Distributed Sensor Fusion for Object Tracking

Alankar Karol and Mary-Anne Williams

Faculty of Information Technology, University of Technology, Sydney, NSW 2007, Australia

Abstract. In a dynamic situation like robot soccer any individual player can only observe a limited portion of their environment at any given time. As such to develop strategies based upon planning and cooperation between different players it is imperative that they be able to share information which may or may not be in any individual player's field of vision. In this paper we propose a method for multi-agent cooperation for perception based upon the Extended Kalman Filter (EKF) which enables players to track objects absent from their field of vision and also to improve the accuracy of position and velocity estimates of objects in their field of vision.

1 Introduction

Robot Soccer is a relatively new research initiative and in terms of its development it is in its infancy. One of the current challenges for robots playing in the RoboCup Four-Legged League is to implement a game based upon cooperation and planning. Planning e.g. behaviour or a strategy based upon game play [2] can be facilitated by the efficient sharing of information among the team players. Players can share information regarding their own pose, estimate of the pose of the opponents and an estimate of the ball position and velocity.

Since each player (generally) knows its own pose with a high degree of accuracy sharing this information is fairly straightforward. However sharing information about the ball and the opponents is not that trivial. In this paper we ignore the problem of tracking opponents since we wish to avoid the problem of data association. We propose a solution for ball tracking by fusing information from multiple robots using an Extended Kalman Filter (EKF).

Kalman filters have been successfully used in robotics to track objects and achieve robot localisation with a high degree of accuracy. For robot localisation, Kalman filters effectively fuse information from position estimating sensors like sonars and vision with odometry [5] [7] [8] [4]. The observations are matched to a map of the local environment and then merged to update the robot pose. In the studies mentioned above, Kalman filters were successfully employed to fuse together information from different sensors located on the same robot. We undertake the task of fusing distributed simultaneous sensor information from different robots.

The problem of ball tracking by fusing multiple simultaneous observations of the same object from distributed vantage points has been tackled in the past [6] [9]. Stroupe *et al.* [6] used the method of Smith and Cheeseman [5] to combine two dimensional Gaussian distributions by a single matrix operation. Due to the associativity and the symmetry of the operation any number of distributions can be combined in any order

A. Bredenfeld et al. (Eds.): RoboCup 2005, LNAI 4020, pp. 504–511, 2006.

to fuse the information from different robots leading to a highly efficient and reactive method. However the parameters of the observation in their case do not correspond to the canonical form of the Gaussian as assumed by the method of Smith and Cheeseman [5]. To overcome this obstacle they have to rotate their observations and then re-rotate them in the end to extract the position and uncertainty of the tracked object. Moreover, the method they describe is used only for locating the position of the ball when the robots are perfectly localised (i. e. ,the different robots know their pose with certainty) and standing still which is achieved artificially by placing the robots in specific locations. Also, since they only locate the position of the ball they are unable to extract information regarding the speed and the direction of the ball's motion.

The solution to the general problem of fusing the multiple simultaneous observations to accurately determine the position and the velocity of the ball is described by Weigel *et al.* [9]. They determined the ball position by a probabilistic integration of all ball measurements coming from the different players using a combination of Kalman Filters and Markov Localisation. Since Grid-Based Markov Localisation is computationally expensive they used only a two dimensional grid which does not allow them to store the velocity of the ball and as such they could not determine the position of the ball accurately when the ball was in motion. However their algorithm does allow for integrating the ball in a global sensor integrator placed off-board.

In the robocup legged-league off-board processing is not allowed and any algorithm for fusing the multiple simultaneous observations wold have to be constrained by the computational power of the AIBO robots. We propose a method for ball tracking to accurately determine the ball position and velocity when both the robots and the ball is in motion. We use an approach based upon Extended Kalman filters which does not make any assumptions of linearity. The algorithm is highly efficient and does not place undue load on an AIBO Robot and can be used either on an ERS-7 or an ERS-210 without degrading performance. The algorithm was successfully used in the RoboCup 2004 championships held in Lisbon where UTSUnleashed! exploited the power of information sharing in the Open Challenge where we demonstrated active passing between players. UTSUnleashed! also used the above algorithm in games to successfully create game plays based upon cooperation which led us to second position in the soccer competition in our second year of competition.

The remainder of the paper is set out as follows. In the next section we give a brief description of the theory of Extended Kalman filter. In Section 3 we present the state model and the algorithm for fusing multiple simultaneous ball observations by different robots. In section 4 we present the results of some analytical experiments and in section 5 we conclude with a brief discussion.

2 Extended Kalman Filter

The problem of ball tracking and robot localisation requires an estimate every time a new measurement is received, which calls for a solution comprising of a recursive filter. This means that the data can be processed sequentially rather than as a batch, which greatly enhances the execution of the filter algorithm since it is not necessary to store the complete data set nor is it necessary to reprocess the existing data when new measurements become available. Such a filter consists of essentially two stages: prediction and update. The prediction stage uses the system model to predict the state probability density function from one measurement time to the next. The state is usually subject to unknown disturbances which are modelled as random noise hence the prediction stage generally translates, deforms and spreads the state probability density function. The update operation uses the latest sensor measurements to modify the predicted probability density function.

Formally we want to consider the evolution of the state sequence $\{x_k, k \in N\}$ given by $x_k = f_k(x_{k-1}, v_{k-1})$ where f_k maybe a nonlinear function of the state x_{k-1} and v_{k-1} is a process noise sequence. The objective of tracking is to recursively estimate x_k from measurements $z_k = h_k(x_k, w_k)$ where h_k maybe a nonlinear function and n_k is a measurement noise sequence. In particular we seek filtered estimates of x_k based on the set of all available measurements $z_{1:k} = \{z_i, i = 1, \cdots, k\}$ up to time k.

From a Bayesian perspective, the tracking problem is to recursively calculate some degree of belief in the state x_k at time k, given the data $z_{1:k}$ up to time k. Thus it is required to construct the probability distribution function $p(x_k|z_{1:k})$. It is assumed that the initial pdf $p(x_0|z_0) \equiv p(x_0)$ is known a priori. Then in principle, the pdf $p(x_k|z_{1:k})$ may be obtained recursively in two stages: prediction and update.

Suppose that the required pdf $p(x_{k-1}|z_{1:k-1})$ at time $(k-1)$ is available. The prediction stage involves using the system model to obtain the prior pdf of the state at time k:$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})dx_{k-1}$. At time step k, a measurement z_k becomes available, and this may be used to update the prior pdf via Baye's rule:$p(x_k|z_{1:k}) = \dfrac{p(z_k|x_k)p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})}$. In the update stage, the measurement z_k is used to modify the prior density to obtain the required posterior density of the current state.

The Extended Kalman filter assumes that the posterior density at every time step is Gaussian and hence, parametrised by a mean and a covariance. If $p(x_{k-1}|z_{1:k-1})$ is Gaussian then it has been shown [1] that $p(x_k|z_{1:k})$ is also Gaussian. It is then possible to write the above by means of matrix operations

$$\hat{F}_k = \left.\frac{df_k(x)}{dx}\right|_{x=m_{k-1|k-1}} \tag{1}$$

$$\hat{H}_k = \left.\frac{dh_k(x)}{dx}\right|_{x=m_{k|k-1}} \tag{2}$$

If we define $\mathcal{N}(x, ; m, P)$ as a Gaussian density with argument x, mean m and covariance P, the Extended Kalman filter algorithm can be viewed as the following recursive relationship:

$$\begin{aligned} p(x_{k-1}|z_{1:k-1}) &\approx \mathcal{N}(x_{k-1}; m_{k-1|k-1}, P_{k-1|k-1}) \\ p(x_k|z_{1:k-1}) &\approx \mathcal{N}(x_k; m_{k|k-1}, P_{k|k-1}) \\ p(x_k|z_{1:k}) &\approx \mathcal{N}(x_k; m_{k|k}, P_{k|k}) \end{aligned}$$

where

$$m_{k|k-1} = f_k(m_{k-1|k-1}) \tag{3}$$

$$P_{k|k-1} = Q_{k-1} + \hat{F}_k P_{k-1|k-1} \hat{F}_k^T \tag{4}$$
$$m_{k|k} = m_{k|k-1} + K_k(z_k - h_k(m_{k|k-1})) \tag{5}$$
$$P_{k|k} = P_{k|k-1} - K_k \hat{H}_k P_{k|k-1}. \tag{6}$$

In the above $(z_k - h_k(m_{k|k-1}))$ is termed as innovation and is the difference between the expected and the actual measurement. K_k is defined as the Kalman gain and is given by $K_k = P_{k|k-1}\hat{H}_k^T S_k^{-1}$. where S_k is the covariance of the innovation term and is given by:$S_k = \hat{H}_k P_{k|k-1} \hat{H}_k^T + R_k$.

3 Ball Tracking

In the case of robot localisation it is fairly simple to have a state model since the motion model which consists of the commands given to the actuators (walk forward, strafe left, turn clockwise) captures the robot's kinematics. The update model consists of the observations (distance, heading and elevation) made by the robot's camera to the landmarks, if seen in the frame. If the robot sees more than one landmark in any given vision frame then the estimate of the robot's pose is highly accurate since the uncertainties introduced by the observation of one landmark are compensated for by the observations to the other landmarks. Thus the more landmarks that a robot sees in any given frame the more accurate the estimate of the pose.

We use a similar logic for the problem of ball tracking by reversing the problem of robot localisation into a problem of ball localisation. Let us assume for the sake of explanation that the ball has a camera on it and the robots are the beacons. The observations made by the robots to the ball can then be viewed as observations made by the imaginary camera on the ball. One can then imagine that the ball uses the EKF described above to localise itself off the robots. The more robots that a ball views the better localised it will be since the uncertainties are compensated for by multiple observations.

The state vector of the ball is given by $(x, y, v_x, v_y)^T$ where x and y are the coordinates of the ball in a global reference frame and v_x and v_y are the speeds of the ball along the x and the y axis respectively. The heading of the ball with respect to the x-axis can then be easily calculated by $\phi_b = \tan^{-1}\left(\frac{v_y}{v_x}\right)$ and the speed of the ball is given by $v_b = \sqrt{v_x^2 + v_y^2}$.

Since it is impossible to capture the exact kinematics of the ball we model the state model by using the standard equations of kinematics.

$$\begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}_f = \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}_i + \begin{bmatrix} (v_x + \frac{a\delta t}{2})\delta t \\ (v_y + \frac{a\delta t}{2})\delta t \\ a\delta t \\ a\delta t \end{bmatrix} \tag{7}$$

In the above system a is the magnitude of the retardation of the ball which is determined by experiments and δt is a time interval.

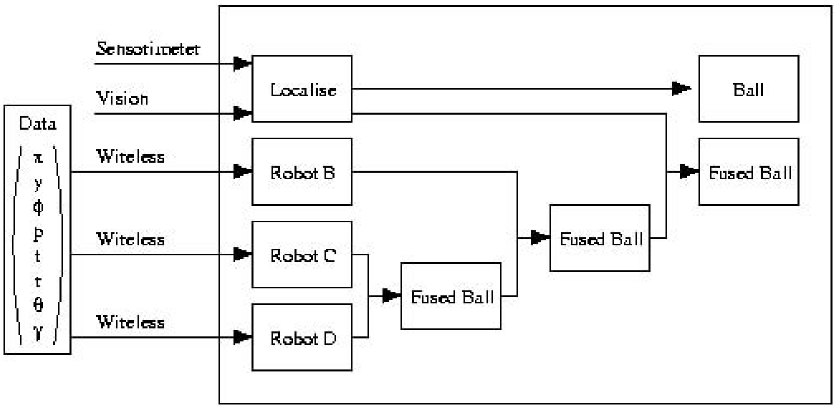

Fig. 1. Diagram representing the algorithm used to fuse information from different robots using the method described above

The measurement model comprises of the distance to the ball, the heading to the ball and the elevation to the ball with respect to the camera center of the robot. Formally:

$$\text{Distance} = \sqrt{(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2}. \tag{8}$$

$$\text{Bearing} = \tan^{-1}\left(\frac{D_v}{D_u}\right). \tag{9}$$

$$\text{Elevation} = \tan^{-1}\left(\frac{D_w}{D_u}\right). \tag{10}$$

Where, D_u, D_v, D_w are the unit vectors from the camera. The coordinates of the robot's camera center (x_c, y_c, z_c) can easily be calculated if the pose of the robot, the neck tilt and the head tilt and pan angles are known.

It is well known that the observations of landmarks leads to a reduction in the uncertainty of the pose when using an EKF [1][3]. Thus if a robot observes a landmark in a vision frame then its uncertainty is reduced. If the robot has not observed a landmark then we increase it's uncertainty by a constant factor determined by experiments. Thus the measure of the uncertainty gives an indication of how reliable the observations of a particular robot are. If the uncertainty of the robot is within some error bound then we consider it's observations to be reliable and the robots transmits it's observations to the team members. If on the other hand the uncertainty is large then it's observations would corrupt the EKF of the team members and hence the robot does not send its information. In this case the robot calculates the position of the ball relative to it's frame of reference. In this situation the robot is still able to react to the ball which is important in a game of robot-soccer.

The robots which satisfy the uncertainty threshold described above then transmit to their team mates information regarding their own pose, their neck tilt and pan angles and the observation of the ball (distance, heading, elevation). We send the neck tilt and pan angles as opposed to the camera coordinates and the unit vectors to reduce the overall transmission on the wireless network.

The receiving robot calculates the camera locations and the unit vectors for each robot with the aid of the above information and then updates the ball position and velocity as described in section (2). As a result of this algorithm the pose of the **fused ball** is unique and is common to all the robots. This pose might be different from the pose which is calculated independently by the robot based on its own observations.

Different issues that arise from such a procedure are tackled as such:

– If the robot can see the ball then behaviour is adopted based upon its own observation of the ball. This is necessary to accommodate for the fact that the data from other robots lags by at least one frame. Moreover if the fused ball is quite distinct from the robot's own perception of the ball then th robot is not forced to make wrong decisions or be frozen as a result of the disparity. This is important in a highly reactive game like robot soccer.
– If the uncertainty in the determination of the robot's own pose is high then it does not send the ball information to the other robots. This normally tends to happen when the robot is chasing a ball and is unable to see a landmark. In this case if the uncertainty in the fused ball is small (compared to some threshold) then it is possible for the lost robot to use the ball as a landmark and update its own pose. Complete localisation takes a few cycles and the robot will inadvertently see a landmark during that time.
– Since the ball identification and measurements are vision based (as opposed to a sonar) it is quite possible that erroneous information may enter the data set. For example a robot might see a false ball or get a wrong distance reading due to reflection. In this case we employ a 2 sigma gate procedure which does not allow any readings which deviate by 2 standard deviations to be integrated into the set.
– If no observation is made for 2 seconds (50 framesin an ERS210 and 60 frames in an ERS7) then the track is deleted and the ball position is reinitialised from observation.

4 Results

Experiments were performed using Sony AIBO ERS-210 robots. Several experiments trying to emulate the dynamic nature of a robot soccer game were carried out. A brief description of a prototypical example is given below and the results reported.

Robots were placed on the field and allowed to move. A ball was rolled along a particular trajectory and the estimates of the ball position and velocity from the robot were recorded. Figure (2) shows the configuration of the field and the position of the robots initially. Robot A was looking outfield and was not allowed to move. However the other robots were allowed to move. The robots were given a few seconds to localise themselves. A ball was then placed on the penalty box of the blue goal and made to roll diagonally to the penalty box of the yellow goal. Robots B, C and D could see the ball at all times with Robot D and C coming quite close to the ball although none of the robots managed to touch the ball. As robot A could not see the ball at any instant, information from the three robots was fused together according to the process described above and the trajectory of the ball according to Robot A's world model was plotted.

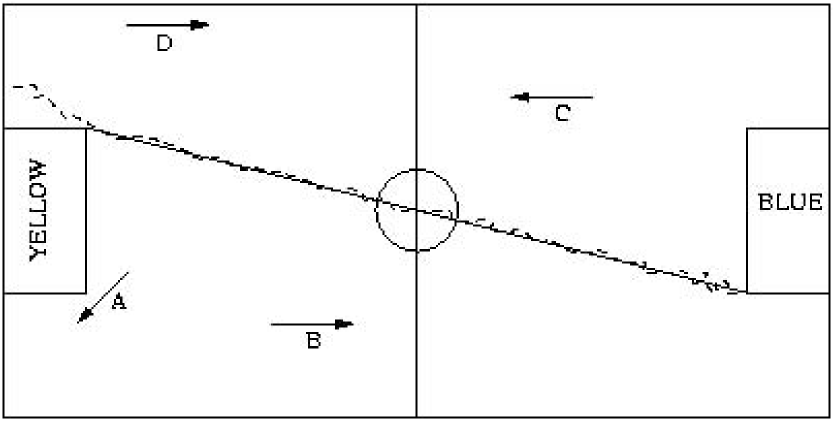

Fig. 2. Trajectory of the ball (solid line) and the trajectory as observed by Robot A (dashed line). Robot A was standing in place while robots B, C and D were allowed to move.

The result is shown in figure (2). As can bee seen from the figure the trajectory of the ball as evaluated by robot A is quite accurate.

Several trials of similar experiments were performed and on an average the mean deviation from the actual path was 5 cm with a standard deviation of 1 cm. Trials performed where all the robots could not see the ball simultaneously gave similar results reflecting the robustness of the algorithm.

5 Discussion

In this paper we present a method of integrating simultaneous observations of a single unique object from different robots using an extended Kalman filter. The computational ease and flexibility of the approach makes it an ideal candidate for object tracking in complex dynamic domains. The approach was tested in the domain of the Robocup where the task is made particularly difficult due to the dynamic and uncertain nature of the domain. The ability of moving robots to fuse information about moving targets enabled UTSUnleashed! to build a highly competitive team at RoboCup 2004.

References

1. Ho, Y.C. and Lee, R.C.K.: A Bayesian Approach to Problems in Stochastic Estimation and Control. IEEE Trans. Automat. Contr., vol. AC-9, 1964. pp. 333–339.
2. Karol,A., Nebel, B., Stanton, C., Williams, M-A.: Case Based Game Play in the Robocup Four-Legged League Part I The theoretical Model. Robocup Symposium 2003.
3. Maybeck, Peter S.: Stochastic Models, Estimation and Control: Volume I. Academic Press, 1979.
4. Sasiadek, J.Z. and Hartana, P.: Sensor data fusion using kalman filter. Proceedings of the third International Conference on Information Fusion, Vol 2, 2000. 19–25

5. Smith, R.C. and Cheeseman, P.:On the Representation and Estimation of Spatial Uncertainty. International Journal of Robotics Research, Vol 5, no. 4, Winter 1986. 56–68.
6. Stroupe, A., Martin, M.C. and Balch, T.: Distributed Sensor Fusion for Object Position Estimation by Multi-Robot Systems. Proceedings of the IEEE International Conference on Robotics and Automation- May, 2001. IEEE, 2001.
7. Tang, C.Y., Hung, Y.P., Shih, S.W. and Chen, Z.: A Feature Based tracker for multiple object tracking. Proceedings of the National Science Council, republic of China, Vol 23(1) Part A, 1999. 151–168.
8. Wang, H., Chua, C.S. and Sim, C.T.: Real Time Object Tracking from Corners. Robotica, Vol 16(1), 1998. 109-116.
9. Weigel, T., Gutmann, J.-S., Dietl, M., Kleiner, A. and Nebel, B.: CS Freiburg: Coordinating Robots for Successful Soccer Playing, IEEE Transactions on Robotics and Automation, 18(5),2002. 685-699.