International Journal of Humanoid Robotics
Vol. 7, No. 3 (2010) 331–356

DOI: 10.1142/S0219843610002180

HAND POSTURE AND FACE RECOGNITION USING A FUZZY-ROUGH APPROACH

P. PRAMOD KUMAR*, PRAHLAD VADAKKEPAT† and LOH AI POH‡

Department of Electrical and Computer Engineering,
4 Engineering Drive 3,
National University of Singapore,
Singapore — 117576
**pramod@nus.edu.sg*
†prahlad@nus.edu.sg
‡elelohap@nus.edu.sg

Received 4 July 2009
Revised 9 June 2010
Accepted 30 June 2010

A novel algorithm based on fuzzy-rough sets is proposed for the recognition of hand postures and face. Features of the image are extracted using the computational model of the ventral stream of visual cortex. The recognition algorithm translates each quantitative value of the feature into fuzzy sets of linguistic terms using membership functions. The membership functions are formed by the fuzzy partitioning of the feature space into fuzzy equivalence classes, using the feature cluster centers generated by the subtractive clustering technique. A rule base generated from the lower and upper approximations of the fuzzy equivalence classes classifies the images through a voting process. Using genetic algorithm, the number of features required for classification is reduced by identifying the predictive image features. The *margin of classification*, which is a measure of the discriminative power of the classifier, is used to ensure the quality of classification process. The fitness function suggested assists in the feature selection process without compromising on the classification accuracy and margin. The proposed algorithm is tested using two hand posture and three face datasets. The algorithm provided good classification accuracy, at a less computational effort. The selection of relevant features further reduced the computational costs of both feature extraction and classification algorithms, which makes it suitable for real-time applications. The performance of the proposed algorithm is compared with that of support vector machines.

Keywords: Hand posture recognition; face recognition; human–robot interaction; feature selection; fuzzy-rough sets; computer vision; biologically inspired vision.

1. Introduction

Hand posture and face recognition are two important areas of research in visual pattern recognition, having wide applications in human–robot interaction (HRI), human–computer interaction (HCI), and virtual reality (VR). Visual interaction is an easy and effective way of interaction, which does not require any physical contact and does not get affected by noisy environments.

Recognition of hand gestures and face is useful for many tasks involving humans and robots. However, designing a recognition system for a robot is a complex process requiring adaptation to the varying and complex backgrounds, person-independent recognition, fast processing speed, low-computational costs, and compatibility with the system architecture and video interface. The challenge of solving this problem reliably and efficiently in realistic settings is what makes research in this area difficult.

Hand gestures can be generally classified as *static* or *dynamic* gestures. Static hand gestures (also called a hand posture) are those in which the hand position does not change during the gesturing period. Static gestures rely only on the information about the flexure angles of the fingers. In dynamic hand gestures (also called a hand gesture), the hand position is temporal and it changes continuously with respect to time. Dynamic gestures rely not only on the finger's flex angles but also on the hand trajectories and orientations. Dynamic gestures can be viewed as actions composed of a sequence of static gestures that are connected by continuous motions. Therefore, a dynamic hand gesture can be expressed as a hierarchical combination of static gestures. In this article, the recognition of static hand gestures is considered.

There are different methods for hand gesture recognition. First attempts to solve this problem resulted in mechanical devices that directly measure hand and/or arm joint angles and spatial position, using glove-based devices. Later vision based non-contact methods developed. The tools used for vision-based hand gesture recognition can be broadly classified as (i) hidden Markov model (HMM)-based methods,[1–6] (ii) neural network- and learning-based methods,[7–15] and (iii) other methods (graph-algorithm-based methods,[16–19] 3D-model-based methods,[20,21] statistical and syntactic methods,[22,23] eigen-space-based method,[24] etc.).

The research efforts in face processing include face detection, face recognition, face tracking, pose estimation, and expression recognition.[25] The major recognition methods applied to face images are eigenface, neural network, dynamic link architecture, HMM, geometrical feature matching, and template matching.[26] The surveys on face recognition[27] and face detection[25] provide the details of these methods.

The present article considers the multi-class recognition of hand postures and human faces. It proposes a fuzzy-rough set-based novel classification algorithm that classifies the images at a less computational cost. The features of the images are extracted using a cortex-like mechanism.[28] The algorithm then partitions the feature space by fuzzy-discretization into fuzzy equivalence classes. The lower and upper approximations of the fuzzy equivalence classes are obtained, which are used to derive the fuzzy if-then rules. These rules are utilized for classifying a new image, through a voting process. The proposed algorithm has a polynomial time complexity.

By identifying the decision attributes, the performance of a classification algorithm can be improved. The cost of classification is sensitive to the number of attributes/features used to construct the classifier. In the present work, genetic algorithm (GA) is utilized to reduce the number of features required for classification, by identifying and removing irrelevant and redundant features. The proposed classifier is a margin classifier and it provides *margin of classification*, a measure of the distance from the classification boundary. The margin of classification is used to define the fitness function. The proposed fitness function assists in the feature selection process without compromising on the classification accuracy and margin. The reduction in number of features needed for classification reduced the number of features to be extracted. This increased the speed of both feature extraction as well as classification processes, which makes the proposed algorithm suitable for real-time applications.

The proposed algorithm is tested using three face datasets (a subset of Yale face database B,[29] a subset of color FERET database,[30] and a subset of CMU face dataset[31]) and two hand posture datasets (a newly created dataset namely NUS hand posture dataset, and Jochen Triesch hand posture dataset[32]). Moreover, the algorithm is tested by capturing the hand images online, and the performance is compared with that of support vector machines (SVM).

Section 2 of this article presents the necessary background information of the proposed algorithm. Section 3 explains the proposed recognition algorithm. The algorithm is tested on different datasets and the results are discussed in Sec. 4. The article is concluded with the remarks provided in Sec. 5.

2. Background

This section presents a review of the concept of fuzzy-rough sets and a brief explanation of the feature extraction system.

2.1. *Fuzzy-rough sets*

Real-world classification problems often involve continuous data that makes the design of reliable classifiers difficult. One way to handle continuous data is by partitioning the data into crisp or discrete intervals. This process of discretization determines how coarsely the data is split into intervals. The crisp discretization is achieved by generating a set of *cuts* of features within the dynamic ranges of the corresponding features. The positions of cuts are very sensitive to the subsets of the information system, which are used to generate the cuts, as well as to the methodology adopted. The position sensitivity of cuts may make the classification accuracy adversely affected. Fuzzy sets, which are generalization of the classical sets, proposed by Zadeh in 1963,[33] offer solutions for tackling such difficulties associated with continuous data. It suggests the fuzzy discretization of the feature space and solves the problems associated with crisp discretization.

Pawlak introduced the rough-set theory in the early 1980s, as a tool to handle inconsistencies among data.[34–36] A rough set is a formal approximation of a vague concept by a pair of precise concepts, the lower and upper approximations.

No additional knowledge about the data, such as prior probability in probabilistic approach or grade of membership in fuzzy-set theory, is required in rough sets. Rough sets handle uncertainty by computing the lower and upper approximations of the concept under consideration.

Fuzzy and rough sets, which are the two computational intelligence tools used for decision making in uncertain situations, can be combined to form rough-fuzzy sets or fuzzy-rough sets.[37–44] Fuzzy- and rough-set theories are considered complementary in that they both deal with uncertainty; vagueness for fuzzy sets, and deal with indiscernibility for rough sets.[38] Combining the two theories provides the concept of lower and upper approximations of fuzzy sets and it is useful for addressing the classification problems.[40,45–47] The present work utilizes the fuzzy-rough set approach for the recognition of hand postures and human faces.

A fuzzy similarity relation is used to replace an equivalence relation in the rough sets, resulting in a deviation of rough-set theory namely the fuzzy-rough sets.[37,38] In fuzzy-rough sets, the concept of crisp equivalence classes, which form the basis for rough-set theory, is extended to fuzzy-set theory to form fuzzy equivalence classes.[38] In fuzzy-rough sets the equivalence class is fuzzy in addition to the fuzziness in the output class.[45]

Let the equivalence classes be in the form of fuzzy clusters $F_1, F_2, \ldots, F_H$, which are generated by the fuzzy partition of the input set X into H number of clusters. Then, the description of any fuzzy set C_c (output class) by means of the fuzzy partitions under the form of a lower and an upper approximation $\underline{C_c}$ and $\overline{C_c}$ is as follows:

$$\begin{aligned} \mu_{\underline{C_c}}(F_j) &= \inf_x\{\max(1-\mu_{F_j}(x), \mu_{C_c}(x))\} \quad \forall\, x \in X \\ \mu_{\overline{C_c}}(F_j) &= \sup_x\{\min(\mu_{F_j}(x), \mu_{C_c}(x))\} \quad \forall\, x \in X. \end{aligned} \tag{1}$$

The tuple $\langle \overline{C_c}, \underline{C_c} \rangle$ is called a fuzzy-rough set. Here, $\mu_{F_j}(x)$ and $\mu_{C_c}(x)$ are the fuzzy membership of the input x to the fuzzy equivalence class F_j and output class C_c, respectively. The fuzzy-roughness appears when a fuzzy equivalence class contains patterns that belong to different classes.

Fuzzy equivalence classes and fuzzy lower and upper approximations have been successfully applied in classification problems.[48–51] Most of these techniques make use of pre-defined fuzzy membership functions for the development of the classifier. The current work proposes an algorithm that automatically generates fuzzy membership functions and corresponding classification rules for pattern classification, directly from the training dataset, based on the fuzzy-rough set approach.

2.2. *Feature extraction*

The features of the image that are to be used for pattern recognition is an ongoing research topic in computer vision. Gabor wavelet (also known as Gabor function or Gabor filter) based features have good discriminative power between different textures and shapes in the image. Gabor filters resemble the receptive fields of neurons in the primary visual cortex of mammals.[52] Use of the 2D Gabor wavelet representation in computer vision was pioneered by Daugman.[53] Serre *et al.* extended this approach[28,54] with a hierarchical system imitating the primate visual system, based on a quantitative theory of the ventral stream of the visual cortex.[55] The features proposed by them, namely the C_2 standard model features (SMFs), are scale and position-tolerant, and the feature extraction algorithm does not require the segmentation of the image. In addition, the number of the extracted features is independent of the input image size. They used the features for robust object recognition.[28,54] Later, these features were used for hand-writing recognition[56] and face recognition.[57] The proposed algorithm utilizes the C_2 features for the multi-class recognition of human faces and hand postures.

The visual object recognition is mediated by the ventral visual object-processing stream in the visual cortex.[55,58] Serre *et al.*[28] proposed a computational model of the ventral stream, based on the standard model of visual object recognition.[55] This model provides an algorithm for feature extraction, which comprises four layers (Table 1), that imitates the feedforward path of object recognition in the ventral stream of primate visual cortex.[55]

Layer 1 (S_1) consists of a battery of Gabor filters with different orientations (4) and sizes (16 sizes divided into 8 bands). This imitates the simple cells in the primary visual cortex (V1) that filters the image for the detection of edges and bars. Layer 2 (C_1) models the complex cells in V1, by applying a MAX operator locally (over different scales and positions) to the first layer results. This operation provides tolerance to different object projection size, its position and rotation in the 2D plane of the visual field. In layer 3 (S_2), radial basis functions (RBFs) are used to imitate the V4 and posterior inferotemporal (PIT) cortex. This aids shape recognition by comparing the complex features at the output of C_1 stage (which corresponds to the retinal image) with patches of previously seen visual image and shape features (in human, these patterns are stored in the synaptic weight of the neural cells). Finally, the fourth layer (C_2) applies a MAX operator (globally, over all scales and positions) to the output of layer S_2, resulting in a representation that expresses the best comparison with previously seen images. The output of layer 4 is the C_2 SMFs, which are used for the classification of the image.

Simple cells in the third layer implement an RBF, which combines bars and edges in the image to more complex shapes. RBFs are a major class of neural network model, comparing the distance between input and a prototype.[59] The response

Table 1. Different layers in the C_2 feature extraction system.

Layer	Process	Represents
S_1	Gabor filtering	Simple cells in V1
C_1	Local pooling	Complex cells in V1
S_2	Radial basis functions	V4 & posterior inferotemporal cortex
C_2	Global pooling	Inferotemporal cortex

of each S_2 unit depends in a Gaussian-like way on the Euclidean distance between a new input and a stored prototype. The prototype patches of different sizes (center of the RBF units) are drawn randomly (random image and position) from the training images at the level of the second layer (C_1). Each patch contains all the four orientations. The third layer compares these patches by calculating the summed Euclidean distance between the patch and every possible crop (combining all orientation) from the image of similar size. This comparison is done separately with each scale-band representation in the second layer.

The final set of shift and scale invariant C_2 responses is computed by taking a global maximum over all scales and positions for each S_2 type, i.e., the value of the best match between a stored prototype and the input image is kept and the rest is discarded. Each C_2 feature corresponds to a specific prototype patch with a specific patch size (in layer 3). The more the number of extracted features the better is the classification accuracy. However, when more number of features are extracted, the computational burden (both for feature extraction and classification) will increase. In the present work, 250 prototype patches with four different patch sizes are used for the feature extraction, similar to that done in the object-recognition task.[28] The total number of features extracted is 1000. The relevant features that are good in discriminating different classes are selected from these 1000 features, which reduce the feature extraction and classification time, by maintaining the classification accuracy.

3. The Proposed Recognition System

The proposed recognition algorithm is discussed in this section. The recognition of a pattern in the image consists of two processes, feature extraction and classification. The aim is to come up with a simple algorithm that is computationally less intensive, while automatically generating a classifier from the training set, using the fuzzy-rough set approach.

Figure 1 shows an overview of the components of the proposed recognition system. The available data is divided into training, validation, and testing datasets. The classification rules are generated in the training phase, after extracting the features of the image. The discriminative features of the images are identified and selected in the feature selection phase. In the testing phase, the images are classified using the generated classification rules.

3.1. *Training phase*

This section explains the various steps associated with the training of the classifier. Figure 2 shows the training phase of the classifier. The features of the image are extracted using the C_2 feature-extraction system explained in Sec. 2.2. The fuzzy discretization of the feature space is done using the cluster centers identified by the subtractive clustering technique.[60] Then, the lower and upper approximations of the fuzzy sets are found out and the classification rules are generated. The following subsections explain each of these phases.

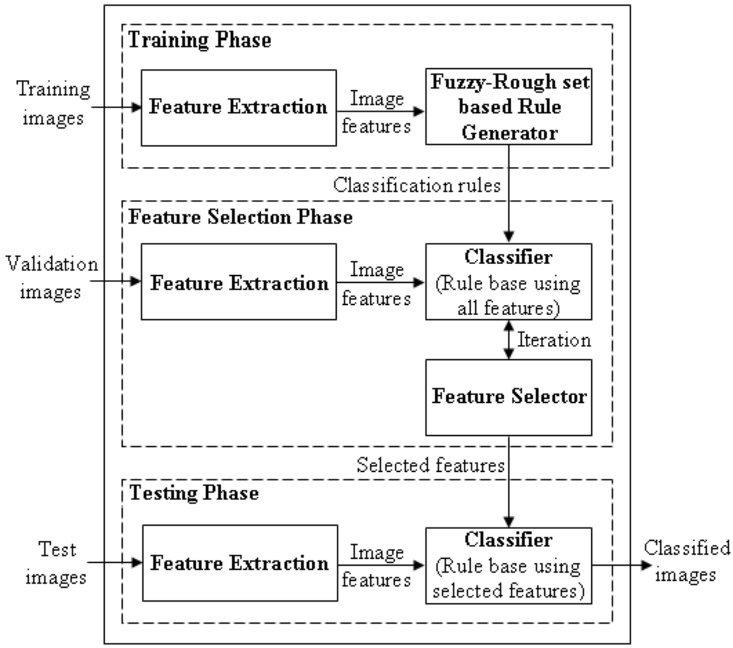


Fig. 1. Overview of the recognition algorithm.

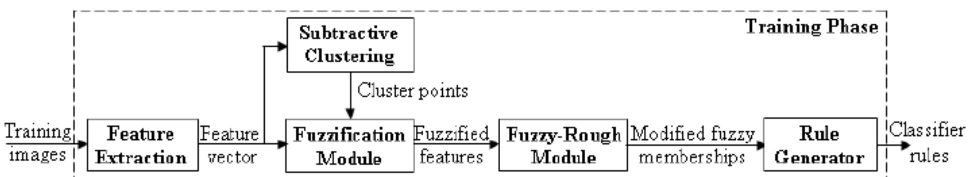


Fig. 2. Training phase of the recognition algorithm.

3.1.1. *Fuzzy membership functions from feature cluster centers*

The fuzzy membership functions are created in this step using the feature cluster centers, and these are utilized to partition the data into fuzzy equivalence classes. The subtractive clustering technique,[60] which is an extension of Yager's mountain clustering method,[61] is utilized to identify the feature cluster center points. Subtractive clustering assumes that each data point is a potential cluster center and calculates a measure of the likelihood that each data point would define the cluster center, based on the density of surrounding data points. The algorithm selects the data point with the highest potential as the first cluster center and then removes all the data points in the vicinity (as specified by the subtractive clustering radius that usually lies within $[0.2, 0.5]$) of the first cluster center. The second data cluster and its center point are identified next. This process is repeated until every data sample is within the radius of one of the cluster centers.

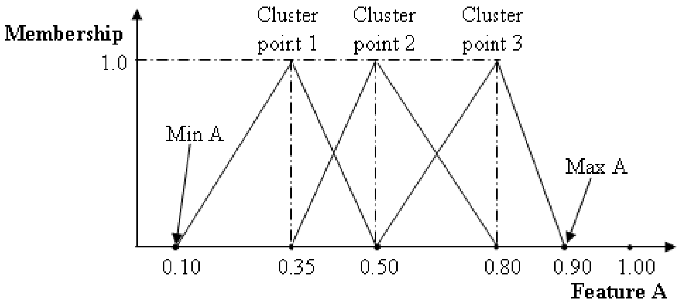


Fig. 3. Formation of membership functions from cluster center points.

The cluster centers are determined for each feature (in the feature vector corresponding to the training images) of each class. In subtractive clustering, the number of clusters increases with smaller radius leading to increased computational requirements. The clustering radius used in the present work is 0.5. The number of cluster centers generated varied from 1 to 3. Once the data clusters are identified, the peaks of the triangular membership functions are placed at these points and, its left/right sides span to adjacent left/right data cluster centers. The minimum/maximum feature values position the left/right sides of the leftmost/rightmost membership functions. For example, if a set of cluster points $cp_1 = \{0.35, 0.5, 0.8\}$ are obtained for a feature A, which is within the range $[0.1, 0.9]$ for class C, then the triangular membership functions for A, for class C are as shown in Fig. 3.

A set of membership functions is obtained for each feature of every class using the cluster center points. The training data is then fuzzified using the generated membership functions and the lower and upper approximations are obtained.

3.1.2. *Modified fuzzy membership functions*

It is found that the same fuzzy equivalence class contains the samples from different output classes, which cause the fuzzy-rough uncertainty. The proposed algorithm finds the lower and upper approximations of the fuzzy equivalence classes. The limiting values of the memberships that partition the definite and possible members of the output class are identified. The classification rules are generated utilizing these membership values. The limiting values of the memberships are calculated as follows (Fig. 4).

Let MF be the fuzzy set associated with a particular cluster center point of a feature in a class and,

$$\mu^{\max}(C_i) = \max\{\mu_{MF}[x_{C_i}(l)]\}, \tag{2}$$

where, $x_{C_i}(l)$ is a sample from the class C_i, and,

$$C_{\max} = \underset{C_i}{\operatorname{argmax}}\{\mu^{\max}(C_i)\}, \tag{3}$$

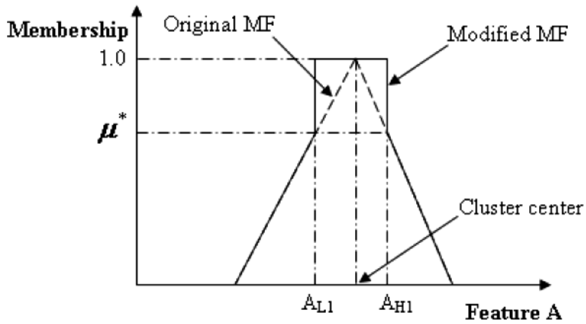


Fig. 4. Modified fuzzy membership function.

then,

$$\mu^* = \max_{C_i \neq C_{\max}} \{\mu^{\max}(C_i)\}. \tag{4}$$

μ^* is the maximum of the membership values associated with data samples belonging to classes other than $C_{\max}$. This limiting value of membership entails the rule: if any $\mu_{MF}(x) > \mu^*$, then the associated class is $C_{\max}$. This rule implies that if the value of the particular feature A is within $[A_{L1}, A_{H1}]$, then the sample belongs to class $C_{\max}$. These membership limits decide whether a particular sample is a definite or possible member of the output class ($C_{\max}$). The rule always holds true for the training samples. However, some of the rules classify only a small number of training samples (say one or two), if the samples from various classes are well mixed. To increase the reliability of the classifier by avoiding the generation of rules from outlying data points, only those rules that classify two or more number of training samples are stored in the rule base. In order to classify a new sample, a generalization of the rule is needed, which is provided in Sec. 3.1.3.

3.1.3. *Generation of classification rules*

The generalized classification rules are discussed in this section. Let $\{\mu^*_{ijk}\}$ be the set of membership values obtained as per Eq. (4), where

i	$1, \ldots, p$	: p — number of classes,
j	$1, \ldots, q$	: q — number of features,
k	$1, \ldots, r$	: r — number of clusters corresponding to the jth feature of the ith class,

then the following two rules are utilized for classifying the patterns. *Rule* 1 is a voting step, whereas *Rule* 2 is the decision-making step.

Rule 1:

$$IF\ [\mu_{MF_{ijk}}(x) > \mu^*_{ijk}]\ THEN\ [N_{C_i} = N_{C_i} + 1], \tag{5}$$

where x is the sample to be classified and N_{C_i} is the number of votes for a particular class C_i.

Rule 2:

$$C = \underset{C_i}{\operatorname{argmax}}\{N_{C_i}\}, \tag{6}$$

where C is the class to which the sample belongs to.

For example, in case of a dataset with three classes, the number of votes in the three classes namely N_{C_1}, N_{C_2}, and N_{C_3} is calculated for the particular sample under consideration (*Rule* 1). *Rule* 2 then identifies the class with maximum votes. *Rules* 1 and 2 serve to form the classifier rule base, keeping the algorithm computationally simple.

The proposed classifier is a *margin classifier* that provides the minimum distance from the classification boundary, namely *margin of classification* (MC), for each sample. The margin of classification of a particular sample for the proposed classifier is defined as,

$$MC = \mathit{number\ of\ positive\ votes} \ - \ \mathit{max} \cdot (\mathit{number\ of\ negative\ votes}). \tag{7}$$

For a sample from class 1, let the values $N_{C_1} = 90$, $N_{C_2} = 5$, and $N_{C_3} = 10$ (in the case of a three-class classification problem). Then, the MC for the sample is $90 - 10 = 80$. In this case, the sample received 90 positive votes.[a] The number of negative votes are five and ten in classes 2 and 3, respectively. A positive margin indicates correct classification whereas negative margin indicates misclassification. In Sec. 3.2, the margin of classification is used to identify the relevant features of the image.

3.2. *Feature selection algorithm*

The number of features used to describe a pattern determines the size of the search space to be explored.[62] An abundance of features increases the size of the search space, thereby increasing the time needed for classification. The cost of classification in any classification process is sensitive to the number of features used to construct the classifier. The available features in a dataset can be categorized into four. (i) Predictive/relevant: the features that are good in discriminating between different classes; (ii) Misleading: the features that affect the classification task negatively; (iii) Irrelevant: the features that provide a neutral response to the classifier algorithm; and (iv) Redundant: the features of a class that have other relevant features for the discrimination. The presence of misleading features will reduce the classification accuracy and the presence of irrelevant and redundant features will increase the computational burden. It is desirable to remove such attributes from the dataset. The removal of misleading, irrelevant, and redundant features reduces

[a]Voting is positive if the voted class and the actual class are the same. Otherwise it is negative.

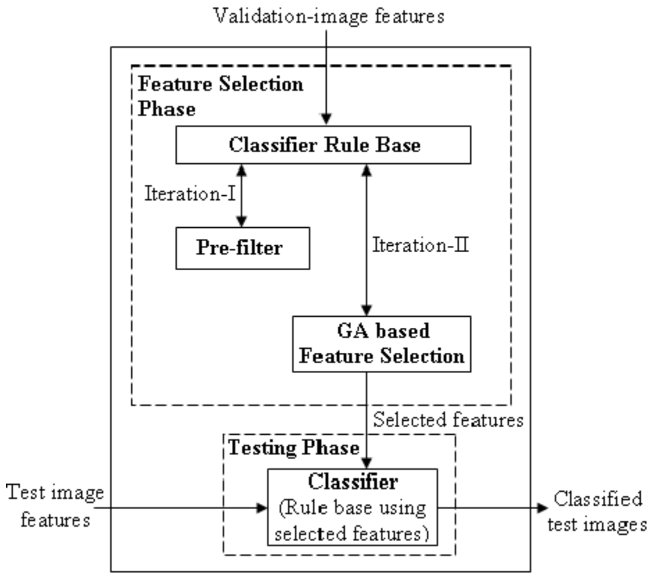


Fig. 5. Feature selection and testing phase.

the size of the rule base of a classifier by preserving the relevant and predictive features. This process is known as attribute reduction[63] or, in the context of machine learning, feature selection.[64]

In the present work, the feature reduction is done in two stages. The first stage is a pre-filter that filters out the misleading features. The second stage is a GA-based feature selection algorithm that removes the irrelevant and redundant features. The feature reduction process is depicted in Fig. 5.

Each classification rule in the proposed classifier corresponds to a particular feature of the image. The classification rules provide positive or negative votes to the samples. This helps to identify the relevant features (the features that provide more positive votes) of the image. The proposed margin of classification (7) is utilized for the feature selection process. The selection of the relevant features improves the performance of the recognition system and reduces the computational burden of the overall algorithm.

3.2.1. *Pre-filter: weeding out the misleading features*

Pre-filter (Fig. 6) identifies and removes the features that give negative votes in the classification process. The features, which negatively vote for two or more samples as per Eq. (5), are weeded out. The validation data is sorted in the ascending order of the margin of classification and those samples that have less *MC* are considered (the samples that have less *MC* have more negative votes and are nearer to the classification boundary). The features that vote these samples negatively are then identified and removed from the dataset. For example, if $x(1)$, $x(2)$, $x(3)$, and $x(4)$

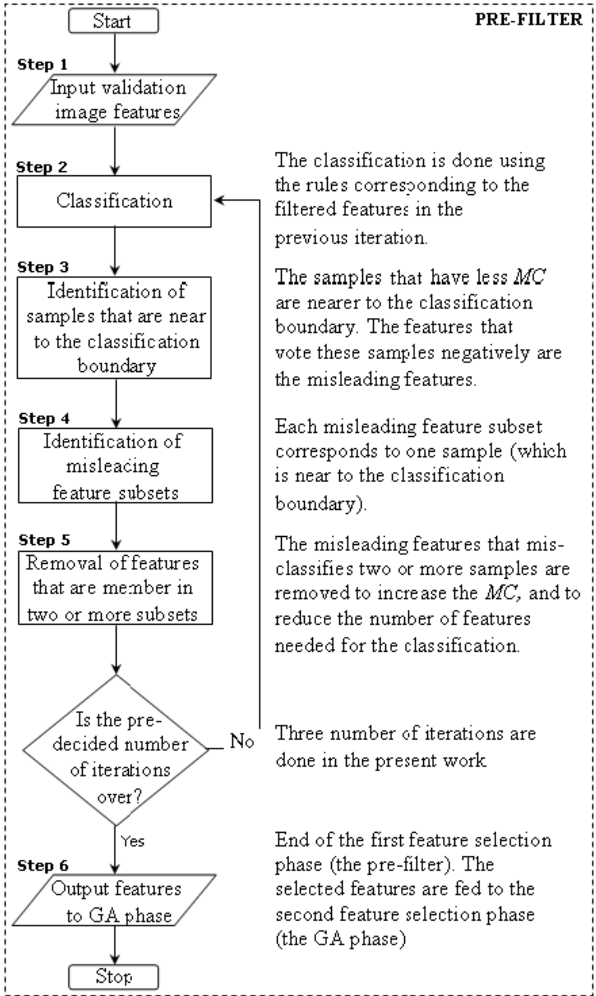


Fig. 6. Flowchart of the pre-filter.

are the first four samples in the sorted list and A_1, A_2, A_3 and A_4 are the subsets of features (the misleading feature subsets) that give negative votes to these samples, respectively, then, the set of features eliminated is A', where A' consists all the features that are members in at least any two of the subsets A_1, A_2, A_3, and A_4. The process is iterated a few times by repeating the validation procedure. The number of iterations needed depends on the nature of the dataset. More number of iterations are undergone for a dataset with large number of misleading features. In the present work, the number of iterations used is three. The elimination process is a simple operation, but it has significant effect in increasing the margin of classification and in reducing the computational load at the second stage (as the number of classification rules is reduced).

3.2.2. *Feature selection using GA*

GAs are widely used for feature selection and extraction in machine learning.[65–69] In the present work, GA is utilized to remove the irrelevant and redundant features. A fitness function is defined, which considers the accuracy, cost, and margin of classification in accordance with the classification algorithm proposed in Sec. 3.1.

Each chromosome in the population represents the set of features used for classification. A chromosome consists of N_p bits and each bit represents a specific feature. When a particular bit is unity, it indicates the presence of the feature and its absence is indicated by a zero. In the present work, the value of N_p is 200. Elitism technique is utilized to create a new population and the fitness function Eq. (8) is minimized.

$$F = w_1 \ * \ N_1 + w_2 \ * \ N_2 - w_3 \ * \ N_3, \tag{8}$$

where

N_1	number of misclassifications,
N_2	number of features used for classification,
N_3	minimum MC (minimum of MCs of different samples),
w_1, w_2, and w_3	weighing factors.

N_1 represents the accuracy of classification and N_2 represents the computational expense (presence of more features leads to more classification rules and so is the computational cost). N_3 provides preference to chromosomes with better margin of classification. The minimum margin of classification is the margin of the sample that is nearest to the classification boundary. w_1, w_2, and w_3 are to be tuned depending on the dataset. An empirical guideline is to select the weights (w_1, w_2, and w_3) such that $w_1 > w_2 > w_3$, which provides the first preference to the accuracy of classification, second preference to the number of features, and third preference to the margin of classification. In the present work, the values of w_1, w_2, and w_3 are 1.0, 0.6, and 0.4, respectively.

The cross-over and mutation rate used are 0.9 and 0.01, respectively. The population size of the GA is kept 100. Iterations are done until the pre-decided maximum number of iterations (50 in the present work) is reached or till a nonzero classification error is originated. If iteration is stopped after the origination of a nonzero classification error, then the features used in the previous iteration is selected. The number of features needed for classification is decided by taking the average over ten runs of GA.

A detailed flowchart of the classifier development algorithm is provided in Fig. 7.

3.3. *Testing phase*

Figure 8 shows the flowchart of the testing phase of the recognition system. The selected features of the unlabeled test images are extracted using the C_2 feature-extraction system. Then, the memberships of these features in different classes are

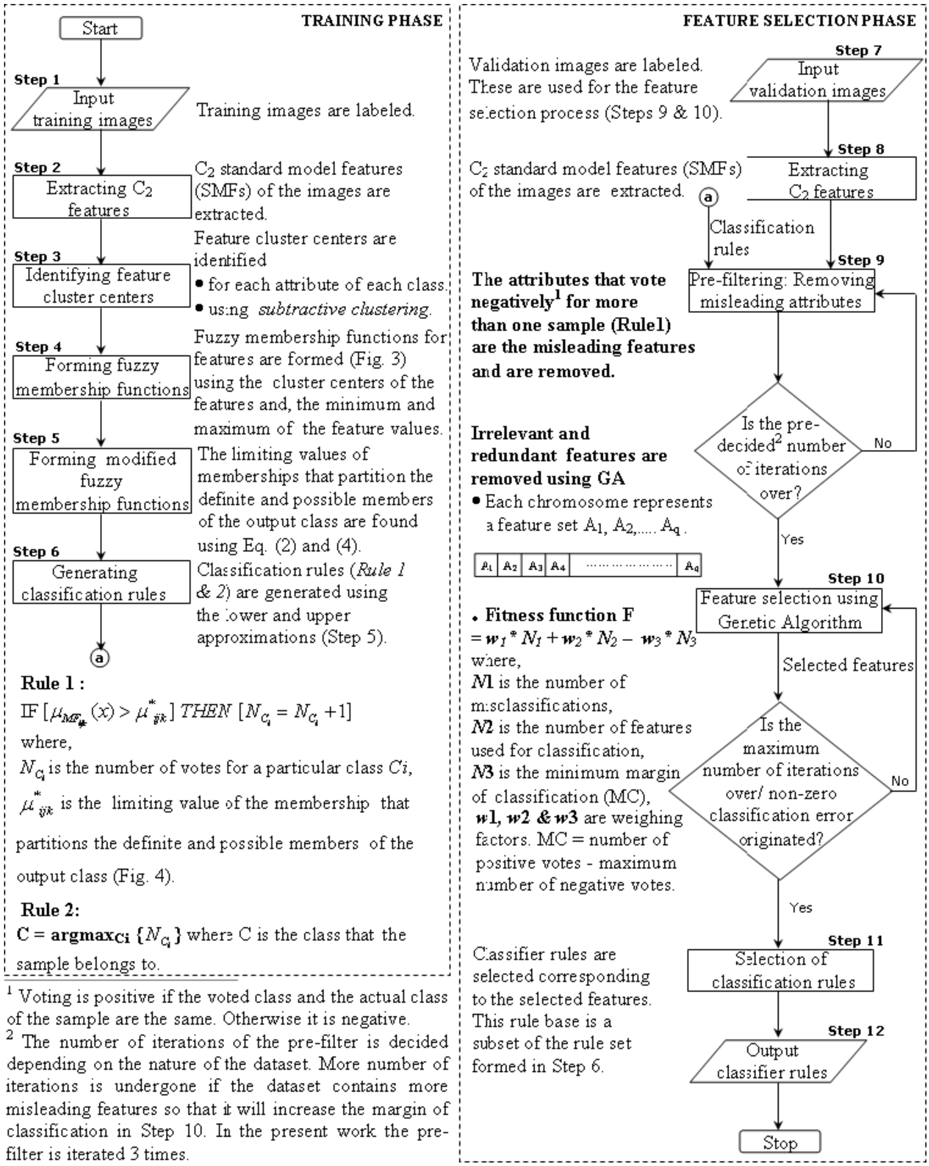


Fig. 7. Flowchart of the classifier development algorithm.

calculated using the membership functions formed in the training phase. This membership values are compared with the limiting value of the membership function (4) and the classification is done using *Rules* 1 and 2. Each of the execution steps of the classifier consists of a few number (which is equal to the number of cluster centers of the feature) of comparisons, using the classification rules, which makes the algorithm computationally simple. The classification results are discussed in Sec. 4.

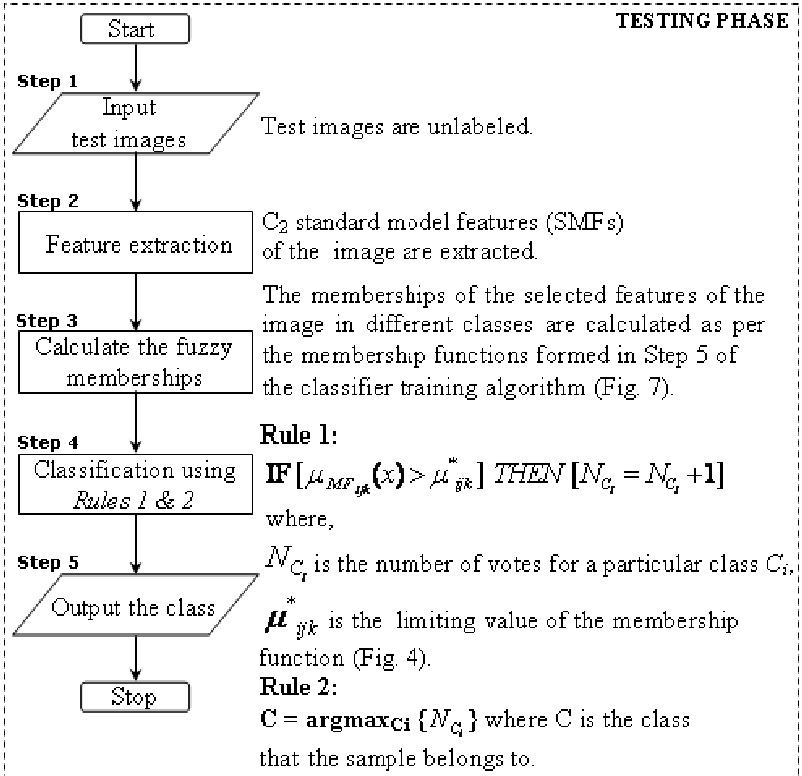


Fig. 8. Flowchart of the testing phase.

3.4. *Computational complexity analysis*

The different parameters at the input of the recognition system are the number of classes and the size of the input image. The complexity of the recognition algorithm has two components, complexity due to the feature extraction system and that due to the proposed classifier. This section discusses the computational complexity of the classifier algorithm.

The fuzzy-rough pattern classifier is formed using the classification rules generated in the training phase. The pseudo code of the classifier is shown in Fig. 9. Let p be the number of classes and q be the number of selected features at the input of the classifier. The complexity of the algorithm is as follows: $O(q)$ for reading the image features, $O(pq)$ for reading the trained parameters of the classification rules, $O(pq)$ for the fuzzy membership calculation and voting process, and $O(p)$ for finding the class index that received the maximum number of votes. Therefore, the overall complexity of the algorithm is polynomial time, $O(pq)$. As the feature extraction algorithm provides a fixed number of features, independent of the input image size, the effective complexity of the classification algorithm reduces to $O(p)$.

```
read {test_image_features} ; % read the selected C2 features.
read {μ*}, {Amin}, {Amax}, {Acenter} ; % read the membership limits, min. & max.
                                       % values of features, and feature cluster centers.
for i = number_of_classes
for j = number_of_features;  % number of selected features.
    calculate {fuzzy_memberships} ; % calculation of μ.
        IF μ > μ*
            vote(i) = vote(i) +1;
        end
    end
end
class = index {max (vote)} ; % find the class index which received maximum votes.
print {class} ;
```

Fig. 9. Pseudo code of the classifier.

4. Experimental Evaluation

The proposed algorithm is tested using three face datasets and two hand posture datasets (Table 2). The datasets contain ten classes with equal number of samples. The dataset is divided into N subsets, with equal number of images. The images in the first subset are used for the development of the classifier while those in the remaining $N-1$ subsets are used for testing. The classifier development is then done using the second subset and the algorithm is tested using the remaining $N-1$ subsets. The experiments are repeated in a similar fashion, N times, until each of the subset is used for the development, and the average accuracy achieved over the N runs is reported. The values of N are two and four ($N = 2, 4$ corresponds to $50\%, 25\%$ images in one subset respectively). The data subset used for the classifier development is equally divided into training and validation sets. The classification results are compared with that of SVMs with polynomial kernel, implemented using LIBSVM.[70]

Table 2. Different datasets used.

Dataset	Source	# images
Face dataset-1	Subset of Yale face database B[29]	640
Face dataset-2	Subset of color FERET database[30]	240
Face dataset-3	Subset of CMU face dataset[31]	240
Hand posture dataset-1	NUS hand posture dataset	240
Hand posture dataset-2	Jochen Triesch hand posture dataset[32]	480

4.1. *Face recognition*

The proposed recognition algorithm is tested using three different face datasets that have variation in lighting direction (Yale face database B[29]), variation in pose (color FERET database[30]), and illumination variation (CMU face dataset[31]).

The first face dataset considered is a subset of the Yale face database B,[29] which contains ten classes of face images, taken from different lighting directions (Fig. 10(a)). It consists of 640 frontal face images. A comparison of the achieved results with that of SVM using C_2 features is provided in Table 3. The fuzzy-rough classifier provided full classification of the dataset, whereas SVM provided 99.38% accuracy, when $N = 2$. The proposed algorithm classified the dataset utilizing only 86 number of features, whereas SVM utilized all the 1000 features.

The second face dataset used is a subset of the color FERET database.[30] The subset contains ten classes of face images, with variations in pose (Fig. 10(b)). It

(a)

(b)

Fig. 10. Sample images from (a) Yale face dataset, (b) FERET face dataset, and (c) CMU face dataset.

(c)

Fig. 10. (*Continued*)

Table 3. Recognition results — face datasets.

Dataset	No. of Training Samples	FRC*		SVM	
		Accuracy (%)	No. of Features	Accuracy (%)	No. of Features
A subset of Yale	160 ($N = 4$)	99.16	95	98.33	
face dataset[29]	320 ($N = 2$)	100.00	86	99.38	
A subset of color	60 ($N = 4$)	88.89	111	88.33	
FERET database[30]	120 ($N = 2$)	91.66	102	90.83	1000
A subset of CMU	60 ($N = 4$)	99.44	62	98.88	
face dataset[31]	120 ($N = 2$)	100.00	55	100.00	

*Fuzzy-rough classifier.

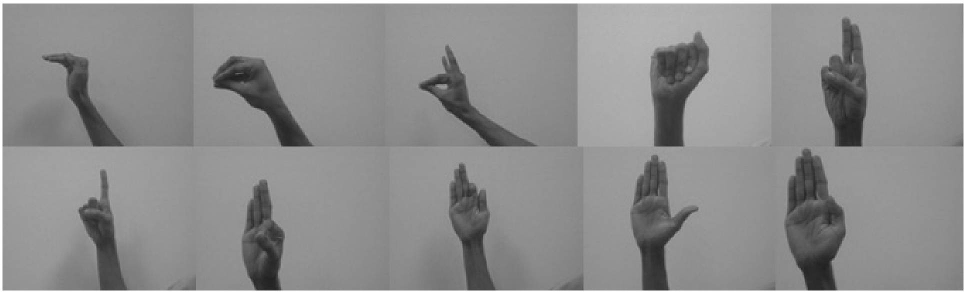
consists of 240 images. A comparison of the results with that of SVM is provided in Table 3. The proposed algorithm needed only 102 features (when $N = 2$) for the classification and it provided better accuracy compared to SVM classifier.

The third face dataset considered is a subset of the CMU face dataset.[31] It has good amount of illumination variation as shown in Fig. 10(c). The dataset consists of 240 frontal face images (24 from each class). The recognition results are provided in Table 3. Both the proposed algorithm and the SVM classifier classified the dataset fully. However, the proposed algorithm did the classification with lesser number of features.

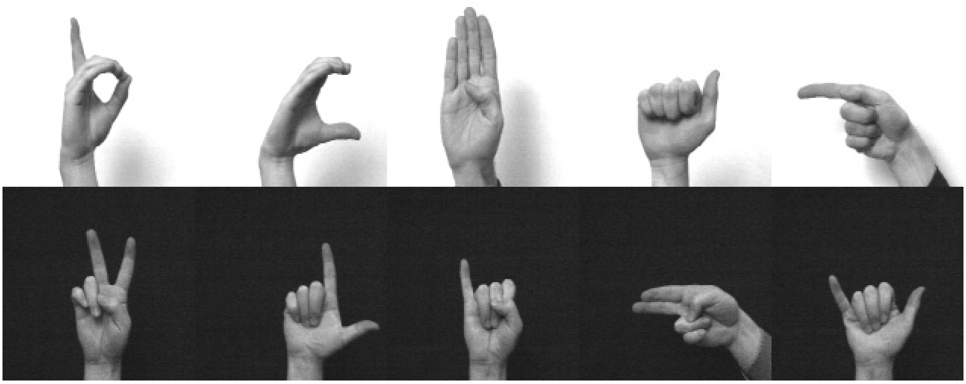
4.2. *Hand posture recognition*

As the number of available hand posture datasets is limited, a new dataset namely the NUS hand posture dataset,[b] with ten classes of hand postures, is created. It

[b]This dataset is available on an e-mail request to prahlad@nus.edu.sg.



(a)



(b)

Fig. 11. Sample hand posture images from (a) NUS dataset and (b) Jochen Triesch dataset.

consists of 24 sample images per posture, which are captured by varying the position and size of the hand within the image frame. These are gray-scale images (160×120 pixels). The hand postures are selected in such a way that the inter class variation in the appearance of the postures is less, which makes the recognition task more challenging. Sample images are shown in Fig. 11(a). The better performance of the proposed algorithm is evident from the recognition results (Table 4). The

Table 4. Recognition results — hand posture datasets.

Dataset	No. of Training Samples	FRC*		SVM	
		Accuracy (%)	No. of Features	Accuracy (%)	No. of Features
NUS hand posture dataset	60 ($N = 4$)	91.66	98	91.80	
	120 ($N = 2$)	93.33	92	92.50	
Jochen Triesch hand posture dataset[17]	120 ($N = 4$)	95.83	72	94.44	1000
	240 ($N = 2$)	98.75	63	97.91	

*Fuzzy-rough classifier.

proposed algorithm utilized only 92 number of features (when $N = 2$) for the classification. The misclassifications occurred are mostly between the classes that are similar.

The second hand posture dataset considered is Jochen Triesch hand posture dataset.[32] The ten classes of hand postures (Fig. 11(b)), performed with 24 different persons, against light and dark backgrounds, were considered for the classification (total of 480 images). The images vary in size of the hand and shape of the postures. The proposed algorithm achieved better classification accuracy on comparison with SVM (Table 4). The classification is done utilizing 63 selected features (when $N = 2$), whereas the SVM classifier utilized all 1000 features.

4.3. *Online implementation and discussion*

The proposed recognition system is implemented online in Windows platform to test the computational performance. The algorithm is developed using the NUS hand posture dataset for the recognition of hand postures. The image to be recognized is accessed using a web-camera, resized and converted to gray scale, similar to the training images.

The algorithm is tested by showing each hand posture 20 times. The hand postures are performed by different persons, with variation in size and shape of the hand posture, and with different lighting conditions. The algorithm recognized the postures with an accuracy of 94.5%.

Table 5 provides a comparison of the average computational time of the proposed algorithm with that of SVM. The total time for recognition is divided into two, the time for the feature extraction, and the time for the classification. In the SVM classifier, all the 1000 features are utilized for the classification, and so 1000 features are extracted. The proposed algorithm needs only a subset of the 1000 features. This reduced both feature extraction and classification time (the proposed classifier took lesser time for classification even when 1000 features are utilized).

Each of the extracted C_2 SMF corresponds to a particular prototype patch with a specific patch size, as explained in Sec. 2.2. The selection of relevant features identified the prototype patches those are good in the inter class discrimination. This enhanced the shape selectivity. In addition, it decreased the processing time for the feature extraction, which is a major limitation of the feature extraction algorithm proposed by Serre *et al.*[28] The significant reduction in the feature extraction and classification time makes the algorithm suitable for real-time applications.

Table 5. Comparison of computational time.

	Proposed Algorithm	SVM
Feature extraction	1.92 s	7.43 s
Classification	0.96 ms	3.24 ms

5. Conclusion

A hand posture and face recognition algorithm using C_2 SMFs, and based on the concepts of lower and upper approximations of fuzzy equivalence classes, is proposed. The fuzzy membership functions and the corresponding classification rules are generated from the training images and the classification is done by a simple voting process. The predictive features in the dataset are selected using a GA-based feature selection algorithm. The proposed fitness function reduces the number of features required for the classification, without compromising on the classification accuracy. The performance of the algorithm is evaluated with some well-known datasets. The recognition results are compared with that of an SVM classifier. The proposed algorithm provided good recognition accuracy for all the datasets considered, at a less computational cost, which makes it suitable for real-time applications.

Acknowledgments

Portions of the research in this article use the FERET database of facial images collected under the FERET program, sponsored by the DOD Counterdrug Technology Development Program Office. We also used the Yale face database, CMU face database, and Jochen Triesch hand posture database for this research.

References

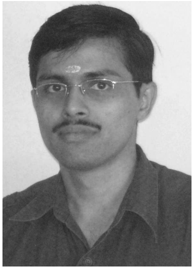
1. K. H. Lee and J. H. Kim, An hmm based threshold model approach for gesture recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21**(10) (1999) 961–973.
2. H. S. Yoon, J. Soh, Y. J. Bae and H. S. Yang, Hand gesture recognition using combined features of location, angle and velocity, *Pattern Recognition* **34** (2001) 1491–1501.
3. A. Ramamoorthy, N. Vaswani, S. Chaudhury and S. Banerjee, Recognition of dynamic hand gestures, *Pattern Recognition* **36** (2003) 2069–2081.
4. A. Just and S. Marcel, A comparative study of two state-of-the-art sequence processing techniques for hand gesture recognition, *Computer Vision and Image Understanding* (2009), doi:10.1016/j.cviu.2008.12.001.
5. F. S. Chen, C. M. Fu and C. L. Huang, Hand gesture recognition using a real-time tracking method and hidden markov models, *Image and Vision Computing* **21** (2003) 745–758.
6. H. D. Yang, A. Y. Park and S. W. Lee, Gesture spotting and recognition for human-robot interaction, *IEEE Transactions on Robotics* **23**(2) (2007) 256–270.
7. M. C. Su, A fuzzy rule-based approach to spatio-temporal hand gesture recognition, *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Application and Reviews* **30**(2) (2000) 276–281.
8. A. Licsar and T. Sziranyi, User-adaptive hand gesture recognition system with interactive training, *Image and Vision Computing* **23** (2005) 1102–1114.
9. S. S. Ge, Y. Yang and T. H. Lee, Hand gesture recognition and tracking based on distributed locally linear embedding, *Image and Vision Computing* **26** (2008) 1607–1620.
10. M. H. Yang, N. Ahuja and M. Tabb, Extraction of 2D motion trajectories and its application to hand gesture recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(8) (2002) 1061–1074.

11. M. Zhao, F. K. H. Quek and X. Wu, Rievl: Recursive induction learning in hand gesture recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(11) (1998) 1174–1185.
12. X. Teng, B. Wu, W. Yu and C. Liu, A hand gesture recognition system based on local linear embedding, *Journal of Visual Languages & Computing* **16** (2005) 442–454.
13. M. Hasanuzzamana, T. Zhanga, V. Ampornaramveth, H. Gotoda, Y. Shirai and H. Ueno, Adaptive visual gesture recognition for human-robot interaction using a knowledge-based software platform, *Robotics and Autonomous Systems* **55**(8) (2007) 643–657.
14. M. Hasanuzzaman, V. Ampornararamveth, T. Zhang, M. A. Bhuiyan, Y. Shirai and H. Ueno, Real-time vision-based gesture recognition for human robot interaction, in *IEEE International Conference on Robotics and Biomimetics* (*ROBIO 2004*) (Shenyang, P. R. China, 2004), pp. 413–418.
15. Y. T. Chen and K. T. Tseng, Developing a multiple-angle hand gesture recognition system for human machine interactions, in *33rd Annual Conference of the IEEE-Industrial-Electronics-Society* (Taipei, Taiwan, 2007), pp. 489–492.
16. J. Triesch and C. Malsburg, A system for person-independent hand posture recognition against complex backgrounds, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23**(12) (2001) 1449–1453.
17. J. Triesch and C. Malsburg, Robust classification of hand postures against complex backgrounds, in *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition, 1996* (Killington, VT, USA, October, 1996), pp. 170–175.
18. J. Triesch and C. Malsburg, A gesture interface for human-robot-interaction, in *Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, 1998* (Nara, Japan, April, 1998), pp. 546–551.
19. A. Shamaie and A. Sutherland, Graph-based matching of occluded hand gestures, in *30th Applied Imagery Pattern Recognition Workshop, AIPR 2001* (Washington, DC, USA, 2001), pp. 67–73.
20. E. Ueda, Y. Matsumoto, M. Imai and T. Ogasawara, A hand-pose estimation for vision-based human interfaces, *IEEE Transactions on Industrial Electronics* **50**(4) (2003) 676–684.
21. X. Yin and M. Xie, Estimation of the fundamental matrix from uncalibrated stereo hand images for 3D hand gesture recognition, *Pattern Recognition* **36** (2003) 567–584.
22. Q. Chen, N. D. Georganas and E. M. Petriu, Hand gesture recognition using haar-like features and a stochastic context-free grammar, *IEEE Transactions on Instrumentation and Measurement* **57**(8) (2008) 1562–1571.
23. W. H. A. Wang and C. L. Tung, Dynamic hand gesture recognition using hierarchical dynamic Bayesian networks through low-level image processing, in *7th International Conference on Machine Learning and Cybernetics* (Kunming, P. R. China, 2008), pp. 3247–3253.
24. K. S. Patwardhan and S. D. Roy, Hand gesture modelling and recognition involving changing shapes and trajectories, using a predictive eigentracker, *Pattern Recognition Letters* **28** (2007) 329–334.
25. M. H. Yang, D. J. Kriegman and N. Ahuja, Detecting faces in images: A survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(1) (2002) 34–58.
26. Y. S. Gao and M. K. H. Leung, Face recognition using line edge map, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(6) (2002) 764–779.
27. R. Chellappa, C. L. Wilson and S. Sirohey, Human and machine recognition of faces — a survey, *Proceedings of the IEEE* **83**(5) (1995) 705–740.

28. T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber and T. Poggio, Robust object recognition with cortex-like mechanisms, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(3) (2007) 411–426.
29. A. S. Georghiades, P. N. Belhumeur and D. J. Kriegman, From few to many: Illumination cone models for face recognition under variable lighting and pose, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23**(6) (2001) 643–660.
30. P. J. Phillips, H. Wechsler, J. Huang and P. Rauss, The FERET database and evaluation procedure for face recognition algorithms, *Image and Vision Computing* **16**(5) (1998) 295–306.
31. T. Sim, S. Baker and M. Bsat, The CMU pose, illumination, and expression database, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(12) (2003) 1615–1618.
32. J. Triesch and C. Malsburg, Sebastien Marcel hand posture and gesture datasets: Jochen Triesch static hand posture database, Available at http://www.idiap.ch/resources/gestures/.
33. L. A. Zadeh, Fuzzy sets, *Information and Control* **8**(3) (1965) 338–353.
34. Z. Pawlak, Rough sets, *International Journal of Computer and Information Science* **11** (1982) 341–356.
35. Z. Pawlak, Rough classification, *International Journal of Man-Machine Studies* **20** (1984) 469–483.
36. Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning About Data* (Dordrecht: Kluwer Academic Publishers, 1991).
37. D. Dubois and H. Prade, Rough fuzzy sets and fuzzy rough sets, *International Journal of General Systems* **17** (1990) 191–209.
38. D. Dubois and H. Prade, Putting rough sets and fuzzy sets together, in R. Slowinski (Ed.), *Intelligent Decision Support: Handbook of Applications and Advances in Rough Sets Theory*, ser. Series D: System Theory, Knowledge Engineering and Problem Solving (Dordrecht, the Netherlands: Kluwer Academic Publishers, 1992), Vol. 11, pp. 203–232.
39. T. J. Li and W. X. Zhang, Rough fuzzy approximations on two universes of discourse, *Information Sciences* **178** (2008) 892–906.
40. M. Sarkar, Rough fuzzy functions in classification, *Fuzzy Sets and Systems* **132** (2002) 353–369.
41. Y. Y. Yao, Constructive and algebraic methods of the theory of rough sets, *Information Sciences* **109** (1998) 21–47.
42. W. Z. Wu, J. S. Mi and W. X. Zhang, Generalized fuzzy rough sets, *Information Sciences* **15** (2003) 263–282.
43. A. M. Radzikowska and E. E. Kerre, A comparative study of fuzzy rough sets, *Fuzzy Sets and Systems* **126** (2002) 137–155.
44. N. N. Morsi and M. Yakout, Axiomatics of fuzzy rough sets, *Fuzzy Set and Systems* **100** (1998) 327–342.
45. M. Sarkar, Fuzzy-rough nearest neighbor algorithms in classification, *Fuzzy Sets and Systems* **158** (2007) 2134–2152.
46. R. Jensen and C. Cornelis, A new approach to fuzzy-rough nearest neighbour classification, in *Proceedings of the 6th International Conference on Rough Sets and Current Trends in Computing* (2008), pp. 310–319.
47. X. Wang, J. Yang, X. Teng and N. Peng, Fuzzy-rough set based nearest neighbor clustering classification algorithm, *Lecture Notes in Computer Science* **3613** (2005) 370–373.

48. T. P. Hong, T. T. Wang and S. L. Wang, Knowledge acquisition from quantitative data using the rough-set theory, *Intelligent Data Analysis* **4** (2000) 289–304.
49. T. P. Hong, T. T. Wang, S. L. Wang and B. C. Chien, Learning a coverage set of maximally general fuzzy rules by rough sets, *Expert Systems with Applications* **19** (2000) 97–103.
50. Q. Shen and A. Chouchoulas, A rough-fuzzy approach for generating classification rules, *Pattern Recognition* **35** (2002) 2425–2438.
51. A. Roy and K. P. Sankar, Fuzzy discretization of feature space for a rough set classifier, *Pattern Recognition Letters* **24** (2003) 895–902.
52. J. P. Jones and L. A. Palmer, An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex, *Journal of Neurophysiology* **58**(6) (1987) 1233–1258.
53. J. G. Daugman, Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters, *Journal of the Society of America A* **2**(7) (1985) 1160–1169.
54. T. Serre, L. Wolf and T. Poggio, Object recognition with features inspired by visual cortex, in C. Schmid, S. Soatto and C. Tomasi (Eds.), *Conference on Computer Vision and Pattern Recognition* (San Diego, CA, 2005), pp. 994–1000.
55. M. Riesenhuber and T. Poggio, Hierarchical models of object recognition in cortex, *Nature Neuroscience* **2**(11) (1999) 1019–1025.
56. T. van der Zant, L. Schomaker and K. Haak, Handwritten-word spotting using biologically inspired features, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **30**(11) (2008) 1945–1957.
57. J. Lai and W. X. Wang, Face recognition using cortex mechanism and SVM, in C. Xiong, H. Liu, Y. Huang and Y. Xiong (Eds.), *1st International Conference on Intelligent Robotics and Applications* (Wuhan, P. R. China, 2008), pp. 625–632.
58. M. Goodale and A. Milner, Separate visual pathways for perception and action, *Trends in Neuroscience* **15** (1992) 20–25.
59. C. Bishop, *Neural Networks for Pattern Recognition* (Oxford University Press).
60. S. Chiu, Fuzzy model identification based on cluster estimation, *Journal of Intelligent and Fuzzy Systems* **2**(3) (1994) 18–28.
61. R. Yager and D. Filev, Generation of fuzzy rules by mountain clustering, *Journal of Intelligent and Fuzzy Systems* **2**(3) (1994) 209–219.
62. J. Yang and V. Honavar, Feature subset selection using a genetic algorithm, in H. Liu and H. Motoda (Eds.), *Feature Extraction, Construction and Selection: A Data Mining Perspective*, ser. The Kulwer International series in Engineering and Computer Science (Boston: Kluwer Academic Publishers, 1998), pp. 117–131.
63. Z. Pawlak, Rough sets and fuzzy sets, in *Proceedings of ACM, Computer Science Conference* (Nashville, Tennessee, 1995), pp. 262–264.
64. D. Tian, J. Keane and X. Zeng, Evaluating the effect of rough sets feature selection on the performance of decision trees, in *Granular Computing, 2006 IEEE International Conference* (2006), pp. 57–62.
65. W. Siedlecki and J. Sklansky, A note on genetic algorithms for large scale feature selection, *IEEE Transactions on Computers* **10** (1989) 335–347.
66. F. Brill, D. Brown and W. Martin, Fast genetic selection of features for neural network classifiers, *IEEE Transactions on Neural Networks* **3**(2) (1992) 324–328.
67. H. D. Kim, C. H. Park, H. C. Yang and K. B. Sim, Genetic algorithm based feature selection method development for pattern recognition, in *Proceedings of SICE-ICASE International Joint Conference 2006* (Bexco, Busan, Korea, 2006), pp. 1020–1025.

68. L. Rokach, Genetic algorithm-based feature set partitioning for classification problems, *Pattern Recognition* **41** (2008) 1676–1700.
69. J. Lu, T. Zhao and Y. Zhang, Feature selection based on genetic algorithm for image annotation, *Knowledge-Based Systems* (2008) doi:10.1016/j.knosys.2008.03.051.
70. C. C. Chang and C. J. Lin, LIBSVM: A library for support vector machines (2001). Available at http://www.csie.ntu.edu.tw/cjlin/libsvm.



P. Pramod Kumar graduated with a bachelor degree in instrumentation and control engineering from Calicut University, Kerala, India, in 2003, with university first rank. He worked as a junior research fellow in the Center for Development of Advanced Computing (C-DAC), from January 2004 to August 2004, and as a control and instrumentation engineer, in NTPC limited, from September 2004 to July 2007, both under the Government of India. He joined the National University of Singapore as a research scholar, in August 2007, for his doctoral studies. His research interests include computational intelligence, fuzzy-rough sets, pattern recognition, computational neuroscience, and human–robot interaction.

Prahlad Vadakkepat received the M.Tech. and Ph.D. degrees from the Indian Institute of Technology, Madras, India, in 1989 and 1996, respectively. From 1991 to 1996, he was a lecturer with the Regional Engineering College, Calicut (currently, the National Institute of Technology, Calicut), Kerala, India. From 1996 to 1998, he was with Korea Advanced Institute of Science and Technology, Daejeon, Korea, as a post-doctoral fellow. He is currently an associate professor with the National University of Singapore, Singapore. His research interests are in humanoid robotics, distributed robotic systems, evolutionary robotics, neuro-fuzzy controllers, and intelligent control techniques.

Dr. Vadakkepat is the Founder Secretary of the Federation of International Robot-Soccer Association [www.fira.net], where he is currently the FIRA General Secretary. He was appointed as the technical activity coordinator of IEEE Region 10 from 2001 to 2002. He is an associate editor of the *International Journal of Humanoid Robotics* since 2003. He is a member of the National Committee of Robotics, Confederation of Indian Industry since 2009.

Loh Ai Poh is currently the deputy head for undergraduate programmes at the Department of Electrical and Computer Engineering, National University of Singapore. She received her bachelor of engineering degree (Electrical, First Class Honours) from the University of Malaya at Kuala Lumpur in 1983, and her doctor of philosophy in Control from Oxford University, in 1986. Her post-graduate work was made possible by a scholarship from the Kuok Foundation. From 1986 to 1989, she was with the University of Auckland, New Zealand, first with a post-doctoral fellowship followed by a lectureship. She has been a lecturer at National University of Singapore since 1989. From 1994 to 1997, she spent three years at the Department of Mechanical Engineering, Massachusetts Institute of Technology, USA, as a visiting lecturer. Her research interests are mainly in the areas of relay feedback systems, nonlinear control, auto-tuning, and fault detection.