Pattern Recognition 43 (2010) 1491–1501

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Analysis of multi-agent activity using petri nets

Matej Perše [a,*], Matej Kristan [a], Janez Perš [a], Gašper Mušič [a], Goran Vučkovič [b], Stanislav Kovačič [a]

[a] Faculty of Electrical Engineering, University of Ljubljana, Tržaška 25, SI-1001 Ljubljana, Slovenia
[b] Faculty of Sport, University of Ljubljana, Gortanova 22, SI-1001 Ljubljana, Slovenia

ARTICLE INFO

Article history:
Received 29 January 2009
Received in revised form
30 October 2009
Accepted 11 November 2009

Keywords:
Multi-agent activity analysis
Activity recognition and evaluation
Trajectories
Basketball analysis

ABSTRACT

This paper presents the use of place/transition petri nets (PNs) for the recognition and evaluation of complex multi-agent activities. The PNs were built automatically from the activity templates that are routinely used by experts to encode domain-specific knowledge. The PNs were built in such a way that they encoded the complex temporal relations between the individual activity actions. We extended the original PN formalism to handle the propagation of evidence using net tokens. The evaluation of the spatial and temporal properties of the actions was carried out using trajectory-based action detectors and probabilistic models of the action durations. The presented approach was evaluated using several examples of real basketball activities. The obtained experimental results suggest that this approach can be used to determine the type of activity that a team has performed as well as the stage at which the activity ended.

1. Introduction

Understanding "What is going on in the video" has been one of the main challenges of the video-analysis domain for more than a decade [1]. The main reason for this is the complexity of natural scenes. This is especially true in cases when more than one object of interest or agent is present in the scene, as the analysis should be able to interpret the different temporal, spatial and logical relations among them. The main problem of such an analysis is the substantial temporal and spatial variability in the way different agents perform similar tasks. This, together with the combinatorial complexity of multi-agent activities, is the main reason why several well-known frameworks such as Bayesian networks or hidden Markov models, which have proven to work well in the case of the activity analysis of individuals, experience problems in the case of complex multi-agent systems.

Our research goal is to develop a method for the automatic trajectory-based analysis of multi-agent activities. In particular, we focus on the recognition and evaluation of highly structured activities that usually occur in the sport domain (e.g., organized activities which are practiced in advance and have to be performed according to some predetermined scenario) or video surveillance (e.g., agent motion in highly secured facilities). To perform such analysis we use petri nets (PNs) since they allow the modeling of several sequential and concurrent events and their temporal synchronization.

1.1. Related work

One of the main problems in the activity-recognition domain is the correct interpretation and evaluation of the observed events. Therefore, it is not surprising that an increasing amount of video-analysis research is dedicated to these problems [2–5].

Several different stochastic and deterministic inference methods for addressing the problem of the trajectory-based semantics of human behavior have been proposed over the past decade. The stochastic approaches involve Bayesian networks, dynamic Bayesian networks and their variations [6–9], hidden Markov models [10,11], propagation networks (P-nets) [12,13], Gaussian mixture models [14], and stochastic grammars [15]. For example, Nair and Clarck [16] used hidden Markov models (HMMs) to develop an automated visual surveillance system that detects suspicious human activity in a scene. Johnson and Hogg [14] used Gaussian mixture models to model and synthesize the stochastic behavior of pedestrians. Li and Woodham [17] presented a system for representing and reasoning about selected hockey plays based on trajectory data, augmented with domain-specific knowledge, such as forward/backward skating and puck possession. Intille and Bobick [6] built models of football plays using belief networks and temporal graphs. Shi et al. used propagation networks (P-nets) [12,13] for the representation and recognition of sequential activities that include parallel streams of actions.

One of the major drawbacks of the above methods is the use of very complex models with a large number of parameters. Such models are hard to build since they have to be learnt from a huge amount of training data or demand a lot of manual work which usually has to be done by the domain expert. This is a serious

* Corresponding author. Tel.: +386 1 4768 876; fax: +386 1 4768 130.
E-mail address: matej.perse@fe.uni-lj.si (M. Perše).

0031-3203/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2009.11.011

disadvantage in situations when only a few training examples can be obtained. Another major drawback of these methods is the modeling of concurrent actions, which usually occur in multi-agent activities. In such cases, commonly used approaches, such as HMMs and DBNs [6,7,9], fail to model precisely all these combinations. A common solution to this problem is to use several different sub-models to represent the temporal relations [6] or to model the activity in a hierarchical manner [7]. However, this again increases the complexity of the obtained activity model. Considering these facts, we can conclude that stochastic frameworks are the most suitable for simple activities whose structure is known in advance and can be used when there is enough training data. On the other hand, for more complex, high-level activities, which include many temporal combinations of events, deterministic inference seems to be more appropriate.

Petri nets [18–20] have been previously used for the recognition of highly structured events [1,21,22]. They have proven to be particularly suited to the modeling of sequential and concurrent events and their synchronization, handling multiple scenarios using the same PN model, modeling the hierarchical structure of activities, and modeling the deterministic and stochastic inference of event occurrences.

Castel et al. [1], Ghanem et al. [22] and Lavee et al. [21] used PNs for the recognition and querying of events in surveillance videos. Two different methodologies for scene modeling were proposed which produce different classes of PN models—*object PN* [1] or *plan PN* [22,21]. In the first case, the places represent the states of the objects, the tokens represent the number of objects and the transitions represent the state changes. In the latter case the places represent the states of the activity, the tokens represent its progress and the transitions represent the activity advancement from one state to another.

1.2. Overview of our approach

This paper presents a method for the automatic evaluation of complex activities that involve several agents and many different actions. The main idea behind our approach is that it is possible to automatically build the activity model using the expert knowledge encoded in the activity template. Similar approaches have previously been used in [21,22]; however, that work did not address several key issues which we try to solve in our work:

- First, in the previously proposed methods the PNs were constructed manually. In our work an automatic procedure for building the PNs from activity templates is proposed. Such templates can be used in various areas of computer vision (e.g. video surveillance, traffic monitoring, sports analysis or human–computer interaction) to encode different real-world scenarios.
- Second, a procedure for the automatic learning of logical and temporal relations among actions and a procedure for the evaluation of these relations is developed.
- Third, the basic PN concept is extended to handle the evaluation of activities by using the tokens as the carriers of the information about *the goodness* of the observed activity.

The proposed approach is extensively tested on several real-world examples obtained from the sport domain. In particular, the focus was put on the analysis of a basketball game since, from the analysis standpoint, basketball represents a very challenging multi-agent environment. The main reason for this is the large number of players involved; since in most cases all five players of the same team are involved in the activity. Additionally, the analysis becomes even more challenging when we consider that these activities can be terminated at different stages of their execution, as players change their tactics as soon as they get a good chance to score.

The remainder of the paper is organized as follows: A short introduction to the multi-agent activity structure is given in Section 2. In Section 3, a short overview of the PN framework is given. Next, the procedure for building the PN from an activity template is presented and the learning of the network's temporal parameters from training samples is described. In Section 4 the experimental setup and the obtained evaluation results are presented. Finally, in Section 5 the results of the experiments are discussed and the final conclusions are drawn.

2. The structure of multi-agent activities

An activity is composed of several elementary actions (e.g. in basketball these actions are *player motion, dribbling, passing, shooting, screening, rebounding, team starting formation*, etc.), which have to be executed in a prescribed temporal order [23]. The idea behind our approach is that it is possible to establish the overall score and the current stage of the observed activity by evaluating how well these individual actions have been performed and whether the elements were performed in the correct temporal order.

Fig. 1 shows an example of the template for a simple basketball activity called "double screen". This activity is composed of six actions: four players' *moves* (players' movements along predefined paths) and two *screens* (close contacts of two

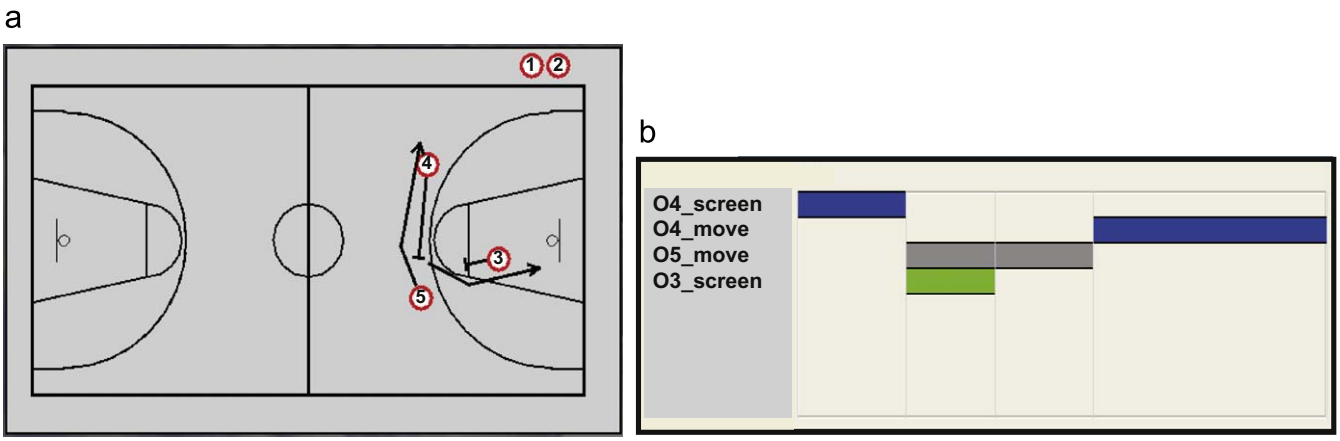

Fig. 1. An example of the spatial (a) and temporal (b) relations of an organized offensive activity called "double screen".

players, where one of the players is standing still). It can be interpreted as follows:

- First, player 4 should move to the position of the screen.
- After player 4 has positioned himself in the position of the screen, player 5 should run next to player 4 and use the screen. At the same time, player 3 should position himself in the position of the screen for player 4.
- Finally, after players 3 and 5 have moved to their new positions, player 4 should move next to player 3 and use the screen.

The above description contains all the information that is relevant for the execution of this basketball activity.

3. Methods

This section presents the basic building blocks to encode, interpret and evaluate the multi-agent activities. First, a short introduction to the PN framework is given. Next, the methods for modeling the temporal relations between the actions are presented and the procedure for building the PN models automatically from the obtained temporal relations is described. Additionally, the methods for modeling the networks' temporal parameters and propagating the activity information along the network are described. Finally, the domain-specific activity detectors that were used in our work to evaluate the spatial properties of individual actions are presented.

3.1. The petri net formalism

Formally, the basic place/transition PN can be described as a five-tuple

$$PN = \{P, T, I, O, M\}, \quad (1)$$

and can be graphically represented by a directed bipartite graph (Fig. 2) which includes two types of nodes: the places P, which are drawn as circles, and the transitions T, which are drawn either as bars or boxes [19].

In Eq. (1) $P = \{p_1, p_2, \ldots, p_n\}$ is a finite set of places, $T = \{t_1, t_2, \ldots, t_m\}$ is a finite set of transitions, $I : (P \times T) \to \mathbb{N}$ is the input arc function which can be represented by the input matrix $\mathbf{I}_{n\times m}$. If there exists an arc with weight k that connects the place p_i to the transition t_j, then $\mathbf{I}(p_i, t_j) = k$, otherwise $\mathbf{I}(p_i, t_j) = 0$. $O : (P \times T) \to \mathbb{N}$ is the output arc function, which can be represented by the output matrix $\mathbf{O}_{n\times m}$. If there exists an arc with weight w that connects the transition t_j to the place p_k, then $\mathbf{O}(t_j, p_k) = w$, otherwise $\mathbf{O}(t_j, p_k) = 0$. $M : P \to \mathbb{N}$ is the current marking of the net and can be represented as a vector $\mathbf{M}_{1\times n}$. M_0 is the initial marking, which denotes the initial state of the net.

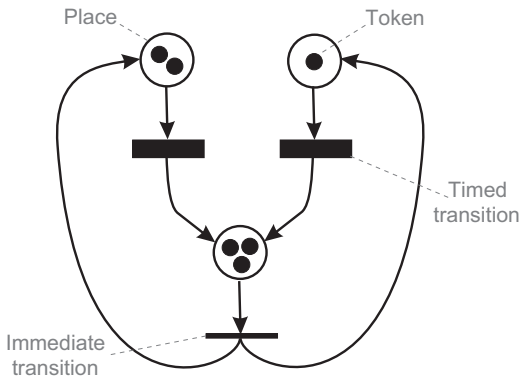

Fig. 2. An example of a petri net [20].

The functions I and O define the weights of the directed arcs. Let the $\bullet t_j \subseteq P$ denote the set of places that are the inputs to the transitions $t_j \in T$. Then, the transition t_j is enabled by a given marking if, and only if, $\mathbf{M}(p_i) >= \mathbf{I}(p_i, t_j)$, $\forall p_i \in \bullet t_j$. An enabled transition can fire and, as a result, remove the tokens (black dots in Fig. 2) from the input places and create tokens in the output places. In this case the new marking $\mathbf{M}_t$ of the net is calculated as

$$\mathbf{M}_t = \mathbf{M}_{t\text{-}1} + (\mathbf{O}\text{-}\mathbf{I}) \cdot \mathbf{E}_t, \quad (2)$$

where $\mathbf{M}_{t\text{-}1}$ is the old marking and $\mathbf{E}_t$ is the vector of the transitions that fired. For further details readers are referred to [19,20].

3.2. Temporal relations

Although the activity presented in Fig. 1 is a relatively simple one, it contains several interdependent actions that should be performed in a specific spatial configuration, either concurrently or in a specific temporal order. These actions can be divided into two groups. The first group contains actions which are performed by a single player (e.g. *player motion, shooting*). The second group consists of a set of actions that are performed by two or more players (e.g. *screening* or *starting player formation*). However, to keep the structure of the templates reasonably general and simple, we describe both groups of actions in a form of single player primitives or ball primitives (e.g. a motion of individual players). To obtain the temporal profiles of multi-player actions (e.g. screens), we perform simulation of those primitives, and detect the time intervals of multi-player as well as single-player actions using sport-specific detectors in a similar manner as we detect those actions from the actual trajectory data. This way we obtain the *activity timeline* (Fig. 3) that defines the actual time intervals in which the actions should occur. Note that this process may split one multi-player action into more actions. For example, a *screen* action is encoded as the *motion* of the player that is making the screen and the actual *screen*.

By observing the starting and ending times of the actions, we can define whether the action has to be executed *before*, *within* or *simultaneously* in accordance with the other actions from the activity.

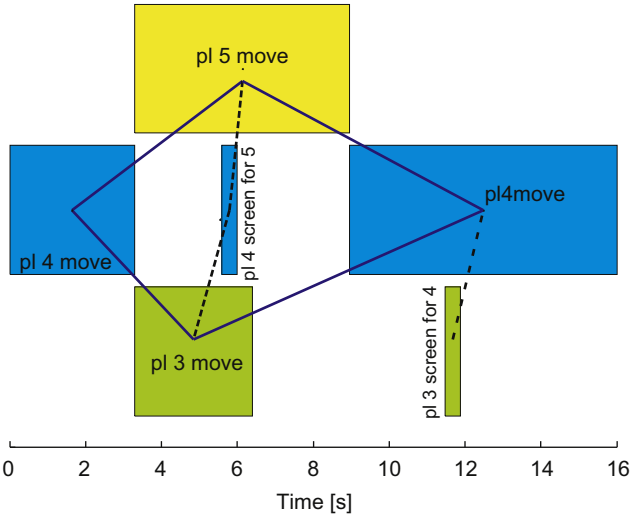

Fig. 3. The timeline for the *double screen* activity. The lines represent the learned temporal relations. Full lines represent relations *before* and dashed lines represent relations *within*.

3.3. Building the petri net model

Our automatic procedure for building PN models from activity templates consists of two steps:

- First, the *action chains* [21], which represent individual actions are constructed.
- Next, the obtained temporal constraints are integrated into the PN activity model. In this way the action chains are linked together by using the knowledge about the temporal relations between elements.

Action chains represent the basic building blocks of the PN and are used to model the individual actions. Following the previously developed ontologies proposed by Ghanem et al. [22] and Lavee et al. [21], we can encode an action as a three-node (Fig. 4a) or a five-node chain (Fig. 4b).

In the first case (Fig. 4a) the actions are encoded as instantaneous events where only the information about the execution or non-execution of the action is obtained. On the other hand, Lavee et al. [21] also encode the duration of the action (Fig. 4b). In this case the starting and the ending points of each action have to be observed. In our case this means that the starting and ending points of relatively short actions have to be observed. From the practical standpoint this is a major drawback, since the analyzed data as well as the action detectors include some degree of uncertainty [24]. Therefore, the obtained starting and ending points may vary significantly from one activity to another. As a consequence, the five-node action chains are not very suitable for modeling short actions and, as a preliminary study suggested, they perform worse in comparison with the three-node chains. For this reason we model each action from the activity timeline as instantaneous time fragments, represented as a three-node chain where:

- The staring node (*precondition place*) represents the preconditions that have to be met in order for the observed action to begin.
- The final node (*action occurred place*) denotes that an action has been observed.
- The middle node—*a timed transition* represents the logical state of an action. It denotes whether an action was or was not observed. When it fires, the token that represents the state of the action moves from the *precondition place* to the *action-occurred place*. The firing of the transition occurs after the logical condition is fulfilled (i.e., the execution of the action) or in the case when the time period allocated for the execution of the action has expired. To test the logical condition we use trajectory-based action detectors (see Section 3.6). If the transition fires due to the expiration of the allocated time, we assume that the action has not been observed.

Once the action chains are created, they are automatically connected into a network using the knowledge about the temporal relations between the actions that were obtained from the timeline (Fig. 3). For this purpose, purely logical, instantaneous *split* and *join* transitions (gray transitions in Fig. 5), are added to the network. The logical transitions can model one or several relations. The number of input and output arcs of the logical transition depends on the type and the number of relations among the actions that the individual logical transition represents. For example, action "pl4 move" is in direct *before* relation to the two other actions—"pl4 move" and "pl5 move". These two relations are modeled as one input and two output arcs to the logical transition that connects the corresponding action chains. Additionally, the second two actions are also in a *within* relation with the screen action "pl4 screen for pl4". For this reason, an extra output arc is added to the previously mentioned logical transition. This way all the action chains are connected such that the final PN model satisfies all the required PN properties [18–20]. That is, they are connected in a way that deadlocks and conflicts between the different action chains are prevented and all the action chains are reachable from the initial marking.

At the end of modeling procedure two additional dummy nodes that represent the start and the end of an activity are added. These dummy nodes are added just in case more activity templates are used for the evaluation at the same time. Using these nodes all the individual activity models can be connected together into a single PN with many parallel single-activity threads.

3.4. Learning the model temporal parameters

Transitions that are part of the action chains can fire for two reasons. The first one is the change of the action state (the

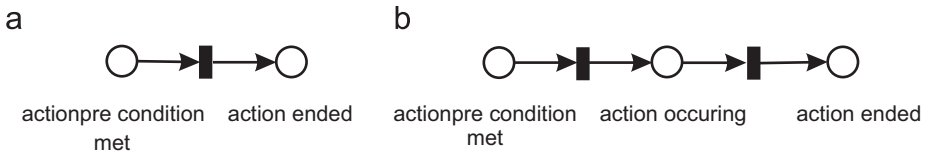

Fig. 4. Different types of action chains used for modeling actions. (a) A three-node action chain. (b) A five-node action chain.

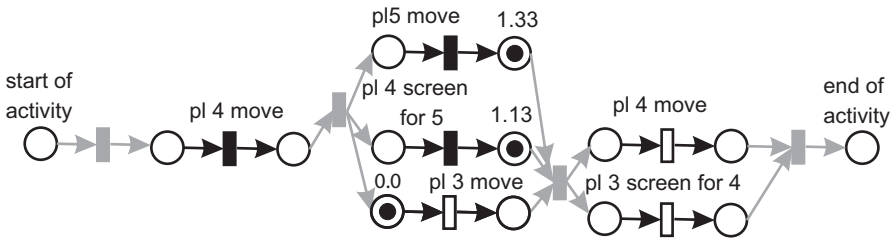

Fig. 5. Petri net model automatically built from the *double screen* template. The black dots inside the places represent tokens that define the current state/marking of the net. The numbers above the tokens represent the accumulated overall score of the particular activity that is analyzed. The black rectangles denote the observed actions, and the white rectangles denote the actions that were not yet observed. The gray elements denote purely logical elements.

observation of the action), and the second is the expiration of the time allocated for its execution. In the second case the time periods for the individual activities have to be known. The simplest solution is to allocate an equal amount of time to each individual action. A better solution is to obtain the overall duration of the observed activity and then divide it in proportion to the relative durations of the actions according to the activity template.

In cases when at least some training samples can be obtained, the actions' temporal durations can be automatically derived from these samples. For this purpose, the training samples can be analyzed using the PN model with the temporal durations set to very high values (e.g., 150 frames). In this way the times when individual transitions are fired can be observed. Then a probabilistic distribution function (pdf) can be fitted onto the obtained data in order to model the temporal distribution and to derive the temporal distribution model of each action (see the examples in Fig. 6). In our case a Gaussian pdf is used to determine the expected action durations (μ) and the upper limits, which are defined as $T^j_{max} = \mu_j + \sigma_j$. Note that other pdfs could be used for the modeling (e.g., the Gamma distribution function); however, due to the small number of training samples and the better spread over all the test data, the Gaussian pdf provided the best results.

3.5. The evaluation scheme

The original PN framework has evolved into several different high-level formalisms, among which the most widely used are colored petri nets and generalized stochastic petri nets. We followed the idea of the colored PN framework [18], where the tokens are used as carriers of information. Usually, tokens carry the information about the properties of objects that are part of the modeled system (e.g., color of a car or the type of installed engine). In our implementation, however, the tokens are used to carry information about the *overall goodness* of the analyzed activity. In terms of the colored petri net terminology a single color set of the type *real* is used [18]. The attached data value is called the *activity score*. The tokens collect the information about the individual action scores, which are defined as the product of the spatial (S_j) and the temporal (T_j) goodness. The activity score is updated every time a transition fires and is calculated as

$$X^i_{new} = X^i_{old} + S_j \cdot T_j, \tag{3}$$

where S_j is the maximum detector response for the action j, T_j is the temporal goodness of the transition j and X^i_{old} is the previous score of the token i that is passing through the transition j. The temporal goodness T_j is obtained from the temporal pdf

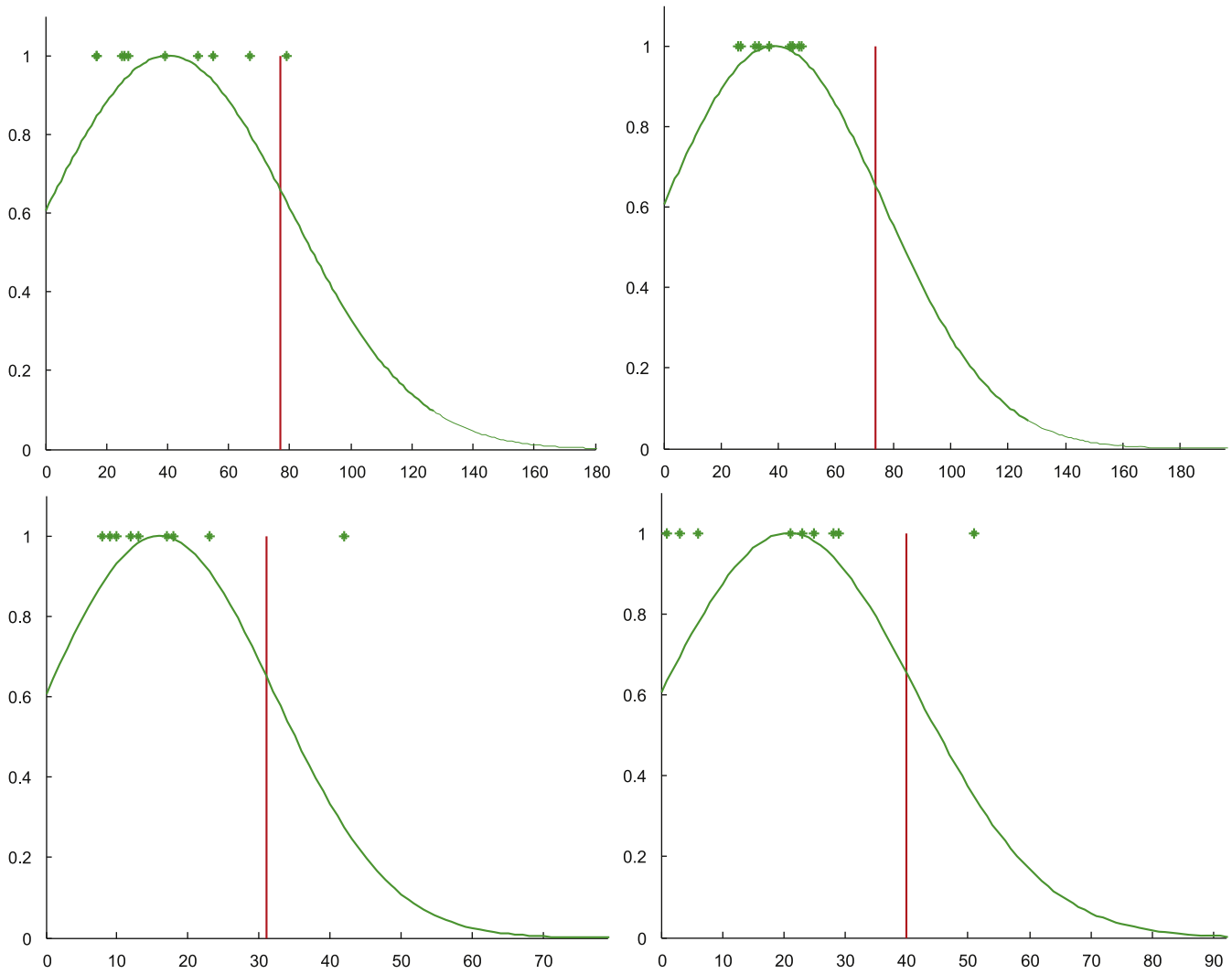

Fig. 6. Gaussian probability density functions for four different actions. The green dots represent the temporal durations of 10 training samples. The red vertical line represents the maximum action duration. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

(Section 3.4) and is defined as

$$T_j = \begin{cases} e^{-\mu_j^2/2\cdot\sigma_j^2}, & t_j < 0, \\ e^{-(\mu_j - t_j)^2/2\cdot\sigma_j^2}, & 0 < t_j < T_{max}^j, \\ 0, & t_j > T_{max}^j, \end{cases} \tag{4}$$

where t_j is the period in which the transition j was enabled, μ_j and σ_j are the mean and standard deviation of the Gaussian pdf for the action j, and the parameter T_{max}^j is the maximum time limit of this action.

The *logical join* and the *logical split* transitions are used to propagate and collect the token information. When n concurrent tokens (X_{old}^i) join into a single token (X_{new}^{i+1}) the new token score is calculated as the sum of the scores of all the incoming tokens

$$X_{new}^{i+1} = \sum_{i=1}^{n} X_{old}^i. \tag{5}$$

In the case of a *logical split* transition, the accumulated score is transferred to one of the newly created tokens. To be able to compare the results from the different templates, the final score obtained at the end node is normalized with the number of action chains in the activity thread.

3.6. Evaluating the spatial properties of the actions

Once the PN model is built, it can be used for an evaluation of the activity performance. In order to do that we have to devise a mechanism that will transform the raw trajectories into meaningful information about the spatial characteristics of the analyzed action. For this purpose we describe different basketball actions (e.g., *dribbling, passing, shooting, player motion, and screening*) using very generic detectors that involve a single player or a collaboration of two players:

- *Screen*. In the basketball literature, the *screen* is defined as a close contact between two players [23], where ideally one player is standing still and the other runs in his near proximity. Thus a certain interaction among two players is more likely to be interpreted as a *screen* if the velocity of the slower player is very low and the distance between the players is small.
  Let d_t be the Euclidian (l_2) distance between the two interacting players and let v_t be the velocity of the slower player. The likelihood function of a *screen* is defined as
  $$\mathcal{L}(screen|d_t, v_t) \triangleq \mathcal{N}(d_t; 0, \sigma_d) \cdot \mathcal{N}(v_t; 0, \sigma_v), \tag{6}$$
  where $\mathcal{N}(\cdot; 0, \sigma)$ is a zero-mean Gaussian function with variance σ^2. The quality of the *screen* is defined as the likelihood ratio
  $$S_{screen} \triangleq \frac{\mathcal{L}(screen|d_t, v_t)}{\mathcal{L}(screen|0, 0)}, \tag{7}$$
  where $\mathcal{L}(screen|d_t, v_t)$ is the likelihood of the *screen* given the current distance and velocity values (d_t, v_t) of the interacting players and $\mathcal{L}(screen|0, 0)$ is the likelihood of an ideal *screen*.
  To obtain the values of the detector parameters, we relied on an extensive study on accuracy of the player tracking [24,25]. Examining the published results, we decided to set the proximity parameter σ_d in Eq. (6) to σ_d = 1 m and the velocity parameter σ_v, which determines the velocity of the player that is "still enough", to $\sigma_v = 0.5$ m/s.
- *Player move*. In the activity template, a player *move* is defined as the exact path that a player should follow. The path is defined by one or more line segments, where each line segment has a starting point (A_i) and an ending point (B_i). Thus, the quality of the player's move is defined as a product of the distance function from the ideal path $\mathcal{N}(d_t; 0, \sigma_d)$ and the path ratio $f_{path}(t)$
  $$S_{move}(t) \triangleq \mathcal{N}(d_t; 0, \sigma_d) \cdot f_{path}(t). \tag{8}$$
  d_t in Eq. (8) denotes the l_2 distance between the player and the closest point on the path (perpendicular distance) and the function $f_{path}(t)$ determines the ratio between the path that the player has covered up to time t and the total length of the path
  $$f_{path}(t) = \frac{\sum_{i=1}^{M} \sum_{j=1}^{t} (\Delta\vec{x_j} \cdot \vec{A_iB_i})}{\sum_{i=1}^{M} \|\vec{A_iB_i}\|}, \tag{9}$$
  where M is the number of line segments, $\|\vec{A_iB_i}\|$ is the length of the path segment i and $\sum_{j=1}^{t}(\Delta\vec{x_j} \cdot \vec{A_iB_i})$ defines the sum of the scalar products of the current player's motion vector $\Delta\vec{x_j}$ and the ideal motion vector of the i th segment $\vec{A_iB_i}$.
- *Ball pass*. A *pass of the ball* is defined as a motion of the ball along a straight line. For this reason it can be regarded in the same way as the motion of a player and therefore the same motion detector that is used to detect the *player move* can be used to detect and evaluate the ball pass.

Fig. 7 illustrates an example of the detector response obtained for a *player motion* action.

4. Experiments and results

To test the performance of our PN-based activity-analysis procedure, several test videos were recorded. To record the basketball plays two cameras, fixed to the ceiling of the sports hall, were used. An image from one of the cameras is shown in Fig. 8.

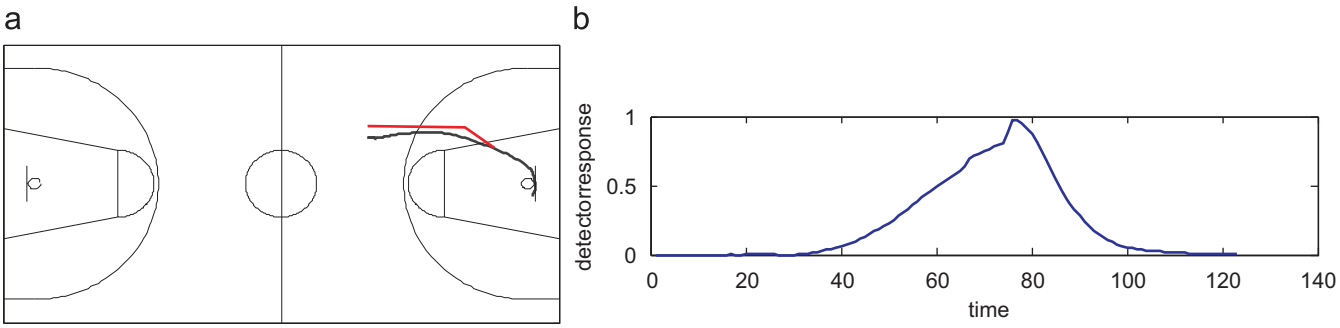

Fig. 7. An example of a detector response for player motion. (a) Player motion on the court. The thick black line represents the player trajectory and the thinner straight red line represents the optimal player path. (b) Detector response for the predefined motion. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The tracking scheme which is based on the color-based particle filter [26,27] that was used to obtain the data is designed so that it does not allow the tracking of the online streaming input videos. The reason for this is that during the tracing several failures can occur due to the frequent player collisions (we estimated that up three tracking mistakes per player can occur in basketball and up to 11 in handball during the entire course of a match [26]). For this reason, an operator supervised the tracking and corrected any 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 to compensate for the radial distortion that is present in the original video data [28]. At the end of the tracking the data were smoothed using a 25-samples-wide symmetric Gaussian filter kernel, which proved to be the most suitable for reducing the tracking jitter and for retaining the measurement accuracy [29,24].

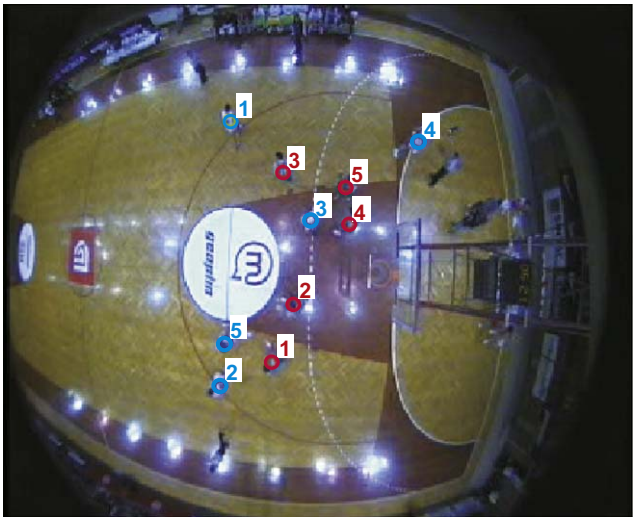

Fig. 8. Operator-supervised tracking in progress.

The evaluation procedure described in the article represents only the last step of a three-step "top-down" analysis process. The entire process involves (a) segmentation, (b) recognition and (c) evaluation. The methods for segmentation and recognition have been published already in [30]. Nevertheless, in this work we have manually segmented the video streams to produce input sequences for PNs. The main reason for that was that the test sequences used were not recorded during an actual basketball game but during a training session, which allowed us to analyze the execution of the activity in several repetitions, under controlled and documented conditions (e.g. with and without defense team). Additionally, manual segmentation allowed us to test the performance of the proposed method in isolation from

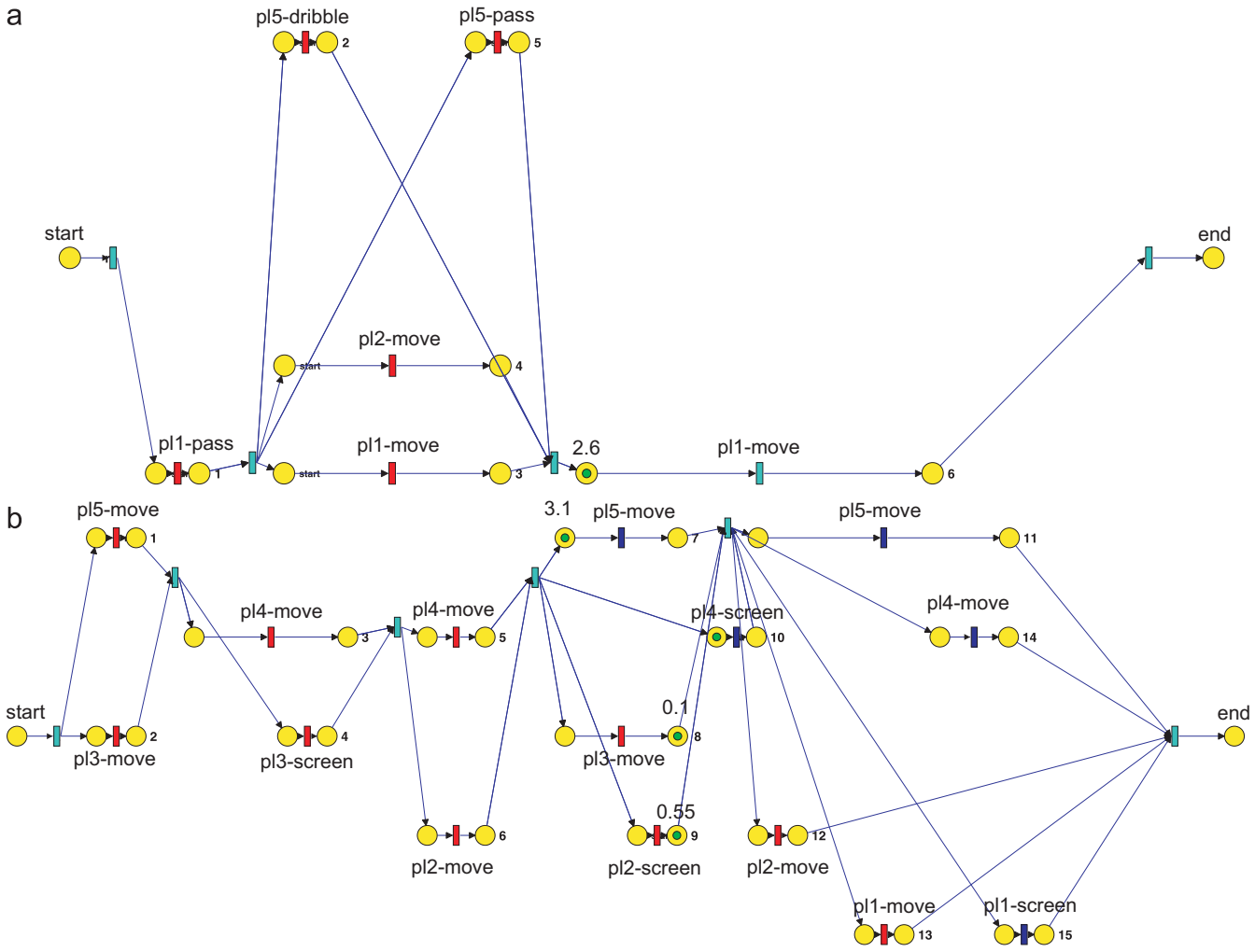

Fig. 9. Two examples of the PNs used in the evaluation procedure. The darker (red) transitions represent the already observed actions. (a) PN model for the "Slovan1" template. (b) PN model for the "flex" template. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Average activity scores when evaluating activities using PNs built from activity templates of different types.

Class of templates	Class of actions						
	52 $\overline{\chi} \pm \sigma$	Flex $\overline{\chi} \pm \sigma$	Motion $\overline{\chi} \pm \sigma$	Slovan1 $\overline{\chi} \pm \sigma$	Slovan2 $\overline{\chi} \pm \sigma$	Slovan3 $\overline{\chi} \pm \sigma$	Slovan4 $\overline{\chi} \pm \sigma$
52	**0.55 ± 0.11**	0.15 ± 0.09	0.17 ± 0.07	0.09 ± 0.03	0.10 ± 0.02	0.18 ± 0.04	0.16 ± 0.04
Flex	0.22 ± 0.06	**0.33 ± 0.09**	0.18 ± 0.09	0.03 ± 0.02	0.05 ± 0.01	0.14 ± 0.04	0.24 ± 0.05
Motion	0.15 ± 0.04	0.20 ± 0.06	**0.40 ± 0.12**	0.06 ± 0.02	0.04 ± 0.02	0.16 ± 0.04	0.13 ± 0.06
Slovan1	0.13 ± 0.05	0.04 ± 0.05	0.03 ± 0.05	**0.78 ± 0.12**	0.51 ± 0.09	0.44 ± 0.17	0.51 ± 0.16
Slovan2	0.05 ± 0.05	0.05 ± 0.07	0.14 ± 0.09	0.51 ± 0.07	**0.57 ± 0.05**	0.50 ± 0.12	0.52 ± 0.13
Slovan3	0.10 ± 0.06	0.13 ± 0.07	0.16 ± 0.07	0.32 ± 0.05	0.32 ± 0.03	**0.60 ± 0.07**	0.55 ± 0.08
Slovan4	0.12 ± 0.03	0.15 ± 0.05	0.18 ± 0.04	0.20 ± 0.03	0.20 ± 0.02	0.48 ± 0.08	**0.60 ± 0.04**

The highest average score is displayed in bold.

other factors, such as tracking or segmentation errors, which would inevitably propagate into the final evaluation results.

In this way, two sets of test data were acquired. The first set was composed of 61 examples of three different basketball offensive plays. There were 20 cases of offense called "52". From these, nine cases were performed without any defense and 11 cases were performed against the defense. The next 22 cases belonged to the "flex" offence: 11 cases were performed without, and 11 against the defense. The last 19 cases belonged to the "motion" offense. From these, 12 were played without, and seven against the defense.

In the second set there were 40 repetitions of a single offense: however, they were performed in such a manner that they ended in different stages of execution. The first 10 examples (Slovan1 activity) ended shortly after the start of the activity and the last 10 (Slovan4 activity) were completed. Each time, five offenses were performed without the defense and the other five were performed against the defense.

Since the evaluation procedure requires that the roles of the players are known (i.e., we have to know which trajectory represents which player role in the template), the players were cast into their respective roles using the methods described in [30]. The experiments were carried out on a modified version of the PN framework presented in [31].

In order to ensure the correctness of the obtained results we always removed the tested sample from the training set that was used to learn the temporal distributions of the individual actions.

The main goal of the experiment was to determine whether the team activity was performed according to the activity template. Additionally, we wanted to establish if it is possible to correctly determine the stage at which the activity ended. To do that, the PN model for all the templates representing the analyzed activities (Slovan1–Slovan4, 52, flex and motion), was built. Two examples of the obtained PN models are presented in Fig. 9.

Table 1 shows the average score for the class of activities belonging to the same type and stage of execution. The videos demonstrating the results of the evaluation procedure are available online at [32].

It is clear from Table 1 that the best scores were obtained in cases when both the type and the stage of the template and the observed activity matched. Furthermore, we can see that even when the stages of the template and the activity mismatched, the obtained score is higher in cases when the type of activity and template matched. This would suggest that the proposed method is very robust even in cases when the activity was concluded too early or too late. Additionally, the results suggest that by using several templates of different lengths it is usually possible to determine the correct stage of the observed activity since the scores are lower in cases when the difference in stages is higher.

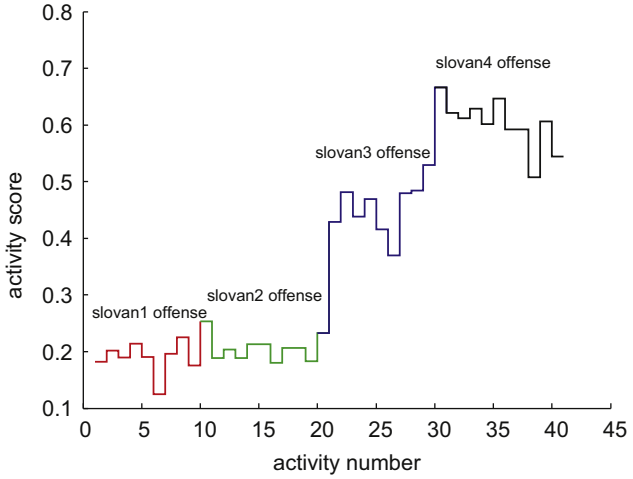

Fig. 10. Evaluation results for individual activities when analyzed using the "Slovan4" template. The text on the graph denotes the type of analyzed activity.

Fig. 10 presents the results of all the activities from the second set of test data when they were analyzed using the longest (Slovan4) template. Here we can see how the results improve as the difference between the stages of the activity and the template becomes smaller.

Fig. 11 shows the evaluation results of the individual activities when different templates are used for the evaluation. Here we can observe that even though the players performed the entire offense they in most cases obtained lower scores when only the first part of the offense was observed. The main reason for this is that when the players perform a longer offense, the spatial and the temporal characteristics of this offense change in comparison with the shorter ones. In addition, we can observe that the scores of the offenses that were performed without any defense are slightly higher than the scores of the offenses that were disrupted by the defense. The main reason for this is that the temporal profile, as well as the path of the players' motion, changes when the defense is present. The reason for this is that the defense disrupts the optimal flow of the activity and as a consequence offensive players also adjust their behavior toward the defensive team. From this, we could conclude that the proposed framework allows some reasonable deviation in the player motion if the value of standard deviation is set correctly (e.g. more than 0.3 m). However, if we wanted to penalize motion deviations more rigorously, we could simply lower the values of the standard deviations. This would additionally differentiate the scores of the two types of activities.

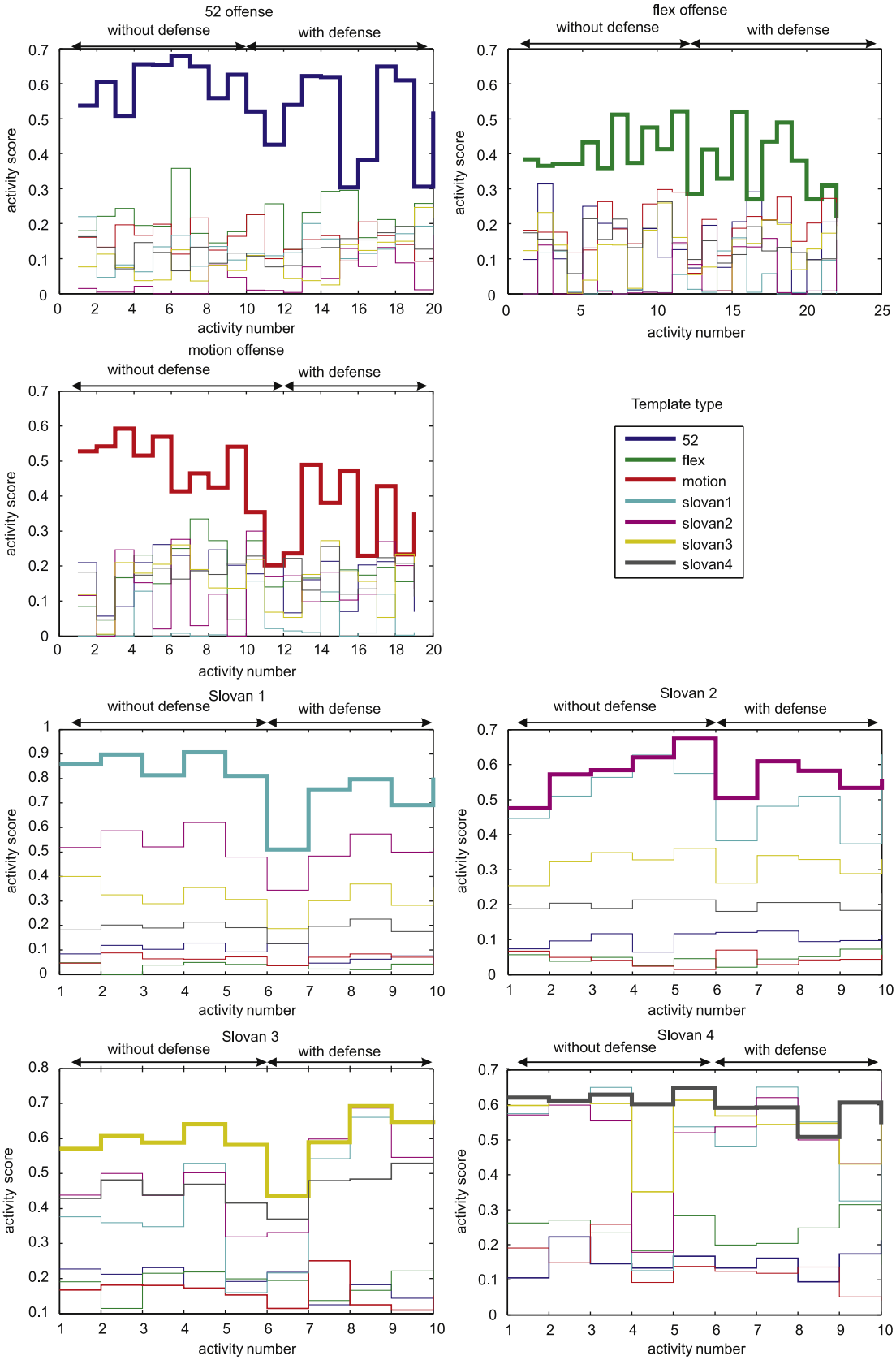

Fig. 11. Evaluation results for individual activities. The names above the graphs denote the type of analyzed activity. The results of the evaluations where the type of activity and the type of template matched are displayed in bold. Individual results can be obtained at [32].

5. Conclusions and future work

An approach to the automatic evaluation of complex, multi-agent activities with petri nets was presented. The PNs were built automatically from the activity templates. The building process was composed of two stages. In the first stage, three-node chains (two places and one transition) were built for each action. In the second stage, the action chains were connected together so that they encode the complex temporal relations between the actions. After the model was built, the temporal durations of the actions were learnt from training data. The strong point of our approach is that it allows learning from only a small amount of training data. In order to evaluate how well the individual actions were performed, trajectory-based action detectors were applied to each transition that represented an action. To obtain the knowledge about the overall activity, a method that allows the propagation of information about activity performance was developed.

Several experiments were performed to evaluate the proposed approach. They were carried out on two sets of trajectory data obtained from two different sets of basketball activities. In total 101 test examples were used for the testing. The obtained experimental results suggest that the presented method can be used to determine the type of activity the team has performed as well as the stage at which the activity ended. The method has proven to be robust, even in cases when the activity ended too early or too late.

Our future work will focus on applying the presented approach to other domains, such as high-security video surveillance, where the proposed approach could be used to observe the unusual behavior of people. Additionally, the evaluation procedure could be refined by taking into account the information about the importance of individual actions, and the allowed deviation in player paths (expressed by the standard deviations in individual action detectors). This information is already contained in the activity templates, however, as it turned out, sport experts had difficulties in choosing an "appropriate" parameter value. Nevertheless, the parameter value is application specific and further studies are needed to enlighten the parameter setting from this perspective as well.

Acknowledgement

This research was supported in part by the Slovenian Research Agency (ARRS), under contracts P2-0095, P2-0214, research grant 1000-05-310094, by the Ministry of Defense of Republic of Slovenia M3-0233 PDR and EU FP7-ICT-215181-IP project CogX.

References

[1] C. Castel, L. Chaudron, C. Tessier, What is going on? A high level interpretation of sequences of images, in: Proceedings of the Workshop on Conceptual Descriptions from Images (ECCV, 96), 1996, pp. 13–27.
[2] C. Cedras, M.A. Shah, Motion-based recognition: a survey, Image and Vision Computing (IVC) 13 (2) (1995) 129–155.
[3] B.T. Moeslund, E. Granum, A survey of computer vision-based human motion capture, Computer Vision and Image Understanding: CVIU 81 (3) (2001) 231–268.
[4] J.K. Aggarwal, S. Park, Human motion: modeling and recognition of actions and interactions, in: 3DPVT '04: Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'04), 2004, pp. 640–647.
[5] B.T. Moeslund, A. Hilton, V. Krüger, 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) (2006) 90–126.
[6] S.S. Intille, A.F. Bobick, Recognizing planned, multiperson action, Computer Vision and Image Understanding: CVIU 81 (3) (2001) 414–445.
[7] M. Perše, J. Perš, M. Kristan, S. Kovačič, Automatic evaluation of organized basketball activity, in: M. Grabner, H. Grabner (Eds.), Computer Vision Winter Workshop 2007, St. Lambrecht, Austria, 2007, pp. 11–18.
[8] J. Muncaster, Y. Ma, Activity recognition using dynamic Bayesian networks with automatic state selection, in: IEEE Workshop on Motion and Video Computing, 2007, WMVC '07, Austin, TX, USA, 2007, 30 pp.
[9] Y. Du, F. Chen, W. Xu, Human interaction representation and recognition through motion decomposition, IEEE Signal Processing Letters 14 (12) (2007) 952–955.
[10] T. Duong, H. Bui, D. Phung, S. Venkatesh, Activity recognition and abnormality detection with the switching hidden semi-Markov model, in: Proceedings of CVPR '05, , 2005, pp. 838–845.
[11] I. Mccowan, D. Gatica-Perez, S. Bengio, G. Lathoud, M. Barnard, D. Zhang, Automatic analysis of multimodal group actions in meetings, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005) 305–317.
[12] Y. Shi, Y. Huang, D. Minnen, A. Bobick, I. Essa, Propagation networks for recognition of partially ordered sequential action, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004), IEEE Computer Society, 2004, pp. 862–870.
[13] Y. Shi, A. Bobick, I. Essa, Learning temporal sequence model from partially labeled data, in: CVPR'06, IEEE Computer Society, 2006.
[14] H. Johnson, D. Hogg, Representation and synthesis of behavior using gaussian mixtures, Image and Vision Computing 20 (2002) 889–894.
[15] A. Bobick, Y. Ivanov, Action recognition using probabilistic parsing, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, , 1998, pp. 196–202.
[16] V. Nair, J. Clark, Automated visual surveillance using hidden Markov models, in: VI02—International Conference on Vision Interface, , 2002, pp. 88–93.
[17] F. Li, R.J. Woodham, Analysis of player actions in selected hockey game situations, in: Proceedings of the 2nd Canadian Conference on Computer and Robot Vision (CRV'05), Canada, 2005, pp. 152–159.
[18] K. Jensen, Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use, vol. 1, Springer, Berlin, 1997.
[19] M. Marsan, G. Balbo, G. Conte, S. Donatelli, G. Franceschinis, Modelling with Generalized Stochastic Petri Nets, Wiley, New York, 1998.
[20] P.J. Haas, Stochastic Petri Nets—Modelling, Stability, Simulation, in: Springer Series in Operations Research and Financial Engineering, 2002.
[21] G. Lavee, A. Borzin, E. Rivlin, M. Rudzsky, Building Petri Nets from Video Event Ontologies, in: ISVC07, 2007, pp. 442–451.
[22] N. Ghanem, D. DeMenthon, D. Doermann, L. Davis, Representation and recognition of events in surveillance video using petri nets, in: EventVideo04, 2004, pp. 112–120.
[23] J. Kresse, R. Jablonski, The Complete Book of Man-To-Man Offense, second ed., Coaches Choice, 2004.
[24] J. Pers, M. Bon, S. Kovacic, Errors and mistakes in automated player tracking, in: Proceedings of the Sixth Computer Vision Winter Workshop, CVWW'01, Bled, Slovenia, 2001, pp. 25–36.
[25] J. Perš, M. Bon, S. Kovačič, M. Šibila, B. Dežman, Observation and analysis of large-scale human motion, Human Movement Science 21 (2) (2002) 295–331.
[26] M. Kristan, J. Perš, M. Perše, S. Kovačič, Closed-world tracking of multiple interacting targets for indoor-sports applications, Computer Vision and Image Understanding 113 (2009) 598–611.
[27] M. Perše, M. Kristan, J. Perš, G. Vučkovič, 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, 2005, pp. 328–333.
[28] J. Perš, M. Kristan, M. Perše, S. Kovačič, ECCV 2008 videos: analysis of player motion in sport games 〈http://eccv2008.inrialpes.fr/videos/videos_newformat/pers-kristan-kovacic.avi〉.
[29] J. Perš, S. Kovačič, Tracking people in sport: Making use of partially controlled environment, in: 9th International Conference on Computer Analysis of Images and Patterns, CAIP 2001, Lecture Notes in Computer Science, vol. 2124, 2001, pp. 374–382.
[30] M. Perše, M. Kristan, J. Perš, G. Vučkovič, S. Kovačič, A trajectory-based analysis of coordinated team activity in a basketball game, Computer Vision and Image Understanding 113 (2009) 612–621.
[31] G. Mušič, T. Löscher, D. Gradišar, An open petri net modelling and analysis environment in Matlab, in: I3M, 2006, pp. 123–128.
[32] Trajectory-based analysis of team sports; petri-net demos: 〈http://vision.fe.uni-lj.si/research/sporta/pn.html〉.

About the Author—MATEJ PERŠE received the Ph.D. degree in Electrical engineering from the University of Ljubljana, Ljubljana, Slovenia 2009, in the field of sport tracking. Currently he works as a senior software developer at Sinergise, laboratory for geographical information systems, Ltd. Slovenija, with interests in computer vision, image processing, human motion analysis.

About the Author—MATEJ KRISTAN received the Ph.D. degree in Electrical engineering from the University of Ljubljana, Ljubljana, Slovenia 2008. Currently he works as a researcher at Machine Vision Laboratory, Faculty of Electrical Engineering and at Visual Cognitive Systems Laboratory, Faculty of Computer and Information Science. His research interests focus on statistical pattern recognition, modelling and estimation, tracking, recognition and visual inspection.

About the Author—JANEZ PERŠ received the Ph.D. degree in Electrical engineering from the University of Ljubljana, Ljubljana, Slovenia 2004. Currently he works as a researcher at Machine Vision Laboratory, Faculty of Electrical Engineering, with interests in image sequence processing and analysis, object tracking and human motion analysis.

About the Author—GAŠPER MUŠIČ received B.Sc., M.Sc. and Ph.D. degrees in electrical engineering from the University of Ljubljana, Slovenia in 1992, 1995, and 1998, respectively. He is Associate Professor at the Faculty of Electrical Engineering, University of Ljubljana. His research work is focused on the area of control systems technology and the area of modelling and simulation of discrete-event and hybrid dynamical systems.

About the Author—GORAN VUČKOVIC received the Ph.D. degree from the Faculty of sport at the University of Ljubljana, Ljubljana, Slovenia 2005. Currently he works as a assistant professor in the Basketball department, Faculty of sport, with interests in motion analysis of players in different sports, sport coaching and performance analysis in sport.

About the Author—STANISLAV KOVAČIČ received the Ph.D. degree in Electrical engineering from the University of Ljubljana, Ljubljana, Slovenia in 1990. Currently he is a vice-dean for Research and professor at the Faculty of Electrical Engineering. His research interests include active vision, image processing and analysis, biomedical and machine vision applications.