Physics-Based Modelling of Human Motion using Kalman Filter and Collision Avoidance Algorithm

Matej Perše, Janez Perš, Matej Kristan,
Stanislav Kovačič
Faculty of Electrical Engineering
University of Ljubljana
Tržaška 25, SI-1000 Ljubljana, Slovenia
{matej.perse}{janez.pers}@fe.uni-lj.si

Goran Vučkovič
Faculty of Sports
University of Ljubljana
Gortanova 22, SI-1000 Ljubljana, Slovenia

Abstract

The paper deals with the problem of computer vision based multi-person motion tracking, which in many cases suffers from lack of discriminating features of observed persons. To solve this problem, a physics based model of human motion is proposed, which includes intertial forces of the persons by the means of the Kalman filter, and the cylindrical envelopes, which produce collision avoiding forces when observed persons come to close proximity. We tested the proposed method on two sequences, one from squash match, and the other from the basketball play and found out that the number of tracker mistakes significantly decreased.

1. Introduction

Vision based people tracking has in last decade become increasingly important technology in several application areas. There are many promising uses of computer vision based people tracking, among which are for example surveillance [3, 6] and sport applications [5, 9]. The main reason for this lies in cheaper and more powerful computer and video equipment, which finally reached the levels where such applications can be commercially attractive.

One of the main problems in computer vision based multi-person human motion tracking is the reliability of the tracking during collisions and occlusions. When the persons being tracked have no obvious discriminating features, the image segmentation phase of the tracker cannot discriminate between two or more persons. As a consequence, the outcome of collision or occlusion of two persons of the similar outfit is mostly random if no additional models of human motion or behavior are present. Similar problems have inspired many researchers to develop various occlusion-proof tracking algorithms, such as [2, 11].

In our work, we are dealing with the tracking of people during the sport matches [9, 10]. The main purpose of tracking people in sport games using computer vision is to acquire data of player's positions on the court. Based on this data, different types of games analysis can be performed. This analysis can assist sport experts to devise better training strategies and game tactics. Since we are dealing with *measurements* of human motion, we depend heavily on the high-quality input video data. Among others, the strict requirement is that, for the sake of accuracy, the cameras should have the bird's eye view of the players. As a side effect, the contacts between players are then visible not as occlusions, but as collisions. Note: by *collision*, we refer to the any contact between the different persons where they visually come so close that the tracking algorithm would get confused.

In the case of "color-blind" tracking algorithms, such as background image subtraction, every collision, regardless of player appearance, is a potential problem with unknown outcome. In the case of advanced tracking algorithms, such as CONDENSATION [7] or simpler color based tracking methods [10], the problem is reduced to collisions of the players of same dress color.

In this paper we present a model of person motion, which is built on two premises:

- that the human body has a certain inertia, which makes the extremely rapid changes in person's motion impossible, and
- that the persons have a tendency towards avoiding head-on collisions and will try to find the path around the other persons blocking their way.

We believe that these two premises are fully valid in tracking of sport players and may have even wider use.

The paper is structured as follows: after a brief discussion regarding image segmentation, we describe how the Kalman filter is used both to simulate the inertia of human body and provide predicted positions for a tracking in the subsequent frame. Next, we describe the spatial model of human body, and the corresponding collision avoidance algorithm. In the last part of the paper, we present four sets of results on two different test video sequences, from squash and basketball, which show that such model of human motion significantly increases the chances of tracker properly choosing which player is which after the critical situations of player collision.

2 Image segmentation

For each video frame, image segmentation is the first step in measuring of players' positions. These initial measurements are fed into the proposed algorithm. We will not discuss image segmentation method in this paper – our framework is general enough to be used with any image segmentation method which is able to provide object position from the given video frame based on the position prediction. The actual image segmentation method used for the experiments is briefly described in the Experiments section.

Let us simply assume that we are dealing with the black box image segmentation method, which requires input image (video frame) and the predicted player position on this frame. As a result, image segmentation method provides the measurements of the positions of the player on the supplied frame.

3 Kalman filter for human motion tracking

A major problem of computer vision based tracking is to precisely define the next player position on the image and thus reducing the search area for a image segmentation algorithm in a subsequent frame. Additionally, good predictions may guide the image segmentation algorithm through "dangerous" situations, where it might jump between the players with similar visual properties. Once the positions are obtained, they usually contain noise, which has an adverse effect on the subsequent trajectory processing. As we will see, the noise has extremely negative influence on our collision avoidance mechanism, which calculates player velocities on-the-fly.

Kalman filter algorithm [4, 12] is an elegant and efficient answer to both problems, as it:

- considerably reduces the noise of the measurement stage and thus provides the more stable data for the the Collision avoidance algorithm
- provides next predicted player position which is then supplied to the image segmentation module to improve chances of finding the player on the next frame.

The Kalman filter algorithm consists of two separate stages, as shown in Fig. 1, the prediction part and the measurement update part. The prediction part is responsible for the projection forward (in time) of the current player's state and error covariance estimate, to obtain the a priori or predicted estimate of the player's state at next time step. The measurement equations are on the other hand responsible for feedback, incorporating the measurement (at next time step) into the predicted player's state estimate in order to obtain a new improved a posteriori estimate from which, given the motion model, a new prediction can be calculated.

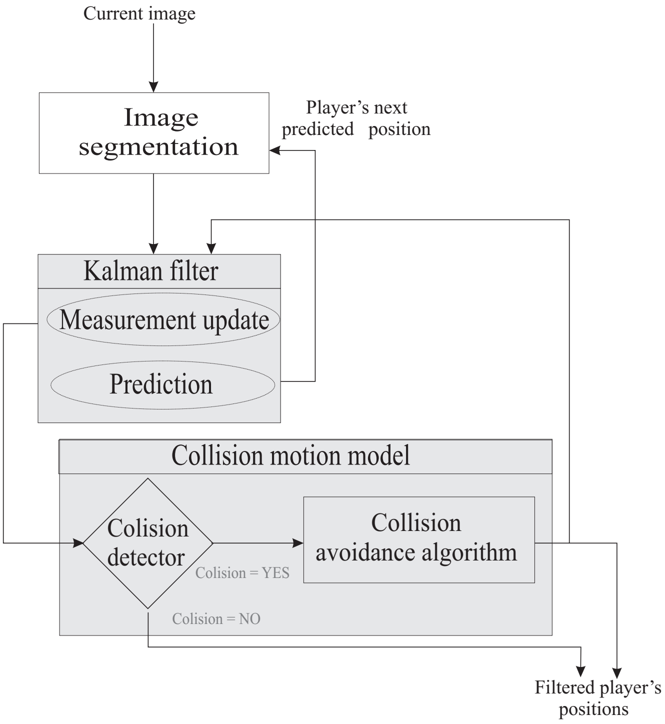

Figure 1. Tracker diagram. Lines and arrows represent the flow of data between parts of the tracker.

In any modelling of human motion, articulated structure of human body poses a difficult problem. The full and detailed body model would be too complex for our use. Kalman filter allows usage of simple motion models (constant velocity or acceleration) and at the same time it allows the knowledge of the modelling error to be built into the model.

In our implementation, we are dealing with the calibrated system. Therefore, we are able to translate image coordinates to the real world coordinates and back on the fly. This way, image segmentation algorithm is working in the image coordinates, yet Kalman filtering and collision avoidance are performed in real world coordinates. This means that all the coefficients and method parameters introduced here have physical meaning and could be set up independently of the actual imaging setup.

3.1 Kalman filter motion model

The primary information gained when tracking players are their positions on the court. That is why the human body movement can be modelled as a motion of a single point, where the selected point represents the gravity center of player's body. Therefore we can write the state space vector of the player's gravity center in the Cartesian coordinate system as:

$$\mathrm{x}_k = \{p_x, v_x, a_x, p_y, v_y, a_y\}^T , \qquad (1)$$

where variables p, v and a represent the current position, velocity and acceleration of gravity center, respectively.

Player's motion model can be defined by two independent equations, the state update equation (2) and the observation equation, (3). The first one describes the state interdependence of two consecutive time steps:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{w}_k, \tag{2}$$

where $\mathbf{x}_{k+1}$ represents the player state at the next time step, $\mathbf{x}_k$ is the player state in the current time step, the matrix $\mathbf{A}$ is the state transition matrix and $\mathrm{w_k}$ is system noise, which is assumed to be white and gaussian. On the other hand, the observation equation (3) gives the connection between the actual position measurements and the state space vector:

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k, \tag{3}$$

where $\mathbf{H}_k$ stands for the observation matrix, $\mathbf{x}_k$ is again the current player state and $\mathbf{v}_k$ is the measurement noise matrix.

Experiments showed that the state transition matrix $\mathbf{A}_k$ can be defined based on the assumption of constant player's acceleration between two consecutive measurements, and that actual change in players' acceleration can indeed be modelled as a Gaussian white noise $\mathrm{w_k}$. Therefore, we defined the state transition matrix as:

$$\mathbf{A}_k = \mathbf{I}_{2\times 2} \otimes \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}, \tag{4}$$

where Δt represents the length of an interval between two consecutive measurements, and can be calculated from the frame rate of the actual video sequence used for tracking. $\otimes$ represents the Kronecker product. The observation matrix $\mathbf{H}_k$ that links the measurements to state vector is defined as:

$$\mathbf{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{5}$$

3.2 System noise covariance

The player's constant acceleration assumption mentioned above is only an approximation of the actual player's dynamics. As we observed from the experiments, the actual player's acceleration is randomly changing all the time. Considering these facts and given the formulation of the system state update equation (2), we are able to statistically determine the properties of the noise and consequently were able to derive the system covariance matrix [1] for Gaussian noise in advance as:

$$\begin{aligned} \mathbf{Q}_k &= E\left[\mathbf{w}_k\mathbf{w}_k^T\right] = & (6)\\ &= q\cdot \mathbf{I}_{2\times 2} \otimes \begin{bmatrix} \frac{1}{20}\Delta t^5 & \frac{1}{8}\Delta t^4 & \frac{1}{6}\Delta t^3 \\ \frac{1}{8}\Delta t^4 & \frac{1}{3}\Delta t^3 & \frac{1}{2}\Delta t^2 \\ \frac{1}{6}\Delta t^3 & \frac{1}{2}\Delta t^2 & \Delta t \end{bmatrix}. & (7) \end{aligned}$$

Detailed derivation of the Equation 7 can be found in [1]. The only free parameter, q, represents the variance of acceleration for player motion, and was determined by analyzing player motion during several squash matches. It is also important to stress that based on our experience, this parameter does not need to be changed for the different types of players, i.e. for players that play either more aggressive or defensive type of game.

3.3 Measurement noise covariance

In Equation (3), another random variable $\mathbf{v}_k$ that represents the system measurement noise can be observed. It represents the observed player's gravity center motion in cases when player is standing still and therefore no motion should be observed. The measurement noise covariance matrix can be derived [1] as:

$$\mathbf{R}_k = E\left[\mathbf{v}_k\mathbf{v}_k^T\right] = \begin{bmatrix} r_{11} & 0 \\ 0 & r_{22} \end{bmatrix}. \tag{8}$$

Measurement noise is a result of several factors, among others the image digitalization and compression artifacts, the noise of the image segmentation phase, and the noise caused by the motion of player's extremities.

4 Collision detection and avoidance

Kalman filter represents good model for a linear human motion and may reduce the possibility of mislabelling the players when they come to close proximity. However, another solution is needed to cope with highly non-linear motion, which is *caused* both by collisions and the human tendency to avoid them.

In real world situations, it is obvious that people will smoothly adapt their motion if they find themselves on a collision path with another person. In some sport games, this observation is perhaps less valid, since some sports require the players to *block* the motion of the opponents. Such sport is for example basketball, where defensive players use their body to block the opponent's way to the basket. On the other hand, the players in the offensive role will still try to avoid the defense players.

However, in some sports, such as squash, players are by the rules of the game *required* to clear the way for the opposite player, or risk a penalty.

4.1 Collision motion model

To address the problem of collision, we devised a simple physical model of human motion when approaching possible collision. The behavior of the players is modelled as non-elastic collision of two cylindrical objects. Since we use bird's eye view of the players and deal with calibrated position data in 2D plane (X and Y coordinates of players on a court plane), the model is essentially reduced to two dimensions.

Fig. 2 shows the situation during the collision of two objects as a projection to a 2D plane (e.g. bird's eye view of two players, or persons in general).

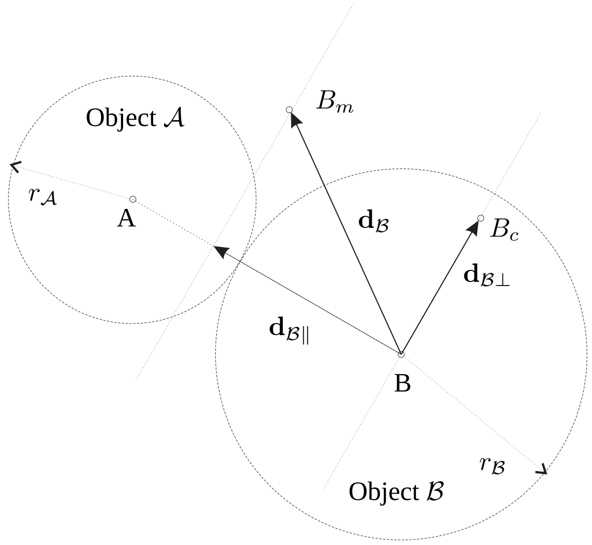

Figure 2. Collision motion model. Detailed explanation in text.

The objects $\mathcal{A}$ and $\mathcal{B}$ are modelled as circles (cylinders in 3D space), with the center points A and B and radiuses $r_{\mathcal{A}}$ and $r_{\mathcal{B}}$, respectively. For the clarity of the presentation, let us assume that the object $\mathcal{A}$ is stationary, and that the object $\mathcal{B}$ is in motion, in the direction of vector $\mathbf{d}_{\mathcal{B}}$, which means that it is on a collision path with object $\mathcal{A}$. Point B_m denotes the measurement of the position of the object $\mathcal{B}$ in a current frame, which is unreliable and may need a correction by the collision avoidance mechanism. Let us also assume, that we have the reliable measurement of position of the point B. $\mathbf{d}_{\mathcal{B}}$ is the motion vector of object $\mathcal{B}$, and $\mathbf{d}_{\mathcal{B}||}$ and $\mathbf{d}_{\mathcal{B}\perp}$ are its orthogonal components. The first one is oriented in the direction of the vector $\overrightarrow{AB}$, which connects the centers of both objects, and the second one is perpendicular to the same vector.

Since the point B_m is actually the measurement of the position of the object $\mathcal{B}$ in this frame (as given for example by the image segmentation method), it is assumed to be wrong, if, as a consequence, the objects $\mathcal{A}$ and $\mathcal{B}$ would overlap.

If the objects would not overlap, the measurement of the position B_m and the object motion vector $\mathbf{d}_{\mathcal{B}}$ are accepted without any further processing. On the other hand, if the objects would overlap, the following condition is true:

$$\mid AB_m \mid < r_{\mathcal{A}} + r_{\mathcal{B}}. \tag{9}$$

In that case, we have to correct initial position measurement B_m in a way that the overlapping does not occur. The simplest way of doing it is by discarding the collision-generating component of the motion vector $\mathbf{d}_{\mathcal{B}||}$. This way, the corrected motion vector is $\mathbf{d}_{\mathcal{B}\perp}$, and the corrected measurement of the center of the object $\mathcal{B}$ is B_c, instead of discarded B_m. Therefore, for each pair from the temporal sequence from the measured input coordinates, $B_m(t) = (x_{\mathcal{B}}(t), y_{\mathcal{B}}(t))$, the algorithm will provide the set of output coordinates, $B'(t) = (x'_{\mathcal{B}}(t), y'_{\mathcal{B}}(t))$, which can be same as input coordinates, or corrected using the procedure described above, when the algorithm has detected a possible collision.

In actual processing of the measurement B_m, the values of A, B, B_m, $r_{\mathcal{A}}$ and $r_{\mathcal{B}}$ are always known, either as constants ($r_{\mathcal{A}}$ and $r_{\mathcal{B}}$) or as the results of previous or actual measurements. Projections $\mathbf{d}_{\mathcal{B}||}$ and $\mathbf{d}_{\mathcal{B}\perp}$, and the corrected position B_c have to be calculated.

Mathematically the projection of vector $\mathbf{b}$ onto vector $\mathbf{a}$ can be calculated as:

$$\mathrm{proj}_{\mathbf{a}}\mathbf{b} = \frac{\mathbf{a} \bullet \mathbf{b}}{\mathbf{a} \bullet \mathbf{a}}\mathbf{a}, \tag{10}$$

where $\bullet$ denotes the scalar product of vectors. Therefore, the projections $\mathbf{d}_{\mathcal{B}||}$ and $\mathbf{d}_{\mathcal{B}\perp}$ can be calculated as follows:

$$\mathbf{d}_{\mathcal{B}||} = \mathrm{proj}_{\overrightarrow{AB}}\mathbf{d}_{\mathcal{B}} \quad = \quad \frac{\overrightarrow{AB} \bullet \mathbf{d}_{\mathcal{B}}}{\overrightarrow{AB} \bullet \overrightarrow{AB}}\overrightarrow{AB} \tag{11}$$

$$\mathbf{d}_{\mathcal{B}}\perp = \mathrm{proj}_{\overrightarrow{AB}_\perp}\mathbf{d}_{\mathcal{B}} \quad = \quad \frac{\overrightarrow{AB}_\perp \bullet \mathbf{d}_{\mathcal{B}}}{\overrightarrow{AB}_\perp \bullet \overrightarrow{AB}_\perp}\overrightarrow{AB}_\perp, \tag{12}$$

where $\overrightarrow{AB}_\perp$ denotes the vector, perpendicular to the vector which connects the object centers, $\overrightarrow{AB}$. This perpendicular vector can be calculated from the vector $\overrightarrow{AB}$ in two dimensional space as follows. Given the two components of the vector $\overrightarrow{AB}$, x and y,

$$\overrightarrow{AB} = \begin{bmatrix} x \\ y \end{bmatrix} \tag{13}$$

the vector $\overrightarrow{AB}_\perp$ can be calculated as:

$$\overrightarrow{AB}_\perp = \begin{bmatrix} -y \\ x \end{bmatrix} \tag{14}$$

4.2 Implementation of collision detection and avoidance mechanism

During the tracking, the measurements of players' positions are continuously obtained by image segmentation methods, coupled with Kalman filter. These measurements are fed into the collision detection and avoidance algorithm, which essentially works as an observer, remembering measurements from the previous step, and internally reconstructing object velocities (that is, motion vectors $\mathbf{d}_{object}$) for all of the objects, as shown in Fig. 3. It is obvious that the collision avoidance algorithms needs the measured positions of all of the objects tracked, in each step t, before the collision avoidance on each object can be performed.

If the collision avoidance algorithm corrects position of a particular player, the corrected position is filtered through same Kalman filter for a second time. This way, both a refined measurement and new prediction of player position are obtained from the collision-corrected position measurement. If collision avoidance algorithm does not perform a correction, this entire step is skipped, and the refined measurement and prediction from the first Kalman step are used.

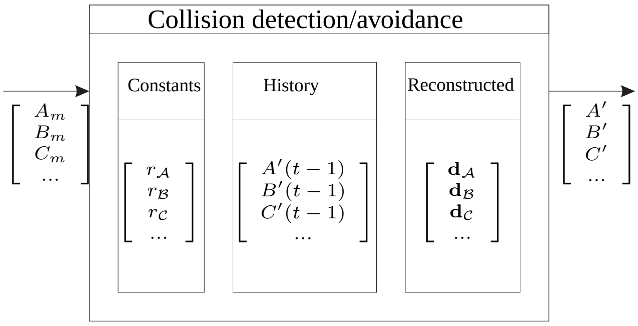
Collision detection/avoidance
Constants
History
Reconstructed
A_m B_m C_m ...
$r_\mathcal{A}$ $r_\mathcal{B}$ $r_\mathcal{C}$ ...
$A'(t-1)$ $B'(t-1)$ $C'(t-1)$ ...
$\mathbf{d}_\mathcal{A}$ $\mathbf{d}_\mathcal{B}$ $\mathbf{d}_\mathcal{C}$ ...
A' B' C' ...

Figure 3. Use of the collision motion model. $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ ... denote different objects.

If such behavior of two objects is simulated using the constant velocity of objects, it can be seen that in the situation which is shown in the Fig. 2, the object $\mathcal{B}$ will slow down and smoothly slide around and along the object $\mathcal{A}$. Smoothness of the motion heavily depends on the sampling rate at which the motion is sampled or simulated. If the sampling rate is low, the object $\mathcal{B}$ will abruptly move right of the object $\mathcal{A}$ and continue its motion.

4.3 Multiple collisions and pile-ups

In contrast to the above discussion, it is usual that more than one object is in motion and that more than two objects are tracked simultaneously. In this case multiple collisions may occur, and the motion correction for some objects may introduce new collisions. This simple collision avoidance algorithm makes no attempts to explicitly solve this scenario. Instead, the detection/correction phase is run in multiple iterations, until either the motion vectors for all objects are set up in a way that there are no collisions anymore, or until certain predefined number of iterations has been reached. This way, the algorithm makes the best effort to solve the possible pile-up of objects, but does not slow down or even block the entire system if the solution is not found quickly.

5 Experiments

Both the Kalman filter and the collision avoidance mechanism were coupled with the image segmentation method to perform the experiments on real world video sequences. We have chosen the CONDENSATION algorithm, similar to [8]and [7], which was modified to take advantage of the static camera setup. The observed players are modelled with adaptive ellipses and the color histogram is used to differentiate between different players. The algorithm does good job of discriminating between players of the different color and the background, but it fails often when the colliding players are of the same color.

Since our work is motivated by the problem from sport tracking domain, we used two sport video sequences to test the performance of the above described methods. Both video sequences have been captured at 25 frames per second and have the resolution of 284×288 pixels. One frame from each video sequence is shown in Fig. 4.

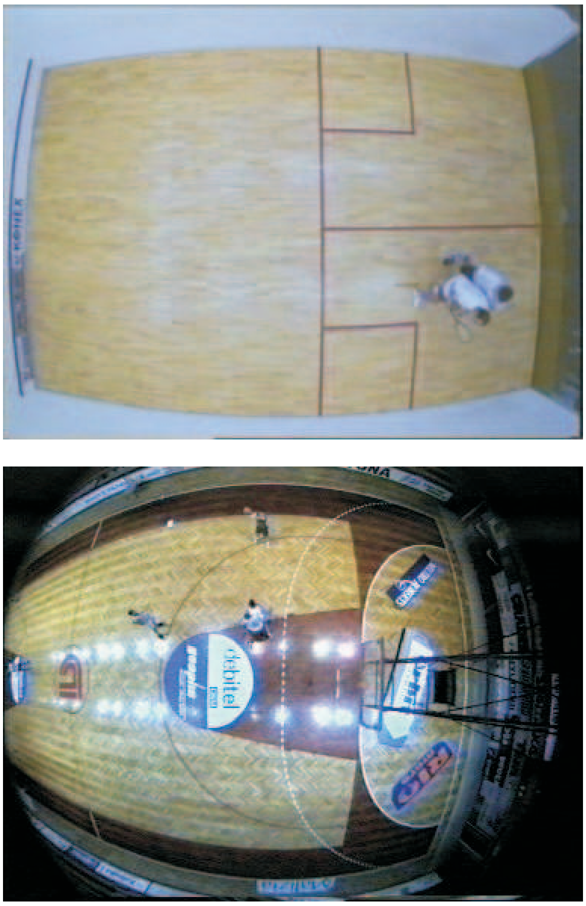

Figure 4. One frame from each of the test video sequences in the moments of players' collision.

The first video shows the bird's eye view of the squash match, and was 10 minutes long. It features two players, both wearing white T-shirts, during the squash game. On many occasions players come to close contact, and after that the Condensation algorithm alone cannot distinguish between them anymore.

The second video shows several players on a basketball court and is two minutes long. We tracked only the two players, which are both wearing white T-shirts, and perform positional dribbling. This video is not a recording of a actual basketball match, and players are essentially wrestling for the better position on the court, which would probably not happen for such a long time in the actual basketball match. The condensation algorithm alone failed completely to distinguish the two players in this sequence (the two positions always converged to one after a just few processed frames), so it is fair to say that this type of motion represents an extremely difficult challenge.

To quantify the improvement, achieved by introducing Kalman filter and the collision avoidance into the scheme, we have done four runs of the tracker for both sequences:

I. The Condensation tracking algorithm alone was used. The Kalman filter was disabled by setting its parameters to such values that it did not modify the input data (very high dynamics). This was done for practical reasons to avoid heavy modification to our code by completely removing it. The collision avoidance algorithm was disconnected.

II. The Kalman filter was enabled with the parameters that have been previously found to work well (on other videos, on other matches). The collision avoidance algorithm was disconnected.

III. The Kalman filter was again disabled, but the collision avoidance algorithm was enabled with radiuses of players set to 20 centimeters for squash and 30 centimeters for basketball.

IV. Both the Kalman filter and the collision avoidance algorithm were enabled with previously described parameters.

For each run, tracking was supervised by operator, which counted the tracker mistakes and stopped tracking when the mistake was made. Then he reinitialized player positions and restarted the tracking. As “mistakes”, we counted the situations where the tracker started tracking another person and stabilized there. If the error in position was only temporary and the situation returned to normal, it was not counted as a mistake.

Table 1 summarizes the results of our experiments. N/A for basketball sequence in the first test run means that the counting of mistakes proved impossible, since the tracker was practically unable to discriminate between the players.

Table 1. Experimental results. The roman numerals denote the test run. The numbers show the number of tracker mistakes.

Test run:	I.	II.	III.	IV.
Squash	45	15	26	9
Basketball dribbling	N/A	38	26	19

6 Conclusion

It can be seen that both the Kalman filter and our collision avoidance algorithm improved reliability of plain condensation tracker, even when used one at a time. However, the best results are achieved, if they are used together. There may be several reasons for this. First, the collision avoidance algorithm works as the observer, and calculates motion vectors internally by differentiation of consecutive player positions. Without the Kalman filtering, the positional data from motion segmentation is noisy, and the motion vectors are very inaccurate. The second probable reason is that the Kalman filter, as a linear model, cannot predict highly nonlinear player motion which is needed to avoid the obstacle. The combination of both the Kalman filter and collision avoidance adds to the tracker both the inertia and the possibility to predict the nonlinear motion when that is needed.

Such algorithm has perhaps even the wider use, although we did not test it further. It could be perhaps useful for tracking persons through the partial occlusions. However, this would require careful tuning of the parameters, since they would probably not correspond to real-world variables (e.g. body size in centimeters as approximate value for the collision radiuses).

7 Acknowledgement

The research, described in this paper has been sponsored by InspireWorks.

References

[1] R. G. Brown and P. Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering: with MATLAB excercises and solutions*. John Wiley & Sons,Inc., 3rd edition, 1997.

[2] R. Cucchiara, C. Grana, G. Tardini, and R. Vezzani. Probabilistic people tracking for occlusion handling. In *Proceedings of IAPR International Conference on Pattern Recognition, ICPR 2004*, pages 132–135, Cambridge, UK, August 23-26 2004.

[3] L. M. Fuentes and S. A. Velastin. People tracking in surveillance applications. In *2nd IEEE International Workshop on Performance Evaluation on Tracking and Surveillance, PETS 2001*, Kauai, Hawaii, USA, 2001.

[4] R. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME, Journal of Basic Engineering*, pages 35–45, 1960.

[5] C. J. Needham and R. D. Boyle. Tracking multiple sports players through occlusion, congestion and scale. In *12th British Machine Vision Conference, BMVC01*, pages 93–102, Manchester, UK, September 2001.

[6] W. Niu, J. Long, and W. Y. F. Human activity detection and recognition for video surveillance. In *Proceedings of the IEEE Multimedia and Expo Conference*, Taipei, Taiwan, 2004.

[7] K. Nummiaro, E. Koller-Meier, and L. Van Gool. An adaptive color-based particle filter. *Image and Vision Computing*, 21(1):99–110, 2002.

[8] P. Perez, J. Vermaak, and A. Blake. Data fusion for visual tracking with particles. 292:495–513, January 2004.

[9] J. Perš, M. Bon, S. Kovačič, M. Šibila, and B. Dežman. Observation and analysis of large-scale human motion. *Human Movement Science*, 21:295–311, 2002.

[10] J. Perš and S. Kovačič. Tracking people in sport: Making the use of partially controlled environment. In W. Skarbek, editor, *Lecture notes in computer science: Proceedings of 9th International Conference on Computer Analysis of Images and Patterns CAIP’2001*, pages 374–382. Springer Verlag, 2001.

[11] A. Senior, H. A., Y. Tian, L. Brown, S. Pankanti, and R. Bolle. Appearance models for occlusion handling. In *2nd IEEE Workshop on Performance Evaluation of Tracking and Surveillance, PETS 2001*, Kauai, Hawaii, USA, December 9 2001.

[12] G. Welch and G. Bishop. An introduction to the kalman filter. Technical Report 95-041, University of North Carolina at Chapel Hill, Department of Computer Science, 2002.