Recognizing Hand Gestures Using Motion Trajectories

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 2-view correspondences. Affine transformations are computed from each pair of corresponding regions to define pixel matches. Pixels matches over consecutive images 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 in hand gestures can be extracted and recognized with high recognition rate using motion trajectories.

1 Introduction

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. The algorithm is used to recognize dynamic visual processes based on spatial, photometric and temporal characteristics. An application of the algorithm is in sign language recognition where an utterance is interpreted based on, for example, hand location, shape, and motion. The performance of the algorithm is evaluated on the task of recognizing 40 complex hand gestures of American Sign Language (ASL).

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 2-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 2-view motion similarity to obtain a motion based segmentation of the video sequence. Only some of the moving regions correspond to visual phenomena of interest. 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. For example, it is sufficient to recognize most gestures in ASL in terms of shape and location changes of palm regions. Therefore, palm and head regions are extracted out in each frame and the palm locations are specified with reference to the usually still head regions.

To recognize motion patterns from trajectories, we use a time-delay neural network (TDNN) [11]. TDNN is a multilayer feedforward network that uses time-delays between all layers to represent temporal relationships between events in time. An input vector is organized as a temporal sequence, where only the portion of the input sequence within a time window is fed to the network at one time. The time window is shifted and another portion of the input sequence is given to the network until the whole sequence has been scanned through. The TDNN is trained using standard error backpropagation learning algorithm. The output of the network is computed by adding all of these scores over time, followed by applying a nonlinear function such as sigmoid function to the sum. TDNNs with two hidden layers using sliding input windows over time lead to a relatively small number of trainable parameters. 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 learned by a time-delay neural network with high recognition rate.

2 Related Work

Since Johansson's seminal work [7] that suggests human movements can be recognized solely by motion information, motion profiles and trajectories have been investigated to recognize human motion by several researchers. In [8] Siskind and Morris conjecture that human event perception does not presuppose object recognition. In other words, they think visual event recognition is performed by a visual pathway which is separated from object recognition. To verify the conjecture, they analyze motion profiles of objects that participate in different simple spatial-motion events. Their tracker uses a mixture of color based and motion based techniques. Color based techniques are used to track objects defined by set of colored pixels whose saturation and value are above certain thresholds in each frame. These pixels are then clustered into regions using a histogram based on hue. Moving pixels are extracted from frame differences and divided into clusters based on proximity. Next, each region (generated by color or motion) in each frame is abstracted by an ellipse. Finally, feature vector for each frame is generated by computing the absolute and relative ellipse positions, orientations, velocities and accelerations. To classify visual events, they use a set of Hidden Markov Models (HMMs) which are used as generative models and trained on movies of each visual event represented by a set of feature vectors. After training, a new observation is classified as being generated by the model that assigns the highest likelihood. Experiments on a set of 6 simple gestures, "pick up," "put down," "push," "pull," "drop," and "throw," demonstrate that gestures can be classified based on motion profiles.

Bobick and Wilson [3] adopt a state based approach to represent and recognize gestures. First, many samples of a gesture are used to compute its principal curve [5] which is parameterized by arc length. A by-product of calculating the curve is the mapping of each sample point of a gesture example to an arc length along the curve. Next, they use line segments of uniform length to approximate the discretized curve. Each line segment is represented by a vector and all the line segments are grouped into a number of clusters. A state is defined to indicate the cluster to which a line segment belongs. A gesture is then defined by an ordered sequence of states. The recognition procedure is to evaluate whether input trajectory successfully passes through the states in the prescribed order. Contrasted to their work where each example of a gesture is a single trajectory in space, each gesture in our work is represented by a set of motion trajectories corresponding to the motions of different parts of, say, the palm, instead of a single representative point. Thus, each example of a gesture in our work is represented by a set of motion trajectories. Our experimental results show that an ensemble of trajectories yields better generalization

Bobick and Wilson [3] adopt a state based approach to represent and recognize gestures. First, many samples of a gesture are used to compute its principal curve [5] which is parameterized by arc length. A by-product of calculating the prototype is the mapping of each sample point of a gesture example to an arc length along the prototype curve. Next, they use line segments of uniform length to approximate the discretized curve. These line segments are grouped into a number of clusters. A state is defined to indicate the cluster to which a line segment belongs. A gesture is then defined by a ordered sequence of states. The recognition procedure is to evaluate whether input trajectory successfully passes through the states in the prescribed order. Contrasted to their work where each example of a gesture is a single trajectory in space, each gesture in our work is represented by a set of motion trajectories corresponding to the motions of different parts of, say, the palm, instead of a single representative point. Thus, each example of a gesture in our work is represented by a set of motion trajectories. Our experimental results show that an ensemble of trajectories yield better generalization in learning, and accuracy in recognition.

Recently, Isard and Blake have proposed the CONDENSATION algorithm [6] as a probabilistic method to track curves in visual scenes. This method is a fusion of the statistical factored sampling algorithm with a stochastic model to search a multivariate parameter space that is changing over time. Objects are modeled as a set of parameterized curves and the stochastic model is estimated based on the training sequence. Experiments on the proposed algorithm have been carried to track objects based on their hand drawn templates. Black and Jepson [2] extend this algorithm to recognize gestures and facial expressions in which human motions are modeled as temporal trajectories of some estimated parameters (which describe the states of a gesture) over time. The major 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 [6] or distinct trackable icons [2]. Motion patterns are then learned from the extracted motion trajectories. No prior knowledge is assumed or required for the extraction of motion trajectories, although domain specific knowledge can be applied for efficiency reasons.

3 Motion Segmentation

To capture the dynamic characteristics of objects, we segment an image frame into regions with uniform motion. Our motion segmentation 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 are 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 [1] is unnormalized Gaussian:

$$\begin{array}{rcl} d_g(\Delta I,\sigma_g) & \sim & \sqrt{2\pi\sigma_g^2}N_{\Delta_I}(0,\sigma_g^2) \\ d_s(\vec{r},\sigma_s) & \sim & \left\{ \begin{array}{ll} \sqrt{2\pi\sigma_s^2}N_{||\vec{r}||}(0,\sigma_s^2), & ||\vec{r}|| \leq 2\sigma_s \\ 0, & ||\vec{r}|| > 2\sigma_s \end{array} \right. \end{array}$$

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.

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 scales, 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

$$\begin{array}{l} \left(\begin{array}{c} x_{ij}^t \\ y_{ij}^t \end{array} \right) \rightarrow R\left[\mathbf{A_k} \left(\begin{array}{c} x_{ij}^t - \bar{x}_i^t \\ y_{ij}^t - \bar{y}_i^t \end{array} \right) + \vec{T_k} + \left(\begin{array}{c} \bar{x_i}^{t+1} \\ \bar{y_i}^{t+1} \end{array} \right) \right] \\ \qquad = \left(\begin{array}{c} \hat{x}_{ij}^t \\ \hat{y}_{ij}^t \end{array} \right)_k \end{array}$$

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, $\mathbf{A_k}$, and a translation vector, $\vec{T_k}$. By defining the indicator function,

$$\lambda_i^t(x,y) = \left\{ \begin{array}{l} 1, (x,y) \in R_i^t \\ 0, else \end{array} \right.$$

the amount of mismatch is measured as

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

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

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 uniform 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 1 shows frames from an image sequence of a complex ASL sign called "cheerleader" and Figure 2 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 uniform motion.

4 Color and Geometric Analysis

Motion segmentation generates regions that have uniform motion. However, only some of these motion regions carry important information for motion pattern recognition. To recognize hand gestures considered here, it is sufficient to extract the motion regions of head and palm regions. Towards this end, we use color and geometric information of palm and head regions.

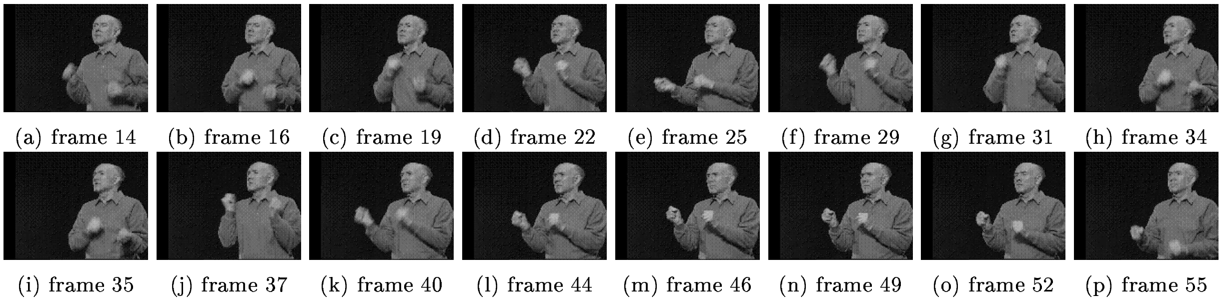
Human skin color has been used and proved to be an effective feature in many applications. We use a Gaussian mixture to model the distribution of skin color pixels from a Michigan database of 2,447 images which consists of human faces from different ethnic groups. We use CIE LUV color space and discard the luminescence value of each pixel to minimize the effects of lighting condition. The parameters in the Gaussian mixture are estimated using an EM algorithm. A motion region is classified to have skin color if most of the pixels have probabilities of being skin color above a threshold. Coupled with motion segmentation, motion regions of skin color can be efficiently extracted from image sequences.

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 is calculated from the axes of 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 thus generate the parameters for the ellipse. The largest elliptic region extracted from an image is identified as human head and the next two smaller elliptic regions are palm regions. Figure 1 shows the image sequence of a complex ASL sign called "cheerleader" and Figure 3 shows the results of color and geometric analysis on the motion regions.

5 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 palm regions for gesture recognition considered in this paper.

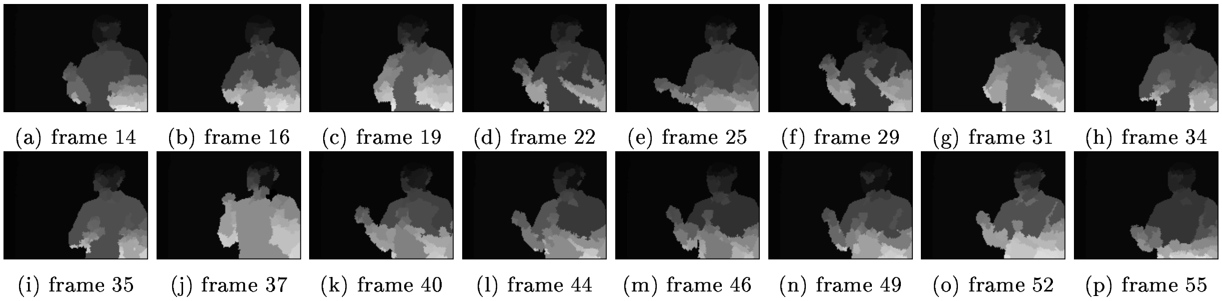
Affine transformation of palm region in each frame pair is computed based on equations in Section 3.3. The affine transformations of successive pairs are then concatenated to construct the motion trajectories of the palm region. Figure 4 shows such trajectories for a



(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

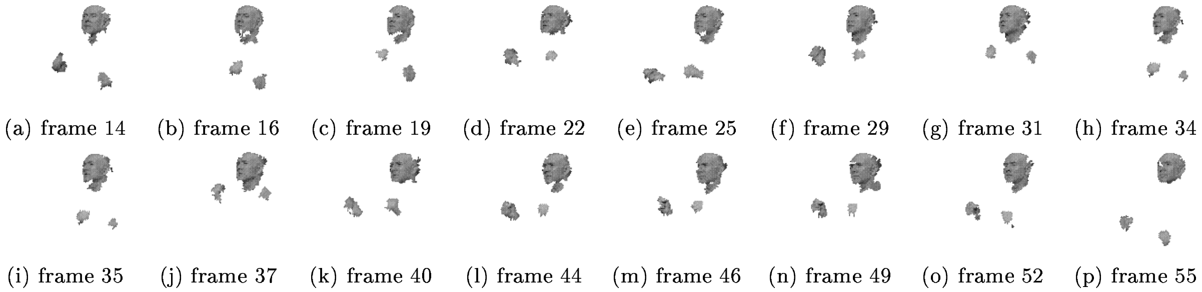
Figure 1: Image sequence of ASL sign "cheerleader"



(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

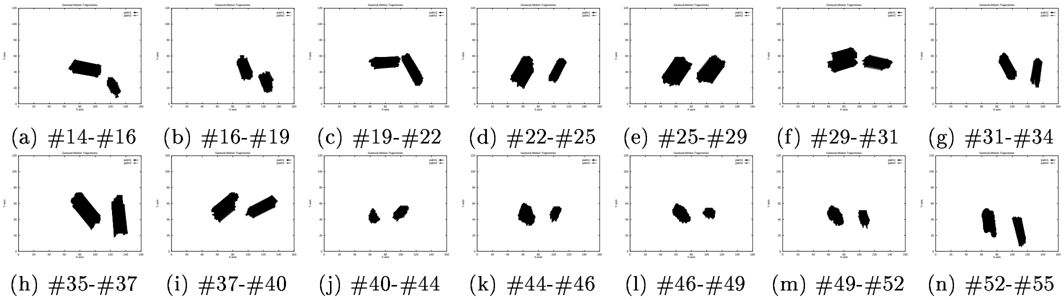
Figure 2: Motion segmentation of the image sequence "cheerleader" (pixels of the same motion region are displayed with same gray level and different regions are displayed with different gray levels)



(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

Figure 3: Extracted head and palm regions from image sequence "cheerleader"

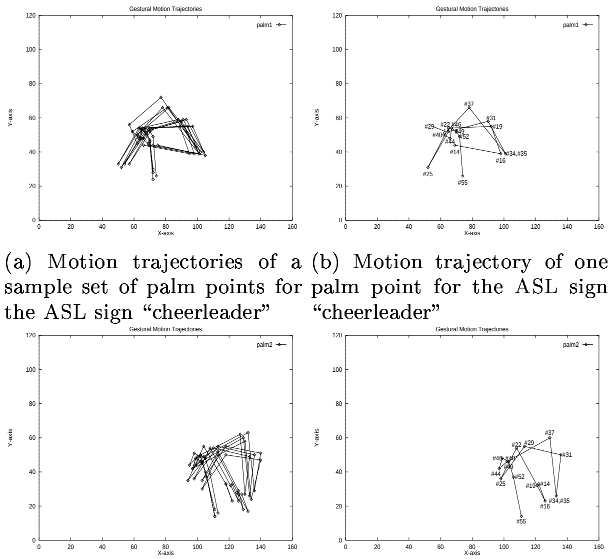


(a) #14-#16 (b) #16-#19 (c) #19-#22 (d) #22-#25 (e) #25-#29 (f) #29-#31 (g) #31-#34

(h) #35-#37 (i) #37-#40 (j) #40-#44 (k) #44-#46 (l) #46-#49 (m) #49-#52 (n) #52-#55

Figure 4: Extracted gestural motion trajectories from segments of ASL sign "cheerleader" (since all pixel trajectories are shown, they form a thick blob)

number of frames in the image sequence "cheerleader." Since all pixel trajectories are shown together, they form a thick blob. Figure 5 shows a 10 to 1 subsampling of the motion trajectories.



(a) Motion trajectories of a sample set of palm points for the ASL sign "cheerleader"

(b) Motion trajectory of one palm point for the ASL sign "cheerleader"

(c) Motion trajectories of a sample set of palm points for the ASL sign "cheerleader"

(d) Motion trajectory of one palm point for the ASL sign "cheerleader"

Figure 5: Extracted gestural motion trajectories (subsampled by a factor of 10) of ASL sign "cheerleader"

6 Motion Pattern Classification

We employ TDNN to classify gestural motion patterns of palm regions since TDNNs have been demonstrated to be very successful in learning spatio-temporal patterns. 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 their seminal work, Waibel et al. [11] demonstrated excellent results for phoneme classification using TDNN and showed that it achieves 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. 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 gray squares and negative values as black squares. The inputs to our TDNN are vectors of (x, y, v, θ) for motion trajectories extracted from a gesture image sequence, where x, y are positions with respect to the center of the head, and v, θ are magnitudes and angle of velocity respectively; the outputs are the gesture classes; and the learning mechanism is error backpropagation.

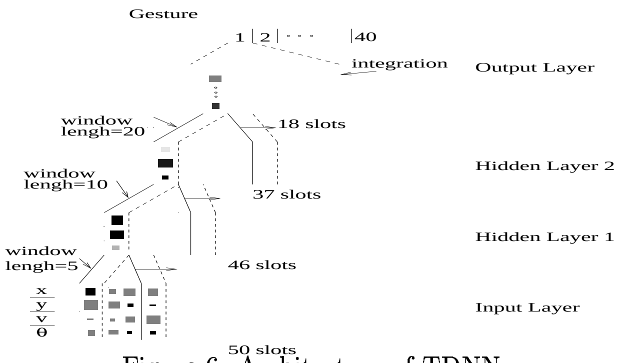


Figure 6: Architecture of TDNN

7 Experiments

We use a video database of 40 ASL signs for experiments. Each video consists of an ASL sign which lasts about 3 to 5 seconds at 30 frames per second with image size of 160 × 120 in Quicktime format. Figure 1 shows one complex ASL gesture from the sequence "cheerleader." Note that the hand movement consists of rotation and repetitions. Each image sequence of the 40 gestures in the experiment has 80 to 120 frames. Discarding the frames in which palms do not appear in the images (i.e. frames in starting and ending phase), each image sequence has about 50 frames. Motion regions with skin color are identified by their chromatic characteristics. These regions are then merged into palm and head regions shown in Figure 3 based on geometric analysis discussed in Section 4. Affine parameters of matched palm regions are computed, which give pixel motion trajectories for each image pair. By concatenating the trajectories for consecutive image pairs, continuous motion trajectories are generated. Figures 4 shows the extracted motion trajectories from a number of frames and Figure 5 shows the trajectories from the whole image sequence. Note that the motion trajectories of palm region match the movement in the real scene well.

Training of TDNN is performed on the corpus of 80% of the extracted dense (38 on the average) trajectories from each gesture, using an error backpropagation algorithm. The rest 20% of the trajectories are then used for testing. Based on the experiments with 40 ASL gestures, the average recognition rate on the training trajectories is 98.14% and the average recognition rate on the unseen test trajectories is

93.42%. Since dense motion trajectories are extracted from each image sequence, the recognition rate for each gesture can be improved by a "voting" scheme (i.e. the majority rules) on the classification result of each individual trajectory. The resulting average recognition rate on the training and testing sets for gesture recognition are 99.02% and 96.21%, respectively.

8 Discussion and Conclusion

We have described an algorithm to extract and recognize motion patterns using trajectories. For concreteness, the experiments have been carried out to recognize hand gestures in ASL. Motion segmentation is performed to generate regions with uniform motion. Moving regions with salient features are then extracted using color and geometric information. The affine transformations associated with these regions are then concatenated to generate continuous trajectories. These motion trajectories encode the dynamic characteristics of hand gestures and are classified by a time-delay neural network. Our experiments demonstrate that hand gestures can be recognized, with high accuracy, using motion trajectories.

The contributions of this work can be summarized as follows. First, a general method that extracts motion trajectories is developed. This is in contrast to much work on gesture recognition that uses color histogram tracker [8] [4] [2], magnetic sensors [3], hand drawn template [6], and stereo [10] to obtain a representation of the gesture. Second, we use a TDNN to recognize gestures based on the extracted trajectories. Using an ensemble of trajectories helps achieve high recognition rates. It would be interesting to compare these recognition rates with those obtained using other recognition methods such as HMM, CONDENSATION algorithm [6] [2] and principal curve [3].

References

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

[2] M. J. Black and A. D. Jepson. A probabilistic framework for matching temporal trajectories: CONDENSATION-based recognition of gesture and expressions. In *Proceedings of European Conference on Computer Vision*, pages 909–924, 1998.

[3] A. F. Bobick and A. D. Wilson. A state-based approach to the representation and recognition of gesture. *IEEE Trans. Pattern. Anal. Mach. Intell.*, 19(12):1325–1337, 1997.

[4] J. L. Crowley and F. Berard. Multi-modal tracking of faces for video communications. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 640–645, 1997.

[5] T. Hastie and W. Stuetzle. Principal curves. *Journal of American Statistical Association*, 84(406):502–516, 1989.

[6] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[7] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception and Psychophysics*, 73(2):201–211, 1973.

[8] J. M. Siskind and Q. Morris. A maximum-likelihood approach to visual event classification. In *Proceedings of the Fourth European Conference on Computer Vision*, pages 347–360, 1996.

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

[10] C. Vogler and D. Metaxas. ASL recognition based on a coupling between HMMs and 3d motion analysis. In *Proceedings of the Sixth International Conference on Computer Vision*, pages 363–369, 1998.

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