Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition 41 (2008) 1124–1137
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/pr

Gesture spotting for low-resolution sports video annotation ☆

Myung-Cheol Roh [a], Bill Christmas [b], Joseph Kittler [b], Seong-Whan Lee [a,*]

[a] Department of Computer Science and Engineering, Korea University, Anam-dong, Seongbuk-ku, Seoul 136-713, Korea
[b] Center for Vision, Speech, and Signal Processing, University of Surrey, Guildford GU2 7XH, UK

Received 13 May 2006; received in revised form 30 May 2007; accepted 16 July 2007

Abstract

Human gesture recognition plays an important role in automating the analysis of video material at a high level. Especially in sports videos, the determination of the player's gestures is a key task. In many sports views, the camera covers a large part of the sports arena, resulting in low resolution of the player's region. Moreover, the camera is not static, but moves dynamically around its optical center, i.e. pan/tilt/zoom camera. These factors make the determination of the player's gestures a challenging task. To overcome these problems, we propose a posture descriptor that is robust to shape corruption of the player's silhouette, and a gesture spotting method that is robust to noisy sequences of data and needs only a small amount of training data. The proposed posture descriptor extracts the feature points of a shape, based on the curvature scale space (CSS) method. The use of CSS makes this method robust to local noise, and our method is also robust to significant shape corruption of the player's silhouette. The proposed spotting method provides probabilistic similarity and is robust to noisy sequences of data. It needs only a small number of training data sets, which is a very useful characteristic when it is difficult to obtain enough data for model training. In this paper, we conducted experiments spotting serve gestures using broadcast tennis play video. From our experiments, for 63 shots of playing tennis, some of which include a serve gesture and while some do not, it achieved 97.5% precision rate and 86.7% recall rate.

Keywords: Posture descriptor; Posture determination; Gesture spotting; Low resolution video annotation

1. Introduction

The development of high-speed digital cameras and video processing technology has resulted in increased attention being given to automated video analysis such as surveillance video analysis, video retrieval and sports video analysis. In particular, applying this technology to sports video creates many potential applications: automatic summaries of play, highlight extraction, winning pattern analysis, adding virtual advertisements, etc. There is some interesting research on ball tracking, player tracking and stroke detection for tennis, baseball, soccer, American football, etc, [1–3].

Although there has been much discussion in the literature on automatic sports video annotation and gesture recognition in a restricted environment, there has been little research into the detection/recognition of a player's gestures in standard off-air video, due to the low resolution of the player's region, the fast motion of the player, and camera motion [3,4]. In sports video, the player's region often has low resolution because the audience wants to watch a broad view of the scene in order to understand the persons' situation and relative position in the field. The same is often true in surveillance video where fixed cameras are used. The camera motion can also make the process of tracking players and extracting players' silhouettes difficult, and low resolution makes the determination of the player's posture unreliable. Hence, because of the unreliable determination of postures, recognition and detection of gestures can also be a difficult task to accomplish.

In this paper, we suggest a new type of posture descriptor to represent the player's silhouette, together with a novel gesture spotting method. Because of the problems of noise and low resolution, the player's silhouette is often significantly corrupted. Therefore, we propose a posture descriptor that is robust to noise and corruption. However, in practice, obtaining a

☆ A preliminary version of this paper has been presented in Ninth European Conference on Computer Vision, Graz, Austria, May 2006.

* Corresponding author. Tel.: +82 2 3290 3197, fax: +82 2 926 2168.
E-mail addresses: mcroh@image.korea.ac.kr (M.-C. Roh), w.christmas@surrey.ac.uk (B. Christmas), j.kittler@surrey.ac.uk (J. Kittler), swlee@image.korea.ac.kr (S.-W. Lee).

0031-3203/$30.00
doi:10.1016/j.patcog.2007.07.013

large set of training gesture data in low resolution video is very difficult to achieve. The proposed spotting method can work very well with a small number of gesture sequences to train it, and is robust to noise. We found the combination of the posture descriptor and spotting method to be efficient and robust to the problems of noise and low resolution that we encountered using standard off-air video.

2. Related works

Sullivan et al. proposed a method of detecting tennis players' strokes, based on the qualitative similarity, which computes the point-to-point correspondence between shapes by combinatorial geometric hashing [4]. They demonstrated that specific human actions can be detected from single frame postures in a video sequence with higher resolution than that typically found in broadcast tennis video. Although they presented interesting results for their video sequence, the method has some shortcomings. The outline of the player will not be extracted accurately in low resolution environments. Often, we can see only the player's back while he or she is playing, so we cannot use the information of the player's arm because of self-occlusion. Kopf et al. proposed a shape-based posture and gesture recognition method using a new curvature scale space (CSS) method in a video recorded by a pan/tilt/zoom camera [5]. Their new CSS representation can describe convex segment of shape as well as the concave one. However, their test sequence was of good quality, with good resolution of the player. They also recognized postures, rather than gestures. To date, there is a considerable quantity of literature on human gesture recognition using 3D, but these methods are difficult to apply to low-resolution video which shows the player's back view; also, they are not computationally very efficient [6,7].

There are many sequence recognition and matching methods which consider time information, and which have given interesting results in particular environments [8,9]. However, in special cases such as broadcast sports video, we may not have sufficient data to train a recognizer such as an HMM. Some methods, such as dynamic time warping (DTW), which computes the distance between two time series of given sample rates, provide similarity measures [10]. However, these methods have a high computational cost and do not associate any probabilities with the results. An extension of DTW, continuous dynamic programming (CDP), was proposed by Oka [11], which is the baseline algorithm used for comparing the results of our proposed gesture spotting algorithm. Alon et al. proposed a gesture spotting CDP algorithm via pruning and sub-gesture reasoning [12]. Their method showed an 18% increase in recognition accuracy over the CDP algorithm for video clips of two users drawing the 10 digits, '0' to'9', in an office environment.

3. Sports player posture determination and gesture spotting

Our gesture spotting system consists of four parts: foreground separation, posture description of the player's silhouette, posture determination and gesture spotting. We define a gesture as a set of postures. Therefore, a gesture is analyzed by using a set of postures. Fig. 1 shows a diagram of our gesture spotting system used to spot certain gestures using a posture set.

Firstly, foreground separation is performed to separate the foreground objects from the original frames using mosaicking [13]. As a result of the foreground separation, silhouettes of the player, ball and noise blobs are obtained and player's position can be tracked. Secondly, posture description is used to extract features which describe the silhouette of the player. Thirdly, posture determination is employed to determine which posture in the database is best matched to the player's silhouette. Fourthly, gesture spotting is done using a history of the determined postures, which is represented as a function of the time domain.

The frames in sports video often have serious motion blur due to camera motion and the player's movement. However, the *fields* in a video *frame* have less motion blur than the *frame* itself. Therefore, in our system, we extract the *fields* from *frames* and use them as input images instead of using *frames*.

3.1. Foreground separation

We assume that a camera moves only around its optical center, i.e. only pan/tilt/zoom are allowed. This presumption is reasonable, since most broadcast cameras are located in a fixed place while panning, tilting and zooming. This assumption makes the use of the mosaic technique feasible. Using the mosaic technique, each frame is projected into a single coordinate system, and a mosaic is created by the median filtering of the pixels. Then the mosaic is used as a background image.

The mosaic technique is divided into three stages. The first stage is correcting the camera geometries, which can be easily solved by projective transform. The second stage involves registering the images or, in other words, finding homographies. In this paper, corner points are used as features for registering the images. The third stage is image composition, which deals with the problem of how to determine the pixel values on the mosaic image. In order to remove moving objects from a sequence, median filtering is applied. This median filtering takes the median value from all corresponding pixels of the projected images, $I_M(x, y) = median(I_{p0}(x, y), I_{p1}(x, y), I_{p2}(x, y), \ldots, I_{pN}(x, y))$, where I_M and N are a pixel in a mosaic image and total number of input images, respectively, and I_{pn}, where $0 \leqslant n < N$, is a pixel in an input frame projected into the mosaic image coordinate. From the mosaic image, the foreground image can be obtained by taking the difference between the frame and mosaic image.

3.2. Posture description

A player's silhouette obtained from the foreground images is a posture and is described by the proposed posture descriptor. The silhouette of a player is extracted from the foreground image obtained in the previous stage. However, because of cam-

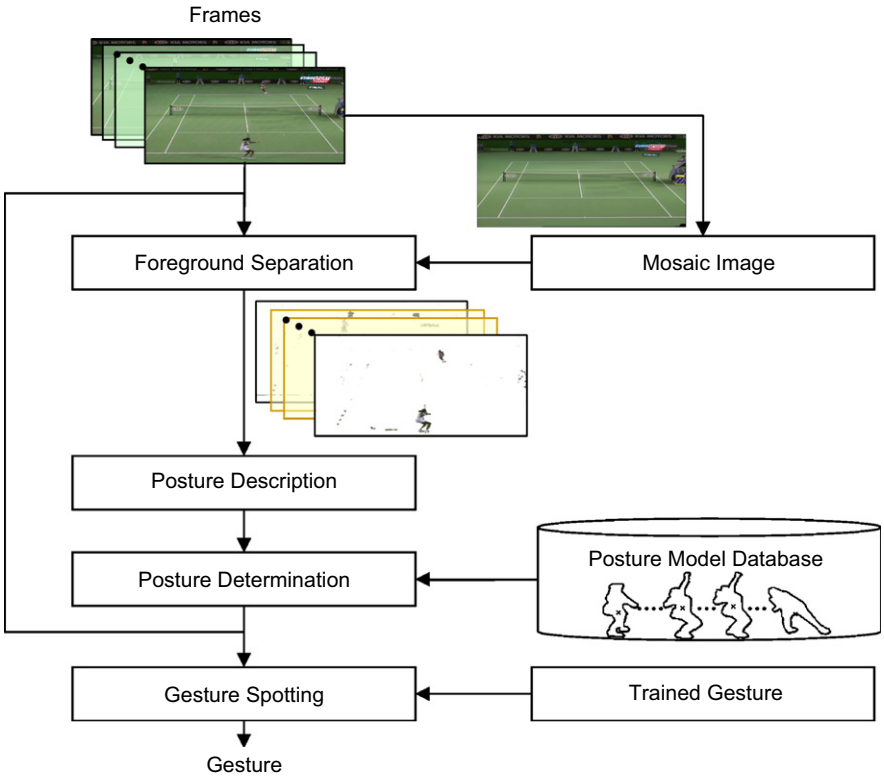

Fig. 1. Player's gesture annotation system.

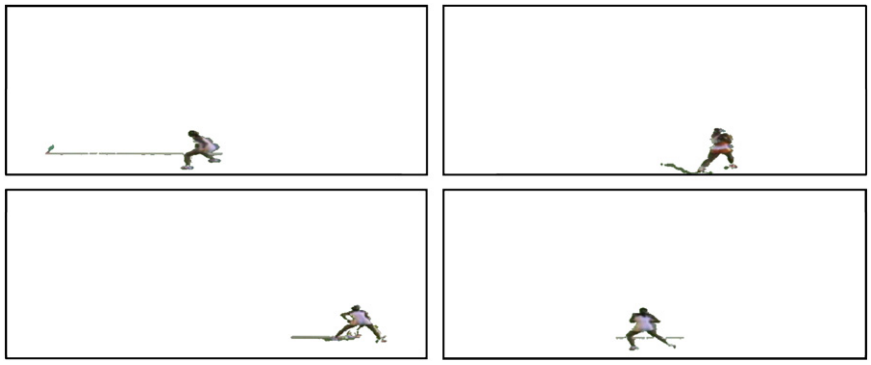

Fig. 2. Corrupted foreground images.

era movement, motion blur and low-resolution images, it is difficult to obtain a clear foreground image. The foreground is frequently corrupted by wrongly separated foreground. Fig. 2 shows some foreground images of a player. Therefore, a posture descriptor that is robust to corruption is needed. In this section, we introduce a well-established shape descriptor, CSS, which is robust to noise, and the proposed descriptor, which is based on the CSS, and is robust to corruption.

3.2.1. Curvature scale space

CSS is a well-established technique for describing shapes used in image retrieval, and is one of the descriptors used in the MPEG-7 standard [14]. We outline the CSS method here, paraphrasing the description in Ref. [14]. The CSS image of a planar curve is computed by convolving a path-based parametric representation of the curve with a Gaussian function of increasing variance σ^2, extracting the zeros of curvature of the convolved curves, and combining them in a scale space representation for the curve. These zero curvature points are calculated continuously while the planar curve is evolved by the expanding Gaussian smoothing function. Let the closed planar curve, r, be represented by the normalized arc length parameter, u:

$$r(u) = \{(x(u), y(u) | u \in [0, 1]\}. \quad (1)$$

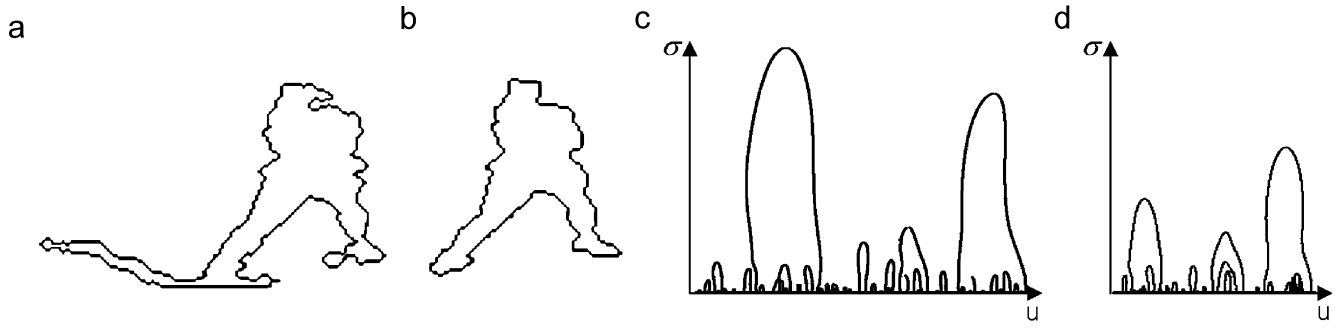

Fig. 3. Examples of foreground silhouette images and CSS images: (a) a silhouette image, (b) a posture model, (c) CSS image of (a), and (d) CSS image of (b).

Then, the evolved curve is represented by Γ_σ:

$$\Gamma_\sigma(u) = \{\chi(u,\sigma), \psi(u,\sigma)\}, \tag{2}$$

where

$$\chi(u,\sigma) = x(u) \otimes g(u,\sigma),$$

$$\psi(u,\sigma) = y(u) \otimes g(u,\sigma),$$

where g denotes a Gaussian function of width σ, and $\otimes$ is the convolution operator.

$$g(u,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-u^2/2\sigma^2}.$$

Then curvature of Γ is defined as

$$\kappa(u,\sigma) = \frac{\chi_u(u,\sigma)\psi_{uu}(u,\sigma) - \chi_{uu}(u,\sigma)\psi_u(u,\sigma)}{(\chi_u(u,\sigma)^2 + \psi_u(u,\sigma)^2)^{3/2}}, \tag{3}$$

where

$$\chi_u(u,\sigma) = \frac{\partial}{\partial u}(x(u) \otimes g(u,\sigma)) = x(u) \otimes g_u(u,\sigma),$$

$$\chi_{uu}(u,\sigma) = \frac{\partial^2}{\partial^2 u}(x(u) \otimes g(u,\sigma)) = x(u) \otimes g_{uu}(u,\sigma),$$

$$\psi_u(u,\sigma) = y(u) \otimes g_u(u,\sigma),$$

$$\psi_{uu}(u,\sigma) = y(u) \otimes g_{uu}(u,\sigma).$$

Then, CSS image I_c provides a multi-scale representation of zero crossing points by

$$I_c = \{(u,\sigma) | \kappa(u,\sigma) = 0, u \in [0,1], \sigma \geqslant 0\}. \tag{4}$$

The CSS image representation is robust to similarity transformation and (to a lesser extent) affine transformation, so significant peaks in CSS shape representation are considered suitable features for similarity-based retrieval. However, the drawback of this representation is that the zero crossing points of CSS are not reliable features, if some parts of the shape are significantly corrupted. Figs. 3(a) and (b) shows examples of a player's silhouette which is corrupted by noise blobs due to the low-quality of a video sequence and a posture model which is extracted manually, respectively. Figs. 3(c) and (d) shows CSS images of a foreground silhouette (a) and a posture model (b). Because there is significant corruption of a part of the silhouette, these two CSS features are not likely to be considered to be the same. Although the CSS descriptor is robust to noise, it is less reliable when describing shapes including significant corruption.

3.2.2. Proposed posture descriptor

The proposed posture descriptor is robust to corruption and distortion as well as local noise. A contour can be characterized and described by some points on the contour. The proposed descriptor describe a contour by feature points which derived from the CSS which characterize the contour by curvature.

Given a threshold, t, a new feature set, F, of a silhouette is defined by:

$$F = \{(x,y) | (x,y) = r(u), (u,t) \in I_c^t\}, \tag{5}$$

where $I_c^t = \{(u,\sigma) | \kappa(u,\sigma) = 0, \sigma = t\}$ and $u \in [0,1]$. The parameter t is determined from experiment and experiences. Although the value of t is not critical, according to our experience, the optimal value depends on the image quality and size of the contour. If t is too low, the computational cost for matching two contours will be increased because of the greater number of feature points. If t is too high, there will only be few feature points which will not be adequate to represent characteristics of the contour.

Fig. 4 shows a brief description of the process of extracting features where the squares indicate selected points. An input silhouette is represented by the CSS. Then, by selecting the threshold t, feature points are determined and transformed back into image space. Fig. 5 shows examples of the feature sets from a silhouette image and posture extracted manually. We can see that many of the feature points correspond very well. The exceptions are the feature points on the wrongly detected part (left part of the left image).

The standard CSS method involves finding zero crossing points of an input image (finding the peaks in the CSS image) and comparing them with those from a reference image. Thus, if there is a corrupted part, the peaks appeared in different locations. The heights of the peaks are used as weights of the

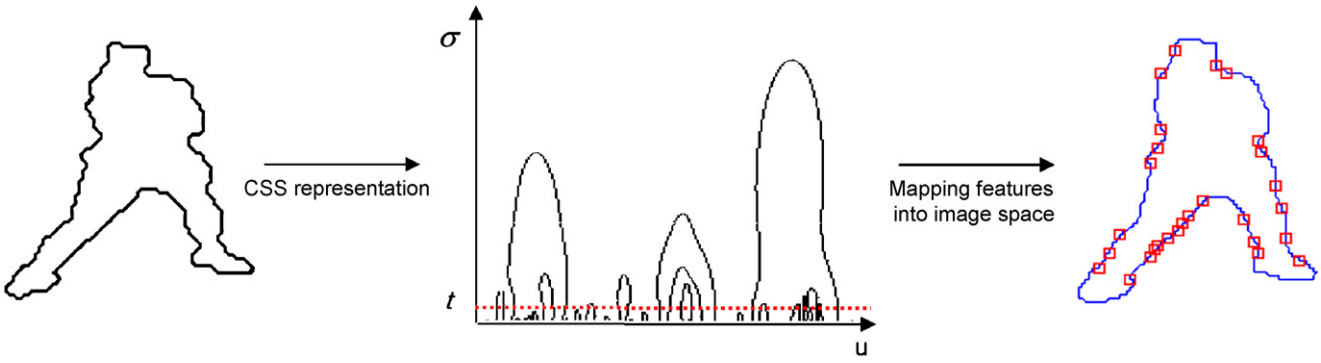

Fig. 4. The proposed feature extraction method.

points when two contours are compared. Therefore, if there is a significant shape corruption, such as Fig. 3(a), the matching is distorted. However, the proposed descriptor samples the silhouette and match the points with those from the reference silhouette. The proposed sampling method is much more efficient than many other sampling methods, because the sampling of the feature points is based on CSS which is itself robust to noise. The weights of the points are same. Therefore, although points on significantly corrupted part are also sampled, the matching is successful because most of the points correspond correctly. It has also low computational cost to match two shape images with small number of feature points.

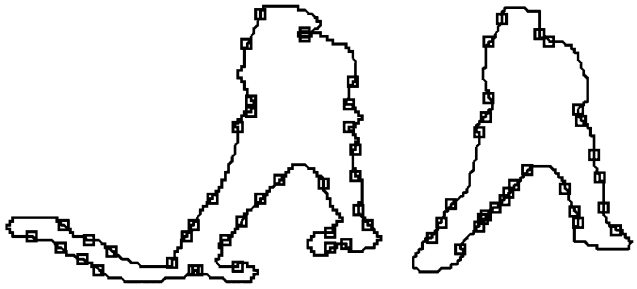

Fig. 5. The proposed features marked by squares.

3.3. Posture determination

A posture model database is generated from the silhouette images extracted manually from representative image sequences among several play shots. Hereafter, we refer to the silhouette images extracted from the test frames as *input images* and the postures in the database as *models*. To determine the input image's posture by comparing it with the models in the database, and measure the similarity, the transformation between them must be estimated. We use the RANSAC algorithm to find the transformation between the two sets of feature points, because of its power and simplicity [15]. We consider affine transformations in this paper, but extensions to other geometric transformations can be made easily. Apart from the translation parameters, the parameter values are assumed to be small on the grounds that there are a sufficient number of images in the database to match shape and size variations. We assume that a large enough proportion of the shape of the input image is preserved well enough to be matched to one of the models.

The affine transform between the model and the input image is computed in two steps: firstly, we calculate the translation, and secondly, we find the remaining parameters of the affine transform. The algorithm used for finding the transformation is shown in Table 1.

After estimating the transformation, B, the corresponding feature point can be found by selecting a pair of feature point for which the distance between the input image's feature point transformed by B and the feature point in the model is smallest. We define a distance metric D as the distance between the feature points in the input image and those in the model. However, the inverse function which maps the feature points from the model to their corresponding points from the input image, and the function, which maps the feature points from the input image to their corresponding points of the model, are not the same. Therefore, we define a symmetric distance metric, M, as follows:

$$D = D_{im} + D_{mi}, \tag{6}$$

$$D_{im} = \frac{1}{|I|} \sum_{\mathbf{x} \in I} \min_{\mathbf{y} \in M} \|\mathbf{B}\mathbf{x} - \mathbf{y}\|, \tag{7}$$

$$D_{mi} = \frac{1}{|M|} \sum_{\mathbf{y} \in M} \min_{\mathbf{x} \in I} \|\mathbf{x} - \mathbf{B}^{-1}\mathbf{y}\|, \tag{8}$$

where M and I are the sets of feature points of the model and the input image, $\mathbf{x}$ and $\mathbf{y}$ are homogeneous coordinates of the feature points, and $|I|$ and $|M|$ are the numbers of feature points in I and M, respectively. The posture of the input silhouette is determined by selecting the model that has the lowest M among the posture models in the database. Fig. 6 shows some examples of input images and their corresponding posture models (yellow contour) determined among the database models. In Fig. 6, the contours of the foregrounds are not extracted accurately because of the shadow and white lines of the court. Nevertheless, the posture (yellow contour) using the proposed features is achieved very successfully, even though the contours are not good enough to match using the standard CSS matching method.

Table 1
The procedure to find transformation between the features of input image and the model

1.	Pick one feature point from the feature set of the input image and another feature point from the feature set of the model.
2.	Estimate translation, **t**, by calculating the vector difference between the two feature points.
3.	Count the number of inliers between the input image and the model with the translation, **t**.
4.	Select the **t** which maximizes the number of inliers.
5.	Iterate the above steps a given number of times or until the number of inliers with the **t** selected above step has converged.
6.	Initialize the other parameters of the affine transformation: if we denote the affine matrix as [**A**|**t**], then initialize **A** as a unit matrix.
7.	Estimate the precise affine transform matrix, **B**, from the inliers using the Levenberg–Marquardt algorithm.

Fig. 6. Foreground (red) and determined posture model (yellow).

3.4. Gesture spotting

In this section, we will introduce the CDP algorithm and the proposed gesture spotting algorithm. In Section 3.4.1, we review the CDP method which is used as a baseline algorithm for the purpose of comparison to the proposed method.

The CDP is an extension of the DTW algorithm [10]. The traditional DTW is a method for finding the optimal matching between a reference model and input sequence based on dynamic programming. It is a matching algorithm for isolated data rather than a spotting algorithm. One of the successful extension methods of the DTW for the purpose of spotting is the CDP. It has been used in hand gesture spotting, speech segmentation and so on. Thus, we use the CDP as a reference method for comparison.

3.4.1. Continuous dynamic programming

Let $f(t)$ and $Z(\tau)$ be variables used to represent inputs which are functions of time t in the input image sequence space and time τ in the reference sequence space, respectively.

Thus, t is unbounded and $\tau \in \{1 \dots T\}$, where T is the length of the reference pattern. The local distance is defined by $d(t,\tau) = |f(t) - Z(\tau)|$ and the minimum accumulated value of local distances $P(t,\tau)$ is initialized by $P(-1,\tau) = P(0,\tau) = \infty$. Then, iteration $(t = 1, 2, \dots)$ is:

for $\tau = 1$

$$P(t,1) = 3 \cdot d(t,1), \tag{9}$$

for $\tau = 2$

$$P(t,2) = \min \begin{cases} P(t-2,1) + 2 \cdot d(t-1,2) + d(t,2), \\ P(t-1,1) + 3 \cdot d(t,2), \\ P(t,1) + 3 \cdot d(t,2), \end{cases} \tag{10}$$

for $\tau \geqslant 3$

$$P(t,3) = \min \begin{cases} P(t-2,\tau-1) + 2 \cdot d(t-1,\tau) + d(t,\tau), \\ P(t-1,\tau-1) + 3 \cdot d(t,\tau), \\ P(t-1,\tau-2) + 3 \cdot d(t,\tau-1) \\ \quad + 3 \cdot d(t,\tau), \end{cases} \tag{11}$$

A section of an input sequence is 'spotted' if the value of $A(t)$ gives a local minimum below a threshold value, where $A(t)$ is given by

$$A(t) = \frac{1}{3 \cdot T} P(t,T). \tag{12}$$

How different a spotted sequence is from a reference sequence is dependent on the threshold value.

3.4.2. Proposed sequence matching algorithm

We propose a new method of sequence matching, which is simple, works with a small amount of training data, and provides a probabilistic output. The need for a small amount of training data is clearly important, as the need for large training sets for the neural networks or hidden Markov models can make these methods unattractive. Also, the typical template matching methods such as DTW, or CDP do not provide probabilistic output measurement. Therefore, these methods are sensitive to variations in the threshold.

The proposed algorithm represents a gesture as a curve in a model index versus input image time sequence space. Let $G = \{g_1, g_2, \dots, g_N\}$ represent a gesture which is an ordered set by time index and the index of the model is also ordered, i.e. $G = \{1, 2, \dots, N\}$. The kth element g_k represents an index of a model or a cluster that has similar shapes as measured by the proposed shape descriptor. The indexes point to their corresponding models. Let the model of interest be g_k where $1 \leqslant n \leqslant N$.

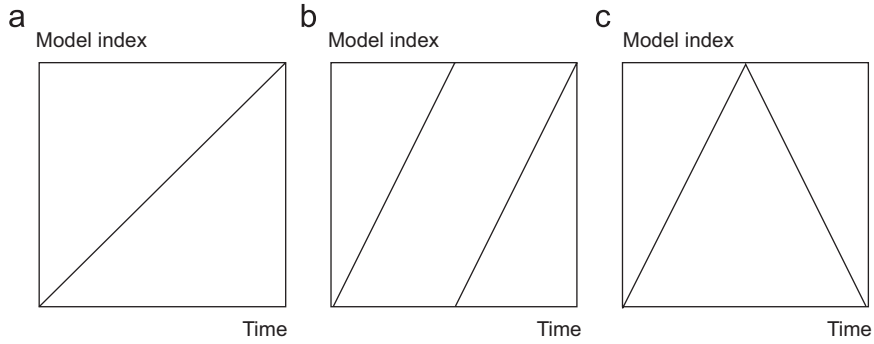

Fig. 7. Examples of gesture curves.

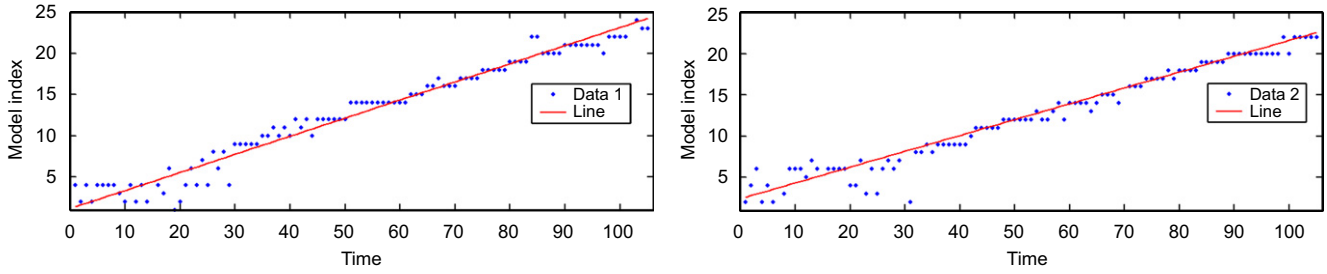

Fig. 8. Examples of line fitting for training a gesture.

Given a curve, C, the re-aligned index, G', can be represented by

$$G' = \{h_1, h_2, \ldots, h_n, (h_{n+1}), (h_{n+2}), \ldots\}, \tag{13}$$

where

$$h_i = C(g_i), \quad i \leqslant n.$$

Fig. 7 shows simple examples of some typical gesture curves. Fig. 7(a) shows a simple gesture such as *the raising up of a right arm* and (b) and (c) show a repetitive gesture, such as *writing 'w'*, and a mirroring gesture, such as *the raising up and down of a right arm*, respectively.

If the line equation, $C(g_k) = ag_k + b$, is used (in the case of Fig. 7(a)), then G' will be aligned linearly and there are only two parameters (a, b) to be trained. The variances (σ_a, σ_b) and means (μ_a, μ_b) of a and b are determined from some training sequences and are used for estimating the likelihood of the input sequence. Given the size, l, of the interval, let $v(s)$ be an interval $v(s) = [s, s+l]$ in the input sequence and s be a starting frame of a sliding window. For each interval, $v(s)$, $a'(s)$ and $b'(s)$, such that $C'(g_j) = a'(s)g_j + b'(s)$ where $g_j \in v(s)$, are calculated using a data fitting algorithm. Then the likelihoods $L_a(s)$ and $L_b(s)$ are calculated as follows:

$$L_a(a'(s)) = \frac{1}{\sigma_a\sqrt{2\pi}} e^{-(a'(s)-\mu_a)^2/2\sigma_a^2}, \tag{14}$$

$$L_b(b'(s)) = \frac{1}{\sigma_b\sqrt{2\pi}} e^{-(b'(s)-\mu_b)^2/2\sigma_b^2}. \tag{15}$$

Then, we define likelihood L of the interval for the trained curve as follows:

$$L(v(s)) = L_a(a'(s)) \times L_b(b'(s)). \tag{16}$$

Spotting is done by using the likelihood, L. By finding the maximum peak value of $L(v(s))$ for the interval, where $L(v(s)) > L_{min\_threshold}$, the gestures can be spotted. The threshold value, $L_{min\_threshold}$, can be roughly estimated, because the performance is not sensitive to it. In our experiments on serve detection in a tennis video, the difference of the peaks in the serve and the non-serve gesture sequences was nearly 1. The maximum likelihood for the non-server gestures among the correctly spotted sequences was 0.75×10^{-8}. Various speeds of gestures can be absorbed by the variance of b determined from the training sequences.

For a robust estimation of C', we need to choose only the inliers of the estimated line parameters, because there are some mis-determined postures in the posture matching, although most are determined correctly. Basically, only those posture indexes that are of interest in estimating C' are used, while those indexes which are of no interest are ignored. The line equation is estimated using a least squared method. Other robust methods of rejecting outliers can be easily adopted. If the number of interesting postures in the interval, $v(s)$, is smaller than a given threshold, it means that the sequence in the interval has little information to be matched to a gesture, in which case we ignore the interval. Fig. 8 shows examples of the lines that are fitted by the least square fitting method, to train a and b using two sets of sequence data.

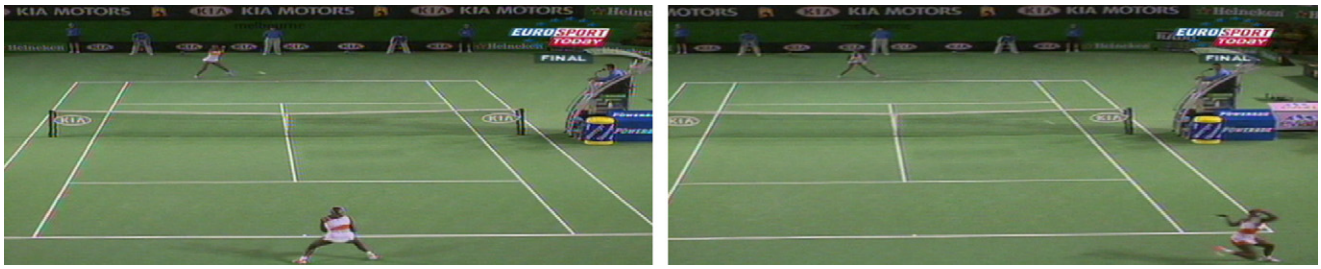

Fig. 9. Examples of input frames.

Fig. 10. Mosaic image.

4. Experimental results

In the experiments, the proposed method is used to detect serves in standard off-air tennis videos. Serve gesture spotting and recognition play important roles in understanding and annotating tennis videos. In contextual analysis of tennis video, there are several cues to annotate the play, such as trajectories of ball and players, court lines, strokes and serves. The serve is the most useful cue to indicate the start of a play shot. The ball tracking and other annotation modules are launched at the start of the annotation, as well as segmentation of the play shots. In this paper, we focused solely on the detection of the serve gesture, among the other cues needed in annotating a tennis video.

4.1. Environment

An off-air interlaced sequence from the 2003 Australian open tennis tournament video was used to evaluate the performance of the proposed posture matching and gesture spotting method. Our target player is the near/bottom player. Initialization of the player's location was conducted using a background subtraction method and by analyzing blobs. To get the posture model database, a single play shot was chosen from the whole collection, and the player's postures are extracted manually from foreground images. Then, the serve gesture was trained using three play shots of this same.

4.2. Foreground separation

Mosaic images are created for each play shot. Figs. 9 and 10 represent the input frames and the mosaic image created from the input frames, respectively. Fig. 11 represents the foreground images of Fig. 9 by subtracting the mosaic image (which is the background image) from the input frame transformed into a mosaic coordinate. The mean width and height of the player regions are about 53 and 97 pixels, respectively.

4.3. Posture determination

The postures are described by the proposed posture descriptor and determined by comparison with the models in the database. The feature set is extracted from the input image and the distance to the posture models in the database is measured. Fig. 12(a) shows a good example of posture determination in the case where the player's silhouette is corrupted significantly. Although the contours of the foregrounds are not extracted accurately, they are correctly matched (yellow contour). Other good examples are also shown in Fig. 6. Fig. 12(b–d) shows some examples of incorrectly matched postures. Although some of the postures are not correctly determined, the gesture spotting is sufficiently robust to determine most of the serve gesture postures correctly.

Fig. 13 represents the normalized similarity values for two serve gestures of two players. The higher the value (red), the more similar it is to the model in the database. The model index corresponds to the model of the serve gesture in the database. For the model index, the models and postures should be matched well because both sequences are serve gestures; this is seen in the figure. Significant peaks in the diagonal are observed, indicating that the proposed posture descriptor is reliable.

Fig. 14 represents postures that are the serve gestures spotted from the test video sequence, using proposed method. The yellow contours represent the determined posture models. When the player is different form the one used for training, most of the postures are also determined correctly, although there are some errors caused by the different serve postures and wrongly separated foreground.

4.4. Serve gesture spotting

We tested the proposed method using 63 sequences, some of which include a serve gesture, while some do not. The sequences are also partitioned by the players' identity ('A' and 'B') and the position of the player on the court ('left' and

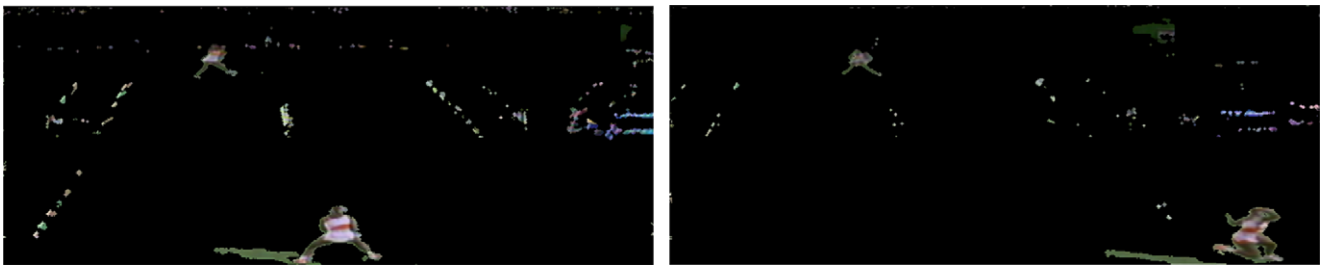

Fig. 11. Foreground images of Fig. 9.

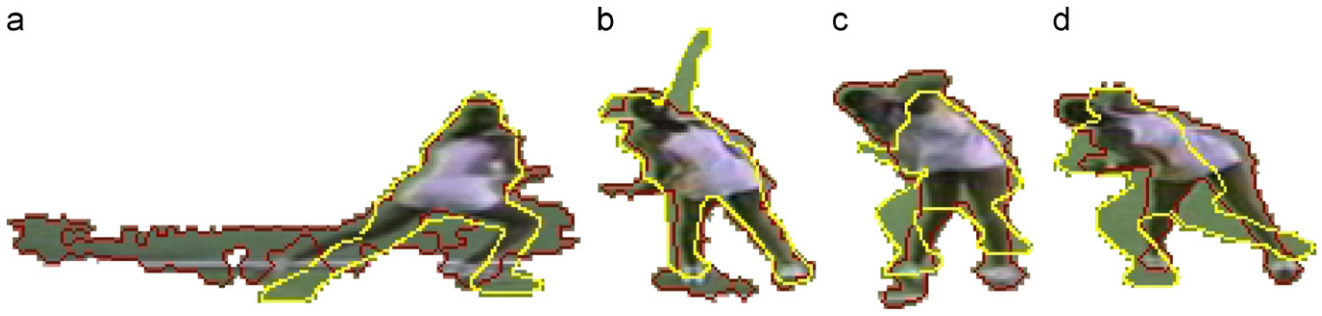

Fig. 12. Foreground(red) and determined posture model(yellow)

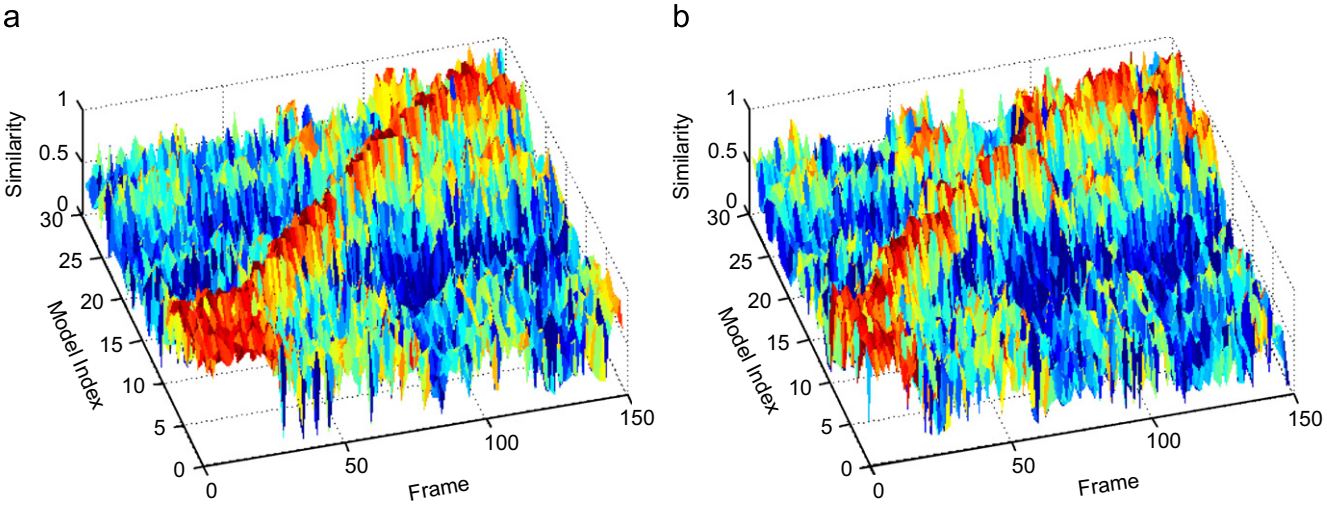

Fig. 13. Normalized similarity matrix for two serve gestures (The higher the red, the more similar it is to the model in the database.)

'right'). Fig. 15 represents postures plotted onto the *frame-model index*. For training, we use a single shot, in which one of the players is serving on the right-hand side of the court.

Fig. 15 represents postures plotted onto the *frame-model index* space, where P represents the index of the posture in the training database. The indexes below the red-dotted line are postures of serve gestures. Therefore, the posture index of interest ranges from 0 to 22, i.e. $N = 22$ in Eq. (13). The indexes above the red-dotted line are outside the interesting postures, and are referred to as garbage postures. These garbage postures are treated as outliers and do not contribute to the calculation of the line parameters for the serve gestures spotting.

Fig. 16 shows the likelihoods of serve gesture for the input sequences corresponding to Fig. 15, where L represents the likelihood. We can see significant peaks in Fig. 16 when there are serve gestures. Figs. 15(a) and 16(a) show results of a short play. The play ended shortly due to fault while a player was

Fig. 14. Foreground and determined postures among the serve gestures for two players.

serving. In Figs. 15(b) and 16(b), the player was bouncing the ball for a while before making a serve, therefore serve gestures are spotted after 16th frame. In Figs. 15(c) and 16(c), there were two serve sequences (starting at the first and 548th frames) in one play shot. The player served twice because the first serve was a fault in the play shot. Figs. 15(d) and 16(d) show the result on one of the play shots used for training.

Table 2 shows the results of the serve gesture spotting. In this table, true positive (TP), true negative (TN), false positive (FP) and false negative (FN) values indicate the cases where a serve gesture is detected in the correct location, where no serve gesture is detected when there is no serve, where a serve gesture is detected when there is no serve and where no serve gesture is detected when there is a serve, respectively. The proposed method generates 90.5% correct results (TP+TN) and 7.8% incorrect results (FP+FN), while the corresponding results for the baseline algorithm, CDP, are 61.9% and 36.7%, respectively.

Since we trained the data on just one serve sequence in which player 'A' is serving on the right side of the court, the results in the 'left' columns are lower than those in the 'right' columns. In fact, the silhouette of a gesture when a player serves on the right side of the court is different from the that of a gesture when a player serves on the left side of the court. The spotting result shows that our method is substantially better than CDP.

Since the evaluation afforded by Table 2 could be vague in this kind of spotting application, we introduce another evaluation method which is usually used to evaluate the detection algorithms. Eqs. (17) and (18) represent the evaluation measurements of the precision and recall rates, respectively. Table 3 represents the gesture spotting results evaluated by the precision and recall rates. According to the results in this table, the proposed method has a much higher value for both measurements.

$$Precision = \frac{\text{Number of correctly spotted sequences}}{\text{Number of spotted sequences as serve}} \quad (17)$$

$$Recall = \frac{\text{Number of correctly spotted sequences}}{\text{Number of serve sequences as serve}}. \quad (18)$$

4.5. Analysis

We proposed a combination of two processes to detect serve gesture; determination of a posture and spotting a sequence. The method can be directly applied to other types of spotting applications. More variation of the speed, starting point and

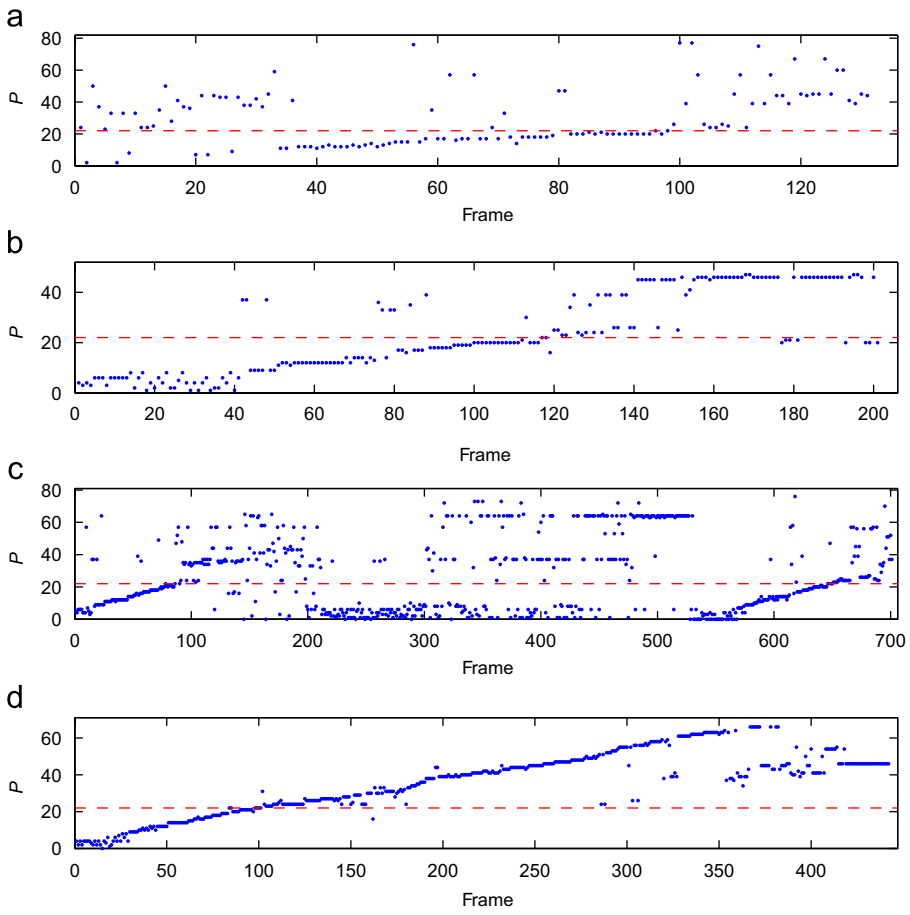

Fig. 15. Determined postures on posture-frame domain space for four play shots including whole tennis gestures, such as running and stroke gestures as well as server gesture. P's indicate postures.

ending point be accommodated by extending the training stage (i.e. σ_a and σ_b, respectively, in Eqs. (14) and (15). As we performed the experiment on the player 'B', we also got good result, although it was little bit lower than that of the player 'A'. Therefore, we can say the variability of players can be reasonably well accommodated in our method. In terms of posture determination, we can apply the proposed method to other applications which can generate a silhouette feature. From the experimental results, the combination of processes gave reasonable accuracy in the tennis video.

Since we used a silhouette-based feature to determine a posture in this paper, it may not directly applicable to other kinds of tennis gestures, such as forehand strokes and backhand strokes. These gestures have many variations of posture, and do not have distinctive postures that can characterize such a gesture. It means that these gestures cannot be analyzed using only silhouettes, and other additional information, such as velocity and acceleration information, should be considered. These considerations are beyond the scope of this paper.

We conducted an additional experiment using another tennis video to see how the accuracy would vary. Twenty play shots from the 2006 WTA Tour video are used. The frames are normalized so that the players have the similar height size with the model in the database we used in the previous experiment. The frames are normalized to facilitate the computation of the affine matrix **B** in Eqs. (7) and (8). Fig. 17 presents some examples of determined postures. The foreground is separated clearer than the previous videos because there is little camera movement in each play shot. The proposed method achieved 65.0% correct and 10.5% incorrect results for the new video. The spotting results are lower than the previous one because the posture determination is not accurate as the previous one due to the different video environments and the players' different characteristics; such as, frame rates, resolutions, camera geometries, types of gesture, height, the lengths of the legs and so on. Fig. 17(d) shows an example of different body characteristic between the player used in training and the player in the new video, although the posture determined correctly. The player in the new video is taller and has longer legs than the player in the database.

5. Conclusion and further research

We presented a robust posture descriptor based on the CSS feature and a gesture spotting method that represents a gesture by a line/curve. The proposed posture descriptor is robust to

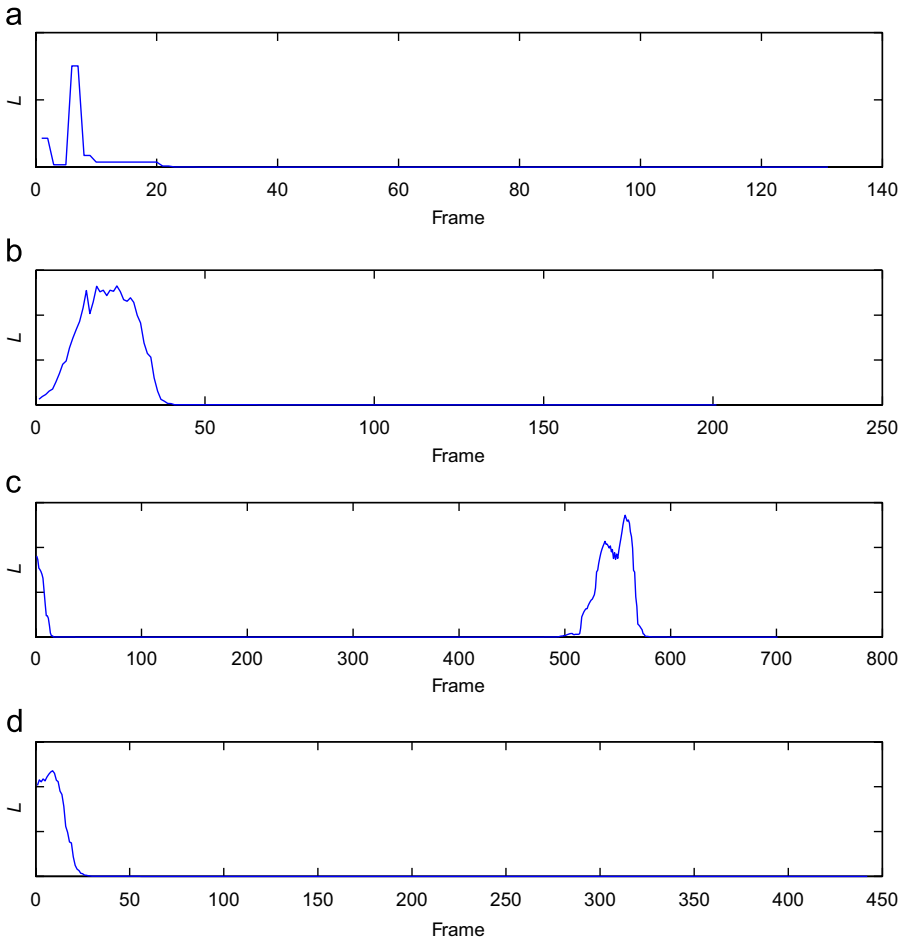

Fig. 16. Likelihood for the play shots corresponding to Fig. 15.

Table 2
Gesture spotting results using continuous dynamic programming and the proposed method

		Player A		Player B		Total
		Left	Right	Left	Right	
CDP	TP	6/12	9/13	5/9	8/11	61.9% correct
	TN	11/18				
	FP	5/12	4/13	4/9	3/11	36.7% incorrect
	FN	6/12	4/13	4/9	3/11	
Proposed method	TP	9/12	12/13	8/9	10/11	90.5% correct
	TN	18/18				
	FP	0/12	0/13	1/9	0/11	7.8% incorrect
	FN	3/12	1/13	1/9	1/11	

Table 3
Gesture spotting results evaluated by precision and recall rates

Method	Evaluation
CDP	Precision rate = 63.6% Recall rate = 62.2%
Proposed Method	Precision rate = 97.5% Recall rate = 86.7%

noise and significant corruption of the player's shape, which occasionally occur in low-resolution sports video. The proposed gesture spotting method is also robust to noise which is caused by mismatched posture. It also needs only a small number of training sets, which is a very important advantage in practical applications.

From our experiments, the proposed spotting method achieved 90.5% correct results and 7.8% incorrect results (97.5% precision and 86.7% recall) while the CDP achieved correct and incorrect results of only 61.9% and 36.7%, respectively (63.6% precision and 62.2% recall). The proposed spotting method is robust to noise in the sequence data, and its computational cost is small enough to be calculated in real time. If the time for posture matching were to be reduced, the whole system could operate in real time with good performance.

In the future, we intend to improve the posture matching speed and extend the serve gesture spotting method to cope with gestures used in daily life, which may contain repeated and complicated motion. There are many potential applications of posture matching and gesture spotting, including sports video annotation, surveillance systems, human–robot interactions, etc.

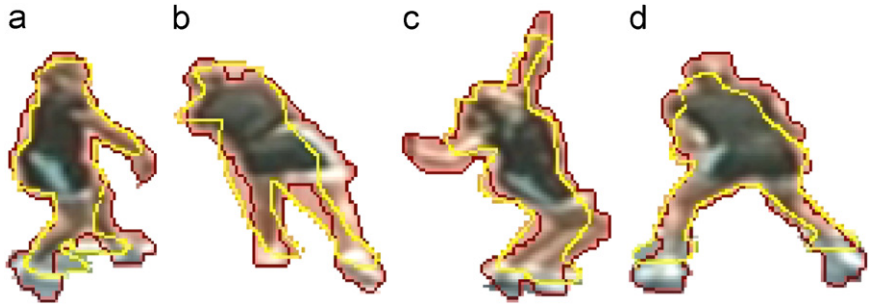

Fig. 17. Foreground(red) and determined posture model(yellow) in another video sequence.

Acknowledgment

This research was supported by the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Commerce, Industry and Energy of Korea, the IST-507752 MUSCLE Network of Excellence, and the IST FP6-045547 VIDI-Video project.

References

[1] W.J. Christmas, A. Kostin, F. Yan, I. Kolonias, J. Kittler, A system for the automatic annotation of tennis matches, Fourth International Workshop on Content based Multimedia Indexing, Riga, June 2005.

[2] F. Yan, W. Christmas, J. Kittler, A tennis ball tracking algorithm for automatic annotation of tennis match, in: Proceedings of BMVC, Oxford, UK, September 2005, pp. 619–628.

[3] J.R. Wang, N. Parameswaran, Survey of sports video analysis research issues and applications, In: Proceedings of Pan-Sydney Area Workshop on Visual Information Processing, vol. 36, Sydney, Australia, December 2004, pp. 87–90.

[4] J. Sullivan, S. Carlsson, Recognising and tracking human action, Proceedings of European Conference on Computer Vision (May 2002) 629–644.

[5] S. Kopf, T. Haenselmann, W. Effelsberg, Shape-base posture and gesture recognition in videos, Electronic Imaging, vol. 5682, San José, CA, January 2005, pp. 114–124.

[6] A.-Y. Park, S.-W. Lee, Gesture spotting in continuous whole body action sequences using discrete hidden Markov models, gesture in human–computer interaction and simulation, Lect. Notes Artif. Intell. 3881 (May 2006) 100–111.

[7] M.-C. Roh, H.-K. Shin, S.-W. Lee, S.-W. Lee, Volume motion template for view invariant gesture recognition, in: Proceedings of the 18th IAPR/IEEE International Conference on Pattern Recognition, Hong Kong, China, vol. 2, August 2006, pp. 1229–1232.

[8] J. Ben-Arie, Z. Wang, P. Pandit, S. Rajaram, Human activity recognition using multidimensional indexing, IEEE Trans. Pattern Anal. Mach. Intell. 24 (8) (2002) 1091–1104.

[9] H.-K. Lee, J.H. Kim, An HMM based threshold model approach for gesture recognition, IEEE Trans. Pattern Anal. Mach. Intell. 21 (10) (1999) 961–973.

[10] L. Rabiner, B.-H. Juang, Fundamentals of Speech Recognition, Prentice-Hall, England Chiffs, NJ, 1993.

[11] R. Oka, Spotting method for classification of real world data, Comput. J. 41 (8) (1998) 559–565.

[12] J. Alon, V. Athitsos, S. Sclaroff, Accurate and efficient gesture spotting via pruning and subgesture reasoning, In: Proceedings of IEEE Workshop on Human–Computer Interaction, Beijing, China, October (2005) pp. 189–198.

[13] M. Irani, P. Anandan, S. Hsu, Mosaic based representations of video sequences and their applications, In: Proceedings of International Conference Computer Vision, MA, USA, June 1995, pp. 605–611.

[14] F. Mokhtarian, M. Bober, Curvature Scale Space Representation: Theory, Applications & MPEG-7 Standardisation, Kluwer Academic, Dordrecht, 2003.

[15] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun ACM 24 (1981) 381–395.

About the Author—MYUNG-CHEOL ROH received his B.S. degree in Computer Engineering from Kangwon University, Chun-Choen, Korea, in 2001, and his M.S. degrees in Computer Science and Engineering from Korea University, Seoul, Korea, in 2003. He is currently a Ph.D. student in the department of Computer Science and Engineering in Korea University. He won the best paper award of the 25th annual paper competition which is supervised by the Korea Information Science Society and is sponsored by Microsoft in 2006. He worked at the Center for Vision, Speech and Signal Processing in the University of Surrey as a collaborate researcher for 1 year since 2004. His present research interests include object tracking, text extraction, face and gesture recognition, robot vision and the pattern recognition related fields.

About the Author—WILLIAM J. CHRISTMAS received the Ph.D. degree from the University of Surrey, Surrey, UK, while working on the use of probabilistic methods for matching geometric features. He holds a University Fellowship in Technology Transfer at the Centre for Vision, Speech, and Signal Processing, University of Surrey. After studying engineering science at the University of Oxford, Oxford, UK, he spent some years with the British Broadcasting Corporation as a Research Engineer, working on a wide range of projects related to broadcast engineering. He then moved to BP Research International as a Senior Research Engineer, working on research topics that included hardware aspects of parallel processing, real-time image processing, and computer vision. His other interests have included region-based video coding and the integration of machine vision algorithms to create complete applications. Currently, he is working on projects concerned with automated, content-based annotation of video and multimedia material.

About the Author—JOSEF KITTLER received the B.A., Ph.D., and D.Sc. degrees from the University of Cambridge in 1971, 1974, and 1991, respectively. He heads the Centre for Vision, Speech, and Signal Processing at the School of Electronics and Physical Sciences, University of Surrey. He teaches and conducts research in the subject area of machine intelligence, with a focus on biometrics, video and image database retrieval, automatic inspection, medical data analysis, and cognitive vision. He has published a Prentice Hall textbook, Pattern Recognition: A Statistical Approach, and several edited volumes, as well as more than 500 scientific papers, including in excess of 150 journal papers. He serves on the editorial board of several scientific journals in pattern recognition and computer vision. He has consulted for many companies and is one of the founders of OmniPerception Ltd. He chairs the OmniPerception Advisory Group.

About the Author—SEONG-WHAN LEE received his B.S. degree in Computer Science and Statistics from Seoul National University, Seoul, Korea, in 1984, and his M.S. and Ph.D. degrees in computer science from KAIST in 1986 and 1989, respectively. From February 1989 to February 1995, he was an assistant professor in the Department of Computer Science at Chungbuk National University, Cheongju, Korea. In March 1995, he joined the faculty of the Department of Computer Science and Engineering at Korea University, Seoul, Korea, as an associate professor, and he is now a full professor. He was the winner of the Annual Best Paper Award of the Korea Information Science Society in 1986. He obtained the First Outstanding Young Researcher Award at the Second International Conference on Document Analysis and Recognition in 1993, and the First Distinguished Research Professor Award from Chungbuk National University in 1994. He also obtained the Outstanding Research Award from the Korea Information Science Society in 1996. He has been the founding co-Editor-in-chief of the International Journal on Document Analysis and Recognition and the associate editor of the Pattern Recognition Journal, the International Journal of Pattern Recognition and Artificial Intelligence, and the International Journal of Computer Processing of Oriental Languages since 1997. He was the Program co-chair of the Sixth International Workshop on Frontiers in Handwriting Recognition, the Second International Conference on Multimodal Interface, the 17th International Conference on the Computer Processing of Oriental Languages, the Fifth International Conference on Document Analysis and Recognition, and the Seventh International Conference on Neural Information Processing. He was the workshop co-chair of the Third International Workshop on Document Analysis Systems and the First IEEE International Workshop on Biologically Motivated Computer Vision. He served on the program committees of several well-known international conferences. He is a fellow of IAPR, a senior member of the IEEE Computer Society and a life member of the Korea Information Science Society. His research interests include pattern recognition, computer vision and neural networks. He has published more than 200 publications in these areas in international journals and conference proceedings, and has authored five books.