Appearance-Based Hand Sign Recognition from Intensity Image Sequences

Yuntao Cui [†] and John Weng[‡]

[†] Siemens Corporate Research
755 College Road East
Princeton, NJ 08540
cui@scr.siemens.com

[‡] Department of Computer Science
Michigan State University
East Lansing, MI 48824
weng@cps.msu.edu

Abstract

In this paper, we present a new approach to recognizing hand signs. In this approach, motion recognition (the hand movement) is tightly coupled with spatial recognition (hand shape). The system uses multiclass, multidimensional discriminant analysis to automatically select the most discriminating linear features for gesture classification. A recursive partition tree approximator is proposed to do classification. This approach combined with our previous work on hand segmentation forms a new framework which addresses the three key aspects of hand sign interpretation: the hand shape, the location, and the movement. The framework has been tested to recognize 28 different hand signs. The experimental results show that the system achieved a 93.2% recognition rate for test sequences that have not been used in the training phase. It is shown that our approach provides better performance than the nearest neighbor classification in the eigen-subspace.

1 Introduction

The ability to interpret hand gestures is essential for computer systems to interact with human users in a natural way. In this paper, we present a new vision-based framework which allows a computer to interact with users through hand signs. Our experimental setup is described as follows. We put a video camera on the top of a computer. The user faces the camera while performing hand signs. We assume an indoor environment where the lighting is fixed.

Since its first known dictionary was printed in 1856 [9], American Sign Language (ASL) is widely used in the deaf community as well as by the handicapped people who are not deaf [6]. The general hand sign interpretation needs a broad range of contextual information, general knowledge, cultural background and linguistic capabilities, which are beyond our capabilities now. In our current research, we select twenty-eight different signs from [7] as shown in Fig. 1 for experiments. These hand signs have following characteristics: 1) they represent a wide variation of hand shapes; 2) they include a wide variation of motion patterns; 3) these hand signs are performed by one hand; 4) recognition of these signs can be done without using contextual information. The gestures which require the hand to perform in a certain environment or to point to a specific object are excluded.

In the linguistic description of ASL, Stokoe used a structural linguistic framework to analyze sign formation [36]. He defined three "aspects" that were combined simultaneously in the formation of a particular sign - what acts, where it acts, and the act. These three aspects translate into building blocks that linguists describe - the hand shape, the location, and the movement. There are two major components in our framework to deal with above three building blocks. We developed a prediction-and-verification scheme to locate hands from complex backgrounds. The spatiotemporal recognition component combines motion understanding (movement) with spatial recognition (hand shape) in an unified framework.

2 Relation to Previous Work

Recently, there has been a significant amount of research on vision-based hand gesture recognition (see [24] for a survey). A vision-based approach acquires visual information of a gesture using a single video camera or a pair of cameras.

The existing approaches typically include two parts, modeling hands and analysis of hand motion. Models of the human hand include three dimensional (3-D) models (e.g., Downton & Drouet [18] and Etoh *et al* [21]: generalized cylindrical model; Kuch & Huang [29]: NURBS-

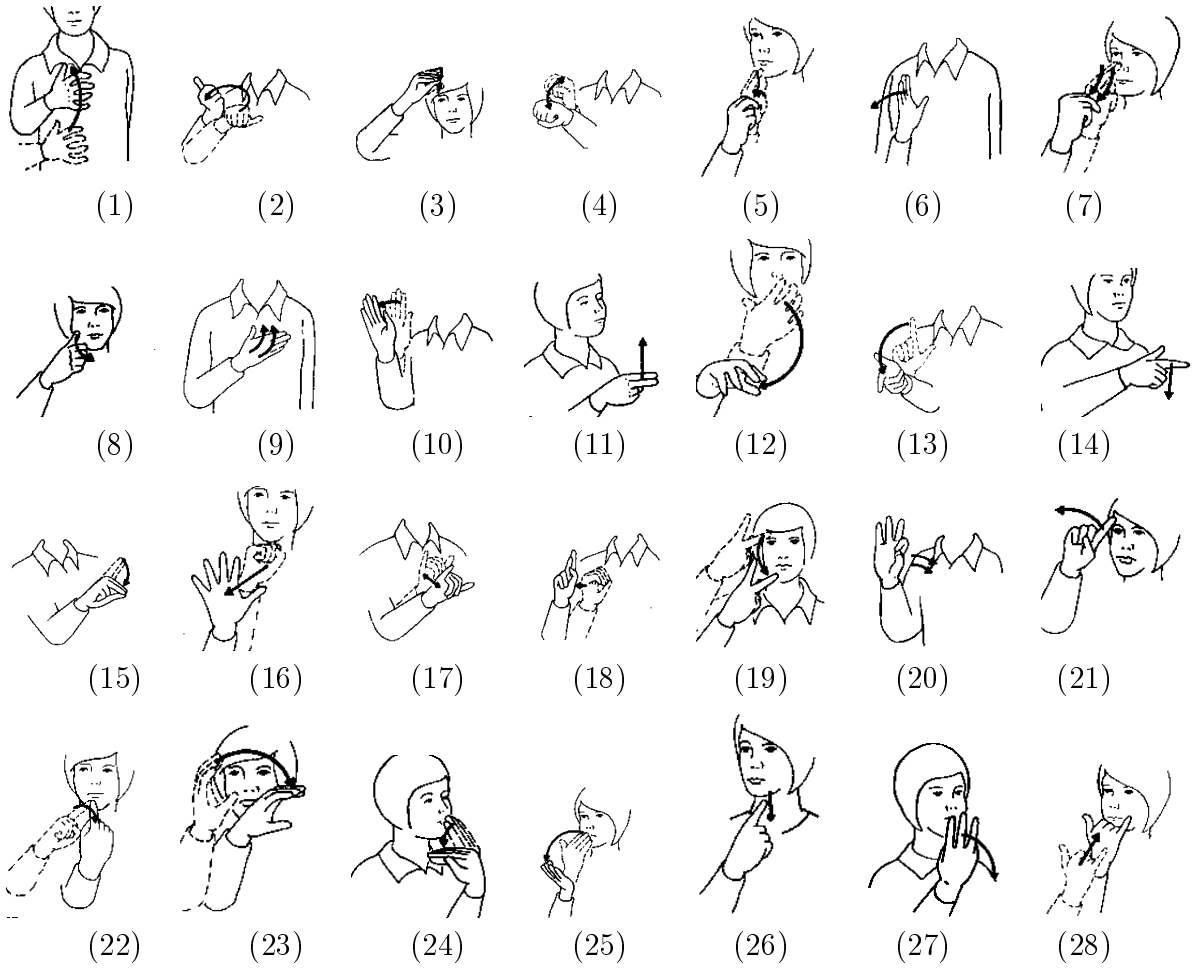


Figure 1: The twenty eight different signs used in the experiment. (1) sign of "angry"; (2) "any"; (3) "boy"; (4) "yes"; (5) "cute"; (6) "fine"; (7) "funny"; (8) "girl"; (9) "happy"; (10) "hi"; (11) "high"; (12) "hot"; (13) "later"; (14) "low"; (15) "no"; (16) "nothing"; (17) "of course"; (18) "ok"; (19) "parent"; (20) "pepper"; (21) "smart"; (22) "sour"; (23) "strange"; (24) "sweet"; (25) "thank you"; (26) "thirsty"; (27) "welcome"; (28) "wrong" (Bornstein and Saulnier 1989).

based hand model), the region-based model (e.g., Darrell & Pentland [16], Bobick & Wilson [5], and Cui *et. al.* [13]), 2-D shape models (e.g., Starner & Pentland [35]: elliptic shape models; Cho & Dunn [11]: line segments shape model; Kervrann & Heitz [26] and Blake & Isard [4]: contour shape model), and fingertip models (e.g., Cipolla *et al* [12] and Davis & Shah [19]). For generality, our hand model is based on 2-D hand appearance, which takes into account both shape and hand texture and allows hand self-occlusion.

Different models for hand require different models for hand motion. A system which uses a 3-D hand model may explicity parameterize 3-D hand kinematics (e.g. [29]), but reliably estimating 3-D motion parameters is very difficult. For a system which uses a two dimensional hand model, the motion can be described as two dimensional rotation, translation and scaling in the image plane [26]. The trajectory of each fingertip can also be used to to represent the motion of a hand [19]. Explicitly modeling both global motion and local motion is possible but estimating the parameters of these explicit models are again very difficult. To use as much information as possible, our global motion model is implicit, represented as a motion vector in what is called attention vector to be explained in the next section. The local motion information is also implicitly coded into the attention vector.

The vision-based approach is one of the most unobtrusive ways which enable users to interact with computers in a natural fashion. However, it faces several challenges. Among them, the task of segmenting a moving hand from sometimes complex backgrounds is perhaps the most difficult one. For this reason, many current systems rely on markers or marked gloves (e.g. [12, 19, 35]). In situations where gesture can be distinguished by the trajectory of a hand, modeling concerns the 2-D trajectory while the hand is modeled as a point (e.g.,[3] [41]). Others methods that do use hand shape more typically assume uniform backgrounds (e.g. [5, 13, 16]). Recently, Moghaddam and Pentland [31] used a maximum likelihood decision rule based on the estimated probability density of the hand and its 2D contour to detect hands from intensity images. The major characteristics of our work lies in 1) dealing with a large number of detailed hand shapes; 2) dealing with complex background, and 3) using shape, motion, and segmentation in a tightly integrated framework. We use a quick indexing scheme to learn a large number of hand shapes. The segmentation scheme in our framework uses subimages from multiple fixations. The search for a valid segmentation is predicted by the training samples and verified by a learning-based interpolation scheme. By coupling motion information with spatial information, we do not separate the hand modeling and the motion

recognition into two different processes, in order to fully use the two types information in an integrated classification stage.

3 Overview of the Approach

A hand gesture is a *spatiotemporal event*. Such a spatiotemporal event involves an object of interest and the motion of the object. The movement of the object (the hand) can be further decomposed into two components: global and local motions. The global motion captures the motion of the entire hand. The local motion could be the motion of fingers or changes of the hand shape, e.g. from palm to fist.

3.1 A three-stage framework

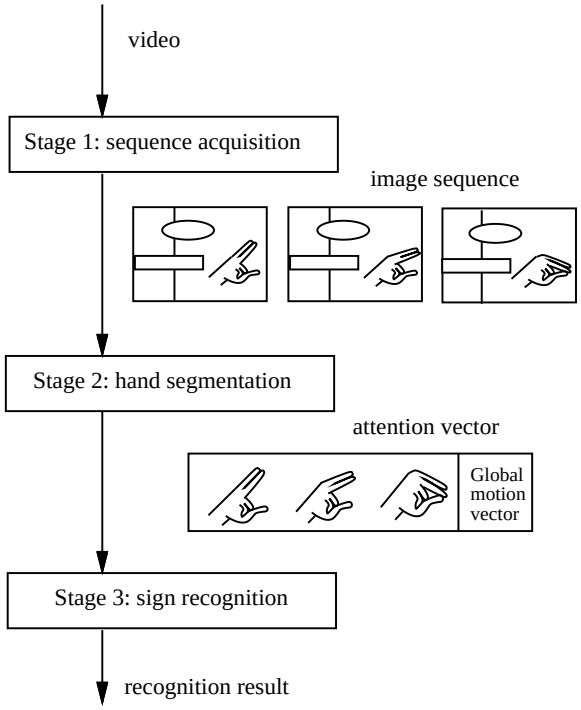


Figure 2: The three-stage framework for spatiotemporal event recognition

In this paper, we propose a three-stage framework for the spatiotemporal event recognition, as illustrated in Fig. 2. The first stage, sequence acquisition, acquires image sequences representing the event. Here, we assume that a hand sign starts and ends with a static hand. Then a simple image difference method is capable of detecting the first and the last images in a hand sign image sequence. We map this temporal window (from start to end) to a standard

temporal length (e.g., 5 frames) to form what is called *motion clip*. In a motion clip, only the temporal dimension is normalized.

The second stage is attention selection and object segmentation. This is one of the major difficulties faced by any vision-based handsign recognition system. At this stage, the system uses motion information to roughly locate the hand in the image frame. A new prediction-and-verification segmentation scheme was used to segment detailed hands from complex background. In the training phase, the system learns the mapping from many local partial views of a hand (that does not include background) to the correct contour of the hand. In the performance phase, this mapping enables the system to predict the contour of a hand from each partial view of a hand detected mainly from motion information. The predicted contour is then used to mask the hand region from the input frame. The resulting hand region is used to verify whether it is a learned hand. This learning-based prediction-and-verification enables the system to segment learned hands of very complex shapes from complex background. Due to the page limit, we are not able to include the segmentation algorithm into this paper. The reader is referred to [14] and [15].

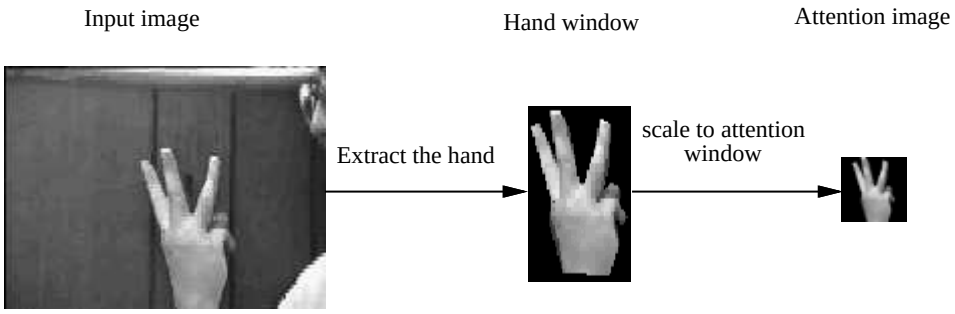


Figure 3: The illustration of constructing attention images.

After the second stage, the object of interest in each image of the sequence is segmented and mapped to an image of a standard fixed size. We call it attention image. Fig. 3 gives an illustration of constructing an attention image. Segmented attention images at different times form a standard *spatiotemporal attention sequence*, in which both temporal and spatial dimensions are normalized. The global motion information of the hand is placed in a global motion vector G, which records the size and the position information in the original image. G is defined as $G = (r^2_{ul} - r^1_{ul}, c^2_{ul} - c^1_{ul}, r^2_{ur} - r^1_{ur}, c^2_{ur} - c^1_{ur}, r^2_{ll} - r^1_{ll}, c^2_{ll} - c^1_{ll}, r^2_{lr} - r^1_{lr}, c^2_{lr} - c^1_{lr}, \cdots, r^p_{ul} - r^1_{ul}, c^p_{ul} - c^1_{ul}, r^p_{ur} - r^1_{ur}, c^p_{ur} - c^1_{ur}, r^p_{ll} - r^1_{ll}, c^p_{ll} - c^1_{ll}, r^p_{lr} - r^1_{lr}, c^p_{lr} - c^1_{lr})$, where r^i_{ul} is row number of the upper-left corner of the ith attention image in the original ith image and c^i_{ul} is column

number of the upper-left corner of the ith attention image in the original ith image. Overall, there are four corners, namely, ul is upper-left, ur is upper-right, ll is lower-left, and lr is lower-right. This vector is necessary because once the object is segmented and mapped to a attention sequence with a standard spatiotemporal size, the global motion information is lost.

For each attention image g with m rows and n columns into an *(mn)*-dimensional vector. For example, the set of image pixels $\{g(i,j) \mid 0 \leq i < m, 0 \leq j < n\}$ can be written as a vector $\mathbf{V} = (v_1, v_2, \cdots, v_d)$ where $v_{mi+j} = g(i,j)$ and $d = mn$. Note that although pixels in an image are lined up to form a 1-D vector $\mathbf{V}$ this way, the 2-D neighborhood information between pixels will be characterized by the scatter matrix of $\mathbf{V}$ to be discussed later. Let p be the standard temporal length and f_i be the hand attention image corresponding to the frame i. Then we create a new vector $\mathbf{X}$, called the *attention vector*, which is concatenation segmented attention images and the global motion vector G,

$$\mathbf{X} = (\mathbf{V}_1, \mathbf{V}_2, ..., \mathbf{V}_p, G). \tag{1}$$

3.2 Processing the attention vector

The third stage, which is the major focus of this paper, is to recognize the spatiotemporal event from the attention vector. The appearance-based mechanism, which has been successfully applied to the pixel-array of intensity images, is extended to the attention vector, which includes not only multiple image frames, but also other information such as the global motion vector G. In other words, we apply powerful statistical tools directly to attention vectors so that they can adaptively decide which components are more important than others and how to weight each component appropriately and automatically for statistical classification. This belongs to what is now called appearance-based approach [27] [38] [32] [37]. This type of approach is characterized by applying statistical tools directly to image vectors to automatically *derive* features instead of relying a human designer to manually define features. Thus, the automatically derived features do not just characterize local features but also global ones, depending on the size of the image window to which the appearance-based approach is applied.

An automatic hand gesture recognition system accepts an input attention vector $\mathbf{X}$ and outputs the recognition result $\mathbf{C}$ which indicates the class to which $\mathbf{X}$ belongs. Thus, a recognition system can be denoted by a function f that maps each element in the space of $\mathbf{X}$ to an element in the space of $\mathbf{C}$. Our objective of constructing a recognition system is equivalent to approximating function $f : S \mapsto C$ by another function $\hat{f} : S \mapsto C$. The error of an approx-

imation can be indicated by certain measure of the error $\hat{f} - f$. One such measure is the L^2 norm:

$$E(\hat{f} - f)^2 = \int_{X \in S} (\hat{f}(\mathbf{X}) - f(\mathbf{X}))^2 dF(\mathbf{X})$$

where $F(\mathbf{X})$ is the probability distribution function $\mathbf{X}$ in S. In other words, $\hat{f}$ can defer a lot from f in parts where $\mathbf{X}$ never occurs, without affecting the error measure. Another measure is the pointwise absolute error $\|\hat{f}(\mathbf{X}) - f(\mathbf{X})\|$ for any point $\mathbf{X}$ in S', where $S' \subset S$ is a subset of S that is of interest to a certain problem.

Of course, f is typically high-dimensional and highly complex. A powerful method of constructing $\hat{f}$ is to use learning. Specifically, a series of cases is acquired as the learning data set:

$$L = \{(\mathbf{X}_i, f(\mathbf{X}_i)) \mid i = 1, 2, \cdots, n\}.$$

Then, the learning task is to construct $\hat{f}$ based on L. For notational convenience, the sample points in L is denoted by $X(L)$:

$$X(L) = \{\mathbf{X}_i \mid i = 1, 2, \cdots, n\}. \tag{2}$$

$X(L)$ should be drawn from the real situation so that the underlying distribution of $X(L)$ is as close to the real distribution as possible.

One popular solution to approximating f is to use the nearest neighbor approximator in the eigenspace based on Karhunen-Loeve projection [30]. We refer the top k ($k \leq 1$) features extracted by Karhunen-Loeve projection as the most expressive features (MEFs) [13] because their power in approximating sample images using a relatively small number of basis vectors [27]. The approach which uses the nearest neighbor approximator in the MEF space has been applied to the problems of face recognition [38]. There are other approximator in the MEF space, such as using manifold for 3D object recognition [32]. The features extracted by Karhunen-Loeve projection, in general, are not the best ones for classification, because the features that describe some major variations in the class are typically irrelevant to how the subclasses are divided as illustrated in Fig. 4.

In this paper, we use the multiclass, multivariate discriminant analysis [25, 39] to select the most discriminating features (MDF's). In the MDF space, we build a recursive partition tree to approximate the function f. The details of the algorithm are in the following section. We also show the convergence property of our new recursive partition tree approximator. Our experiments demonstrated a better performance of the recursive partition tree approximator

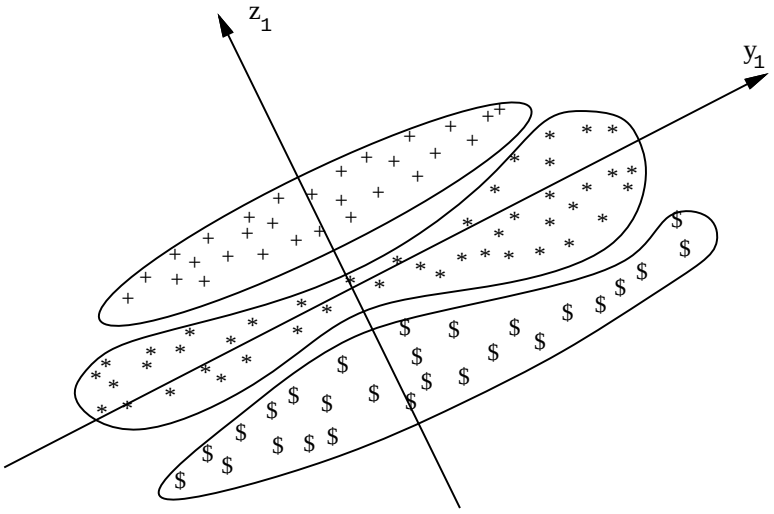


Figure 4: A 2D illustration of the most discriminating features (MDF). The MDF is projection along z_1. The Karhunen-Loeve projection along y_1 can not separate the three classes indicated by three different symbols.

in the MDF space than the nearest neighbor approximator in the MEF subspace computed using the principal component analysis (PCA).

4 Approximation Using Recursive Partition Tree in the MDF Space

The multiclass, multivariate discriminant analysis [25, 39] is used to derive the features in this paper. The discriminant analysis is characterized as follows: one has two types of multivariate observations. The first, called *training samples*, are those whose class identity are known. The second type, referred to as *test samples*, consists of observations for which class identity are unknown and which have to be assigned to one of the class. The discriminant analysis consists of two stages. The first stage, concerned solely with the training samples, is to find a representation of these observations so as to, in some sense, clearly separate the groups. The second stage is concerned with assigning the test samples to one of the specific class.

4.1 The Most Discriminating Features (MDF)

We use the multiclass, multivariate linear discriminant analysis (LDA). Suppose samples of $\mathbf{Y}$ are m-dimensional random vectors from c classes. The ith class has a probability p_i, a mean vector $\mathbf{m}_i$ and a scatter matrix Σ_i. The *within-class scatter matrix* is defined by

$$S_w = \sum_{i=1}^{c} p_i E\{(\mathbf{Y} - \mathbf{m}_i)(\mathbf{Y} - \mathbf{m}_i)^t | \omega_i\} = \sum_{i=1}^{c} p_i \Sigma_i. \tag{3}$$

The *between-class scatter matrix* is

$$S_b = \sum_{i=1}^{c} p_i(\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^t, \tag{4}$$

where the grand mean $\mathbf{m}$ is defined as $\mathbf{m} = E\mathbf{Y} = \sum_{i=1}^{c} p_i M_i$. The *mixture scatter matrix* is the covariance matrix of all the samples regardless of their class assignments:

$$S_m = E\{((\mathbf{Y} - \mathbf{m})(\mathbf{Y} - \mathbf{m})^t\} = S_w + S_b. \tag{5}$$

Suppose we use k-dimensional linear features $\mathbf{Z} = W^t\mathbf{Y}$ where W is an $m \times k$ rectangular matrix whose column vectors are linearly independent. The above mapping represents a linear projection from m-dimensional space to k-dimensional space. The samples $\mathbf{Y}_1, \mathbf{Y}_2, \cdots, \mathbf{Y}_n$ are projected to a corresponding set of samples $\mathbf{Z}_1, \mathbf{Z}_2, \cdots, \mathbf{Z}_n$ whose within-class scatter, and between-class scatter matrices are S_{Z_w} and S_{Z_b}, respectively. For details of computing W^t, the reader is referred to [23]. We call the feature extracted by the above method the most discriminating features (MDFs).

4.2 Curse of dimensionality and the DKL projection

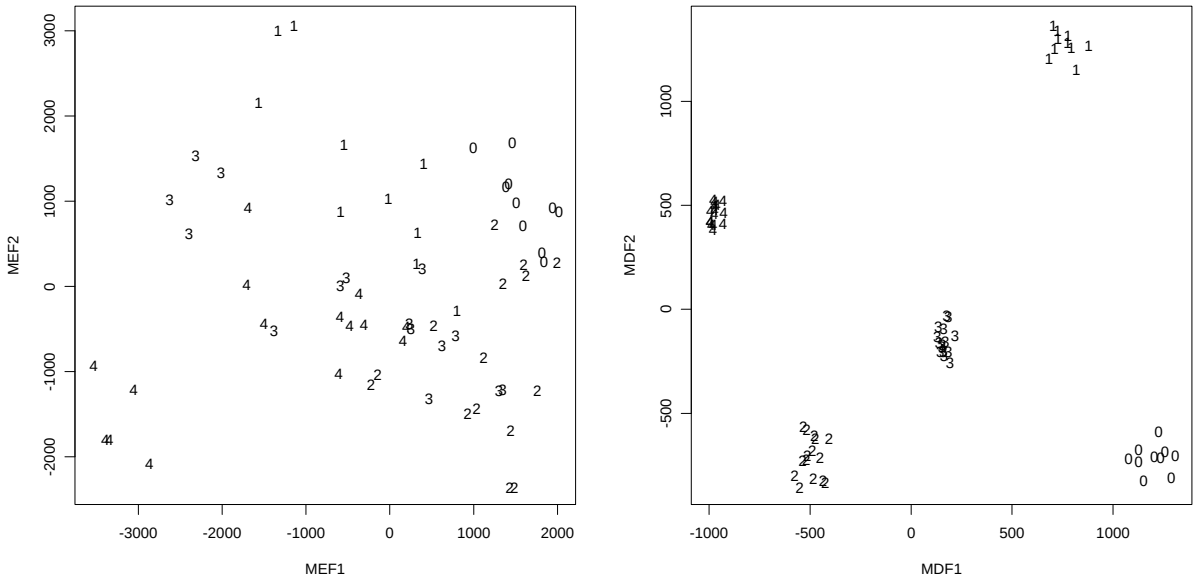
The discriminant analysis procedure breaks down when the within-class scatter matrix S_w becomes degenerate, which is true in our case due to a high dimension of the input image and a much smaller number of training samples. We use the *DKL projection* (short for Discriminant Karhunen-Loeve projection) [40]. The DKL projection consists of two projections, the first is the Karhunen-Loeve projection and the second is LDA projection. In this subspace, which keeps almost all the variance (the number of MEF is such that 95% or more of variance in the original image space is kept), the discriminant analysis can be performed since the degeneracy does not occur. The overall DKL projection can be represented by projection matrix $M^t = W^tV^t$ where V consists of MEF eigenvectors and W consists of MDF eigenvectors in the resulting MEF subspace.

4.3 Why MDF

In this section, we show some experimental results to indicate quantitatively how the MEFs (obtained by the Karhunen-Loeve projection) and the MDF may perform very differently in classifying hand signs.

4.3.1 Clustering effects

We computed MEF's and MDF's, respectively, using 50 sequences (10 for each signs). These signs are obtained from different subjects and the viewing positions are slightly different. Fig. 5 (a) shows the samples in the subspace spanned by the first two MEFs and Fig. 5 (b) shows them in the subspace spanned by the first two MDFs. As clearly shown, in the MEF subspace,



(a) (b)

Figure 5: The difference between MEF and MDF in representing samples. (a) Samples represented in the subspace spanned by the first two MEFs. (b) Samples represented in the subspace spanned by the first two MDFs. The numbers in the plot are the class labels of the samples.

samples from a single class spread out widely and samples of different classes are not far apart. In fact, some samples from different classes mingle together. However, in the MDF subspace, samples of each class are clustered more tightly and samples from different classes are farther apart. Thus in the MDF space, a classification scheme (such as the nearest neighbor rule) works more effectively. Therefore, the MDFs are better in terms of classification of signs.

4.3.2 Intuitive meaning of the MDF

In the MDFs, factors that are not related to classification are discarded or weighted down, which is accomplished by minimizing the within-class scatter; factors that are crucial to clas-

sification are emphasized, which is achieved by maximizing the between-class scatter. In this experiment, we show an example to indicate that the MDFs can capture the important geometric features represented as intensity appearance.

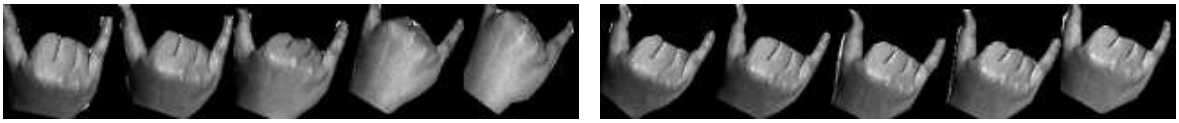
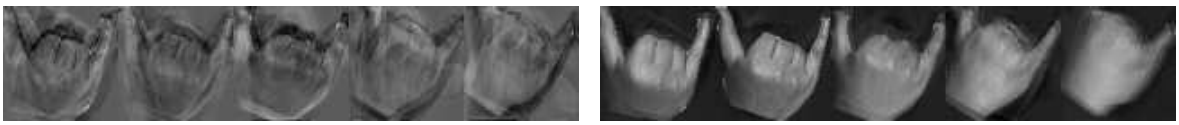


Figure 6: Two sample sequences of sign "of course" (left) and "wrong" (right).

In our gesture vocabulary, the image sequences of two signs: "of course" and "wrong" are visually very similar. Fig. 6 illustrates two sample sequences of the above signs. The nearest neighbor approximator in MEF generally has difficulty to distinguish them, but not the recursive partition tree approximator in the MDF space. Fig. 7 shows the difference between the MEF and the MDF. The left sequence in Fig. 7 is a reconstruction of the sequence "of course" based on the first MDF and the right sequence is a reconstruction of the same sequence using MEFs that retain 95% of the sample variance. We can see that the MEFs are good in terms of preserving the absolute intensity of images. On the other hand, the first MDF captures the feature locations (edges) because it accounts for the major between-sign variation.



(a) (b)

Figure 7: The difference between the MEF and MDF. (a) Reconstruction based on the first MDF. (b) Reconstruction based on MEFs. MDFs characterize patterns of edge locations better than MEFs.

4.4 Recursive partition tree

For a complicated problem, a single-level feature space may not be best for a specific set of classes as illustrated in 8. A distribution of samples from one hand sign obtained under various lighting conditions and viewing directions usually is not linearly separable. In order to handle this situation, we propose a hierarchal structure, represented by a tree. Each node of the tree represents a new MDF space. This tree also drastically reduces the time complexity of finding a good match from a larger number of training samples to approximately logarithmic: $O(\log(n))$ where n is the number of leave nodes.

Our hierarchal structure shares many common characteristics with the well known tree

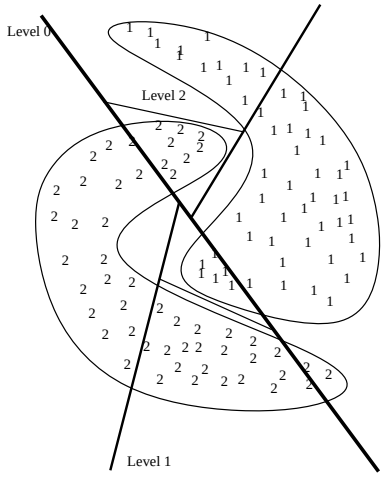


Figure 8: The distributions of class "1" and "2". A single level MDF space can not separate these two classes using a single hyperplane. The hierarchical classification tree partitions regions into unions of simpler regions whose boundaries are piecewise hyperplanes.

classifiers and the regression trees in the mathematics community [8], the hierarchical clustering techniques in the pattern recognition community [20, 25] and the decision trees or induction trees in the machine learning community [33]. The major differences between our tree and those traditional trees are:

1. Our method automatically derives features directly from training images, while all the traditional trees work on a human pre-selected set of features.

2. The traditional trees either (a) at each internal node, search for a partition of the corresponding samples to minimize a cost function (e.g., ID3 [33] and clustering trees [25]), or (b) simply select one of the remaining unused features as the splitter (e.g., the k-d tree). Option (a) results in an exponential complexity that is way too computationally expensive for learning from high-dimensional input like images. Option (b) implies selecting each pixel as a feature, which simply does not work for image inputs (in the statistics literature, it generates what is called a dishonest tree [8]). Our tree directly computes the most discriminating features (MDF), using the Fisher's multi-class, multi-dimensional linear discriminant analysis [20, 23, 39], for recursive space partitioning at each internal node.

4.4.1 Construction of recursive partition tree

The algorithm of constructing a recursive partition tree is illustrated as follows.

Given a set of n training samples $\mathbf{X} = \{\mathbf{X}_1, \cdots, \mathbf{X}_n\}$
if $\mathbf{X}$ belong to more than one classes, **then**
1) Compute the DKL projection matrix.
2) Project $\mathbf{X}_i$ into the DKL space, denoted as $\mathbf{Y}_i$.
3) Select an radius r, such that there are k samples $\{\mathbf{Y}'_1, \cdots, \mathbf{Y}'_k\}$ in $\mathbf{Y}$
where $1 < k < T$ and for any $i \neq j$, $\|\mathbf{Y}'_i - \mathbf{Y}'_j\| > r$;
T is the maximum number of branches;
4) For each $\mathbf{Y}'_i$, we have a subset $C_i = \{\mathbf{X}_m\}$,
the partition cell for $\mathbf{Y}'_i$. That is,
any $j \neq i$, and its corresponding $\mathbf{Y}_m$,
$\|\mathbf{Y}_m - \mathbf{Y}'_j\| > \|\mathbf{Y}_m - \mathbf{Y}'_i\|$.
5) Recursively partition each C_i as above
until C_i consists of samples from a single class.
end if

The graphic description in Fig. 9 gives an simplified but intuitive explanation of the hierarchical structure. The structure is a tree. The root corresponds to the entire space of all the possible inputs. The children of the root partition the space into large cells, as shown by thick lines in Fig. 9. The children of a parent subdivide the parent's cell further into smaller cells, and so on. As indicated in Fig. 9, although the MDF is a linear feature (hyperplane as the separator), the tree structure enables the nonlinear classification boundary by using piecewise linear boundary segments.

4.4.2 Approximation for recognition

During the construction of our tree, the MDF's are computed locally. For each subregion $P_{i,j}$, we obtain DKL projection matrices $V_{i,j}$ and $W_{i,j}$ and mean vector $\mathbf{M}_{i,j}$ based on the training samples within $P_{i,j}$, where $V_{i,j}$ is the projection matrix to the MEF space and $W_{i,j}$ is the projection matrix to the MDF space as defined previously. The leaves of the partition tree correspond to the regions which contain the training samples from a single class. The approximator uses the following decision rule to classify the query attention vector $\mathbf{X}$ to the class of a leaf cell.

Definition 1 *Given a training set of attention vectors $L = \{F_1, F_2, \cdots, F_n\}$ the corresponding recursive partition tree does the following for any query attention vector $\mathbf{X}$. If the current level is not a leaf, the recursive partition tree approximator (RPTA) selects the cell with center C_i*

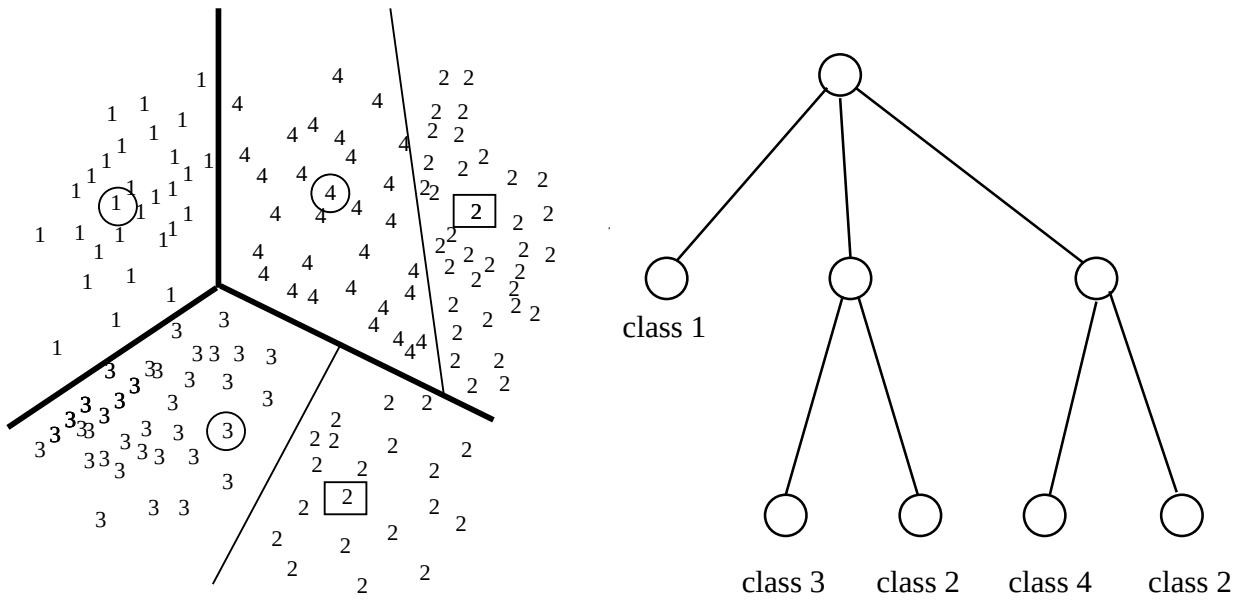


Figure 9: A 2-D illustration of a recursive partition tree. The samples surrounded by circles or rectangulars are the centers of the subregions. Left: the partition. Right: the corresponding recursive partition tree.

for recursive label assignment, where for any other cell with center C_j*, we have* $R_d(\mathbf{X}, C_i) < R_d(\mathbf{X}, C_j)$*. If the current level is a leaf node, the RPTA designates the label of the leaf to the query* $\mathbf{X}$*.*

Since each local cell node of RPTA has its own DKL projection, in order to properly compare the distance across different subspaces, we use a measurement called Mixture Distance (R_d).

Definition 2 *Let* C *be the center of the region* P*,* V *be the projection matrix to the MEF space and* W *be the projection matrix from the MEF space to the MDF space. The Mixture Distance (MD) from a query* $\mathbf{X}$ *attention vector of the center* C *is defined as follows.*

$$R_d(\mathbf{X}, C) = \sqrt{\|\mathbf{X} - VV^t\mathbf{X}\|^2 + \|MM^tC - MM^t\mathbf{X}\|^2}, \tag{6}$$

where $M^t = W^tV^t$*.*

Intuitively, what is being measured can be seen in Fig. 10. In Fig. 10, the original image space is a 3D space, the MEF space is a 2D subspace, and the MDF space is 1D subspace since two classes are well separated along the first MDF vector. The first term $\|\mathbf{X} - VV^t\mathbf{X}\|^2$ in equation (6) is the distance from $\mathbf{X}$ to the MEF space which indicates how well the MEF subspace represents the query vector $\mathbf{X}$. This term is necessary since it is entirely possible that a query vector that is far away from a particular cell's MEF subspace would project very near

to the cell's center. The second term $\|MM^tC - MM^t\mathbf{X}\|^2$ indicates the distance between the MDF components of the query vector and the MDF components of the center vector in the original image space.

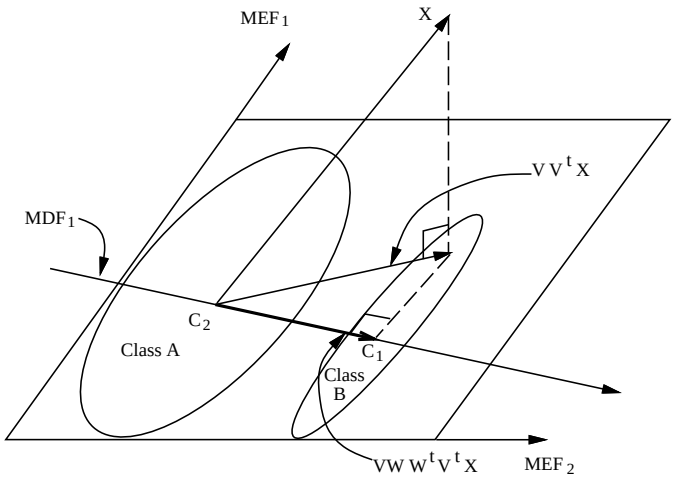


Figure 10: Illustration of components in the Mixture Distance in a 3D original space.

4.5 Convergence of the Approximators

An important issue to study here is how well the above approximators can approximate a function f. Its answer is closely related to the way samples are generated for the learning set L. In this section, we show that our recursive partition tree approximator converges correctly pointwise in probability and thus, there is no local minima problem that is typical with a local search method.

Due to a high complexity and undetermined nature of the way in which a learning set L is drawn from the real world, it is effective to consider that $X(L)$, the set of samples in S, is generated randomly. We know that a fixed L is a special case of random L in that the probability distribution is concentrated at the single location. Thus, we consider $\mathbf{X}$ in $X(L)$ as a random sample from S. The learning set L is generated by acquiring samples from S with a d-dimensional probability distribution function $F(\mathbf{X})$.

Definition 3 *A point $\mathbf{X}_0 \in S$ is positively supported if for any $\delta > 0$ we have $P\{\|\mathbf{X} - \mathbf{X_0}\| \leq \delta\} > 0$, where $P\{e\}$ denotes the probability of the event e.*

If S consists of a finite number of discrete points, a point $\mathbf{X}$ in P is positively supported means that the probability of selecting $\mathbf{X}$ as a sample is not a zero-probability event. If S

consists of infinitely many points, a point $\mathbf{X}$ in P is positively supported means that in any small neighborhood centered at $\mathbf{X}$, the probability of selecting any point in the neighborhood is not a zero-probability event. In practice, we are not interested in cases that almost never appears in a real-world application. An approximate function $\hat{f}$ can assume any value in subregions of S that will never be used in the application, without hurting the real performance of the system. Thus, we just need to investigate how well the approximation can do at points $\mathbf{X}$'s that are positively supported.

Definition 4 *A vector* $\mathbf{X_0}$ *is positively supported by its class in a region* S_M *if for any* $\delta > 0$, *we have* $P\{\mathbf{X} \mid \mathbf{X}$ *and* $\mathbf{X_0}$ *belong to the same class and* $\mathbf{X} \in S_M\} = \eta > 0$.

Given a recursive partition tree RPT, constructed from a training set L of size n, each query $\mathbf{X_0}$ is projected into a subspace at the corresponding leaf node l_n with a projection matrix $M_n = W_n^t V_n^t$, where n denotes the size of the training set. Typically, the larger the number n, the smaller the probability for leaf l_n to receive samples from more than one class, since the tree partitions the space S into finer and finer cells each represented by a leaf node which corresponds to samples from a single class. This is not true if regions of each class has a fractal structure, which is generally not the case in practice.

Theorem 1 *Suppose that a querying vector* $\mathbf{X_0}$ *is positively supported by its class in the region* S_M *of the corresponding leaf node* l_n, *for all sufficient large* n. *Then, given any small number* $\epsilon > 0$, *there is a number* $N > 0$, *so that as long as we independently draw* $n > N$ *learning samples, the approximator* $\hat{f}_n$ *determined by the recursive partition tree has the following property:*

$$P\{\hat{f}_n(\mathbf{X_0}) \neq f(\mathbf{X_0})\} < \epsilon.$$

Note that we consider an RPT as a hierarchical partition for the underlying class boundaries. The proof is presented in Appendix A. This theorem means that the RPT can classify the class of any point $\mathbf{X_0}$ with any arbitrary high $P = 1 - \epsilon$ probability, as long as we have a sufficiently large training set, given the conditions are satisfied.

5 Experimental Results

The framework has been applied to recognize the twenty eight different signs as illustrated in the Fig. 1. The image sequences were obtained while subjects were performing hand signs in

front of a video camera. Since each subject stands roughly the same position related to the camera, the variation of hand size in images is limited. Two different lighting conditions were used. In the current implementation, each hand sign was represented by five images sampled from the video. Figure 11 shows several examples of these sequences.

5.1 Hand segmentation

The method and results of segmentation of hands from complex backgrounds is presented in [14]. Fig. 12 shows some segmentation results.

5.2 Recognition of Hand Sign

The segmentation result was used as the input for sign recognition. The problem is now how to deal with the sequences which has some images that have been rejected by the segmentation routine. In this case, we still output those sequences because there are still good chances that they can be recognized if only one or two images in the sequences are rejected while the rest of them are fine. The number of images used in the training is 3300 (660 sequences). The number of testing images is 805 (161 sequences).

5.2.1 Results of the nearest neighbor approximator in the MEF space

For comparison purpose, we show some experimental results to indicate the performance of the nearest neighbor approximator in the MEF space. We computed MEF's using 660 training sequences. Fig. 14 shows top 10 MEF's.

The number of MEF's was selected based on the variation ratio $r = \sum_{i=1}^{m} \lambda_i / \sum_{i=1}^{n} \lambda_i$, between the explained variance $\sum_{i=1}^{m} \lambda_i$ and the total variance $\sum_{i=1}^{n} \lambda_i$, where λ_i is the ith largest eigenvalue [13]. Table 1 shows the number of MEF's corresponding to the variation ratio.

Table 1: The number of MEF's vs. the variation ratio

The variation ratio	The number of MEF's
10%	1
20%	2
40%	6
80%	48
95%	125

Figure 11: Eight sample sequences. From top to bottom, they represent the signs “happy”, “hot”, “nothing”, “parent”, “pepper”, “smart”, “welcome”, and “yes”, respectively

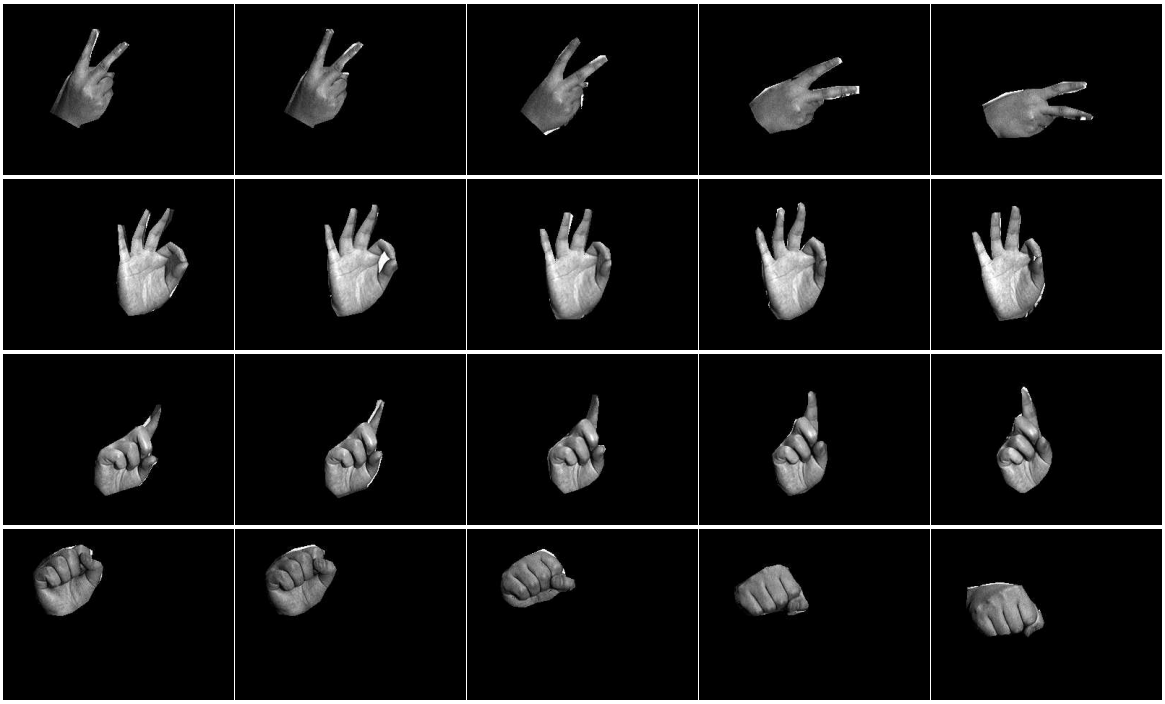


Figure 12: The results of the segmentation are shown after masking off the background.

Fig. 13 shows the performance of the nearest neighbor approximator under the different variation ratios. The performance first improves when the ratio r increases. Then, at the point $r = 0.4$, the performance saturates at the recognition rate 87.0%.

5.2.2 Time of the nearest neighbor query

The nearest neighbor problem which is also known as the *post-office* problem [28] has been studied extensively in the past. There are efficient query algorithms $O(\log n)$ for two- or three- dimensional cases [10, 17]. However, there is still a lack of efficient solutions for the dimensionality higher than three. k-d tree based nearest neighbor algorithms have been widely used in computer vision [2, 42]. k-d trees are extremely versatile and efficient to use in low dimensional cases. However, the performance degrades exponentially in high dimensional cases. R-tree and its variants [22, 34, 1] have similar performance of nearest neighbor searches in high dimensions. In [14], we present an efficient algorithm which uses a hierarchical quasi-Voronoi diagram to search for the nearest neighbor.

Table 2 shows average computation time for each sequence on the SGI INDIGO 2. The time was obtained based on the two different nearest neighbor query approaches, namely, the linear search and the hierarchical quasi-Voronoi diagram in [14]. As shown in the figure, the

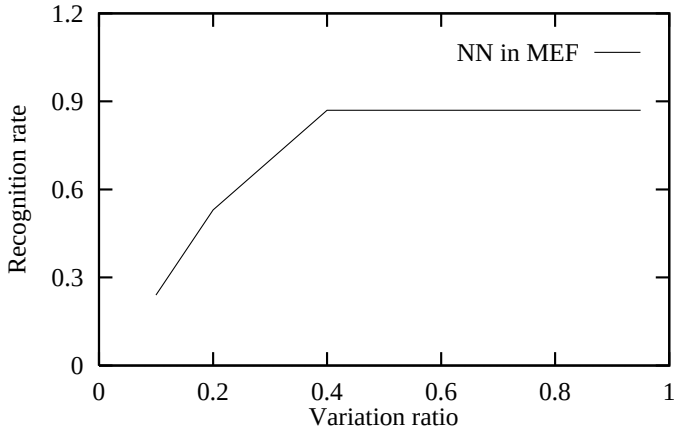


Figure 13: Performance of the nearest neighbor approximator in the MEF space. The performance is given as a function of the number of MEF's used.

use of the hierarchical quasi-Voronoi diagram approach dramatically shortens the query time.

Table 2: Time of two nearest neighbor query approaches

Variation ratio	Time (sec., linear)	Time (sec., hierarchical)
0.1	1.39	0.021
0.2	1.56	0.026
0.4	1.87	0.034
0.6	7.28	0.140
0.95	10.64	0.266

5.2.3 Result of the recursive partition tree approximator in the MDF space

In this experiment, 660 training sequences were used to build a recursive partition tree. The size of each attention image is 32×32, and the length of a sequence is 5. So the dimension of the attention vector is $5 \times 32 \times 32 + 4 \times 8$ (global motion vector) $= 5248$. We used 95% variation ratio for MEF, so the dimension for MEF space is 125 as shown in Table 5.2.2. Top 10 MDF is used to construct the tree. For each region represented by a nonterminal node, we selected an adaptive radius r as defined in Section 4.4 to split the region into subregions. Given r for a nonterminal region, we get $k > 1$ training samples and the distance between each pair of these k samples is greater than r. These k samples were the centers to generate the next level partition. The distance here is the Euclidean distance in the MDF space corresponding to the region. Fig. 15 shows the top 10 MDF's at the root level. If we compare the MDF's in Fig. 15 with the MEF's in Fig. 14, it seems that the top MDF's capture feature locations (edges)

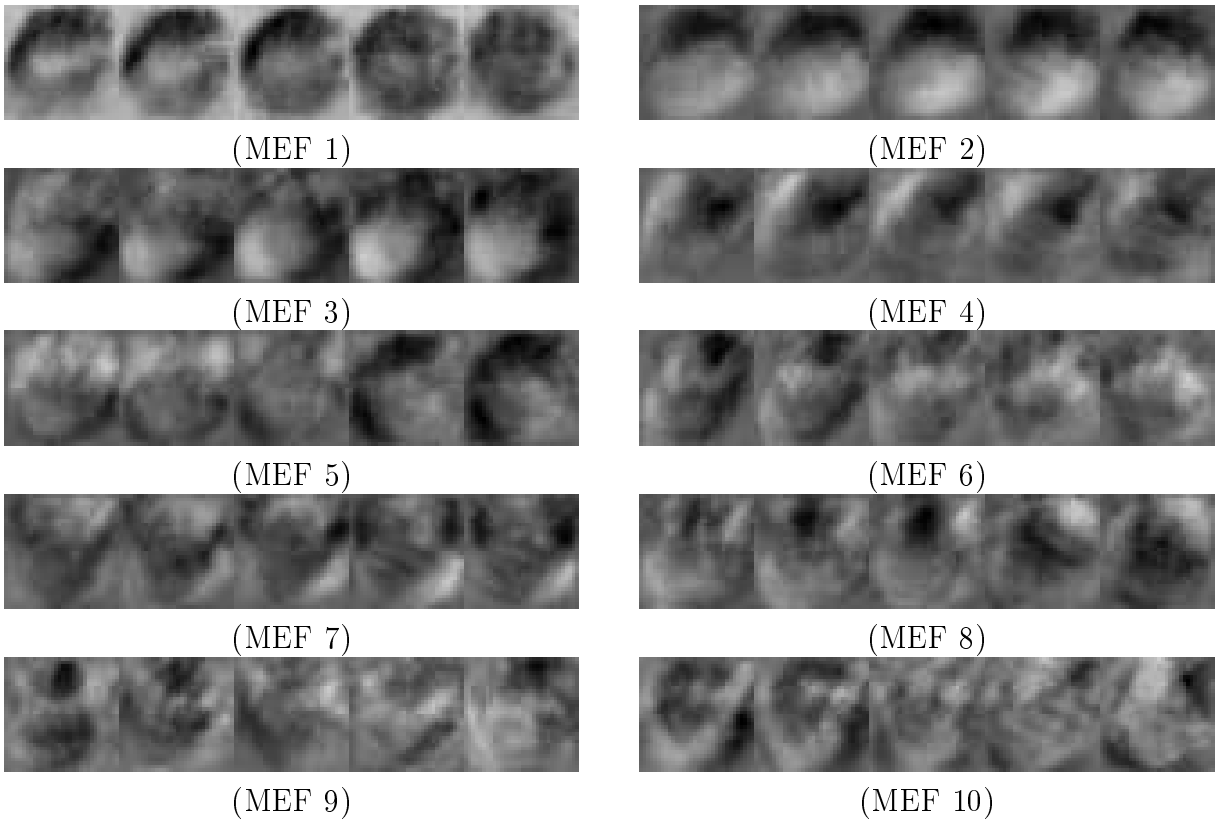


Figure 14: Top ten MEF's

since the intensity transitions are faster in Fig. 15 than in Fig. 14.

Table 3: Summary of the experimental data for RPTA

Training		*Testing*	
Number of training samples	660 (3300 images)	Number of testing sequences	161 (805 images)
Height of the tree	7	Recognition rate	93.2% (87% for MEF)
Number of nodes	90	Time per sequence (sec.)	0.63

Once we have created the recursive partition tree, we used it to recognize the sign. As we did in the experiments for the nearest neighbor approximator in the MEF space. The segmentation result was used as the input for sign recognition. The results are summarized in Table 3. The correct recognition rate of 161 testing sequences is 93.2% which is better than the recognition rate (87.0%) of the nearest neighbor approximator in the MEF space. The average recognition time per sequence is 0.63 second on the SGI INDIGO 2. The time is longer than

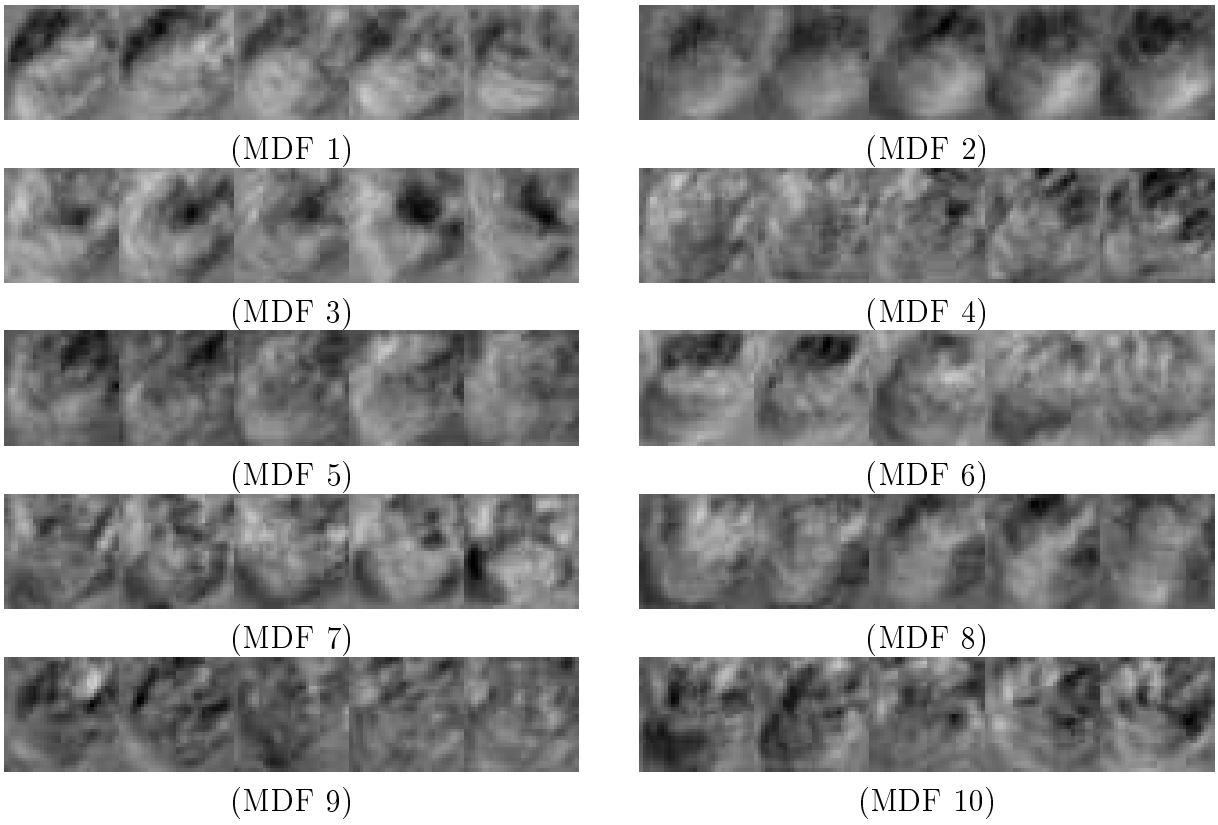


Figure 15: Top ten MDF's

the time (0.27 seconds) of nearest neighbor approximator when the quasi-Voronoi diagram is used in the query. This is because each nonterminal node in the recursive partition tree has its own version of DKL projection matrices, each requires a projection, whereas in the case of the nearest neighbor approximator, we only need one projection to the MEF space.

6 Conclusion

In this paper, we have presented a new approach to recognizing hand signs. In our approach, motion understanding (the hand movement) is tightly coupled with spatial recognition (hand shape). To achieve a high applicability and adaptability to various conditions, we do not impose priori features that the system must use, but rather the system automatically derives features from images during learning using the principle of multiclass, multidimensional discriminant analysis. The recursive partition tree guarantees that the best subset features are selected to distinguish the specific subset classes. This approach combined with our previous work on the hand segmentation forms a new framework which addresses three key aspects of the hand sign interpretation, that is, the hand shape, the location, and the movement. The framework has

been tested to recognize 28 different hand signs. The experimental results have shown that the system achieved a 93.2% recognition rate. It is shown that our approach provides better performance than the nearest neighbor classification in the PCA subspace.

Acknowledgments

The authors would like to thank Dan Swets and Shaoyun Chen for providing us the programs for MEF and MDF computation. Thanks also go to Yu Zhong, Kal Rayes, Doug Neal, and Valerie Bolster for making themselves available for the experiments. This work was supported in part by NSF grant No. IRI 9410741 and ONR grant No. N00014-95-1-0637.

Appendix A

Proof of Theorem 1. Consider the samples in S_M. They all fall into the same leaf node as $\mathbf{X_0}$. Let E_i be the event that the ith sample in $\mathbf{X}(L_i)$ does not belong to the class of $\mathbf{X_0}$. Let D_n be the event that E_i is true for all $i = 1, 2, \cdots, n$. In other words, D_n is the event that $\mathbf{X_0}$ is misclassified: no desired sample goes into S_M.

$$\begin{aligned} P\{\hat{f}_n(\mathbf{X_0}) \neq f(\mathbf{X_0})\} &= P\{D_n\} \\ &= \prod_{i=1}^{n} P\{E_i\} \\ &= \prod_{i=1}^{n} (1-\eta) \\ &= (1-\eta)^n \end{aligned} \tag{7}$$

where $\eta = P\{\mathbf{X} \mid \mathbf{X}$ and $\mathbf{X_0}$ belong to the same class, and $\mathbf{X} \in S_M\} > 0$ as in the definition for $\mathbf{X_0}$ to be positively supported by its class in S_M. Since η is not a function of n and samples are independently drawn, we have $(1-\eta)^n \to \infty$. Thus, there is $N > 0$, such that when $n > N$,

$$P\{\hat{f}_n(\mathbf{X_0}) \neq f(\mathbf{X_0})\} < \epsilon.$$

References

[1] N. Beckmann, H. Kriegel, R. Schneider, and B. Seeger, "The r^*-tree: an efficient and robust access method for points and rectangles", in *Proceedings of the 1990 ACM SIGMOD*, pp. 322-331, 1990.

[2] P. J. Besl and N. D. Mckay, "A method for registration of 3-d shapes", in *IEEE Trans. on PAMI*, vol. 14, pp. 239-256, 1992.

[3] M. J. Black and A. D. Jepson, "Recognizing tremporal trajectories using the condensation algorithm," in *Int'l Conf. Automatic Face and Gesture Recognition*, Nara, Japan, pp. 16-21, April 14-16, 1998,

[4] A. Blake and M. Isard, "3D position, attitude and shape input using video tracking of hands and lips", in *Proceedings of SIGGRAPH 94*, pp. 185-192, 1994.

[5] A. Bobick and A. Wilson, "A state-based technique for the summarization and recognition of gesture", in *Proc. 5th Int'l Conf. Computer Vision*, pp. 382-388, Boston, 1995.

[6] J. Bonvillian, K. Nelson, and V. Charrow, "Language and language related skills in deaf and hearing children", in *Sign Language Studies*, vol. 12, pp. 211-250, 1976.

[7] H. Bornstein and K. Saulnier, *The Signed English Starter*, CLERC BOOKS, Gallaudet University Press, Washington, D.C., 1989.

[8] L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone, *Classification and Regression Trees*, Chapman & Hall, New York, 1993.

[9] J. S. Brown, *A Vocabulary of Mute Signs*, Baton Rouge, Louisiana, 1856.

[10] B. Chazelle, "How to search in history", in *Inf. Control*, vol. 64, pp. 77-99, 1985.

[11] K. Cho and S. M. Dunn, "Learning shape classes", in *IEEE Tran. on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 882-888, 1994.

[12] R. Cipolla, Y. Okamoto and Y. Kuno, Robust structure from motion using motion parallax, in *IEEE Conf. Computer Vision and Pattern Recog.*, pp. 374-382, 1993.

[13] Y. Cui, D. Swets and J. Weng, "Learning-Based Hand Sign Recognition Using SHOSLIF-M", in *Proc. Int'l Conf. Computer Vision*, pp. 631-636, Boston, MA, 1995.

[14] Y. Cui and J. Weng, "Hand segmentation using learning-based prediction and verification for hand-sign recognition", in *Proc. IEEE Conf. Computer Vision and Pattern Recog.*, pp. 88-93, San Francisco, CA, June, 1996.

[15] Y. Cui and J. Weng, "A learning-based prediction-and-verification segmentation scheme for hand sign image sequences," *IEEE Trans. Pattern Analysis and Machine Intelligence*, accepted and to appear.

[16] T. Darrell and A. Pentland, "Space-time gestures", in *IEEE Conf. Computer Vision and Pattern Recog.*, pp.335-340, 1993.

[17] D. P. Dobkin and R. J. Lipton, "Multidimensional searching problems", in *SIAM J.*

Comput., vol. 5, pp. 181-186, 1976.

[18] A. C. Downton and H. Drouet, "Image analysis for model-based sign language coding", in *Progress in image analysis and processing II: Proc. of the 6th International Conference on Image Analysis and Processing*, pp. 79-89, 1991.

[19] J. Davis and M. Shah, Visual gesture recognition, in *IEE Proc. Vis. Image Signal Process*, vol. 141, No. 2, pp. 101-106, April 1994.

[20] R.Duda and P.Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.

[21] M. Etoh, A. Tomono and F. Kishino, "Stereo-based description by generalized cylinder complexes from occluding contours", in *Systems and Computers in Japan*, vol. 22, no. 12, pp. 79-89, 1991.

[22] A. Guttman, "R-trees: a dynamic index structure for spatial searching", in *ACM SIGMOD*, pp. 905-910, 1984.

[23] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd Edition, Academic Press Inc., San Diego, CA, 1990.

[24] T.S. Huang and V. I. Pavlovic, "Hand gesture modeling, analysis, and synthesis", in *Proc. International Workshop on Automatic Face- and Gesture- Recognition*", pp. 73-79, 1995.

[25] A.K. Jain and R.C. Dubes, *Algorithms for Clustering Data*, Prentice-Hall, New Jersey, 1988.

[26] C. Kervrann and F. Heitz, "A hierarchical statistical framework for the segmentation of deformable objects in image sequences", in Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 724-728, 1994.

[27] M. Kirby and L. Sirovich, "Application of the Karhunen-Loeve procedure for thecharacterization of human faces", in *IEEE Tran. on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 103-108, 1990.

[28] D. Knuth, *The Art of Computer Programming III: Sorting and Searching*, Addison-Wesley, Reading, Mass., 1973.

[29] J. J. Kuch and T. S. Huang, "Vision based hand modeling and tracking", in *Proc. International Conference on Computer Vision*, June, 1995.

[30] M.M. Loeve, *Probability Theory*, Princeton, NJ: Van Nostrand, 1955.

[31] B. Moghaddam and A. Pentland, "Maximum likelihood detection of faces and hands", in Proc. International Workshop on Automatic Face- and Gesture- Recognition", pp. 122-128, June 1995.

[32] H. Murase and S. K. Nayar, "Illumination planning for object recognition in structured environments," in Proc *IEEE Conf. Computer Vision and Pattern Recognition*, Seattle, WA, pp. 31 - 38, June 1994.

[33] J. Quinlan, "Introduction of decision trees", in *Machine Learning*, vol. 1, num. 1, pp. 81-106, 1986.

[34] T. Sellis, N. Roussopoulos and C. Faloutsos, "The r+-tree: a dynamic index for multidimensional objects", in *Proceedings of 13th International Conference on VLDB*, pp. 507-518, 1987.

[35] T.E. Starner and A. Pentland, "Visual recognition of American sign language using hidden markov models", in *Proc. International Workshop on Automatic Face- and Gesture-Recognition*", pp. 189-194, June 1995.

[36] W. Stokoe, "Sign language structure: an outline of the visual communication system of the American deaf", *Studies in Linguistics Occasional Paper No. 8*, 1960.

[37] D. L. Swets and J. Weng, "Using discriminant eigenfeatures for image retrieval," *IEEE Trans. Pattern Recog. and Machine Intell.*, vol. 18, no. 8, pp. 831-836, 1996.

[38] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991.

[39] S.S.Wilks, *Mathematical Statistics*, Wiley, New York, 1963.

[40] J. Weng, On comprehensive visual learning, in *Proc. NSF/ARPA Workshop on Performance vs. Methodology in Computer Vision*, pp. 152-166, Seattle, WA, June 24-25, 1994.

[41] M. Yang and N. Ahuja, "Extracting gestural motion trajectories," in *Int'l Conf. Automatic Face and Gesture Recognition*, Nara, Japan, pp. 10- 15, April 14-16, 1998,

[42] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces", in *International Journal of Computer Vision*, vol. 13, pp. 119-152, 1994.