CVBASE '06 - Proceedings of ECCV Workshop on Computer Vision
Based Analysis in Sport Environments, Graz, Austria, May 12th, 2006, pp. 71-82

A Template-Based Multi-Player Action Recognition of the Basketball Game

Matej Perše, Matej Kristan, Janez Perš, and Stanislav Kovačič

University of Ljubljana, Faculty of Electrical Engineering
Tržaska 25, 1000 Ljubljana, Slovenia
{matej.perse, matej.kristan, janez.pers, stanislav.kovacic}@fe.uni-lj.si
http://vision.uni-lj.si

Abstract. In this paper we present a method for fully automatic trajectory based analysis of basketball game in the form of large and small scale modelling of the game. The large-scale game model is obtained by dividing the game into several game phases. Every game phase is then individually modelled using mixture of Gaussian distributions. The Expectation-Maximization algorithm is used to determine the parameters of the Gaussian distributions. On the other hand, the small-scale modelling of the game deals with specific basketball actions which can be defined in the form of action templates that are used by the basketball experts to pass their instructions to the players. For the recognition purposes we define the basic game elements which are the building blocks of the more complex game actions. These elements are then used to semantically describe the observed basketball actions and the templates. To establish if the observed action corresponds to the template, the similarity of descriptions is calculated using Levenstein distance measure. Experiments show that the proposed method could become a powerful tool for the recognition of various basketball actions.

1 Introduction

Every coach's goal is to be in possession of information that could give him a tactical advantage over his opponents. This is the reason for increasing volume of research concentrated on the problem of automatic or semi-automatic game analysis using different sources of data, e.g. using statistical data produced during the game [1], or manually labelled events from game videos [2]. Even though these methods proved to be very successful in discovering different complex interactions which are sometimes hidden even to the eye of experienced coach [2], they all had a crucial deficiency of tedious manual event detection and labelling. Labelling usually has to be done by one or more sport experts - not only very time consuming and potentially very expensive task. This has motivated our work towards a system, which would be able to automatically recognize and evaluate team behavior from the player trajectories obtained by the means of computer vision based tracking algorithms [3], see Fig. 1.

For this purpose, we decided to build a game model on two scales - a large and a small scale. The main purpose of the large-scale model is to segment the

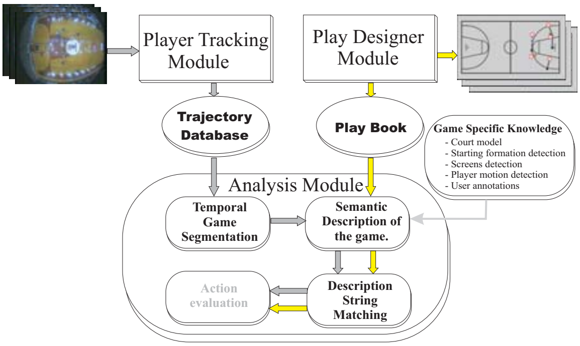


Fig. 1. System configuration and game processing pipeline.

game into several phases (e.g. offense, defense). This is important for small-scale game recognition, since there is a fundamental difference between analyzing for example the offensive phase or the defensive phase of the game.

In this work we present the recognition of small-scale events in the offensive phase of the game. We adopt the concept of event recognition in the form of action templates, which can be interactively defined by the game expert (Fig. 2).

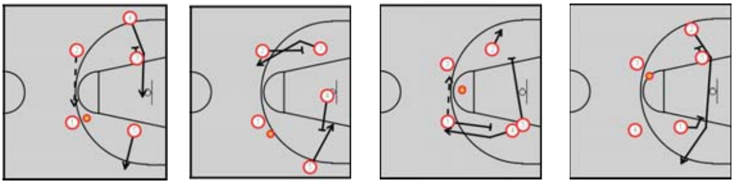


Fig. 2. Example of a "Flex" action template designed with Play designer module (see Fig 1).

We developed a special tool - the Play Designer module, that allows us to enter the spatial (the position on the court) and temporal (the duration and order of the events) properties of the basic events that are the building blocks of a more complex structure which is called *an (offensive) action*. Once the properties of these events are determined they are stored in the action database (playbook). The reason for using this approach is that the game experts usually do not possess the engineering knowledge, but are already used to design and to communicate their knowledge in the form of templates such as the one shown in Fig. 2.

This paper is structured as follows. In Section 2 the modelling and recognition of the large-scale basketball game is presented. Then, we present the detectors of the basic basketball elements and describe how these detectors can be used to semantically describe the offensive phase of the game and use these descriptions to recognize the more complex actions. In Section 4 we present the experimental results of our method. Finally, we discuss the advantages and disadvantages of the proposed game analysis approach.

2 Modelling the large-scale team behavior

Although different sports have very different rules and the goals of the game are diverse, there are some common large-scale properties, which can be observed in almost any type of a multi-player sport game. Many sport games can be regarded as a process consisting of some number of discrete phases such as offensive game, defensive game, time outs, inactive game, etc. We decided to build the game model consisting of three game phases: offensive game (m_1), defensive game (m_2) and time outs (m_3):

$$M = [m_1, m_2, m_3]^T \; ; \quad (i = 1, 2, 3) \; . \tag{1}$$

The probabilistic phase model was defined as a mixture of Gaussian model:

$$p(\mathbf{x}|m_i) = \sum_{i=1}^{n} \alpha_i p(\mathbf{x}|\mu_i, \Sigma_i) \; , \tag{2}$$

where parameters α_i represent mixing coefficients such that $\sum_{i=1}^{n} \alpha_i = 1$, and the multivariate Gaussian density function is defined as

$$p(\mathbf{x}|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(\mathbf{X}-\mu)^T \Sigma^{-1} (\mathbf{X}-\mu)} \; . \tag{3}$$

We base our phase models solely on the observation of player positions and their movement. Similar to the work of Erdmann [4], we model the game phases simply by observing the team gravity center. Therefore the state vector $\mathbf{x}$ is

$$\mathbf{x} = [x_t, y_t, \delta x_t, \delta y_t]^T \; , \tag{4}$$

where x_t and y_t represent the position of the team gravity center and δx_t and δy_t represent the velocity of the gravity center at time t:

$$x_t = \frac{1}{m} \sum_{i=1}^{m} x_i \; , \quad y_t = \frac{1}{m} \sum_{i=1}^{m} y_i \; , \tag{5}$$

where m stands for the number of active players on the court.

To determine the number of Gaussian distributions used in the phase model and their parameters we use the Expectation Maximization (EM) algorithm [5] on manually labelled sequences. The labelling of every sample into one of the three phases was done by using the expert knowledge.

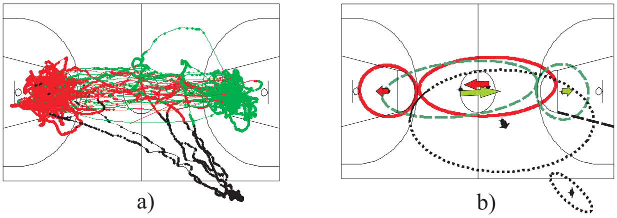


Fig. 3. *(a)* Labelling of sequences into one of the three phases. *(b)* Likelihood functions of the probabilistic game phase models obtained with EM algorithm ($p(x|m1)$ - full line, $p(x|m2)$ - dashed line, $p(x|m3)$ - dotted line). The arrows show the direction and size of velocity component in the state vector.

Once the model of the game is built, we can classify a given sample at time t into one of the three game phases simply by determining the highest *a posteriori* probability using Bayes formula:

$$m_*(t) = \max_{i \in \mathrm{M}}(p(m_i|\mathbf{x}(t))) = \max_{i \in \mathrm{M}} \left\{ \frac{p(\mathbf{x}(t)|m_i)p(m_i)}{p(\mathbf{x}(t))} \propto p(\mathbf{x}(t)|m_i)p(m_i) \right\}, \quad (6)$$

where $p(m_i)$ is the *a priori* probability that sample $\mathbf{x}$ belongs to the phase m_i. These probabilities were set in advance using expert knowledge,

$$p(M) = \left[0.45, 0.45, 0.10\right]^T . \quad (7)$$

Since the analysis is an off-line process, the information of past and future samples can be used. Therefore, once all the samples are individually labelled into one of the three categories, a nonlinear filter is used to incorporate the time continuity of the game into the model. This way, the t-th sample is assigned the phase label that receives the highest score among all the phases inside the observed filter window of length N,

$$m_{**}(t) = \max_{i \in \mathrm{M}} \left\{ \sum_{k=t-N/2}^{t+N/2} D(k) \right\}, \quad D(k) = \left\{ \begin{array}{l} 1;\, m_*(k) = m_i \\ 0;\, m_*(k) \neq m_i \end{array} \right\}. \quad (8)$$

The filter window should not be too wide, but should be wide enough to cover the shortest possible phase length. This way we reduce the unwanted effect of rapid model switching produced by not taking into account the neighboring samples and at the same time we preserve the phase boundaries.

3 Definition and detection of small-scale team behavior

Once the large-scale segmentation of the game is completed the analysis of small-scale team behavior can take place.

At this scale of the game, the specific game elements can be observed. These elements vary significantly from sport to sport and depend on the game rules and the ultimate goal of the team. For example in the offensive part of the basketball or handball game there are typical situations where a player tries to make the space for his teammate by blocking his teammate's defender and by doing so giving him a better chance to fulfill a given task. This element is called *screening*. Another example (*cutting*) can be observed in basketball or soccer, where players try to cut their way into the empty space on the court where they have a better chance of receiving the ball and scoring.

The sequences of such events define a play. Plays are trained in advance and can only be successful if all the players as a team are performing well. The sports experts call this "*team tactics*" and can be defined as the coordinated player activity aiming to achieve the given task. Although the team action can adapt according to the actual situation on the court, the general concept of the team tactics always remains the same. As it will be shown, it is possible to recognize the predefined team actions just by observing the sequence of the basic game elements.

For the purpose of small-scale action recognition the game-specific semantic description of the activities during the observed period of time was implemented. There are several reasons for selecting this approach:

– The influence of temporal variability of observed action and its elements is reduced since only the information about the order of the events is retained.
– There is no need of reanalyzing the trajectories when a new action definition is added to the database of templates (playbook) and thus significantly reducing the analysis time.
– Different players can play different roles in actions of the same type. That is why players have to be "casted" into their respective roles during the recognition process.
– The same approach can be used to compare semantic description of two different actions.

3.1 Game specific knowledge

Since we are dealing only with players trajectories and have no knowledge about possession of the ball, we turn our attention to the recognition of trajectory-based game elements. The main information that can be extracted this way is the types of the basic elements which compose the more complex action and the locations of their occurrences on the court. That is, player X movement (M) into some court region A can be described as

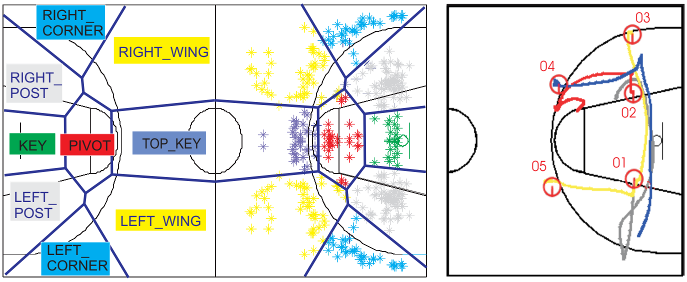
PlayerX_M_CourtRegionA.

For this purpose we need to define these game specific elements and build a model of the court to limit the possible number of regions.

Court modelling

Studying different sources of basketball literature and consulting several experts about the possible court division, we did not come across any consistent information about the type of the court model or any other kind of court parcelling. The main reason for this is that couches usually describe the game in the form of playing positions (eg. point guard, forward or center), which do not have deterministic edges such as our court regions should have had. This is why we decided to build our own model which would, as precisely as possible reflect the actual playing regions.

To build the court model we used 2x17 (left and right version) different definitions of offensive actions - action templates [6]. For every single action template, the positions of the basic events were observed and then used to divide the court into some number of non-overlapping playing areas using the k-means clustering algorithm. Fig. 4(a) shows the extracted points and court regions, nine in this case.



a) b)

Fig. 4. *(a)* The right side of the court shows the points extracted from the 34 action definitions. The left side shows the names of the regions obtained using k-means clustering algorithm. *(b)*Actual trajectories of "Flex" offense. Red circles denote players' starting positions.

Definition of basic events

Coupling the control theory with the knowledge of the basketball expert, we define a sport game as a sequence of events. In the basketball offense these events are: team starting formation, screening, player motion on the court, and ball passing. Since we have no information about the ball we only observe the first three elements.

– **Team starting formation** is the most distinctive indicator that an offensive action has been initiated. The probability that the formation has been observed can be defined as a sum of a product of Gaussian distributions over all players on the court, normalized with the factor of all players standing at their optimal positions defined by the playbook

$$p_{formation} = \frac{\sum_{i=1}^{n} N(r_i, \sigma_r) \cdot N(d_i, \sigma_d)}{\sum_{i=1}^{n} N(r_i, \sigma_r) \cdot N(0, \sigma_d)} \ , \qquad (9)$$

where r_i is the players current distance from the basket and d_i is the distance from his optimal position in Euclidean terms

$$r = \sqrt{(x - x_{basket})^2 + (y - y_{basket})^2} \ , \qquad (10)$$

$$d = \sqrt{(x - x_{ref})^2 + (y - y_{ref})^2} \ . \qquad (11)$$

This definition of the formation detector is particularly suitable as it takes into account that players which are further from the basket do not have as many points of orientation such as free throw line, and as a consequence they tend to position themselves further from their optimal position.

– **A screen** is defined as a close contact of two players [6] where the player who is making the screen stands as still as possible and the player who is taking the screen runs close nearby the first player. This way a screen probability can be calculated as

$$p_{screen} = N(d, \sigma_d) \cdot N(v_{min}, \sigma_v) \ . \qquad (12)$$

Here the variable d defines the distance between the two players and v_{min} defines the velocity of the slowest player.

– The court model described in the beginning of this section is used to quantize and semantically describe the **payer motion on the court**. The court model was designed to account for the variability in player's movement relative to the ideal path in the way that the variability inside the region is not important for the player's performance. Using this model it is possible to describe the player's motion simply be observing the regions of the court the player has covered.

Using the above definitions we can obtain the following semantic description (Table 1) of trajectories from Fig. 4(b).

Table 1. Initial part of semantic description of trajectories from Fig. 4.The numbers denote the occurrence order.

(1) P1_M_RIGHT_POST,	(7) FORM_flex1,	(12) P4_M_LEFT_POST,
(2) P2_M_LEFT_POST,	(8) P3_M_LEFT_POST,	(13) SCR_2_4_LEFT_POST,
(3) P3_M_LEFT_CORNER,	(9) SCR_2_3_LEFT_POST,	(14) FORM_flex1 inv,
(4) P4_M_LEFT_WING,	(10) P1_M_RIGHT_CORNER,	(15) P2_M_LEFT_WING,
(5) P5_M_RIGHT_WING,	(11) P3_M_KEY,	(16) SCR_2_3_LEFT_POST ...

3.2 Action recognition using action templates

The main goal of action recognition with templates is to establish the correspondence between the observed action and the predetermined action template, such as the one from Fig. 2.

On the contrary to the work of Intille and Bobick [7] or Jug et al. [8], where authors built their game models using belief networks and temporal graphs to represent the action models and use these model for recognition and evaluation of multi-agent actions from noisy trajectory data, we adopted much simpler but effective way of comparing the semantic descriptions of the analyzed action and all action templates from the playbook. This way we avoid the painstaking and sometimes very subjective decision making of determining the temporal and logical relationships of every belief network [7] and their mutual dependencies. Since there are usually only few or none training examples, it is impossible evaluate these parameters from training set and as a consequence they are subjected to the personal experience and technical knowledge of the person who designs them. We designed our system based on the preposition that the person who will be using the system has very little or no technical background, but is able to pass his knowledge to the system in the for of game templates. By using the proposed semantic descriptions, we are able to automatize the process of quantifying and condensing the information to only those key elements that are needed for a successful recognition of the selected action.

Another advantage of the proposed framework in comparison to those mentioned above is that our approach considers the possible mismatching between the predetermined players roles and actual ones and assigns role labels to players automatically (not manually as in [7] from starting positions), based on the observed situation on the court. These way we reduce influence of player mismatches. This is very important since we obtain the analyzed data by video tracking of multiple interacting targets [9] which may be, event though supervised by the operator, subjected to errors.

3.3 Matching trajectories to action templates

After the semantic descriptions are obtained, they can be compared using different procedures, e.g. dynamic programming [10]. To do this, we first divide the real trajectories and each template semantic description into five single player role descriptions called *player agendas*. The agendas are built by observing if a player was involved in a particular action event. For example, all players are involved in the starting formation, so the formation name is assigned to all agendas. The screen label is assigned only to agendas of those two players that are involved in the screen and the player motion label is assigned only to the agenda that represents that player. Considering these rules we can transform the semantic description from Table 1 to the five agendas in Table 2.

Table 2. Five player agendas built from the semantic description from Table 1. The number of agendas match the original player numbers.

ag1	ag2	ag3
(1) M_RIGHT_POST	(1) M_LEFT_POST	(1) M_LEFT_CORNER
(2) FORM_flex1	(2) FORM_flex1	(2) FORM_flex1
(3) M_LEFT_POST	(3) SCR_LEFT_POST	(3) SCR_LEFT_POST
(4) M_RIGHT_CORNER	(4) SCR_LEFT_POST	(4) M_KEY
(5) FORM_flex1 inv ...	(5) FORM_flex1 inv	(5) FORM_flex1 inv
	(6) M_LEFT_WING	(6) SCR_LEFT_POST ...
	(7) SCR_LEFT_POST...	

ag4	ag5
(1) M_LEFT_WING	(1) M_RIGHT_WING
(2) FORM_flex1	(2) FORM_flex1
(3) M_LEFT_POST	(3) FORM_flex1 inv...
(4) SCR_LEFT_POST	
(5) FORM_flex1 inv...	

Once the agendas are obtained it is necessary to establish the correspondence between different player agendas as defined in action template, and the agendas observed from real trajectories (casting). This is done by cross-validating the five agendas from the template with those from the observed trajectories, that is calculating $m! = 5! = 120$ different possible matchings (P_l). In our case a simple Levenstein distance [10] is used as a similarity measure of two different agendas which is normalized with the sum of agendas lengths $(l+k)$. The highest overall similarity (smallest Levenstein distance $D(R,T)$) of all real trajectories and template agendas can be calculated as

$$D(R,T) = min\{\sum_{i=1}^{m} \frac{D_l(agR_i, agT_j)}{l+k}\}; \quad j = P_l(i), \quad l = 1 \dots 120; \qquad (13)$$

where $D_l(agR_i, agT_j)$ stands for the Levenstein distance between i^{th} player agenda (agR_i) and j^{th} template agenda (agT_j). The $D_l(agR_i, agT_j)$ indicates how well has player i performed the elements of j^{th} role in the template. The match $D(R,T)$ that has the smallest distance represents the optimal assignment of player roles to roles from the template and at the same time to shows how well is the observed action performed according to the compared action template.

3.4 Assigning template label to the observed action

To determine the template that is the most similar to the observed action, we first repeat the action-template matching procedure for every definition from the playbook database and then calculate the average template penalty. Finally the unknown action is assigned the label of template or templates that have the smallest penalty but only if their penalty is 10 % smaller then the average one. This way we prevent the wrong recognition of action in cases when the observed action is not in the database. Figure 5 shows the results of matching 17 different templates to a "Flex" offense action.

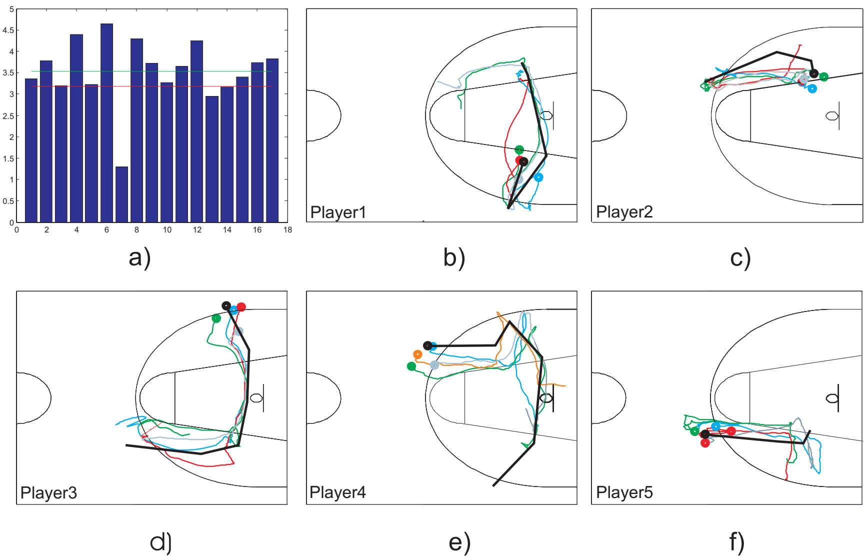


Fig. 5. *(a)* Results of matching "Flex" action to 17 different templates. Green (upper) straight line shows the average penalty and red (lower) line shows the 10% boundary. *(b-f)* Figures show the ideal player paths as defined in the "Flex" template (black straight line) and successfully matched player trajectories of four different "Flex" actions.

4 Experimental results

To test the recognition accuracy on large and small scales of the game we designed two separate sets of experiments. For the evaluation of large-scale recognition, we acquired about 40 minutes of real game trajectories (60700 trajectory samples). Our goal was to determine how the amount of training data and the number of Gaussian models used to represent the game phases influence the final recognition result. As it can be observed in Table 3, the best classification results were obtained when the mixture of two Gaussian models was used to model each game phase (Fig. 3 b). The explanation for this is that when using more components the model tends to over-fit the training data. The two components mixture could also be explained as a model of the transition from one phase to another and the model of the main behavior during that phase. We can also observe that there is very small decrease in the performance results even when a relatively small percentage of data is used for training purposes.

Table 3. Experimental results of game segmentation using different ratios for teaching the game model and different number of Gaussian distributions for modelling game phases with constant (five seconds) width of non-linear filtering window.

	The number of Gaussian models used for modelling single game phase $\{ m_1, m_2, m_3 \}$				
train/test ratio [%]	{1,1,1}	{2,2,2}	{3,3,3}	{2,2,1}	{3,3,2}
30/70	92.5	94.0	93.4	93.3	92.4
50/50	92.8	94.2	93.4	92.8	93.4
100/100	92.8	94.5	95.6	92.9	94.3

In our second experiment, we tested the accuracy of the proposed small-scale (action) recognition method using four different types of court models (See Table 4). Two of them were produced using court modelling described in Section 3.1 and two of them were produced simply by dividing the court into n identical regions (grid model). To obtain the testing sequences we asked a local basketball team to perform three different basketball offenses 71 times. Among these 71 offenses 39 were played with and 32 without defensive team. The offenses were matched to 34 (2x17) different action templates, which were obtained from various basketball literature such as [6], and by the local team coach.

Table 4. Experimental results of 65 different offensive basketball actions recognition with different types of court model. The percentages do not sum up to 100 % because some actions matched to two different templates.

	K-means model (9 regions)	K-means model (16 regions)	Grid model (9 regions)	Grid model (18 regions)
Correctly recognized	71 (100 %)	66 (92.96 %)	64 (90.14 %)	67 (94.37 %)
Mislabelled	0 (0 %)	5 (7.04 %)	7 (9.86 %)	4 (5.63 %)

The experiments show that there are small variations in the recognition results when different types of court models are used. As it can be seen the k-means clustering of the court improves the recognition performance.

5 Conclusion

We presented a novel approach towards the automatic trajectory-based analysis of the basketball game. We demonstrated that by modelling the game just by the gravity center of the teams, it was possible to segment the basketball game into different game phases. Furthermore we showed that the best results were obtained when every game phase was modelled as a two component Gaussian mixture model. We also demonstrated that only a small amount of data was sufficient to estimate the model parameters without significantly reducing the recognition rate.

In the second part of the article, we presented a method for detecting specific complex basketball actions, which can be defined by the user in the form of action templates. The presented results show that by comparing the semantic description of the observed trajectories and action templates it was possible to successfully match the observed trajectories to the templates. They also suggest that the recognition result are improved if the court model that maximizes the possible variance of key action elements is selected. In our case this model was obtained by clustering the starting and ending positions of key action elements that have been extracted from the predetermined action templates.

References

1. Bhandari, I., Colet, E., Parker, J., Pines, Z., Pratap, R., Ramanujam, K.: Advanced scout: Data mining and knowledge discovery in NBA data. Data Mining and Knowledge Discovery **1** (1997)
2. Borrie, A., Jonsson, G., Magnusson, M.: Temporal pattern analysis and its applicability in sport: an explanation and exemplar data. Journal of Sports Sciences **20** (2002)
3. Perše, M., Perš, J., Kristan, Vučkovič, M.G., Kovačič, S.: Physics-based modelling of human motion using kalman filter and collision avoidance algorithm. ISPA05 (2005) 328–333
4. Erdmann, W.: Gathering of kinematic data of sport event by televising the whole pitch and track. R. Rodano, Editor, Proceedings of 10th ISBS symposium, International Society of Biomechanics in Sports, Milan, Italy (1992) 159–162
5. McLachlan, G., Krishnan, T.: The EM Algorithm and Extensions. Wiley Series in Probability and Statistics. Wiley, New York (1997)
6. Kresse, J., Jablonski, R.: The Complete Book of Man-To-Man Offense. 2nd edn. Coaches Choice (2004)
7. Intille, S., Bobick, A.: A framework for recognizing multi-agent action from visual evidence. In: National Conference on Artificial Intelligence (AAAI). (1999)
8. Jug, M., Perš, J., Dežman, B., Kovačič, S.: Trajectory based assessment of coordinated human activity. In: Computer Vision Systems, Proceedings of Third International Conference ICVS 2003, Graz, Austria, (Lecture Notes in Computer Science, vol.2626), Springer (2003) 534–543
9. Kristan, M., Perš, J., Perše, M., Bon, M., Kovačič, S.: Multiple interacting targets tracking with application to team sports. In: 4th International Symposium on Image and Signal Processing and Analysis ISPA. (2005) 322–327
10. Makinen, V., Navarro, G., Ukkonen, E.: Transposition invariant string matching. Proc. 20th International Symposium on Theoretical Aspects of Computer Science (STACS 2003), in: Lecture Notes in Comput. Sci., vol. 2607, Springer-Verlag. (2003) 191202