In Proceedings of the IEEE CVPR, pp. 892-897, Santa Barbara, 1998

Extraction and Classification of Visual Motion Patterns for Hand Gesture Recognition

Ming-Hsuan Yang and Narendra Ahuja
Beckman Institute and Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801

Abstract

We present a new method for extracting and classifying motion patterns to recognize hand gestures. First, motion segmentation of the image sequence is generated based on a multiscale transform and attributed graph matching of regions across frames. This produces region correspondences and their affine transformations. Second, color information of motion regions is used to determine skin regions. Third, human head and palm regions are identified based on the shape and size of skin areas in motion. Finally, affine transformations defining a region's motion between successive frames are concatenated to construct the region's motion trajectory. Gestural motion trajectories are then classified by a time-delay neural network trained with backpropagation learning algorithm. Our experimental results show that hand gestures can be recognized well using motion patterns.

1 Introduction

This paper is concerned with the problem of detecting two-dimensional motion across image frames and classifying motion patterns associated with certain hand actions. Classification is aimed at the recognition of the action represented by the motion pattern. Such a capability is quite central to human vision, and useful in many application domains. For concreteness, both extraction of motion patterns and their interpretations are carried out for the domain of hand gesture recognition in this work. However, the results can be easily extended to other scenarios. Most of the past work on gesture recognition focuses on static hand gestures, with much less attention given to the dynamic characteristics of gestures. Our work is aimed at recognizing gestures of American Sign Language (ASL) using spatio-temporal analysis of gestural motion trajectories.

We perform motion segmentation to group pixels of similar motion into regions and find region correspondences across frames. Although there are many motion regions in each frame, the movements of palm regions contain significant information about the gesture meaning, and therefore palm motion is extracted. Further, ASL experts have pointed out that gesture interpretation requires not only the motion pattern but also relative locations of the hand with respect to other landmark parts of the body such as the head or shoulders [2]. Therefore, human head region is extracted for use as reference in gesture recognition. To distinguish among the different moving regions, we use color and geometric characteristics. Both head and palm regions have skin color and have similar elliptic shapes, but differ in size. Motion regions with skin color are first identified, and a connected component analysis is then performed to merge neighboring regions until the shape of the merged region is approximately elliptic. The head and palm regions are discriminated based on difference in size. The palm regions are extracted from each frame and the affine transformations between corresponding regions in successive frames are computed and concatenated to obtain gestural motion trajectories.

Recognition of the motion patterns is performed using a time-delay neural network (TDNN) with an error backpropagation learning algorithm. Several studies have shown that such networks are capable of classifying spatio-temporal signals [11]. TDNNs are appropriate for recognizing motion patterns because the input data are organized as a temporal sequence, where the data sampled during a time window are input to the network simultaneously. To get a time sequence of output data, this window is moved stepwise in time. Our experiments verify that gestural motion patterns can be classified by TDNN.

2 Related Work

Gesture recognition consists of two major components: pattern extraction and classification. Many of the gesture recognition applications use specialized colored gloves or markers [4]. Electromagnetic sensors and stereo vision have also been experimented with to locate the signer in video [12]. Pfinder [3] adopts a maximum a posteriori probability approach to detection and tracking of the human body using simple 2D models and uses it for recognizing ASL signs [8]. Other than color cues, motion is applied for signer localization in [5]. However, these approaches require the signer to wear specialized gloves and the back-

ground color is restricted, so these systems do not provide excellent means of human-computer interaction. To overcome the limitations of the individual cues, fusion of cues is explored for face localization in video, but not fully exploited for hand localization [6]. We describe a method that combines motion, color and geometric analysis for hand localization and motion trajectory computation. This combination helps to achieve robustness and accuracy in the extraction of motion trajectories, which in turn helps in recognizing complex hand gestures.

To classify extracted pattern as a gesture, several approaches have been proposed in recent years. Schlenzig, Hunter and Jain [7] use Hidden Markov Model (HMM) and a rotation-invariant image representation to recognize visual gestures such as "hello" and "good-bye." HMMs are also utilized by Starner and Pentland [8] to recognize ASL signs, with or without special colored gloves. Darrell and Pentland [3] apply dynamic time warping to model correlation for recognizing hand gestures from video. Recently, Wilson and Bobick [12] extend the standard HMM method of gesture recognition to include a global parametric variation in the output probabilities of the states in order to recognize and interpret parametric gestures. TDNNs [11] have been applied successfully in speech recognition, spelling recognition and forecasting of time series. It has been shown that TDNN achieves lower error rates in phoneme classification than a simple HMM-based recognizer [11]. In this work, we utilize TDNN to recognize gesture motion patterns.

3 Motion Segmentation

In order to capture the dynamic characteristics of hand gestures, we segment an image frame into regions with similar motion. The algorithm processes an image sequence two successive frames at a time. For a pair of frames, (I_t, I_{t+1}), the algorithm identifies regions in each frame comprising the multiscale intraframe structure. Regions at all scales are then matched across frames. Affine transforms re computed for each matched region pair. The affine transform parameters for region at all scales are then used to derive a single motion field which is then segmented to identify the differently moving regions between the two frames. The following sections describe the major steps in the motion segmentation algorithm.

3.1 Multiscale Image Segmentation

Multiscale segmentation is performed using a transform descried in [1] which extracts a hierarchy of regions in each image. The general form of the transform, which maps an image to a family of attraction force fields, is defined by

$$\mathbf{F}(x,y;\sigma_g(x,y),\sigma_s(x,y)) = \int\int_R d_g(\Delta I,\sigma_g(x,y))\cdot d_s(\vec{r},\sigma_s(x,y))\frac{\vec{r}}{||\vec{r}||}dwdv$$

where $R = domain(I(u,v))\backslash\{(x,y)\}$ and $\vec{r} = (v-x)\vec{i} + (w-y)\vec{j}$. The parameter σ_g denotes a homogeneity scale which reflects the homogeneity of a region to which a pixel belongs and σ_s is spatial scale that controls the neighborhood from which the force on the pixel is computed. The homogeneity of two pixels is given by the Euclidean distance between the associated m-dimensional vectors of pixel values (e.g., $m = 3$ for a color image):

$$\Delta I = |I(x,y) - I(v,w)|$$

The spatial scale parameter, σ_s, controls the spatial distance function, $d_s(\cdot)$, and the homogeneity scale parameter, σ_g, controls the homogeneity distance function, $d_g(\cdot)$. One possible form for these functions satisfying criteria discussed in [10] are unnormalized Gaussian

$$\begin{aligned} d_g(\Delta I,\sigma_g) &\sim \sqrt{2\pi\sigma_g^2}N_{\Delta I}(0,\sigma_g^2) \\ d_s(\vec{r},\sigma_s) &\sim \begin{cases} \sqrt{2\pi\sigma_s^2}N_{||\vec{r}||}(0,\sigma_s^2), & ||\vec{r}|| \le 2\sigma_s \\ 0, & ||\vec{r}|| > 2\sigma_s \end{cases} \end{aligned}$$

The force field encodes the region structure in a manner which allows easy extraction. Region boundaries correspond to diverging force vectors in $\mathbf{F}$ and region skeletons correspond to converging force vectors in $\mathbf{F}$. An increase in σ_g causes less homogeneous structures to be encoded and an increase in σ_s causes large structures to be encoded.

The leftmost image in Figure 1 shows a frame from a video sequence. The following three images show the segmented frame with all pixels in regions at three different scales ($\sigma_g = 6, 20, 40$) replaced by their respective average gray values. The extracted region boundaries align well with the perceived boundaries at different scales. To obtain the segmentations, all parameters of the transform are selected automatically, eliminating the need to make *a priori* assumptions about either the geometric or homogeneity characteristics of the structure.

Figure 1: Results of multiscale segmentation

3.2 Region Matching

The matching of motion regions across frames is formulated as a graph matching problem at four different scales where scale refers to the level of detail captured by the image segmentation process. Three partitions of each image are created by slicing through the multiscale pyramid at three preselected values of σ_g. Region partitions from adjacent frames are matched from coarse to fine, with coarser scale matches guiding the finer scale matching. Each partition is represented as a region adjacency graph, within which each region is represented as a node and region adjacencies are represented as edges. Region matching at each scale consists of finding the set of graph transformation operations (edge deletion, edge and node matching, and node merging) of least cost that create an isomorphism between the current graph pair. The cost of matching a pair of regions takes into account their similarity with regard to area, average intensity, expected position as estimated from each region's motion in previous frames, and the spatial relationship of each region with its neighboring regions.

Once the image partitions at the three different homogeneity scales have been matched, matchings are then obtained for the regions in the first frame of the frame pair that were identified by the motion segmentation module using the previous frame pair. The match in the second frame for each of these motion regions is given as the union of the set of finest scale regions that comprise the motion region. This gives a fourth matched pair of image partitions, and is considered to be the coarsest scale set of matches that is utilized in affine estimation. The details of the algorithm can be found in [9].

3.3 Affine Transformation Estimation

For each pair of matched regions, the best affine transformation between them is estimated iteratively. Let R_i^t be the ith region in frame t and its matched region be R_i^{t+1}. Also let the coordinates of the pixels within R_i^t be (x_{ij}^t, y_{ij}^t), with $j = 1 \ldots |R_i^t|$ where $|R_i^t|$ is the cardinality of R_i^t, and the pixel nearest the centroid of R_i^t be $(\bar{x}_i^t, \bar{y}_i^t)$. Each (x_{ij}^t, y_{ij}^t) is mapped by an affine transformation to the point $(\hat{x}_{ij}^t, \hat{y}_{ij}^t)$ according to

$$\begin{pmatrix} x_{ij}^t \\ y_{ij}^t \end{pmatrix} \rightarrow R\left[\mathbf{A_k} \begin{pmatrix} x_{ij}^t - \bar{x}_i^t \\ y_{ij}^t - \bar{y}_i^t \end{pmatrix} + \vec{T}_k + \begin{pmatrix} \bar{x}_i{}^{t+1} \\ \bar{y}_i{}^{t+1} \end{pmatrix} \right] = \begin{pmatrix} \hat{x}_{ij}^t \\ \hat{y}_{ij}^t \end{pmatrix}_k$$

where the subscript k denotes the iteration number, and $R[\cdot]$ denotes a vector operator that rounds each vector component to the nearest integer. The affine transformation comprises a $2x2$ deformation matrix, $\mathbf{A_k}$, and a translation vector, $\vec{T}_k$. By defining the indicator function,

$$\lambda_i^t(x,y) = \begin{cases} 1, (x,y) \in R_i^t \\ 0, else \end{cases}$$

the amount of mismatch is measured as

$$(M_i^t) = \sum_{x,y} |I_t(x,y) - I_{t+1}(\hat{x},\hat{y})| \cdot \left[\lambda_i^t(x,y) + \lambda_i^{t+1}(\hat{x},\hat{y}) - \lambda_i^t(x,y) \cdot \lambda_i^{t+1}(\hat{x},\hat{y}) \right]$$

The affine transformation parameters that minimize M_i^t are estimated iteratively using a local descent criterion [9].

3.4 Motion Field Integration

The computed affine parameters give a motion field at each of the four scales. These motion fields are then combined into a single motion field by taking the coarsest motion field and then performing the following computation recursively at four scales. At each matched region, the image prediction error generated by the current motion field and the motion field at the next finer scale are compared. At any region where the prediction error using the finer scale motion improves by a significant amount, the current motion is replaced by the finer scale motion. The result is a set of "best matched" regions at the coarsest acceptable scales.

3.5 Motion Field Segmentation

The resulting motion field $\vec{M}_{t,t+1}$ is segmented into areas of similar motion. We use a heuristic that considers each pair of best matched regions, R_i^t and R_j^t, which share a common border, and merges them if the following relation is satisfied for all (x_{ik}^t, y_{ik}^t) and (x_{jl}^t, y_{jl}^t) that are spatially adjacent to one another:

$$\frac{||\vec{M}_{t,t+1}(x_{ik}^t, y_{ik}^t) - \vec{M}_{t,t+1}(x_{jl}^t, y_{jl}^t)||}{max(||\vec{M}_{t,t+1}(x_{ik}^t, y_{ik}^t)||, ||\vec{M}_{t,t+1}(x_{jl}^t, y_{jl}^t)||)} < m_{\sigma_g}$$

where m_{σ_g} is a constant less than 1 that determines the degree of motion similarity necessary for the regions to merge.

The segmented motion regions are each represented in $MS_{t,t+1}$ by a different value. Because each of the best matched regions have matches, the matches in frame $t+1$ of the regions in $MS_{t,t+1}$ are known and comprise the coarsest scale regions that are used in the affine estimation module for the next frame pair.

It should be noted that the motion segmentation does not necessarily correspond to the moving objects in the scene because the motion segmentation is done over a single motion field. Nonrigid objects, such as humans, are segmented into multiple, piecewise rigid regions. In addition, fast objects moving at rates less than one pixel per frame cannot be identified. Handling both these situations requires examining the motion field over multiple frames.

Figure 2 shows frames from an image sequence and Figure 3 shows the results of motion segmentation. Different motion regions are displayed with different gray levels. Notice that there are several motion regions within the head and palm regions because these piecewise rigid regions have similar motion.

4 Color and Geometric Analysis

Motion segmentation generates regions that have similar motion. However, only some of these motion regions carry important information for gesture recognition. To recognize the hand gestures considered here, it is sufficient to extract the motion regions of head and palm regions. We use color segmentation to find motion regions that have skin-like color because of their unique chromaticity. Meanwhile, we use CIE LUV color space in order to minimize the dependence on luminance. A look-up table is created based on statistical analysis of the (u, v) values of skin color pixels in training images. A region is classified to have human skin color if most of its pixels fall into the trained LUV skin color cluster. Coupled with motion segmentation, motion regions of skin color can be efficiently extracted from video.

Since the shape of human head and palm can be approximated by ellipses, and the human hand is a thin, rectangular region, motion regions that have skin color are merged until the shape of the merged region is approximately elliptic or rectangular. The parameters of a rectangular shape can be obtained from the bounding box of each region easily. The orientation of an ellipse are calculated by the least moment of inertia. The extents of the major and minor axes of the ellipse are approximated by the extents of the region along the axis directions and the degree of fit of the ellipse is determined by the number of pixels that fall into that shape specified by the computed parameters. That elliptic region which is larger than the rest is viewed as human head and the palm regions are the elliptic regions that are smaller than the head region while the hand is a rectangular area with its size in between. Figure 4 shows the results of color segmentation and geometric analysis on the motion regions.

5 Motion Trajectory

Although motion segmentation generates the affine transformations that capture motion details by matching finest regions, it is sufficient to use the coarser motion trajectories of identified palm regions for gesture recognition.

Affine transformation of palm region in each frame pair is computed based on equations in Section 3.3. These affine transformations are then concatenated to construct the motion trajectory of the palm region. Figure 7 shows the motion trajectories of the palm region from the image sequence "any."

6 Motion Pattern Classification

Since gesture recognition is a pattern recognition problem of spatio-temporal signals and TDNNs have been demonstrated to have been very successful at such tasks, we employ TDNN to classify gestural motion patterns of palm regions. TDNN is a dynamic classification approach in that the network sees only a small window of the motion pattern and this window slides over the input data while the network makes a series of local decisions. These local decisions have to be integrated into a global decision at a later time. In a seminal paper, Waibel et al. [11] demonstrated excellent results for phoneme classification using a TDNN and showed that it achieved lower error rates than those achieved by a simple HMM recognizer.

The design of TDNN is attractive because its compact structure economizes on weights and makes it possible for the network to develop general feature detectors. Also, its hierarchy of delays optimizes these feature detectors by increasing their scope at each layer. Most importantly, its temporal integration at the output layer makes the network shift invariant (i.e. insensitive to the exact positioning of the gesture). Figure 6 shows our TDNN architecture for the experiments, where positive values are shown as black squares and negative values as gray squares. The inputs to our TDNN are vectors of (x, y, v, a) for each of the 50 frames from gesture image sequence, where x, y are positions with respect to the head, and v, a are magnitudes of velocity and acceleration respectively; the outputs are the gesture classes; and the learning mechanism is an error backpropagation learning algorithm. Our experimental results show that motion patterns can be classified by TDNN accurately and efficiently.

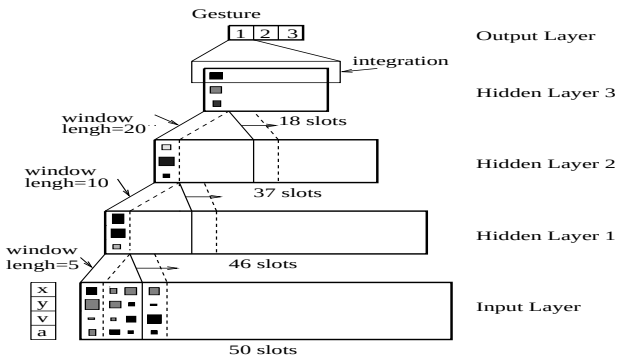

Figure 6: Architecture of TDNN

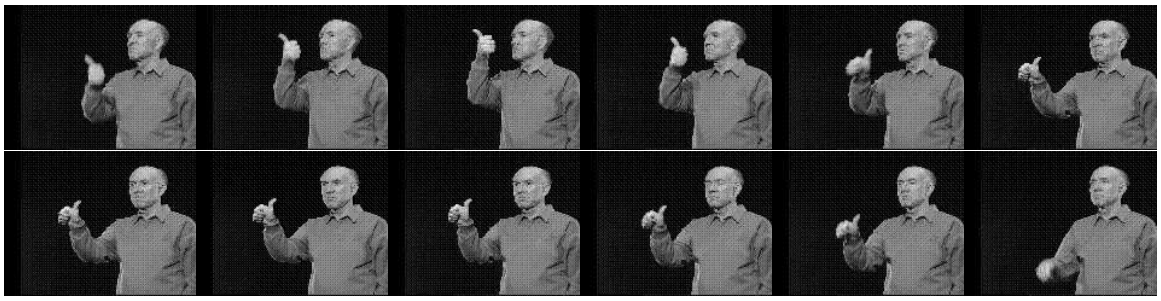

Figure 2: Image sequence of ASL sign "any" (time increases left to right and top to bottom)

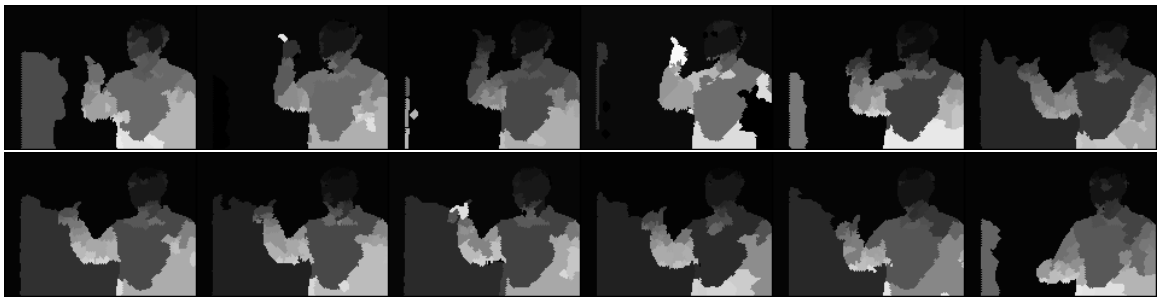

Figure 3: Motion segmentation of the sequence in Figure 2 (time increases left to right and top to bottom)

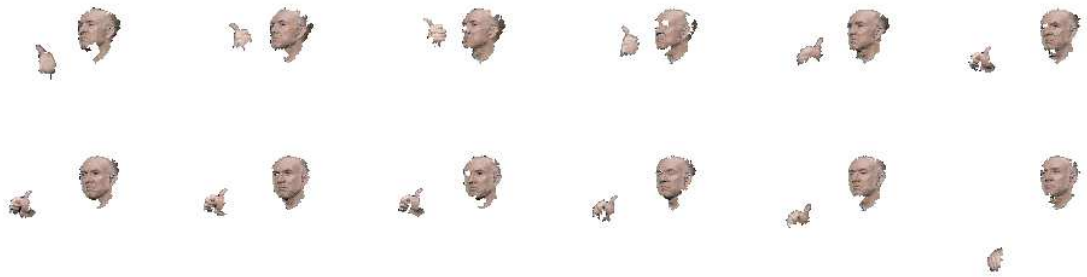

Figure 4: Extracted human head and palm regions in the sequence of Figure 2

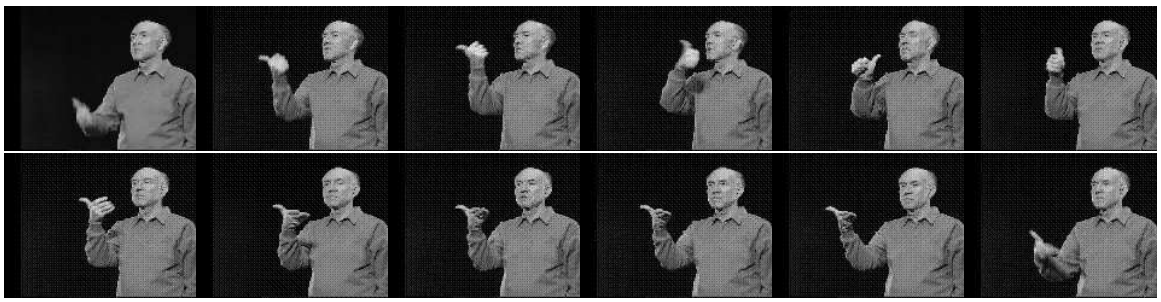

Figure 5: Image sequence of ASL sign "anything" (time increases left to right and top to bottom)

7 Experimental Results

We use a video database of ASL signs for experiments. Each video consists of an ASL sign which lasts about 3 to 5 seconds at 30 frames per second. Figures 2 and 5 show several key frames from a video of ASL sign "any" and "anything." Figure 3 shows the results of motion segmentation on an image sequence "any." Note that the head and palm of the signer consist of several motion regions because motion segmentation is done over a single motion field as discussed previously.

Motion regions with skin color are identified because of their chromatic characteristics. These regions are then merged into palm and head regions based geometric analysis as shown in Figure 4. Affine parameters of matched palm regions are computed, thereby yielding motion trajectories shown in Figure 7. Note that the motion trajectory of palm region matches the movement in the real scene well.

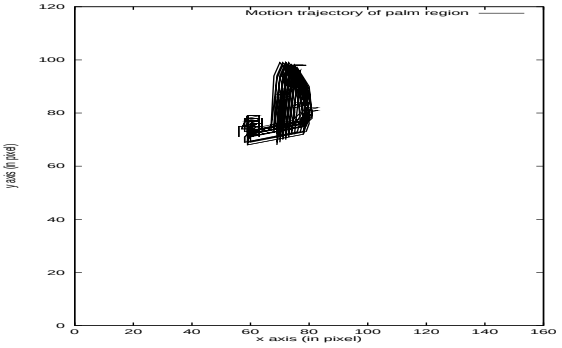

Figure 7: Motion patterns of gesture "any"

For experiments with the classification scheme, we extract motion patterns for these gesture signs (numbered from 1 to 3): "any," "anything" and "accompany." Figures 7 and 8 show the motion patterns of gesture sign 1 and 2. For each sign, we use 80% of the extracted motion patterns of video sequences for training and the rest for testing. Table 1 shows the classification results of the gestures.

Table 1: Experimental results of classification tests

	Gesture 1	Gesture 2	Gesture 3
Recognition rate	100%	96.7%	98.1%
# test patterns	34	30	52

8 Conclusion

We have proposed a method to recognize hand gestures based on motion patterns derived from image sequences. A new system has been developed to extract motion patterns based on motion segmentation, color segmentation, and geometric analysis. These motion

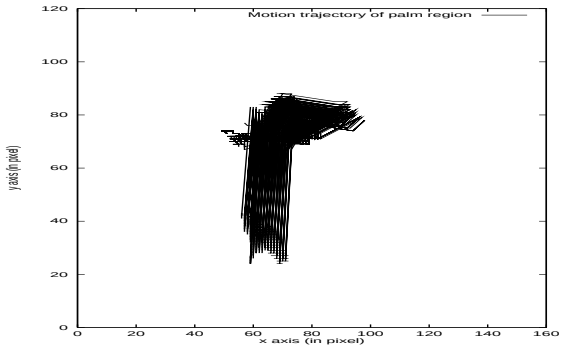

Figure 8: Motion patterns of gesture "anything"

patterns encode the dynamic characteristics of hand gestures and are classified by a time-delay neural network. Our experiments show promising results. Work is ongoing to recognize ASL signs from a database of more than 200 signs and index these patterns for content-based querying.

Acknowledgments

The support of the Office of Naval Research under grant N00014-96-1-0502 is gratefully acknowledged.

References

[1] N. Ahuja. A transform for multiscale image segmentation by integrated edge and region detection. *IEEE PAMI*, 18(12):1211–1235, 1996.

[2] R. Battison. *Lexical Borrowing in American Sign Language*. Linstok Press, Silver Spring, MD, 1978.

[3] T. Darrell, I. Essa, and A. Pentland. Task-specific gesture analysis in real-time using interpolated views. *IEEE PAMI*, 18(12):1236–1242, 1996.

[4] S. S. Fels and G. E. Hinton. Glove-talk: A neural network interface between a data-glove and a speech synthesizer. *IEEE Trans. Neural network*, 4:2–8, January 1993.

[5] W. T. Freeman and C. D. Weissman. Television control by hand gestures. In *Proceedings of the International Workshop on Automatic Face and Gesture Recognition*, pp. 179–183, 1995.

[6] V. I. Pavlovic, R. Sharma, and T. S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE PAMI*, 19(7):677–695, 1997.

[7] J. Schlenzig, E. Hunter, and R. Jain. Vision based hand gesture interpretation using recursive estimation. In *Proceedings of the Twenty-Eighth Asilmoar Conference on Signals, Systems and Computers*, 1994.

[8] T. E. Starner and A. Pentland. Real-time american sign language recognition from video using hidden markov models. In *Proceedings of the International Symposium on Computer Vision*, 1995.

[9] M. Tabb and N. Ahuja. 2-d motion estimation by matching a multiscale set of region primitives. *IEEE PAMI*, 1997. submitted.

[10] M. Tabb and N. Ahuja. Multiscale image segmentation by integrated edge and region detection. *IEEE Trans. Image Processing*, 6(5):642–655, 1997.

[11] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang. Phoneme recognition using time-delay neural networks. *IEEE Trans. ASSP*, 37(3):328–339, 1989.

[12] A. D. Wilson and A. F. Bobick. Recognition and interpretation of parametric gesture. In *Proceedings of the Sixth ICCV*, pp. 329–336, 1998.