Reactive Visual Control of Multiple Non-Holonomic Robotic Agents *

Kwun Han **Manuela Veloso**
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

We have developed a multiagent robotic system including perception, cognition, and action components to function in a dynamic environment. The system involves the integration and coordination of a variety of diverse functional modules. At the sensing level, our complete multiagent robotic system incorporates detection and recognition algorithms to handle the motion of multiple mobile robots in a noisy environment. At the strategic and decision-making level, deliberative and reactive components take in the processed sensory inputs and select the appropriate actions to reach objectives under the dynamic and changing environmental conditions. At the actuator level, physical robotic effectors execute the motion commands generated by the cognition level. In this paper, we focus on presenting our approach for reactive visual control of multiple mobile robots. We present a tracking and prediction algorithm which handles visually homogeneous agents. We describe our non-holonomic control for single robot navigation, and show how it applies to dynamic path generation to avoid multiple moving obstacles. We illustrate our algorithms with examples from our real implementation. Using the approaches introduced, our robotic team won the RoboCup-97 small-size robot competition at IJCAI-97 in Nagoya, Japan.

1 Introduction

Our multiagent robotic system addresses the robotic soccer task. Robotic soccer offers a complex environment in which multiple agents need to collaborate in the presence of adversaries to achieve specific objectives. Robotic soccer offers a challenging research domain to investigate a large spectrum of issues of relevance to the development of complete autonomous agents [1].

The fast-paced nature of the domain necessitates real-time sensing coupled with quick behaving and decision making. The behaviors and decision making processes can range from the most simple reactive behaviors, such as moving directly towards the ball, to arbitrarily complex reasoning procedures that take into account the actions and perceived strategies of teammates and opponents.

We have been pursuing research in the robotic soccer domain within the RoboCup initiative [5], which, in 1997, included a simulator league and small-size and medium-size robot leagues. Our research involves both the simulator league and the small-size robot league [10]. In this paper, we focus on the small-size robot league presenting our approach for reactive visual control of multiple objects and results. Our team, CMUnited, won the RoboCup-97 small-robot competition at IJCAI-97 in Nagoya, Japan.

2 System Overview

The small-size robot setup is viewed as an overall complete autonomous framework composed of the physical navigational robotic agents[1], a video camera over-looking the playing field connected to a centralized interface computer, and several clients as the minds of the small-size robot players. Figure 1 sketches the building blocks of the architecture.

The complete system is fully autonomous consisting of a well-defined and challenging processing cycle. The global vision algorithm perceives the dynamic environment and processes the images, giving the positions of each robot and the ball. This information is sent to an off-board controller and distributed to the different agent algorithms. Each agent evaluates the world state and uses its strategic knowledge to decide what to do next. Actions are motion commands that are sent by the off-board controller through radio fre-

*We thank Peter Stone and Sorin Achim for their contributions to the development of the robotic team. This research is sponsored in part by the Defense Advanced Research Projects Agency (DARPA), and Rome Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-95-1-0018 and in part by the Department of the Navy, Office of Naval Research under contract number N00014-95-1-0591. Views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing official policies or endorsements, either expressed or implied, of the Air Force, of the Department of the Navy, Office of Naval Research or the United States Government.

[1]For hardware details and specifications of the robots, please see [11].

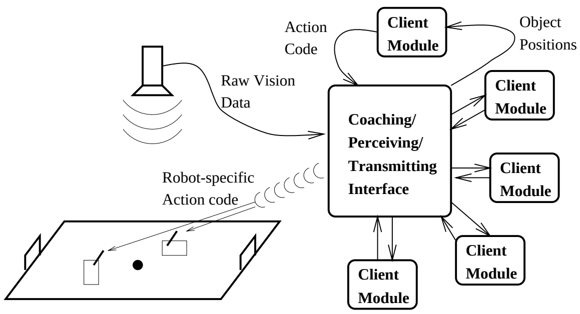

Figure 1: CMUnited Architecture with Global Perception and Distributed Reaction.

quency communication. Commands can be broadcast or sent directly to individual agents. Each robot has an identification binary code that is used on-board to detect commands intended for that robot. Motion is not perfectly executed due to inherent mechanical inaccuracies and unforeseen interventions from other agents. The effects of the actions are therefore uncertain. This complete system is fully implemented.

The physical robots themselves are of size $15cm \times 12cm \times 10cm$. Figure 2 shows our robots. Differential drive mechanism is used in all of the robots. Two motors with integrated gear boxes are used for the two wheels. Differential drive was chosen due to its simplicity. The size constraints do not allow the realization of other more complex drive mechanisms. The size of our robots conforms to RoboCup Competition rules[2]. Employing the differential drive mechanism means that the robot is non-holonomic, which makes the robot control problem considerably more challenging.

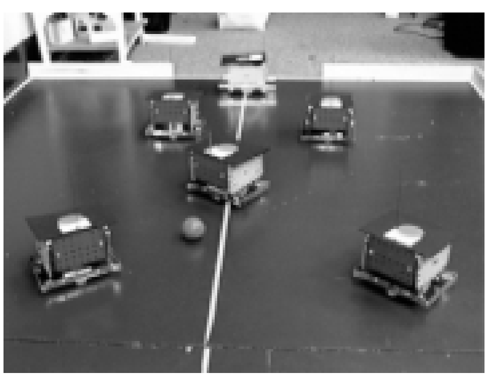

Figure 2: Our robots, showing the two color patches on top of the robots and the differential drive wheels at the sides.

Although it may be possible to fit an on-board vision system onto robots of small size, in the interest of being able to quickly move on to strategic multiagent issues, we have opted for a global vision system.

The fact that perception is achieved by a video camera overlooking the complete field offers an opportunity to get a global view of the world state. Although this setup may simplify the sharing of information among multiple agents, it presents a challenge for reliable and real-time processing of the movement of multiple mobile objects — the ball, five robots on our team, and five robots in the opponent team.

3 Real-Time Perception for Multiple Agents

This section focusses on presenting our vision processing algorithm whose accuracy makes it a major contribution towards the success of our robotic team.

3.1 Color-based Detection

The vision requirements for robotic soccer have been examined by different researchers [7, 8]. Systems with on-board and off-board types have appeared in recent years. All have found that the reactiveness of soccer robots requires a vision system with a high processing cycle time. However, due to the rich visual input, researchers have found that dedicated processors or even DSPs are often needed [2, 7]. Our current system uses a frame-grabber with frame-rate transfer from a 3CCD camera. A relatively slow processor (166MHz Pentium) was at the heart of the vision system, performing all vision computations.

The RoboCup rules specify well defined colors for different objects in the field and these are used as the major cue for object detection. The RoboCup rules specify a green color field with specific white markings. The ball is an orange golf ball (see Figure 2). It also specifies a yellow or blue colored circle on the top of the robots, one color for each team. A single color patch on the robot is not enough to provide orientation information. Thus, an additional pink color patch was added to each robot. These colors can be differentiated reliably in color-space.

The set of detected patches are unordered. The detected color patches on the tops of the robots are then matched by their distance. Using the constant distance between the team-color (blue or yellow) and the pink orientation patch, our detection algorithm matches patches that are this distance apart. Two distance-matched patches are detected as a robot.

Noise is inherent in all vision systems. False detections in the current system are often of a magnitude of 100 spurious detections per frame. The system eliminates false detections via two different methods. First, color patches of size not matching the ones on the

[2] see http://www.robocup.org/RoboCup/

robots are discarded. This technique filters off most "salt and pepper" noise. Second, by adding the distance matching mechanism described above, all false detections are eliminated.

3.2 Data Association

The detection scheme described above returns an unordered list of robots for each frame. To be able to control the robots, the system must associate each detected robot in the field with a robot identification.

Each of the robots is fitted with the same color tops and no attempts are made to differentiate them via color hue. Experience has shown that, in order to differentiate 5 different robots by hue, 5 significantly different hues are needed. However, the rules of the RoboCup game eliminate green (field), white (markings), orange (ball), blue and yellow (team and opponent) from the list of possibilities. Furthermore, inevitable variations in lighting conditions over the area of the field and noise in the sensing system are enough to make a hue-based detection scheme impractical.

With each robot fitted with the same color, visually, all robots in each team look identical to the visual system. Detecting and differentiating multiple objects simultaneously is trivial if the objects were non-homogeneous. However, problems arise when the objects to be identified are homogeneous such that distinguishing between them is not possible.

Data association addresses the problem of retaining robot identification in subsequent frames. We devised an algorithm to retain association based on the spatial locations of the robots.

We assume that the starting positions of all the robots are known. This can be done trivially by specifying the location of the robots at start time. However, problems arise when subsequent frames are processed, the locations of the robots have changed due to robot movements. Association can be achieved by making two complementary assumptions: 1) Robot displacements over consecutive frames are local; 2) The vision system can detect objects at constant frame rate. By measuring the maximum robot velocity, we can know that in subsequent frames, the robot is not able to move out of a 3cm radius circular region. This provides the basis of our association technique.

3.3 Greedy Association

With these assumptions in mind, a minimum distance scheme can be used to retain association between consecutive frames.

During consecutive frames, association is maintained by searching for objects within a minimum displacement. Current robot positions are matched with the closest positions from the previous frame.

The following is the pseudo-code of a greedy association procedure:

let $prev[1..n]$ be the array of robot locations
  from the previous frame
let $cur[1..m]$ be the array of robot locations
  from the current frame
let ma be triangular array of size $n-1$ s.t.
$ma[i][j] = \text{dist}(prev[i], cur[j])$
for $i := 1$ to m **do**
  find smallest element $ma[i][j]$
  save (i, j) as a matched pair
  set all elements in row i and column j to be ∞
end
if $m < n$ **then**
  forall $prev[i]$ unmatched, save $(prev[i], prev[i])$
return the set of saved pairs as the set of matchings.

This algorithm searches through all possible matches, from the smallest distance pair upwards. Whenever a matched pair is found, it greedily accepts it as a matching pair.

Due to noise, it is possible for the detection system to leave a robot or two undetected (i.e.. in the pseudo-code $m < n$). In this case, some locations will be left unmatched. The unmatched location will then be carried over to the current frame, and the robots corresponding to this location will be assumed to be stationary for this one frame.

This algorithm was implemented and was used in RoboCup97. Although the implementation was very robust, we present an improvement that allows for a globally optimal associatio.

3.4 Globally Optimal Association

The greedy association algorithm, as described above, fails in some cases, for example as the one illustrated in Figure 3. In the figure, a greedy algorithm incorrectly matches the closest square and circular objects.

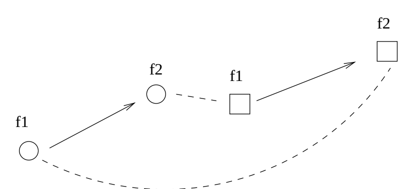

Figure 3: A case in which greedy association fails but global optimal association performs well. The arrow indicates the actual object displacement over subsequent frames f1 and f2. The dotted lines indicate the wrong matches returned by greedy association.

An improved algorithm was devised to handle the situation depicted above. The new algorithm generates all possible sets of matching and calculates the fitness of the each of the sets globally according to the following least square criteria:

$$\sum_{i=1}^{N}(dist(prev_i, cur_i))^2,$$

where $(prev_i, cur_i)$ are the i^{th} matching pair. And the function $dist(x, y)$ is the Euclidean distance. The set of matches that minimizes the above criteria is selected as the best matches.

While these algorithms do not theoretically guarantee perfect associations, in particular with noisy perception and cluttered environments, the implementation has proved to be very robust.

3.5 Tracking and Prediction

In the setting of a robot soccer game, the ability to detect merely the locations of objects on the field is often not enough. Like for real soccer players, it is often essential for robots to predict future locations of the ball (or even of the other players). We have used an Extended Kalman filter (EKF) for such a purpose[4], which is very suitable since the detection of the ball's location is noisy.

The EKF is a recursive estimator for a possibly non-linear system. It involves a two-step iterative process, namely *update* and *propagate*. The current best estimate of the system's state and its error covariance is computed on each iteration. During the update step, the current observations are used to refine the current estimate and recompute the covariance. During the propagate step, the state and covariance of the system at the next time step are calculated using the system's equations. The process then iteratively repeats, alternating between the update and the propagate steps.

We capture the ball's state into 5 variables: the ball's x and y location, the ball's velocities in the x and y direction and a friction parameter (λ_k) for the surface.

These variables are related via the following set of non-linear difference equations:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \dot{x}_{k+1} \\ \dot{y}_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + \dot{x}_k \cdot \Delta t \\ y_k + \dot{y}_k \cdot \Delta t \\ \dot{x}_k \cdot \lambda_k \\ \dot{y}_k \cdot \lambda_k \\ \lambda_k \end{bmatrix}$$

The above equations model the ball with simple Newtonian dynamics. λ_k is a friction term which discounts the velocity at each time step. Δt is the time-step size.

The prediction equations are:

$$\begin{aligned} x_{k+n} &= x_k + \dot{x}_k \cdot \Delta t \cdot \alpha_{kn} \\ y_{k+n} &= y_k + \dot{y}_k \cdot \Delta t \cdot \alpha_{kn} \\ \alpha_{kn} &= \begin{cases} 1, & \text{if } \lambda_k = 1 \\ (1-(\lambda_k)^n)/(1-\lambda_k), & \text{otherwise} \end{cases} \end{aligned}$$

The prediction equations are derived by solving the recursive equation obtained by substituting the value of x_{k+i} where i decreases from n to 1. We are only interested in the predicted spatial location of the ball thus we do not explicitly calculate the predicted velocity.

Through a careful adjustment of the filter parameters modelling the system, we were able to achieve successful tracking and, in particular prediction of the ball trajectory, even when sharp bounces occur.

Our vision processing approach worked perfectly during the RoboCup-97 games. We were able to detect and track 11 objects (5 teammates, 5 opponents and a ball) at 30 frames/s. The prediction provided by the EKF allowed the goal-keeper to look ahead in time and predict the best defending position. During the game, no goals were suffered due to miscalculation of the predicted ball position.[3]

4 Single-Agent Control

In order to be able to successfully collaborate, agents require robust basic skills. These skills include the ability to go to a given place on the field. The ability to direct the ball in a given direction, and the ability to intercept a moving ball are all built on top of this simple behavior.

The non-holonomic path planning problem has been addressed by many researchers, e.g., [6, 3]. However, most of the algorithms deal with static worlds and generate pre-planned global paths. In the robot soccer domain, this is not possible for the following reasons: 1) The domain is inherently dynamic, as robots move around the field; 2) The fast reaction time (less than 1/150 second for each robot) needed during a game prohibits the use of computationally expensive algorithms. Furthermore, the robots mechanics are noisy and possible interference from other robots occurs (e.g., pushing), making precisely mapped out paths ineffective and unnecessary.

We devised and implemented a reactive controller for our system, which has the following advantages: 1) It is computationally inexpensive; 2) It by nature deals with dynamic environments; 3) It recovers from noisy command execution and possible interferences.

[3] In RoboCup-97 we scored a total of thirteen goals and only suffered one.

A reactive controller also has possible disadvantages: 1) It may get caught in a local path space minima; 2) It may generate sub-optimal paths. However, the advantage outweighs the disadvantage. To handle the possibility of failure, a failure recovery routine was devised.

4.1 Reactive Non-Holonomic Control

The navigational movement control is done via reactive control. The control rules described below are non-linear control strategy that smoothly guides the robot to face the target and maintain a smooth transition between the rotational and translational velocity components.

$$(v, \dot{\theta}) = \begin{cases} (\alpha \cdot \cos\theta, \beta \cdot \sin\theta) & \text{if } |\theta| < \frac{\pi}{4} \text{ or } |\theta| > \frac{3\pi}{4} \\ (0, \mathrm{sgn}(\theta) \times \beta_0) & \text{otherwise} \end{cases}$$

where v and $\dot{\theta}$ are the desired translational and rotational velocities, respectively, θ is the direction of the target relative to the robot ($-\pi < \theta < \pi$), β_0 is the in place rotational velocity, and α and β are the base translational and rotational velocities, respectively. The translational and rotational velocities can be translated to differential drive parameters via a simple, invertible linear transform. This set of control formulae differs from the love vehicle in that it takes into account the orientation of the robot with respect to the target and explicitly adds rotational control. This is illustrated in Figure 4.

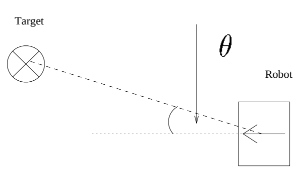

Figure 4: The geometry of the robot control algorithm. θ is the direction of the target relative to the robot.

This set of control rules implicitly allows for heading independence, i.e., the control rule allows for both forward and backward movements, whichever one is most efficient to execute.

Figure 5 shows an actual run of the reactive control algorithm described above. The target point is marked with a cross.

5 Dynamic Path Generation

The multi-object tracking, the prediction of the ball's direction and speed, and the effective reactive non-holonomic control for single robot navigation are of great relevance to the successful performance of our team of robots [10]. In particular, we developed

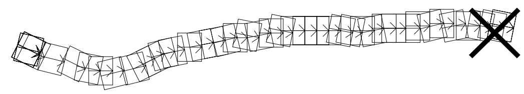

Figure 5: Sample trace of the execution of the reactive control algorithm. The target point is marked with a cross.

a dynamic path generation algorithm that applies to highly dynamic environments, such as robotic soccer.

The path planning problem in a dynamic environment and related problems has been addressed by many researchers [3, 9]. The requirement that we have is more strigent in the way that our domain requires that we must control multiple non-holonomic robots in a environment with dynamic obstacles and half of which are adversaries who we cannot control nor know the behaviour of.

In the robotic soccer field, there are always multiple non-static robots between a robot and its target location. Our robots try to avoid collisions by planning a path around the other obstacles. Due to the highly dynamic nature of the environment, our obstacle avoidance algorithm uses closed-loop control by which the robots continually replan their goal positions around obstacles. The algorithm consists of an incremental generation of different possible intermediate target points. The selection of which intermediate target point to go to is based on an evaluation of the obstructness of the paths through it. In a nutshell, our dynamic path generation algorithm aims at finding an intermediate target point that both deviates as little as possible from its straight line to the final target, and that avoids in the best way the detected obstacles considered. Figure 6 illustrates a typical situation where the algorithm chooses to go around the obstacle through the side that is less crowded with obstacles.

The algorithm takes into account, i.e., incorporates lookahead for a pre-specified number of obstacles. Intermediate target points are generated at increasing distances from each detected obstructing obstacle to allow for the robot to go around multiple objects, according to its lookahead. The algorithm takes also into account the walls of the field in its generation and evaluation of the intermediate target points. A lookahead of a single obstacle actually also produces good results, as the environment is rather dynamic and other robots are continually moving around.

Multiple robots have different target locations, they employ the same algorithm which together plans a set of path which lead the robots to their targets. However, even with this algorithm in place, the robots can

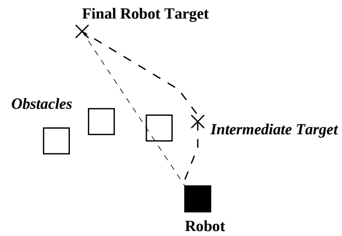

Figure 6: The robot starts by trying to go towards its final target along a straight line. When it comes across an obstacle within a certain distance of itself, it aims at an intermediate target, considering several obstacles according to a pre-specified lookahead. Using reactive control to approach this dynamic environment of moving obstacles, the robot continually recomputes its path until it obtains an unobstructed path to the final target.

occasionally get stuck against other robots or against the wall. Particularly if adversarial robots do not use obstacle avoidance, collisions are inevitable. When unable to move, our robots identify the source of the problem as the closest obstacle and "unstick" themselves by moving away. Once free, normal control resumes.

Figure 7 shows an actual run of the dynamic path generation algorithm described above. The target point is shown with a cross. The obstacle is another robot sitting between the robot's initial location and the target point.

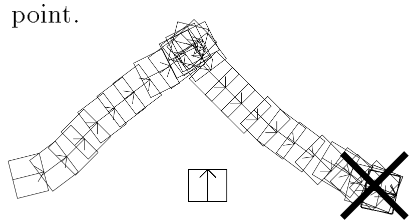

Figure 7: Execution trace of a robot reaching the target while avoiding an obstacle (another robot).

6 Conclusion

We presented the reactive visual control algorithm of our robotic soccer team. The team consists of five non-holonomic robots which need to operate in a dynamic and noisy environment. We introduced our association and tracking approach that allows for the reliable continuous identification of robots that are visually homogeneous. We also introduce a non-holonomic control that reactively plans the robot trajectory, by using translation and rotational motion parameters. Our team is fully implemented and we illustrated our algorithms with examples from real runs of the robots.

For a video of our robot team, including its performance in the RoboCup'97 games, please visit our Homepage in http://www.cs.cmu.edu/˜robosoccer.

References

[1] Minoru Asada, Yasuo Kuniyoshi, Alexis Drogoul, Hajime Asama, Maja Mataric, Dominique Duhaut, Peter Stone, and Hiroaki Kitano. The RoboCup physical agent challenge: Phase-i. *Applied Artificial Intelligence*, 12, 1998.

[2] Minoru Asada, Shoichi Noda, Sukoya Tawaratumida, and Koh Hosoda. Purposive behavior acquisition for a real robot by vision-based reinforcement learning. *Machine Learning*, 23:279–303, 1996.

[3] Kikuo Fujimura. *Motion Planning in Dynamic Environments*. Springer-Verlag, 1991.

[4] Kwun Han and Manuela Veloso. Physical model based multi-objects tracking and prediction in robosoccer. In *Working Note of the AAAI 1997 Fall Symposium*. AAAI, MIT Press, 1997.

[5] Hiroaki Kitano, Yasuo Kuniyoshi, Itsuki Noda, Minoru Asada, Hitoshi Matsubara, and Eiichi Osawa. RoboCup: A challenge problem for AI. *AI Magazine*, 18(1):73–85, Spring 1997.

[6] Jean-Claude Latombe. *Robot motion planning*. Kluwer Academic Publishers, 1991.

[7] Michael K. Sahota, Alan K. Mackworth, Rod A. Barman, and Stewart J. Kingdon. Real-time control of soccer-playing robots using off-board vision: the dynamite testbed. In *IEEE International Conference on Systems, Man, and Cybernetics*, pages 3690–3663, 1995.

[8] Randy Sargent, Bill Bailey, Carl Witty, and Anne Wright. Dynamic object capture using fast vision tracking. *AI Magazine*, 18(1):65–72, Spring 1997.

[9] T. Tsubouchi and S. Arimoto. Behaviour of a mobile robot navigated by an iterated forecast and planning scheme in the presence of multiple moving obstacles. In *Proccedings of IEEE Int. Conf. on Robotics and Automation*, pages 2470–2475, May 1994.

[10] Manuela Veloso, Peter Stone, and Kwun Han. The cmunited-97 robotic soccer team: Perception and multiagent control. In *Submitted to the Second International Conference on Autonomous Agents*, 1998.

[11] Manuela Veloso, Peter Stone, Kwun Han, and Sorin Achim. The CMUnited-97 small robot team. In *Proceedings of the First International Workshop on RoboCup*, Nagoya,Japan, August 1997.