Computer Vision Winter Workshop 2008, Janez Perš (ed.)
Moravske Toplice, Slovenia, February 4–6
Slovenian Pattern Recognition Society, Ljubljana, Slovenia

Temporal Segmentation of Group Motion using Gaussian Mixture Models

Matej Perše[1], Matej Kristan[1], Janez Perš[1], Goran Vučkovič[2], and Stanislav Kovačič[1]

[1]Faculty of Electrical Engineering, University of Ljubljana,
Tržaška 25, SI-1000 Ljubljana, Slovenia
matej.perse@fe.uni-lj.si

[2]Faculty of Sport, University of Ljubljana,
Gortanova 22, SI-1000 Ljubljana, Slovenia

Abstract

This paper presents a new trajectory-based approach for probabilistic temporal segmentation of team sports. The probabilistic game model is applied to the player-trajectory data in order to segment individual game instants into one of the three game phases (offensive game, defensive game and time-outs) and a nonlinear or Gaussian smoothing kernel is used to enforce the temporal continuity of the game. The presented approach is compared to the Support Vector Machine (SVM) classifier on three basketball and three handball matches. The obtained results suggest that our approach is general and robust and as such could be applied to various team sports. It can handle unusual game situations such as player exclusions, substitution or injuries which may happen during the game.

1 Introduction

Development of video technologies in last decade has enabled the sport experts to obtain large quantities of video material. At the same time the increase in computational power and the advancement of video technology have made possible for the experts to obtain large quantities of data about human behavior and have shifted the research focus from human recognition and tracking [1, 2] towards video-based behavior analysis and behavior understanding [3, 4]. Most popular research domains in the field of sport include video summarization [5], segmentation [6, 7], indexing [8], highlight extraction [9, 10], as well as complex recognition of behavior patterns [11], and skill evaluation of individual players and whole teams [12, 13]. The main goal of these research activities is to assist sport experts in their work by reducing the amount of data that has to be analyzed. For example in production logging [6] it is crucial that user obtains only the relevant video clips which are then included in the program. Similarly, in the field of player behavior and game tactics analysis [13] the experts would usually want to obtain the specific contextual parts of the game (e.g. offense and defense or active and passive game) and are less interested in specific video clips.

The aim of this paper is to show the efficiency of Gaussian mixture model (GMM) for temporal segmentation of group motion. We demonstrate the superiority of this approach in comparison to the Support Vector Machine (SVM) [14] classifier even in the cases when relatively small amount of training data is available. The idea of the proposed approach is to develop a universal segmentation method which can be trained to adapt to particular type of motion. We will show that by using a pre-computed game model and a two step segmentation procedure it is possible to segment the game into individual game phases. Two examples from the sport domain will be used to demonstrate that fairly simple features that are based solely on the players' positions can be used for segmentation.

The rest of this paper is organized as follows. Section 2 presents some problems of the multi-player game segmentation and provides details about our segmentation procedure. We report on experimental results in section 3. Finally, in Section 4 the conclusions and the future work are discussed.

2 Temporal segmentation of team sports

Team sports are determined both by the game rules and by the collective aims, which the teams must pursue to defeat their opponents. In many popular team sports the teams' activities alternate between offense and defense, with minor interruptions, such as time outs, free throws or free kicks. Therefore, the game can be regarded as a process consisting of a certain number of discrete phases. These phases correspond to offensive play, defensive play, time outs, inactive play, free throws, free kicks, and other miscellaneous activities.

In most team sports (e.g. basketball, handball, hockey, etc.) the state of the team can be defined by the possession of the ball or, in the case of hockey, possession of the puck. However, due to the size, speed and occlusions of these objects in video, it is nontrivial to obtain accurate trajectories even when the most advanced video processing technology is used. Moreover, even if the ball position would be known it would still be very difficult to disambiguate the ball possession and therefore the state of the teams. For these reasons, we decided not to use the information about the ball for game segmentation.

Our segmentation procedure is divided into two steps:

- In the first step each individual time segment is labelled into one of the game phases using a probabilistic model of the game. We use two different classification approaches. In the first approach the game model is built using a Gaussian Mixture Model (GMM). In the second approach, the Support Vector Machine (SVM) is used for classification.
- In the second step the label of each time segment is recomputed by considering the neighboring samples. This way we enforce the temporal continuity of the game and significantly reduce the effect of undesirable phase switching. By doing so, we are able to additionally improve the segmentation results.

To obtain a general game model, we first have to design a trajectory-based state vector which would differentiate well among the individual game phases and would not depend on the number of players on the court or specific sport-related rules.

2.1 The state vector

In the case of basketball and handball, our model assumes that the play consists the following three phases: offensive play (m_1), defensive play (m_2) and time outs (m_3):

$$M = \{m_1, m_2, m_3\} . \tag{1}$$

Our basic assumption is that there exists an intrinsic relation between the parameters of player motion (position, velocity and direction) and the phases of the game. Therefore, our model is based solely on the observation of the players' motion.

Our initial idea was to build the separate game models for each team [11]. Similarly to the work of Erdmann [15], we calculated the *collective* position of the active players by calculating the team centroid (x_t, y_t) (i.e. the mean position of the players belonging to the same team)

$$x_t = \frac{1}{n}\sum_{i=1}^{n} x_i \quad , \quad y_t = \frac{1}{n}\sum_{i=1}^{n} \cdot y_i. \tag{2}$$

Additionally, we also encoded the two-dimensional motion of this centroid across the court $(\Delta x_t, \Delta y_t)$ as a position difference of two consecutive centroid positions that are calculated from previously smoothed player trajectories (see Section 3.1). We defined the *team state vector* **x**(t) as

$$\mathbf{x}(t) = [x_t, y_t, \Delta x_t, \Delta y_t]^T . \tag{3}$$

Although the described approach yielded reasonably good results [11], we have observed that a slightly better results were obtained when the trajectories of the same team were used for training and testing and they dropped when the teams were different. Additionally, we have observed that this model failed in the cases of tracking errors or unusual game situations such as player injuries, exclusions or substitutions. In such situations the excluded or injured player is regarded as an outlier since he/she is not actually involved in the game [1].

[1]Some examples of such situations can be found at http://vision.fe.uni-lj.si/research/SportA/segmentation.html

For these reason we have decided to improve the state vector as follows:

- Since the two teams have the opposite states but they move in a very similar manner regardless of the game state, we compute the *overall game model* by considering all (n) active players from both teams. Using the obtained *game model* one can calculate the state of the first team and simply invert the offensive and defensive states for the other team.
- We introduce weight w_i which indicates how well an individual player fits into the flow of the game. For this purpose we fit a Gaussian distribution $\mathcal{N}(\cdot; \mu, \sigma)$ with mean $\mu(t) = [\mu_x(t), \mu_y(t)]^T$ and covariance $\Sigma(t)$ on the positions of the active players. The weight of each individual player position is defined as a likelihood of that position under the fitted Gaussian distribution.

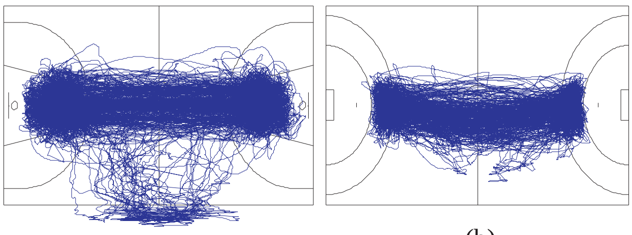
$$w_i(t) = \frac{\mathcal{N}(x_i(t); \mu(t), \Sigma(t))}{\sum_1^n \mathcal{N}(x_i(t); \mu(t), \Sigma(t))}, \tag{4}$$

where $x_i(t) = [x_i, y_i]^T$ is the position of the $i\!-\!th$ player.

The centroid position is then computed as

$$x_t = \sum_{i=1}^{n} w_i(t) \cdot x_i(t) \quad , \quad y_t = \sum_{i=1}^{n} w_i(t) \cdot y_i(t). \tag{5}$$

As an illustration, Figure 1 shows the trajectories of the centroid for the basketball and handball games.



(a) (b)

Figure 1: Trajectory of the centroid: (a) Basketball. (b) Handball.

2.2 Building the game model

We use two different approaches for building the game model. The first one is Gaussian Mixture Model (GMM) [11] and the second is Support Vector Machine (SVM) classifier.

In both cases the model is built in advance and is meant to be as general as possible so that it is capable to handle different game irregularities as well as different strategies of different teams (e.g. defensive or offensive play).

2.2.1 The Gaussian mixture model We define a probabilistic model of the game phases using a mixture of Gaussians [16]

$$p(\mathbf{x}|m_i) = \sum_{k=1}^{n} \alpha_k^{(i)} \cdot \mathcal{N}(\mathbf{x}; \mu_k^{(i)}, \Sigma_k^{(i)}), \qquad (i = 1, 2, 3), \tag{6}$$

where m_i represents the game phase, α_k represent the mixing coefficients, such that $\sum_{k=1}^{n} \alpha_k^{(i)} = 1$, $\mathcal{N}(\mathbf{x}|\mu_k^{(i)}, \Sigma_k^{(i)})$ is the k-th Gaussian density function with mean $\mu_k^{(i)}$ and the covariance matrix $\Sigma_k^{(i)}$, and n represents the number of Gaussian density functions used to model each phase of the game. To determine the parameters of the Gaussian density functions we use the Expectation Maximization (EM) algorithm [17] on manually labeled training sequences.

Using the above model, we can calculate the probability of the model m_i given the current *flow vector* **x**(t) using Bayes formula

$$p(m_i|\mathbf{x}(t)) = \frac{p(\mathbf{x}(t)|m_i)p(m_i)}{p(\mathbf{x}(t))}, \qquad (i = 1, 2, 3). \tag{7}$$

Provided that the probability $p(\mathbf{x}(t))$ remains constant for all models m_i, the classification of a given sample $\mathbf{x}(t)$ at time t is expressed as

$$m_*(t) = \arg\max_{m_i \in \mathrm{M}} \{p(\mathbf{x}(t)|m_i)p(m_i)\}, \qquad (i = 1, 2, 3), \tag{8}$$

where $p(m_i)$ is the *a priori* probability of phase m_i. These probabilities were estimated in advance by roughly estimating the amounts of time teams spent in each individual phase of the game.

$$p(M) = \begin{cases} [0.4,\ 0.4,\ 0.2]^T; & basketball \\ [0.48,\ 0.48,\ 0.04]^T; & handball \end{cases} \tag{9}$$

2.2.2 Support Vector Machines Support Vector Machines (SVMs) are a well known and useful technique for data classification which is often used for machine learning and pattern recognition [14, 18, 19]. The main objective of the SVM is to find a hyperplane, which separates the feature space into two parts, each containing the majority of samples from each of the two classes. Further details on the subject of SVMs can be found in different literature [14, 19, 20, 21].

SVMs were originally designed for binary classification. However, several methods have been proposed where a multi-class classifier in constructed by combining several binary classifiers [22]. The two best known approaches for combining the binary classifiers are *one-against-all* and *one-against-one* approach. In the first case a (K-1) classifiers are built using samples from one class as positive examples and semples from all other classes as negative examples. In this case the sample is assigned to the class that receives the highest vote among the voting classifiers

$$m_*(t) = \max_{k=1:K-1} y_k(\mathbf{x}(t));. \tag{10}$$

In the second case a classifier is built for every pair of clases, building altogether $K(K-1)/2$ classifiers. The test sample at time t is classified into the class m_* that receives the higher number of ”votes” among the voting classifiers.

In our case a SVM with a RBF kernel and the *one-against-one* approach presented in [21] is used for the multi-class classification.

2.3 Enforcing the temporal continuity

The two classifiers described above provide a reasonably good classification for the individual time instants. However, when used for trajectory segmentation, they produce a number of faulty short segments, since they do not encode any temporal continuity. To enforce this requirement we smooth the output of the first step of segmentation using one of the following kernels:

- In the first case, a nonlinear kernel

$$m_{**}(t) = \arg\max_{m_i \in \mathrm{M}} \left\{ \sum_{k=t-K}^{t+K} D_{m_*, m_i}(k) \right\}, \tag{11}$$

where

$$D_{m_*, m_i}(k) = \begin{cases} 1; & m_*(k) = m_i \\ 0; & otherwise \end{cases} \tag{12}$$

is used to smooth the labels. In this case the t-th sample is assigned a label that receives the highest score among the observed individual labels inside the kernel window of length $2K + 1$. The kernel width was determined experimentally (see Section 3.2 for details) and is set to twice the length of the theoretically shortest possible segment of play, which in basketball as well as in handball, according to the experts' opinion, corresponds to approximately three to four seconds.

- In the second case, a Gaussian kernel is used to smooth the labels

$$m_{**}(t) = \arg\max_{m_i \in \mathrm{M}} \left\{ \sum_{k=t-3\sigma}^{t+3\sigma} \mathcal{N}(k; t, \sigma) \cdot D_{m_*, m_i}(k) \right\}. \tag{13}$$

In this case the individual neighboring label are first weighted using the Gaussian distribution and the observed sample is classified into the class with the highest sum of weighted label scores.

3 Experimental results

We present the experiments which were carried out on two datasets of basketball and handball matches. Trajectory data from three basketball and three handball matches were acquired using the method described in subsection 3.1. This way we have obtained about 376000 trajectory samples for each basketball player and 331000 trajectory samples for each handball player. Several experiments were performed to test different aspects of the proposed segmentation method.

3.1 Data acquisition and preparation

We have used two 25fps PAL video cameras, fixed to the ceiling of the sports hall. An image from one of the cameras is shown in Figure 2.

To obtain motion data we performed operator-supervised tracking on the obtained videos. We used a modified color-histogram-based *CONDENSATION* algorithm [23, 24] as the tracking engine, built into a user-friendly graphical interface. The sport expert supervised the tracking and corrected the errors that appeared during the tracking process. The tracking was coupled with the appropriate calibration, which made it possible to map the image coordinates to the real-world (court) coordinates and compensate for the radial distortion that is present in the original video data. At the end of the tracking the data were smoothed using a 25-samples-wide symmetric Gaussian filter kernel [25], which proved to be most suitable for reducing the tracking jitter and retaining the measurement accuracy.

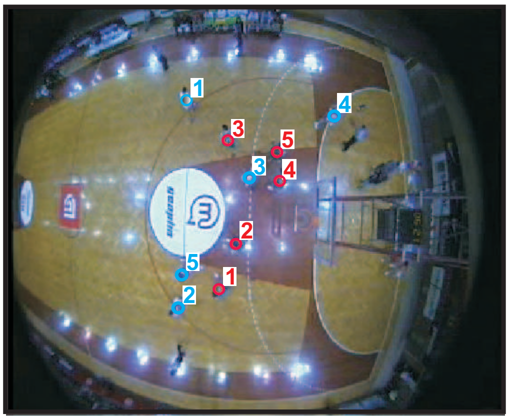


Figure 2: Operator-supervised tracking in progress.

3.2 Defining the parameters of GMM

Our segmentation procedure has two free parameters which have to be determined prior to the segmentation procedure. The first one is the number of components of the GMM that are used to model each of the game phases and the other is the length of the smoothing kernel.

To determine the number of components of GMM, we used nonlinear smoothing kernel and fixed its width to 180 frames. We selected different number of components for each of the game phases (see Figure 3). For each set of components, a six-fold cross-validation test was performed, so that each subset of the original database represented one half of the game (approximately 54000 samples) and the average classification rate was calculated.

By studying carefully Figure 3, one can observe that, in the case of basketball, the classification rate does not vary significantly when two or more components are used to model the individual game phase. On the other hand, in the case of handball, the increase in the number of components significantly influences the segmentation results. The main reason for these is that the handball rules permit many more

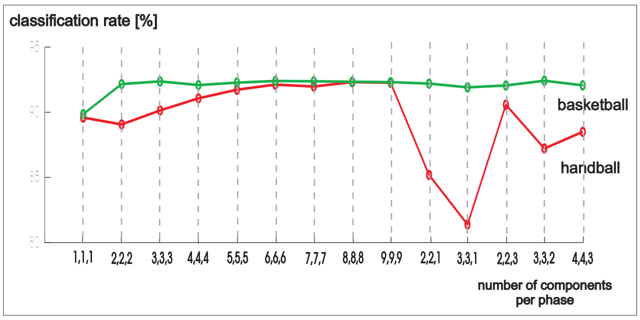


Figure 3: Classification rate for different number of components. Numbers on the x-axis represent the number of components per phase (e.g. 1,1,1 - denotes that one component was used to model each phase of the game).

unusual events such as player exclusions (i.e. the team is playing with less players) or rapid substitution during the active game, which have to be incorporated into the model. By considering the obtained results and considering the fact that the time needed to build the model and to perform the segmentation increase when a more complex model is used, we decided to model each phase of the game as a six-component mixture of Gaussians (Figures 4 and 5).

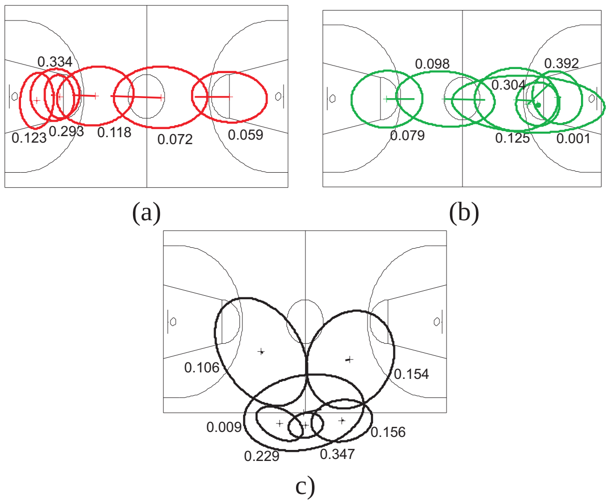


Figure 4: Game model for basketball: (a) offense, (b) defense, (c) time-out. Numbers on the ellipses denote the values of the mixing coefficients.

The aim of our second experiment was to determine the best type (nonlinear or Gaussian) and the optimal width of the smoothing kernel. For this purpose, we repeated the above experiment by fixing the number of GMM components to six and selected different width of the two smoothing kernels. Our aim was to determine how the kernel length influences the overall classification rate (Figure 6-a) as well as the smoothing capabilities of the two kernels which can be determined as a ration between the number of game segments produced by our method and true number of segments which were determined by the game expert (Figure 6-b).

In Figure 6(a) one can observe that the two kernels produce very similar segmentation results. However, by com-

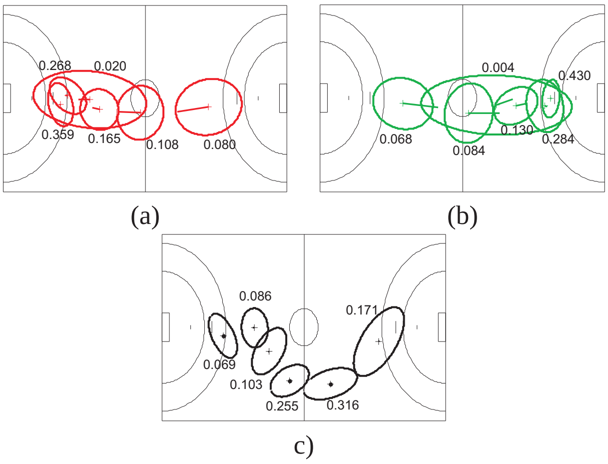


Figure 5: Game model for handball: (a) offense, (b) defense, (c) time-out. Numbers on the ellipses denote the values of the mixing coefficients.

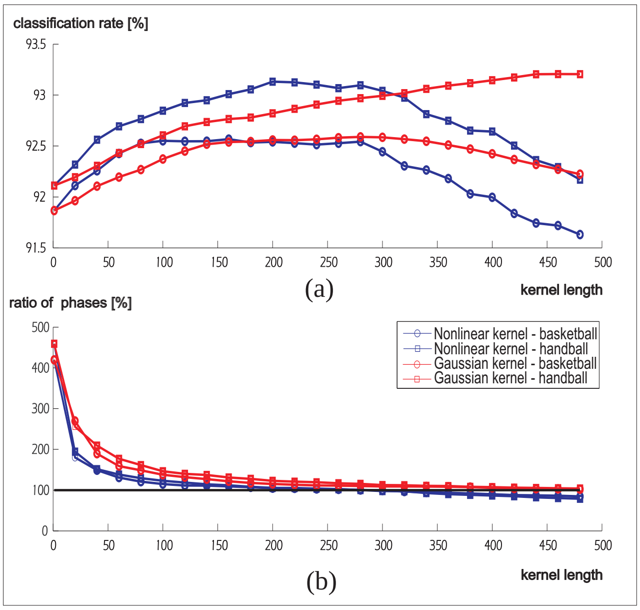


Figure 6: Segmentation results for different lengths of nonlinear and Gaussian smoothing kernel. (a) Classification rate. (b) Ration between number of phases obtained with our procedure and number of manually labelled phases.

paring the results by sport one can observe, that when the nonlinear kernel is used in both sports the peak is obtained when the kernel length is around 200 samples or eight seconds which is exactly twice the length of the theoretically shortest possible game phase. By observing the results when the Gaussian kernel is used, one can notice that although the obtained results are more stable for different lengths of kernel, the peak result is radically different when the type of the analyzed sport changes.

By studying carefully Figure 6(b), one can observe that both kernels significantly reduce the number of false, usually very short segments from almost five times the true number to approximately the same number. Additionally, one can observe that the number of segments is equal when the width of nonlinear kernel is between 200 and 300 frames which is exactly where the peak in the classification rate occurs. In the case of the Gaussian kernel the optimal width is between 400 and 500.

Given the above results, one can conclude that the nonlinear kernel is a more suitable choice since it allows a single selection of the kernel length for both sports. However, if the stability of the results would be the main selection criteria, then the Gaussian kernel with a different width for each sport would be a more suitable choice.

3.3 Segmentation results

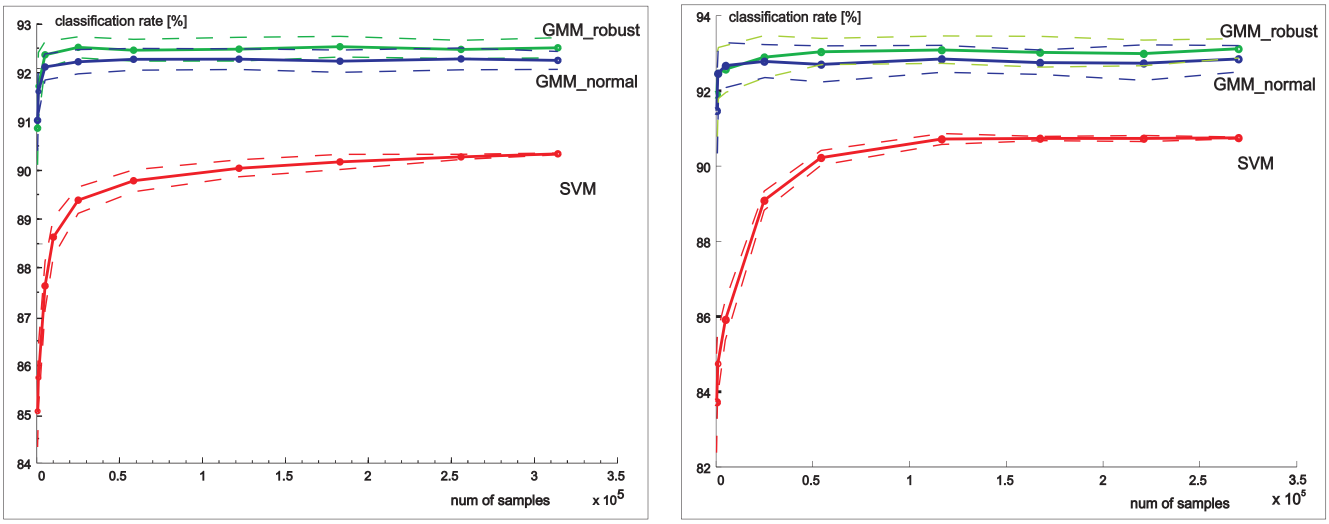
In our last experiment we investigated how the number of training samples that are used to build the game model influences the segmentation results. The main purpose of this experiment was to determine if it is possible to obtain good results in cases when the game model is built from relatively small amount of data. For this purpose, we repeated the six-fold cross-validation test several times. The amount of training samples varied from 500 to 270000 samples. Since the samples were selected randomly, we repeated the experiment 20 times for each selection of sample amount. To demonstrate the efficiency of the GMM, we performed the same experiment by replacing the GMM classifier with a SVM classifier. Additionally, we used different types of state vector to model the game flow. In all cases, a 200-samples-wide nonlinear kernel was used to smooth the labels obtained from the classifiers.

Type of Model	Basketball	Handball
Single team GMM model	89.18	90.55
SVM Model	90.05	90.73
Normal Joined GMM Model	92.18	92.83
Robust Joined GMM Model	92.47	93.11

Table 1: Average classification rate when 270000 samples were used for training the model.

Table 1 presents the classification results for four different segmentation approaches when maximum amount of data was used for training. First row shows the results for a two-component GMM model with a single team state vector which was used in our previous research [11]. In the second, the results for SVM which uses the robust state vector that considers players from both teams are shown. The last two rows present the result for a six-component GMM model with a normal and robust state vector computed from players from both teams. Additionally, Figure 7 presents the average segmentation results for different amounts of training data and Figure 8 shows the times needed to build the game model and the times needed to perform the segmentation of one half of a handball game.

By studying the segmentation results for the four methods (Table 1), one can conclude that a significantly better result can be obtained when a GMM model with a normal or robust joined state vector is used for classification. If the results for the two joined state vectors are compared, a slightly better result is obtained when the robust joined state vector is used, although the difference is not statistically significant.



(a) (b)

Figure 7: Segmentation results when using different amount of data were used to train the model. The dashed lines represent the variability in the results. a) Basketball. b) Handball.

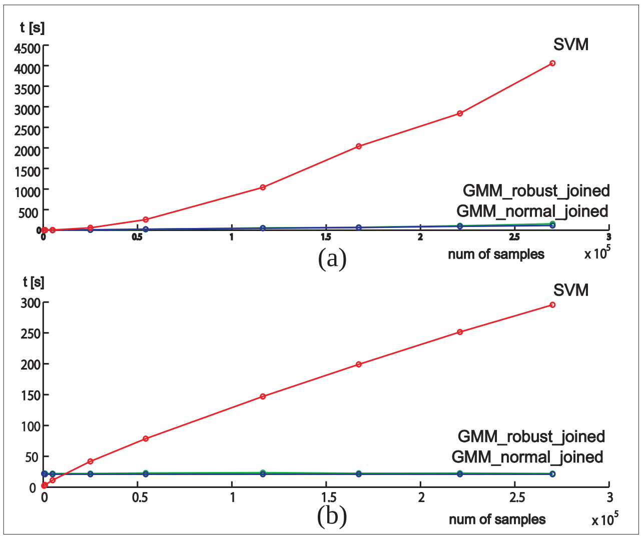


Figure 8: Times needed to learn the game model (a) and segment an entire halftime of a handball game (b).

Additionally, by studying Figure 7, it can be observed that the results do not change significantly in case when smaller amount of data (e.g. 500 samples) is used to build the game model. Although in general the game model needs to be built only once, this information suggests that a large training database is not needed to obtain a good segmentation model. This is especially important in cases when a method would be applied to new sports or a different analysis domain.

By studying Figure 8, it can be observed that if SVM is used for classification, the training and testing times increase from a few seconds up to a few hours in the training stage and from a few second up to a few minutes in the testing stage. On the other hand, if a GMM model is used, the learning time only increases from a few second up to a minute and the testing time does not change. The reason for this is that the hyperplane of the SVM model is encoded in the form of the support vectors. With the increase of the learning samples the number of support vectors that define the hyperplane also increases. Thus with the increased complexity of the learned SVM model the time to classify individual test sample also increases. On the other hand, in the case of the GMM the testing time does not increase since the complexity of the model changes with the number of GMM component and not with the amount of the training data.

Although our approach was principally developed for the off-line segmentation, one can observe that average time to process a single frame of the game is approximately 0.39 ms (21.09 seconds for on average 54000 samples). This would suggest that the described approach could also be used for the *near realtime* segmentation since the whole procedure would only have to be delayed for a half of the kernel width. Moreover, in the case when only the classification step would be used, the method could also be used for real-time segmentation.

However there is also a downside of the GMM model in comparison to the SVM. The main drawback of the GMM is the minimum amount of data that is needed to build the model. In the case of the SVM this number can be very small (less then 50 samples). Although the obtained result would be very poor, the method would still work. However, this is not the case when GMM is used. The main reason for this is that the GMM is a statistical model and if the number of training samples is small the problem becomes numerically ill-posed.

Given the obtained results one can conclude that by using either of the two classifiers, it is possible to segment the multi-player games reasonably well since in both cases the obtained classification rate is near or over 90 %. The GMM model with a robust joined state vector is more suitable for segmentation since it produces better results, is less sensitive to the amount of the training data and is less computationally demanding.

4 Conclusion and future work

A method for automatic trajectory-based group motion segmentation was presented. Our ultimate goal was an automatic segmentation of the sport-related trajectory data into meaningful game phases (offense, defense and time outs). We have shown that by observing the centroid position of the players on the court, it is possible to segment different game sports with reasonable accuracy. We have shown how a robust, team and sport independent feature vector can be computed. Additionally, we have presented a two-step segmentation method which can be applied to different sports. The first step represents the classification of individual time instances using a pre-computed six-component Gaussian mixture model. In the second step the temporal continuity of the game is enforced by smoothing individual game labels using a nonlinear or Gaussian smoothing kernel.

The presented results suggest that the use of Gaussian Mixture Model in the first step of segmentation is significantly more efficient when compared with the Support Vector Machines and that by considering all player on the court the segmentation result can be additionally improved. Moreover, we have shown that even by using a very small amount of training data, it is possible to derive a general game model, which can be used to successfully segment trajectories into the game phases without any tuning.

Even though we only tested the proposed segmentation method on sport data, we believe that the same approach could be applied to other domains. For example in video surveillance this approach could be used to segment scenes where groups of people move in different directions (e.g. streets or subways).

Our future work will be focused on developing new game models for other sports such as soccer, hockey, volleyball or beach volley, where the structure of the player motion is very similar to the motion in basketball or handball. Finally, since the presented segmentation framework can be easily extended to consider new game features, we will try to improve it by consider new features such as distance between the offensive and defensive players or position of the ball.

Acknowledgement

This research was supported in part by Slovenian Research Agency (ARRS) contracts P2-0232 and L5-6274. Special thanks to prof. Dr. Petra Platen and her team from Ruhr-University Bochum, Germany, for providing the handball data and video annotation support.

References

[1] C. Cedras and M. A. Shah. Motion-based recognition: A survey. *Image and Vision Computing (IVC)*, 13(2):129–155, March 1995.

[2] B. T. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding: CVIU*, 81(3):231–268, 2001.

[3] J. K. Aggarwal and Sangho Park. Human motion: Modeling and recognition of actions and interactions. In *3DPVT '04: Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium on (3DPVT'04)*, pages 640–647, 2004.

[4] B. T. Moeslund, A. Hilton, and V. Krger. A survey of advances in vision-based human motion capture and analysis. *Special issue on modeling people: Vision-based understanding of a person's shape, appearance, movement, and behaviour, Computer Vision and Image Understanding*, 104(2):90–126, November 2006.

[5] A. Ekin, A.M. Tekalp, and R. Mehrotra. Automatic soccer video analysis and summarization. *Image Processing, IEEE Transactions on*, 12:796–807, July 2003.

[6] J. Assfalg, M. Bertini, C. Colombo, A. Del Bimbo, and W. Nunziati. Semantic annotation of soccer videos: automatic highlights identifcation. *Computer Vision and Image Understanding*, 92:285–305, 2003.

[7] P. Xu, L. Xie, S. Chang, A. Divakaran, A. Vetro, and H. Sun. Algorithms and systems for segmentation and structure analysis in soccer video. In *IEEE International Conference on Multimedia and Expo (ICME), Tokyo, Japan*, pages 184–187, 2001.

[8] C.G.M. Snoek and M. Worring. Multimodal video indexing: A review of the state-of-the-art. *Multimedia Tools and Applications*, 25(1):5–35, 2005.

[9] P. Chang, M. Han, and Y. Gong. Extract highlights from baseball game video with hidden markov models. In *International Conference on Image Processing*, volume 1, pages 609–612, 2002.

[10] M. Baillie and J.M. Jose. An audio-based sports video segmentation and event detection algorithm. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW04)*, pages 110–120, 27-02 June 2004.

[11] M. Perše, M. Kristan, J. Perš, G. Vučkovič, and S. Kovačič. A template-based multi-player action recognition of the basketball game. In *In: Janez Pers, Derek R. Magee (eds.), Proceedings of the ECCV Workshop on Computer Vision Based Analysis in Sport Environments, Graz, Austria*, pages 71–82, May 2006.

[12] A. F. Bobick. Movement, Activity and Action: The Role of Knowledge in the Perception of Motion. *Royal Society of London Philosophical Transactions Series B*, 352:1257–1265, August 1997.

[13] M. Perše, J. Perš, M. Kristan, and S. Kovačič. Automatic evaluation of organized basketball activity. In *In: Michael Grabner, Helmut Grabner (eds.), Computer Vision Winter Workshop 2007, St. Lambrecht, Austria*, pages 11–18, February 2007.

[14] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[15] W. S. Erdmann. Gathering of kinematic data of sport event by televising the whole pitch and track. In *Proceedings of 10th ISBS symposium, International Society of Biomechanics in Sports, Rome, Italy*, pages 159–162, 1992.

[16] R. A. Redner and H. F. Walker. Mixture densities, maximum likelihood and EM algorithm. *SIAM Review 26*, 195-202, 1984.

[17] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley Series in Probability and Statistics.

John Wiley & Sons, Inc., New York, 1997.

[18] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[19] V. N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, Inc., New York, 1998.

[20] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines (and other kernel-based learning methods)*. Cambridge University Press, 2000.

[21] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[22] C. W. Hsu and C. J. Lin. A comparison of methods for multiclass support vector machines. *IEEE TRANSACTIONS ON NEURAL NETWORKS*, 13(2):415–425, March 2002.

[23] M. Kristan, J. Perš, M. Perše, M. Bon, and S. Kovačič. Multiple interacting targets tracking with application to team sports. In *4th International Symposium on Image and Signal Processing and Analysis ISPA*, pages 322–327, September 2005.

[24] M. Perše, M. Kristan, J. Perš, G. Vučkovič, and S. Kovačič. Physics-based modelling of human motion using kalman filter and collision avoidance algorithm. In *International Symposium on Image and Signal Processing and Analysis, ISPA05, Zagreb, Croatia*, pages 328–333, September 2005.

[25] J. Perš, M. Bon, and S. Kovačič. Errors and mistakes in automated player tracking. In *B. Likar (ed.), Proceedings of the Sixth Computer Vision Winter Workshop CVWW'01, Bled, Slovenia*, number 2536, February 2001.