Extraction of 2D Motion Trajectories and Its Application to Hand Gesture Recognition

Ming-Hsuan Yang, *Member*, *IEEE*, Narendra Ahuja, *Fellow*, *IEEE*, and Mark Tabb, *Member*, *IEEE*

Abstract—We present an algorithm for extracting and classifying two-dimensional motion in an image sequence based on motion trajectories. First, a multiscale segmentation is performed to generate homogeneous regions in each frame. Regions between consecutive frames are then matched to obtain two-view correspondences. Affine transformations are computed from each pair of corresponding regions to define pixel matches. Pixels matches over consecutive image pairs are concatenated to obtain pixel-level motion trajectories across the image sequence. Motion patterns are learned from the extracted trajectories using a time-delay neural network. We apply the proposed method to recognize 40 hand gestures of American Sign Language. Experimental results show that motion patterns of hand gestures can be extracted and recognized accurately using motion trajectories.

Index Terms—Motion segmentation, motion analysis, motion trajectory, American Sign Language, hand gesture recognition, time-delay neural network.

1 Introduction

Numerous approaches to understanding human motion have been developed since Johansson's seminal work on motion perception using moving light displays [21]. Notwithstanding demonstrated successes in various applications (e.g., cyclic motion analysis, action understanding, and hand gesture recognition), most existing methods have been developed for specific tasks. It is not clear how the same algorithms might be applied to other domains since these methods usually rely on applications-specific knowledge or models to extract motion information [31], [7], [1].

In this paper, we present an algorithm for extracting two-dimensional motion fields of objects across a video sequence and classifying each as one of a set of a priori known classes. Aside from labeling the motion patterns of interest, no prior knowledge is assumed or required for the extraction of motion fields. The algorithm is designed to recognize dynamic visual processes based on spatial, photometric, and temporal characteristics. An application is presented in which a sign (hand gesture) utterance is recognized and interpreted based on, for example, hand location, shape, and motion. The performance of our algorithm was evaluated on the basis of a task involving recognition of 40 complex hand gestures of American Sign Language (ASL) in which each gesture represents one English word.

The algorithm consists of two major steps. First, each image is partitioned into regions using a multiscale segmentation method. Regions between consecutive frames are then matched to obtain two-view correspondences. Affine transformations are computed from each pair of corresponding regions to define pixel matches. Pixel matches over consecutive image pairs are concatenated to obtain pixel-level motion trajectories across the video sequence. Pixels are also grouped based on their two-view motion similarity to obtain a motion-based segmentation of the video sequence. Both the intrinsic properties of the objects represented by image regions and their dynamics represented by the motion trajectories determine whether they comprise an event of interest. Usually, only some of the moving regions correspond to visual phenomena of interest. For example, studies of sign readers suggest humans need few hand details (e.g., shapes and locations) to interpret ASL signs. [27], [33]. Therefore, hand and head regions are extracted from each frame and hand locations are specified with reference to a head region.

To recognize motion patterns from trajectories, we use a time-delay neural network (TDNN) [42], a multilayer feed-forward network that uses shift windows between all layers to represent temporal relationships between events in time. An input vector is organized as a temporal sequence and at any instance only the portion of an input sequence within a time window is fed to the network. Consequently, a TDNN has small receptive fields (i.e., small number of weights to be learned). The time window is shifted and another portion of the input sequence is given to the network until the whole sequence has been scanned. A TDNN is trained using the standard error back-propagation learning algorithm and the output of the network is computed by adding all of these scores over time, followed by applying a nonlinear function (e.g., sigmoid function) to the sum. We adopt TDNN to recognize motion patterns because gestures are spatio-temporal sequences of feature vectors defined along motion trajectories. Our experimental results show that motion patterns can be accurately recognized by a time-delay neural network with extracted motion trajectories.

The remainder of this paper is organized as follows: Section 2 reviews previous works on motion pattern recognition and hand gesture recognition. Motivation of this work and differences from other methods are presented.

- *M.-H. Yang is with Honda Fundamental Research Labs, 800 California St., Mountain Vew, CA 94041. E-mail: myang@hra.com.*
- *N. Ahuja is with the Department of Computer Science and Beckman Institute, University of Illinois at Urbana-Champaign, Urbana, IL 61801. E-mail: ahuja@vision.ai.uiuc.edu.*
- *M. Tabb is with Vexcel Corporation, 4909 Nautilus Court, Boulder, CO 80301. E-mail: tabb@vexcel.com.*

Manuscript received 5 Sept. 2000; revised 28 Aug. 2001; accepted 3 Jan. 2002.
Recommended for acceptance by A. Kundu.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 112811.

Next, Section 3 describes a generic motion segmentation algorithm that divides images into regions of similar motion. This algorithm also tracks regions across frames and estimates the associated affine transformations. As a test bed, we apply the generic motion segmentation algorithm to recognize ASL hand gestures. Toward this end, we propose a method to extract regions of interest from the results of motion segmentation using skin color and geometric cues in Sections 4 and 5. Section 6 presents a method to extract gestural motion trajectories from image sequences and Section 7 gives an architectural view of the time-delay neural network which we apply to recognize gestures. Our experimental results on a set of 40 ASL gestures are detailed in Section 8. Finally, we conclude this paper with remarks and suggestions for future work.

2 Related Work

Since Johansson's seminal work [21] suggesting that human movements can be recognized solely by motion information, methods based on motion profiles and trajectories have been proposed to analyze and understand human movements [31], [7], [1]. In [32], Siskind and Morris conjectured that human event perception does not presuppose object recognition. In other words, they argued that visual event recognition is performed by a visual pathway which is separated from object recognition. To verify this conjecture, they analyzed motion profiles of objects that participate in several spatial motion events. Their tracker used a mixture of color-based and motion-based techniques. To classify visual events, a set of Hidden Markov Models (HMMs) are trained with feature vectors extracted from movies of several visual events. A new observation was classified as being generated by a trained model that assigns the highest likelihood. Experiments on a set of six simple gestures, "pick up," "put down," "push," "pull," "drop," and "throw," demonstrated that visual events can be classified based on motion profiles.

Bobick and Wilson [6] adopted a state-based approach to represent and recognize gestures. For each gesture, a number of samples were used to compute its principal curve [15] which is parameterized by arc length. A by-product of computing the curve was the mapping of each sample point of a gesture example to an arc length along the curve. Next, they used line segments of uniform length to approximate the discretized curve. Each line segment was represented by a vector and all the line segments were grouped into a number of clusters. A state was defined to indicate the cluster to which a line segment belongs and a gesture was then defined by an ordered sequence of states. Similar to [10], they used the Dynamic Time Warping algorithm to match an input state sequence against the previously learned state sequences of gestures. In other words, the recognition procedure evaluated whether the input trajectory passes through the states in a prescribed order. They subsequently extended their method to recognize parameterized gestures (e.g., size gestures) [45].

Vogler and Metaxas described a 3D camera system to recognize ASL signs [40]. The data samples were obtained by using physics-based 3D tracking methods and were used to train HMMs for gesture recognition. At every image frame and for each body part, a subset of the cameras that provide the most informative views for tracking was derived. This time-varying subset was selected based on the visibility of a part and the observability of its predicted motion from a certain camera. Once a set of cameras was selected to track each part, points on the occluding contour to points on the 3D shape model were related based on projective geometry. They used the predicted motion of the model from an extended Kalman filter at each frame to establish point correspondences between occluding contours and the 3D model. The outputs were a set of rotation and translation parameters and were used as inputs to the HMMs for recognition. Subsequently, they proposed a method with HMM extensions to address the scale problem when the vocabulary size is increased [41].

The CONDENSATION algorithm, proposed by Isard and Blake [20], fuses a statistical factored sampling method with a stochastic model to search for a multivariate parameter space that is changing over time. It has been successfully applied to numerous visual tracking and localization problems with success. Objects can be modeled, for example, as a set of parameterized curves, in which case the stochastic model is estimated based on the training sequence. Black and Jepson [5] extended the CONDENSATION algorithm to recognize gestures and facial expressions in which human motions were modeled as temporal trajectories of some estimated parameters (which describe the states of a gesture or an expression) over time.

Many gesture recognition methods used colored gloves or markers to track hand movements. Fels and Hinton used data gloves and Polhemus sensors to extract 3D hand location, velocity, and orientation. Feature vectors were then formed to represent hand gestures and used to train a multilayer neural network for translating hand gestures to synthesized speech in GloveTalk [13] and GloveTalk II [14]. Schlenzig et al. [30] used HMM and a rotation-invariant image representation (based on Zernike moments) to recognize a set of six hand gestures such as "hello" and "good-bye" in a restrictive background. Starner and Pentland used colored gloves to track hand position [35] and, subsequently, adopted a blob representation to track human hands in which a blob is expanded by merging pixels of skin tone using a skin color model [34]. Similarly to [30], they used HMMs to recognize ASL signs [35], [34] in experiments that consist of a 40-word lexicon selected from ASL. One system, which observes the signer from a desk mounted camera, achieved 92 percent word accuracy. Another system, which observes the user from a camera mounted on a cap worn by the signer, achieved 98 percent accuracy [34]. In [22], a method was developed to spot and recognize hand gestures for a human computer interface. Skin color was used to track hand regions and to represent each gesture in terms of the direction of hand movements. A modified HMM was adopted to recognize 10 hand gestures for presentation control commands (e.g., "start," "first," "next," etc.).

It should be noted that numerous researchers have developed methods to recognize static hand gestures, e.g., [47], [9], whereas the focus of this paper is on extraction and recognition of generic motion patterns such as hand gestures. Table 1 summarizes the most relevant works (See also [26], [7] for a review of hand gesture recognition methods). Previous research of hand gesture recognition has been applications-specific and, consequently, difficult to extend to other domains. The main difference between our approach and these methods is that we propose a method to extract motion trajectories from an image sequence without hand drawn templates [20] or distinct trackable markers [5]. Underpinning our method is a generic motion segmentation algorithm that does not impose restrictive assumptions regarding the motions in an image sequence. Motion

TABLE 1
Gesture Recognition Methods

Method	Extraction algorithm	Recognition algorithm	Summary
Schlenzig, Hunter, and Jain [30]	Image moments	Hidden Markov Model	Recognize a set of 6 user defined hand gestures in a restrictive background.
Starner and Pentland [35] [34]	Colored glove or skin tone blobs	Hidden Markov Model	Specifically designed to recognize a set of 40 ASL hand gestures using a color-based tracker.
Fels and Hinton [13] [14]	Data glove	Multilayer neural network	Extract motion information using data gloves and train neural networks for speech synthesizer control.
Siskind and Morris [32]	Colored marker	Hidden Markov model and predicate calculus	Recognize 6 visual events based on predicate calculus in which each predicate is maximally estimated from trained hidden Markov models.
Wilson and Bobick [6]	Data glove and template	Principal curve and Dynamic Time Warping	Recognize various gestures using motion trajectories extracted from magnetic sensors or eigen templates of hand images in restrictive background.
Wilson and Bobick [45]	Data glove	Parameterized hidden Markov Model	Extract 3-D locations of hands using stereo cameras and skin color to recognize a set of 32 size gestures.
Black and Jepson [5]	Colored marker	Modified CONDENSATION algorithm	Extend CONDENSATION algorithm to recognize 6 gestures with colored marker in an office white board environment.
Vogler and Metaxas [40] [41]	3-D camera system with physics model	Hidden Markov Model	Develop parallel HMMs to recognize 22 ASL gestures with a 3-D camera system.
Lee and Lim [22]	Skin color	Hidden Markov Model	Track hand location using skin color to recognize 10 gestures for presentation control.

patterns are learned directly from the extracted motion trajectories. No prior knowledge is assumed or required for the extraction of motion trajectories, although domain knowledge can improve efficiency. We have utilized the proposed segmentation algorithm to understand the motion contents of video sequences with complex and unknown backgrounds such as football game, aerial, video conferencing, and traffic scenes (See Fig. 1 for an example). In this paper, we apply this motion segmentation algorithm to hand gesture recognition and exhibit its advantages. The same method can be easily extended to recognize motion patterns in other domains.

3 Motion Segmentation

We first discuss prior art on motion estimation which motivates our motion segmentation algorithm. An overview of our motion segmentation algorithm, based on [3], [37], is then presented, followed by algorithmic details and examples.

3.1 Motivation

Previous works on 2D motion estimation can be classified as either pixel-based (intensity-based) or feature-based. Pixel-based approaches, often referred to as optical flow methods, assume a direct relationship between object motion and intensity changes within an image sequence. In other words, these methods assume that motion causes variations in intensity and vice versa. Consequently, motion estimation is formulated as an optimization or Bayesian problem where the motion field corresponds to the operator which best accounts for the intensity variations, given certain restrictions. Such methods include algorithms which utilize constraints based on local spatial and temporal derivations [19], [39], [17], [43], and the block-based correlation algorithms [2]. Pixel-based methods generate dense motion estimates. These methods generally perform well in textured areas of the scene, especially when the motion of individual objects is slow relative to their size and the scene consists of only a few moving objects. However, they perform poorly when a scene consists of quickly moving small objects or when the implicit assumption of equivalence between intensity change and motion is violated.

Feature-based methods extract features from images and then match them across image frames to obtain a displacement field. Such features include points defined by local intensity extrema [4], edges [16], [44], corners [25], and regions [28], [36]. These algorithms usually result in sparse motion fields. They rely on single scale segmentation to extract features (e.g.,

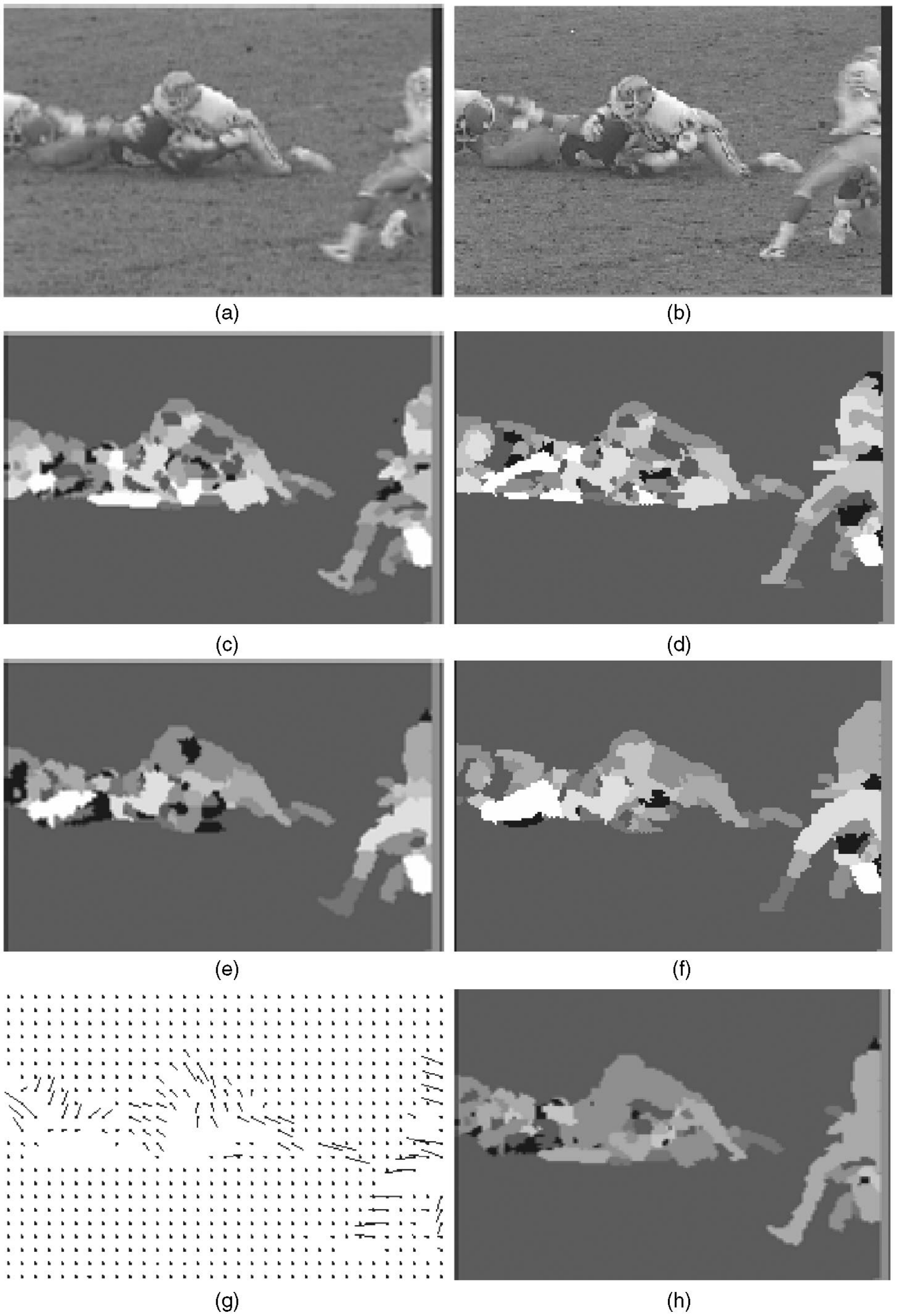


Fig. 1. (a) and (b) Two consecutive frames of a football game sequence. (c) and (d) Matched regions at fine homogeneity scale ($\sigma_g = 9$) and at coarse homogeneity scale ($\sigma_g = 21$) (e) and (f). (g) Estimated motion field shown downsampled. (h) Segmentation of motion field. Pixels of same regions are displayed with the same intensity value, and adjacent regions are displayed with different intensity values.

edges, corners, regions). Consequently, segmentation errors increase the difficulty of finding feature correspondences across frames. Furthermore, previous feature-based methods use fairly simple approaches to obtain correspondences and, thus, may not work well.

Our motion segmentation algorithm estimates a 2D motion field by matching a multiscale set of region primitives [37], thereby differing from previous pixel-based and feature-based methods in several respects. First, region-based motion algorithms (including ours) perform well in situations where pixel-based methods fail. For example, motion information in areas with little intensity variation is contained in the contours of the regions associated with such areas. Our algorithm computes correspondences for such regions and finds the best affine transformation that accounts for the change in contour shape. This transform then represents the estimated motion for the pixels within the region interior as well. Further, region primitives are fairly robust to noise and illumination changes, so differences in shape and position of the region contours across time are generally caused by motion. Second, we use a multiscale set of regions to find their correspondences. The reason is that a multiscale algorithm provides a much richer description of regions available for matching. Both structural changes and noise within a certain area of an image may cause an absence of a match for a region within that area at a particular scale. However, it is often the case that matches can be found within that area at other scales. Consequently, a multiscale method is able to find region correspondences over a larger fraction of an image than most feature-based methods that extract regions at a single scale. Third, previous region-based methods use fairly simple approaches to find region correspondences. In contrast, we formulate region correspondence problem as finding isomorphism between two planar graphs that minimizes a global cost function. This method takes similarity of region size, shape, intensity, and neighboring region into account whereas most methods do not.

3.2 Overview of the Algorithm

To capture dynamic characteristics of objects, we segment an image frame into regions with uniform motion. Our motion segmentation algorithm processes two successive frames of an image sequence 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 (i.e., it identifies structure information within each frame at multiple scales and does not use temporal information among regions). Regions at each scale are then matched across frames. Affine transforms are computed for each matched region pair. The affine transform parameters for region at every scale are used to derive a single motion field, which is then segmented to identify differently moving regions between two frames. The following sections describe the major steps of the motion segmentation algorithm. Also, see [37] for technical discussions and implementation details.

3.3 Multiscale Image Segmentation

Multiscale segmentation is performed using a new transform described in [3] which extracts a hierarchy of regions in each image. In contrast to most segmentation algorithms that consider scale and structure independently, the transform is a nonlinear function that aims to deal with scale and structure simultaneously. Furthermore, the parameters of this transform are selected automatically.

The general form of the transform, which maps an image to a family of attraction force fields, is defined by

$$\begin{aligned} &F(x,y;\sigma_g(x,y),\sigma_s(x,y)) = \\ &\int\int_R d_g(\Delta I,\sigma_g(x,y))\cdot d_s(\vec{r},\sigma_s(x,y))\frac{\vec{r}}{||\vec{r}||}dwdv, \end{aligned} \tag{1}$$

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 a 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 color images):

$$\Delta I = |I(x,y) - I(v,w)|. \tag{2}$$

In this paper, homogeneity is measured by intensity similarity between pixels. 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 that satisfies criteria is unnormalized Gaussian:

$$\begin{aligned} d_g(\Delta I,\sigma_g) &\sim \sqrt{2\pi\sigma_g^2}N_{\Delta_I}\left(0,\sigma_g^2\right) \\ d_s(\vec{r},\sigma_s) &\sim \begin{cases} \sqrt{2\pi\sigma_s^2}N_{||\vec{r}||}\left(0,\sigma_s^2\right), & ||\vec{r}|| \le 2\sigma_s \\ 0, & ||\vec{r}|| > 2\sigma_s. \end{cases} \end{aligned} \tag{3}$$

The transform computes, at each pixel $I(x,y)$, a vector sum of pairwise affinities between $I(x,y)$ and all other pixels. The resultant vector at $I(x,y)$ defines both the direction and magnitude of attraction experienced by the pixel from the rest of the image.

The force field encodes region structures in a manner which allows easy extraction. Consider a region whose boundary is given by a closed curve V, where ∇V is the outward normal of V. Let F^- denote the field immediately on the interior of V and F^+ denote the field immediately on the exterior, V satisfies two relations (from the property of contracting flow, i.e., inward force vectors),

$$\nabla V \cdot F^- \le 0, \quad \nabla V \cdot F^+ \ge 0 \tag{4}$$

since every point on a boundary curve separates at least two areas of contracting flow. With the above definition of the force field F, pixels are grouped into regions whose boundaries correspond to diverging force vectors in F and region skeletons correspond to converging force vectors in 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 readers are referred to [3], [38] for details of the transform and automatic parameter selections (e.g., σ_g and σ_s) for multiscale image segmentation.

3.4 Region Matching

Matching motion regions across frames is formulated as a graph isomorphism 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 a multiscale pyramid at three preselected values of σ_g. Region partitions from adjacent frames are matched from coarse to fine scales, with coarser scale matches guiding the finer scale matching. Each partition is represented by a region adjacency graph within which each region is denoted by a node and region adjacencies are denoted by 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 [18]. 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 spatial relationship of each region with its neighboring regions.

Once image partitions have been matched at three different homogeneity scales, matchings are then obtained for regions in the first frame of a frame pair that were identified by the motion segmentation module using a 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. See [37] for details of the region matching method.

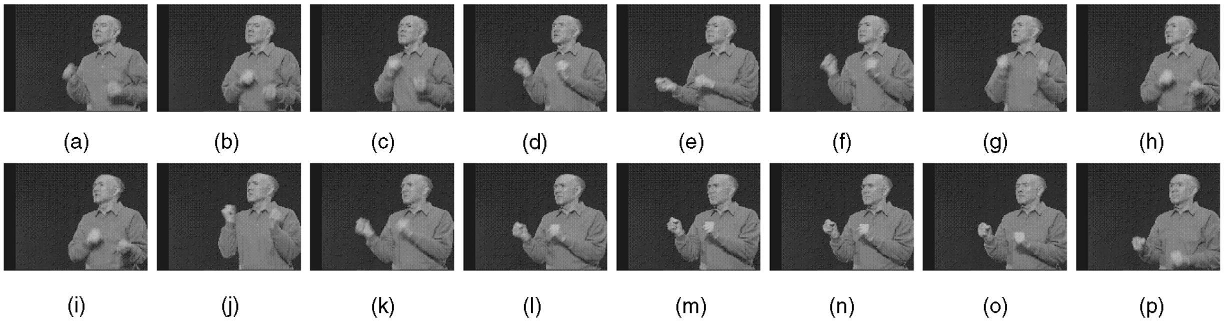


Fig. 2. Image sequence of ASL sign "cheerleader." This gesture contains complex movements of both hands. (a) Frame 14. (b) Frame 16. (c) Frame 19. (d) Frame 22. (e) Frame 25. (f) Frame 29. (g) Frame 31. (h) Frame 34. (i) Frame 35. (j) Frame 37. (k) Frame 40. (l) Frame 44. (m) Frame 46. (n) Frame 49. (o) Frame 52. (p) Frame 55.

3.5 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 $j = 1 \dots |R_i^t|$, 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

$$\begin{pmatrix} x_{ij}^t \\ y_{ij}^t \end{pmatrix} \rightarrow R\left[A_k \begin{pmatrix} x_{ij}^t - \bar{x}_i^t \\ y_{ij}^t - \bar{y}_i^t \end{pmatrix} + \vec{T}_k + \begin{pmatrix} \bar{x}_i^{t+1} \\ \bar{y}_i^{t+1} \end{pmatrix} \right] = \begin{pmatrix} \hat{x}_{ij}^t \\ \hat{y}_{ij}^t \end{pmatrix}_k, \tag{5}$$

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 2×2 deformation matrix, A_k, and a translation vector, $\vec{T}_k$. By defining the indicator function,

$$\lambda_i^t(x,y) = \begin{cases} 1, & (x,y) \in R_i^t \\ 0, & \text{else} \end{cases} \tag{6}$$

the amount of mismatch is measured by

$$\begin{gathered} \left(M_i^t\right) = \textstyle\sum_{x,y} |I_t(x,y) - I_{t+1}(\hat{x},\hat{y})| \cdot \\ \left[\lambda_i^t(x,y) + \lambda_i^{t+1}(\hat{x},\hat{y}) - \lambda_i^t(x,y) \cdot \lambda_i^{t+1}(\hat{x},\hat{y}) \right]. \end{gathered} \tag{7}$$

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

3.6 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 computations recursively at four scales. At each matched region, the image prediction error generated by the current motion field and the motion field at 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.

3.7 Motion Field Segmentation

The resulting motion field $\vec{M}_{t,t+1}$ is segmented into areas of uniform motion, denoted by $MS_{t,t+1}$. 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:

$$\frac{\|\vec{M}_{t,t+1}(x_{ik}^t, y_{ik}^t) - \vec{M}_{t,t+1}(x_{jl}^t, y_{jl}^t)\|}{\max(\|\vec{M}_{t,t+1}(x_{ik}^t, y_{ik}^t)\|, \|\vec{M}_{t,t+1}(x_{jl}^t, y_{jl}^t)\|)} < m_{\sigma_g}, \tag{8}$$

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

Each segmented motion region in $MS_{t,t+1}$ is represented 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.

Note that regions generated by motion segmentation do not necessarily correspond to the moving objects in the scene because motion segmentation is done over a single motion field. Non-rigid 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 a motion field over multiple frames.

3.8 Examples

We applied the motion segmentation algorithm to extract motion contents in image sequences. Figs. 1a and 1b displays the first two frames of a football game sequence. The results from region matching at a finer homogeneity scale ($\sigma_g = 9$) are shown in Figs. 1c and 1d and the results at a coarser scale ($\sigma_g = 21$) are shown in Figs. 1e and 1f. The regions in a given matched set are displayed with the same intensity value, and the neighboring regions are displayed with different intensity values. Any unmatched region is displayed with black pixels. The resulting motion field after integrating all the motion fields at different scales is shown in Fig. 1g. The segmentation results of the motion field in Fig. 1g are shown in Fig. 1h where the segmented regions of the football, hands, and feet perceptually match the motions well.

We also applied the motion segmentation algorithm to hand gesture recognition. Fig. 2 shows frames from an image sequence of a complex ASL sign "cheerleader" and Fig. 3 shows results of motion segmentation, where different motion regions are displayed with different intensity values. Notice that there are several motion regions within the head and hand areas because these piecewise rigid regions have uniform motion. Another image sequence of ASL sign "any" is shown in Fig. 6, and the results of motion segmentation are displayed in Fig. 7.

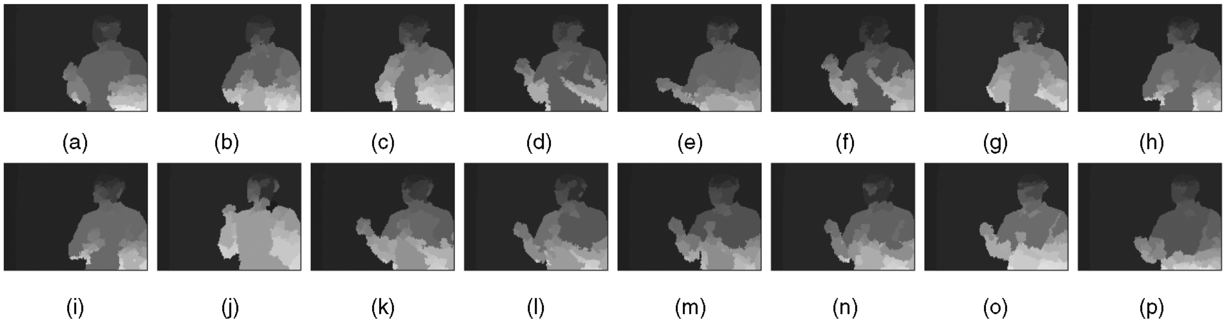


Fig. 3. Motion segmentation of the "cheerleader" image sequence (pixels of the same motion region are displayed with same intensity value and different regions are displayed with different intensity values). (a) Frame 14. (b) Frame 16. (c) Frame 19. (d) Frame 22. (e) Frame 25. (f) Frame 29. (g) Frame 31. (h) Frame 34. (i) Frame 35. (j) Frame 37. (k) Frame 40. (l) Frame 44. (m) Frame 46. (n) Frame 49. (o) Frame 52. (p) Frame 55.

4 Skin Color Model

Motion segmentation generates regions that have uniform motion. In general, only some of these regions are of importance for motion pattern recognition. For example, our generic motion segmentation generates numerous regions, in image sequences shown in Figs. 1, 3, and 7, but clearly most regions (e.g., background and torso areas) have no relevance to motion analysis. Since the regions of interest are application-specific (e.g., hands and legs for human motion analysis in football game scene and hand regions for gesture recognition) and many of them have no relevance to motion pattern recognition, it is futile to analyze all the motion regions in an image sequence. Studies in experimental psychology have also suggested that few hand details are necessary for humans to interpret sign language [27], [33]. To recognize the ASL hand gestures considered in this paper, it is therefore sufficient to extract the motion regions of head and hand regions. The movements of hand regions contain semantic meanings for certain gestures while the head region in a frame is used as a reference point to describe hand locations. Towards this end, we use color and geometric cues to extract these regions for hand gesture recognition.

It should be emphasized that our motion segmentation algorithm is a generic method that does not depend on color information (See Figs. 1, 3, and 7, as well as examples in [37]). Furthermore, the main focus of this paper is on presentation of a generic motion segmentation and its advantages. We use hand gesture recognition as one test bed since the motion contents (i.e., hand movements) to be analyzed in this problem are well defined. Nevertheless, color information is used as a cue to extract the motion regions of interest (i.e., with semantic meanings) for the application considered in this paper. We may use other visual cues, such as texture or shape, to analyze the motion patterns in the football game sequence (See Fig. 1). In contrast to other methods that use color to segment and track regions of interest (for hand gesture recognition, human motion analysis, etc.), our motion segmentation algorithm is generic and can be applied to any image sequence.

Human skin color has been used and proven to be an effective feature in numerous applications. We used a Gaussian mixture [12] to model skin color distribution in CIE LUV color space from a database of 2,447 images which consists of faces of different ethnic groups. The luminance value of each pixel was discarded to minimize the effects of lighting conditions and the parameters of the Gaussian mixture were estimated using an EM algorithm [11], [29]. A motion region was classified to be skin tone if the probabilities of being skin color of most pixels were above a threshold (where each pixel was tested independently). Coupled with motion segmentation, motion regions of skin tone were efficiently extracted from image sequences.

5 Geometric Analysis

Since the shapes of human head and hands can be approximated by ellipses, motion regions of skin tone were merged until the shape of the merged region is approximately elliptic. In contrast to existing work on region merging, our method and application is simpler since color and shape cues can be used to group motion regions of skin tone into elliptic-shaped objects. We used an iterative grow and select merging method which is similar to [23], [24] in spirit. First, we sorted the skin-tone regions based on their size. Among the largest regions, a region MS_p was randomly selected as a seed. Sorted by region size, a neighbor of MS_p was iteratively merged if the goodness fit of an elliptic function of the resulting region did not decrease by a threshold. The process continued to select and merge regions if the number of grouped regions did not exceed a preselected maximum value. A resulting merged region was then considered as a candidate. The whole process repeated several times with different random seeds to generate multiple candidates and the largest merged region was selected. All the parameters were selected empirically for the hand gesture recognition application considered in this paper.

The orientation of an ellipse was estimated from the axes of the least moment of inertia. The extents of the major and minor axes of the ellipse were approximated by the extents of the region along the axis directions, thereby obtaining the ellipse parameters. The largest elliptic region extracted from an image was identified as a human head and the next two smaller elliptic regions were deemed as hand regions.

Fig. 2 shows the image sequence of a complex ASL sign "cheerleader" and Fig. 4 shows the results after applying color and geometric analysis to the segmented motion regions shown in Fig. 3. The results show that head and hand regions can be extracted well by our method. Fig. 9 shows another example where the head and hand regions are extracted from an ASL image sequence "any" shown in Fig. 6.

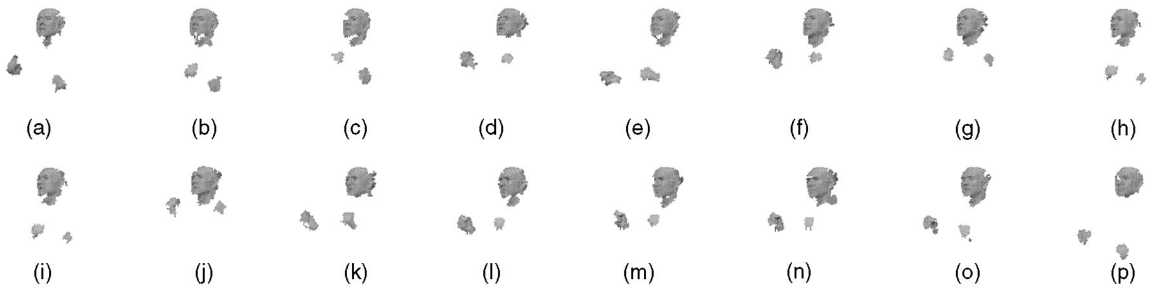


Fig. 4. Extracted head and hand regions from the "cheerleader" image sequence after motion segmentation, color segmentation, and geometric analysis. (a) Frame 14. (b) Frame 16. (c) Frame 19. (d) Frame 22. (e) Frame 25. (f) Frame 29. (g) Frame 31. (h) Frame 34. (i) Frame 35. (j) Frame 37. (k) Frame 40. (l) Frame 44. (m) Frame 46. (n) Frame 49. (o) Frame 52. (p) Frame 55.

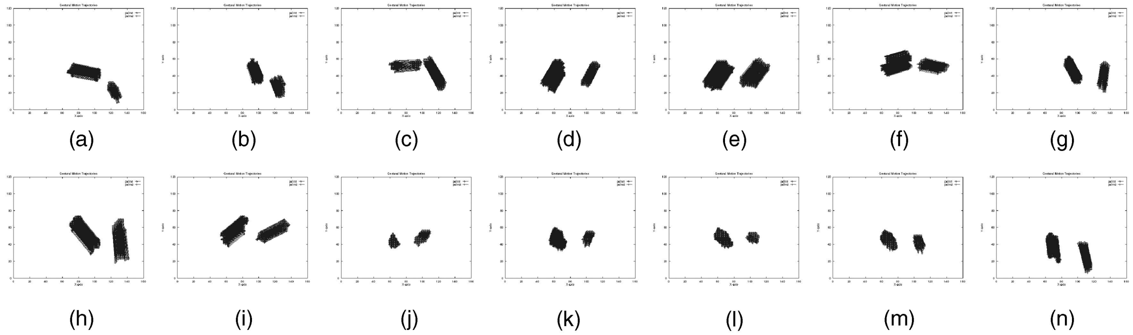


Fig. 5. Extracted gestural motion trajectories from segments of the "cheerleader" image sequence (since all pixel trajectories are plotted, they form a thick blob). The concatenated motion trajectories from each pair of frames constitute motion trajectories for a hand gesture. (a) #14-#16, (b) #16-#19, (c) #19-#22, (d) #22-#25, (e) #25-#29, (f) #29-#31, (g) #31-#34, (h) #35-#37, (i) #37-#40, (j) #40-#44, (k) #44-#46, (l) #46-#49, (m) #49-#52, and (n) #52-#55.

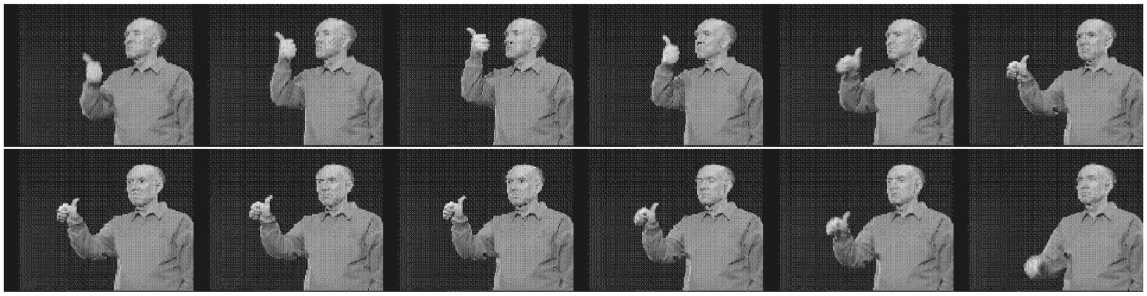


Fig. 6. Image sequence of ASL sign "any" (time increases left to right and top to bottom). This hand gesture shows one sign uttered by the movements of one hand.

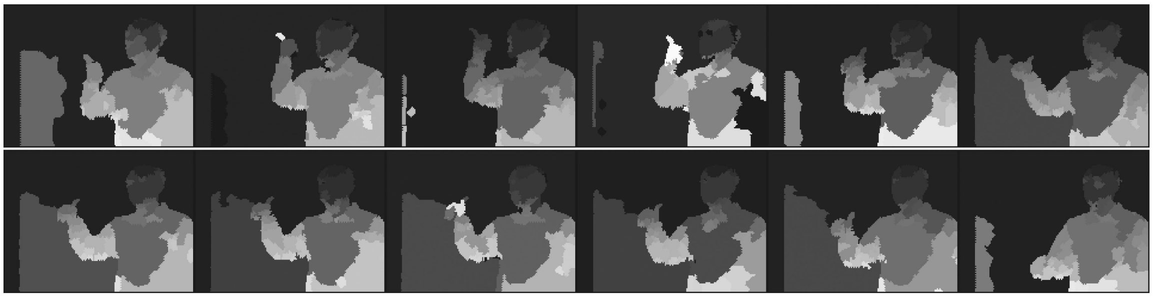
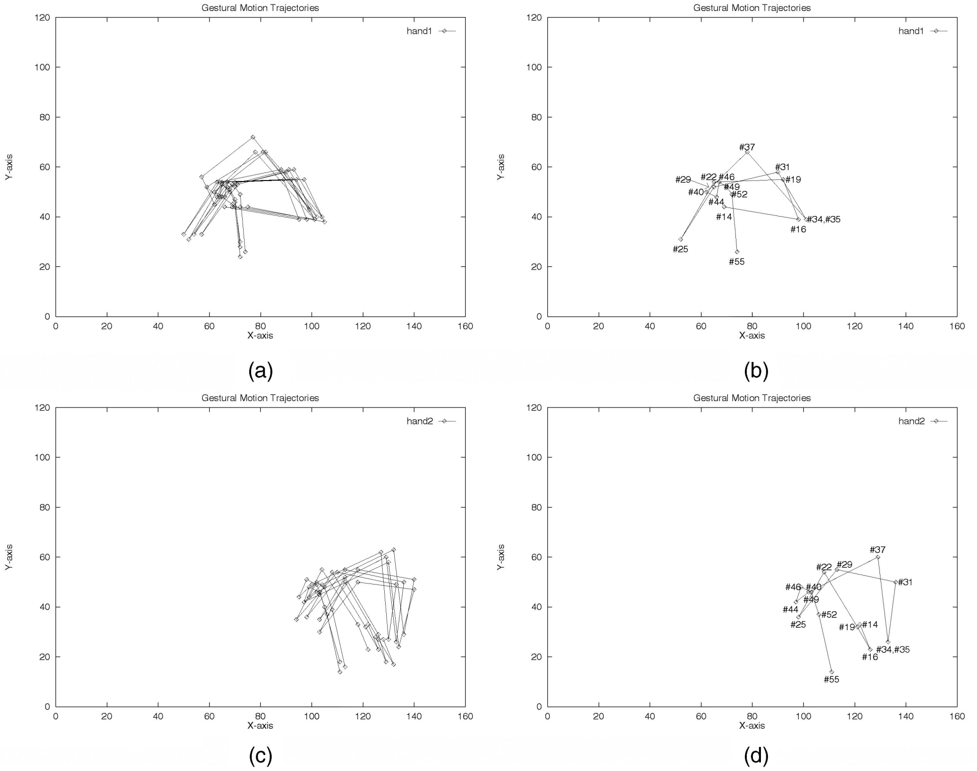


Fig. 7. Motion segmentation of the "any" image sequence shown in Fig. 6 (time increases left to right and top to bottom). Note there are many motion regions in an image frame, however not all of them are useful for hand gesture recognition. Note also that there are several motion regions in the head and hand areas.

6 Motion Trajectories

Although motion segmentation generates affine transformations that capture motion details by matching regions at fine scales, it is sufficient to use coarser motion trajectories of identified hand regions for gesture recognition considered in this paper (which has been suggested by studies of human sign readers [27], [33]).

Affine transformation of a hand region in each frame pair is computed based on equations described in Section 3.5. The affine transformations of successive pairs are then



(a)

(b)

(c)

(d)

Fig. 8. Extracted gestural motion trajectories (subsampled by a factor of 10) of the "cheerleader" image sequence. This gesture shows an example with complex hand movements. (a) A set of pixel motion trajectories of one hand in the "cheerleader" image sequence. (b) One pixel motion trajectory of one hand in the "cheerleader" image sequence. (c) A set of pixel motion trajectories of the other hand in the "cheerleader" image sequence. (d) One pixel motion trajectory of the other hand in the "cheerleader" image sequence.

concatenated to construct motion trajectories of the hand region. Fig. 5 shows such trajectories for a number of frames in the image sequence "cheerleader" where they form thick blobs since all pixel trajectories are displayed together. Fig. 8 shows downsampled (by a factor of 10) motion trajectories of the same image sequence. The results show that extracted motion trajectories perceptually match the motions well. For example, Fig. 5a shows that one group of trajectories moves sideways and the other group of trajectories moves downwards; these trajectories match the hand motions in frames 14 and 16 of Fig. 2 as well. This hand gesture shows one ASL sign that is formed by movements of both hands.

Fig. 9 shows the extracted human head and hand regions from ASL sign "any" image sequence (see Fig. 6). The extracted motion trajectories from ASL "any" and "anything" image sequences (see Figs. 6 and 11) are shown in Figs. 10 and 12. Although these two gestures make circular motions, they do not end at the same location, as shown in the extracted trajectories of Figs. 10 and 12. These gestures show examples in which only one hand motion is required to utter an ASL sign.

7 Recognizing Motion Patterns Using Time-Delay Neural Network

We employed TDNN to classify gestural motion patterns of hand regions since it has been successfully applied to learn spatio-temporal patterns [42]. TDNN is a dynamic classification approach in that the network sees only a small window of the input 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. There are two good properties about TDNN. First, TDNN is able to recognize patterns from poorly

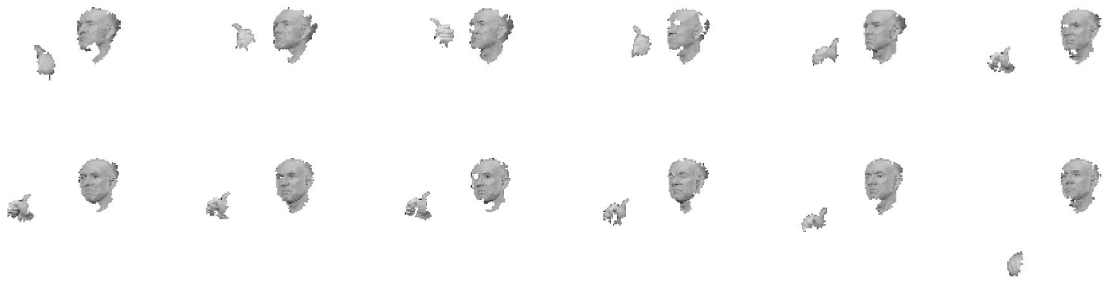


Fig. 9. Extracted head and hand regions in the ASL sequence "any" shown in Fig. 6.

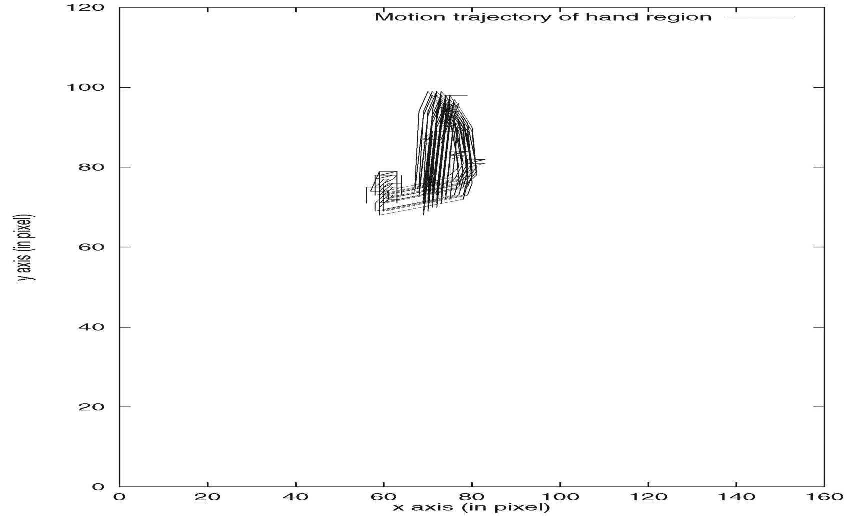


Fig. 10. Gestural motion trajectories of gesture ASL sequence "any" (See Fig. 6 for image frames).

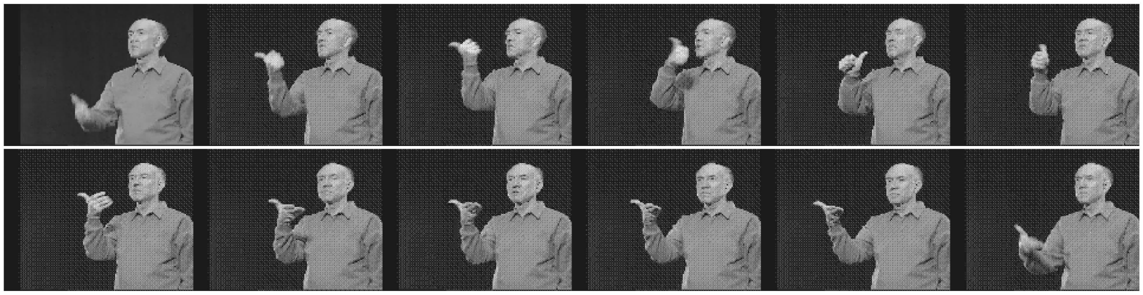


Fig. 11. Image sequence of ASL sign "anything" (time increases left to right and top to bottom). This sequence has hand movements similar to ASL sign "any" (See Fig. 6 for images) but with different circular motions and locations.

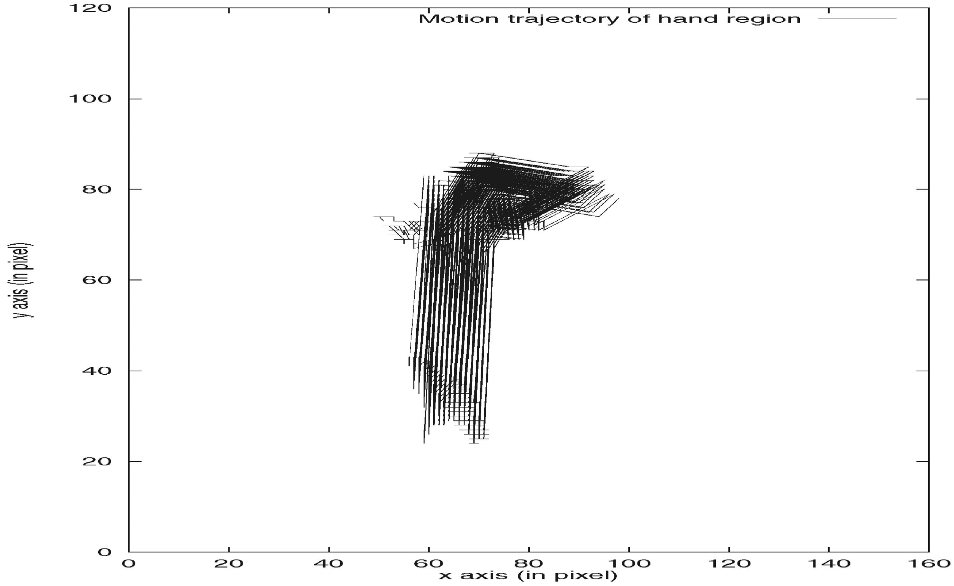


Fig. 12. Gestural motion trajectories of gesture "anything" shown in Fig. 11.

aligned training examples. This is important since examples of gesture image sequence have slight variation in time duration. On one hand, we want to recognize gestures with slight time variation as the same gesture. On the other hand, gestures with the same movements but different execution time should be recognized with different meanings. It has been noted that some gestures have similar movements but have different execution time, e.g., size gestures [45]. Such gestures have different meanings in ASL. TDNNs have been applied successfully to speech recognition where the patterns vary slightly in time [42]. Second, the total number of weights in the network is relatively small since only a small window of

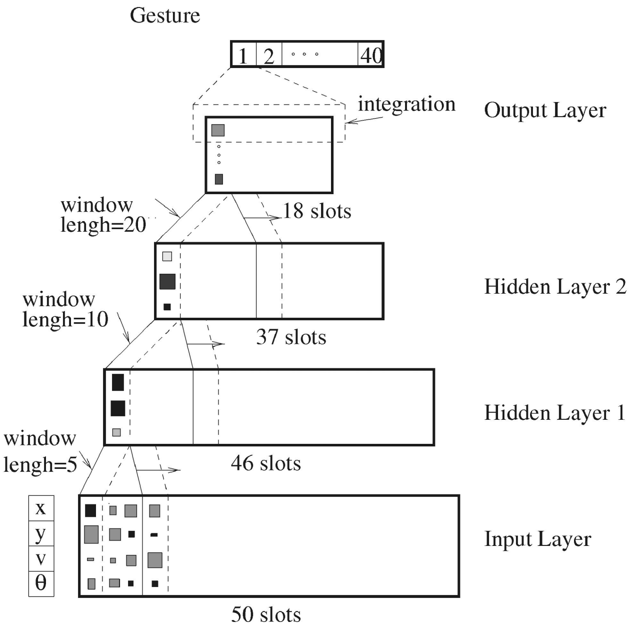


Fig. 13. Architecture of TDNN: A portion of an input vector is fed to the network and propagated to the output layer at any time instance. The results from portions of an input vector are then integrated to output a hand gesture label at a later time.

the input pattern is fed to TDNN at any instance. In other words, TDNN has small receptive fields. This in turns helps reduce training time due to a small number of weights in each receptive field.

In contrast to other methods on hand gesture recognition which use HMMs for recognition, 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). In a seminal work, Waibel et al. [42] demonstrated excellent results for phoneme classification using TDNN and showed it has lower error rates than that achieved by a simple HMM recognizer.

Fig. 13 shows our TDNN architecture (e.g., number of nodes in each layer, number of hidden layers and window size etc.) for the experiments, where the weights are shown using the Hinton diagram. All parameters in the TDNN were selected empirically after numerous experiments on a training set. For each point on a motion trajectory, we formed a vector, $\mathbf{f}_i = (x_i, y_i, v_i, \theta_i)$, where x_i, y_i were positions with respect to the center of a head region at time instance i, and v_i, θ_i were magnitudes and angle of velocity, respectively. All points on a n-point motion trajectory j were stacked next to each other to form a feature matrix for that gesture, i.e., $F_j = (\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n)$. The output for motion trajectory j was the gesture class and the learning mechanism in TDNN was a standard error back-propagation algorithm.

8 Experiments

We first discuss implementation details and experiment setups. The empirical results on a set of 40 ASL hand gestures are then presented and analyzed, followed by some observations.

8.1 Building Skin Color Model

To build a skin color model, we collected 2,447 face images of persons of different ethnic backgrounds. Each image was segmented using the multiscale transform algorithm [3] and the skin color regions were manually selected, thereby resulting in a collection of 9,565,862 skin color pixels represented in RGB color space. To reduce the effects on lighting conditions, each sample was transformed from RGB to CIE LUV color space and the luminance value was discarded. Fig. 14 shows the resulting 2D histogram

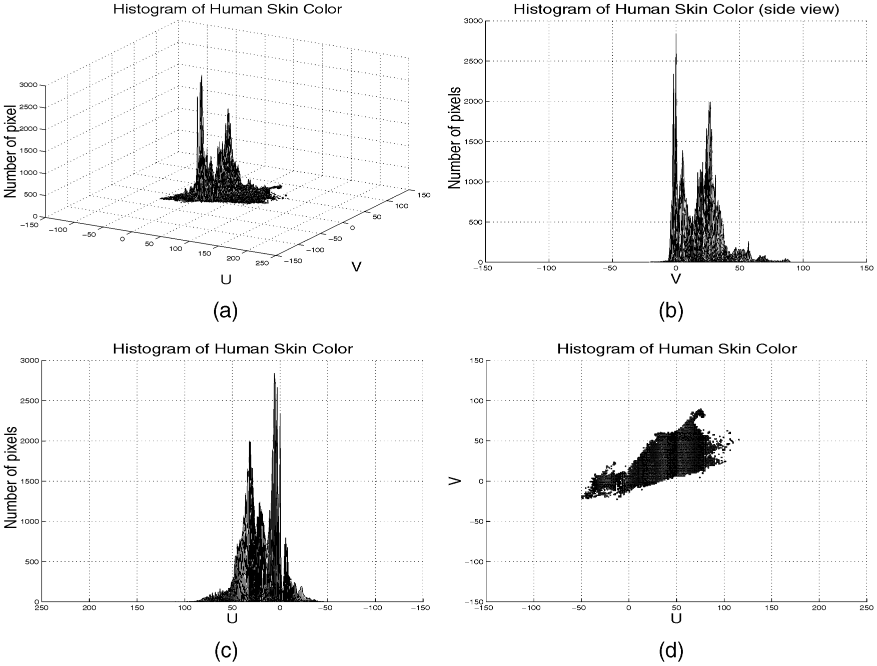


Fig. 14. Histogram of skin color (downsampled by a factor of 10) viewed from different angles.

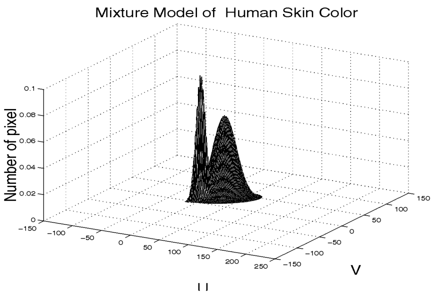


Fig. 15. Estimated density function. The estimated density function was used to determine skin-tone region.

(downsampled by a factor of 10). It is clear that a single Gaussian density function is not sufficient to model the skin color distribution.

8.2 Estimated Density Function

We used an EM algorithm to estimate the parameters of a Gaussian mixture [11]. The samples were initially labeled using k-means clustering where k is empirically set to 2 as an observation of the histogram in Fig. 14. The parameters of a mixture of Gaussian were estimated using the E-step (expectation) and M-step (maximization) iteratively [29]. The estimated density function (Fig. 15) perceptually fits the histogram of the samples (Fig. 14a), making it evident that a finite Gaussian mixture model is more appropriate for estimating a density function of skin color. Statistical tests on the normality and the number of components were performed to justify parameter selections mentioned above [46].

A close inspection of the results shown in Fig. 15 reveals that one hump in the Gaussian mixture corresponds to the skin color distribution of Caucasians and the other one represents the skin color distribution of Asians, Blacks, and other races. The results also validate the use of a Gaussian mixture to model skin color. Nevertheless, further experiments with large and representative data sets will be required to reach firm conclusions.

8.3 Recognizing Hand Gestures

We used a video database of 40 ASL signs for our experiments. (See Table 2.) Each video consisted of an ASL sign lasting about 3 to 5 seconds at 30 frames per second and the image size was 160×120 pixels. Fig. 2 shows one complex ASL gesture from the "cheerleader" sequence. Note that the hand movements consisted of rotation and repeated motions (left-right-left hand movements). An image sequence in the experiments typically had 80 to 140 frames. Discarding the frames in which hands did not appear in the images (i.e., frames in starting and ending phases), each image sequence had an average of 60 frames. In the meanwhile, each image sequence generated an average of 47 motion trajectories of hand areas. Motion regions having skin color were identified by the motion segmentation algorithm and their chromatic characteristics. These regions then were merged into hand and head regions (e.g., Fig. 4) based on the geometric analysis discussed in Section 4. Affine parameters of matched hand regions were computed, which gave pixel motion trajectories for each image pair. By concatenating the trajectories extracted from consecutive image pairs, continuous motion trajectories were generated. Fig. 5 shows the extracted motion trajectories from a number of frames and Fig. 8 shows the trajectories obtained from the whole image sequence. Note that the gestural motion trajectories of both hands corresponded with the movements in the real scene well.

The experiments were conducted with five-fold cross validation. Training of TDNN was performed on the corpus of 80 percent of the extracted dense trajectories from each gesture (each image sequence has an average of 38 trajectories), using an error back-propagation algorithm. The remaining 20 percent of the trajectories were then used for tests. Table 3 summarizes the experimental results. Based on the experiments with 40 ASL gestures, the average recognition rate on the training trajectories was 98.14 percent and the average recognition rate on the unseen test trajectories was 93.42 percent. Since dense motion trajectories were extracted from each image sequence, the recognition rate for each gesture could be improved by applying a simple "voting" scheme (i.e., the majority rule) to the classification result of each individual trajectory. The resulting average recognition rates on the training and test sets for gesture recognition were 99.02 percent and 96.21 percent, respectively. We noted that the errors in the experiments were caused by motion segmentation rather than TDNN. Our training and test videos were recorded at a normal frame rate (30 frames per second) and some images were blurry due to rapid hand movements. As a consequence of image blurs, motion trajectories might not have been generated correctly.

8.4 Discussion

Contrasted to hand gesture recognition methods in the literature [34], [13], [14], [40], [41], [6], [45], our method is able to extract motion trajectories from image sequences with fewest constraints. Most methods in the literature are specifically designed for hand gesture recognition, making them difficult to employ in other applications. These methods usually apply color blobs or special data gloves to extract hand motions (e.g., position, velocity, and angle). Our method first applies a generic motion segmentation algorithm to segment images into regions of uniform motion and then uses color and geometry cues (as an efficient feature method) to extract regions of interest. The underlying motion segmentation algorithm is generic and

TABLE 2
40 ASL Gestures Used in Our Experiments

"Any", "Anything", "Accompany", "Baseball", "Boat", "Cheerleader", "Collision", "Diet", "Doubt", "Experiment", "Explode", "Fast", "Flush", "God", "Fast", "Fish", "Heavy", "Hit", "Infant", "Invite", "Knock-on-a-surface", "Lecture", "Light", "Many", "Mountain", "Neck", "Night", "Obey", "Overnight", "Plan" "Progress", "Rebel", "Request", "Strong", "Superior", "Tall", "Trust", "Upper-Class", "Valley", "Will".

TABLE 3
Recognition Rates

	Using one trajectory in hand areas	Using all trajectories in hand areas
Recognition rate of training set	93.42%	98.14%
Recognition rate of test set	96.21%	99.02%

can be adapted to recognize motion patterns in different applications. It has been shown to be able to segment football game sequence into regions of uniform motion (see Fig. 1). More motion segmentation results of various image sequences with complex backgrounds can be found in [37].

9 Conclusion

We have described an algorithm for extracting motion trajectories and its application to hand gesture recognition. Motion segmentation is performed to generate regions with uniform motion. Moving regions with salient features are then extracted using color and geometric cues. The affine transformations associated with these regions are then concatenated to generate continuous trajectories. These motion trajectories encode dynamic characteristics of hand gestures and are classified by a time-delay neural network. Our experiments demonstrate that hand gestures can be recognized accurately using motion trajectories.

The contributions of this work can be summarized as follows: First, a general method that extracts motion trajectories of hand gestures is presented. This is in contrast to existing methods on gesture recognition that use color histogram trackers [32], [8], [5], magnetic sensors [6], hand drawn templates [20], or stereo cameras [40] to track hand movements. Second, we use a TDNN to recognize gestures based on the extracted trajectories. Using an ensemble of trajectories helps achieve high recognition rates. In this paper, we have emphasized a generic method to extract motion trajectories of hand gestures and to utilize TDNN for recognition. Given that our method is able to extract gestural motion trajectories, we believe that hand gestures can also be recognized by other methods based on HMMs [35], CONDENSATION algorithm [5], or principal curves [6].

References

[1] J.K. Aggarwal and Q. Cai, "Human Motion Analysis: A Review," *Computer Vision and Image Understanding,* vol. 73, no. 3, pp. 428-440, 1999.

[2] J.K. Aggarwal and N. Nandhakumar, "On the Computation of Motion from Sequences of Images: A Review," *Proc. IEEE,* vol. 76, no. 8, pp. 917-935, 1988.

[3] N. Ahuja, "A Transform for Multiscale Image Segmentation by Integrated Edge and Region Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 12, pp. 1211-1235, 1996.

[4] S. Barnard and W. Thompson, "Disparity Analysis of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 2, no. 4, pp. 333-340, 1980.

[5] M.J. Black and A.D. Jepson, "A Probabilistic Framework for Matching Temporal Trajectories: CONDENSATION-Based Recognition of Gesture and Expressions," *Proc. Fifth European Conf. Computer Vision,* pp. 909-924, 1998.

[6] A.F. Bobick and A.D. Wilson, "A State-Based Approach to the Representation and Recognition of Gesture," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 12, pp. 1325-1337, Dec. 1997.

[7] *Computer Vision for Human-Machine Interaction,* R. Cipolla and A. Pentland, eds., Cambridge Univ. Press, 1998.

[8] J.L. Crowley and F. Beard, "Multimodal Tracking of Faces for Video Communications," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 640-645, 1997.

[9] Y. Cui and J. Weng, "A Learning-Based Prediction-and-Verification Segmentation Scheme for Hand Sign Sequence," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, no. 8, pp. 798-804, Aug. 1999.

[10] T. Darrell and A. Pentland, "Space-Time Gestures," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 335-340, 1993.

[11] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc.,* vol. 39, no. 1, pp. 1-38, 1977.

[12] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification.* New York: Wiley-Interscience, 2001.

[13] S.S. Fels and G.E. Hinton, "Glove-Talk: A Neural Network Interface between a Data-Glove and a Speech Synthesizer," *IEEE Trans. Neural Networks,* vol. 4, no. 1, pp. 2-8, Jan. 1993.

[14] S.S. Fels and G.E. Hinton, "Glove-Talk II: A Neural Network Interface which Maps Gestures to Parallel Format Speech Synthesizer Controls," *IEEE Trans. Neural Networks,* vol. 9, no. 1, pp. 205-212, 1997.

[15] T. Hastie and W. Stuetzle, "Principal curves," *J. Am. Statistical Assoc.,* vol. 84, no. 406, pp. 502-516, 1989.

[16] S. Haynes and R. Jain, "Detection of Moving Edges," *Computer Vision, Graphics, and Image Understanding,* vol. 21, no. 3, pp. 345-367, 1980.

[17] F. Heitz and P. Bouthemy, "Multimodal Estimation of Discontinuous Optical Flow Using Markov Random Fields," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 12, pp. 1217-1232, Dec. 1993.

[18] J. Hopcroft and R. Tarjan, "Isomorphism of Planar Graphs," *Complexity of Computer Computations,* R. Miller and J. Thatcher, eds., pp. 131-152, New York: Plenum Press, 1972.

[19] B. Horn and B. Schunck, "Determining Optical Flow," *Artificial Intelligence,* vol. 17, nos. 1-3, pp. 185-203, 1981.

[20] M. Isard and A. Blake, "CONDENSATION—Conditional Density Propagation for Visual Tracking," *Int'l J. Computer Vision,* vol. 29, no. 1, pp. 5-28, 1998.

[21] G. Johansson, "Visual Perception of Biological Motion and a Model for Its Analysis," *Perception and Psychophysics,* vol. 73, no. 2, pp. 201-211, 1973.

[22] H.-K. Lee and J.H. Kim, "An HMM-Based Threshold Model Approach for Gesture Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, no. 10, pp. 961-973, Oct. 1999.

[23] A. Leonardis, A. Gupta, and R. Bajcsy, "Segmentation as the Search for the Best Description of the Image in Terms of Primitives," *Proc. Third IEEE Int'l Conf. Computer Vision,* pp. 121-125, 1990.

[24] D. Marshall, G. Lukacs, and R. Martin, "Robust Segmentation of Primitives from Range Data in the Presence of Geometric Degeneracy," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 3, pp. 304-314, Mar. 2001.

[25] H. Nagel, "Displacement Vectors Derived from Second-Order Intensity Variations in Image Sequences," *Computer Vision, Graphics, and Image Understanding,* vol. 21, no. 1, pp. 85-117, 1983.

[26] V.I. Pavlovic, R. Sharma, and T.S. Huang, "Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 7, pp. 677-695, July 1997.

[27] H. Poizner, U. Bellugi, and V. Lutes-Driscoll, "Perception of American Sign Language in Dynamic Point-Light Displays," *J. Experimental Psychology: Human, Perception and Performance,* vol. 7, no. 2, pp. 430-440, 1981.

[28] K. Price and R. Reddy, "Matching Segments of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 1, no. 1, pp. 110-116, 1979.

[29] R.A. Render and H.F. Walker, "Mixture Densities, Maximum Likelihood and the EM Algorithm," *SIAM Rev.,* vol. 26, no. 2, pp. 195-239, 1984.

[30] J. Schlenzig, E. Hunter, and R. Jain, "Vision Based Hand Gesture Interpretation Using Recursive Estimation," *Proc. 28th Asilomar Conf. Signals, Systems, and Computers,* 1994.

[31] *Motion Based Recognition,* M. Shah and R. Jain, eds. Kluwer Academic Publishers, 1997.

[32] J.M. Siskind and Q. Morris, "A Maximum-Likelihood Approach to Visual Event Classification," *Proc. Fourth European Conf. Computer Vision,* pp. 347-360, 1996.

[33] G. Sperling, M. Landy, Y. Cohen, and M. Pavel, "Intelligible Encoding of ASL Image Sequences at Extremely Information Rates," *Computer Vision, Graphics, and Image Understanding,* vol. 31, no. 2, pp. 335-391, 1985.

[34] T. Starnder, J. Weaver, and A. Pentland, "Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 12, pp. 1371-1375, Dec. 1998.

[35] T.E. Starner and A. Pentland, "Visual Recognition of American Sign Language Using Hidden Markov Models," *Proc. First Int'l Workshop Automatic Face and Gesture Recognition,* pp. 189-194, 1995.

[36] S. Sull and N. Ahuja, "Integrated 3D Analysis and Analysis-Guided Synthesis of Flight Image Sequences," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 16, no. 4, pp. 357-372, Apr. 1994.

[37] M. Tabb, "Multiscale Structure Detection and Its Application to Image Segmentation and Motion Analysis," PhD thesis, Univ. of Illinois at Urbana-Champaign, 1996.

[38] M. Tabb and N. Ahuja, "Multiscale Image Segmentation by Integrated Edge and Region Detection," *IEEE Trans. Image Processing,* vol. 6, no. 5, pp. 642-655, 1997.

[39] A. Verri, F. Girosi, and V. Torre, "Differential Techniques for Optical Flow," *J. Optical Soc. Am.,* vol. 7, no. 5, pp. 912-922, 1990.

[40] C. Vogler and D. Metaxas, "ASL Recognition Based on a Coupling between HMMs and 3D Motion Analysis," *Proc. Sixth IEEE Int'l Conf. Computer Vision,* pp. 363-369, 1998.

[41] C. Vogler and D. Metaxas, "A Framework for Recognizing the Simultaneous Aspects of American Sign Language," *Computer Vision and Image Understanding,* vol. 81, no. 3, pp. 358-384, 2001.

[42] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme Recognition Using Time-Delay Neural Networks," *IEEE Trans. Acoustics, Speech, and Signal Processing,* vol. 37, no. 3, pp. 328-339, 1989.

[43] Y. Weiss, E. Adelson, "A Unified Mixture Framework for Motion Segmentation: Incorporating Spatial Coherence and Estimating the Number of Models," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 321-326, 1996.

[44] J. Weng, N. Ahuja, and T. Huang, "Matching Two Perspective Views," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 14, no. 8, pp. 806-825, Aug. 1992.

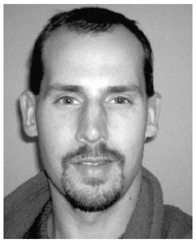
[45] A.D. Wilson and A.F. Bobick, "Parametric Hidden Markov Models for Gesture Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, no. 9, pp. 884-900, Sept. 1999.

[46] M.-H. Yang and N. Ahuja, *Face Detection and Hand Gesture Recognition for Human-Computer Interaction.* Kluwer Academic Publishers, 2001.

[47] M. Zhao, F.K.H. Quek, and X. Wu, "RIEVL: Recursive Induction Learning in Hand Gesture Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 11, pp. 1174-1185, Nov. 1998.

Ming-Hsuan Yang received the PhD degree in computer science from the University of Illinois at Urbana-Champaign in 2000. He studied computer science and power mechanical engineering at the National Tsing-Hua University, Taiwan; computer science and brain theory at the University of Southern California; artificial intelligence and electrical engineering at the University of Texas at Austin. In 1999, he received the Ray Ozzie fellowship for his research work. Since 2000, he has been working on vision problems related to humanoid robots at the Honda Fundamental Research Labs. He has coauthored the book *Face Detection and Gesture Recognition for Human-Computer Interaction* (Kluwer Academic Publishers 2001) and is one of the guest editors for a special issue on face recognition of computer vision and Image understanding, 2003. His research interests include computer vision, computer graphics, pattern recognition, cognitive science, neural computation, and machine learning. He is a member of the IEEE and the IEEE Computer Society.

Narendra Ahuja (F'92) received the BE degree with honors in electronics engineering from the Birla Institute of Technology and Science, Pilani, India, in 1972, the ME degree with distinction in electrical communication engineering from the Indian Institute of Science, Bangalore, India, in 1974, and the PhD degree in computer science from the University of Maryland, College Park, in 1979. From 1974 to 1975, he was the scientific officer in the Department of Electronics, Government of India, New Delhi. From 1975 to 1979, he was at the Computer Vision Laboratory, University of Maryland, College Park. Since 1979, he has been with the University of Illinois at Urbana-Champaign where he is currently the Donald Biggar Willet Professor in the Department of Electrical and Computer Engineering, the Coordinated Science Laboratory and the Beckman Institute. His interests are in computer vision, robotics, image processing, image synthesis, sensors, and parallel algorithms. His current research emphasizes integrated use of multiple image sources of scene information to construct three-dimensional descriptions of scenes, the use of integrated image analysis for realistic image synthesis, parallel architectures and algorithms and special sensors for computer vision, extraction and representation of spatial structure, e.g., in images and video, and use of the results of image analysis for a variety of applications including visual communication, image manipulation, video retrieval, robotics, and scene navigation. He received the 1999 Emanuel R. Piore award from the IEEE, and the 1998 Technology Achievement Award from the International Society for Optical Engineering. He was selected as associate (1998-1999) and Beckman associate (1990-1991) in the University of Illinois Center for Advanced Study. He received the University Scholar Award (1985), the Presidential Young Investigator Award (1984), the National Scholarship (1967-1972), and the President's Merit Award (1966). He has co-authored the books *Pattern Models* (Wiley, 1983), and *Motion and Structure from Image Sequences* (Springer-Verlag, 1992), and co-edited the book *Advances in Image Understanding*, (IEEE Press, 1996). He is a fellow of the IEEE and a member of the IEEE Computer Society, the American Association for Artificial Intelligence, the International Association for Pattern Recognition, Association for Computing Machinery, the American Association for the Advancement of Science, and the International Society for Optical Engineering. He is a member of the Optical Society of America. He is on the editorial boards of the journals *IEEE Transactions Pattern Analysis and Machine Intelligence, Computer Vision, Graphics, and Image Processing, Journal of Mathematical Imaging and Vision, Journal of Pattern Analysis and Applications, International Journal of Imaging Systems and Technology,* and *Journal of Information Science and Technology*, and a guest coeditor of the *Artificial Intelligence* journal's special issue on vision.



Mark Tabb received the BS degree in electrical engineering from Cornell University in 1991, and the MS and PhD degrees in electrical and computer engineering from the University of Illinois at Urbana-Champaign in 1993 and 1996, respectively. From 1991-1995, he was a member of the research staff at the Beckman Institute where his research concerned multiscale feature extraction, image segmentation, and motion estimation from video sequences. From 1996-1997, he was with the Image Understanding Technology Group at Lockheed Martin Astronautics where he helped develop a model-based SAR recognition system. Since joining Vexcel Corporation in 1998 as a senior research scientist, Dr. Tabb has worked on a variety of problems involving radar and signal processing, including 1D and 2D PGA autofocus, and automated feature extraction and land-use classification using interferometric SAR (IFSAR). More recently, he has been one of the major developers of the field of polarimetric SAR interferometry. He is a member of the IEEE and IEEE Computer Society.