Robust Player Gesture Spotting and Recognition in Low-Resolution Sports Video

Myung-Cheol Roh[1], Bill Christmas[2], Joseph Kittler[2] and Seong-Whan Lee[1]*

[1] Center for Artificial Vision Research, Korea Univ., Seoul, Korea
{mcroh, swlee}@image.korea.ac.kr
[2] Center for Vision, Speech, and Signal Processing, Univ. of Surrey, Guildford, UK
{w.christmas, j.kittler}@surrey.ac.uk

Abstract. The determination of the player's gestures and actions in sports video is a key task in automating the analysis of the video material at a high level. In many sports views, the camera covers a large part of the sports arena, so that the resolution of player's region is low. This makes the determination of the player's gestures and actions a challenging task, especially if there is large camera motion. To overcome these problems, we propose a method based on curvature scale space templates of the player's silhouette. The use of curvature scale space makes the method robust to noise and our method is robust to significant shape corruption of a part of player's silhouette. We also propose a new recognition method which is robust to noisy sequences of data and needs only a small amount of training data.

1 Introduction

The development of high-speed digital cameras and video processing technology has attracted people's attention to automated video analysis such as surveillance video analysis, video retrieval, sports video analysis. Specifically, applying this technology to sports video has many potential applications: automatic summary of play, highlight extraction, winning pattern analysis, adding virtual advertizement, etc. There are interesting works on ball tracking, player tracking and stroke detection for tennis, baseball, soccer, American football, etc[2, 13, 14].

Although there has been much discussion in the literature on automatic sports video annotation and gesture recognition in restricted environment, there is little on player's gesture detection/recognition in standard off-air video, due to low resolution of the player's region, fast motion of the player and camera motion[11, 14].

In sports video, the player's region often has low resolution because the audience wants to watch a wide 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. Camera motion can also make tracking players and extracting players' silhouettes hard, and low resolution makes matching

* To whom all correspondence should be addressed.

player's posture to trained models unstable. Then, because of unstable matched postures, recognition and detection of gestures can also be hard.

J. Sullivan et al. proposed a method for detecting tennis players' strokes, based on qualitative similarity that computes point to point correspondence between shapes by combinatorial geometric hashing. 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[11]. Although they presented interesting results from their video sequence, the method has some shortcomings. The outline of player will not be extracted accurately when the resolution is low. Often we can see only the player's back while playing, so that we cannot use information of the player's arm because of self occlusion. S. Kopf et al. proposed shape-based posture and gesture recognition using a new curvature scale space (CSS) method in a video sequence which was recorded by pan/tilt/zoom camera[5]. Their new CSS representation can describes convex segment of shape as well as concave. However, their test sequence is of good quality, with good resolution of the player. Also they recognized postures, rather than gesture which is a set of postures. To date, there is much literature in human gesture recognition using 3D, but these methods are difficult to apply to low-resolution video which shows mainly player's back posture; also they are not computationally efficient[9].

There are many sequence recognition and matching methods considering time information, and they have given interesting result in particular environments[3, 6]. But, in special cases such as broadcast sports video, we may not have enough 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, gives a similarity measure[10]. But it needs high computational cost and does not give probability measurement. An extension of DTW, Continuous Dynamic Programming (CDP) was proposed by Oka[8], which is our baseline algorithm for comparing with our proposed gesture matching/spotting algorithm.

J. Alon et al. proposed a gesture spotting CDP algorithm via pruning and sub-gesture reasoning . Their method shows an 18% increase in recognition accuracy over the CDP algorithm for video clips of two users gesturing the ten digits 0-9 in an office environment[1].

In this paper, we suggest a new type of feature to represent the player silhouette, together with a novel gesture spotting method. We found the combination to be efficient and robust to the problems of noise and low resolution we encountered using standard off-air video.

2 Sports Player Posture Matching and Gesture Spotting

Our system consists of four parts: foreground separation, silhouette feature extraction, player posture matching and gesture detection. Fig. 1 shows a diagram of our gesture spotting system. Foreground separation is to separate foreground objects from original frames using mosaicing. As a result of foreground sepa-

ration, we get players' silhouette, ball silhouette and noise blobs roughly, and can track player's position using particle filter. Silhouette matching is to match player's silhouette which also includes wrong separated area, to trained silhouettes in database. Gesture detection is done using history of matched silhouettes and database, which is a function of time domain.

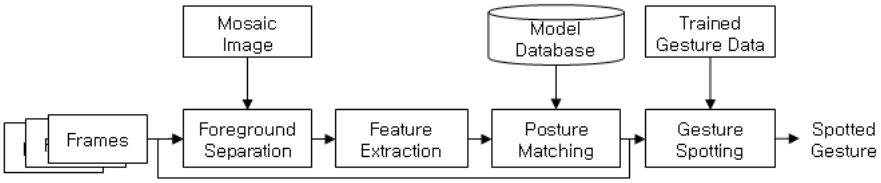


Fig. 1. Player's gesture annotation system

2.1 Foreground Separation

Background generation We assume a pan/tilt/zoom camera in this method, which is the situation in many sports events. It also makes feasible the use of a mosaicking technique: each frame is projected into a single coordinate system, and a mosaic is created by median filtering of the pixels, creating a background image.

Foreground separation By warping the mosaic to match the current frame, the foreground image is extracted simply by taking the difference between the frame and mosaic. Fig. 2 shows input frames, mosaic image and foreground images.

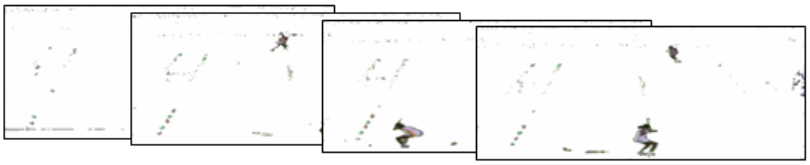
2.2 Silhouette Feature Extraction

We extract the silhouettes from the foreground image of the previous stage by identifying the two largest blobs. The silhouettes are used for matching a posture to posture models in a database, using the following steps.

Curvature scale space(CSS) [7] CSS is a well-established technique for shape representation used in image retrieval, and is one of the descriptors used in the MPEG-7 standard. We outline the method here, paraphrasing the description in [7]. 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 evolving by the

a) Input video frames

b) Mosaic image



c) Foreground image

Fig. 2. Mosaic image and foreground images separated from input frames

expanding Gaussian smoothing function. Let the closed planar curve r be represented by the normalized arc length parameter u:

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

Then the evolved curve is represented by Γ_σ :

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

where

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

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

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

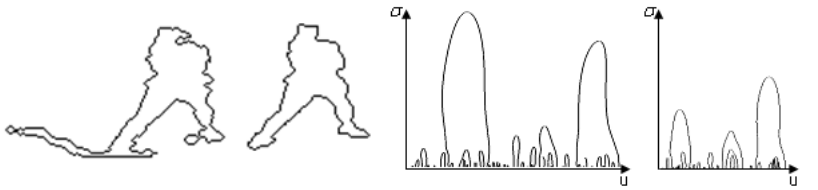
$$\psi_u(u,\sigma) = y(u) \otimes g_u(u,\sigma)$$

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

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

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

The CSS image representation is robust to a similarity transformation and (to a lesser extent) an affine transformation, so significant peaks in the CSS shape representation are considered as suitable features for similarity-based retrieval. But the drawback is the zero crossing points of CSS are not reliable features, if part of the shape is corrupted significantly. Fig. 3 (a) and (b) show examples of foreground silhouettes which are corrupted by noise blobs due to low-quality video sequence and a posture model to be matched, respectively. Fig. 3 (c) and (d) shows CSS images of foreground silhouette(a) and posture model(b). These two CSS images are not likely to be considered the same.



a) A Silhouette image b) A silhouette image c) CSS image of a) d) CSS image of b)

Fig. 3. Examples of foreground silhouette and CSS represented images

Proposed feature extraction We propose a new feature which is based on CSS. Given threshold t, new feature set F of a curve is defined by:

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

where

$$I_c^t = \{(u,\sigma)|\kappa(u,\sigma) = 0, u \in [0,1], \sigma = t\}$$

Fig. 4 represents the processing of extracting features. Fig. 5 shows an example of the new feature set from silhouette images. In contrast to the CSS and the sampling method which is sampling some points on the contour with fixed number[12], the proposed feature is more robust to local noise and significant shape corruption of a part of silhouette and has low computational cost to match two shape images.

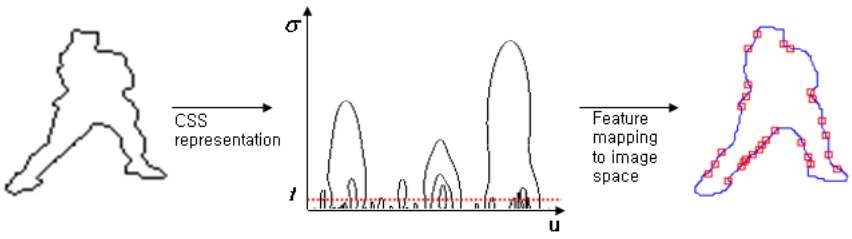


Fig. 4. Proposed feature extracting

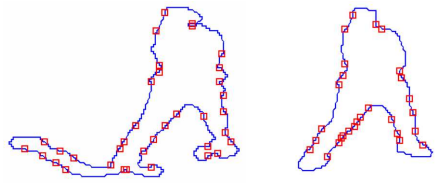


Fig. 5. The proposed features based on CSS. The squares indicate feature locations.

2.3 Posture Matching

A posture model database using the proposed feature is constructed from a set of silhouette images of which are extracted manually from representative image sequences. We call silhouette images extracted from test frames *input images* and the silhouettes in the database *models* from now on. To match input image with models in the database and measure the difference, the transformation between them must be estimated. RANSAC algorithm is used for finding transformation of two sets of feature points because of its powerfulness and simplicity[4]. We considered affine transformations in this paper, but extensions to other geometric transformations can be made easily. Apart from the translation parameters, parameter values in the affine transform matrix 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 proportion of the shape of the input image is preserved well enough to be matched to one of the models.

The affine transform between model and input image is computed in two steps: firstly the translation is calculated, and secondly the remaining parameters of the affine transform are estimated. The algorithm for finding transformation is defined as follows:

1. Pick one feature point from the feature set of the input image and other feature point from the feature set of the model.
2. Calculate the translation $\mathbf{t}$.
3. Count the number of inliers between the input image and the model with the $\mathbf{t}$.
4. Repeat above steps k times and find $\mathbf{t}$ which has the biggest number of inliers.

5. Initialize the other parameters of the affine transformation : if we denote the affine matrix as $(\mathbf{A}|\mathbf{t})$, then initialise $\mathbf{A}$ as a unit matrix.
6. Find the precise affine transform matrix of the inliers using the Levenberg-Marquardt algorithm.

After finding the transformation $\mathbf{A}$, corresponding feature points in the input image can be found by selecting feature points of the model, transformed by the transform matrix. Let F_{mi} be a function mapping feature points from the model to their corresponding points from the input image, and let F_{im} be the converse. Then, we define D_{mi} as the mean distance between feature points from the model and their corresponding points from the input image, and similarly for D_{im}.

$M = D_{im} + D_{mi}$ is used as a measurement, and a matched model which has lowest M is selected. Fig. 7 shows some examples of input images and models matched to input images.

2.4 Gesture Spotting

We will introduce our Sequence Matching/spotting algorithm and the continuous dynamic programming(CDP) algorithm which is the baseline algorithm to which we compare our proposed algorithm.

Continuous dynamic programming [8] CDP is an extension of the Dynamic Time Warping (DTW) algorithm. Let $f(t)$ and $Z(\tau)$ be variables 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 a 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 $\tau = 1$

$$P(t,1) = 3 \cdot d(t,1) \tag{6}$$

for $\tau = 2$

$$P(t,2) = \min \begin{cases} P(t-2,1) + 2 \cdot d(t-1,2) + d(t,2) \\ P(t-1,1) + 3 \cdot d(t,2) \\ P(t,1) + 3 \cdot d(t,2) \end{cases} \tag{7}$$

for $\tau \leq 2$

$$P(t,2) = \min \begin{cases} P(t-2,\tau-1) + 2 \cdot d(t-1,\tau) + d(t,\tau) \\ P(t-1,\tau-1) + 3 \cdot d(t,\tau) \\ P(t-1,\tau-2) + 3 \cdot d(t,\tau-1) + 3 \cdot d(t,\tau) \end{cases} \tag{8}$$

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:

$$A(t) = \frac{1}{3 \cdot T} P(t,T) \tag{9}$$

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

Proposed Sequence Matching Algorithm We propose a new method of sequence matching which is simple and works with a small amount of training data. The need for a small amount of training data is clearly important, as the large training sets typically needed for the commonly used Neural Networks and Hidden Markov Models can be hard to come by. In this paper, we represent a gesture (which is a sequence of postures) as a curve in a 2D Cartesian space of reference time τ versus input image time sequence t. Let $D = \{g_1, g_2, \ldots, g_N\}$ represent a gesture ordered by time index. Thus the kth element g_k represents a member of a cluster of models which have same posture. Let the object $n \leq N$ is the model index of interest (the gesture models to be trained). Given a curve C, a re-aligned index D' can be represented by:

$$D' = \{h_1, h_2, \ldots, h_n, (h_n + g_{n+1}), \ldots, (h_n + g_N)\} \tag{10}$$

where

$$h_i = C(g_i), \qquad i \leq n$$

If line equation, $C(g_k) = ag_k + b$, is used , then D' will be aligned linearly and only two parameters(a, b) need to be trained. Variances (a_σ, b_σ) and means (a_μ, b_μ) of a and b are trained on some training data and these are used for estimating the likelihood of the input sequence. Thus we can say the dimensionality of gesture is reduced to 2. Given the size l of 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 spotting, we estimate a' and b' such that $C'(g_j) = a'g_j + b'$ for each interval v_s where $g_j \in v_s$ and calculate the likelihoods L_a, L_b as follows:

$$L_a(a') = -\frac{1}{2\pi a_\sigma} e^{\frac{(a'-a_\mu)^2}{2a_\sigma^2}} \tag{11}$$

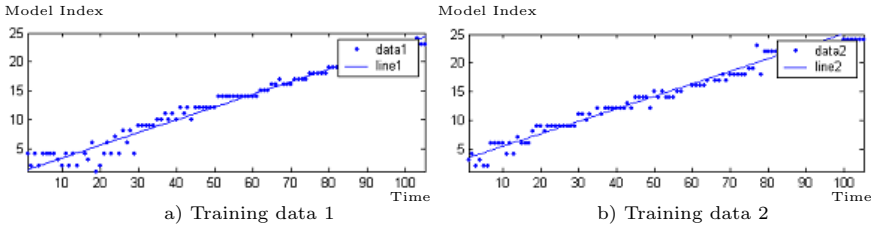
$$L_b(b') = -\frac{1}{2\pi b_\sigma} e^{\frac{(b'-b_\mu)^2}{2b_\sigma^2}} \tag{12}$$

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

$$L(v_s) = L_a(a') \times L_b(b') \tag{13}$$

Although the size of interval is fixed in implementation, various speed of gestures can be absorbed by the variance of b, which is determined from the training sequences. Finding maximum value of $L(v_s)$ for the interval where $L(v_s) > L_{threshold}$, we can spot gestures. The threshold value $L_{threshold}$ can be determined roughly because it does not affect the performance seriously. In our experiments on serve detection in tennis video, the difference of peaks of serve and non-serve gesture sequence was larger than 10^{-11} at least. For a robust estimation of C', we need to choose inliers of the estimated line parameters

because there are mismatched postures in posture matching, although most of them are matched correctly. Simply, we use posture indexes which are of interest for estimating C' and ignore the indexes which are out of interest.Of course we can use other robust methods to reject outliers. 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 few information to be matched a gesture, in which case we can discard the interval. Fig. 6 shows examples of lines which are fitted from 2 sets of training data by a least square fitting method.



a) Training data 1 b) Training data 2

Fig. 6. Examples of line fitting to training data set

3 Experimental Results

3.1 Environment

To evaluate the performance of the proposed posture matching and gesture spotting, we used an off-air interlaced sequence from the 2003 Australian open tennis tournament video. We separated out the fields from the frames, due to player's fast motion; some examples are shown in Fig. 2. Our target player is the near (or bottom) player. Initialization of the player's location was achieved by using a simple background subtraction method. The average width and height of the player are 53 and 97 pixels, respectively. To create the posture model database, we chose one play shot from the whole collection of play shots, and extracted player's posture manually. A model in database is represented by the curve coordinates of silhouette, zero-crossing points' locations in the image space, and additionally, zero-crossing points' height in the curvature scale space. For training gestures, i.e. estimating line parameters, we used only three sequences of a gesture data.

3.2 Posture Matching

Posture matching is achieved by using our new CSS-based features. We extracted the feature set and compared the distance to the posture models in the database. Fig. 7 shows some examples of foreground extracted from the input frames and matched models. In Fig. 7(a-c), the contours (yellow) of the foregrounds are

not extracted accurately because of shadows and white lines of the court. Nevertheless, we can see good matched results with our new features even though the contours are not good enough to match with the standard CSS matching method. Sometimes, matching are failed such as in Fig. 7(d), but in the 5 best matched models we could find proper model.

a) b) c) d)

Fig. 7. Foreground and matched model

3.3 Serve Gesture Spotting

We tested using 50 sequences, of which some include a serve gesture, some do not. The sequences are also partitioned by the players' identity ('A' and 'B') and position of player on the court('Left' and 'Right'). Fig. 6 shows a sequence projected onto the time-model space for a serve gesture. For training, we use a single shot, of which one of the players is serving on the right-hand side of the court.

Fig. 8 shows postures plotted onto the time-model space (first rows), likelihood graph versus time (second rows) and some snap-shots of sequences which are detected as a serve gesture(third rows). The gray area in the first rows indicates the which model indices are outliers (do not contribute to calculating line parameters for matching serve gesture). The yellow contours in the third rows shows a matched posture model. We can see that gesture spotting is achieved successfully, even though posture matching sometime fails. In Fig. 8(a), the player is bouncing the ball for a while, so the serve gestures is spotted at the 16th frame. In Fig. 8(b), there are two serve sequences (starting from 1st and 548th frame) in a shot. The player served twice because the first serve was a fault. Table 1 shows results of serve gesture spotting. In the table, true positive(TP), true negative(TN), false positive(FP) and false negative(FN) indicate serve gesture detected in correct location, no serve gesture detected where there is no serve, serve gesture detected where there is no serve and serve gesture not detected where there is a serve, respectively. Our method generates 90% correct result(TP+TN)and 5.2% incorrect result(FP+FN), while the results for the baseline algorithm, CDP, are 62% and 36.8%. The spotting results show that our method is substantially better than CDP.

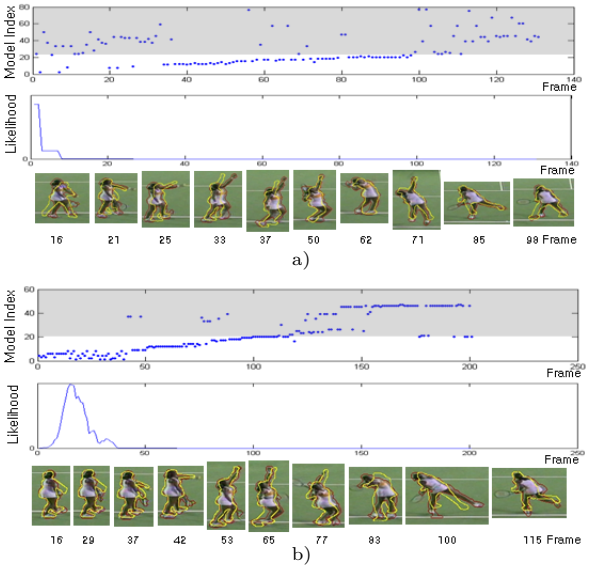


Fig. 8. Likelihood and some snapshots of spotted gesture for representative shots

4 Conclusions

In this paper, we presented a robust posture matching method and a gesture spotting method. For matching posture from low-resolution video frames, we proposed a feature based on CSS, which is robust to noise and significant shape corruption of the player's silhouette. For gesture spotting under a sequence of matched postures which may include mismatches, we calculate parameters for a curve of gestures plotted on the time-model space, and then estimate the likelihood using these trained parameters for spotting. According to our experiments, our spotting method generates 90% correct results and 5.2% incorrect results while the figures for the continuous dynamic programming algorithm are

Table 1. Gesture spotting result using CDP and the proposed method

		Player A		Player B		Total
		Left	Right	Left	Right	
CDP	TP	5/11	8/11	4/8	7/8	62% correct
	TN	7/12				
	FP	4/11	3/11	4/8	3/8	36.8% incorrect
	FN	6/11	3/11	4/8	1/8	
Proposed method	TP	8/11	11/11	7/8	7/8	90% correct
	TN	12/12				
	FP	0/11	0/11	0/8	0/8	5.2% incorrect
	FN	2/11	0/11	1/8	1/8	

62% and 36.8%, respectively. The proposed spotting method is robust to noise in the sequence data, with computational costs small enough to be calculated in real time.

Acknowledgements. This work was supported by the Korea Science and Engineering Foundation (KOSEF).

References

1. J. Alon, V. Athitsos, and S. Sclaroff, Accurate and Efficient Gesture Spotting via Pruning and Subgesture Reasoning, Proc. the IEEE Workshop on Human-Computer Interaction, Beijing, China, Oct. (2005) 189-198
2. W. J. Christmas, A. Kostin, F. Yan, I. Kolonias and J. Kittler. A System for The Automatic Annotation of Tennis Matches, Fourth International Workshop on Content-based Multimedia Indexing, Riga, June (2005)
3. A. Corradini, Dynamic Time Warping for Off-line Recognition of A Small Gesture Vocabulary, Proc. the IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, Vancouver, Canada (2001) 82-89
4. M. A. Fischler and R. C. Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, Comm. of the ACM, Vol. 24 (1981) 381-395
5. S. Kopf, T. Haenselmann and W. Effelsberg, Shape-base Posture and Gesture Recognition in Videos, Electronic Imaging, 5682, San José, CA, January (2005) 114-124
6. H.-K. Lee and J. H. Kim, An HMM-Based Threshold Model Approach for Gesture Recognition, the IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 21, No. 10 (1999) 961-973
7. F. Mokhtarian and M. Bober, Curvature Scale Space Representation: Theory, Applications & MPEG-7 Standardisation, Kluwer Academic (2003)
8. R. Oka, Spotting method for classification of real world data, The Computer Journal, Vol. 41, No. 8 (1998) 559-565
9. A.-Y. Park, and S.-W. Lee, Gesture Spotting in Continuous Whole Body Action Sequences Using Discrete Hidden Markov Models, Gesture in Human-Computer Interaction and Simulation, Lecture Notes in Computer Science, Vol. 3881 (2005) 100-111
10. L. Rabiner and B.-H. Juang, Fundamentals of Speech Recognition, Prentice-Hall (1993)
11. J. Sullivan and S. Carlsson, Recognising and Tracking Human Action, Proc. European Conf. on Computer Vision, Copenhagen, Denmark, May (2002) 629-644
12. B. J. Super, Improving Object Recognition Accuracy and Speed through Non-Uniform Sampling, Proc. SPIE Conf. on Intelligent Robots and Computer Vision XXI: Algorithms, Techniques, and Active Vision, Providence, RI (2003) 228-239
13. F. Yan, W. Christmas and J. Kittler, A Tennis Ball Tracking Algorithm for Automatic Annotation of Tennis Match, Proc. British Machine Vision Conference, Oxford, UK, Sep. (2005) 619-628
14. J. R. Wang and N. Parameswaran, Survey of Sports Video Analysis: Research Issues and Applications, Proc. Pan-Sydney Area Workshop on Visual Information Processing, Vol. 36, Sydney, Australia, Dec. (2004) 87-90