Watch their Moves: Applying Probabilistic Multiple Object Tracking to Autonomous Robot Soccer

Thorsten Schmitt, Michael Beetz, Robert Hanek, and Sebastian Buck
Munich University of Technology,
Department of Computer Science,
D-80290 Muenchen
{schmittt,beetzm,hanek,buck}@in.tum.de

Abstract

In many autonomous robot applications robots must be capable of estimating the positions and motions of moving objects in their environments. In this paper, we apply probabilistic multiple object tracking to estimating the positions of opponent players in autonomous robot soccer. We extend an existing tracking algorithm to handle multiple mobile sensors with uncertain positions, discuss the specification of probabilistic models needed by the algorithm, and describe the required vision-interpretation algorithms. The multiple object tracking has been successfully applied throughout the RoboCup 2001 world championship.

Introduction

In many autonomous robot applications robots must be capable of estimating the positions and motions of moving objects in their environments. Autonomous cars driving on a highway, for example, must know where the other cars are and what they do (Dickmanns 1997). Robots navigating through crowded regions must keep track of the individual persons (Thrun *et al.* 2000). Autonomous robots playing soccer must know where their opponents are in order to make the right moves and plays (Beetz *et al.* 2002).

In most cases, the robots employ specific state estimation algorithms for keeping track of the moving objects in their environments. For various reasons, these kinds of tracking systems are difficult to realize. Observations of the robots are inaccurate and incomplete. Sometimes the sensors hallucinate objects. Often the robots cannot perceptually distinguish the individual objects in their environments. To reliably estimate the positions and motions of the objects despite these perturbations, researchers have proposed multi object tracking algorithms. Tracking algorithms use motion models of the objects and sequences of observation to distinguish real object observations from clutter and can thereby keep track of object positions both more reliably and more accurately.

Some of the most successful approaches to the multi object tracking problem are probabilistic tracking algorithms, such as Multiple Hypothesis Tracking (MHT) (Reid 1979; Cox & Hingorani 1996) and Joint Probabilistic Data Association Filter (JPDAF) (Bar-Shalom & Fortmann 1988; Schulz *et al.* 2001). Using probabilistic motion and sensing models these algorithms maintain probabilistic estimates of the objects' positions and update these estimates with each new observation. Probabilistic tracking algorithms are attractive because they are concise, elegant, well understood, and remarkably robust.

In this paper, we apply MHT, one of the probabilistic tracking algorithms introduced above, to estimating the positions of opponent players in autonomous robot soccer based on image data. In robot soccer (middle-size league) two teams of four autonomous robots play soccer against each other. The tracking problem in autonomous robot soccer confronts probabilistic tracking methods with a unique combination of difficult challenges. The state is to be estimated by multiple mobile sensors with uncertain positions, the soccer field is only partly visible for each sensor, occlusion of robots is a problem, the robots change their direction and speed very abruptly, and the models of the dynamic states of the robots of the other team are very crude and uncertain.

In this paper we show how the MHT algorithm can be applied to opponent tracking in autonomous robot soccer. This application requires programmers to equip the robots with sophisticated mechanisms for observing the required information, and to provide probabilistic domain descriptions that the algorithm needs for successful operation. These probabilistic descriptions include motion models and sensing models, such as the probability of the robot detecting an object within sensor range. We show that such mechanisms enable the MHT to reliably and accurately estimate the positions of opponent robots using passive vision-based perception where the cameras have a very restricted field of view. In addition, we will show that the cooperation between robots provides the robots with a more complete estimate of the world state, a substantial speed up in the detection of motions, and more accurate position estimates.

In the remainder of the paper we proceed as follows. The next section describes the MHT algorithm. In the subsequent section we provide a detailed account of how to apply the MHT to autonomous robot soccer. Then we discuss how good parameter settings for autonomous robot soccer can be found and how cooperation between the robots can be exploited to obtain better performance. We conclude with a discussion of these findings and a discussion of related work.

Multiple Hypothesis Tracking

Multiple hypothesis tracking considers the following state estimation problem. The world is populated with a set of stationary and moving objects. The number of objects may vary and they might be occluded and out of sensor range. Robots are equipped with sensing routines that are capable of detecting objects within sensor range, of estimating the positions of the detected objects, and of assessing the accuracy of their estimate.

The objective of the MHT algorithm is to keep a set of object hypotheses, each describing a unique real object and its position, to maintain the set of hypotheses over time, and to estimate the likelihood of the individual hypotheses.

The basic data structure used by the MHT algorithm is the object hypothesis. An object hypothesis consists of an estimated position, orientation, and velocity of an object, a measure of uncertainty associated with the estimation, and a second measure that represents the degree of belief that this hypothesis accurately reflects an existing object. Because the number of objects might vary new hypotheses might have to be added and old ones might have to be deleted.

Before we dive into the details of the MHT algorithm let us first get an intuition of how it works. The MHT algorithm maintains a forest of object hypotheses, that is a set of trees. The nodes in the forest are object hypotheses and represent the association of an observed object with an existing object hypothesis. Each hypothesis has an association probability, which indicates the likelihood that observed object and object hypothesis refer to the same object. In order to determine this probability the motion model is applied to the object hypothesis of the previous iteration, in order to predict where the object will be now. Then the association probability is computed by weighing the distance between the predicted and the observed object position. Thus in every iteration of the algorithm each observation is associated with each existing object hypothesis.

The MHT Algorithm

Our MHT algorithm is an extension of Reid's algorithm (Reid 1979). It extends Reid's version in that it can handle multiple mobile sensors with uncertain positions. The computational structure of the algorithm is shown in Fig. 1.

An iteration begins with the set of hypotheses of object states $H^k = \{h_1^k, \ldots, h_m^k\}$ from the previous iteration k. Each h_i^k is a random variable ranging over the state space of a single object and represents a different assignment of measurements to objects, which was performed in the past. The algorithm maintains a Kalman filter for each hypothesis.

With the arrival of new sensor data (6), $Z(k+1) = \{z_1(k+1), \ldots, z_{n_{k+1}}(k+1)\}$, the motion model (7) is applied to each hypothesis and intermediate hypotheses $\widehat{h}_i^{k+1}$ are predicted. Assignments of measurements to objects (10) are accomplished on the basis of a statistical distance measurement, such as the Mahalanobis distance. Each subsequent child hypothesis represents one possible interpretation of the set of observed objects and, together with its parent hypothesis, represents one possible interpretation of all past observations. With every iteration of the MHT probabilities

```
algorithm MULTIPLEHYPOTHESISTRACKING()
 1  let Ĥ^k = {ĥ_1^k, ..., ĥ_{m_k}^k}  % pred.hypos.
 2     Z(k) = {z_1(k), ..., z_{n_k}(k)}   % ob.feat.
 3     H^k = {h_1^k, ..., h_{o_k}^k}  % new hypos.
 4     X^{k-N}  % world state at time k-N.
 5  do for k ← 1 to ∞
 6     do Z(k) ← INTERPRETSENSORDATA();
 7        Ĥ^k ← APPLYMOTIONMODEL(H^{k-1}, M);
 8        for i ← 1 to n_k
 9        do for j ← 1 to m_k
10           do h_ij^k ← ASSOCIATE(ĥ_j^k, z_i(k));
11              COMPUTE(P(h_ij^k|Z(k)))
12        for j ← 1 to n_k
13        do H^k ← H^k ∪ {GENERNEWHYP(z_j(k))};
14        PRUNEHYPOTHESIS(H^k);
15        X^{k-N} ← {x_1^{k-N}, ..., x_{o_{k-N}}^{k-N}}
```

Figure 1: The multiple hypothesis tracking algorithm.

(11) describing the validity of an hypothesis are calculated. Furthermore for every observed object an new hypothesis with associated probability is created (13).

In order to constrain the growth of the hypothesis trees the algorithm prunes improbable branches (14). Pruning is based on a combination of ratio pruning, i.e. a simple lower limit on the ratio of the probabilities of the current and best hypotheses, and the N-scan-back algorithm (Reid 1979). This algorithm assumes that any ambiguity at time k is resolved by time $k+N$. Consequently if at time k hypothesis h_i^{k-1} has m children, the sum of the probabilities of the leaf notes of each branch is calculated. The branch with the greatest probability is retained and the others are discarded. After pruning the world state of X^{k-N} can be extracted (15). Please note that this world state is always N steps delayed behind the latest observations. However, in the Application section we will demonstrate that this delay can be overcome by N observers performing observations in parallel.

Computing the Likelihood of Hypotheses

Obviously, the heart of the MHT algorithm is the computation of the likelihood of the different hypothesis-observation associations, $P(h_{ij}^{k+1}|Z(k))$, in line 11 of the algorithm in figure 1. In this section we derive the formula that is used in order to compute this probability. The derivation of this formula is critical because it tells us which probabilities must be specified by programmers in order to apply the algorithm to specific applications.

Let Z^k be the sequence of all measurements up to time k. A new hypothesis of an object at time k is made up of the current set of assignments (also called an event), $\theta(k)$, and a previous state of this hypothesis, h_j^{k-1}, based on observed features up to time step $k-1$ inclusively.

We can transform the probability of an object's hypothesis $P(h_i^t|Z^k)$ using Bayes' rule and the Markov assumption in

order to obtain an easier expression.

$$\begin{aligned} P(h_i^k|Z^k) &= P(\theta(k), h_i^{k-1}|Z(k), Z^{k-1}) && (1)\\ &= P(\theta(k), h_i^{k-1}|Z(k), H^k) && (2)\\ &= \alpha * p(Z(k)|\theta(k), h_j^{k-1}, Z^{k-1}) && (3)\\ & \quad P(\theta(k)|h_j^{k-1}, Z^{k-1})P(h_j^{k-1}|Z^{k-1}) \end{aligned}$$

Here α is a normalization factor ensuring that $P(h_i^k|Z^k)$ sums up to one over all h_i^k. The last term of this equation is probability of the parent global hypothesis that has been computed in the previous iteration. The second factor can be evaluated as follows (Bar-Shalom & Fortmann 1988):

$$P(\theta(k)|h_j^{k-1}, Z^{k-1}) = \frac{\phi!\nu!}{m_k!}\mu_F(\phi)\mu_N(v) \quad (4)$$
$$\prod_t (P_D^t)^{\delta_t}(1-P_D^t)^{1-\delta_t}(P_T^t)^{\tau_t}(1-P_T^t)^{1-\tau_t}$$

where $\mu_F(\phi)$ and $\mu_N(\nu)$ are prior probability mass functions of the number of spurious measurements and new geometric features. P_D^t and P_T^t are the probabilities of detection and termination of track t (originating from hypothesis h_t^{k-1}) and δ_t and τ_t are indicator variables. δ_t (τ_t) is 1, if track t is detected (deleted) at time k and 0 otherwise. The indicator variable δ_t depends on the observing robots camera orientation. It is 1, if the track t is within the sensors field of perception and track t is not occluded by another team mate. $P_T^{\tau_t}$ is used to model the declination of an unobserved hypothesis probability over time. It is defined as $P_T^{\tau_t} = 1 - e^{-\frac{\Delta k}{\lambda_T}}$. Δk is the number of consecutive time steps an hypothesis was not observed. λ_T determines the speed of the declination process. Larger λ_T result in a slower declination of the hypothesis probability.

The first term on the right hand side of equation 3 denotes the association probability of a measurement and an hypothesis. In order to determine this term is is assumed that a measurement z_i^k has a Gaussian probability density function if it is associated with object t_i.

$$N_{t_i}(z_i(k)) = N_{t_i}(z_i(k), \widehat{h}_i^k, S_{t_i}^k) = \quad (5)$$
$$|2\pi S_{t_i}^k|^{-\frac{1}{2}} e^{-\frac{1}{2}\{(z(k)-\widehat{h}_i^k)^T\{S_{t_i}^k\}^{-1}(z(k)-\widehat{h}_i^k))\}}$$

Here $\widehat{h}_i^k$ denotes the predicted measurement for hypothesis t_i and $S_{t_i}^k$ is the associated innovation covariance. The probability of a new object and a spurious measurement are taken to be uniformly distributed over the observation volume V. In our implementation the observation volume V is the intersection of the field of view (neglecting occlusions) and the soccer field. Thus V is a function of the robot's pose estimate and the camera's field of view.

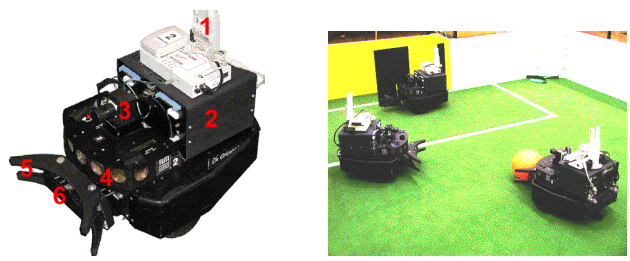
$$p(Z(k)|\theta(k), h_j^{k-1}, Z^{k-1}) = \prod_i^{m_k} [N_{t_i}(z_i(k))]^{\kappa_i} V^{-(1-\kappa_i)} \quad (6)$$

ϕ and ν are the total numbers of false alarms and new geometric features, respectively and κ_i is another indicator variable which is 1, if $z_i(k)$ came from a known track and 0 otherwise.

Applying MHT to Autonomous Robot Soccer

Robotic soccer has become a standard "real-world" testbed for autonomous multi robot control. In robot soccer (mid-size league) two teams of four autonomous robots — one goal keeper and three field players — play soccer against each other. The soccer field is four by nine meters big surrounded by walls. The key characteristics of mid-size robot soccer is that the robots are completely autonomous. Consequently, all sensing and all action selection is done on board of the individual robots. Skillful play requires our robots to recognize objects, such as other robots, field lines, and goals, and even entire game situations.

The AGILO RoboCup team consists of four Pioneer I robots; one of them is depicted in Fig. 1(a). The robot is equipped with a single on board linux computer (2), a wireless Ethernet (1) for communication, and several sonar sensors (4) for collision avoidance. A color CCD camera with an opening angle of 90° (3) is mounted fix on the robot. The robot also has a dribbling (5) and a kicking device (6) that enable the robot to dribble and shoot the ball. Autonomous robot soccer confronts object tracking mechanisms with challenging research problems. The camera system with an opening angle of 90° and pointed to the front gives an individual robot only a very restricted view of the game situation. Therefore, the robot needs to cooperate to get a more complete picture of the game situation. Vibrations of the camera, spot light effects, and poor lighting conditions cause substantial inaccuracies. Even small vibrations that cause jumps of only two or three pixel lines cause deviations of more than half a meter in the depth estimation, if the objects are several meters away. The opponent robots change their speed and moving directions very quickly and therefore an iteration of the tracking algorithm has to be very fast such that the inaccuracies of the motion model does not have such a huge effect.



(a) (b)

Figure 2: An AGILO soccer robot (a) and a game situation (b).

Sensor Data Interpretation for Robot Soccer

Let us first look at the sensor data interpretation process for tracking in robot soccer. The information needed for object tracking is provided by the perception system and includes the following kinds of information: (1) partial state estimates broadcasted by other robots, (2) feature maps extracted from captured images, and (3) odometric information. The estimates broadcasted by the team mates comprise the respective robot's location and the locations of the oppo-

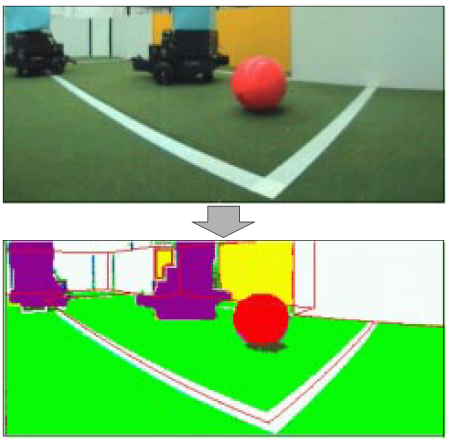


Figure 3: The figure shows an image captured by the robot and the feature map that is computed for self, ball, and opponent localization.

nents. From the captured camera images the feature detectors extract problem-specific feature maps that correspond to (1) static objects in the environment including the goal, the borders of the field, and the lines on the field, (2) a color blob corresponding to the ball, and (3) the visual features of the opponents. The working horse of the perception component are a color classification and segmentation algorithm that is used to segment a captured image into colored regions and blobs (see Fig. 3). The color classifier is learned in a training session before tournaments in order to adapt the vision system to specific lighting conditions and effects. We are currently working on the next version of the classifier, which will be capable of automatically adjusting itself to changing lighting conditions during the game.

The color segmented image is then processed by a feature extraction algorithm that estimates the 3D positions Ψ and the covariances C_ψ of the objects of interest. At present it is assumed that the objects are colored black, constructed in the same way and have approximately circular shape. Object detection is performed on the basis of blob analysis. The position of an object is estimated on the basis of a pinhole camera model. Due to rotations and radial distortions of the lenses this model is highly non-linear. The uncertainty estimation process is based on the unscented transformation (Julier & Uhlmann 1997). This allows the use of non-linear measurement equations, the incorporation of parameters describing the measurement uncertainty of the sensor at hand as well as an efficient way of propagating the uncertainty of the observing robots pose. A detailed description of the feature extraction algorithm and uncertainty estimation process can be found in (Schmitt *et al.* 2001).

Implementation and Parameter Selection

Every robot of the soccer team is equipped with its own computer (Pentium III 500 MHz). On this computer several processes are running in order to perform the image processing, path planning, action selection and object tracking. Currently every robot processes approx. 15 frames per second. From every frame a set of opponent observations is extracted and sent via a wireless Ethernet to all other robots of the team. Every robot iterates the MHT algorithm once for every set of opponent observations (own and team mates) and generates a new world state. This state serves as input for action selection and path planning.

In the following we will discuss the remaining parameters of the MHT algorithm. The probability of detection P_D represents the probability that a track is detected by an observing robot, if it is within it's field of view. Special image interpretation routines handle the case of partially occluded objects by detecting and analyzing cascaded object blobs. Later in section "Predictive Models" we will look at the issue of providing a more informative probabilistic model for P_D. During the experiments in section "Results from the RoboCup" we have set P_D to 0.9.

By default we set the termination likelihood λ_T to 40. This allows an unconfirmed hypothesis to survive for approx. 2 seconds. Remembering objects that are currently not visible is important to avoid collisions and thereby reduce the risks of being charged for fouls. The mean rates of false observations λ_F and new tracks λ_N are 0.0002 and 0.04 respectively. This means that no observations are ignored. The depth N of the tracking tree is set to four. This is the minimal depth that allows each team mate to contribute an observation to the hypothesis fusion process. The empirical investigations will show that the update-times of the MHT within our application domain are fast enough to handle the observations of all four robots. Thus the first set of global hypotheses, after the initialization of the MHT, is already created after every robot has performed only one local opponent observation. Since these observations are performed in parallel and integrated through the MHT in real time (before the next opponent observations are performed) a depth of four contributes sustainable to the quality of the global opponent hypotheses. The maximum number of hypothesis was limited to 50 and the value for ratio pruning was set to 0.001.

Empirical Investigation

The multiple object tracking algorithm described in this paper has been employed by our AGILO robot soccer team in the fifth robot soccer world championship in Seattle (2001). The team has played six games for a total of about 120 minutes. The team advanced to the quarter finals. Unfortunately, in middle size robot soccer there is no external sensing device which records a global view of the game and can be used as the ground truth for experiments. Thus for the experimental results in this section we can only use the subjective information of our robots and argue for the plausibility of their behavior and belief states. To do so, we have written log files and recorded the games using video cameras in order to evaluate our algorithm.

Results from the RoboCup

The analysis of the log files from RoboCup 2001, revealed that an average MHT update takes between 6 to 7 msecs. This allows our implementation to process all observations of all robots (max. frame rate: 25Hz) in real time. The minimum and maximum iteration times were measured to be

1.1 msecs and 86 msecs respectively. On average the MHT tracked 3.2 opponents. This is a reasonable number since there are maximal 4 opponent players and players can be send off or have to be rebooted off field. In breaks of the games (when people get on to the field) or when there are crowds of robots the MHT successfully tracked up to 11 objects. A typical result of the AGILO game state estimator is

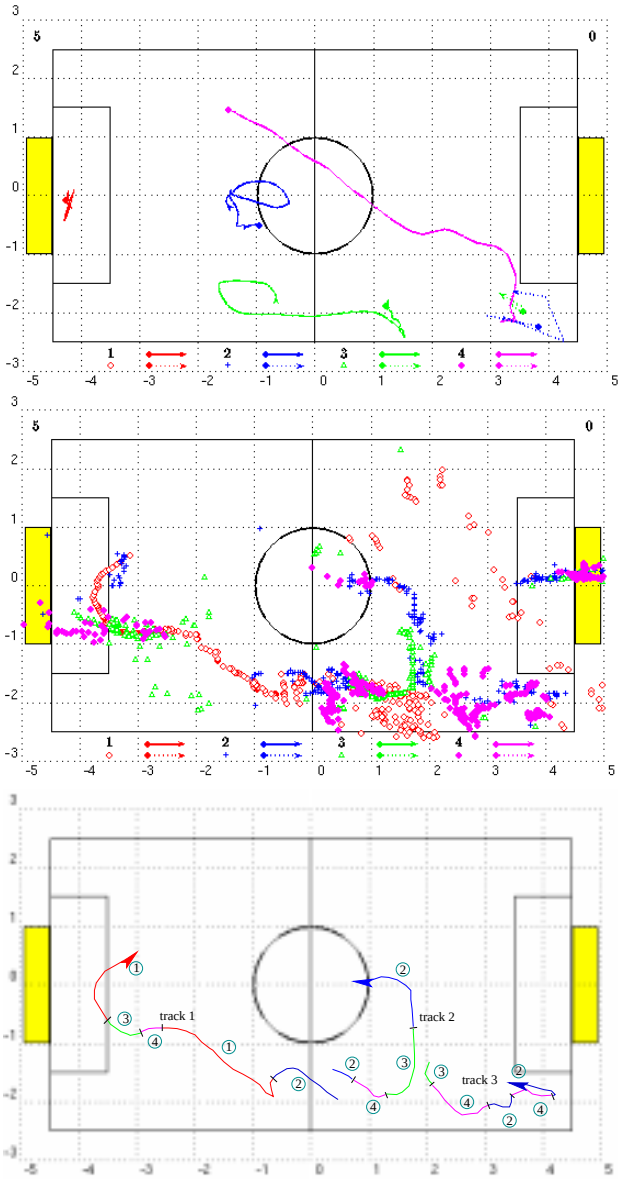


Figure 4: Opponent observations and resolved tracks.

shown in Fig. 4. The upper picture shows the positions of the AGILO players of the own team, computed through vision-based self localization (Hanek & Schmitt 2000). The middle picture shows the individual observations of the opponent robots. The tokens indicate which AGILO robot made the observations. It is good visible how the individual observations of the different robots are merged into a consistent track. In the lower picture the tracks as they were resolved by the MHT are displayed. They are divided into subsections. The number of the robot that contributed the most observations to this part of the track is denoted next to the track.

Qualitatively, we can estimate the accuracy of the game state estimation by looking for the jumps in the tracked lines. The tracks of the opponents look very reasonable. They are less accurate and sometimes only partial. This is due to the high inaccuracy and incompleteness of the sensory data. However, it is observable that several tracks resulted from merging the observations of different robots. In addition, the merging of the different observations results in fewer hallucinated obstacles and therefore allows for more efficient navigation paths. Several wrong opponent observations made by the goal keeper (1) were correctly omitted by the MHT and not assigned to a track. We have cross checked the tracks computed by the algorithm using video sequences recorded during the matches. The tracks are qualitatively correct and seem to be accurate. A more thorough evaluation is only possibly based on the ground truth for the situations. We are currently implementing tracking software for a camera mounted above the field that allows us to compute the ground truth for the next RoboCup championship.

The cooperation of the different robots increases both, the completeness and the accuracy of state estimation. Accuracy can be substantially increased by fusing the observations of different robots because the depth estimate of positions are much more inaccurate than the lateral positions in the image. This can be accomplished through the Kalman filter's property to optimally fuse observations from different robots into global hypotheses with smaller covariances.

The completeness of state estimation can be increased because all the robots can see only parts of the field and can be complemented with observations of the team mates. The other effect we observed was that cooperation allowed to maintain the identity of opponent players over an extended period of time, even though the field of view of the observing robots is limited. This point is well illustrated in Fig. 4. The three opponent field players were tracked successfully over a period of 30 seconds.

A Predictive Model for the Use of Observations

In section "Applying MHT to Robot Soccer" we have assumed that the probabilities needed as parameters by the MHT can be simply supplied by the programmer. In this section we want to take a more thorough look at this issue. The first characteristic of our parameterization is that we have provided the different parameters as a priori probabilities. This does not match our intuitions because we would expect, for example, that observations of the robots are much more inaccurate and unreliable in situations where the robot turns quickly, mainly because of time delays between capturing the camera image and taking odometric readings.

To test this hypothesis we first defined a set of features which we expected to influence the quality of observations, and then learned a decision tree[1] in order to predict whether

[1] Given a feature language and a set of examples Quinlan's C4.5 is capable of devising a set of rules of the form <u>if</u> $f_1 \wedge \ldots \wedge f_2$ <u>then</u> $class$.

an observation made in a given context can be expected to be reliable and accurate. Again the lack of ground truth is hampering a thorough investigation of this issue. Therefore we used a much weaker notion than reliability and accuracy. We take as the concept to be learned whether the observation can be expected to contribute to a long track or not. We chose this concept under the assumption that, if an observation can be assigned to a long and well established track, then it must have features identifying it as a good observation.

In order to describe an observation the following feature language $F = \{R, T, V, W, O, D, U\}$ was used. R specifies the number of the observing robot. The observation type, T, describes whether an observation is an unknown robot, an opponent robot and a cascaded robot. V and W are continuous features representing the translational and rotational velocities of the observing robot. The observational angle O, and the distance D describe the relative polar coordinates of an observation within the field of view. The continuous feature U represented the area of the covariance ellipse of an observation. These features were computed for every observation and an off-line run of the MHT filter with hypothesis trees of depth $N = 20$ generated positive and negative training examples. We then applied C4.5 to this set, which derived the following rules:

<u>if</u> $W \leq 41$ <u>then</u> $use\ observation\ for$ MHT
<u>if</u> $W > 41 \wedge D > 2.4$ <u>then</u> $use\ observation\ for$ MHT
<u>if</u> $W > 41 \wedge D < 2.4$ <u>then</u> $discard\ observation$

The rules state that an observation of an object closer than 2.4 meters performed by an robot rotating with more than 41 degrees per second should be discarded. This rule is reasonable as the image processing routines are computational expensive and observations of close objects with high rotational velocities are errors-prone. All other features, except D and W, were omitted by C4.5 and were not included in the decision rules.

This is a strong indication that the only features influencing the quality of an observation are D and W. One rationale for this is, that all robots are equipped with identical sensors. Furthermore, these results indicate that the performance of the algorithm can be greatly improved by substituting the a priori probabilities through probabilities conditioned on the context of observations. To obtain these probabilities we need more accurate sensing models. These can be learned by using the camera providing the ground truth and computing the probability distributions over observations based on experiences acquired in the soccer games.

Related Work

Related work comprises work done on object tracking in the robot soccer domain and probabilistic and vision-based tracking of moving targets. To the best of our knowledge no probabilistic state estimation method has been proposed for tracking the opponent robots in robot soccer or similar application domains. Dietl, Guttmann and Nebel 2001 estimate the positions of the opponents and store them in the team world model but they do not probabilistically integrate the different pieces of information. Probabilistic tracking of multiple moving objects has been proposed by(Schulz *et al.* 2001). They apply sample-based JPDAF estimation to the tracking of moving people with a moving robot using laser range data. The required computational power for the particle filters is opposed by the heuristic based pruning strategies of the MHT algorithm. Hue et al. 2000 are also tracking multiple objects with particle filters. In their work data association is performed on the basis of the Gibbs sampler. Our approach to multiple hypothesis tracking is most closely related to the one proposed by Cox and Hingorani (Cox & Hingorani 1996). Indeed our algorithm is based on their implementation. We extend their work on multiple hypothesis tracking in that we apply the method to a much more challenging application domain where we have multiple moving observers with uncertain positions. In addition, we perform object tracking at an object rather than on a feature level.

Conclusions

In this paper, we have extended and analyzed a probabilistic object tracking algorithm for a team of vision-based autonomously moving robots. Our results suggest that purely image-based probabilistic estimation of complex game states is feasible in real time even in complex and fast changing environments. Finally, we have seen how the state estimation modules of individual robots can cooperate in order to produce more accurate and reliable state estimation. Besides an empirical analysis of the parameter settings and learning accurate sensing models, we intend to compare in future work the MHT algorithm with the JPDAF implementation of (Schulz *et al.* 2001).

References

Bar-Shalom, Y., and Fortmann, T. 1988. Tracking and data association. Ac. Press.

Beetz, M.; Buck, S.; Hanek, R.; Schmitt, T.; and Radig, B. 2002. The AGILO autonomous robot soccer team: Computational principles, experiences, and perspectives. In *AAMAS*

Cox, I., and Hingorani, S. 1996. An Efficient Implementation of Reid's Multiple Hypothesis Tracking Algorithm and Its Evaluation for the Purpose of Visual Tracking. *IEEE PAMI* 18(2):138–150.

Dickmanns, E. D. 1997. Vehicles capable of dynamic vision. In *IJCAI*, 1577–1592.

Dietl, M.; Gutmann, J.-S.; and Nebel, B. 2001. Cooperative sensing in dynamic environments. In *IROS*, 1706–1713.

Hanek, R., and Schmitt, T. 2000. Vision-based localization and data fusion in a system of cooperating mobile robots. In *IROS*, 1199–1204.

Hue, C.; Le Cadre, J.-P.; and Perez, P. 2000. Tracking multiple objects with particle filtering. Technical Report 1361, IRISA.

Julier, S., and Uhlmann, J. 1997. A new extension of the kalman filter to nonlinear systems. The 11th Int. Symp. on Aerospace/Defence Sensing, Simulation

Reid, D. 1979. An algorithm for tracking multiple targets. IEEE Trans. on Automatic Control, 24(6):843–854.

Schmitt, T.; Hanek, R.; Buck, S.; and Beetz, M. 2001. Cooperative probabilistic state estimation for vision-based autonomous mobile robots. In *IROS*, 1630–1638.

Schulz, D.; Burgard, W.; Fox, D.; and Cremers, A. 2001. Multiple object tracking with a mobile robot. In *CVPR*, 371–377.

Thrun, S.; Beetz, M.; Bennewitz, M.; Cremers, A.; Dellaert, F.; Fox, D.; Hähnel, D.; Rosenberg, C.; Roy, N.; Schulte, J.; and Schulz, D. 2000. Probabilistic algorithms and the interactive museum tour-guide robot minerva. *International Journal of Robotics Research* 19(11):972–999.