Gesture Recognition: A Survey

Sushmita Mitra, *Senior Member, IEEE*, and Tinku Acharya, *Senior Member, IEEE*

***Abstract*—Gesture recognition pertains to recognizing meaningful expressions of motion by a human, involving the hands, arms, face, head, and/or body. It is of utmost importance in designing an intelligent and efficient human–computer interface. The applications of gesture recognition are manifold, ranging from sign language through medical rehabilitation to virtual reality. In this paper, we provide a survey on gesture recognition with particular emphasis on hand gestures and facial expressions. Applications involving hidden Markov models, particle filtering and condensation, finite-state machines, optical flow, skin color, and connectionist models are discussed in detail. Existing challenges and future research possibilities are also highlighted.**

***Index Terms*—Face recognition, facial expressions, hand gestures, hidden Markov models (HMMs), soft computing, optical flow.**

I. Introduction

IN THE PRESENT day framework of interactive, intelligent computing, an efficient human–computer interaction is assuming utmost importance in our daily lives. Gesture recognition can be termed as an approach in this direction. It is the process by which the *gestures* made by the user are *recognized* by the receiver.

Gestures are expressive, meaningful body motions involving physical movements of the fingers, hands, arms, head, face, or body with the intent of: 1) conveying meaningful information or 2) interacting with the environment. They constitute one interesting small subspace of possible human motion. A gesture may also be perceived by the environment as a compression technique for the information to be transmitted elsewhere and subsequently reconstructed by the receiver. Gesture recognition has wide-ranging applications [1] such as the following:

- developing aids for the hearing impaired;
- enabling very young children to interact with computers;
- designing techniques for forensic identification;
- recognizing sign language;
- medically monitoring patients' emotional states or stress levels;
- lie detection;
- navigating and/or manipulating in virtual environments;
- communicating in video conferencing;
- distance learning/tele-teaching assistance;
- monitoring automobile drivers' alertness/drowsiness levels, etc.

Generally, there exist many-to-one mappings from concepts to gestures and vice versa. Hence, gestures are *ambiguous* and *incompletely specified*. For example, to indicate the concept "stop," one can use gestures such as a raised hand with palm facing forward, or, an exaggerated waving of both hands over the head. Similar to speech and handwriting, gestures vary between individuals, and even for the same individual between different instances.

There have been varied approaches to handle gesture recognition [2], ranging from mathematical models based on hidden Markov chains [3] to tools or approaches based on soft computing [4]. In addition to the theoretical aspects, any practical implementation of gesture recognition typically requires the use of different imaging and tracking devices or gadgets. These include instrumented gloves, body suits, and marker-based optical tracking. Traditional 2-D keyboard-, pen-, and mouse-oriented graphical user interfaces are often not suitable for working in virtual environments. Rather, devices that sense body (e.g., hand, head) position and orientation, direction of gaze, speech and sound, facial expression, galvanic skin response, and other aspects of human behavior or state can be used to model communication between a human and the environment.

Gestures can be static (the user assumes a certain pose or configuration) or dynamic (with prestroke, stroke, and poststroke phases). Some gestures also have both static and dynamic elements, as in sign languages. Again, the automatic recognition of natural continuous gestures requires their temporal segmentation. Often one needs to specify the start and end points of a gesture in terms of the frames of movement, both in time and in space. Sometimes a gesture is also affected by the context of preceding as well as following gestures. Moreover, gestures are often language- and culture-specific. They can broadly be of the following types:

1) *hand and arm gestures*: recognition of hand poses, sign languages, and entertainment applications (allowing children to play and interact in virtual environments);
2) *head and face gestures*: some examples are: a) nodding or shaking of head; b) direction of eye gaze; c) raising the eyebrows; d) opening the mouth to speak; e) winking, f) flaring the nostrils; and g) looks of surprise, happiness, disgust, fear, anger, sadness, contempt, etc.;
3) *body gestures*: involvement of full body motion, as in: a) tracking movements of two people interacting outdoors; b) analyzing movements of a dancer for generating matching music and graphics; and c) recognizing human gaits for medical rehabilitation and athletic training.

Typically, the meaning of a gesture can be dependent on the following:

- spatial information: where it occurs;
- pathic information: the path it takes;

- symbolic information: the sign it makes;
- affective information: its emotional quality.

Facial expressions involve extracting sensitive features (related to emotional state) from facial landmarks such as regions surrounding the mouth, nose, and eyes of a normalized image. Often dynamic image frames of these regions are *tracked* to generate suitable features. The location, intensity, and dynamics of the facial actions are important for recognizing an expression. Moreover, the intensity measurement of spontaneous facial expressions is often more difficult than that of *posed* facial expressions. More subtle cues such as hand tension, overall muscle tension, locations of self-contact, and pupil dilation are sometimes used.

In order to determine all these aspects, the human body position, configuration (angles and rotations), and movement (velocities) need to be sensed. This can be done either by using sensing devices attached to the user. Those may be magnetic field trackers, instrumented (data) gloves, and body suits, or by using cameras and computer vision techniques.

Each sensing technology varies along several dimensions, including accuracy, resolution, latency, range of motion, user comfort, and cost. Glove-based gestural interfaces typically require the user to wear a cumbersome device and carry a load of cables connecting the device to a computer. This hinders the ease and naturalness of the user's interaction with the computer. Vision-based techniques, while overcoming this, need to contend with other problems related to occlusion of parts of the user's body. While tracking devices can detect fast and subtle movements of the fingers when the user's hand is moving, a vision-based system will at best get a general sense of the type of finger motion. Again, vision-based devices can handle properties such as texture and color for analyzing a gesture, while tracking devices cannot. Vision-based techniques can also vary among themselves in: 1) the number of cameras used; 2) their speed and latency; 3) the structure of environment (restrictions such as lighting or speed of movement); 4) any user requirements (whether user must wear anything special); 5) the low-level features used (edges, regions, silhouettes, moments, histograms); 6) whether 2-D or 3-D representation is used; and 7) whether time is represented. There is, however, an inherent loss in information whenever a 3-D image is projected to a 2-D plane. Again, elaborate 3-D models involve prohibitive high-dimensional parameter spaces. A tracker also needs to handle changing shapes and sizes of the gesture-generating object (that varies between individuals), other moving objects in the background, and noise. Good review on human motion analysis is available in literature [5], [6].

In this paper, we provide a survey on different aspects of gesture recognition. Section II outlines various tools often used for gesture recognition. Section III is devoted to hand and arm gestures, with particular emphasis on hidden Markov models (HMMs), particle filtering and condensation, finite-state machine (FSM), and neural network. This is followed by facial gesture recognition in Section IV including a coverage on approaches employing HMMs, principal component analysis (PCA), contour models, feature extraction, Gabor filtering, optical flow, skin color, and connectionist models. Finally, Section V indicates a few of the existing challenges and future research possibilities.

II. Tools for Gesture Recognition

Gesture recognition is an ideal example of multidisciplinary research. There are different tools for gesture recognition, based on the approaches ranging from statistical modeling, computer vision and pattern recognition, image processing, connectionist systems, etc. Most of the problems have been addressed based on statistical modeling, such as PCA, HMMs [3], [7], [8], Kalman filtering [9], more advanced particle filtering [10], [11] and condensation algorithms [12]–[14]. FSM has been effectively employed in modeling human gestures [15]–[18].

Computer vision and pattern recognition techniques [19], involving feature extraction, object detection, clustering, and classification, have been successfully used for many gesture recognition systems. Image-processing techniques [20] such as analysis and detection of shape, texture, color, motion, optical flow, image enhancement, segmentation, and contour modeling [21], have also been found to be effective. Connectionist approaches [22], involving multilayer perceptron (MLP), time-delay neural network (TDNN), and radial basis function network (RBFN), have been utilized in gesture recognition as well.

While static gesture (pose) recognition can typically be accomplished by template matching, standard pattern recognition, and neural networks, the dynamic gesture recognition problem involves the use of techniques such as time-compressing templates, dynamic time warping, HMMs, and TDNN. In the rest of this section, we discuss the principles and background of some of these popular tools used in gesture recognition.

A. HMM

A time-domain process demonstrates a Markov property if the conditional probability density of the current event, given all present and past events, depends only on the jth most recent event. If the current event depends solely on the most recent past event, then the process is termed a first order Markov process. This is a useful assumption to make, when considering the positions and orientations of the hands of a gesturer through time.

The HMM [3], [7] is a double stochastic process governed by: 1) an underlying Markov chain with a finite number of states and 2) a set of random functions, each associated with one state. In discrete time instants, the process is in one of the states and generates an observation symbol according to the random function corresponding to the current state. Each transition between the states has a pair of probabilities, defined as follows:

1) transition probability, which provides the probability for undergoing the transition;
2) output probability, which defines the conditional probability of emitting an output symbol from a finite alphabet when given a state.

The HMM is rich in mathematical structures and has been found to efficiently model spatio–temporal information in a natural way. The model is termed "hidden" because all that can be

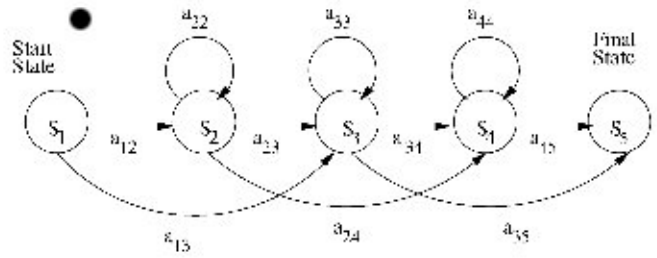

Fig. 1. Five-state left-to-right HMM for gesture recognition.

seen is only a sequence of observations. It also involves elegant and efficient algorithms, such as Baum–Welch and Viterbi [23], for evaluation, learning, and decoding. An HMM is expressed as $\lambda = (A, B, \Pi)$ and is described as follows:

- a set of observation strings $O = \{O_1, \ldots, O_T\}$, where $t = 1, \ldots, T$;
- a set of N states $\{s_1, \ldots, s_N\}$;
- a set of k discrete observation symbols $\{v_1, \ldots, v_k\}$;
- a state-transition matrix $A = \{a_{ij}\}$, where a_{ij} is the transition probability from state s_i at time t to state s_j at time $t+1$

$$A = \{a_{ij}\} = \text{Prob}(s_j \text{ at } t+1 | s_i \text{ at } t), \quad \text{for } 1 \le i, j \le N. \tag{1}$$

- an observation symbol probability matrix $B = \{b_{jk}\}$, where b_{jk} is the probability of generating symbol v_k from state s_j;
- an initial probability distribution for the states

$$\Pi = \{\pi_j\}, j = 1, 2, \ldots, N, \text{ where } \pi_j = \text{Prob}(s_j \text{ at } t = 1).$$

The generalized topology of an HMM is a fully connected structure, known as an *ergodic* model, where any state can be reached from any other state. When employed in dynamic gesture recognition, the state index transits only from left to right with time, as depicted in Fig. 1. The start state s_1 and final state s_N, for $N = 5$, are indicated on the figure. Here, the state-transition coefficients $a_{ij} = 0$ if $j < i$, and $\sum_{j=1}^{N} a_{ij} = 1$. The Viterbi algorithm is used for evaluating a set of HMMs and decoding by considering only the maximum path at each time step instead of all paths.

The global structure of the HMM is constructed by parallel connections of each HMM $(\lambda_1, \lambda_2, \ldots, \lambda_M)$, whereby insertion (or deletion) of a new (or existing) HMM is easily accomplished. Here, λ corresponds to a constructed HMM model for each gesture, where M is the total number of gestures being recognized.

HMMs have been applied to hand and face recognition. Usually, a 2-D projection is taken from the 3-D model of the hand or face, and a set of input features are extracted experimentally. The spatial component of the dynamic gesture is typically neglected, while the temporal component (having a start state, end state, and a set of observation sequences) is mapped through an HMM classifier with appropriate boundary conditions. A set of data are employed to train the classifier, and the test data are used for prediction verification.

Given an observation sequence, the following are the key issues in HMM use:

1) *evaluation*: determining the probability that the observed sequence was generated by the model (Forward–Backward algorithm);
2) *training or estimation*: adjusting the model to maximize the probabilities (Baum–Welch algorithm);
3) *decoding*: recovering the state sequence (Viterbi algorithm).

B. Particle Filtering and Condensation Algorithm

Particle-filtering-based tracking and its applications in gesture recognition systems became popular very recently [10]–[14]. Particle filters have been very effective in estimating the state of dynamic systems from sensor information. The key idea is to represent probability densities by set of samples. As a result, it has the ability to represent a wide range of probability densities, allowing real-time estimation of nonlinear, non-Gaussian dynamic systems. This technique was originally developed to effectively track objects in clutter [12], [13]. The state of a tracked object at time t is described by a vector X_t, where the vector Y_t represents all the samples of *observations* $\{y_1, y_2, \cdots, y_t\}$. The posterior density $P(X_t|Y_t)$ and the observation density $P(Y_t|X_t)$ are often non-Gaussian.

Basically, the particle filters are a sample-based variant of Bayes filters. The key idea is to approximate the probability density distribution by a weighted sample set $S_t = \{\langle x_t^{(i)}, w_t^{(i)} \rangle | \ i = 1, \cdots, N_p\}$. Here, each sample $x_t^{(i)}$ represents a hypothetical state of the object, and $w_t^{(i)}$ represents the corresponding discrete sampling probability of the sample $x_t^{(i)}$ such that

$$\sum_{i=1}^{N_p} w_t^{(i)} = 1. \tag{2}$$

The particle filtering in its basic form actually realizes the recursive Bayes filter according to a sampling procedure, often called sequential importance sampling with resampling (SISR) [14]. The iterative evolution of the sample set is described by propagating each sample according to a system model. Each sample element in the set is weighted in terms of the observations, and N_p samples are drawn with replacement by choosing a particular sample with posterior probability $w_t^{(i)} = P(y_t | X_t = x_t^{(i)})$. In each step of iteration, the mean state (sample) of an object is estimated as

$$E(S_t) = \sum_{i=1}^{N_p} w_t^{(i)} x_t^{(i)}.$$

Since it models uncertainty (as posterior probability density), particle filtering provides a robust tracking framework suitable for gesture recognition systems. Based on the above principle, the condensation algorithm (conditional density propagation over time) was originally proposed to deal with the problem of tracking rapid motion in clutter [13]. Rather than attempting to fit a specific equation to the observed sensory data, it uses the N_p weighted samples to approximate the curve described by the observed data. When applied to tracking, each sample

represents the *state* of the object being tracked, e.g., its velocity and location. Given such a randomly sampled state S_t at time t, a prediction of a new state S_{t+1} at time $t+1$ is made using a predictive model.

An extension of the condensation algorithm has been proposed [24] to automatically switch between various prediction motion models. Because of the usage of multiple such models to predict different types of motion of the objects, this significantly improves the performance of the tracker. It also enables the extraction of the most likely particular model that correctly represents the motion observed at a given time t. This is very relevant and useful for gesture recognition. If each competing model represents a single gesture, then the most likely model predicts which gesture is being observed.

C. FSM Approach

In the FSM approach, a gesture can be modeled as an ordered sequence of states in a spatio–temporal configuration space [15]–[18]. The number of states in the FSM may vary between applications. The gesture is recognized as a prototype trajectory from an unsegmented, continuous stream of sensor data constituting an ensemble of trajectories. The trajectories of the gestures are represented as a set of points (e.g., sampled positions of the head, hand, and eyes) in a 2-D space.

Usually, the training of the model is done off-line, using many possible examples of each gesture as training data, and the parameters (criteria or characteristics) of each state in the FSM are derived. The recognition of gestures can be performed online using the trained FSM. When input data (feature vectors such as trajectories) are supplied to the gesture recognizer, the latter decides whether to stay at the current state of the FSM or jump to the next state based on the parameters of the input data. If it reaches a final state, we say that a gesture has been recognized.

The state-based representation can be extended to accommodate multiple models for the representation of different gestures, or even different phases of the same gesture. Membership in a state is determined by how well the state models can represent the current observation. If more than one model (gesture recognizer) reach their final states at the same time, we can apply a winning criteria to choose the most probable gesture.

The concept of motion energy has been used [17] to extract the temporal signature of hand motion from a limited set of dynamic hand gestures. This is subsequently analyzed and interpreted by a deterministic FSM. The relative change of direction of motion, rather than the relative motion such as quickly or slowly, is considered to be more important for determining the temporal signature. Adaptation with cross-cultural gestures can be achieved by redefining the FSM according to the relevant rules of the society. Inclusion of new gestures is achieved by the construction of additional FSMs portraying the corresponding motion profile.

D. Soft Computing and Conectionist Approach

Soft computing is a consortium of methodologies that works synergistically and provides flexible information processing capability for handling real-life ambiguous situations [25]. Its aim is to exploit the tolerance for imprecision, uncertainty, approximate reasoning, and partial truth in order to achieve tractability, robustness, and low-cost solutions.

Sensor outputs are often associated with an inherent uncertainty. Relevant, sensor-independent, invariant features are extracted from these outputs, followed by gesture classification. The recognition system may be designed to be fully trained when in use, or may adapt dynamically to the current user. Soft computing tools [26], such as fuzzy sets, artificial neural networks (ANNs), genetic algorithms (GAs), and rough sets, hold promise in effectively handling these issues. Management of uncertainty at different levels can be made by fuzzy sets and rough sets, while dimensionality reduction is taken care of by self-organizing maps. Fuzzy sets can also be used to model simultaneous partial membership to multiple gesture classes.

The adaptive nature of ANNs enable connectionist approaches to incorporate learning in data-rich environment. This characteristic, coupled with robustness, is useful in developing recognition systems. The various connectionist models in literature [27]–[30], for hand gesture and facial expression recognition, include MLP, TDNN, and RBFN.

III. Hand and Arm Gestures

Human gestures typically constitute a space of motion expressed by the body, face, and/or hands. Of these, hand gestures are often the most expressive and the most frequently used. This involves: 1) a posture: static finger configuration without hand movement and 2) a gesture: dynamic hand movement, with or without finger motion. Gestures may be categorized as given in the following list, such that as we proceed downward this list, their association with speech declines, language properties increase, spontaneity decreases, and social regulation increases:

- *gesticulation*: spontaneous movement of hands and arms, accompanying speech. These spontaneous movements constitute around 90% of human gestures. People gesticulate when they are on telephone, and even blind people regularly gesture when speaking to one another;
- *languagelike gestures*: gesticulation integrated into a spoken utterance, replacing a particular spoken word or phrase;
- *pantomimes*: gestures depicting objects or actions, with or without accompanying speech;
- *emblems*: familiar signs such as "V for victory," or other culture-specific "rude" gestures;
- *ign languages*: well-defined linguistic systems. These carry the most semantic meaning and are more systematic, thereby being easier to model in a virtual environment.

Hand gesture recognition consists of *gesture spotting* that implies determining the start and end points of a meaningful gesture pattern from a continuous stream of input signals and, subsequently, segmenting the relevant gesture. This task is very difficult due to: 1) the *segmentation ambiguity* and 2) the *spatio–temporal variability* involved. As the hand motion switches from one gesture to another, there occur intermediate movements as well. These transition motions are also likely to be segmented and matched with reference patterns, and need to be eliminated

by the model. Moreover, the same gesture may dynamically vary in shape and duration even for the same gesturer.

Orientation histograms have also been used as a feature vector for fast, simple hand gesture classification and interpolation [31]. It is based on computing the Euclidean distance from the prototypes in terms of these features. The histogram of orientations, representing an orientation in terms of its angle, provides for translational invariance.

A. HMMs for Hand Gesture Recognition

HMM is a rich tool used for hand gesture recognition in diverse application domains. Probably, the first publication addressing the problem of hand gesture recognition is the celebrated paper by Yamato *et al*. [7]. In this approach, a discrete HMM and a sequence of vector-quantized (VQ)-labels have been used to recognize six classes of tennis strokes. Before applying the HMM, the image sequence goes through several preprocessing steps such as low-pass filtering to reduce the noise, background subtraction to extract the moving objects, and binarization of the moving objects in order to generate blobs. The blobs roughly represent the poses of the human. The features are the amounts of object (black) pixels. These features are vector quantized, such that the image sequence becomes a sequence of VQ-labels, which are then processed by a discrete HMM.

Subsequently, several other applications of hand gesture recognition have been developed based on HMMs. In the following sections, we elaborate some of these, as available in the literature.

1) Sign Language Recognition: Sign language is a visual language. Usually, sign language consists of three major components: 1) finger-spelling; 2) word-level sign vocabulary; and 3) nonmanual features [32]. The finger-spelling is used to spell words letter by letter. The word-level sign vocabulary is used for majority of communication, while the nonmanual features consist of facial expression, position of tongue, mouth, and body.

A real-time HMM-based system has been designed [33], [34] for recognizing sentence-level American Sign Language, without explicitly modeling the fingers. Since each gesture in a sign language has an already assigned meaning, strong rules of context and grammar may be applied to make the recognition tractable. Typically, the gestures represent whole words. A five-state HMM is used to recognize data strings, and is combined with statistical grammar to incorporate context during training and recognition. The Viterbi algorithm is used both with and without a strong grammar, based on the known forms of the sentences. However, once the recognizer starts, the subject must conduct only sign languages. This is because the model cannot distinguish undefined hand motions. The subject wears distinctly colored gloves on both hands, and sits in a chair in front of the camera to aid hand tracking. An eight-component feature vector consists of the hand's 2-D coordinates, axis angle of least inertia, and the eccentricity of the bounding ellipse. Considering that all human hands have approximately the same hue and saturation but vary in their brightness, in another approach, the hands are tracked based on skin tone. The leftmost and rightmost hands are assigned to be "left" and "right," respectively.

The HMM-based approach, described above, requires extensive training sets for modeling. It was found to be effective for practically just around 50 words and required heavily constrained artificial grammar on the structure of the sentences [33], [35]. Bowden *et al*. [32], [35] made significant progress in sign language interpretation by structuring the classification model around a linguistic definition of signed words, rather than an HMM. In this vision-based approach, the "visemes" of sign (*aka* phase representation of constituent motions) are defined in a manner similar to that in the sign linguistic dictionary.

Viseme is a visual equivalent of a phoneme. This enables signs to be learnt reliably from just a handful of training examples. The system is designed based on a novel two-stage classification. In the first stage, the raw image sequences are segmented in order to extract shapes and trajectories of the hands in the binary image sequence. These features are then converted into a viseme representation: 1) position of the hands relative to each other (HA); 2) position of hands relative to key body locations (TAB); 3) relative movement of the hands (SIG); and 4) the shape of the hands (DEZ). In the second stage of classification, each sign is modeled as a first order Markov chain in which each state in the chain represents a particular set of feature vectors generated from the classification in stage one. The Markov chain encodes temporal transitions of the hands. In order to allow minor variations over the sign instances and optimal feature-to-symbol mapping, independent component analysis (ICA) is employed to separate the correlated features from uncorrelated noise. Once the ICA transformation matrix has been learnt, a lookup table (LUT) is generated from the training data to map the ICA transformed features to symbols for use in Markov chain.

The salient feature of the above work [32], [35] is that the Markov chains can be built from as little as a single training example, or alternatively, a hand-coded description of the sign, due to generalization of the features. It still yields results comparable to the state-of-the-art.

2) Graphic Editor Control: Another HMM-based model [36] uses hand localization, hand tracking, and gesture spotting at preprocessing for hand gesture recognition. Hand candidate regions are located on the basis of skin color and motion. The centroids of the moving hand regions are connected to produce a hand trajectory, which is then divided into real and meaningless segments (categories). Input feature codes are extracted in terms of combined weighted location, angle, and velocity. This is followed by c-means clustering [37] to generate the HMM codebook. Left-to-right HMM with ten states is used for recognizing hand gestures to control a graphic editor. The gestures modeled include 12 graphic elements (circle, triangle, rectangle, arc, horizontal line, and vertical line) and 36 alphanumeric characters (ten Arabic numerals and 26 alphabets).

A charge-coupled device (CCD) camera placed in front of a monitor gives a sequence of gesture images from an image-capture board. The I and Q components of the YIQ color system are used here to extract hand areas from input images. *A priori* knowledge about the hand location of a previous video image, the usual face location, and the size of the hand region are used

to distinguish the hand region from multiple candidate regions. The basic hand location algorithm is outlined as follows:

1) color system conversion from RGB to YIQ;
2) estimation of similarity measures between model and input regions;
3) thresholding similarity measures;
4) noise removal and dilation;
5) detection of hand candidate regions;
6) selection of hand region.

Garbage movements that come before and after a pure gesture are removed by using a spotting rule, whereby the user intentionally stops for a while (2–3 s) at the start and end of the gesture. An eight-connectivity counterclockwise directional chain code is used to convert the orientation angles into feature codes. The velocity component takes care of the fact that while a simple "circle" gesture may have an almost nonvarying speed, a complex "q" or "w" gesture generation can involve varying speeds of movement.

3) Robot Control: A combination of static shape recognition, Kalman-filter-based hand tracking, and an HMM-based temporal characterization scheme is developed [38] for reliable recognition of single-handed dynamic hand gestures. Here, the start and end of gesture sequences are automatically detected. Hand poses can undergo motion and discrete changes in between the gesture, thereby enabling one to deal with a larger set of gestures. However, any continuous deformation of hand shapes is not allowed. The system is robust to background clutter, and uses skin color for static shape recognition and tracking. A real-time implementation is developed for robot control.

The user is expected to bring the hand to a designated region for initiating a gestural action to be grabbed by a camera. The hand should be properly illuminated, and the user should avoid sudden jerky movements. When the user moves the hand away from the designated region, it signals the end of the gesture and the beginning of the recognition process. The grabber and tracker are operated as synchronized threads. The five hand gestures and the corresponding instructions modeled for the robot are: 1) closed to open forward: move forward; 2) closed to open right: move forward then right; 3) closed to open left: move forward then left; 4) open to closed right: move backward then right; and 5) open to closed left: move backward then left.

Static hand shapes are described by their contours, specified through mouse clicks on the boundary of the image, and subsequently fitting a B-spline curve. Translated, scaled, and rotated versions of these shapes are also added to the prior. For a test shape, a matching is made with these stored priors. The closest contour match is chosen for tracker initialization.

A gesture is considered as a sequence of epochs, where each epoch is characterized by a motion of distinct hand shapes. Kalman filter is used for hand tracking, to obtain motion descriptors for the HMM. The moving hand is approximated as a planar rigid shape, under the assumption that the fingers are not being flexed, and the perspective effects are not significant. The left-to-right HMM, with four states and an out degree of three, proceeds by doing the following:

- extracting symbolic descriptors of the gesture at regular intervals from the tracker and hand shape classifier;
- training HMMs by the sequence of symbolic descriptors corresponding to each gesture;
- finding the model, which gives maximum probability of occurrence of the observation sequence generated by the test gesture.

The gesture recognition algorithm is outlined as follows.

1) Detect hand for boot-strapping the tracker.
2) Recognize the starting hand shape, and initialize tracker with its template.
3) **While** hand is in view **repeat**
   a) Track the hand and output encoded motion information **until** shape change is detected.
   b) Recognize the new shape and initialize the tracker with template of the recognized shape.
4) Using HMM, find the gesture, which gives the maximum probability of occurrence of observation sequence composed of shape templates and motion information.

B. Condensation Algorithm

The condensation algorithm was developed based on the principle of particle filtering. It was originally applied effectively in tracking rapid motion of objects in clutter [13]. A mixed-state condensation algorithm has been extended to recognize a greater number of gestures based on their temporal trajectories [39]. Here, one of the gesture models involves an augmented office white-board with which a user can make simple hand gestures to grab regions of the board, print them, save them, etc. In this approach, the authors allow compound models that are very like HMMs, with each state in the HMM being one of the defined trajectory models.

The other part deals with human facial expressions, using the estimated parameters of a learned model of mouth motion [39].

C. FSMs for Hand Gesture Recognition

As discussed in Section II-C, a gesture can be modeled as an ordered sequence of states in a spatio–temporal configuration space in the FSM approach. This has been used to recognize hand gestures [15], [17], [18].

A method to recognize human-hand gestures using a FSM-model-based approach has been used in [15]. The state machine is used to model four qualitatively distinct phases of a generic gesture—static start position (static at least for three frames), smooth motion of the hand and fingers until the end of the gesture, static end position for at least three frames, and smooth motion of the hand back to the start position. The hand gestures are represented as a list of gesture vectors and are matched with the stored gesture vector models based on vector displacements.

Another state-based approach to gesture learning and recognition has been presented in [18]. Here, each gesture is defined to be an ordered sequence of states, using spatial clustering and temporal alignment. The spatial information is first learned from a number of training images of the gestures. This information is used to build FSMs corresponding to each gesture. The FSM is

then used to recognize gestures from an unknown input image sequence.

In [17], a human performs a gesture in the field of view of a strategically placed camera, attached to a standard frame grabbing and digitizing hardware, against a simple stationary background. The gesture is started from any arbitrary spatio–temporal position and is a continuous stream of captured motion. The grabbing system captures the data with a spatial resolution of 256×256 pixels, at a rate of 15 frames/s. The gesture data are temporally segmented into subsequences involving uniform dynamics along a single direction (such as left, right, etc.). The system is highly reconfigurable, and no training concept is involved. The FSM has five states, viz., start (S), up (U), down (D), left (L), and right (R). The length of the signature pattern is independent of the duration over which the gesture is performed, but depends on the number of changes in the dominant direction of motion. Self-loops are essential to accommodate the idleness of hand movement while changing the direction of hand waving. Symbolic commands such as come closer, go far, move left, move right, and emergency stop are recognized. For example, come closer is modeled by repeatedly sweeping one hand toward the body and then slowly away (say, by S-D-U-U-D-U-D-D-U-D). Again, move right may be represented by moving the hand continuously toward the right direction and then the left (say, by S-L-R-R-L-L-R-L-R-L).

A lexicon is constructed to model the motion profile of the hand for each gesture. This knowledge is utilized for signature representation in the form of simple production rules, followed by their interpretation. These can be used as input to a robot programming language for generating machine-level instructions, in order to mimic the intended operation of the corresponding gesture.

D. Connectionist Approach to Hand Gesture Recognition

TDNN has been applied in [27] to recognize gestures related to American Sign Language (ASL). The gesture recognition system in this approach has been divided into two distinct steps. In the first step, multiscale motion segmentation is applied to track movements of objects between the frames. Regions between two consecutive frames are matched to obtain two-view correspondences. Then, *affine* transformations are computed to define pixel matches and recognize the motion regions or trajectories. In the second step, a TDNN is used to match the trajectory of movements to a given gesture model.

To recognize the sign language, only those object areas of motion where skin color is detected are determined first. Then, the detected regions of motion are merged until the shape of the merged region is either an ellipse or a rectangle. This is because sign languages are typically described by the relation between head (a large elliptical shape), palm (small elliptical shape), and (or) closed hand (a rectangular shape). The TDNN with two hidden layers is employed to classify the motion of various regions over time as a particular gesture (sign) in the sign language (ASL).

The experiment was done with a database of 40 complex hand gestures of ASL and each sign with around 38 instances on the average. Each video consists of an ASL sign which lasts for about 3 to 5 s at 30 frames/s with 160×120 pixels image resolution of each frame. The recognition rate achieved on the training and testing sets for gesture recognition are a high 99.02% and 96.21%, respectively.

IV. Face and Head Gestures

Face is a unique feature of a human being. Humans can detect and identify faces in a scene with little or no effort. Their robustness is tremendous, considering the large changes inherent in the visual stimulus due to: 1) viewing conditions (such as variation in luminance); 2) facial expression; 3) aging; 4) gender; 5) occlusion; or 6) distractions such as glasses, hair style or other disguise.

Human faces are nonrigid objects with a high degree of variability in size, shape, color, and texture. The goal of face detection is to efficiently identify and locate human faces regardless of their positions, scales, orientations, poses, and illumination. Any automated system for face and facial gesture recognition will have immense potential in criminal identification, surveillance, missing children retrieval, office security, credit card verification, video document retrieval, telecommunication, high-definition television (HDTV), medicine, human–computer interfaces, multimedia facial queries, and low-bandwidth transmission of facial data [40]–[42]. While frontal recognition is the classical approach, profile recognition schemes are practical for a fast, coarse presearch of large databases in order to reduce computational complexity for a subsequent sophisticated algorithm.

One needs to concentrate on the extraction of appropriate image attributes with useful query functionality, retrieval methods on similarity-based (instead of exact) match, query-by-image example (content-based image retrieval (CBIR)), query refinement, and high-dimensional database indexing. There are two major approaches to automated face recognition [43].

- *Analytic:* here flexible mathematical models are developed to incorporate face deformation and illumination changes. Discrete local (geometrical) features, such as irises and nostrils, are extracted for retrieving and identifying faces. The position of these features, with respect to one another, determine the overall location of the face. Standard statistical pattern recognition techniques such as HMMs [8], may be applied on these measurements. Other approaches include active contour models (Snakes) [21], wavelets [44], and knowledge- or rule-based techniques such as facial action coding system (FACS) [28], [45].
- *Holistic:* this involves gray-level template matching using global recognition. Here a feature vector is used to represent the entire face template. This approach includes ANNs [28]–[30], [46], [49], linear discriminants, PCA, singular value decomposition (SVD) using eigenfaces [48], [49], and optical flow [30], [50]–[52].

Research indicates that a fusion of both approaches often produces better (more stable) results, given an observation sequence, as compared to either approach alone [53]. Facial features can, again, be of two types [54]. They are: 1) permanent

facial features (such as eyes, eyebrows, lips, cheeks, tissue texture, facial hair, and permanent furrows) that are always present in the face, but may be deformed due to facial expressions and 2) transient facial features (such as wrinkles, furrows, and bulges) that occur in the forefront and the regions surrounding the mouth and eyes.

Variation between two face images can be twofold, viz.: 1) interpersonal—difference in appearance due to different identity and 2) intrapersonal—change in appearance of the same person due to different facial expressions or lighting. The first category corresponds to face recognition, while the second one pertains to facial expression recognition. Facial expressions can again be classified in terms of the following: 1) facial actions that cause an expression; 2) nonprototypic expressions such as "raised brows"; or 3) prototypic expressions such as in different emotions. It is interesting to note that the upper face features play a more important role in expression classification as compared to the lower face features.

The issue of CBIR has been addressed on the facial recognition technology (FERET)[1] facial database [47]. The FERET database consists of frontal shots as well as right and left profiles of human faces. A difficulty posed by this database is that the images were often taken at different times, locations, and under different imaging conditions. Moreover, the "gallery" subset (training set) and the "probe" subset (test set) were captured about a week apart and exhibit differences in clothing, hair, and lighting.

A. HMMs in Face Recognition

HMMs have been applied for interpersonal face recognition [8]. A sliding window is applied from the top to bottom of a 2-D image, while extracting the brightness value of the window as the 1-D feature vectors for the HMM. Successive windows are overlapped to avoid cutting of significant facial features and to provide contextual information for the sequence of feature vectors. The face is modeled as a linear left-to-right HMM, with five states corresponding to the important regions such as forehead, eyes, nose, mouth, and chin. The algorithm is outlined as follows.

- Build a code book from the feature vectors of a set of images.
- Quantize feature vectors from this code book.
- Train HMM for every person in the database.
- Recognize a test image, preprocessed using the first two steps.

B. PCA

The eigenfaces method [48] is based on the statistical representation of the face space. It finds the principal components (Karhunen–Loeve expansion) of the facial image distribution, or, the eigenvectors of the covariance matrix of the set of face images. These eigenvectors, representing a set of macrofeatures (that are generated *a posteriori* on a statistical basis) characterizing the face, constitute the eigenfaces. Fisher's linear discriminant is used to develop a set of feature vectors, constituting *Fisherfaces*, where the interpersonal variations are emphasized as compared to the intrapersonal variations.

Essa and Pentland [49] employed eigenfaces approximated by PCA, to locate faces in an arbitrary scene. The distance of an observed image from the face space is calculated using the projection coefficients and the signal energy. Facial motion is extracted using holistic dense optical flow, coupled with 3-D motion and *muscle-based* face models. Faces are detected from an image sequence by performing a spatio–temporal filtering, followed by thresholding, in order to analyze "motion blobs" that represent human heads. A control-theoretic method is used to extract the spatio–temporal motion-energy representation of facial motion for an observed expression. The Euclidean distance is computed from the "ideal" motion-energy templates for the six different expressions, involving two facial actions (viz., smile and raised eyebrows) and four emotional expressions (viz., surprise, disgust, sadness, and anger). A correct recognition rate of 98% is reported with 52 frontal-view image sequences of eight people.

C. FACS

FACS [45] was designed to help human observers detect independent subtle changes in facial appearance caused by contractions of the facial muscles. FACS provides linguistic rules describing all possible, visually detectable facial changes in terms of 44 action units (AUs). Thirty AUs are anatomically related to the contractions of specific facial muscles, *viz*, 12 for the upper face and 18 for the lower face. The AUs can occur either singly or in combination. For example, in FACS, an "inner brow raiser" corresponds to AU1 while a "jaw drop" refers to AU26. Using these rules, a trained human FACS coder decomposes a facial expression into specific AUs describing it. However, here it is not possible to classify an expression into multiple emotion categories; for example, raised eyebrows and smiling mouth being expressed as a blend of surprise and happiness.

D. Contour Models

Facial features such as lips, eyebrows, and nose are often extracted using active contour models (Snakes) [21]. This is an energy-minimizing spline guided by external constraint forces and affected by image forces that pull it toward features such as lines and edges. The local minima of the energy function correspond to the set of solutions. Addition of energy terms push the active model toward the desired solution when placed near it. Snakes are initialized by placing them on the facial features that are to be tracked. They lock onto nearby edges by localizing them accurately. and are able to deform and accurately track the features.

E. Facial Feature Extraction for Gesture Recognition

A combined approach is used for facial feature extraction and determination of gaze direction [55], as applied to facial expression recognition. The features under consideration here

[1]http://www.itl.nist.gov/iad/humanid/feret/

are the eyebrows, eyes, nostrils, mouth, cheeks, and chin. The model employs the following:

1) an improved variation of adaptive Hough transform [56] for geometrical shape parameterization, involving curve detection based on ellipse containing the main (oval) connected component of the image (related to cheeks and chin detection);
2) minima analysis of feature candidates corresponding to the low-intensity regions of the face (extracting eyes, nostrils, and mouth);
3) template matching for inner facial feature localization, using an appropriate binary mask on an area restricted by the eyes (extracting upper eyebrow edges that may not otherwise be uniformly described by a geometric curve);
4) dynamic deformation of active contours for inner face contour detection;
5) projective geometry properties for accurate pose determination, along with analysis of face symmetry properties for determination of gaze direction.

The skinlike regions in the image are detected from the hue-saturation-value (HSV) color space representation.

Several classes of perceptual cue to emotional states are displayed on the face. These include the following:

- the relative displacement of features [44] such as opening the mouth;
- quasi-textural changes in the skin surface such as furrowing the brow [44];
- changes in skin hue such as blushing;
- temporal aspect of these signals, involving motion [30], [51].

While motion extraction approaches directly focus on facial changes occurring due to facial expressions, the deformation-based methods rely on neutral face images (or models) to extract facial features that are relevant to facial actions and are not caused by (say) intransient wrinkles due to old age.

1) Unified Parameterized Facial Appearance Model for Recognition: Lanitis *et al.* [57] have designed a compact unified parameterized model of facial appearance that takes into account different sources of variability such as individual appearance, 3-D pose, facial expression, change in lighting condition, etc. The unified approach is very suitable for interpretation and coding of face images. There exist two main phases.

- *Modeling*: In this phase, flexible models of facial appearance are generated. The shapes of facial features and their spatial relationships are modeled from a set of training images. The model is generated by statistical analysis of the positions of landmark points over the training set in order to describe a mean shape from therein. This training set is representative of variations due to differences between individuals as well as changes in their expressions and 3-D poses. The shape model is augmented with the gray-level information of the face images. As a result, the shape and gray-level models represent the overall appearance of each face image. This is collectively called the appearance parameter of the face images.
- *Interpretation*: During this phase, the models are used for coding and interpretation of the face images. When a new query image is presented, the facial features are located automatically based on the flexible appearance model obtained during the training and modeling. These features are transformed into shape model parameters. The new query face is then deformed to the mean face shape, and the gray-level appearance is transformed into the appearance parameters. Hence, the new face image is coded in terms of these appearance parameters only. This resulting appearance parameters can then be used for personal identification including gender recognition, expression recognition, 3-D pose recovery as well as reconstruction of the face images.

Less than 100 parameters are required to describe each image sufficiently well to generate a good quality reconstruction of the face, inspite of different types of variability. Experimental results are presented for a database of 690 face images including partially occluded test faces. This unified approach is very generic in nature and can be easily adopted for different applications. A similar strategy can also be employed in automatic interpretation of hand gestures, sign reading, lip-reading, etc. [57].

Active appearance model (AAM) is a generative model of a certain visual phenomenon. Although linear in both shape and appearance, yet overall, the AAMs are nonlinear parametric models in terms of the pixel intensities. Fitting an AAM to an image consists of minimizing the error between the input image and the closest model instance; i.e., solving a nonlinear optimization problem. Baker *et al.* have developed facial recognition, interpretation, and coding techniques based on the AAM in [58]–[61].

F. Gabor Filtering

From the standpoint of neurobiology, subtle changes in the shape and texture of the face (that are essential to facial expression discrimination) are best represented using the spatially localized receptive fields typical of primary visual cortex cells. Gabor wavelet functions [62] are found to approximately model such behavior [44]. A complex-valued 2-D Gabor function is a plane wave restricted by a Gaussian envelope and is expressed as

$$\Psi(k,x) = \frac{k^2}{\sigma^2}\exp\left(-\frac{k^2x^2}{2\sigma^2}\right)\left[\exp(ik,x) - \exp\left(-\frac{\sigma^2}{2}\right)\right].$$

Here the multiplicative factor k^2 ensures that filters tuned to different spatial frequency bands have approximately equal energies. The term $\exp(-\sigma^2/2)$ is subtracted to render the filters insensitive to the overall level of illumination. The Gabor wavelet representation allows description of spatial frequency structure in the image, while preserving information about spatial relations. The complex amplitude of the transforms is used as features to test for the presence of spatial structure, restricted to a band of orientations and spatial frequencies, within the Gaussian envelope. The amplitude information degrades gracefully, with shifts in the image location at which it is sampled over the spatial scale of the envelope.

Gabor-wavelet-labeled elastic graph matching has been combined with the *eigenface* algorithm, for facial image

classification [44]. Facial images are transformed using a multiscale, multiorientation set of Gabor filters. Rectangular as well as *fiducial* (with nodes located at easily identifiable landmarks of the face) grids are registered with the face. This is followed by extraction of complex-valued Gabor transform coefficients sampled at the grid points and combined into a labeled graph vector. PCA is applied to reduce the dimensionality of the feature space. Finally, linear discriminant analysis is employed for clustering these Gabor-coded images, based on the different facial attributes. The facial registration grid parameters include the 2-D coordinates of the center of mass, and the horizontal and vertical grid line spacing.

The system [44] was implemented on a set of 193 facial images of Japanese females from the Japanese Female Facial Expression (JAFFE) database.[2] Fundamental facial expressions such as being happy, sad, angry, fearful, surprised, and disgusted were classified, along with a "neutral" face. Each test image was convolved with a set of Gabor filters, having highly correlated responses that are redundant at neighboring pixels. A sparse fiducial grid was positioned by manually clicking on 34 easily identifiable points of each facial image. The regions of the eyes and mouth were found to be the most critical areas for determining facial expressions. The projections of the filter responses along discriminant vectors, obtained from the training set, were compared at corresponding spatial frequency, orientation, and locations with the test image, using the normalized dot product to measure the similarity of the two Gabor response vectors.

It was observed that the horizontally oriented filters were more useful as compared to the vertically oriented ones. This also corroborates with the most noticeable expressive motions of the face such as: 1) opening and closing of mouth and 2) raising and lowering of eyebrows. The Gabor representation showed a significant degree of psychological plausibility, matching with the semantic rating of images by human observers, and demonstrating a promise for suitable human–computer interface.

G. Optical Flow

Pixel-based approaches to motion estimation are often referred to as optical flow methods [63]. Here, a direct relationship is assumed between object motion and intensity changes within an image sequence. Motion estimation is formulated as an optimization problem where the motion field corresponds to the operator, which best accounts for the intensity variations, given certain restrictions. These methods typically generate dense motion estimates. They perform well when the motion of individual objects is relatively slow, and the scene consists of only a few moving objects. Hence, the dynamics of facial expressions is a good candidate for estimation using optical flow.

Optical flow has been used to estimate facial muscle action [50], which are triggered by nerve impulses generated by emotions. The muscle actions cause the movement and deformation of facial skin and features such as eyes, mouth, and nose. Since the facial skin at cheek and forehead has a fine-grained texture, it is suitable for extracting the optical flow. These are then correlated with different facial expressions. The advantage of the optical flow algorithm is that there is no need to extract and trace particular points in the image sequence. The subtle texture of the facial skin is sufficient to extract skin motion in terms of optical flow.

In a top–down approach, the optical flow fields of skin movement is evaluated in muscle windows, each of which defines one primary direction of muscle contraction. However, it is necessary to collaborate with psychologists to construct knowledge-bases to categorize facial expressions from the muscle movement descriptions. In a bottom–up approach, a 15-dimensional feature vector is used to represent the most active points in terms of the first and second moments of velocity pattern through time at local regions derived from optical flow data. Evenly divided rectangular regions are used to cover the entire face as muscle windows, and the most active regions are selected from the sample data to generate the feature vectors. The k-nearest-neighbors rule [37] is then employed for facial expression recognition.

Let us consider an $M \times N$ image sequence with a period of T frames for the kth expression sample. The image is evenly divided into $m \times n$ regions, where $r = M/m = N/n$. The horizontal and vertical optical flow at pixel (x, y) in time frame t are given as $u_t(x, y)$ and $v_t(x, y)$, respectively. The means of optical flow of horizontal and the vertical components at the (i, j)th divided region $R(i, j)$ are expressed as

$$\mu_{u,i,j} = \frac{1}{T}\frac{1}{r^2} \sum_{0<t\leq T} \sum_{(x,y)\in R(i,j)} u_t(x, y) \tag{5}$$

and

$$\mu_{v,i,j} = \frac{1}{T}\frac{1}{r^2} \sum_{0<t\leq T} \sum_{(x,y)\in R(i,j)} v_t(x, y) \tag{6}$$

respectively. The corresponding covariances of the optical flow are computed as

$$\sigma_{uu,i,j} = \frac{1}{T}\frac{1}{r^2} \sum_{0<t\leq T} \sum_{(x,y)\in R(i,j)} (u_t(x, y) - \mu_{u,i,j})^2 \tag{7}$$

$$\sigma_{uv,i,j} = \frac{1}{T}\frac{1}{r^2} \sum_{0<t\leq T} \sum_{(x,y)\in R(i,j)} (u_t(x, y) - \mu_{u,i,j}) * (v_t(x, y) - \mu_{v,i,j})$$

and

$$\sigma_{vv,i,j} = \frac{1}{T}\frac{1}{r^2} \sum_{0<t\leq T} \sum_{(x,y)\in R(i,j)} (v_t(x, y) - \mu_{v,i,j})^2.$$

This results in a $K(= 5mn)$-dimensional vector, whose dimensionality is then reduced to $K = 15$ using feature selection. The parameter values used in the implementation were $M = 256$, $N = 240$, $m = 16$, $n = 15$, and $T = 30$.

Apart from their vulnerability to image noise and nonuniform lighting, the holistic optical flow methods typically have large computational requirements and are sensitive to motion discontinuities.

[2] http://www.mis.atr.co.jp/~mlyons/jaffe.html

H. Skin Color Modeling

A multiscale transform is utilized in [64] to segment images into homogeneous regions, at multiple scales, followed by skin region extraction based on a skin color model. The CIE LUV color space is used after discarding the luminance value. These regions are then merged, from the coarsest to the finest scale, until the shape is approximately elliptic. Postprocessing is done to include nonskin color facial features such as eyes and mouth. Experimental results demonstrate that human faces can be detected in color images, regardless of size, orientation, and viewpoint.

Dai *et al.* [52] demonstrate monitoring of patients on bed by analyzing and recognizing facial expressions. Initially, the facial skin regions are extracted by converting the RGB color image to its YIQ representation, and using the I component. The difference between two successive frames in the facial skin region is used to judge whether any facial action has occurred. Then, the optical flows for the facial action sequence of an expression are computed. The flow magnitudes are thresholded to reduce the effect of small random motions due to noise. Facial action features are next extracted from the optical flow projection histogram on the x and y coordinate axes, corresponding to the mouth and eye regions. The expressions modeled include happiness, easiness, uneasiness, disgust, suffering, and surprise for a set of 40 subjects on bed. A set of 40 facial action features were extracted. Application is envisaged in computerized monitoring by medical personnel of the facial expressions of patients, on bed in hospital, in order to evaluate their levels of uneasiness and suffering.

I. Connectionist Approach To Facial Gesture Recognition

Soft computing tools broadly encompass ANN, fuzzy sets and GAs. Connectionist approach to facial gesture recognition, constituting feedforward ANN models such as MLP [28], [29], [46], and RBFN [30], [47] has considerable representation in literature.

1) MLP: MLP-based classifier has been trained to distinguish between "face" and "nonface" patterns, for the purpose of vertical frontal-view face detection in cluttered scenes [46]. ANNs are typically example-based learners, and do not involve domain-specific knowledge. The network parameters and thresholds are automatically derived from a data-rich environment of input–output examples. An image is searched exhaustively over multiple scales for square patches of human face, with the upper boundary lying just above the eyes and the lower edge falling just below the mouth. At each image location and scale, the network classifies the local image pattern as being either a face or a nonface, based on a set of local distance (Mahalanobis and Euclidean) feature measurements to the face prototype. Preprocessing such as masking (of irrelevant background pixels), illumination gradient correction, and histogram equalization are followed by clustering, using an elliptical c-means algorithm, for generating the face prototypes.

Rowley *et al.* [29] use a retinally connected neural network to examine small (20×20) windows of an image in order to decide whether or not they contain a face (providing an output ranging from 1 to -1). A bootstrap algorithm is employed to collect negative (nonface) examples during back-propagation training. Multiple neural-network-based filters examine each location of the image at several scales, while looking for sites that might contain a face. The system arbitrator merges the results obtained by these networks and eliminates overlapping detections. To detect faces larger than the window size, the input image is repeatedly subsampled and the filter applied at each size.

Tian *et al.* [28] automatically analyze facial expressions based on permanent features (brows, eyes, mouth) and transient features (such as deepening of facial furrows). The output is determined in terms of AUs of the FACS [45], consisting of six upper and ten lower face AUs along with the neutral expression. Multistate face and facial component models are developed for tracking and modeling various facial features such as lips, eyes, brows, cheeks, and furrows. In the upper face, 15 parameters describe the shape and motion of the brows and cheeks, eye state, and furrows. In the lower face, nine parameters describe the shape, motion and state of lips, and furrows. These parameters are geometrically normalized to compensate for image scale and in-plane head motion. Two neural network models are employed for recognizing 16 frequently used AUs of the upper and lower face, respectively. Average recognition rates of 96.4% and 96.7% are reported for the upper and lower face AUs, respectively.

2) RBFN: A hybrid ensemble of radial basis function neural networks and inductive decision trees (C4.5) has been employed [47] on the FERET database for face (and hand) gesture recognition. Queries such as the following are addressed: 1) find all subjects wearing glasses and 2) find individual ID probe with/without glasses.

A hierarchical system of RBFN has been used for facial expression (gesture) recognition [30]. Here, the: 1) highest level identifies the different emotions; 2) mid level determines motion of the facial features; and 3) lowest level recovers the motion directions. The input is the pixel-level image of the face. The emotions modeled at the output layer include happiness, sadness, surprise, fear, anger, and disgust, along with eye blinking.

Trained subnetworks are dedicated for each of the modeled emotions. These subnetworks are further partitioned to specialize in a particular facial component (or feature). Motion in the image of a face allows emotions to be identified with minimal information about the the spatial arrangement of the facial features. A Gaussian weighted output vector, positioning its peak on the current stage of an emotion, is employed at the outer layer of the network to model the network's confidence in the classified emotion. Prominent facial features, such as mouth, nose, eyes, and eyebrows, are located and tracked.

Optical flow is used to identify the direction of rigid and nonrigid motions caused by these features, corresponding to the different human facial expressions. A sequence of facial images are used for the purpose of generating the flow and training the networks. Computational methods are developed to interpret face region motion, i.e., changes in images of facial features caused by facial actions corresponding to feature deformations on the 3-D surface of the face. Motion associated with the edges of the mouth, nose, eyes, and eyebrows is used as cue for action recovery.

A mid-level symbolic representation is modeled, based on linguistic and psychological considerations, to describe the spatio–temporal actions. For example, the action of "raising" a corner of the component "mouth" can also be a combination of raising the "upper" or "lower lip." Hence, interpretation precedence is incorporated to rank these actions. Classification rules are applied to group the actions into one of the facial expression classes. Whole mouth actions, lip actions, and mouth corner actions are ranked in this decreasing sequence.

The face is always considered from a near-frontal view. It is assumed that the overall rigid motion of the head is small, and the nonrigid motions are the result of spatially bounded face deformations between any two consecutive frames. The flow magnitudes are first thresholded to reduce the effect of noisy small motion. This is followed by quantization into the four principal directions (up, down, right, and left), and subsequent filtering (spatial and temporal) to capture directional motion. Polar coordinate representation is used to express the pixel motion. Temporal information, regarding what happened to a feature before a particular movement, is of great value in categorizing an emotion. For example, during expression of "surprise" the "eyebrows" move downward at the end of the emotion while in "anger," this movement happens at the beginning of the emotion.

The performance of the connectionist model is evaluated in terms of its ability of: 1) retention—performing successfully on familiar sequences (88%); 2) extrapolation—performing successfully on unfamiliar sequences or sequences of unfamiliar faces (73%); and 3) rejection—rejecting a sequence that does not display the expression for which the network was trained (79%).

V. Conclusion and Discussion

The importance of gesture recognition lies in building efficient human–machine interaction. Its applications range from sign language recognition through medical rehabilitation to virtual reality. In this article, we have provided a survey on gesture recognition, with particular emphasis on hand gestures and facial expressions. The major tools surveyed for this purpose include HMMs, particle filtering and condensation algorithm, FSMs, and ANNs. A lot of research has been undertaken on sign language recognition, mainly using the hands (and lips). Facial expression modeling involves the use of eigenfaces, FACS, contour models, wavelets, optical flow, skin color modeling, as well as a generic, unified feature-extraction-based approach.

A hybridization of HMMs and FSMs is a potential study in order to increase the reliability and accuracy of gesture recognition systems. HMMs are computationally expensive and require large amount of training data. Performance of HMM-based systems could be limited by the characteristics of the training dataset. On the other hand, the statistically predictive state transition of the FSM might possibly lead to more reliable recognition. An interesting approach worth exploring is the independent modeling of each state of the FSM as an HMM. This can be useful in recognizing a complex gesture consisting a sequence of smaller gestures.

Soft computing tools [26] pose another promising application to static hand gesture identification. For large datasets, neural networks have been used for representing and learning the gesture information. Both recurrent and feedforward networks, with a complex preprocessing phase, have been used for recognizing static postures. The dynamic movement of hand has been modeled by HMMs and FSMs. The similarity of a test hand shape may be determined with respect to prototype hand contours, using fuzzy sets. TDNN and recurrent neural networks offer promise in capturing the temporal and contextual relations in dynamic hand gestures.

With the advent of multimedia data mining [4], the need for intelligent storage, search, processing, and retrieval of information from large, heterogeneous databases through the application of user friendly interfaces is assuming utmost importance. This promises wide-ranging applications in fields from photojournalism through medical technology to biometrics. Due to the increasing involvement of pictorial information, the need for image compression and subsequent querying of online image databases is becoming all the more essential. Analysis of compressed multimedia databases for gesture identification and/or recognition is another promising area of investigation. The concept of wavelets may be employed for face recognition at a coarse level of resolution, followed by finer-level detection of facial expression at the appropriate regions of interest by employing increased resolution.

Fuzzy sets and rough sets provide a natural framework for dealing with uncertainty or imprecise data. Since "pure" emotional expressions are seldom elicited, people typically demonstrate "blends" of these expressions. Fuzzy sets can be suitably employed to represent simultaneous finite membership to multiple emotional categories such as sad–angry or angry–afraid expressions. One can also model different degrees of a particular expression by employing fuzzy membership values to it. For example, the expression surprise can consist of subcategories such as dazed surprise, questioning surprise, slight surprise, moderate surprise, etc. In addition, transition from one expression to another often requires passage through several other quantifiable grades of expression (transition from a happy to angry face will require passage through a neutral face). This may be suitably expressed in fuzzy linguistic terms, thereby enhancing understandability and user-friendliness.

Moreover, not all facial expressions can be completely classified into the six defined categories. There should exist possibility of learning new categories for clustering, and then interpreting each and every encountered facial expression. Similarity-based matching of the retrieved images may be performed on these clusters, using concepts from approximate reasoning, searching, and learning.

References

[1] C. L. Lisetti and D. J. Schiano, "Automatic classification of single facial images," *Pragmatics Cogn.*, vol. 8, pp. 185–235, 2000.

[2] V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual interpretation of hand gestures for human computer interaction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 677–695, Jul. 1997.

[3] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–285, Feb. 1989.

[4] S. Mitra and T. Acharya, *Data Mining: Multimedia, Soft Computing, and Bioinformatics*. New York: Wiley, 2003.

[5] D. M. Gavrila, "The visual analysis of human movement: A survey," *Comput. Vis. Image Understanding*, vol. 73, pp. 82–98, 1999.

[6] J. K. Aggarwal and Q. Cai, "Human motion analysis: A review," *Comput. Vis. Image Understanding*, vol. 73, pp. 428–440, 1999.

[7] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time-sequential images using hidden Markov model," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recogn.*, Champaign, IL, 1992, pp. 379–385.

[8] F. Samaria and S. Young, "HMM-based architecture for face identification," *Image Vis. Comput.*, vol. 12, pp. 537–543, 1994.

[9] G. Welch and G. Bishop, "An introduction to the Kalman filter," Dept. Comput. Sci., Univ. North Carolina, Chapel Hill, Tech. Rep. TR95041, 2000.

[10] S. Arulapalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2001.

[11] C. Kwok, D. Fox, and M. Meila, "Real-time particle filters," *Proc. IEEE*, vol. 92, no. 3, pp. 469–484, Mar. 2004.

[12] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," in *Proc. Eur. Conf. Comput. Vis.*, Cambridge, U.K., 1996, pp. 343–356.

[13] ——, "CONDENSATION—Conditional density propagation for visual tracking," *Int. J. Comput. Vis.*, vol. 1, pp. 5–28, 1998.

[14] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo in Practice*. NewYork: Springer-Verlag, 2001.

[15] J. Davis and M. Shah, "Visual gesture recognition," *Vis., Image Signal Process.*, vol. 141, pp. 101–106, 1994.

[16] A. F. Bobick and A. D. Wilson, "A state-based approach to the representation and recognition of gesture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 12, pp. 1235–1337, Dec. 1997.

[17] M. Yeasin and S. Chaudhuri, "Visual understanding of dynamic hand gestures," *Pattern Recogn.*, vol. 33, pp. 1805–1817, 2000.

[18] P. Hong, M. Turk, and T. S. Huang, "Gesture modeling and recognition using finite state machines," in *Proc. 4th IEEE Int. Conf. Autom. Face Gesture Recogn.*, Grenoble, France, Mar. 2000, pp. 410–415.

[19] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[20] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Reading, MA: Addison-Wesley, 1992.

[21] M. Kass, A. Witkin, and D. Terzopoulos, "SNAKE: Active contour models," *Int. J. Comput. Vis.*, pp. 321–331, 1988.

[22] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York: Macmillan, 1994.

[23] A. J. Viterbi, "Error bounds for convolution codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967.

[24] M. Isard and A. Blake, "A mixed-state condensation tracker with automatic model-switching," in *Proc. 6th Int. Conf. Comput. Vis.*, Mumbai, India, 1998, pp. 107–112.

[25] L. A. Zadeh, "Fuzzy logic, neural networks, and soft computing," *Commun. ACM*, vol. 37, pp. 77–84, 1994.

[26] S. K. Pal and S. Mitra, *Neuro-fuzzy Pattern Recognition: Methods in Soft Computing*. NewYork: Wiley, 1999.

[27] M. S. Yang and N. Ahuja, "Recognizing hand gesture using motion trajectories," in *Proc. IEEE CS Conf. Comput. Vis. Pattern Recogn.*, vol. 1, Fort Collins, CO, Jun.1998, pp. 466–472.

[28] Y. L. Tian, T. Kanade, and J. F. Cohn, "Recognizing action units for facial expression analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 2, pp. 97–115, Feb. 2001.

[29] H. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 1, pp. 23–38, Jan. 1998.

[30] M. Rosenblum, Y. Yacoob, and L. S. Davis, "Human expression recognition from motion using a radial basis function network architecture," *IEEE Trans. Neural Netw.*, vol. 7, no. 5, pp. 1121–1138, Sep. 1996.

[31] W. T. Freeman and M. Roth, "Orientation histograms for hand gesture recognition," in *Proc. IEEE Int. Workshop Autom. Face Gesture Recogn.*, Zurich, Switzerland, Jun.1995, pp. 296–301.

[32] R. Bowden, A. Zisserman, T. Kadir, and M. Brady, "Vision based interpretation of natural sign languages," in *Proc. 3rd Int. Conf. Comput. Vis. Syst.*, Graz, Austria, Apr. 2003, pp. 391–401.

[33] T. Starner and A. Pentland, "Real-time American Sign Language recognition from video using hidden Markov models," MIT Media Lab., MIT, Cambridge, MA, Tech. Rep. TR-375, 1995.

[34] J. Weaver, T. Starner, and A. Pentland, "Real-time American Sign Language recognition using desk and wearable computer based video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 1371–1378, Dec. 1998.

[35] R. Bowden, D. Windridge, T. Kadir, A. Zisserman, and M. Brady, "A linguistic feature vector for the visual interpretation of sign language," in *Proc. 8th Eur. Conf. Comput. Vis.*, New York: Springer-Verlag, 2004, pp. 391–401.

[36] H. S. Yoon, J. Soh, Y. J. Bae, and H. S. Yang, "Hand gesture recognition using combined features of location, angle and velocity," *Pattern Recogn.*, vol. 34, pp. 1491–1501, 2001.

[37] J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles*. London, U.K.: Addison-Wesley, 1974.

[38] A. Ramamoorthy, N. Vaswani, S. Chaudhury, and S. Banerjee, "Recognition of dynamic hand gestures," *Pattern Recogn.*, vol. 36, pp. 2069–2081, 2003.

[39] M. J. Black and A. D. Jepson, "A probabilistic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions," in *Proc. 5th Eur. Conf. Comput. Vis.*, vol. 1, 1998, pp. 909–924.

[40] A. Samal and P. Iyengar, "Automatic recognition and analysis of human faces and facial expressions," *Pattern Recogn.*, vol. 25, pp. 65–77, 1992.

[41] R. Chellappa, C. L. Wilson, and S. Sirohey, "Human and machine recognition of faces: A survey," *Proc. IEEE*, vol. 83, no. 5, pp. 705–740, May 1995.

[42] J. Daugman, "Face and gesture recognition: An overview," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 675–676, Jul. 1997.

[43] M. Pantic and L. J. M. Rothkranz, "Automatic analysis of facial expressions: The state of the art," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 12, pp. 1424–1445, Dec. 2000.

[44] M. J. Lyons, J. Budynek, and S. Akamatsu, "Automatic classification of single facial images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 12, pp. 1357–1362, Dec. 1999.

[45] P. Ekman and W. V. Friesen, *Facial Action Coding System (FACS): Manual*. Palo Alto: Consulting Psychologists Press, 1978.

[46] K. K. Sung and T. Poggio, "Learning human face detection in cluttered scenes," in *Proc. IEEE 6th Int. Conf. CAIP'95*, Prague, Czech Republic, 1995, pp. 432–439.

[47] S. Gutta and H. Wechsler, "Face recognition using hybrid classifiers," *Pattern Recogn.*, vol. 30, pp. 539–553, 1997.

[48] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cogn. Neurosci.*, vol. 3, pp. 71–86, 1991.

[49] I. Essa and A. Pentland, "Coding, analysis, interpretation, recognition of facial expressions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 757–736, Jul. 1997.

[50] K. Mase, "Recognition of facial expression from optical flow," *IEICE Trans.*, vol. E 74, pp. 3474–3483, 1991.

[51] Y. Yacoob and L. Davis, "Recognizing human facial expressions from long image sequences using optical flow," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 6, pp. 636–642, Jun. 1996.

[52] Y. Dai, Y. Shibata, T. Ishii, K. Hashimoto, K. Katamachi, K. Noguchi, N. Kakizaki, and D. Cai, "An associate memory model of facial expressions and its applications in facial expression recognition of patients on bed," in *Proc. IEEE Int. Conf. Multimedia Expo*. Aug. 22–25, 2001, pp. 772–775.

[53] R. Brunelli and T. Poggio, "Face recognition: Features versus templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 10, pp. 1042–1052, Oct. 1993.

[54] B. Fasel and J. Luettin, "Automatic facial expression analysis: A survey," *Pattern Recogn.*, vol. 36, pp. 259–275, 2003.

[55] A. Nikolaidis and I. Pitas, "Facial feature extraction and pose determination," *Pattern Recogn.*, vol. 33, pp. 1783–1791, 2000.

[56] J. Illingworth and J. Kittler, "The adaptive Hough transform," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 9, no. 5, pp. 690–698, May. 1987.

[57] A. Lanitis, C. J. Taylor, and T. F. Cootes, "Automatic interpretation and coding of face images using flexible models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 743–756, Jul. 1997.

[58] S. Baker, I. Matthews, and J. Schneider, "Automatic construction of active appearance models as an image coding problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 10, pp. 1380–1384, Oct. 2004.

[59] I. Matthews and S. Baker, "Active appearance models revisited," *Int. J. Comput. Vis.*, vol. 60, pp. 135–164, 2004.

[60] J. Xiao, S. Baker, I. Matthews, and T. Kanade, "Real-time combined 2D+3D active appearance models," *Comput. Vis. Pattern Recogn.*, vol. 2, pp. 538–542, 2004.
[61] R. Gross, I. Matthews, and S. Baker, "Appearance-based face recognition and light-fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 4, pp. 449–465, Apr. 2004.
[62] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *J. Opt. Soc. Amer. A*, vol. 2, pp. 1160–1169, 1985.
[63] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, pp. 185–203, 1981.
[64] M. S. Yang and N. Ahuja, "Detecting human faces in color images," in *Proc. IEEE Int. Conf. Image Process.*, Chicago, IL, Oct. 1998, pp. 127–130.

Sushmita Mitra (S'91–M'99–SM'00) received the B.Tech and M.Tech degrees from the University of Calcutta, Kolkata, India, in 1987 and 1989, respectively, and the Ph.D. degree from the Indian Statistical Institute, Kolkata, in 1995, all in computer science.

She is currently a Professor at the Machine Intelligence Unit, Indian Statistical Institute. From 1992 to 1994, she was with the Rheinisch-Westfälischen Technischen Hochschule (RWTH), Aachen, Germany, as a German Academic Exchange Service (DAAD) Fellow. She was a Visiting Professor in the Computer Science Departments of the University of Alberta, Edmonton, AB, Canada, in 2004, Meiji University, Japan, in 1999, 2004, and 2005, and Aalborg University Esbjerg, Denmark, in 2002 and 2003. She is the author of *Neuro-Fuzzy Pattern Recognition: Methods in Soft Computing* and *Data Mining: Multimedia, Soft Computing, and Bioinformatics* (New York: Wiley). She has been a Guest Editor for special issues of journals and is an Associate Editor of *Neurocomputing*. She has more than 100 research publications in refereed international journals. Her current research interests include data mining, pattern recognition, soft computing, image processing, and bioinformatics.

Dr. Mitra received the IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award in 1994 for her pioneering work in neuro-fuzzy computing, and the CIMPA-INRIA-UNESCO Fellowship in 1996. She has served as the Program Chair, Tutorial Chair, and a member of program committees of several international conferences.

Tinku Acharya (M'96–SM'01) received the B.Tech and M.Tech degrees from the University of Calcutta, Kolkata, India, in 1987 and 1989, respectively, and the Ph.D. degree from the University of Central Florida, Orlando, in 1994, all in computer science.

He is a Co-Founder and Chief Technology Officer of Avisere Inc., Tucson, AZ. He is also an Adjunct Professor in the Department of Electrical Engineering, Arizona State University, Tempe. During 1996–2002, he was with Intel Corporation, Arizona, where he led several research and development teams in numerous projects toward the development of algorithms and architectures in image and video processing, multimedia computing, PC-based digital camera, high-performance reprographics architecture for color photocopiers, biometrics, multimedia for third-generation cellular mobile telephony, microprocessors, etc. Before joining Intel Corporation in 1996, he was a Consulting Engineer with AT&T Bell Laboratories (1995–1996), NJ, a Research Faculty in the Institute of Systems Research, Institute of Advanced Computer Studies, University of Maryland at College Park (1994–1995), a Systems Analyst at the National Informatics Centre, Planning Commission, Government of India (1988–1990), and a Visiting Professor at the Indian Institute of Technology, Kharagpur (in several occasions during 1998–2001), in addition to many other professional assignments. He has collaborated in research and development with the Palo Alto Research Center (PARC) of Xerox Corporation and the Kodak Corporation. He is an inventor of 83 awarded U.S. patents and 15 European patents in different areas of research and development. He is the author of *Image Processing: Principles and Applications*, *JPEG2000 Standard for Image Compression: Concepts, Algorithms and VLSI Architectures*, and *Data Mining: Multimedia, Soft Computing, and Bioinformatics* (New York: Wiley). He is a Co-Editor of *Information Technology: Principles and Applications* (New Delhi, India: Prentice-Hall). He has over 75 refereed technical papers published in international journals, conferences, and book chapters. His current research interests include computer vision for enterprise applications, multimedia computing, biometrics, image processing, very large scale integration architectures, and algorithms.

Dr. Acharya was awarded the Most Prolific Inventor in Intel Corporation Worldwide in 1999 and the Most Prolific Inventor in Intel Corporation, Arizona, for five consecutive years (1997–2001). He is a Life Fellow of the Institution of Electronics and Telecommunication Engineers. He has served in the U.S. National Body of Joint Photographic Experts Group 2000 Standard Committee (1998–2002), and in the program committees of several international conferences and academic and industrial organizations.