data; they are closely related to facial movements. For this, we allow only one principal component to vary at a time keeping others at the default neutral position. Then we apply an inverse transformation (Equation 3) to obtain the 3D position of the markers. From these 3D positions, the MPEG-4 FAPs are extracted.

Figure 3 shows the influence of the first four principal components. They are related to the opening jaw (a), lip protrusion (b), lips opening (c), and vertical movement of lip corners (d) as in smiling. Note that these movements are not local, meaning that the jaw opening does not only result in the vertical displacement of the jaw, but also the rotation movement, as in real life. Also, the lip opening affects movement of cheeks. These principal components can be used to define a new parameter space for facial animation and a more intuitive interface for the designers of the facial animation. These parameters will be directly linked to the correlated facial movements.

5. SPEECH ANIMATION

In this section, we turn to the most important result of the analysis explained so far. Speech animation typically makes use of temporized phonemes extracted from either real or synthetic speech as the building blocks. These phonemes are mapped onto visemes for which facial animation parameters are pre-defined. In this section we address the problem of realistic speech animation by using the *viseme space* to obtain smooth transition between visemes thus resulting into realistic speech animation.

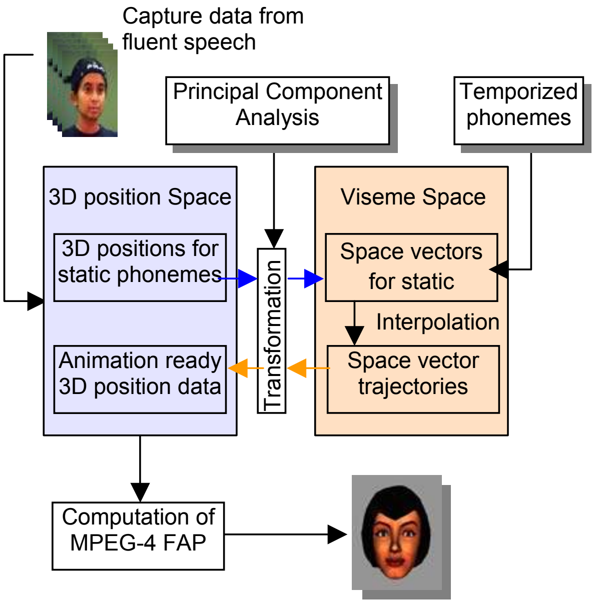
Capture data from fluent speech
Principal Component Analysis
Temporized phonemes
3D position Space
3D positions for static phonemes
Animation ready 3D position data
Transformation
Viseme Space
Space vectors for static
Interpolation
Space vector trajectories
Computation of MPEG-4 FAP

Figure 4: Generating facial animation from speech

We look at an application where synthetic or real speech is used along with phoneme segmentation to generate speech animation. As explained in Section 2, for all the phonemes (or visemes) used for speech animation, we transform the 3D position vectors into the newly generated *viseme space*. These essentially form the key-frames of the speech animation, one key-frame corresponding to each phoneme in the speech. These key-frame visemes have been captured separately during the same session as that of the continuous speech capture. Thus, the captured speech data is mainly used for analysis and for deriving the *viseme space*, and not for directly obtaining the keyframes. The key-frames are positioned at 30% of the total duration from the start of that phoneme. This value was chosen experimentally, and can be controlled in an animation. We then use the cubic spline interpolation to get the trajectories for all the frames of the PCs for a speech animation sequence. The interpolated trajectories are then transformed back to the 3D position space and the FAPs are calculated. This "marching" through the *viseme space* results in realistic speech animation. This process is not an alternative to "co-articulation" (*e.g.* the overlaps are not considered); however, the results obtained demonstrate the high degree of realism in the resulting speech animation. Thus, this method can be used as a sub-step in co-articulation algorithms to enhance the overall results.

Figure 4 demonstrates the whole processing pipeline. In case of the previously used co-articulation algorithms, the interpolation between the phonemes is done in the 3D position space, or on the facial animation parameters. In such case, it is difficult to explicitly consider the inter-relations between these parameters with respect to each other. When interpolation is done in the *viseme space*, this interrelation is automatically considered. The interpolation in *viseme space* takes into account the actual transitions reflected in the 3D position space, as each principal component is responsible for highly correlated facial movements. This automatically ensures realistic animation.

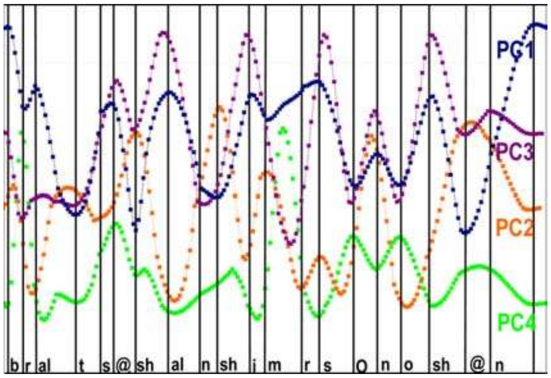

Figure 5: Spline interpolation for the Principal Components

Figure 5 shows the interpolated principal components in the *viseme space*. The phoneme segmentation for the sentence "Bright sunshine shimmers on ocean" was extracted using automatic speech recognition. One key-frame is assigned per viseme. Note that these PCs are not expressed to the scale here, but we show the overall distribution for fluent speech sample.

6. CONCLUSION

We have carried out statistical analysis of the facial feature point movements. As the data is captured for fluent speech, the analysis reflects the dynamics of the facial movements related to speech production. The results of the analysis were successfully applied for a more realistic speech animation. Use of MPEG-4 feature points for data capture and facial animation enabled us to restrict the quantity of data being processed, at the same time offering more flexibility with respect to the facial model.

We would like to further study the effect of using various time envelopes for the transitions in the *viseme space*. It is important to combine the viseme transition method proposed here with the co-articulation methods, instead of using linear or non-linear time envelopes that are manually defined. It is further interesting to observe that mixing of expressions during speech animation can be handled using this technique. In this case, the addition between the expressions and viseme vectors should be done in the *viseme space*. Current efforts are in the direction of evaluating the results obtained for such “expressive speech”.

7. ACKNOWLEDGEMENTS

This work is supported by the EU ACTS Interface project. We are thankful to the MIRALab staff for their valuable support and help. We would like to give special thanks to Chris Joslin for proof reading this paper and to Tom Molet for his help in data capture.

8. REFERENCES

1. F. Parke, "Parameterized Models for Facial Animation", *IEEE Computer Graphics and Applications*, Vol.2, No. 9, pp 61-68, November 1982.
2. K. Waters, "A Muscle Model for Animation Three Dimensional Facial Expression", *Computer Graphics*, Vol. 21, No. 4, pp 17-24, July 1987.
3. D. Terzopoulos, K. Waters, "Physically Based Facial Modelling, Analysis and Animation", *Journal of visualization and Computer Animation*, Vo. 1, No. 2, pp 73-90, 1990.
4. E. Friesen WV (1978), *Facial Action Coding System: A Technique for the Measurement of Facial Movement,* Palo Alto, California: Consulting Psychologists Press.
5. B. Grandstrom, "Multi-modal speech synthesis with applications", in G. Chollet, M. Di Benedetto, A. Esposito, M. Marinaro, *Speech processing, recognition, and artificial neural networks*, Springer, 1999.
6. D. R. Hill, A. Pearce, B. Wyvill, "Animating speech: an automated approach using speech synthesized by rule", *The Visual Computer*, 3, pp. 277-289, 1988.
7. C. Pelachaud (1991), *Communication and Coarticulation in Facial Animation*, PhD thesis, University of Pennsylvania, 1991.
8. M. M. Cohen, D.W. Massaro, "Modelling co-articulation in synthetic visual speech", in N. M. Thalmann and D. Thalmann, *Models and techniques in Computer Animation*, Springer-Verlag, 1993, pp. 139-156.
9. T. Kuratate, H. Yehia, E. V-Bateson, “Kinematics-based synthesis of realistic talking faces”, *Proceedings AVSP’98*, pp. 185-190.
10. L. M. Arsal, D. Talkin, “3-D face point trajectory synthesis using an automatically derived visual phoneme similarity matrix”, *Proceedings AVSP’98.*
11. Specification of MPEG-4 standard, Moving Picture Experts Group, http://www.cselt.it/mpeg/.
12. S. Kshirsagar, S. Garchery, N. Thalmann, “Feature Point Based Mesh Deformation Applied to MPEG-4 Facial Animation”, *Deformable Avatars,* Post-Proceedings of Deform’2000 Workshop on Virtual Humans, Geneva, Switzerland, Nov. 29-20 2000, Kluwer Academic Publishers, pp. 24-34.
13. VICON Motion Systems http://www.vicon.com
14. TIMIT Acoustic-Phonetic Continuous Speech Corpus, http://www.ldc.upenn.edu/Catalog/LDC93S1.html.
15. W. Martin and J. Aggarwal, *Motion Understanding Robot and Human Vision*, Kluwer Academic Publishers, 1988.
16. I. T. Jollife, *Principal Component Analysis*, Springer Verlag, New York, 1986.