Computer Vision and Image Understanding **85,** 189–208 (2002)
doi:10.1006/cviu.2002.0967

A Real-Time Approach to the Spotting, Representation, and Recognition of Hand Gestures for Human–Computer Interaction

Yuanxin Zhu and Guangyou Xu

Department of Computer Science and Technology, Tsinghua University,
Beijing 100084, People's Republic of China
E-mail: zhu4@scripps.edu

and

David J. Kriegman

Beckman Institute for Advanced Science and Technology and Department of Computer Science, University of Illinois at Urbana–Champaign, 405 N. Mathews Avenue, Urbana, Illinois 61801

Received June 19, 2001; accepted June 7, 2002

Aiming at the use of hand gestures for human–computer interaction, this paper presents a real-time approach to the spotting, representation, and recognition of hand gestures from a video stream. The approach exploits multiple cues including skin color, hand motion, and shape. Skin color analysis and coarse image motion detection are joined to perform reliable hand gesture spotting. At a higher level, a compact spatiotemporal representation is proposed for modeling appearance changes in image sequences containing hand gestures. The representation is extracted by combining robust parameterized image motion regression and shape features of a segmented hand. For efficient recognition of gestures made at varying rates, a linear resampling technique for eliminating the temporal variation (time normalization) while maintaining the essential information of the original gesture representations is developed. The gesture is then classified according to a training set of gestures. In experiments with a library of 12 gestures, the recognition rate was over 90%. Through the development of a prototype gesture-controlled panoramic map browser, we demonstrate that a vocabulary of predefined hand gestures can be used to interact successfully with applications running on an off-the-shelf personal computer equipped with a home video camera. © 2002 Elsevier Science (USA)

Key Words: human–computer interaction; gesture recognition; gesture spotting; integration of multiple cues; spatiotemporal representation; robust image motion regression; template-based classification.

1. INTRODUCTION

This paper presents a real-time approach to the spotting, visual appearance modeling, and recognition of hand gestures from a single video stream. In recent years, hand gesture recognition has become a very active research theme because of its potential use in human–computer interaction (HCI), image/video coding, and content-based image/video retrieval, for example. Furthermore, a successful hand gesture recognition system will provide valuable insight into how one might approach other similar pattern recognition problems such as facial expression interpretation, lip reading, and human action identification. However, recognizing hand gestures from a real-time video stream is a challenging task for at least three major reasons: (i) hand gestures are dynamic and are characterized by the changing hand shape and its motion; (ii) the human hand is a complex nonrigid object; and (iii) for each type of gesture, there is quite a bit of variability in how each person performs the gesture.

Approaches to hand gesture recognition from video streams can be classified into two major groups: those based on 3-D modeling of the hand (with or without the arm) [1–6] and those based on directly modeling the visual appearance changes in an image sequence caused by hand motion [7–26]. Generally speaking, approaches in the first group are applicable to almost all kinds of hand gestures while those in the second group may be only suitable for communicative hand gestures. However, the computational cost of fully recovering the 3-D hand/arm state is prohibitive for real-time recognition and may be unnecessary to accomplish the task. Even worse, there are many approximations involved in the process of 3-D modeling, and so the recovery of model parameters is seldom stable. By contrast, appearance-based approaches have many attractive features including low computational cost, real-time processing, and as shown here high accuracy for a modest gesture vocabulary.

Hand detection is certainly a first step of fully automatic gesture recognition. For appearance-based approaches, color and motion are the most frequently used visual cues for detecting hands. Previous works usually use these visual cues individually. Most of the color-based detection techniques rely on histogram matching [4, 7, 26], or use a simple look-up table [5, 8, 12] obtained by training image data from the skin and the surrounding nonskin areas. These techniques frequently suffer from the variability of the skin color under different lighting conditions. Although the problem can be partially removed using restrictive backgrounds and clothing, wearing uniquely colored gloves, or putting markers on the hand and/or fingers [13, 14], the intrusiveness of these techniques is unreasonable for natural HCI.

The approach presented in this paper uses multiple cues including color, hand motion, and shape for the spotting, representation, and recognition of hand gestures. In addition to validating the method off-line on a test set of gestures, a prototype on-line gesture recognition system—a gesture-controlled panoramic map browser—was built to demonstrate the application of the proposed approach to gesture recognition for HCI. Figure 1 illustrates the processing architecture of the system. First, skin color analysis and coarse image motion detection are joined to perform robust gesture spotting, namely identifying an image sequence within the video stream that contains a hand gesture and locating image frames corresponding to the start and end of the gesture. Then a compact spatiotemporal representation, characterizing both the motion and shape appearance changes in the image sequence, is extracted using robust parameterized image motion regression and shape analysis of the hand image region. Afterwards, the spatiotemporal representation is passed to the

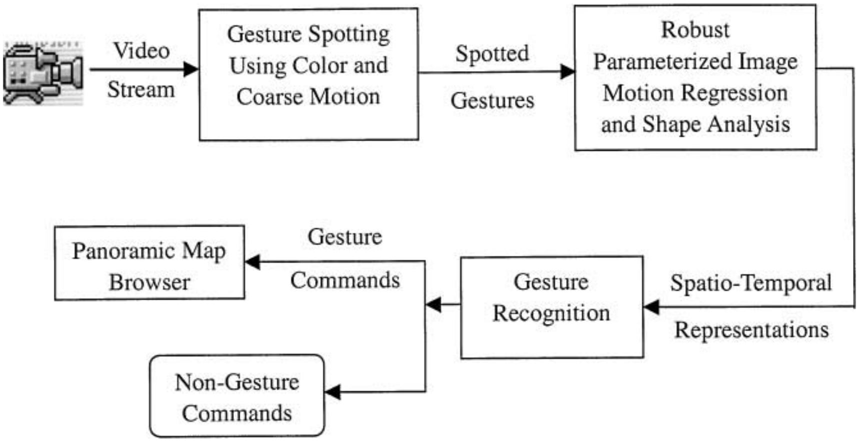


FIG. 1. Processing architecture of the prototype gesture-controlled panoramic map browser.

recognition module. If the representation is recognized as a predefined gesture command, then the panoramic map browser will respond to the predefined function of the gesture command.

In order to realistically expect hand gestures to be used for HCI, the gesture recognition module must not only be sufficiently accurate but as part of a larger system it must be very efficient since only a small portion of system resources are likely to be devoted to the module. Hence many of the design decisions (e.g., linear resampling vs dynamic time warping) were made in favor of faster computation when there was negligible degradation of recognition performance.

The rest of the paper is organized as follows. Section 2 provides some related work. Section 3 describes the technique for real-time spotting of hand gestures in video streams. The spatiotemporal appearance modeling and recognition technique are given respectively in Section 4 and Section 5. Section 6 reports the results of both off-line (isolated gesture) and on-line experiments. A brief summary and some discussion are provided in Section 7. An initial, short version of this paper was presented in [27].

2. RELATED WORK

Work on recognizing hand postures or gestures for HCI has been surveyed recently [28, 29]. Some efforts focus on interpreting static hand postures. For example, using statistical shape models, five hand poses were recognized and used to simulate a mouse manipulating a windows interface [18]. To use hand gestures for navigation in a virtual environment, Maggioni [11] first exploited the color cue for hand segmentation, then extracted moments of the hand contour as features to recognize six hand postures. In Ref. [30], region direction histograms were used for hand tracking, and simple hand postures were identified and used to control games, appliances, etc.

Others focus on recognizing hand gestures from a video stream. For example, Becker [31] developed the Sensi system that can recognize five T'ai Chi gestures by using the velocity of the hand in 3-D space as features combined with hidden Markov models (HMMs). Culter and Turk [21] built a virtual playground for children in which six types of gestures were interpreted using the motion and shape features of motion blobs and rule-based recognition.

Another area of effort is recognition of sign language, especially American Sign Language (ASL). For example, Vogler and Metaxas [32] developed a 3-D camera system that can recognize a 53-word lexicon with 89.9% accuracy by coupling HMMs and 3-D motion analysis; Starner *et al.* [20] described two experiments of recognizing sentence-level ASL selected from a 40-word lexicon with higher than 90% accuracy.

As noted by many other researchers, it is difficult to make a clear and convincing comparison between currently reported gesture recognition systems because almost no two systems are intended for the same application. As a result, the complexity of each system also varies in terms of its gesture command set, assumptions about the background, system settings, etc. Also unlike face detection and face recognition, there is also a lack of standard databases with which to compare different gesture recognition systems. Therefore, it is probably more appropriate and useful to point out some salient characteristics of our system. These include: (i) the system has a larger command set than most of those systems intended for manipulating virtual objects within HCI context even though it has a smaller vocabulary than those of sign language recognition systems; (ii) the system can spot, analyze, and recognize hand gestures with high accuracy as well as give feedback in real-time; (iii) users are allowed to continuously issue gesture commands from any location within the view field of the camera; (iv) hand gestures are recognized in real-time from video without resorting to any special marks, restricted or uniform background, or particular lighting conditions; and (v) a consumer personal computer and an uncalibrated home-use video camera are the only required equipment.

3. HAND GESTURE SPOTTING

Akin to word spotting in speech recognition, the goal of hand gesture spotting is first to delineate a moving hand (if it appears in a video stream) from the rest of the image and then to locate the start and end frames of the image sequence containing a hand gesture. The temporal characteristics of hand gestures play an important role in gesture spotting. Psychological studies are fairly consistent about the temporal nature of hand gestures; it has been established that three phases make a gesture: preparation, stroke, and retraction [33]. Yet, this still leaves quite a bit of temporal variability in hand gestures, and provides a challenge for gesture spotting. A trade-off between the complexity of gesture recognition and the naturalness of performing gestures must be made. As Baudel and Beaudouin-Lafon [34] noticed, users must largely follow a set of rules—the interaction model—when communicating with computers using gestures.

Within an HCI context, we propose an interaction model in which an image sequence is considered to contain a hand gesture if the sequence has the following four characteristics: (i) a moving hand appears in the sequence; (ii) the moving hand is the dominant moving object; (iii) the movement of the hand follows a three-stage process, namely starts from rest, then moves smoothly for a short period of time, and finally slows down; and (iv) the duration of the moving stage is longer than L_1 frames, but not more than L_2 frames for a given sampling rate. Integrating the interaction model with motion and color cues gives us a real-time gesture-spotting algorithm, which is explained in detail in the following.

As a first step, skin regions are segmented from the rest of the image using color, and numerous skin color detectors have been developed. Depending upon the segmentation method (e.g., metrics used), different color spaces may be preferable. (Note: when Bayesian classification is used, the color space may be unimportant [35].) Under a Euclidean norm, skin color is usually more distinctive and less sensitive to illumination changes in the HSI (hue, saturation, and intensity) space rather than in the RGB space. Off-line, a hue–saturation look-up table of skin color is learned using a supervised learning technique on data collected from the skin and its surrounding nonskin areas (see Ref. [36] for details about the learning technique). On-line, each pixel is labeled according to its similarity to the skin color by indexing into the table constructed off-line.

The on-line video stream containing hand gestures can be considered as a signal, $S(x, y, t)$, where (x, y) denote image coordinates, and t denotes time. After converting the original signal from RGB space to HSI, the intensity signal $I(x, y, t)$ is extracted. The above-mentioned look-up table approach is used to label skin pixels and form a binary image sequence $M'(x, y, t)$ that we call the region mask sequence. Figure 4 shows example image sequences along with the region mask sequences. A second binary image sequence $M''(x, y, t)$ that reflects motion information is produced by thresholding the difference images between every consecutive pair of intensity images. See again Fig. 4 for examples. Both $M'(x, y, t)$ and $M''(x, y, t)$ are contaminated by noise, and so morphological filtering operations of erosion and dilation are applied to eliminate isolated pixels and fill out holes.

Given the skin color mask sequence $M'(x, y, t)$ and the motion mask sequence $M''(x, y, t)$, a new binary image sequence delineating the moving skin region, $M(x, y, t)$, is produced by performing a logical AND operation between the corresponding frames of the two mask sequences. The algorithm then determines whether a moving hand appears in each image based on the area ratio of each region in $M(x, y, t)$ to that of the whole image. Under the assumed interaction model, a hand gesture is spotted if a moving hand is detected in more than L_1 but not more than L_2 consecutive frames. The result is an image sequence containing a segmented moving hand.

4. SPATIO-TEMPORAL APPEARANCE MODELING

Given a spotted gesture (an intensity image sequence $I(x, y, t)$ and a region mask sequence $M(x, y, t)$ where each mask covers the moving hand region within its corresponding intensity image), a spatiotemporal representation of the gesture is extracted by combining robust parameterized image motion regression and a shape descriptor of the hand image region. The extracted representation is a trajectory of feature vectors consisting of motion and shape components that characterize, respectively, motion and shape appearance changes in the image sequence.

4.1. Robust Parameterized Image Motion Regression

As Bergen *et al.* [37] noted, parameterized models of image motion make explicit the assumptions about the spatial variation of the optical flow within an image region, and typically it is assumed that the flow can be represented by a low-order polynomial. Among the widely used parameterized models are the translation, affine, and planar models. Black and Anandan [38] presented a robust estimation framework for parameterized image motion regression, especially robust affine motion regression. The framework has advantages such as a concise representation, high computational efficiency, and stability. We adopt this framework, and show in the paper how it can be successfully applied to modeling the image motion of hand gestures.

For completeness, we briefly explain the robust estimation framework. The eight parameter planar model of image motion [39] can be written

$$\begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \begin{bmatrix} a_0 \\ a_3 \end{bmatrix} + \begin{bmatrix} a_1 & a_2 \\ a_4 & a_5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} x^2 & xy \\ xy & y^2 \end{bmatrix} \begin{bmatrix} a_6 \\ a_7 \end{bmatrix}, \tag{1}$$

where a_i $(i = 0, \ldots, 7)$ are constant with respect to the image region, and $u(x, y)$ and $v(x, y)$ are respectively the horizontal and vertical motion components at pixel $\mathbf{x}$ with coordinates (x, y).

For convenience of notation, we define

$$\mathbf{X}(\mathbf{x}) = \begin{bmatrix} 1 & x & y & 0 & 0 & 0 & x^2 & xy \\ 0 & 0 & 0 & 1 & x & y & xy & y^2 \end{bmatrix},$$

$\mathbf{T} = (a_0, 0, 0, a_3, 0, 0, 0, 0)^{\mathrm{T}}$, $\mathbf{A} = (a_0, a_1, a_2, a_3, a_4, a_5, 0, 0)^{\mathrm{T}}$, and $\mathbf{P} = (a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7)^{\mathrm{T}}$. The translation, affine, and planar motions models can then be expressed as $\mathbf{u}(\mathbf{x}, \mathbf{T}) = \mathbf{X}(\mathbf{x})\mathbf{T}$, $\mathbf{u}(\mathbf{x}, \mathbf{A}) = \mathbf{X}(\mathbf{x})\mathbf{A}$, and $\mathbf{u}(\mathbf{x}, \mathbf{P}) = \mathbf{X}(\mathbf{x})\mathbf{P}$ respectively.

Let $\Re$ be the set of image points within a region (e.g., within the masked region $M(x, y, t)$), and let $\boldsymbol{\Theta}$ be the parameter vector of the motion model ($\boldsymbol{\Theta}$ could be $\mathbf{T}$, $\mathbf{A}$, or $\mathbf{P}$). The brightness constancy assumption for the analysis region states that

$$I(\mathbf{x}, t) = I(\mathbf{x} - \mathbf{X}(\mathbf{x})\boldsymbol{\Theta}, t + 1), \quad \forall \mathbf{x} \in \Re, \tag{2}$$

where I is the image intensity function, and t represents time. Taking the Taylor series expansion of the right-hand side, simplifying, and dropping terms above first order gives

$$\nabla \mathbf{I}^{\mathrm{T}}(\mathbf{X}(\mathbf{x})\boldsymbol{\Theta}) + I_t = 0, \quad \forall \mathbf{x} \in \Re, \tag{3}$$

where $\nabla \mathbf{I} = [I_x, I_y]^{\mathrm{T}}$ is the intensity gradient, I_x, I_y, and I_t are respectively partial derivatives of image intensity with respect to the spatial coordinates and time. Then the parameter vector, $\boldsymbol{\Theta}$, is estimated by minimizing the objective function

$$E(\boldsymbol{\Theta}) = \sum_{\mathbf{x} \in \Re} \rho(\nabla \mathbf{I}^{\mathrm{T}}(\mathbf{X}(\mathbf{x})\boldsymbol{\Theta}) + I_t, \sigma), \tag{4}$$

where ρ is some robust error norm, and σ is a scale parameter.

Numerous ρ functions are used in the computer vision literature, and for the experiments in this paper we follow [40] and use the German–McClure function,

$$\rho(r, \sigma) = \frac{r^2}{\sigma^2 + r^2} \tag{5}$$

where $r = \nabla \mathbf{I}^{\mathrm{T}}(\mathbf{X}(\mathbf{x})\boldsymbol{\Theta}) + I_t$. Equation (4) is then minimized using a simultaneous over-relaxation scheme with a continuation method (see Ref. [38] for detail). Specifically, the iterative update equation for minimizing $E(\boldsymbol{\Theta})$ at step $n + 1$ is simply

$$a_i^{n+1} = a_i^n - \omega \frac{1}{T(a_i)} \frac{\partial E(\boldsymbol{\Theta})}{\partial a_i}, \quad i = 0, 1, \ldots, 7, \tag{6}$$

where $0 < \omega < 2$ is the relaxation parameter. When $0 < \omega < 2$ the method can be shown to converge. However, the rate of convergence is sensitive to the exact value of ω. The term $T(a_i)$ is an upper bound on the second partial derivative of $E(\mathbf{\Theta})$, that is,

$$T(a_i) \geq \frac{\partial^2 E(\mathbf{\Theta})}{\partial a_i^2}, \quad i = 0, \ldots, 7.$$

The control parameter, σ, is lowered according to a factor during each interation. The effect of this procedure is that initially all points in the region contribute to the solution and gradually the influence of the outlying residuals is reduced.

To cope with large motions a coarse-to-fine strategy is used in which a Gaussian pyramid is constructed [37]. Starting at the lowest spatial resolution with the motion parameters, $\mathbf{\Theta}$, being initialized as zero, the change in the motion estimate, $\Delta\mathbf{\Theta}$, is computed. The new motion parameters, $\mathbf{\Theta} + \Delta\mathbf{\Theta}$, are then projected onto the next level in the pyramid (scaled as appropriate), and the image at time t is warped toward the image at time $t + 1$ using the current motion parameters. The warped image is then used to compute $\Delta\mathbf{\Theta}$ at this level. The process is repeated until the finest level.

4.2. Interframe Motion Appearance

We use a parameterized image motion model to approximate the projected 3-D motions of the hand onto the image plane. For instance, the affine model would be appropriate when the difference in depth caused by the hand motion is small relative to the distance of the hand from the camera. With the intensity image region covered by the mask image as the analysis region, robust parameterized image motion regression is used to recover the motion model parameters. Since almost all pixels in the region belong to the image of the moving hand, the regressed motion is the hand motion. At the same time, by identifying inliers and rejecting outliers according to the result of regression, a better segmentation of the moving hand can be generated as a by-product [23].

As Cipolla *et al.* [41] and Black and Yacoob [40] indicated, the parameters of image motion models can be decomposed into independent components that have simple geometric interpretations. The specific interpretation of the parameters or their combinations is given in the following:

- Pure horizontal translation (m_1): $m_1 = a_0$
- Pure vertical translation (m_2): $m_2 = a_3$
- Isotropic expansion (m_3) specifying a change in scale: $m_3 = a_1 + a_5$
- Pure shear or deformation (m_4) that describes the distortion of the image region:

$$m_4 = \sqrt{(a_1 - a_5)^2 + (a_2 + a_4)^2}$$

- 2-D rigid rotation (m_5) specifying the change in orientation: $m_5 = -a_2 + a_4$
- Yaw (m_6) about the view direction: $m_6 = a_6$
- Pitch (m_7) about the view direction: $m_7 = a_7$.

The rotation, divergence, and the deformation are scalar invariant and do not depend on the particular choice of image coordinate system. Therefore, a feature vector $\mathbf{m}_t = [m_1, m_2, m_3, m_4, m_5, m_6, m_7]^{\mathrm{T}}$ for characterizing the image motion (motion appearance) between consecutive frames at time t is constructed.

4.3. Intraframe Shape Appearance

Geometric moments are a succinct description of a region's shape. In this paper, a spatial covariance matrix

$$\begin{bmatrix} \tilde{c}_{2,0} & \tilde{c}_{1,1} \\ \tilde{c}_{1,1} & \tilde{c}_{0,2} \end{bmatrix} \tag{7}$$

of the region is formed (central moments $\tilde{c}_{p,q}$ of order p, q). Then the major axis (a) and minor axis (b) of the ellipse fitted to the region and the angle (θ) between the major axis and the horizontal axis of the image plane are estimated from the eigenvalues and corresponding eigenvectors of the covariance matrix. Given the parameters of the ellipse, a vector $\mathbf{s}_t = [a, a/b, \theta]^{\mathrm{T}}$ for describing the static hand shape (shape appearance) in the frame at time t is constructed. Note that a/b is the length ratio of the major axis to the minor one.

4.4. Spatiotemporal Representation

Given an image sequence with L frames containing a hand gesture, let I_t ($t = 0, \ldots, L - 1$) denote the tth frame. After having estimated the motion appearance $\mathbf{m}_t$ between frame I_t and I_{t+1}, and the shape features $\mathbf{s}_t$ for frame I_t, we model the interframe appearance change by combining the motion and shape features as $\mathbf{f}_t = [\mathbf{m}_t, \mathbf{s}_t]^{\mathrm{T}} = [m_1, m_2, m_3, m_4, m_5, m_6, m_7, a, a/b, \theta]^{\mathrm{T}}$. The spatiotemporal representation ($\mathbf{G}$) of the entire gesture is then defined as a trajectory of interframe appearance feature vectors ($[\mathbf{f}_0, \mathbf{f}_1, \ldots, \mathbf{f}_{L-2}]$).

5. GESTURE RECOGNITION

5.1. Time Normalization

Since the gesturing rate may vary between different individuals and at different times for the same individual, elimination of this fluctuation has been an important issue in recognition of hand gestures. In speech recognition, dynamic time warping (DTW) was proposed to match a test pattern with a reference pattern if their time scales were not perfectly aligned [42]. DTW assumes that the endpoints of the two patterns have been accurately located and formulates the problem as finding the optimal path from the start to the end on a finite grid. The optimal path can be found by dynamic programming.

However, results of our experiments found that DTW was rather ineffective for matching two spatiotemporal representations. Unlike the high sampling rate in speech recognition, the sampling rate is usually 10 Hz in hand gesture recognition. Comparatively, the local fluctuation in the time axis of hand gesture patterns is much sharper than that of speech patterns (see Table 4). Traditional DTW is insufficient for dealing with patterns with sharp fluctuation. On the other hand, as a result of enforcing the afore-mentioned interaction model, the extracted spatiotemporal representations can be linearly reparametrized. Keeping computational efficiency in mind, we propose a linear resampling technique for warping the spatiotemporal representations. Each warped representation has a fixed temporal period (called the *warping-length*). The normalization technique can maintain the essential spatiotemporal distribution of the original representations. Although different warping-lengths may be used for different types of gestures, the same warping-length will be applied to all instances of the same type of gesture.

TABLE 1

Motion-Feature-Warping Algorithm

▷ Input: $m_{i,t}, i = 1, \ldots, 7, t = 0, \ldots, L - 1$.

▷ Output: $\tilde{m}_{i,k}, i = 1, \ldots, 7, k = 0, \ldots, K - 1$.

Step 1: Compute the accumulated motion of the ith motion component $A_{i,t}$ from time 0 until time t,

$$A_{i,t} = \sum_{n=0}^{t} m_{i,n}, t = 0, \ldots, L - 1.$$

Step 2: Divide the temporal axis into $K - 1$ intervals delimited by times t_k

$$t_k = \frac{L(k+1)}{K} - 1.$$

Step 3: Using linear interpolation, estimate the total amount of motion $\tilde{A}_{i,t_k}$ of the ith motion component from time 0 until time t_k according to

$$\tilde{A}_{i,t_k} = (t_k - \lfloor t_k \rfloor) A_{i,t+1} + (\lceil t_k \rceil - t_k) A_{i,t},$$

where $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ are respectively the floor and ceiling operations.

Step 4: Estimate the warped motion of each component, $\tilde{m}_{i,k}$,

$$\tilde{m}_{i,k} = \tilde{A}_{i,t_k} - \tilde{A}_{i,t_{k-1}},$$

where $\tilde{A}_{i,-1} = 0$.

Note. The motion-feature-warping algorithm is explained using an original sequence of L frames being warped to K frames ($K \leq L$).

We warp the motion components $\mathbf{m}_t$ and the shape components $\mathbf{s}_t$ in different ways since the image motion is a velocity measurement while the shape appearance characterizes a static configuration of the hand shape. Tables 1 and 2 describe respectively the motion-appearance-warping and shape-appearance-warping algorithms. Using the two algorithms, we can warp the motion components $m_{i,t}$ and shape components $s_{j,t}$ to temporally normalized ones $\tilde{m}_{i,k}$ and $\tilde{s}_{j,k}$, respectively. To see more clearly the functionality of the proposed warping algorithms, Fig. 2 shows us the trajectory of a motion component of an original spatiotemporal representation and the corresponding trajectory generated by the proposed motion-appearance-warping algorithm.

TABLE 2

Shape-Feature-Warping Algorithm

▷ Input $s_{j,t}, j = 1, 2, 3; t = 0, \ldots, L - 1$.

▷ Output $\tilde{s}_{j,k}, j = 1, 2, 3; k = 0, \ldots, K - 1$.

Step 1: Divide the temporal axis into $K - 1$ intervals delimited by times t_k

$$t_k = \frac{L(k+1)}{K} - 1.$$

Step 2: Using linear interpolation estimate the shape component $\tilde{s}_{j,k}$ at time instance t_k, according to

$$\tilde{s}_{j,t_k} = (t_k - \lfloor t_k \rfloor) s_{j,t+1} + (\lceil t_k \rceil - t_k) s_{j,t},$$

where $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ are respectively the floor and ceiling operations.

Note. The shape-feature-warping algorithm is explained using an original sequence of L frames being warped to K frames ($K \leq L$).

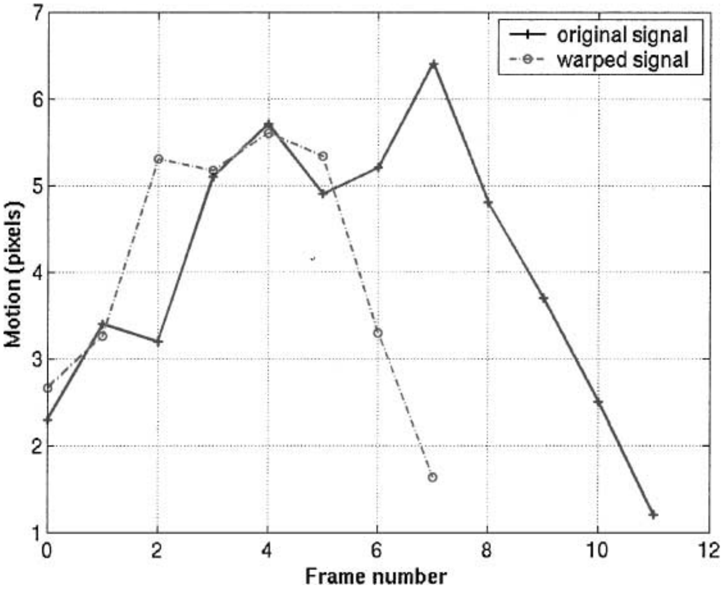


FIG. 2. Illustration of the functionality of the motion-feature-warping algorithm by applying it to the motion component $m_{2,t}$ of an example spatialtemporal representation. The temporal length of the original representation and that of the warped one are 12 and 8 frames, respectively.

5.2. Measuring Distance

Using the above time normalization technique, a distance measure between two spatiotemporal representations can be established using the normalized correlation between their warped representations. Given two spatiotemporal representation $\mathbf{A} = (a_{ij})_{10\times T_1}$ and $\mathbf{B} = (b_{ij})_{10\times T_2}$ with T_1 and T_2 frames long, respectively, we define the distance between them as

$$D(\mathbf{A},\mathbf{B}) = 1 - \frac{\sum_{j=0}^{K-1}\sum_{i=0}^{9}(w_i\tilde{a}_{ij})(w_i\tilde{b}_{ij})}{\sqrt{\sum_{j=0}^{K-1}\sum_{i=0}^{9}(w_i\tilde{a}_{ij})^2}\sqrt{\sum_{j=0}^{K-1}\sum_{i=0}^{9}(w_i\tilde{b}_{ij})^2}}, \tag{14}$$

where $\tilde{\mathbf{A}} = (\tilde{a}_{ij})_{10\times K}$ and $\tilde{\mathbf{B}} = (\tilde{b}_{ij})_{10\times K}$ are the representations warped to K frames. The weights w_i ($i = 0, \ldots, 9$) are taken to be the inverse of the standard deviation of the features computed over the entire training set.

5.3. Template-Based Recognition

The final step of our gesture recognition system is the creation of reference templates for use in the recognition phase. Assume we are given a training set of gestures, for each of the C types of hand gestures in the vocabulary, there are J samples made by different subjects. To create a single reference template for each gesture, the J versions of the gesture must somehow be combined. Given that the variation of the same gesture as performed by different subjects can be quite large, simply averaging the time-normalized spatiotemporal representations of each type of gesture was found to be ineffective. We therefore use a minimax selection technique, which has been successfully applied to spoken word recognition [43] and landmark selection [44], to select a reference template for each type of hand gesture. Under the minimax criterion, the reference template is chosen so that its maximum distance to any of the J samples is minimized. Essentially, the training sample most representative is used as a reference template for the gesture.

Let the spatiotemporal representations of all gesture samples be $\mathbf{G}_{c,j}$, $c = 1, \ldots, C$, $j = 1, \ldots, J$; the reference template for gesture c is taken to be $\mathbf{G}_{c,\hat{j}}$ where

$$\hat{j} = \underset{j}{\arg\min}\ \underset{k}{\arg\max}\ D(\mathbf{G}_{c,j}, \mathbf{G}_{c,k}).$$

With $\mathbf{G}_{c,\hat{j}}\,(c = 1, \ldots, C)$, we apply template-based classification techniques to gesture recognition in the following manner. Given a test image sequence, first extract the gesture's spatiotemporal representation, and then compute its distance to each of the C reference templates; afterward, find the minimal distance and record the corresponding template. If the minimal distance is below a prescribed threshold, then the test gesture belongs to the class in which the reference template resides; otherwise it is rejected.

6. EXPERIMENTAL RESULTS AND ANALYSIS

Two types of experiments were performed: off-line isolated hand gesture recognition and real-time on-line gesture recognition. The goal of the first set of experiments was to examine the acceptability and effectiveness of the proposed approach. In a second set, a prototype gesture-controlled panoramic map browser was built to demonstrate the use of hand gestures for HCI. Since the error rate of real-time on-line gesture recognition depends greatly on the gesture reference templates generated in the first stage, experiments in the first stage were executed carefully. Additional experiments were also conducted in the first stage to investigate the impact of the image motion models and the warping-length on the recognition performance.

The prototype system, illustrated in Fig. 3, is running on a Pentium II (266 MHz) PC with Windows 98. The computer is equipped with a Matrox Meteor board and a Sony Video Hi8 Pro video camera. When using the gesture-controlled map browser, users control the movement of a panoramic map by issuing gesture commands. The command set consists of

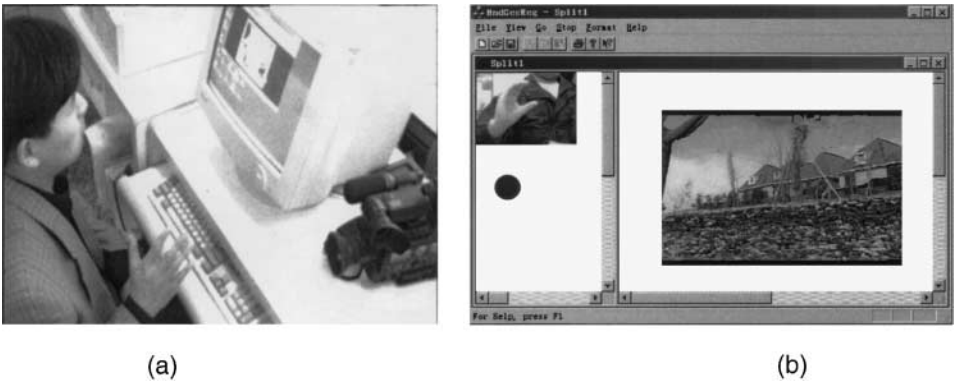
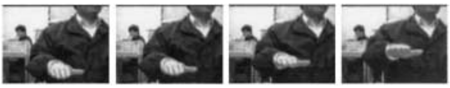
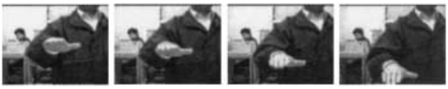
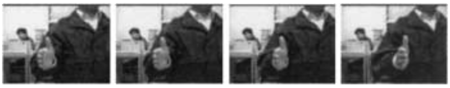
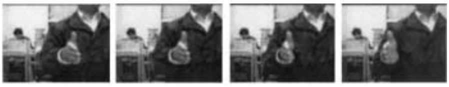
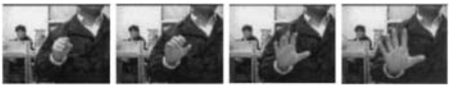
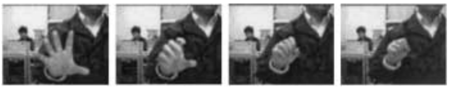
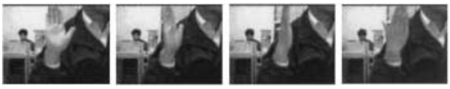
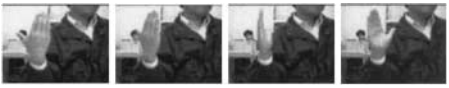
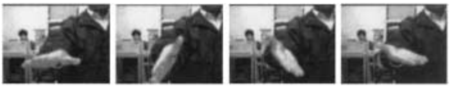
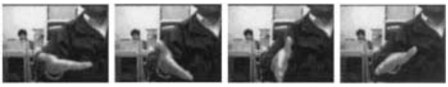
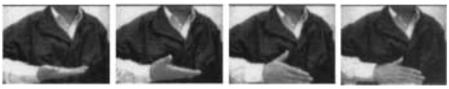
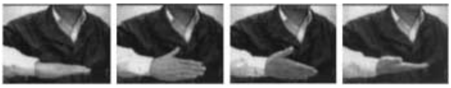


FIG. 3. Experimental setting and user interface of the prototype gesture-controlled panoramic map browser. (a) A user sits in front of a desktop computer and gestures at a video camera connected to the computer. (b) The whole window screen is split into two child windows. The real-time captured video and the system state indicator are displayed respectively in the upper and lower parts of left child window. The right child window shows the visible part of the panoramic map constructed on-site from the image sequence ("flower garden") using image mosaic.

TABLE 3
The Twelve Types of Hand Gesture Commands Used in the Prototype System

No.	Name	Function	Sample image frames
1	Scroll-up	Scroll up by a unit height.	
2	Scroll-down	Scroll down by a unit height.	
3	Scroll-left	Scroll left by a unit width.	
4	Scroll-right	Scroll right by a unit width.	
5	Zoom-in	Zoom in by a unit.	
6	Zoom-out	Zoom out by a unit.	
7	Yaw-right	Yaw right by a unit angle.	
8	Yaw-left	Yaw left by a unit angle.	
9	Rotate-clockwise	Rotate clockwise by a unit angle.	
10	Rotate-anticlockwise	Rotate anticlockwise by a unit angle.	
11	Pitch-down	Pitch down by a unit angle.	
12	Pitch-up	Pitch up by a unit angle.	

12 types of gestures. The name, function definition, and some sample frames of each type of gesture command are listed in Table 3.

6.1. Isolated Hand Gesture Recognition

Five subjects were asked to perform the 12 types of gestures in front of the video camera (see Fig. 3). Each gesture was repeated twice. Therefore, a total of 120 gesture samples (color image sequences) were spotted from video and collected as the experimental data set. Each sample gesture lasts about 1 s. Images are 160×120 pixels with 24 bits per pixel (taken at 10 Hz). To effectively utilize the available samples, we use the leave-one-out cross-validation method (L-method), detailed in [45], to evaluate the proposed approach. Specifically, one

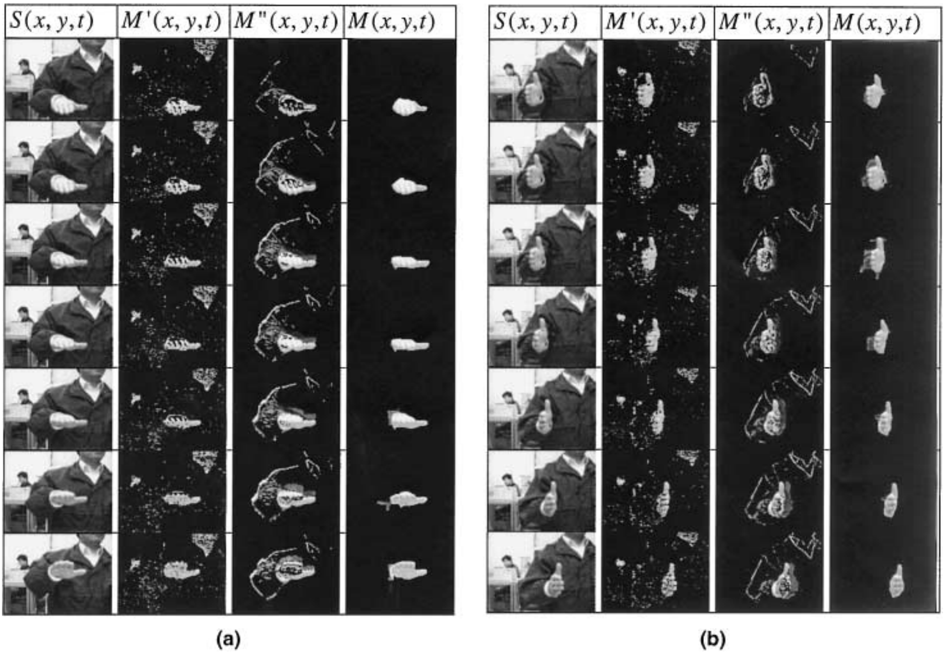


FIG. 4. Examples of real-time spotted gesture instances obtained using the algorithm described in Section 3. (a)–(f) illustrate an example gesture from respectively the scroll-up, scroll-left, zoom-out, rotate-clockwise, yaw-left, and pitch-down types of gesture commands. Each of the six examples has four columns representing, from left to right respectively, the original color image sequence $S(x, y, t)$, the skin region image sequence $M'(x, y, t)$ generated using the color cue, the moving region image sequence $M''(x, y, t)$ produced using motion cue, and the final image sequence $M(x, y, t)$ containing only the moving hand region from which both the human faces and moving clothes are successfully separated using multiple cues. The image sequences in the forth columns are passed on to the spatiotemporal feature extraction module.

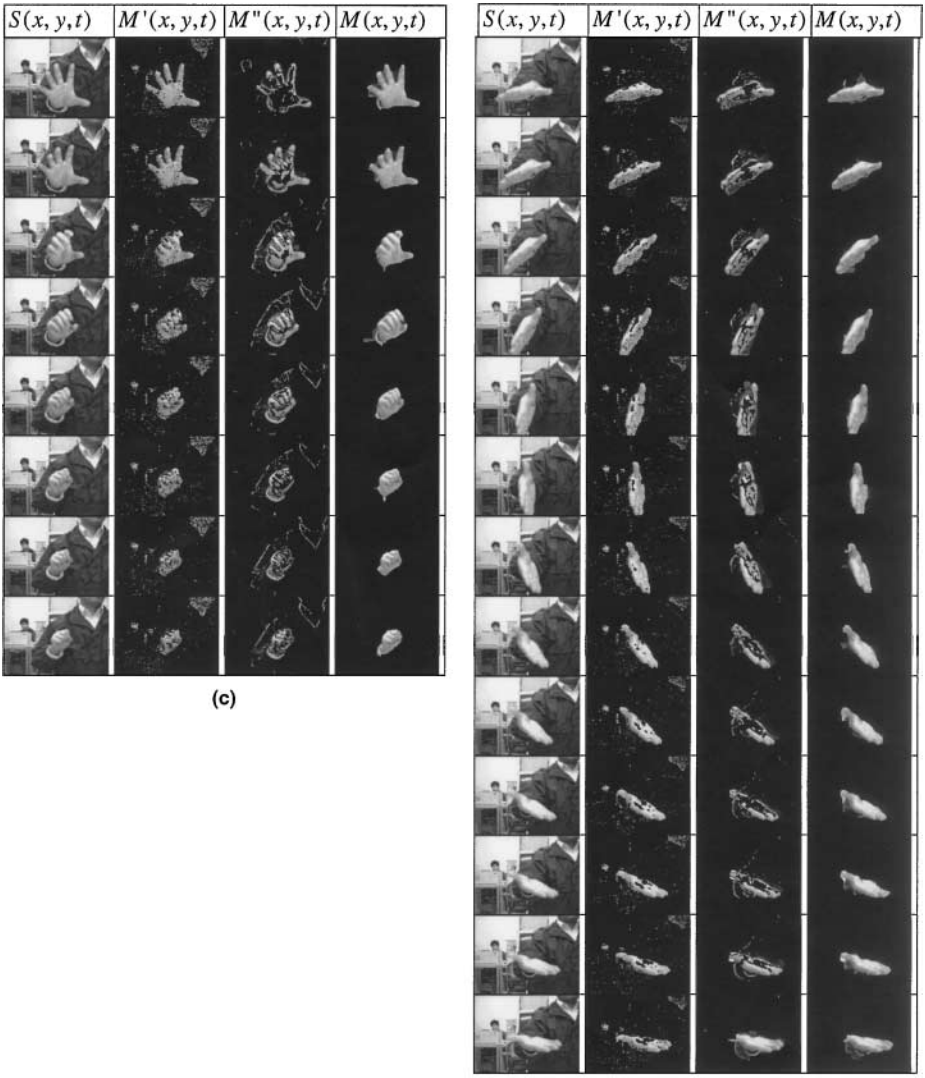
sample is excluded for testing the classifier, which in turn is trained on the remaining 9 samples. This operation is repeated 10 times to test all 10 samples for each type of gestures. Then the number of misclassified samples is counted to estimate the average error. Time for spotting, feature extraction, and recognition of one gesture instance is about 2 s.

6.1.1. Real-Time Hand Gesture Spotting

The first step of the gesture recognition system is spotting gestures from a real-time video stream using the algorithm described in Section 3. Figure 4 shows six examples of the spotted gesture instances. Images in Fig. 4 indicate that the moving hand image segmented using multiple cues is quite clean and complete even though the intermediate results generated using a single cue are noisy and/or unreliable. For example, both the human face and moving clothes (jacket) image regions were successfully separated from the moving hand region. The results generated using the spotting algorithm is sufficient for higher level processing—extraction of the spatiotemporal representations.

6.1.2. Determining Warping-Length

This experiment examines how varying the value of warping-length affects the system's performance. Table 4 lists the statistics of temporal lengths of all gesture samples, showing that the gesturing rate varies quite often, sometimes even significantly, from person to



(c)

(d)

FIG. 4—*Continued*

person as well as from time to time. For instance, within the training set, the longest instance of gesture No. 11 has 13 frames while the shortest one has only 7 frames. The statistics demonstrate that the recognition technique must be able to cope with the variation in gesturing rate.

In order to determine an appropriate warping-length, we first need to find out the range of the warping-length. Table 4 tells us that the longest gesture instance has 15 frames. The longest duration of all extracted spatiotemporal representations on the data set, therefore, is 14 frames. As such, the possible warping-length ranges from 1 to 14 frames. One by one, we examine the recognition performance of the system on the data set using the possible warping-lengths with the planar model of image motion. The experiment results, shown in Fig. 5, indicate that the average recognition accuracy is 83.33% when the warping-leangth is 1 frame (in other words, no temporal information is exploited). The average recognition

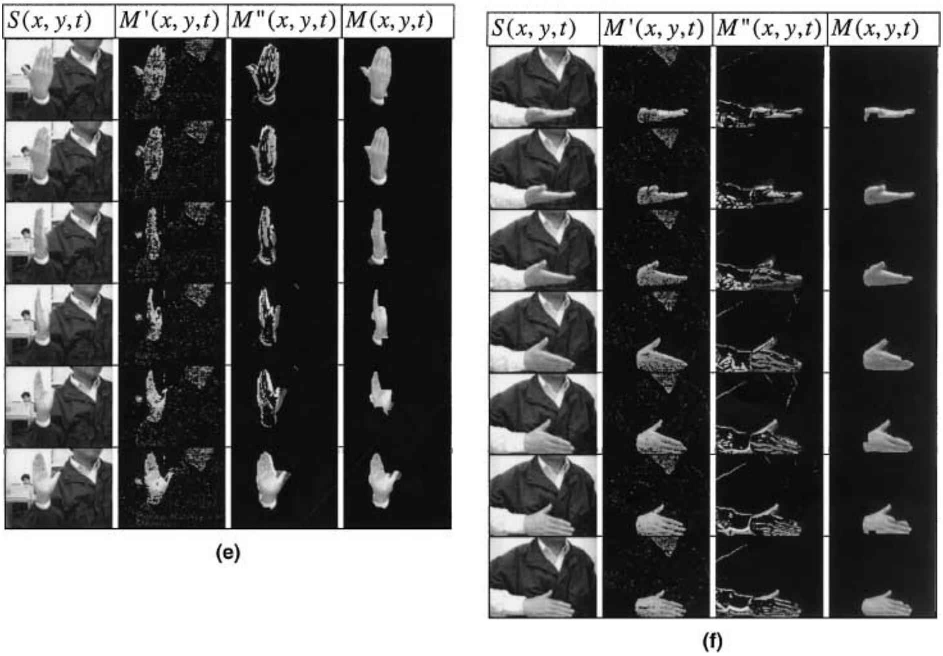


FIG. 4—*Continued*

accuracies reach the highest (90.83%) when the warping-length is 4, 6, or 8 frames. Since a smaller warping-length requires less computation, we set the warping-length to be 4 frames in most other experiments.

6.1.3. Selecting Appropriate Image Motion Model

Different image motion models have different discriminative capacities and involve different computational requirements. For modeling the motion of video containing hand gestures, should we use the planar model, the affine model, or even the translation model?

TABLE 4
Statistics of Temporal Length (Number of Frames) of Gesture Instances in the Data Set

Gestures no.	Maximum	Minimum	Mean	Variance
1	11	7	9.00	2.80
2	10	6	8.50	2.30
3	12	7	9.33	3.87
4	10	7	8.33	1.07
5	10	6	8.17	1.77
6	10	6	7.83	2.57
7	10	7	8.17	0.97
8	8	6	6.83	0.57
9	13	10	11.83	1.37
10	13	9	10.67	1.87
11	13	7	10.00	4.00
12	15	8	10.33	6.27

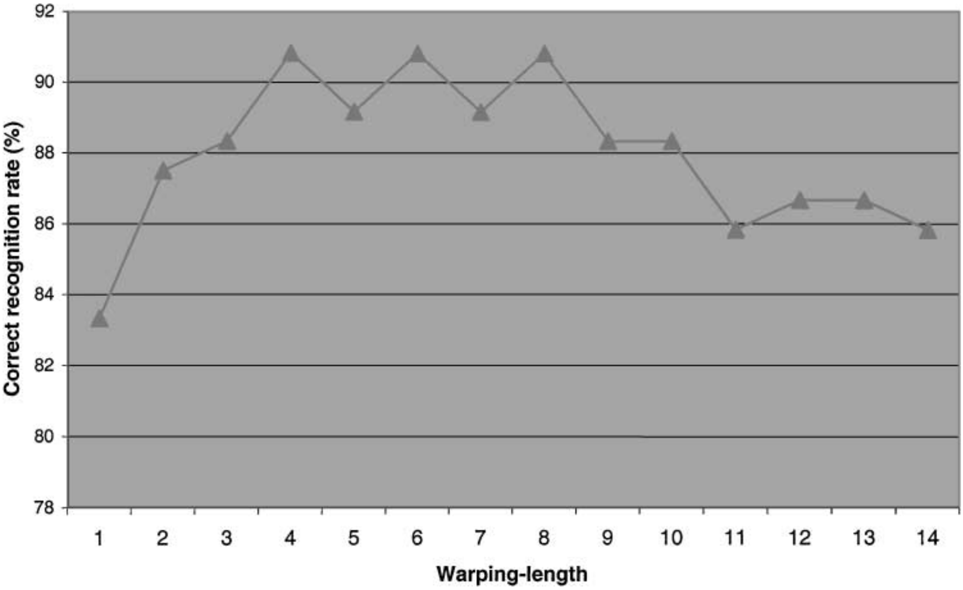


FIG. 5. Correct recognition rates vs warping-length ranged from 1 to 14 frames, obtained using the leave-one-out cross-validation method. The result was achieved on the data set containing 120 gesture instances (there are 12 type of gestures, each has 10 samples). The best recognition performance (90.83% accuracy on average) was attained when the warping-length is 4, 6, or 8 frames.

The answer depends on the size of the gesture command sets, usage of hand gestures, expectation of the recognition performance, and requirement of processing speed. We have experimented with different image motion models separately. Experimental results, listed in Table 5, reveal that the planar model is most appropriate in terms of recognition performances for the gesture command set used in our experiment. Considering the significant difference between the computational complexity of recovering the planar model parameters and that of the affine model parameters, the latter may be more desirable since their average accuracies are close to one another.

6.1.4. Motion Appearance versus Shape Appearance

The spatiotemporal appearance is modeled by combining both motion and shape appearances. In order to explore the individual discriminative capacity of motion information vs shape information, we did two additional experiments. In the first experiment, only the components characterizing motion were used for recognition while in the second experiment only the components characterizing shape appearances were used. The average recognition accuracies are, respectively, 82.50 and 68.33%, using the planar motion model and a warping-length of four frames. Recall that the average accuracy obtained using both shape

TABLE 5
Recognition Performances vs Motion Models Obtained Using the Cross-Validation Leave-One-Out Method

Image motion model	Translation	Affine	Planar
Average accuracy (%)	73.33	87.50	90.83

and motion information is 90.83%; it is obvious that shape appearance is indispensable for achieving higher recognition performance even though the discriminative capacity of motion appearance is dominant.

6.2. Real-Time Gesture-Controlled Panoramic Map Browser

To demonstrate using gestures in HCI, we designed and implemented a prototype gesture-controlled panoramic map browser (see Fig. 3). The system couples two functional parts seamlessly, namely the module for building a panoramic map from multiple views of a scene, and the gesture recognition module that interprets input gestures as control commands. In order to have an entire impression of the panoramic map, users navigate in a 3-D virtual space by issuing gesture commands.

For on-line gesture recognition, all the available gesture instances in the data set were used to create reference templates. Five different users were asked to use the real-time gesture-controlled panoramic map browser. The accuracy of the on-line hand gesture recognition ranged from 80 to 90% for an instantly trained user.

7. SUMMARY AND DISCUSSION

7.1. Summary

An efficient approach for spotting, spatiotemporal appearance modeling, and recognition of hand gestures from real-time video stream has been presented. First, an interaction model for using gestures in HCI is suggested. By integrating the interaction model, color cue-based hand detection, and coarse image motion detection, image sequences containing a single gesture are segmented from video streams in real-time. Then the image sequences are modeled with spatiotemporal representations that characterize both the motion and shape appearance changes in the sequences caused by the hand motion. The representations are trajectories of interframe feature vectors consisting of both motion and shape components. The motion components are estimated using robust parameterized image motion regression, and the shape components are recovered using the geometrical features of an ellipse fitted to the hand image region. Finally, a fast linear time normalization technique for eliminating temporal variations of the gesture representations is developed. The technique can maintain the essential temporal and spatial distribution of the original representations. After time normalization, gestures are successfully recognized using a template-based classification technique.

Both off-line (isolated gesture) and on-line experiments were conducted and reported. For isolated gesture recognition, we did additional experiments to investigate the impact of the choice of the image motion models and the warping-lengths (a key parameter used in the linear time normalization) on the performance of recognition. For the on-line experiment, a prototype gesture-controlled panoramic map browser was designed and implemented to demonstrate using gestures in HCI.

7.2. Discussion

After examining the misclassified gestures instances, we found that most of the misclassified gesture instances are made by one particular subject. To further test the performance of the proposed approach, we generated a new data set by removing the samples made by this subject from the original data set. As a result, there are 9 samples left for each of the 12 types of gestures. We then ran the afore-mentioned cross-validation on the new data set using all possible warping-length. As we expected, better performances were achieved (the highest average correct recognition rate is 93.52%).

The successful application of HMMs in speech recognition [46] has led their way into the community of gesture recognition. Many researchers have explored using HMM in sign language recognition. The statistical nature of HMM provides a good framework for gesture recognition, but as Bobick and Wilson mentioned [19], it precludes a rapid training phase, and many parameters need to be adjusted for the model to relate to gestures. By contrast, the simple time normalization technique proposed in this paper provides a fast training process and efficient computation. On the other hand, we have noticed that the warping-length, the key parameter used in the technique, has a close relationship to the number of states, also a key parameter used in HMMs. It would be interesting to compare the recognition rates achieved on a same data set using the two methods. Finally, we emphasize that to couple HMMs into the work presented here is straightforward since the spatiotemporal representations of hand gestures can be extracted successfully using the techniques reported in the paper.

Our work adopted an interaction model under which 12 types of predefined gestures were recognized with satisfactory accuracy. This kind of framework is suitable for applications similar to that present in the paper. As observed in [29], current systems are unable to recognize natural gestures that people spontaneously make. In most cases, the communication between user and computer is restricted to a few “special” gestures. Considering the three reasons mentioned in the beginning of the paper, we argue that it is more feasible and applicable for computers to interpret gestures within an interaction model even in the near future. A deeper understanding of gestures and user’s intent is needed in order to develop a proper interaction model capturing the best trade-off between the naturalness of the gestures and the necessary computational complexity.

In this paper, a moving object with skin color in the scene is assumed to be the gesturing hand, which could be invalid when there are other moving objects with the similar color, for instance, a moving human face. Obviously, introducing a model of the entire human body can alleviate this problem. It will be necessary to have more gesture commands in order to practically use gestures in HCI. Some commands may be more reasonably input by hand postures. Others may be better replaced by speech input. We are currently working on cooperating hand gesture recognition into a multimodal human–computer interface.

ACKNOWLEDGMENTS

The authors thank Professor Xieyin Lin and Dr. Yu Huang for discussions on various issues, Dr. Linmi Tao for his color segmentation code, Lin Song for his panorama building module, and Haibing Ren, Xiaoping Zhang, and Zhen Wen for their help in doing experiment. This research was funded by National 863 Hi-Tech Program of China under Contract 863-306-03-011. D. Kriegman was partially funded under NSF ITR NSF CCR 00-86094.

REFERENCES

1. J. Davis and M. Shah, Toward 3-D gesture recognition, *Internat. J. Pattern Recognit. Artif. Intell.* **13**, 1999, 381–393.

2. R. Koch, Dynamic 3D scene analysis through synthetic feedback control, *IEEE Trans. Pattern Anal. Mach. Intell.* **15**, 1993, 556–568.

3. E. Clergue, M. Goldberg, N. Madrane *et al.*, Automatic face and gesture recognition for video indexing, in *Proceedings, 1st Int’l Workshop on Automatic Face and Gesture Recognition*, 1995, pp. 110–115.

4. D. M. Gavrila and L. S. Davis, Towards 3D model-based tracking and recognition of human movement: A multi-view approach, in *Proceedings, 1st Int'l Workshop on Automatic Face and Gesture Recognition*, 1995, pp. 272–277.
5. J. J. Kuch, Vision-based hand modeling and tracking for virtual teleconferencing and telecollaboration, in *Proceedings, Int'l Conf. Computer Vision*, 1995, pp. 666–671.
6. J. M. Rehg and T. Kanade, Model-based tracking of self-occluding articulated objects, in *Proceedings, Int'l Conf. Computer Vision*, 1995, pp. 612–617.
7. J. Schlenzing, E. Hunter, and R. Jain, Recursive identification of gesture inputs using hidden markov models, in *Proceedings, 2nd IEEE Workshop Application of Computer Vision*, 1994, pp. 187–194.
8. J. L. Crowley, F. Berard, and J. Coutaz, Finger tracking as an input device for augmented reality, in *Proceedings, 1st Int'l Workshop on Automatic Face and Gesture Recognition*, 1995, pp. 195–200.
9. Y. Cui and J. J. Weng, Hand sign recognition from intensity image sequences with complex backgrounds, in *Proceedings, 2nd Int'l Conf. Automatic Face and Gesture Recognition, Killington*, 1996, pp. 259–264.
10. W. T. Freeman, Orientation histogram for hand gesture recognition, in *Proceedings, 1st Int'l Workshop on Automatic Face and Gesture Recognition*, 1995, pp. 296–301.
11. C. Magginoi, Gesture computer-New ways of operating a computer, in *Proceedings, 1st Int'l Workshop on Automatic Face and Gesture Recognition*, 1995, pp. 166–171.
12. F. K. H. Quek, Eyes in the interface, *Image Vision Comput.* **13**, 1995, 511–525.
13. T. J. Darrell, I. A. Essa, and A. Pentland, Task-specific gesture analysis in real-time using interpolated views, *IEEE Trans. Pattern Anal. Mach. Intell.* **18**, 1996, 1236–1242.
14. R. Kjeldsen and J. Kender, Toward the use of gesture in traditional user interfaces, in *Proceedings, 2nd Int'l Conf. Automatic Face and Gesture Recognition, Killington*, 1996, pp. 151–156.
15. V. I. Pavlovic, R. Sharma, and T. S. Huang, Gestural interface to a visual computing environment for molecular biologists, in *Proceedings, Int'l Conf. Automatic Face and Gesture Recognition, Killington*, 1996, pp. 30–35.
16. A. Bobick and J. Davis, Real-time recognition of activity using temporal templates, in *Proceedings, 3rd IEEE Workshop Application of Computer Vision*, 1996, pp. 39–42.
17. R. Cipolla and N. J. Hollinghurst, Human-robot interface by pointing with uncalibrated stereo vision, *Image Vision Comput.* **14**, 1996, pp. 171–178.
18. T. Ahmad, C. J. Taylor, A. Lanitis, *et al.*, Tracking and recognizing hand gestures, using statistical shape models, *Image Vision Comput.* **15**, 1997, 345–352.
19. A. F. Bobick and A. D. Wilson, A state-based approach to the representation and recognition of gesture, *IEEE Trans. Pattern Anal. Mach. Intell.* **19**, 1997, 1325–1337.
20. T. Starner, J. Weaver, and A. Pentland, Real-time American Sign Language recognition using desk and wearable computer based video, *IEEE Trans. Pattern Anal. Mach. Intell.* **20**, 1998, 1371–1375.
21. R. Culter and M. Turk, View-based interpretation of real-time optical flow for gesture recognition, in *Proceedings, 3rd Int'l Conf. Automatic Face and Gesture Recognition*, 1998.
22. Y. Zhu, Y. Huang, G. Xu, *et al.*, Vision-based interpretation of hand gestures by modeling appearance changes in image sequences, in *Proceedings, IAPR Workshop Machine Vision Applications, Tokyo*, 1998, pp. 573–576.
23. Y. Zhu, Y. Huang, G. Xu, *et al.*, Motion-based segmentation scheme to feature extraction of hand gestures, in *Proceedings, SPIE Vol. 3545*, 1998, pp. 228–231.
24. Y. Huang, Y. Zhu, G. Xu, *et al.*, Spatial-temporal features by image registration and warping for dynamic gesture interpretation, in *Proceedings, IEEE Conf. Systems, Man, and Cybernetics, California*, 1998, pp. 4498–4503.
25. H.-K. Lee and J. H. Kim, An HMM-based threshold model approach for gesture recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* **21**, 1999, 961–973.
26. G. Brakski, B.-L. Yeo, and M. M. Yeung, Gesture for video content navigation, in *Proceedings, SPIE Vol. 3656*, 1998, pp. 230–242.
27. Y. Zhu, H. Ren, G. Xu, and X. Lin, Toward real-time human-computer interaction with continuous dynamic hand gestures, in *Proceedings, 4th IEEE Int'l Conf. Automatic Face and Gesture Recognition, Los Alamitos, CA*, 2000, pp. 544–549.

28. V. Pavlovic, R. Sharma, and T. S. Huang, Visual interpretation of hand gestures for human-computer interaction: A review, *IEEE Trans. Pattern Anal. Mach. Intell.* **19**, 1997, 677–695.

29. A. Pentland, Looking at people: Sensing for ubiquitous and wearable computing, *IEEE Trans. Pattern Anal. Mach. Intell.* **22**, 2000, 107–119.

30. W. T. Freeman, K. Tanaka, J. Ohta, *et al.*, Computer vision for computer games, in *Proceedings, Int'l Conf. Automatic Face and Gesture Recognition, Killington*, 1996, pp. 100–105.

31. D. A. Becker, Sensi: A Real-time recognition, feedback and training system for T'ai Chi gestures, Technical Report TR-426, MIT Media Lab, 1997.

32. C. Vogler and D. Metaxas, ASL recognition based on a coupling between HMMs and 3D motion analysis, in *Proceedings, Int'l Conf. Comput. Vision, Bombay*, 1998.

33. D. McNeill, *Hand and Mind*, Univ. of Chicago Press, Chicago, IL, 1992.

34. T. Baudel and M. Beaudouin-Lafon, Charade: Remote control of objects using free-hand gestures, *Commun. ACM* **36**, 1993, 28–35.

35. M. J. Jones and J. M. Rehg, Statitical color models with application to skin detection, in *Proceedings of IEEE Int'l Conference on Computer Vision and Pattern Recognition (CVPR'99)*, Vol. I, 1999, pp. 274–280.

36. L. Tao and G. Xu, Computer color constancy in two steps, *J. Tsinghua Univ.* **40**, 2000, 101–104.

37. J. R. Bergen, P. Anandan, K. J. Hanna, *et al.*, Hierarchical model-based motion estimation, in *Proceedings, 2nd European Conf. Comput. Vision*, 1992, pp. 237–252.

38. M. J. Black and P. Anandan, The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields, *Comput. Vision Image Understand.* **63**, 1996, 75–104.

39. G. Adiv, Determing three-dimensional motion and structure from optical flow generated by several moving objects, *IEEE Tans. Pattern Anal. Mach. Intell.* **7**, 1985, 384–401.

40. M. J. Black and Y. Yacoob, Tracking and recognizing rigid and non-rigid face motion using local parametric models of image motion, in *Proceedings, Int'l Conf. Computer Vision*, 1995, pp. 374–381.

41. R. Cipolla, Y. Okamoto, and Y. Kuno, Robust structures from motion using motion parallax, in *Proceedings, IEEE Int'l Conf. Comput. Vision*, 1993, pp. 374–382.

42. H. Sakoe and S. Chiba, Dynamic programming optimization for spoken word interpretation, *IEEE Trans. Acoust. Signal Speech Patterns* **26**, 1978, 43–49.

43. L. R. Rabiner, On creating reference templates for speaker independent recognition of isolated words, *IEEE Trans. Acoust. Signal Speech Patterns* **26**, 1978, 34–42.

44. M. Knapek, R. Swain, and D. Kriegman, Selecting promising landmarks, in *Proceedings, IEEE Conference on Robotics and Automation*, 2000, pp. 3771–3777.

45. K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed., Academic Press, San Diego, 1990.

46. L. R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, *Proc. IEEE.* **77**, 1989, 257–286.