TOWARDS THE *FACECODER*: DYNAMIC FACE SYNTHESIS BASED ON IMAGE MOTION ESTIMATION IN SPEECH

Christian Kroos, Saeko Masuda, Takaaki Kuratate and Eric Vatikiotis-Bateson

ATR International – Information Sciences Division
2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, JAPAN
E-mail: {ckroos,smasuda,kuratate,bateson}@isd.atr.co.jp

ABSTRACT

The (digital) transmission of talking faces requires a high bandwidth that not every target channel is able to provide, even if powerful image compression algorithms are used. Therefore, a special *face coding* algorithm would be highly desirable. Unfortunately, development of such an algorithm has been hindered by the general problem of image motion estimation. In this paper we present a video-based system for face motion processing similar to the well-known voder-vocoder system for processing and coding acoustic speech signals. Like the vocoder, our 'face coder' consists of two independent parts: an analysis part for tracking non-rigid face motion, and a synthesis part for producing face animations. Results are shown for face motion tracking and the subsequent animation derived from either the raw motion data or the outcome of Principal Component Analysis. The automatic tracking results were evaluated by comparison with a set of manually tracked points.

1. INTRODUCTION

It has been one of the primary goals underlying research in audio-visual speech processing to find methods to reduce the bandwidth needed for transmitting talking faces. Analogous to the *Vocoder* invented by H. Dudley in the mid-1930s for Bell Laboratories, an algorithm is needed that reduces the spatially complex and temporally variable input signal, associated with the speaking face combined with the speech acoustics, to a simpler set of parameters that (ideally) varies slowly over time. Using an appropriate synthesis algorithm, it should then be possible to reconstruct the original signals from the reduced parameter set.

The *face coding* approach presented here achieves the pairing of face motion analysis and synthesis, consisting of an analytic algorithm that tracks non-rigid face motion and a simple animation method. There are other systems for coding face motion such as FACS (facial action coding system), developed by Ekman and Friesen [1] and adopted as the MPEG4 standard. FACs is supposedly able to code *all* kinds of face motion, although to our knowledge it has never been adequately tested for speech behavior. Our system was specifically created to code the motion of speaking faces. Our tracking method differs from most other methods by globally tracking the face surface rather than selected features (e.g, see [2], [3], and for an exception of this rule [4], [5]). On the synthesis side considerable effort in recent years has been expended developing text-to-audio-visual-speech systems. Typically these systems use morphing between visemes (e.g., [6]) or concatenation of the visual complement of triphones (e.g., [7]). Our method uses the actual face kinematics to synthesize realistic image sequences.

In contrast to the acoustic domain, transmission difficulties in the image domain do not arise from temporal variablility, since conventional video samples motion at a relatively low rate (NTSC: 60 fields/s, PAL: 50 fields/s). Rather, it is the spatial complexity of most natural images - and certainly faces - that necessitates high spatial resolution and complicates image processing.

How then do we code a speaking face? The first step is to separate the two-dimensional (2D) image, or texture map, of the face from the face motion, or kinematics. In order to appear natural to viewers, texture maps must have sufficient spatial detail. Indeed, the cosmetic requirements of viewers may be quite a bit higher than the *functional* ones. Recent studies have shown that identification and speech reading can be achieved at low spatial frequencies (e.g., [8]).

Since the kinematic information is crucial for speech, we assume that the functionally important spatiotemporal behavior of the face can be modeled with relatively sparse sets of anchor points. Furthermore, the change in location over time of these anchor points at a coarse level might be a good prediction for the behavior of a much denser set at a finer level, due to the fact that most of the face is a connected continuous surface. The anchor points, of course, are assumed to vary rapidly, while the texture map presumably changes quite slowly with the exceptions of the opening and closing of the oral aperture and the eyes, and the appearance and disappearance of certain wrinkles [9] [1] .

2. VIDEO-BASED FACE MOTION TRACKING

We have developed an image motion estimation algorithm that was explicitly designed to track face motion in video sequences. It exploits specific properties of the appearance of the face in a video sequence. Since a detailed description of the method already exists [10] [11], we give here only a brief overview. The method can be broken down into two independent parts: initialization and motion tracking

2.1. Initialization

The purpose of the initialization was to fit to the face in one image the three-dimensional (3D) ellipsoid mesh model that was then used by the tracking procedure to register the face in the video image sequence. This was done manually by marking a minimum of six points in the image frame that defined the outline and orientation of the face. An ellipse fitting procedure then assigned a closely fitting ellipse to the face from which the parameters of the ellipsoid were derived (see Figure 1). The procedure was required only once for the entire input sequence and could be easily replaced by an automatic face detection method (e.g. [12]).

2.2. Motion tracking

Measurement of the face motion was done on a frame-to-frame basis by determining the location changes of small parts of the face surface using a multi-resolution analysis of the video image data.

Head motion naturally accompanies speech and other expressive orofacial behaviors. However, it imparts relative motion to the rest of the face that must be removed in order to obtain accurate measures of the face motion alone. Although it is possible to recover the rigid body head motion from video image sequences using techniques similar to those in [13] and [14], our primary interest had been to make accurate measures of the non-rigid face motion. Therefore, we used OPTOTRAK (Northern Digital, Inc.) to make precise time-varying measures of head orientation and position.

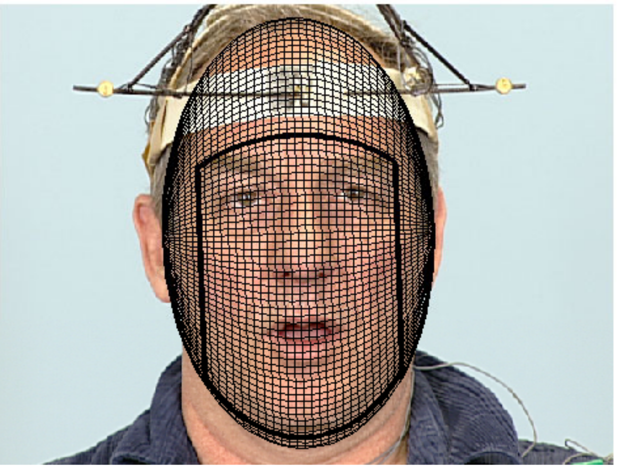

Figure 1. Starting frame with undeformed mesh. The bold lines delimit the tracked area, the remaining nodes were fixed.

The first processing step for any incoming video frame was to apply a two-dimensional discrete wavelet transformation (DWT, see [15]). A cascade of digital half-band filters were applied to the image, where the filters have specific properties (*quadrature mirror filters*). From the resulting signals at each level the so-called *subbands* were chosen for tracking. If not sub-sampled, as in our case, the subbands comprise a version of the original image that is bandlimited in its spatial frequency content. For our specific implementation, the filters conformed to a biorthogonal scheme with cubic spline wavelets of compact support (see [16]).

The ellipsoid mesh was then projected onto the subband images using a primitive perspective camera model and at the resolutions appropriate to the spatial frequencies of the subbands. The tracking algorithm employed a coarse-to-fine strategy: A coarse mesh was applied first to capture the motion of large face areas; then successively finer meshes were applied, capturing the motion of progressively smaller areas. For any two consecutive frames, the goal was to find the most likely positions in the second frame of a number of 'search segments' derived from the filtered texture map in the first frame. In our approach these 'search segments' were defined as the area enclosed by the four neighboring nodes surrounding a center mesh node, which served as anchor point for the analysis (and later the synthesis).

A simple warping process involving an affine transformation of each quadrant of the search segment was used to register already known changes - e.g. the motion tracking results on a coarser level. Then, the search segment was compared with a confined area around its predicted position in the second frame using a 2D cross-correlation. The necessary shift vectors (or lags) were extracted from the seg-

[1] We are concerned only about wrinkles that appear and disappear as the face surface deforms during face motion, not those that appear, but do not disappear, as a result of aging.

ment itself in order to avoid searching through the whole image space. The shift value yielding the highest correlation was selected as the actual motion vector for the center node of the search segment, and the mesh model representation of the second frame was deformed accordingly.

So long as the desired final resolution had not been reached, the next to last step entailed bilinear interpolation of the node coordinates of the next finer mesh.

After passing through all selected wavelet levels, thereby deforming progressively finer representations of the mesh model in a stepwise manner, the perspective distortion of the projected final mesh model was reversed and the pose deviation due to head motion was corrected. The result was a sequence of stabilized ellipsoid mesh models where the time-varying changes in the location of the mesh nodes represented the face motion. Note that 2D tracking resulted from the constraint that the mesh nodes lie on the surface of the ellipsoid at all times.

3. TALKING FACE SYNTHESIS

In this section we describe two ways to synthesize 2D talking faces using the motion tracking results. The first method used the raw motion measures directly. The second method used motion coefficients derived from analysis of all the motion tracking data available at the time of the synthesis.

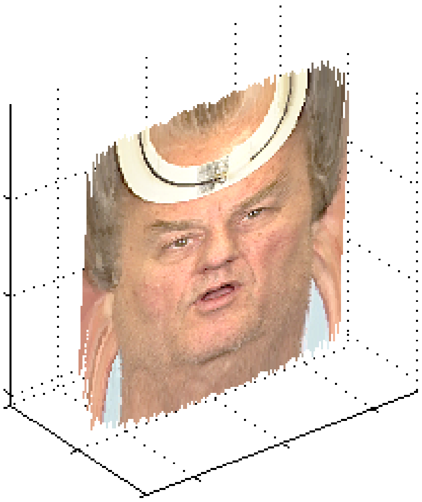

Figure 2. The ellipsoid surface (consisting of the texture pixels) 'unfolded'

3.1. Based on raw motion tracking results

First, a high resolution texture map of the ellipsoid mesh model must be extracted. This was taken (arbitrarily) from the input frame where the motion tracking started. At this stage the mesh was rotated in the image plane to fit the face and projected, but was not deformed with respect to the motion tracking. That is, the coordinates of all mesh nodes in the image were already known and existed in a normalized, 'unfolded' 2D matrix corresponding to the two-coordinate matrices that were used to generate the model in the initialization process. These correspond to the variables u and v in the parameterization formula for the ellipsoid from standard analytical geometry[2]:

$$\begin{pmatrix} x = a \ \cos u \ \cos v \\ y = b \ \sin u \ \cos v \\ z = c \ \sin v \end{pmatrix} \tag{1}$$

where a, b and c are the parameters of the ellipsoid's main axes.

For any pixel lying inside the projected mesh's outline, its image plane coordinates were easily obtained by reversing rotation and projection. However, the corresponding depth values had to be computed by solving the ellipsoid equation

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1 \tag{2}$$

taking into account the rotation, translation and projection of the mesh in the reference frame. Then, reversing the parameterization formulae and 'unfolding' the ellipsoid surface containing the pixel values yielded the normalized 2D coordinates (see Figure 2). Using the u and v values of the mesh nodes, arbitrarily coarse or fine texture maps could then be obtained via interpolation. Of course, the information content of the texture maps was limited by the pixel density of the video frame. Figure 3 shows an example. The texture map was then applied to the sequence of deformed mesh models using the graphics capabilities of 3D rendering programs, library functions or - as in our case - MATLAB.

3.2. Based on principal components

When there is a considerable amount of data already available at the time of the synthesis, the synthesis can also be done using Principal Component Analysis (PCA), where the face motion data are decomposed into a series of orthogonal components (PC's). The series is ordered according to the proportion of the total variability accounted for by each component. There are several potential benefits to PCA. By choosing a subset of the strongest PC's, the dimensionality of the data can be reduced while retaining the

[2] Usually equally spaced u and v values would be used, ranging from $-\pi/2$ to $\pi/2$ for v and from 0 to 2π for u (or from 0 to π to create a half-ellipsoid, which is enough for our purposes). This formula generates mesh nodes that are equally space on the 3D ellipsoid, but not in the 2D image plane where, for the purposes of defining search segments, mesh nodes should be spaced as nearly equal as possible. Therefore, mesh node spacing was normalized using the inverse of the ellipsoid surface. This resulted in the straight vertical lines of the ellipsoid mesh and the almost square shaped areas in the central part of the mesh (see Figure 1) .

relevant aspects of the behavior. This leads to a noticeable reduction in noise. Furthermore, some PC's may characterize functional or structural aspects of the behavior such as the vertical motion of the jaw or shaping of the lips. These components can then be systematically varied and their effects on human audio-visual perception tested. Similarly, modulation of such componenents (e.g., amplification, distortion) might augment intelligibility in multimodal speech transmission and automatic multi-modal speech recognition.

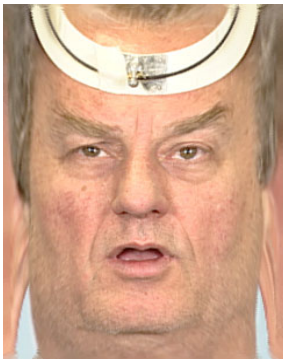

Figure 3. Full interpolated texture map

For this paper we analyzed video clips of a male American speaker uttering the first fifteen sentences of the CID 'Everyday' corpus. Each clip contained one sentence and the clip's length ranged from 39 to 153 (average: 91) video fields. Only the area within the bold lines in Figure 1 was tracked. We conducted a PCA based on the covariance matrix of the motion tracking results at the middle level of the tracking to keep the number of nodes (= variables) manageable at 546.

Bartlett's test was used to test for equality of the last k eigenvalues [17]. Using α=0.001 as the threshold for significance, the procedure indicated that the first 470 of the 546 PC's were significantly different from each other. The remaining PC's represent only the inherent variation (i.e., no covariance) or noise associated with a few isolated mesh nodes. Since *Bartlett's test* is conservative, sometimes keeping more PC's than necessary, we compared it with *Velicer's partial correlation procedure* [17]. Velicer's method examines the partial correlations among the original variables with one or more PC's removed. Starting with all but the first PC removed, PC's are added until additional PC's would represent more variance than covariance. *Velicer's method* is particularly useful for our purposes, because the motion tracking algorithm becomes more sensitive to tracking noise than data at the higher spatial frequencies. Thus, we accepted *Velicer's method* recommendation that only 166 PC's were needed to describe the motion data. These 166 PC's were then used to estimate the individual mesh node coordinates over time.

Subsequently, the synthesis procedure for PC's was almost identical to the one applied to the raw motion tracking data, but with one crucial difference in the way the reference frame is defined: When synthesizing motion from raw data, a different reference frame was used in the motion tracking procedure for each utterance clip; but when basing the synthesis on the PCA results, a single reference must be used for all utterance clips. In the latter case, the temporal displacement between reference and measured frames was quite large for almost all utterances. This violates two assumptions pertinent to the success of the tracking algorithm.

(i) The image brightness function is assumed to be constant from one frame to the next. That is, under ideal recording conditions, the intensity values of objects in the image do not change over time. However, even recording indoors, changes in absolute head orientation and the like will cause the image brightness function to change. Thus, the greater the temporal gap between reference and measured frame, the more likely it is for the image brightness function to vary.

(ii) The face motion is assumed to be smooth in the captured video sequence. To avoid jumps between the global reference frame and the starting frame of an individual clip, a starting frame can be selected that contains a face posture similar to the one in the global reference frame. However, this was not always possible - e.g., sometimes the mouth changed between open and closed between the reference and the starting frames.

In general the algorithm coped with these violations, thus enabling the synthesis to be done using a single, global reference frame.

4. RESULTS AND DISCUSSION

The Quicktime movie `[fm_trak.mov]` shows the motion tracking results for 'Our janitor sweeps the floors every night'. The mesh is superimposed on the grayscale conversion of the original video sequence. The synthesis generated from the raw motion tracking results is shown in the Quicktime movie `[fsyn_raw.mov]`, with the original and synthesized faces shown on the right and left hand sides, respectively. Note that no post processing whatsoever was applied; thus, the synthesis accurately reflects the raw motion tracking data. Similarly, the the PCA-based synthesis is shown by the Quicktime movie `[fsyn_pca.mov]`.

The quality of the animation depends crucially on the accuracy of the motion tracking. Since the tracking for this experiment was based on fields (60 Hz) rather than frames (30 Hz), high speed events such as the onsets (and offsets) of eye blinks were more consistently captured. On

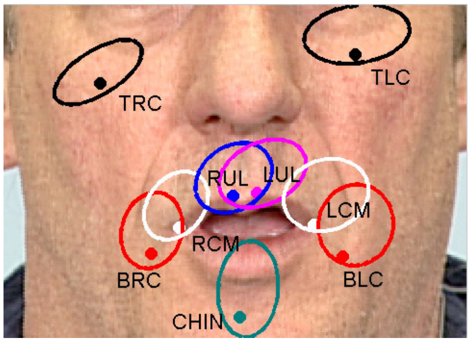

Figure 4. Location and movement range of the manually marked points (head motion not removed)

the other hand using fields reduces the vertical image resolution due to the interlacing of successive fields (corresponding to even and odd lines of the image). More important, in order to track motion continuously through time, twice as many images have to be processed. As with most time-varying estimation techniques, error accumulates and degrades the performance of our algorithm over time, as can be seen towards the end of the sample movies.

Since visual inspection of the resulting animations is not sufficient to judge the accuracy of the tracking algorithm, we manually tracked nine points on the subject's face for the first 8 of 15 sentences (877 of 1363 image fields). Ideally for this test, the points should be distributed randomly on the face, but it was impossible to find enough arbitrarily assigned points that could be identified reliably in every frame. Therefore, landmark coordinates were used comprising the mouth corners, two points on the upper lip, and five points on the cheeks and chin marked by small blemishes (not visible in the images reproduced for this paper). Figure 4 shows the locations (filled circles) and the movement ranges (including head motion). For the ellipses enclosing each location, size was three times the standard deviation and axis orientation was derived from PCA of the manually tracked position coordinates.

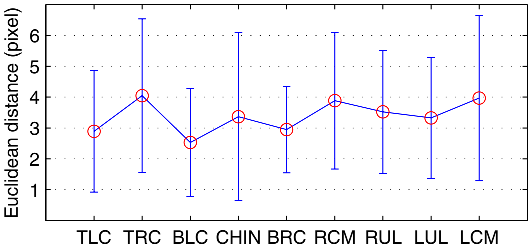

Figure 5. Means and standard deviations of the discrepancy between manual and automatic tracking of nine points on the cheeks (TLC, TRC, BRC, BLC), upper lip (RUL, LUL), lip corners (RCM, LCM), and the chin.

The manual and the automatic tracking results were compared by calculating the Euclidian Distance between each manually marked point and the mesh node closest to it in the *global* reference frame, setting it to zero, and then computing the distances between node-point pairs over time. Figure 5 shows the discrepancy between the methods for the nine points for all image fields tested. As can be seen, there is a mean discrepancy (change in Euclidean Distance) of 3-4 pixels. Neither the mean nor the standard deviation of discrepancy seems to depend on the location or degree of face motion – e.g., compare the relatively motionless upper cheek with highly mobile chin. It is clear, however, that the discrepancy increases over time. Figure 4 shows the generally monotonic growth in mean discrepancy for *all* markers over the time course of *each* sentence.

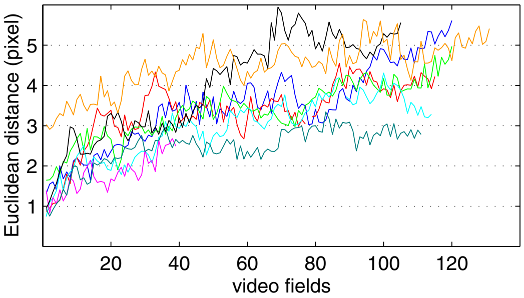

Figure 6. Mean discrepancy of all points over time for each sentence

5. OUTLOOK

Obviously, the algorithm has to be evaluated for a wider range of speakers and conditions (gender, language, etc.), and several issues need to be addressed in order to improve the accuracy of the motion tracking upon which the subsequent synthesis depends. As can be seen in the animation movies, the algorithm underestimates the fast closing movements of the mouth and the eyelids. This is caused by a constraint limiting the amount of frame-to-frame node displacement (preventing overlap of nodes). This problem can be solved by specifying intermediate nodes that are used to interpolate between tracked nodes, but are not tracked themselves.

Additionally, a more sophisticated camera model and a better warping technique for the adaptation of the search segment should be used. Use of a windowing function on the search segment would further improve the tracking, since it assigns more weight to pixel values closer to the center node (the anchor point for tracking and synthesis) than to outlying pixels.

Further, we must determine whether or not the cross-correlation used to determine the search segment's position in an incoming frame is the optimal method. An alternative would be an optical flow method constrained to an affine model as used by [2] for the tracking of facial features.

We currently use spline wavelets, because they exhibit linear phase. This ensures that the mesh is projected on the same area of the face at every wavelet level. However, a different wavelet might be more suitable for the tracking; e.g., the one corresponding to a set of orthonormal, maximally flat FIR filters described in [18]. But that will also depend on finding an appropriate alignment of the output signal across the different wavelet levels.

Last but not least there is the question of whether the intensity values of the search segments must be updated with each frame. Currently, this is largely responsible for the accumulation of estimation error over time. An extreme alternative would be to use only the values from the starting (reference) frame (as in the synthesis), but then lighting changes could lead to unrecoverable errors. We made preliminary investigations, but mistracking in the area of the opened/closed mouth caused a disintegration of parts of the mesh. Using *robust statistics* does not work, since even a 'perfect' error norm does not prevent single mesh nodes from *gradually* (over several frames) moving in the wrong direction and creating an overlap. A middle-ground solution in which search segments are updated only occasionally would be best. That is, assuming that the face motion is faster on average than changes of intensity, problems of the lighting model might be reduced by temporal low pass filtering during the search segment update.

6. CONCLUSION

We presented an approach to *coding* talking faces. In the analysis, the face motion was tracked using a multi-resolution analysis of the image data and a set of parametrized ellipsoid mesh models with variable node density. The deformation of the mesh over time represents the face motion and the coordinates of the mesh nodes serve as coding parameters for the spatiotemporal behavior of the face. For face motion synthesis, the texture map of a reference frame was extracted and then applied to the sequence of deformed mesh models. Faces can be animated from either raw motion measures or principal components. Comparison of manual and automatically tracked motion showed that measurement discrepancy increases over time.

References

[1] P. Ekman and W. V. Friesen. *The facial action coding system (FACS): A technique for the measurement of facial action*. Consulting Psychologists Press, Palo Alto, CA, 1978.

[2] M. Black and Y. Yacoob. Recognizing facial expressions in image sequences using local parameterized models of image motion. *Int. Journal of Computer Vision*, 25(1):23–48, 1997.

[3] K. Mase. Recognition of facial expression from optical flow. *IEICE Transactions*, E74:3474–3483, 1991.

[4] Y.-T. Wu, T. Kanade, J. Cohn, and C. Li. Optical flow estimation using wavelet motion model. In *International Conference on Computer Vision*, pages 992–998, Indian, January 1998.

[5] I. Essa and A. Pentland. Coding, analysis, interpretation and recognition of facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19 (7), July 1997.

[6] T. Ezzat and T. Poggio. Videorealistic talking faces: a morphing approach. In *Proc. of the First ESCA Workshop on Audio-Visual Speech Processing, AVSP'97*, pages 141–144, Rhodes, Greece, September 1997.

[7] C. Bregler, M. Covell, and M. Slaney. Video rewrite: Visual speech synthesis from video. In *Proc. of the First ESCA Workshop on Audio-Visual Speech Processing, AVSP'97*, pages 153–156, Rhodes, Greece, September 1997.

[8] K. G. Munhall, C. Kroos, and E. Vatikiotis-Bateson. Band-pass filtered faces and audiovisual speech perception. *Journal of the Acoustical Society of America*, 109 (Suppl.1):2314, 2001.

[9] L. Revèret, G. Bailly, and P. Badin. MOTHER: a new generation of talking heads providing a flexible articulatory control for video-realistic speech animation. In *International Conference on Speech and Language Processing*, Beijing, China, October 2000.

[10] C. Kroos, T. Kuratate, and E. Vatikiotis-Bateson. Listen to the face - measuring the face kinematics of speech from video sequences. In *5th Seminar on Speech Production: Models and Data*, Bavaria, Germany, 2000.

[11] C. Kroos, T. Kuratate, and E. Vatikiotis-Bateson. Video-based face motion measurement. *Journal of Phonetics (special issue)*, to appear.

[12] A. Nefian, M. Khosravi, and M. Hayes. Realtime detection of human faces in uncontrolled environments. In *Proceedings of SPIE conference on Visual Communications and Image Processing, Vol. 3024*, pages 211–219, San Jose, California, USA, 1997.

[13] M. La Cascia, J. Isidoro, and S. Sclaroff. Head tracking via robust registration in texture map images. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 1998.

[14] A. Schödl, A. Haro, and I. A. Essa. Head tracking using a textured polygonal model. Technical report, GIT-GVU-98-24, 1998.

[15] G. Kaiser. *A friendly guide to wavelets*. Birkhäuser, Boston, 1994.

[16] G. S. Sánchez, N. G. Prelic, and S. J. G. Galán. *Uvi_Wave. Wavelet Toolbox for use with Matlab*. Departamento de Tecnoloxías das Comunicacións. Universidade de Vigo, Vigo, second edition, July 1996.

[17] J. E. Jackson. *A user's guide to principal components*. John Wiley & Sons, New York, 1991.

[18] P. P. Vaidyanathan. *Multirate systems and filter banks*. Prentice Hall, 1993, pages 532–536.