A Measurement Association Model Based on Particle Filters for Automatic Tracking of Occluded Soccer Players

A. Papo[1], A. Averbuch[2], B.Z. Bobrovsky[1]

[1]Department of Electrical Engineering - Systems

School of Electrical Engineering

Faculty of Engineering, Tel Aviv University, Tel Aviv 69978, Israel

[2]School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel

Abstract

Automatic tracking of sport players in soccer video sequences is very important for analyzing soccer games especially when interactive digital TV spreads to the consumer level. This problem has gained interest among researchers in the past few years. Several solutions have been suggested but still no complete solution was offered, which can track players in scenarios where occlusion occurs. A different tracking solution is proposed in this paper, which increases the probability to automatically track players in games with occlusion.

This problem is challenging because players may have similar features (for example, players from the same team wear the same colored uniforms). Moreover, in a situation of occlusion, the players' features and position change, thus making it extremely difficult to automatically distinguish between the players.

The input video sequence in this paper is processed by common methods in order to separate the players' contours from the background in every frame. The players' contours can be originated from a single player, or in case of occlusion between players, can be originated from more than one player. Five measurements are produced from each contour: four measurements of the highest and lowest pixels of the contour in the vertical and horizontal axes and another measurement, which represents the contour's area, which is not covered by the players. The main idea for tracking each player is to estimate the players' states by considering all possible events that associate the players with the contours. In a situation of occlusion, only the contour's measurements, which are best

fitted to the player's state, are assigned to the player. This way each player's state is estimated by the contour's pixels that most likely belong to him.

The suggested solution is based on a coupled sample based joint probabilistic data association method (CSBJPDA). This method, which was recently suggested, combines between particle filter method, which is a Monte Carlo method that estimates non-Gaussian, non-linear state-space models and joint probabilistic data association method (JPDA), which is a multi-target tracking method that evaluates all measurements-to-target associations. Due to the unique model under investigation, some changes in the CSBJPDA method were made.

The above method is tested on real soccer video sequences and the results show that it successfully increases the probability to track and distinguish between players even when it is applied on complex situations, where two or more players from the same team occlude one another and abruptly change their directions.

1 Overview

This paper investigates and analyzes video sequences that are captured from a single fixed (static) camera to track soccer players. The ability to track the players is important in the sport's science industry to obtain information about the players that includes their locations, speeds, accumulated statistics, etc. This information is necessary for planning future game strategies by analyzing the team's performance. The individual player's performance can be assessed as well thus enabling the coach to design an individual program to improve the player's techniques.

Tracking of moving objects has attracted the attention of researchers for many years. This problem requires to estimate the state of an object that rapidly changes its position and its shape at a short time interval. An optimal solution for this problem uses Bayesian filter, which is a recursive solution that uses the Bayes theorem to calculate the posterior density. The problem with this solution is that in most cases it can not be determined analytically. An analytic solution for this problem is the Kalman filter or the extended Kalman filter [13]. Kalman filter produces an optimal solution to the tracking problem but it assumes that the system is linear and the state's object is Gaussian distributed. The extended Kalman filter is a suboptimal solution of Kalman filter, which does not require linearity. In this method, a local linearization of the equations is performed. In some cases, this linearization yields unsatisfying results. Recently, the unscented Kalman filter [4], an approach based on Kalman filter for dealing with the non-linearity, was developed. This method overcomes the problem of non-linearity by defining a set of sampled points which are used to parameterize the mean and covariance for the non-linear system.

A different recent approach is the sequential Monte Carlo approach also known as the condensation algorithm or by the name particle filtering [1]. This approach implements a recursive Bayesian filter by Monte Carlo simulations. The main idea is to approximate the posterior density, which is the density required to estimate the state object by a set of random samples with their associated weights. This method does not require the assumptions of non-linearity and non-Gaussian distribution, thus making it a powerful method for tracking moving objects.

The problem of tracking soccer players can be classified as a multiple target tracking problem since it requires tracking more than one object at the same time. Common multiple tracking methods are the nearest neighbor method (NN) and the joint probabilistic data association filter (JPDAF) [2]. The NN approach is used to determine which measurement belongs to each target by calculating the distance between the measurements and the predicted measurement of each target and minimizing the total measurement-to-target distance. In the JPDAF approach, the measurement-to-target association probabilities are jointly evaluated through all targets to estimate the current state. Both approaches use Kalman filter to estimate the state targets, thus limiting its applicability to linear and Gaussian case. Recently, methods, which combine NN and JPDAF with particle filter [3, 5, 11], were developed. They deal with non-linear and non-Gaussian multi-target tracking problems. Such a method is the coupled sample based JPDA(CSBJPDA) [3]. In this approach, sets of samples are generated for a group of targets. The weight of each set of samples is calculated by summing over all associated hypothesis probabilities.

Tracking soccer players from video sequences can be treated as a visual contour tracking problem since it requires to track the players' contour at each frame. This is opposed to a pointwise tracking problem, where the measurements are just points such as measurements from a laser sensor. The measurements, which assume visual contour tracking, contain the images and the contours of each object. Therefore, models such as active shape models [6], which are flexible shape models, are used to model the dynamics and the measurements of the shape. Such models are often non-linear and non-Gaussian and require methods such as particle filters ([8]).

Tracking soccer players that use active shape and snake models were implemented in [7]. A method, which uses pattern-matching, is proposed in [9]. Another tracking method, which is based on particle filters, is presented in [10]. In this method, each player is fitted into a model and the probability of a group of samples are calculated as a function of the fitness score of each player's model. A particle filter method, which uses an occlusion alarm probability, is suggested in [12]. To prevent particles from changing their locations from one player to another when occlusion occurs, an occlusion alarm probability is applied. This probability is used to repel particles of different players when they get close.

However, in many cases when occlusion between players is present, these methods fail. The reason for this failure is that the soccer players being tracked can not always be fitted into an individual model. First, soccer players from the same team wear the same uniform and therefore it is hard to distinguish between them. Second, when players are occluded, their shapes change as a result of concealing each other and only a contour, which originated from more than one player, can be segmented. Third, when players leave an occlusion, they may have different shapes and also different sizes as a result of a changing distance from the camera. Occlusion alarm probability method also fails in some cases of occlusion. When more than two players get close to each other, the occlusion alarm probability can cause the player's particles to repel to another player and therefore tracking fails (see section 7 for tracking examples that use this method).

The algorithm that tracks soccer players in this paper is based on the CSBJPDA approach. Although the JPDA approach is usually applied on pointwise problems, it is applied in this paper for contour tracking as well. The input is a video sequence, which contains the contours of the players that are separated from the background by common segmentation methods. Each contour provides five different measurements. The first four measurements are the highest and lowest pixels in the horizontal and vertical axes of the contour. These are linear measurements of the state targets. The fifth measurement represents the contour's area which is not covered by the players. This measurement is not linear and therefore uses particle filter. The main idea is to consider all possible events of association between the contours and the players and in each event to assign for a given player only the linear measurements which are most likely originated from the player.

The algorithm presented in this paper is tested on a real soccer video sequence and shows promising results. Automatic tracking of players is successfully accomplished when occlusion is present. Cases when the contour and the speed of the players rapidly change, are also successfully treated by the proposed algorithm.

The paper has the following structure: Section 2 describes the proposed model that achieves automatic tracking. Section 2.1 presents the dynamical model of the state player and section 2.2 presents the measurements model which are produced from the players' contours. Section 3 implements the JPDA method for tracking multiple targets. Since the problem of contour tracking does not fit the regular model that the JPDA assumes, we modify the method to fit our needs. Section 3.2 describes the entire algorithm which is based on the CSBJPDA method.

Section 7 provides experimental results for tracking in a soccer video sequence and section 8 contains concluding remarks. A summary on particle filters is given in appendix A.

2 The tracking model

2.1 Dynamical model

It is common in most of all tracking applications to use a certain model for the dynamics of the object being tracked. In this paper the soccer players are the objects to be tracked. In order to model theirs dynamics, we define a state vector $\vec{x} = [x_x \quad x_y \quad v_x \quad v_y \quad h_x \quad h_y]^T$ for each player, where x_x and x_y are the player's coordinates, v_x and v_y are its velocities in both axes and h_x and h_y are its width and height, respectively. This is illustrated in Fig. 2.1.

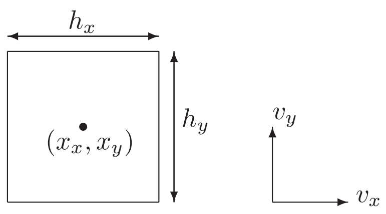


Figure 2.1: Player's state vector

The player's motion model is assumed to have a constant velocity with a zero mean Gaussian distributed acceleration. The player's width and height are assumed to be constant with zero mean Gaussian distributed derivatives given by

$$\begin{aligned} \dot{x}_x(t) &= v_x(t), \quad & \dot{x}_y(t) &= v_y(t) \\ \dot{v}_x(t) &= w_x(t), \quad & \dot{v}_y(t) &= w_y(t) \\ \dot{h}_x(t) &= wh_x(t), \quad & \dot{h}_y(t) &= wh_y(t) \end{aligned} \tag{2.1}$$

where $w_x(t)$, $w_y(t)$, $wh_x(t)$ and $wh_y(t)$ are Gaussian distributed white noise processes given by

$$w_x(t) \sim \mathcal{N}(0, \sigma_x), \quad w_y(t) \sim \mathcal{N}(0, \sigma_y)$$

$$wh_x(t) \sim \mathcal{N}(0, \sigma h_x), \quad wh_y(t) \sim \mathcal{N}(0, \sigma h_y).$$

The automatic tracking of soccer players from a video sequence is a discrete problem where the players' positions are estimated in every frame. Therefore, the model has to be discrete with discrete noises and variables. In order to get a discrete Gaussian distributed noise, it is divided by $\sqrt{\Delta t}$ where Δt is the differential time between two consecutive frames. This way the power density of the noise does not depend on Δt. The continuous derivatives in Eq. 2.1 are replaced by discrete deviations

$$\begin{array}{ll} \frac{x_x(k+1)-x_x(k)}{\Delta t} = v_x(k), & \frac{x_y(k+1)-x_y(k)}{\Delta t} = v_y(k) \\ \frac{v_x(k+1)-v_x(k)}{\Delta t} = \frac{w_x(k)}{\sqrt{\Delta t}}, & \frac{v_y(k+1)-v_y(k)}{\Delta t} = \frac{w_y(k)}{\sqrt{\Delta t}} \\ \frac{h_x(k+1)-h_x(k)}{\Delta t} = \frac{wh_x(k)}{\sqrt{\Delta t}}, & \frac{h_y(k+1)-h_y(k)}{\Delta t} = \frac{wh_y(k)}{\sqrt{\Delta t}} \end{array} \tag{2.2}$$

where $k \in \mathbb{N}$ is the time step.

Let

$$A \triangleq \begin{pmatrix} 1 & 0 & \Delta t & 0 & 0 & 0 \\ 0 & 1 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad B \triangleq \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \Delta t & 0 & 0 & 0 \\ 0 & \Delta t & 0 & 0 \\ 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & \Delta t \end{pmatrix}$$

and

$$\vec{w}(k) \triangleq [w_x(k) \quad w_y(k) \quad wh_x(k) \quad wh_y(k)]^T \sim \mathcal{N}(0, Q(k))$$

where

$$Q(k) = \begin{pmatrix} q_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & q_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & q_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & q_6 \end{pmatrix}.$$

then the dynamical Eq. in 2.2 is

$$\vec{x}(k+1) = A\vec{x}(k) + B\vec{w}(k) \tag{2.3}$$

2.2 Measurements models

2.2.1 Linear measurement: Single target tracking

In order to determine the players' position their observations have to be taken into account. These observations, which are taken from the processed image frames of the video sequence, are the measurements used to estimate the players' positions. The processed image frame contains the contours of the soccer players. Each contour can be originated from either one player only or from a number of players, which occlude each other, or from other objects in the field. When a contour is originated from other objects in the field such as the ball, it is usually a small contour in comparison to the players. Therefore, only contours which have more pixels than a threshold, are taken into consideration in the measurements model. When a contour is originated from one player only, the player's measurements are taken from the contour's pixels. Four pixels are taken as measurements for estimating the player's position: $z_1(x, y)$ - the highest pixel in the vertical axis, $z_2(x, y)$ - the lowest pixel in the vertical axis, $z_3(x, y)$ - the highest pixel in the horizontal axis and $z_4(x, y)$ - the lowest pixel in the horizontal axis. Figure 2.2 illustrates the four measurements of the player's contour.

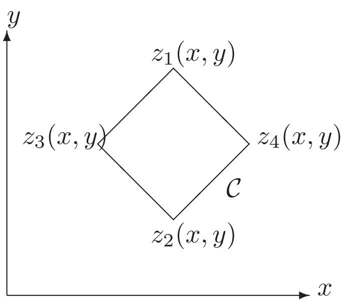


Figure 2.2: $z_1(x, y)$, $z_2(x, y)$, $z_3(x, y)$ and $z_4(x, y)$ are the four measurements of the player which belong to contour $\mathcal{C}$.

When occlusion occurs, it is difficult to estimate the prayers' positions by using pixels inside the contour. Therefore, the pixels, which are being used for estimating the players' position, are the contour's extremist pixels in the x, y axes. This way, when a contour is originated from more than one player, each player can be associated with only the contour's measurements that are, most likely, originated from him.

Each of the four players' measurements are a linear combination of the players' state vector $\vec{x}$ and are assumed to have a Gaussian white noise. The linear equations between the measurements and the player's state vector are given by

$$\begin{array}{ll} z_1^x(k) = x_x(k) + e_1^x(k), & z_1^y(k) = x_y(k) + \frac{1}{2}h_y(k) + e_1^y(k) \\ z_2^x(k) = x_x(k) + e_2^x(k), & z_2^y(k) = x_y(k) - \frac{1}{2}h_y(k) + e_2^y(k) \\ z_3^x(k) = x_x(k) - \frac{1}{2}h_x(k) + e_3^x(k), & z_3^y(k) = x_y(k) + e_3^y(k) \\ z_4^x(k) = x_x(k) + \frac{1}{2}h_x(k) + e_4^x(k), & z_4^y(k) = x_y(k) + e_4^y(k) \end{array} \tag{2.4}$$

where $e_i^x(k), e_i^y(k),\ i = 1, \ldots, 4$ are Gaussian distributed white noises with zero mean and variance $\sigma_i^x(k), \sigma_i^y(k),\ i = 1, \ldots, 4$.

Denote

$$\vec{z}(k) \triangleq [z_1^x(k) \quad z_1^y(k) \quad z_2^x(k) \quad z_2^y(k) \quad z_3^x(k) \quad z_3^y(k) \quad z_4^x(k) \quad z_4^y(k)]^T,$$

$$\vec{e}(k) \triangleq [e_1^x(k) \quad e_1^y(k) \quad e_2^x(k) \quad e_2^y(k) \quad e_3^x(k) \quad e_3^y(k) \quad e_4^x(k) \quad e_4^y(k)]^T$$

and

$$H \triangleq \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \frac{1}{2} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -\frac{1}{2} \\ 1 & 0 & 0 & 0 & -\frac{1}{2} & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Then, the measurement model equation for a single player is given by

$$\vec{z}(k) = H\vec{x}(k) + \vec{e}(k), \quad \vec{e}(k) \sim \mathcal{N}(0, R(k)) \tag{2.5}$$

where

$$R(k) = \begin{pmatrix} r_1^x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & r_1^y & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & r_2^x & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r_2^y & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & r_3^x & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & r_3^y & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r_4^x & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_4^y \end{pmatrix}.$$

2.2.2 Linear measurement: Measurement association

Tracking soccer players becomes difficult when occlusion between players occurs, since more than one player can be assigned to a single contour in the image. In this case, the four measurements of the contour (Eq. 2.4), do not necessarily belong to one player. Each player associated with the contour can be associated with all four measurements as in the case of a single player tracking, or can be associated with a portion of the four measurements, or with no measurements.

Association of measurements-to-players is done by using the players' estimated position and size, conditioned by the measurements of the previous frame. This information is used to estimate which players have the highest and lowest pixels in horizontal and vertical axes.

Let $i = 1, \ldots, N$ be the players associated with contour $\mathcal{C}$. Assume that $I_j(k, \mathcal{C})$, $j = 1, \ldots, 4$, are the players in frame k, which are associated with measurements z_j of contour $\mathcal{C}$. Then

$$I_1(k, \mathcal{C}) = \{i : \max_i [\hat{x}^i_y(k|k-1) + \frac{1}{2}\hat{h}^i_y(k|k-1)]\}$$

$$I_2(k, \mathcal{C}) = \{i : \min_i [\hat{x}^i_y(k|k-1) - \frac{1}{2}\hat{h}^i_y(k|k-1)]\}$$

$$I_3(k, \mathcal{C}) = \{i : \min_i [\hat{x}^i_x(k|k-1) - \frac{1}{2}\hat{h}^i_x(k|k-1)]\}$$

$$I_4(k, \mathcal{C}) = \{i : \max_i [\hat{x}^i_x(k|k-1) + \frac{1}{2}\hat{h}^i_x(k|k-1)]\}.$$

The measurements' equations for players $I_1(k, \mathcal{C}), \ldots, I_4(k, \mathcal{C})$ are

$$\begin{aligned} \vec{z}_1(k) = H_1\vec{x}_{I_1}(k) + \vec{e}_1(k), \quad \vec{z}_2(k) = H_2\vec{x}_{I_2}(k) + \vec{e}_2(k) \\ \vec{z}_3(k) = H_3\vec{x}_{I_3}(k) + \vec{e}_3(k), \quad \vec{z}_4(k) = H_4\vec{x}_{I_4}(k) + \vec{e}_4(k) \end{aligned} \tag{2.6}$$

where

$$\vec{z}_1(k) = [z^x_1(k) \quad z^y_1(k)]^T, \quad \vec{e}_1(k) = [e^x_1(k) \quad e^y_1(k)]^T$$

$$\vec{z}_2(k) = [z^x_2(k) \quad z^y_2(k)]^T, \quad \vec{e}_2(k) = [e^x_2(k) \quad e^y_2(k)]^T$$

$$\vec{z}_3(k) = [z^x_3(k) \quad z^y_3(k)]^T, \quad \vec{e}_3(k) = [e^x_3(k) \quad e^y_3(k)]^T$$

$$\vec{z}_4(k) = [z^x_4(k) \quad z^y_4(k)]^T, \quad \vec{e}_4(k) = [e^x_4(k) \quad e^y_4(k)]^T$$

$$H_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \frac{1}{2} \end{pmatrix}, \quad H_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -\frac{1}{2} \end{pmatrix}$$

$$H_3 = \begin{pmatrix} 1 & 0 & 0 & 0 & -\frac{1}{2} & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad H_4 = \begin{pmatrix} 1 & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

When $i \neq I_j(k, \mathcal{C})$ for all $j = 1, \ldots, 4$, player i is associated with no measurements and therefore tracking is done only by the player's dynamics.

Figure 2.3 illustrates an occlusion between three players which forms the contour $\mathcal{C}$ where $I_1 = 2$, $I_2 = 3$, $I_3 = 3$ and $I_4 = 2$. This means that player 2 has two measurements: z_1 and z_4, player 3 has two measurements: z_2 and z_3 and player 1 has no measurements.

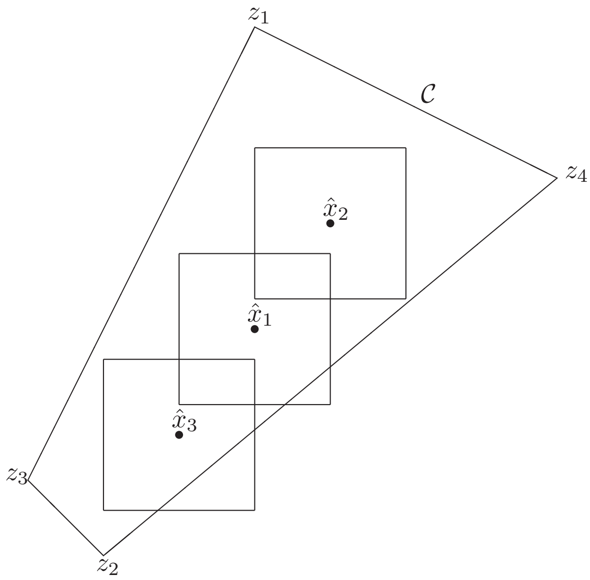


Figure 2.3: A contour created by occlusion between three players x_1, x_2 and x_3 (the camera is placed on the side of the field).

2.2.3 Nonlinear measurement

Another measurement, besides the four measurements which were discussed in section 2.2.1, is taken from the players' contour. Due to the fact that the contour is originated solely from the soccer players, all its pixels have to be originated from at least one player. Therefore, the number of pixels, which do not belong to any player relatively to the total number of pixels in the contour, has to be measured.

Let $X^N = \{x(i)\}_{i=1}^N$ be the states of the players associated with contour $\mathcal{C}$ and $z_5(k, \mathcal{C})$ is the non-linear measurement in frame k associated with contour $\mathcal{C}$. Let $|R_z(X^N, \mathcal{C})|$ be the number of pixels in contour $\mathcal{C}$ that do not belong to any player and $|R_o(\mathcal{C})|$ be the total number of pixels in the contour. Then, we define the function $\Psi(X^N, \mathcal{C})$, which is a function of the contour and the state vectors of players associated with the contours, by

$$\Psi(X^N, \mathcal{C}) \triangleq \frac{|R_z(X^N, \mathcal{C})|}{|R_o(\mathcal{C})|}. \tag{2.7}$$

The likelihood of $z_5(k, \mathcal{C})$ given X^N is

$$p[z_5(k, \mathcal{C})|X^N] = \frac{c_0}{\sqrt{2\pi\sigma_5^2}} exp - \frac{\Psi(X^N, \mathcal{C})^2}{2\sigma_5^2} \tag{2.8}$$

where c_0 is a constant.

3 Data association

3.1 Validation gate

In multiple target tracking applications, it is beneficial to define a validation gate [2] for each target to be tracked. The validation gate (also called validation region) is defined as a region in the measurement space, where the measurement inside it has a true probability to be associated with the target. Any measurement outside the gate has zero probability to be associated with the target and therefore should not be considered in the calculations. Defining a validation gate for each target helps to reduce the number of measurements to be taken into account and therefore reduces the computations.

When two or more validation gates intersect each other, there may be situations in which the same measurement lies inside more than one validation gate. In this case, this measurement is a candidate of more than one target. Validation gates of two targets and their measurements are illustrated in Fig. 3.4. $\hat{z}_1$ and $\hat{z}_2$ are the predicted target positions and z_1, z_2 and z_3 are the measurement positions. As shown, measurement z_3 has to be considered for both $\hat{z}_1$ and $\hat{z}_2$ targets.

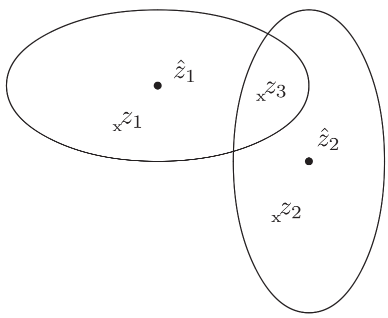


Figure 3.4: Illustration of targets' validation gates

Let $x(k)$ be the target state of frame k and $z(k)$ be the measurement of frame k. The tracking model at time step k is $x(k) = Ax(k-1) + v(k)$, $z(k) = h[x(k)] + w(k)$. The state prediction is

$$\hat{x}(k \mid k-1) \triangleq E[x(k) \mid z(k-1)] \tag{3.9}$$

and the measurement prediction is $\hat{z}(k \mid k-1) \triangleq E[z(k) \mid z(k-1)]$. The innovation between the measurement and the predicted measurement is defined by $v(k) \triangleq z(k) - \hat{z}(k \mid k-1)$, where the innovation covariance is $S(k) \triangleq E[v(k)v(k)^T]$. By assuming that the true measurement is

Gaussian distributed, the probability density function of the measurement innovation is given by $p[v(k)] = \mathcal{N}[v(k); 0, S(k)]$. The probability density of the innovation determines the validation gate, thus, the validation gate is defined as $\mathcal{V}(k, \gamma) \triangleq \{z(k) : v(k)^T S(k)^{-1} v(k) \leq \gamma\}$ where γ is the gate threshold which determines the minimum probability of the innovation to be still inside the region.

Validation gates are usually defined in pointwise problems where each target and each measurement are single points. In this paper, tracking the soccer players is done by the processed images which include the players' contours. Therefore, as shown in section 2.2, each player has four different linear measurements that determine his position and thus the validation gate has to be calculated by a different approach.

The validation gate should determine if a contour is a true measurement of the target player. In order to calculate the validation gate, four innovations for each contour are defined by

$$v_1(k) \triangleq -z_1(k) + \hat{z}_1(k \mid k-1), \quad v_2(k) \triangleq z_2(k) - \hat{z}_2(k \mid k-1)$$

$$v_3(k) \triangleq z_3(k) - \hat{z}_3(k \mid k-1), \quad v_4(k) \triangleq -z_4(k) + \hat{z}_4(k \mid k-1).$$

The four innovation covariances are defined as

$$S_1(k) \triangleq E[v_1(k)v_1(k)^T], \quad S_2(k) \triangleq E[v_2(k)v_2(k)^T]$$

$$S_3(k) \triangleq E[v_3(k)v_3(k)^T], \quad S_4(k) \triangleq E[v_4(k)v_4(k)^T].$$

The four measurements z_1, z_2, z_3 and z_4, that were defined for each contour in Eq. 2.6, are not necessarily measurements of the same target. Each contour in the image can be associated with more than one player. Therefore, the defined innovations are not necessarily real innovations between measurements and the predicted measurements. In order for a target to be associated with a contour, the validation gate defined for each contour has to satisfy

$$\mathcal{V}(k, \gamma) = \{C(k, z_i) : v_i(k)^T S_i(k)^{-1} v_i(k) \leq \gamma\} \tag{3.10}$$

for all innovations $v_i(k)$ that exist $v_i^y(k) > 0$, $i = 1, 2$ and $v_i^x(k) > 0$, $i = 3, 4$. $C(k, z_i)$ is the observed contour.

A target can be either entirely or partly inside the contour or can be completely outside a contour. Innovation i less than zero means that the state i of the target is inside the contour or the target is completely outside the contour as illustrated in Fig. 3.5. When state i of the target is inside the contour, it means that the contour can be a true measurement and therefore should satisfy Eq. 3.10. When the target is completely outside the contour v_j, the innovation of the measurement on the same axis as v_i is greater than zero and is taken into account in the validation gate calculations instead of v_i (it is assumed that the contour is usually larger or equal size of the target and therefore v_j is usually greater than or equal to v_i). Innovation i greater than zero means that the target is partly or completely outside the contour and therefore should be taken into account in the gate calculations.

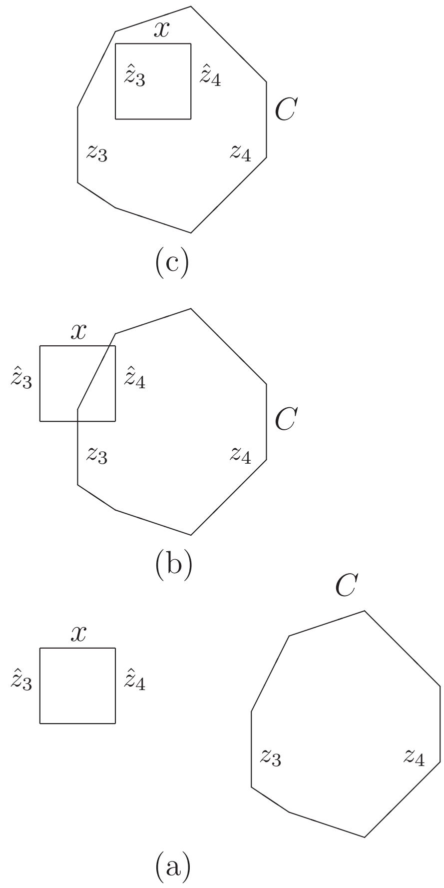


Figure 3.5: Possible states for an estimated player x and its estimated contour C (The camera is placed from the side of the field). (a) The player is completely outside the contour: $v_3^x > 0$, $v_4^x < 0$. (b) The player is partly inside the contour: $v_3^x > 0$, $v_4^x < 0$. (c) The player is completely inside the contour: $v_3^x < 0$, $v_4^x < 0$.

3.1.1 Joint probabilistic data association (JPDA)

As shown in section 2.2, measurements are taken from the players' contours in each frame and are associated with the players. We have to know how to determine the players that belong to each contour.

The problem of associating targets to their measurements is often solved by applying data association methods such as the nearest neighbor method (NN) and joint probabilistic data association method (JPDA) [2]. In the NN method, data association is done by calculating the distance between the measurements and the predicted measurement of each target and then minimizing the total measurement-to-target distance. In the JPDA approach, the measurement to target association probabilities are jointly evaluated through all targets to estimate the current state.

How to determine the players that belong to each contour is in some way different from an ordinary data association problem. In this problem, the targets have to be associated with the contour and not with the contour's measurements. More than one player can be associated with a single contour and players associated with a contour do not necessarily associate with the contour's measurements as shown in section 2.2.

The method being used in this problem is based on the JPDA method. In JPDA, validation gates are used in order to reduce the number of calculations. The players and the players' contours in each frame are divided into joint association groups $R_1, R_2, \ldots, R_M$ where M is the number of groups. Each group contains players which have the same contours inside their validation regions. The group configuration process is described as follows:

1. Let $k = 1$. Let N be the number of players and M be the number of contours. Let S be the group configuration matrix, which is a binary matrix of size $N \times M$ whose elements are defined by

$$S(i,j) \triangleq \begin{cases} 1, & \text{contour } j \text{ lies inside the validation region of player } i \\ 0, & \text{otherwise.} \end{cases} \tag{3.11}$$

2. Let $Q = \{1, \ldots, N\}$

3. Move the element of index k from group Q to a new group R_k.

4. For each element q of group Q and r of group R_k, if exists j that holds satisfies $S(q,j) = S(r,j) = 1$ then move element q from Q to R_k.

5. Go to step 4 until no more players join group R_k.

6. Increase k by one and go to step 3 until $Q = \{\emptyset\}$.

The group configuration process is performed on every frame to determine which players and contours are jointly participate in the algorithm.

For each group in the frame, the joint association events θ are defined as

$$\theta \triangleq \bigcap_{\{t_j\}, j \in R_l} \theta_{j\{t_j\}} \tag{3.12}$$

where R_l is the group l and $\theta_{j\{t_j\}}$ is the event that associates the set of players $\{t_j\}$ to contour j. The group index l will be omitted from now on for simplicity.

In order for θ to be feasible joint events it should satisfy two rules:

1. Each contour should be associated with at least one player.
2. Players are associated only with contours which are in their validation regions

The joint association event probabilities in frame k are defined as $P\{\theta(k)|Z^k\}$ where Z^k are the measurements of frames 1 to k. By using Bayes formula, the joint association event probability becomes

$$\begin{aligned} P\{\theta(k)|Z^k\} &= P\{\theta(k)|Z(k), m(k), Z^{k-1}\} \\ &= \frac{1}{c} p[Z(k)|\theta(k), m(k), Z^{k-1}] P\{\theta(k)|m(k), Z^{k-1}\} \\ &= \frac{1}{c} p[Z(k)|\theta(k), m(k), Z^{k-1}] P\{\theta(k)|m(k)\} \end{aligned}$$

where $m(k)$ is the number of contours and $Z(k)$ is the measurement in frame k.

The probability $P\{\theta(k)|m(k)\}$ does not depend on $\theta(k)$ due to the fact that there is no clutter model (false measurements) and the probability of detection in this model is one (the probability for a player to have no contour at all is zero). Therefore, the joint association event probability is given by $P\{\theta(k)|Z^k\} = c' p[Z(k)|\theta(k), m(k), Z^{k-1}]$ where c' is a constant.

The likelihood function of the measurements conditioned by the joint association event is $p[Z(k)|\theta(k), m(k), Z^{k-1}] = \prod_{j=1}^{m(k)} p[z^j(k)|\theta_{j\{t_j\}}(k), Z^{k-1}]$ where $z^j(k)$ are the measurements of contour j.

As shown in section 2.2, each contour consists of five different measurements. Therefore, the conditional density of the contour is given by

$$p[z^j(k)|\theta_{j\{t_j\}}(k), Z^{k-1}] = \prod_{n=1}^{5} p[z_n^j(k)|\theta_{j\{t_j\}}(k), Z^{k-1}]$$

where

$$p[z_n^j(k)|\theta_{j\{t_j\}}(k), Z^{k-1}] = \mathcal{N}[z_n^j(k); \hat{z}_n^{\{t_j\}}(k|k-1), S^{\{t_j\}}(k)] \quad n = 1 \ldots 4$$

and $\hat{z}_n^{\{t_j\}}(k|k-1)$ is the n^{th} predicted measurement of players $\{t_j\}$ in contour j with innovation covariance $S^{\{t_j\}}(k)$. The conditional density of $z_5^{\{t_j\}}(k)$, which is the non-linear predicted measurement of players $\{t_j\}$ in contour j, will be discussed in section 3.2.

The probability β_{jl}, that assigns player l to contour j, is now calculated by summing up all the joint events in which player l is assigned to contour j as follows

$$\beta_{jl} \triangleq P\{\theta_{jl}(k)|Z^k\} = \sum_{\theta:\theta_{j\{t_j\}}\in\theta, l\in\{t_j\}} P\{\theta(k)|Z^k\}.$$

3.2 Particle filter implementation

Particle filters [1] (see appendix A) are applied when dealing with non-linear, non-Gaussian estimation problems. Data association methods such as JPDA method have recently been combined with particle filters [3, 5, 11] making it possible to solve non-linear, multi target tracking problems. Since the algorithm proposed in this paper requires solving a multi-target problem with non-linear measurements, it is implemented by particle filters.

3.3 Initial conditions

In order to use particle filter, N samples should be generated for each player's state vector. These samples are produced from the initial state distribution. The initial state distribution is determined by the measurements in the first frame.

The processed first frame contains the players' contours. It is assumed that each contour in the first frame is originated from only one player and therefore the four linear measurements taken from a contour belong (Eq. 2.5) to a single player. These measurements are taken from each contour as explained in section 2.2.

Let $(z_1^x, z_1^y)_j, (z_2^x, z_2^y)_j, (z_3^x, z_3^y)_j$ and $(z_4^x, z_4^y)_j$ be the measurements of contour j. Then, the estimated position of player i is $\hat{x}_1^i(0) = \frac{1}{2}(z_3^x + z_4^x)$, $\hat{x}_2^i(0) = \frac{1}{2}(z_1^y + z_2^y)$. The estimated x and y velocities of player i are zero ($\hat{x}_3^i(0) = \hat{x}_4^i(0) = 0$) and the estimated width and height of player i are $\hat{x}_5^i(0) = z_4^x - z_3^x$, $\hat{x}_6^i(0) = z_1^y - z_2^y$. The initial state distribution of each player is given by

$$p_0^i = \mathcal{N}[\vec{\hat{x}}^i(0), R_0] \tag{3.13}$$

where

$$\vec{\hat{x}}^i(0) = [\hat{x}_1^i(0) \quad \hat{x}_2^i(0) \quad \hat{x}_3^i(0) \quad \hat{x}_4^i(0) \quad \hat{x}_5^i(0) \quad \hat{x}_6^i(0)]$$

and

$$R_0 = \begin{pmatrix} r_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & r_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & r_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & r_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & r_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & r_6 \end{pmatrix}.$$

Let $\{\vec{x}^{i,(n)}(0)\}_{n=1}^{N}$ be the N samples which are drawn from the initial distribution of player i. The total number of samples, which are being generated, are $\{\vec{x}_{1:T}^{(n)}(0)\}_{n=1}^{N}$ where T is the number of players in the first frame that is equal to the number of contours in the this frame.

4 Validation gate

As mentioned in section 3.1, a validation gate should be defined for each player in order to detect the possible contours that can be associated with the players. After determining the validation gate of each player, joint association groups, which contain the players and their contours, are defined as well. These groups determine which targets and measurements are jointly considered in the algorithm.

In the SIR filter (see appendix A), which is the particle filter employed in the paper, an importance sampling distribution is used to produce the samples of the particle filter. This importance sampling distribution is the distribution of the state given the past state $p(x_k|x_{k-1})$, which is drawn from the dynamical model of the state vector. Therefore, each sample of the player's estimated state vector is defined for each player i and a set of particles n as $\vec{x}^{i,(n)}(k) \triangleq A\vec{x}^{i,(n)}(k-1) + B\vec{W}^{i,(n)}(k-1)$ where A and B are the matrices of the dynamical model (Eq. 2.3) and $\vec{W}^{i,(n)}(k-1)$ is the n^{th} sample of the state noise generated from the state noise distribution.

As shown in section 3.1, the validation gate is drawn from the distribution of the measurement's innovation. Due to the fact that the measurement model is a Markov model, the distribution $p[\vec{z}(k)|Z^{k-1}]$ can be replaced by $p[\vec{z}(k)|X^{k-1}]$ that determines the validation gate. Using the samples $\vec{x}^{i,(n)}(k)$ and the linear measurements model and by applying Bayes rule, the estimated density $p[\vec{z}(k)|X^{k-1}]$ is

$$\begin{aligned} p[\vec{z}(k)|X^{k-1}] &= \sum_{n=1}^{N} p[\vec{z}(k)|\vec{x}_i(k) = \vec{x}^{i,(n)}, X^{k-1}]p[\vec{x}_i(k) = \vec{x}^{i,(n)}|X^{k-1}] = \\ &= \frac{1}{N}\sum_{n=1}^{N} \mathcal{N}[\vec{v}^{i,(n)}(k); 0, R(k)] \end{aligned}$$

where the innovation vector of the n^{th} sample of player i is given by $\vec{v}^{i,(n)}(k) = \vec{z}(k) - H\vec{x}^{i,(n)}(k)$ and $R(k)$ is the covariance matrix of the measurements noise.

Since each contour contains four points as linear measurements, the validation gate is only calculated for positive innovations.

Let $z_1(k), z_2(k), z_3(k)$ and $z_4(k)$ be the four measurements of contour $C(k)$ and $v_1^{i,(n)}(k), v_2^{i,(n)}(k), v_3^{i,(n)}(k)$ and $v_4^{i,(n)}(k)$ be the n^{th} sample innovation vectors of the measurements. Let R_1, R_2, R_3 and R_4 be the four 2×2 covariance matrices of the measurement noise

$$R_1 = \begin{pmatrix} r_1 & 0 \\ 0 & r_2 \end{pmatrix} \quad , \quad R_2 = \begin{pmatrix} r_3 & 0 \\ 0 & r_4 \end{pmatrix}$$

$$R_3 = \begin{pmatrix} r_5 & 0 \\ 0 & r_6 \end{pmatrix} \quad , \quad R_4 = \begin{pmatrix} r_7 & 0 \\ 0 & r_8 \end{pmatrix}.$$

Then, the particle version of the validation gate for player i, which is give in Eq. 3.10 is

$$\mathcal{V}(k, \gamma) = \{C(k, z_l) : \frac{1}{N}\sum_{n=1}^{N} \mathcal{N}[v_l^{i,(n)}(k); 0, R_l] \leq \gamma\}$$

for all innovations $v_l^{i,(n)}(k)$ that exist $v_l^{y,i,(n)}(k) > 0$, $l = 1, 2$ and $v_l^{x,i,(n)}(k) > 0$, $l = 3, 4$.

After calculating the validation gate of each player, the group configuration process which is described in section 3.1.1, is performed. Each group contains the players and contours which jointly participate in the algorithm.

5 Identifying the targets in each frame

Tracking the players in each frame is performed by the players' estimated states in the previous frame and by the players' contours in the current frame. Since each frame does not contain the entire soccer field, the players inside the scope of the current frame are not necessarily the same as the players inside the scope of the previous frame. Players which are positioned inside the scope of the previous frame can be positioned outside the scope of the current frame and therefore their state should not be estimated anymore. Moreover, players which are positioned outside the scope of the previous frame can be positioned inside the scope of the current frame and therefore should be added to the players which are being tracked.

The identification of the players, which are to be added or omitted from application of the tracking algorithm, is done by the group configuration matrix S which is described in section 3.1.1, Eq. 3.11.

Let $S(k|k-1)$ be the configuration matrix of frame k that was generated before the identification process. Let $S_i(k|k-1),\ i = 1, \ldots, N_{k-1}$, be the i^{th} row of the matrix $S(k|k-1)$ where N_{k-1} is the number of players in the $(k-1)^{th}$ frame and $S^j(k|k-1),\ j = 1, \ldots, M_k$, is the j^{th} column of the matrix $S(k|k-1)$ where M_k is the number of contours in the k^{th} frame. The players, which are omitted from the tracking algorithm, are the players that have no contour in their validation region. Therefore, $PO(k)$, the group omitted players is given by $PO(k) = \{i : S_i(k|k-1) = \vec{0}^T\}$. The players, which are added to the tracking algorithm, are the contours that do not lie inside the validation region of any player. Therefore, $PI(k)$, the group of added players, is given by $PI(k) = \{j : S^j(k|k-1) = \vec{0}\}$. In order to create the configuration matrix S_k, each row in the players group $PO(k)$ is removed from the matrix $S(k|k-1)$. In addition, each contour in $PI(k)$ forms a new row in the configuration matrix S_k, which contains zero elements in all columns except in the contour column where it has the value one.

New players are generated when $PI(k) \neq \emptyset$. Each contour in the group $PI(k)$ belongs to a new player which is drawn from the initial state distribution (see Eq. 3.13).

6 The particle weights

We want to represent the density $p(\vec{x}(k)|Z^k)$ with a set of particles where each particle consists of the state vector $\vec{x}^n(k)$ and its weight $\vec{w}^n(k)$ (see Appendix Eq. 1.15). The weights are calculated by the coupled sample based JPDA (CSBJPDA) method ([3]).

Let $\theta^l(k)$ be the joint association events of group l as described in section 3.1.1, Eq. 3.12[1]. By applying the Bayes rule, the joint association event probability is given by

$$\begin{aligned} P\{\theta(k)|Z^k\} &= P\{\theta(k)|z(k), Z^{k-1}\} = P\{\theta(k)|z(k), X^k\} \\ &= \frac{1}{c} p[Z(k)|\theta(k), X^k] P\{\theta(k)|X^k\}. \end{aligned}$$

[1]The group index l will be omitted from now on for simplicity

Assume that the probability of the joint association events, that is given by the state vectors is, a constant. We assume no clutter model and the probability of detection is one. Then, the joint association event probability is $P\{\theta(k)|Z^k\} = c'p[Z(k)|\theta(k), X^k]$ where c' is a constant.

The likelihood function of the measurements conditioned by the joint association events is given by

$$p[Z(k)|\theta(k), X^k] = \prod_{j=1}^{m_k} p[z^j(k)|\theta_{j\{t_j\}}(k), X^k] = \prod_{j=1}^{m_k}\prod_{l=1}^{5} p[z_l^j(k)|\theta_{j\{t_j\}}(k), X^k]$$

where m_k is the number of contours in frame k, $l = 1, \ldots, 5$ is the measurement number and $\{t_j\}$ is the set of players associated with contour j.

As described in section 3.3, N sets of T samples are generated for the players' state vectors where T is the number of players. For each set of particles n, the joint association event probability is given by

$$P\{\theta(k)|Z^k\}^{(n)} = c' \prod_{j=1}^{m_k} p[z^j(k)|\theta_{j\{t_j\}}(k), X^k]^{(n)} = c' \prod_{j=1}^{m_k}\prod_{l=1}^{5} p[z_l^j(k)|\theta_{j\{t_j\}}(k), X^k]^{(n)}.$$

The probability for each linear measurement $z_l^j(k),\ l = 1, \ldots, 4$, of contour j associated with the set of players $\{t_j\}$ given the players' state vectors is
$p[z_l^j(k)|\theta_{j\{t_j\}}(k), X^k]^{(n)} = \mathcal{N}[v_l^{j,(n)}(k); 0, R_l(k)]$ where $v_l^{j,(n)}(k)$, the innovation of measurement l, is defined as $v_l^{j,(n)}(k) \triangleq z_l^j(k) - H_l x_l^{j,(n)}(k)$ and $x_l^{j,(n)}(k)$ is the state vector of the player associated with measurement $z_l^j(k)$.

The probability of measurement $z_5^j(k)$ given the players' state vectors of the set of particles n (see Eq. 2.8) is

$$p[z_5^j(k)|\theta_{j\{t_j\}}(k), X^k]^{(n)} = \frac{c_0}{\sqrt{2\pi\sigma_5^2}} exp - \frac{\Psi[\mathcal{C}_j, \theta_{j\{t_j\}}(k), x_l^{j,(n)}(k)]^2}{2\sigma_5^2}$$

where $\mathcal{C}_j$ is the contour j and $\Psi(\mathcal{C}_j, \theta_{j\{t_j\}}(k), X^k)$ which is a function of the contour and the state vectors of players associated with the contours, is defined by Eq. 2.7 as

$$\Psi[\mathcal{C}_j, \theta_{j\{t_j\}}(k), x_l^{j,(n)}(k)] \triangleq \frac{|R_z[\mathcal{C}_j, x_l^{j,(n)}(k)]|}{|R_o[\mathcal{C}_j]|}$$

where $|R_z[\mathcal{C}_j, x_l^{j,(n)}(k)]|$ is the number of pixels in the contour $\mathcal{C}_j$ which do not belong to any player and $|R_o[\mathcal{C}_j]|$ is the total number of pixels in the contour.

The samples' weights are now calculated by summing over the joint association event probabilities of all the possible events θ

$$w^{(n)}(k) = \frac{1}{\alpha} \sum_{\theta} P\{\theta(k)|Z^k\}^{(n)}$$

where a possible event occurs when players are assigned to contours in their validation gate and each contour is assigned to at least one player. α is the normalization factor $\alpha = \sum_{n=1}^{N} w^{(n)}(k)$.

After the particle weights are computed, the state vector of each player is calculated by averaging over all the state vector samples by

$$\hat{x}^i(k) = \sum_{n=1}^{N} w^{(n)}(k) x^{i,(n)}(k). \tag{6.14}$$

6.1 Resampling

A common problem in particle filters is that most of the particles' weights except few become zero after few iterations. In order to prevent it, a systematic resampling [1] is used. The systematic resampling is done by generating a new set of particles with equal weights $\frac{1}{N}$ from the particles and their weights. This way the particles with small weights are removed. The algorithm for the systematic resampling is described in appendix A.

7 Experimental Results

The tracking method, which that is described in this paper, was tested on a real soccer video sequence. The input raw data properties of the soccer video are:

Duration:	26 seconds
Number of frames:	383
Frame rate:	15 frames/sec
Resolution:	320 $\times$ 240 pixels

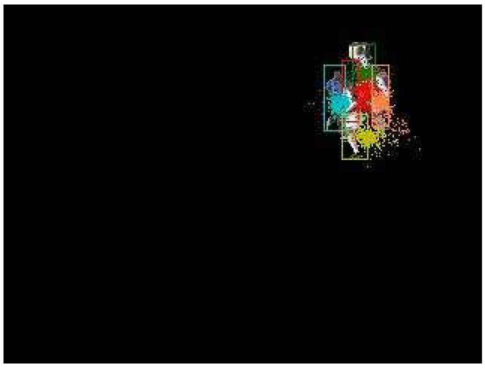
Figures 7.6, 7.7 and 7.8 demonstrate a tracking of soccer players when occlusion is present. Several processed frames from the video sequence are displayed. In each frame a rectangle in a different color is drawn from the estimated state of each player. The colored dots drawn for each player are its samples. Each player is sampled by 1000 points.

The example in Fig. 7.6 displays the tracking results of an occlusion between five players. In frame 15 there are four contours of four different players. In frame 20 there are five contours and therefore a new player is entered the scene. The five players in frame 30 are merged into one contour. All the players except the player with the red rectangle are associated with one of the four linear measurements of the contour. This player is not associated with any linear measurement and therefore the main contribution to its estimated state comes from its dynamical model (Eq. 2.3). Frame 45 demonstrates that the players are successfully tracked after they emerge from a state of occlusion. The fact that, in this example the players motion is homogenous, helps the tracking algorithm to succeed.

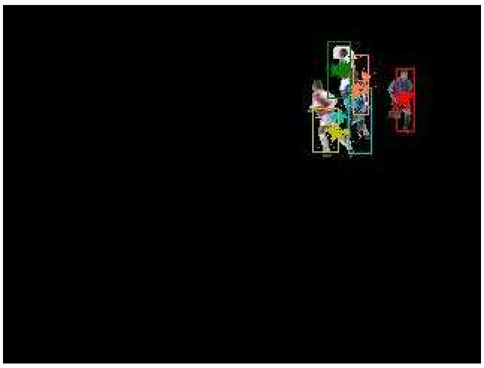
(a) frame 15

(b) frame 20

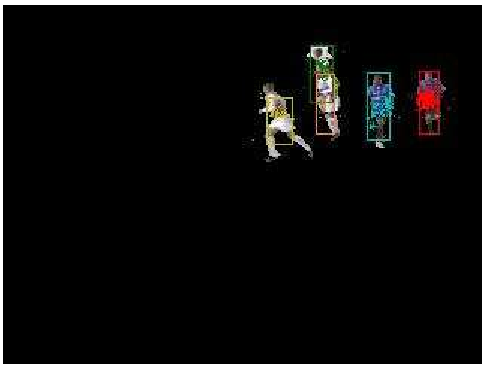
(c) frame 29



(d) frame 36



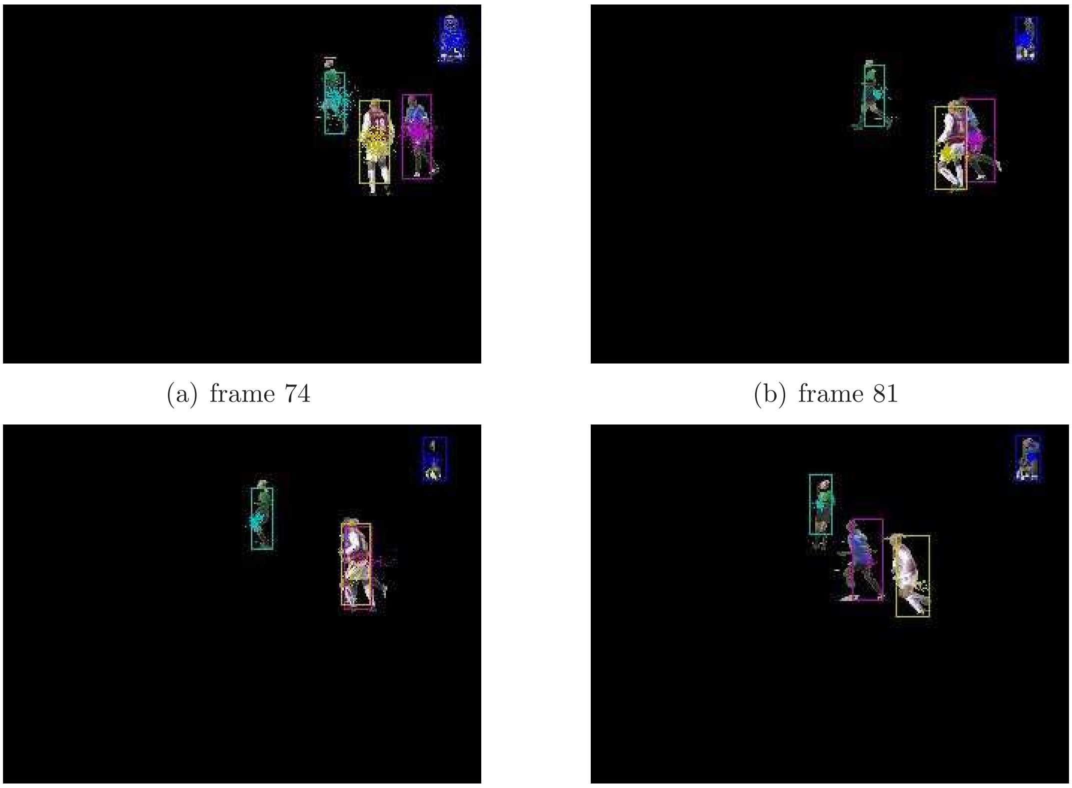
(e) frame 40



(f) frame 45

Figure 7.6: An example of tracking soccer players where occlusion between five players is present.

The example in Fig. 7.7 shows how the tracking can handle the presence of occlusion between two players. In this example, the player with the yellow rectangle and the player with the purple rectangle are moving in the same direction. Due to the fact that the player with the purple rectangle moves faster than the player with the yellow rectangle, it causes them to become occluded in frame 85. The players keep the same velocity in their motions and since the dynamical model in the algorithm considers the players' velocity, we see in frame 93 that the tracking is successfully accomplished.



(a) frame 74

(b) frame 81

(c) frame 85

(d) frame 93

Figure 7.7: An example of tracking soccer players where occlusion between two players is present.

Figure 7.8 displays the tracking results when occlusion between players, which do not move at a constant velocity and have sudden changes of directions in their movements, is present. In this case, tracking the players becomes more difficult. The predicted state of each player (Eq. 3.9) given the past states is based on the dynamical model (Eq. 2.3) and thus has a large prediction error. As shown, the player with the yellow rectangle do not move at the same velocity and changes his direction in frame 116. Tracking the player is accomplished because the non-linear measurement, which represents the number of pixels in the contour which belongs to at least one player relatively to the total number of pixels in the contour, should be one. Therefore, the player's estimated yellow rectangle in frame 124 includes the contour's pixels which do not lie inside other estimated rectangles and thus tracking succeeds. In frame 129, the player with the yellow rectangle changes his direction again and starts falling down. Tracking the player's position is accomplished since the contour's linear measurement z_2 (Eqs. 2.4) is associated with the player.

Figure 7.9 displays an example of tracking soccer players with a method that uses an occlusion alarm probability [12]. To handle occlusion between players, an occlusion alarm probability, which attracts or repels particles, is added. This way the particles' weights of a player are reduced according to their distances to other players' particles. Four players are tracked in this example. The player with the red rectangle is not successfully tracked and in frame 360 the estimated position of this player is moved to another player. The reason that this tracking fails is that the alarm probability caused the particles, which are close to the blue rectangulared player, to have small weights and therefore the estimated player's position becomes far away from the blue rectangulared player and closer to the light blue rectangulared player.

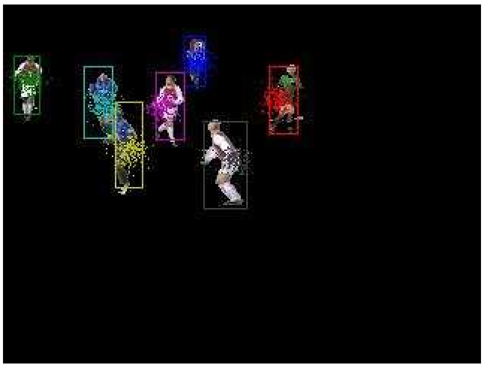
Figure 7.10 displays tracking results of the same scene as in Figure 7.9, but here we use the Monte Carlo data association method which is introduced in the paper. As shown in this example, tracking succeeds. The player with the red rectangle is associated with the contour's linear measurement z_2 in frame 357, thus the player can be successfully tracked.

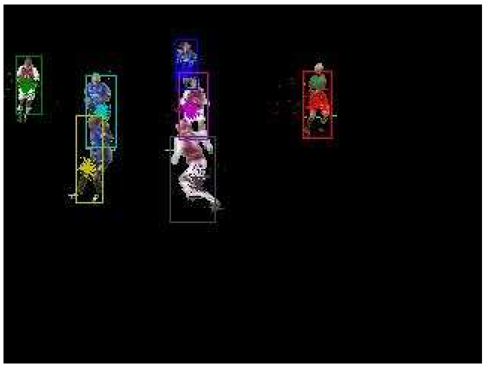
Figure 7.11 shows an example of tracking which fails due to incomplete and inaccurate contours that was generated in the image processing phase. The player with the green rectangle has a smaller number of pixels in frame 27 because some of his pixels were removed during the image processing phase. Since a contour must have a number of pixels that is greater than a threshold, the player's contour is not taken into account in the algorithm. Therefore, as shown in frame 30, the estimated position of the green rectangulared player is inside the contour which belongs to the player with the blue rectangle. This kind of tracking failure is problematic because it can influence other scenes by failing them as well.

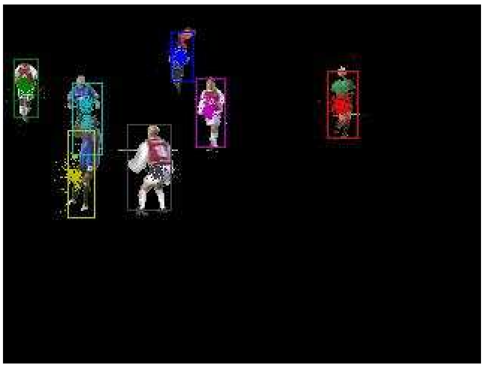
8 Discussion and conclusions

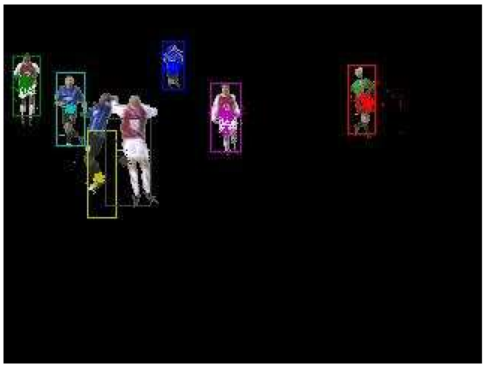
The paper presents a new method for tracking soccer players. The algorithm produces five measurements from each segmented contour in each video frame. A measurement model in which each player is associated with only the contour's pixels, which are most likely originated from him, is suggested. A multi-tracking method which considers all the hypotheses of associating contours to players, is applied using particle filter.

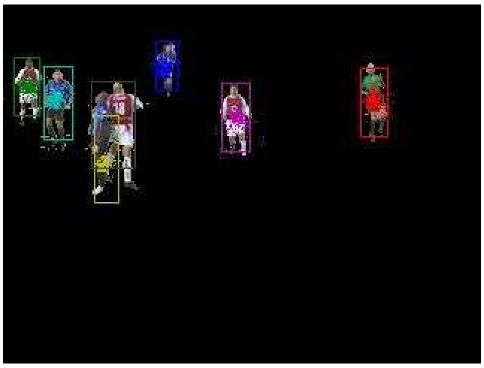
The results achieve good performance in tracking soccer players even when occlusion is present. Situations where more than two players are occluded and situations where players change their directions when getting close to one another, were examined and successfully tracked. The results show that when the players' contours are not properly separated from


(a) frame 111

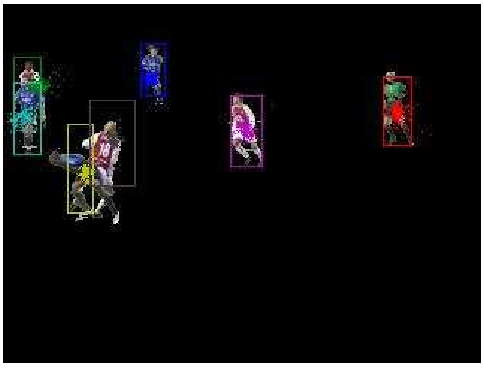

(b) frame 116


(c) frame 120


(d) frame 124

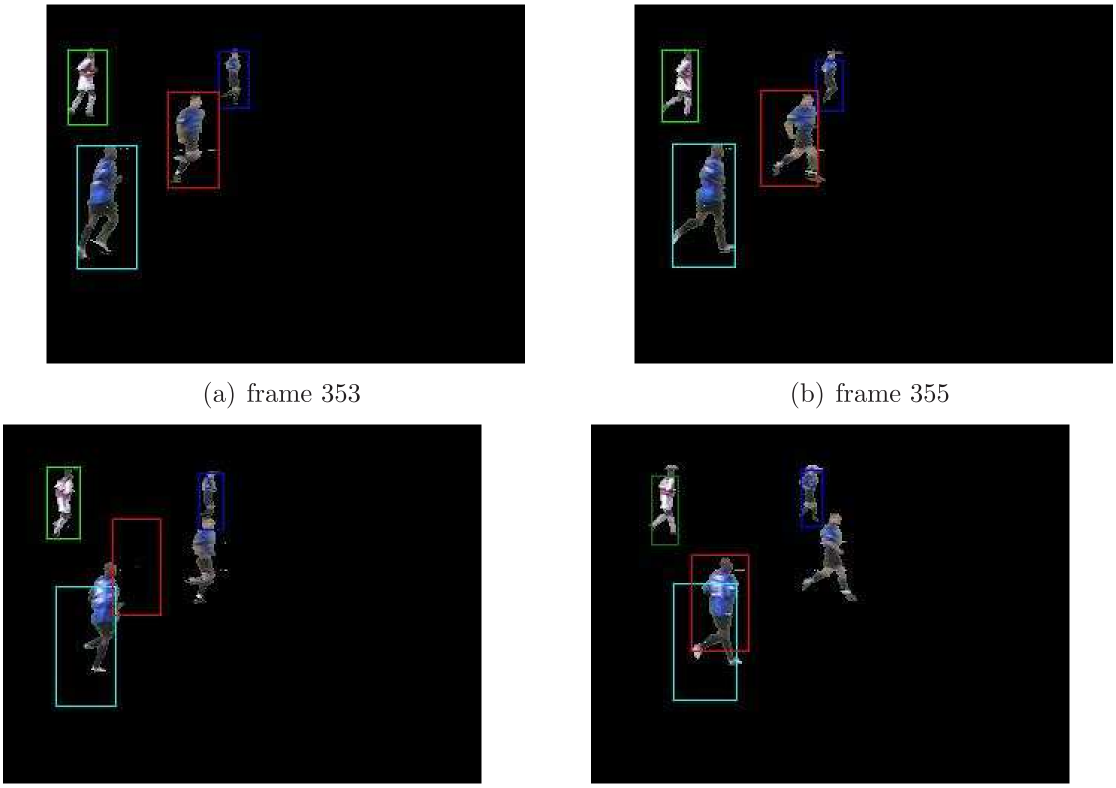

(e) frame 127


(f) frame 129


(g) frame 136


(h) frame 142

Figure 7.8: An example of occlusion between players that change their velocities and their directions.



(a) frame 353 (b) frame 355

(c) frame 357 (d) frame 360

Figure 7.9: An example of tracking soccer players with the occlusion alarm probability method.

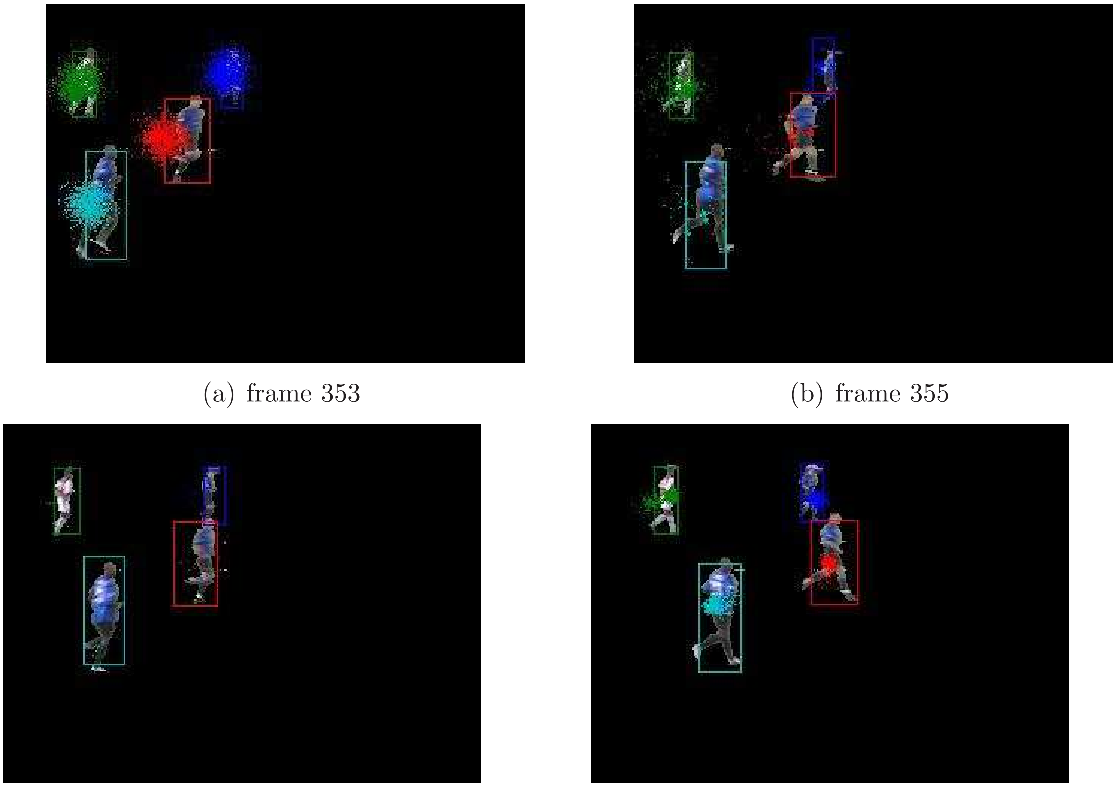
the background in the segmentation phase, then tracking fails. Future work should reduce the sensitivity of the algorithm to the outcome from segmentation.

References

[1] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online non-linear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing,* vol. 50, no. 2, pp. 174-188, Feb. 2002.

[2] Y. Bar Shalom and X. R. Li, *Multitarget-multisensor Tracking: Principles and Techniques.* Urbana, IL: YBS, 1995.

[3] O. Frank, J. Nieto, J. Guivant, and S. Scheding, "Multiple target tracking using sequential monte carlo methods and statistical data association," *Proc. Int Conf. Intelligent Robots and Systems,* vol. 3, oct 2003, pp. 2718- 2723 .



(a) frame 353 (b) frame 355

(c) frame 357 (d) frame 360

Figure 7.10: An example of tracking soccer players using the Monte Carlo data association method.

[4] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to linear system," *Proc. of Aerosense: The 11th Internetional Symposium on Aerospace/Defence sensing, Simulation and Controls, Orlando, Florida,* pp. 182-193, 1997.

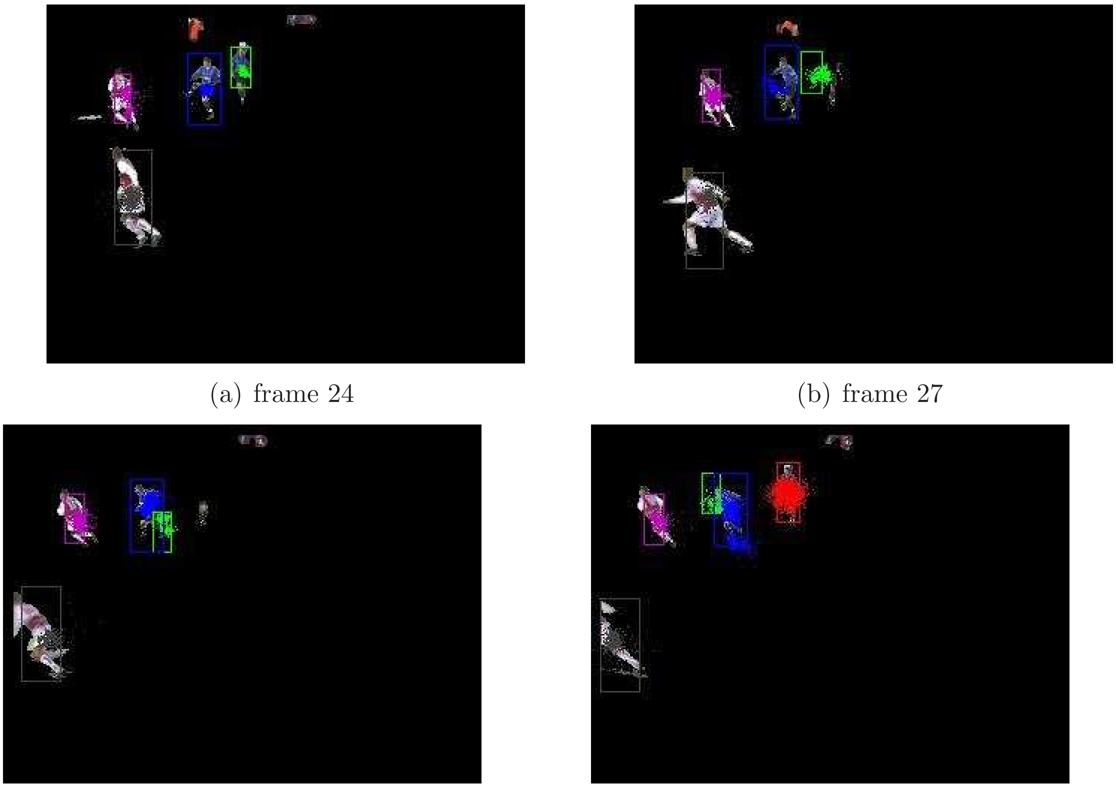
[5] R. Karlsson and F. Gustafsson, "Monte Carlo data association for multiple target tracking," *IEE Target tracking: Algorithms and applications conf.,* Oct 2001.

[6] M. Kass, A. Witkins, and D. Terzopoulos, "Snakes: Active conour models," *Journal of Computer Vision,* vol. 1, no. 4, pp. 321-331, Jan. 1988.

[7] S. Lefevre, J. G. Gerard, A. Piron, and N. Vincent, "An extended snake model for real-time multiple object tracking," *Proc. Int Conf. Workshop on Advanced Concepts for Intelligent Vision Systems,* 2002, pp. 268-275.

[8] P. Li, T. Zhang, and A. E. C. Pece, "Visual contour tracking based on particle filters," *Image and Vision Computing,* vol. 21, no. 1, pp. 111-123, Jan. 2003.

[9] T. Misu, M. Naemura, Z. Wentao, Y. Izumi, and K. Fukui, "Robust tracking of soccer players based on data fusion," *Proc. Int Conf. Pattern Recognition,* 2002, pp. 556- 561.



(a) frame 24 (b) frame 27

(c) frame 30 (d) frame 32

Figure 7.11: An example of tracking which fails due to wrong contours that were generated in the image processing phase.

[10] C. J. Needham and R. D. Boyle, ”Tracking multiple sports players through occlusion, congestion and scale,” *Proc. British Machine Vision Conf.,* 2001, pp. 93-102.

[11] D. Schultz, W. Burgard, D. Fox, and A. B. Cremers, ”People tracking with mobile robots using sample-based joint probabilistic data association,” *International Journal of Robotics Research,* vol. 22, no. 2, pp. 99-116, Feb. 2003.

[12] Y. Seo, and K. S. Hong, ”Occlusion reasoning by occlusion alarm probability for multiple football players tracking,” *IEIce Transactions on Information and Systems,* vol. E87-D, no. 9, pp. 2272-2276, Sept. 2004.

[13] G. Welch and G. Bishop, *An intoduction to Kalman filter*, TR 95-041, University of North Carolina at Chapel Hill, 2001.

A Particle filter methods

A common problem in scientific applications is to estimate a state of a system that changes over time. In order to solve such a problem, there is a need to define a model, which describes the dynamics of the system's state over time (system model) and a model which describes the measurements of the state (measurements model).

Let $\{x_k, k \in \mathbb{N}\}$ be the state sequence of a target, then the system model equation is $x_k = f_k(x_{k-1}, v_{k-1})$ where f_k is a possibly non-linear function and v_k is an i.i.d noise sequence process.

The measurement model equation is $z_k = h_k(x_{0:k}, w_k)$ where h_k is a possibly non-linear function and w_k is an i.i.d noise sequence process.

Bayesian filter provides an optimal solution for this problem, which is a recursive solution that uses the Bayes theorem to calculate the posterior density $p(x_{0:k}|z_{1:k})$. In most cases this solution can not be analytically determined.

Kalman filter is an example for an existing analytical optimal solution. The Kalman filter is an optimal solution for the estimation problem with the assumptions that the system model and the measurement model are both linear with additive white noises which are Gaussian distributed.

In many situations the above assumptions do not hold and therefore Kalman filter can not be used and a suboptimal solution is required. The extended Kalman filter (EKF) is a suboptimal solution based on Kalman filter, which does not require the assumption for the system and measurements equations to be linear. In this method, a local linearization of the equations is done in order to approximate the posterior density of the state by a Gaussian and to use the Kalman filter equations. In some cases, the true posterior density is non-Gaussian and therefore the Gaussian approximation yields an unsatisfying performance. In such cases, particle filter can be used.

Particle filter approach, known as the sequential importance sampling (SIS) approach and the condensation algorithm, is a method which implements a recursive Bayesian filter by Monte Carlo simulations. The main idea is to approximate the posterior density by a set of random samples with their associated weights. In this way, as the number of samples increases, the approximation becomes more satisfying and the filter approaches the optimal Bayesian estimate.

In order to approximate the posterior density, a set of N samples and their associated weights $\{x^i_{0:k}, w^i_k\}_{i=1}^N$ are generated and the associated weights are normalized such that $\sum_{i=1}^N w^i_k = 1$. The posterior density at time step k is approximated by

$$p(x_{0:k}|z_{1:k}) \approx \sum_{i=1}^{N} w^i_k \delta(x_{0:k} - x^i_{0:k}). \tag{1.15}$$

The weights are calculated using the technique of importance sampling. Suppose $p(x_{0:k}|z_{1:k})$ is a probability density from which it is difficult to draw samples. Let $q(x_{0:k}|z_{1:k})$ be a density from which it is easy to draw samples and let x^i_k be the samples generated from $q(x_{0:k}|z_{1:k})$. The proposal density $q(x_{0:k}|z_{1:k})$ is called importance density. The associated weights used to approximate $p(x_{0:k}|z_{1:k})$ in Eq. 1.15 are

$$w^i_k \propto \frac{p(x^i_{0:k}|z_{1:k})}{q(x^i_{0:k}|z_{1:k})} \tag{1.16}$$

If the importance density satisfies

$$q(x_{0:k}|z_{1:k}) = q(x_k|x_{0:k-1}, z_{1:k})q(x_{0:k-1}|z_{1:k-1}) \tag{1.17}$$

then the samples $x^i_{0:k}$ can be derived from the existing samples $x^i_{0:k-1}$ and the new state x^i_k. Using the Bayes law, the posterior density becomes

$$\begin{aligned} p(x_k|z_{1:k}) &= \frac{p(z_k|x_{0:k}, z_{1:k-1})p(x_{0:k}|z_{1:k-1})}{p(z_k|z_{1:k-1})} \\ &= \frac{p(z_k|x_{0:k}, z_{1:k-1})p(x_k|x_{0:k-1}, z_{1:k-1})p(x_{0:k-1}|z_{1:k-1})}{p(z_k|z_{1:k-1})} \\ &= \frac{p(z_k|x_k)p(x_k|x_{k-1})p(x_{0:k-1}|z_{1:k-1})}{p(z_k|z_{1:k-1})} \\ &\propto p(z_k|x_k)p(x_k|x_{k-1})p(x_{0:k-1}|z_{1:k-1}). \end{aligned} \tag{1.18}$$

By substituting Eqs. 1.17 and 1.18 into Eq. 1.16 the weight update equation is

$$\begin{aligned} w_k^i &\propto \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)p(x_{0:k-1}^i|z_{1:k-1})}{q(x_{0:k-1}^i|z_{1:k})q(x_k^i|x_{0:k-1}^i, z_{1:k-1})} \\ &= w_{k-1}^i \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{0:k-1}^i, z_{1:k})} \end{aligned}$$

In addition, if $q(x_k|x_{0:k-1}, z_{1:k}) = q(x_k|x_{k-1}, z_k)$ then the associated weights can be calculated by

$$w_k^i \propto w_{k-1}^i \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i, z_k)}. \tag{1.19}$$

A common problem in particle filters is the degeneracy of the weights. In many applications of the particle filter, all the particles except one have negligible weights after few iterations. This way, most of the particles do not contribute to the posterior density approximation and the approximation becomes unsatisfying.

The degeneracy effect can be reduced by choosing a good importance density. The importance density affects the degeneracy and a good choice of the importance density can reduce the degeneracy. Another way to overcome the problem of degeneracy is to use resampling. The main idea of resampling is to remove the samples which have small weights by generating a new set of samples which have equal weights.

A common particle filter is the sampling importance resampling (SIR) filter which is a special case of the SIS algorithm. In the SIR filter, the importance density is chosen to be the prior density

$$q(x_k|x_{k-1}^i, z_k) = p(x_k|x_{k-1}^i) \tag{1.20}$$

by substituting Eq. 1.20 into Eq. 1.19, the weight update equation is given by $w_k^i \propto w_{k-1}^i p(z_k|x_k^i)$.

In the SIR filter, a resampling procedure is performed in every time step. Therefore, $w_{k-1}^i = \frac{1}{N}$ for all i and the modified weight is $w_k^i \propto p(z_k|x_k^i)$.

The SIR particle filter algorithm is described in Algorithm 1. The reasampling algorithm, also known as systematic resampling, is described in Algorithm 2.

Algorithm 1: SIR Particle Filter

1. For $i = 1 : N$
   Draw $x_k^i \sim p(x_k|x_{k-1}^i)$
   $\tilde{w}_k^i = p(z_k|x_k^i)$

End for

2. $S = \sum_{i=1}^{N} \tilde{w}_k^i$

3. For each i let: $w_k^i = \frac{1}{S}\tilde{w}_k^i$

4. Resample using algorithm 2

Algorithm 2: Resampling Algorithm

1. Let $c_1 = 0$

2. For $i = 2 : N$
   $c_i = c_{i-1} + w_k^i$
   End for

3. $u_1 \sim \mathrm{U}[0, N^{-1}]$ (uniform distribution)

4. For $j = 1 : N$
   $u_j = u_1 + N^{-1}(j-1)$
   While $u_j > c_i$
   $i = i + 1$
   End while
   $x_k^{j*} = x_k^j$
   $w_k^j = \frac{1}{N}$
   $i^j = i$
   End for