Merging Probabilistic Observations for Mobile Distributed Sensing

Ashley W. Stroupe, Martin C. Martin, Tucker Balch

CMU-RI-TR-00-30

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

December 2000

Acknowledgements

The authors would like to acknowledge several people for their contributions to this work: Jim Bruce (vision), Scott Lenser (localization), Kevin Sikorski (camera calibration), Hans Moravec (statistics and camera calibration), and Manuela Veloso. The authors would also like to acknowledge DARPA's Mobile Autonomous Robot Software Program (MARS) and Control of Agent Based Systems Program (CoABS) for support of this research.

Merging Probabilistic Observations for Mobile Distributed Sensing

Ashley W. Stroupe, Martin C. Martin, Tucker Balch

CMU-RI-TR-00-30

Abstract

We present a method for representing, communicating, and fusing multiple noisy observations of an object made by multiple distributed robots. The approach relies on re-parameterization of the canonical two-dimensional Gaussian distribution to a form that corresponds more naturally to the observation space of a robot. The approach enables two or more observers to achieve greater effective sensor coverage of the environment and improved accuracy in object position estimation. We demonstrate empirically that, when using our approach, more observers achieve more accurate estimations of an object's position. Quantitative evaluations of the technique in use on mobile robots are provided. The method is applied in three application areas, including object location, object tracking, and ball position estimation for robotic soccer. Preliminary qualitative results from these sample applications are discussed.

Table of Contents

1. Introduction

This paper presents a method of distributed sensing from mobile robot platforms. Object positions collected by individual robots are modeled as Gaussian distributions, which are then combined to produce more accurate position estimates. Positions of objects visible to only one robot can be passed along to teammates to increase the effective field of view of all robots.

Typically, individual robots can observe only part of their environment at any moment in time. In dynamic environments, information previously collected about currently unobservable parts of the environment grows stale and becomes inaccurate. Sharing information between robots can increase the effective instantaneous visibility of the environment, allowing for more accurate modeling (at whatever level) and more appropriate response. If it is processed effectively, information collected from multiple points of view can provide reduced uncertainty, improved accuracy, and increased tolerance to single-point failures in estimating the location of observed objects.

In order to meet the time demands of a highly dynamic environment (robotic soccer, for example), the information transmitted between robots and the computational demands to combine their observations must be minimal. Our approach makes use of a few easily obtainable parameters describing an observation and simple computations to meet these needs. We use two-dimensional statistical representations of target location observations generated by individual robots. Each robot independently combines the multiple observations provided it in order to produce improved estimates of target locations (a process interchangeably referred to as *merging*, *fusing*, or *multiplying* in this document). The local computation allows for robust use of all available information with graceful degradation as communication fails.

Potential applications of this method of distributed sensing include localizing landmarks for multi-robot mapping of unknown environments (such as planetary surfaces), precise target localization for multi-robot manipulation (such as soccer or movement of large objects), and localization of lost team members by observing robots.

2. Background And Related Work

Two areas of robotics research are related to this work: localization and distributed sensing. Localization of objects in the world is clearly relevant; additionally approaches to locating a robot within an environment can frequently be adapted to locating other objects. This approach to object localization is based on distributed sensing, making approaches to this also relevant.

Two common probabilistic methods used for localization in robotics are Kalman filters [23] and Monte Carlo Localization[11]. Kalman filters have been used to localize robots within mapped environments. Typically, this approach is used to fuse position estimates from multiple synchronous or asynchronous sensors on a single robot [12][15][17][22]. Monte Carlo localization has been specifically applied to the multiple robot localization problem [11]. While most of these approaches rely on previously obtained maps of the environment, some recent research has been ongoing in simultaneously mapping the environment while localizing the robot within it [4][10]. Kalman filters have also been used for object tracking in several methods. A few examples are: estimation of error between different coordinate frames to determine the relative locations of objects [23], identification and prediction of future object locations [21][24], real-time tracking and camera pointing [25], and tracking for grasp using proximity sensors [19].

The ability to rapidly share distributed observations is critical in distributed dynamic tasks like robotic soccer. Most robot soccer team approaches use vision and/or sonar to localize and vision to locate objects in the environment. Some share information for planning and dynamic role assignment (ART [18]) and others share information to fill in blank areas in the world model (CS Freiburg[12][13], RMIT [6], 5dpo [9]). Sharing and merging overlapping information for object localization has received little attention. Two of the few examples are the use of multiple sequential robot observations as input to Kalman filter tracking [7][8] by Cai et al. and updating of grid cell occupancy probabilities using sequential observations from multiple robots [14] by Hanek et al. Another domain in which multiple, mobile platforms track objects is radar systems. In both military and civilian applications radar signals from aircraft, ships, ground vehicles, and fixed stations may be combined to produce accurate estimates of multiple target positions and velocities. In most radar applications, the merging of data is done at the track level, not at the observation level [3].

The task we address is distinct from the others described above. We focus on fusing multiple simultaneous observations of the same object from distributed vantage points (as opposed to observations from the same vantage point over multiple instants in time). Our objective is to provide more accurate instantaneous estimations of the location of dynamic objects that are simultaneously visible by multiple robots without relying on historical data. Additionally, most probabilistic methods rely on decomposing the space into discrete cells. Our approach does not require discretization of the world into cells and instead works in the continuous spatial domain (to the resolution of the sensors).

3. Fusing Gaussian Distributions

3.1. Overview

We represent a single observation of an object as a two-dimensional Gaussian distribution (Figure 1). The distribution represents the conditional probability that the object is in any location given the observation. The mean of the distribution is the estimated location of the object. The standard deviations along the major and minor axes of the distribution correspond to estimates of the uncertainty (or noise) in the observation along each axis: error in depth determines major axis deviation and error in angle uses absolute distance to determine minor axis deviation using basic trigonometry: $s_{min} = d \sin(s_q)$.

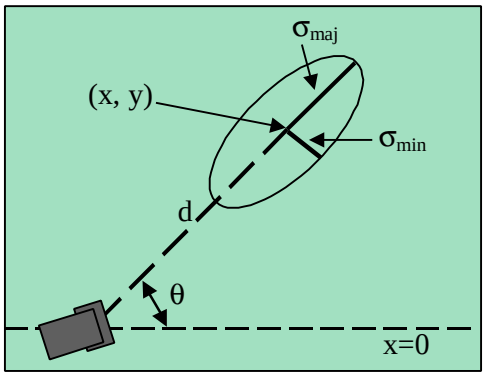

Figure 1. Illustration of the Gaussian distribution parameter definitions. (x, y): position of the mean in global coordinates. q: angle of major axis relative to global x-axis. s_{maj} and s_{min}: standard deviations along major and minor axes, respectively. d: distance from robot to mean.

Provided two observations are independent and drawn from normal distributions, the observations can be merged into an improved estimate by multiplying the distributions. To meet the cycle time requirements of a highly reactive system, an efficient method of multiplying distributions is necessary. We use a two-dimensional statistical approach, first introduced by Duffin [1], based on Bayes' Rule and Kalman filters. In this approach, multi-dimensional Gaussian distributions can be combined using simple matrix operations. Since multiplying Gaussian distributions results in a Gaussian distribution, the operation is symmetric, associative, and can combine any number of distributions in any order.

Our approach is to collect observations of multiple robots, (as in Figure 2), then merge the corresponding Gaussian distributions to yield a better estimate of the location and uncertainty of the observed object. Returning the final result to a coordinate frame aligned with the major and minor axes provides best and worst case uncertainties, and allows a consistent representation for pre- and post-processing estimates.

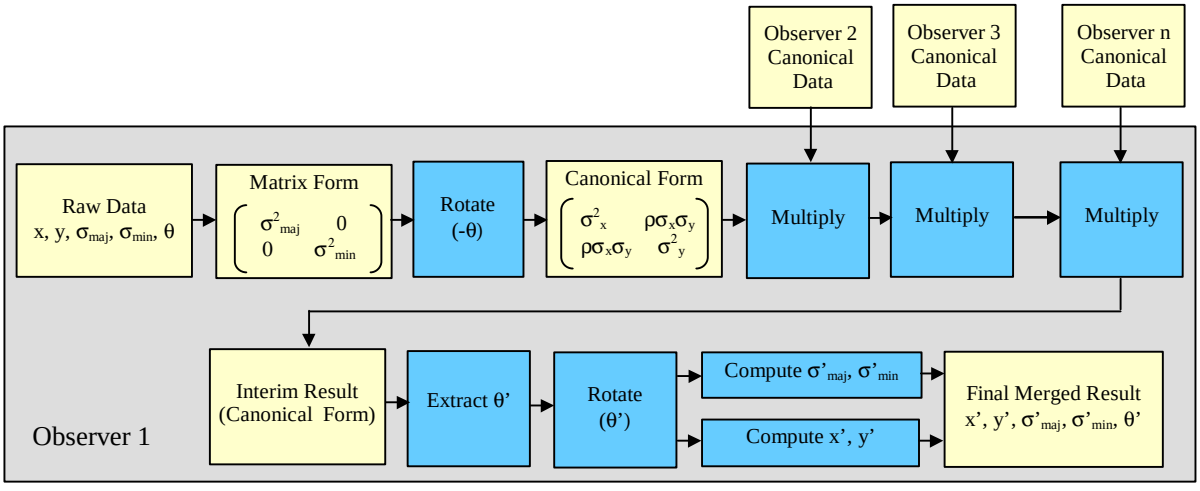

Figure 2. Block diagram of the multi-distribution merging process. Multiplication is conducted using the mathematical formulation described above. Each subsequent distribution is merged with the previous result, and the final parameters are extracted. Blue (dark) boxes represent operations; yellow (light) boxes represent data. Values with an apostrophe (') are post-processing values.

The canonical form of the two-dimensional Gaussian distribution depends on standard deviations, σ, (within the covariance matrix, C), and the mean, $\bar{X}$, as shown [1][23]:

$$p(X) = \frac{1}{2p\sqrt{|C|}} \exp\left(-\frac{1}{2}(X - \bar{X})^T C^{-1}(X - \bar{X}) \right) \quad \text{Eq. 1}$$

where

$$C = \begin{bmatrix} s_x^2 & r s_x s_y \\ r s_x s_y & s_y^2 \end{bmatrix} \quad \text{Eq. 1b}$$

and s_x, s_y, and r are the standard deviations along those axis and the correlation factor, respectively.

The parameterization of the Gaussian distribution in this representation does not correspond the parameters of our observations (Figure 1). We address the problem through a transformation of parameters from observation form to canonical form. In this form, the Kalman filter can be applied to merge the distributions. After the observations are merged, we extract the mean and standard deviations from the merged result (these correspond to the estimated location and uncertainty of the observed object).

3.2. Mathematical Details

We wish to determine the mean, standard deviations, and angle of the combined distribution to estimate object position and characterize the quality of the estimate. We compute the mean, standard deviations, and angle of measurement distributions sensor readings (mean and angle) and models of sensor error (deviations). Thus, we require a method of determining combined parameters from those of individual distributions.

The matrix form of merging distributions adopted by Smith and Cheeseman makes this computation relatively simple [23]. Since the mean, standard deviations, and orientation of the major axis are relative to the shape of the distribution and are independent of the normalization coefficient in front of the exponential, they can be extracted from the resulting merged covariance matrices without computing the distribution.

The covariance matrix of an observation, C, is initially determined from the major and minor axis standard deviations. This matrix is relative to the coordinate frame aligned with the major and minor axes. In this coordinate frame (designated L), the correlation factor is zero.

$$C_L = \begin{bmatrix} s_{maj}^2 & 0 \\ 0 & s_{min}^2 \end{bmatrix} \quad \text{Eq. 2}$$

Since observations may be oriented arbitrarily with respect to the global coordinate frame, they be transformed to this frame. Rotation of X in equation 1 by leads to the following relationship:

$$C^{-1} = R(-q)^T C_L^{-1} R(-q) \quad \Rightarrow \quad C = R(-q)^T C_L R(-q) \quad \text{Eq. 3}$$

where q is the angle of the distribution's principal axis with respect to the global x-axis. This rotation accomplishes the transformation from observation parameters to the canonical form. Once the observation is in the canonical form, we continue with Smith and Cheeseman's approach to merging.

The covariance matrices of two distributions (each from Eq. 3) is combined into a single covariance matrix representing the combined distribution. This covariance is relative to the global x- and y-axes.

$$C' = C_1 - C_1[C_1 + C_2]^{-1} C_1 \quad \text{Eq. 4}$$

The mean of the resulting merged distribution (*x'*, *y'* in Figure 2) is computed from the individual distribution means and covariance matrices.

$$\bar{X} = \bar{X}_1 + C_1\left[C_1 + C_2\right]^{-1}\left(\bar{X}_2 - \bar{X}_1\right) \quad \text{Eq. 5}$$

The resulting major angle axis (θ' in Figure 2) is obtained from the merged covariance matrix (Eq. 4):

$$q' = \frac{1}{2}\tan^{-1}\left(\frac{2B}{A-D}\right) \quad \text{Eq. 6}$$

where *A*, *B*, and *D* represent the components of the covariance matrix (upper left, upper right/lower left, and lower right, respectively).

Lastly, the resulting major and minor axis standard deviations are extracted by rotating the merged covariance matrix (Eq. 4) to align with the merged distribution's major and minor axes:

$$C_L' = R(q)^T C' R(q) \quad \text{Eq. 7}$$

and the resulting major and minor axis standard deviations can be directly computed from the covariance matrix by reversing Equation 2, taking the square root of the non-zero diagonal terms.

The derivations of Equations 3, 4, and 5 are provided in the Appendix.

3.3. Simulated Example

Two robots at different locations observe a target object (Figure 3). Each observation produces a Gaussian distribution of possible locations for the object; typically, each distribution will provide greater accuracy along a different direction than the other distributions.

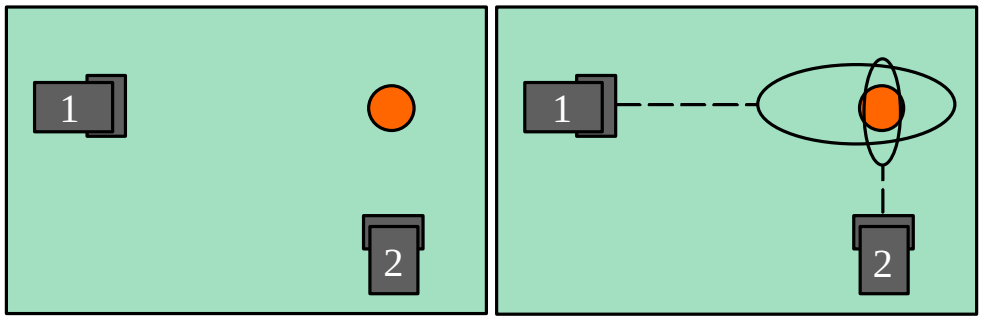

Figure 3. *Left:* Two robots see the same target from different views. *Right:* Observations of the target are modeled as Gaussian distributions; uncertainty (represented by 1-σ ovals) increases with distance.

For this example, the robots are positioned with relative headings 90 degrees apart and looking directly at the target. The target is located at a position (10,10). The two simulated robot observations were drawn from a random normal distribution centered at the object's true position. The major and minor axis standard deviations of these distributions were (5,3) for robot 1 and (3,1) for robot 2. Robot 1 reports a mean of (12.34, 9.02) and Robot 2 reports a mean of (9.90, 11.69). In Figure 4, the distribution resulting from the single measurements by Robot 1 and Robot 2 are shown.

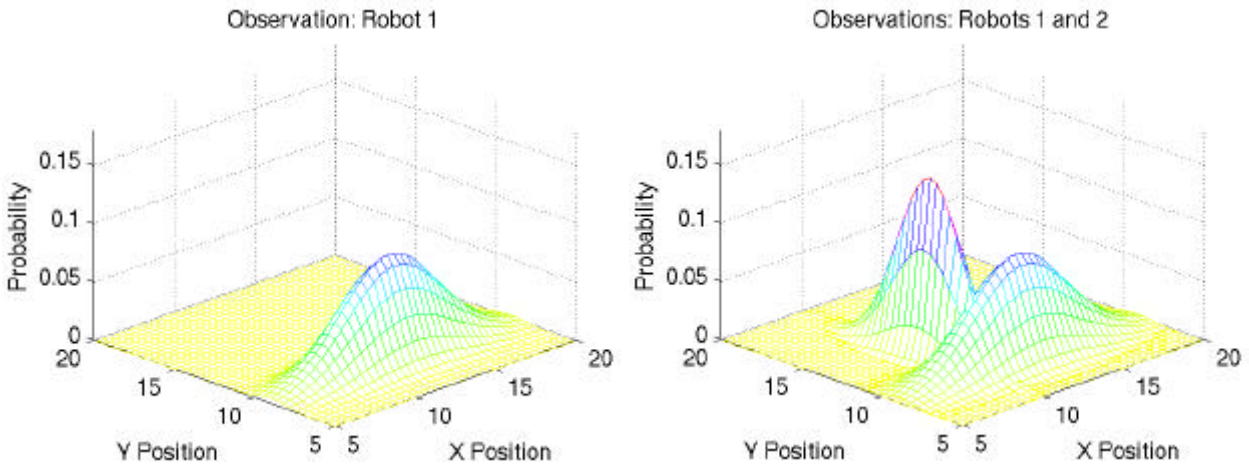

Figure 4. *Above:* Single robot observation distribution. *Below:* Two single robot observations to be merged. Note that robot 1 reports a broader distribution due to larger uncertainty at larger distances. Also note that the uncertainty in the local horizontal dimension is less than that along the local depth dimension.

The distribution resulting from merging the measurement distributions is shown in Figure 5. It is easily observed that the implied uncertainty is reduced, and the resulting distribution is centered more accurately relative to the actual target position. The merged mean is (9.97, 9.57), with major and minor axis standard deviations (0.89, 0.49).

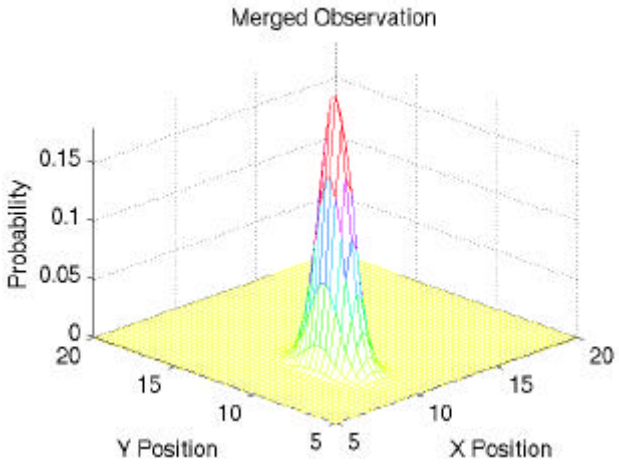

Figure 5. The resulting merged distribution, with smaller standard deviations (error) and higher accuracy in the mean. Note also that the uncertainty along both dimensions is more balanced, as the two observations complement each other.

4. Assumptions

The primary assumption upon which the mathematics is abased is that sensor errors are independent and distributed normally. As demonstrated by the camera calibration procedure (see section 5.3), this assumption approximately holds for this system.

The robot coordinate frames are assumed to be coincident. Without this, data are incompatible and the merging is meaningless.

We assume robots to be perfectly localized. Robot positional uncertainty is not taken into account in the generation of target location distributions, as at the current time our method of localization does not provide uncertainty estimates. If the errors in localization can be modeled as a Gaussian, the mathematics to take this uncertainty into account is similar to that of the merging of measurement distributions. The errors in robot configuration (x, y, q) are geometrically mapped to standard deviations along axes oriented parallel to the major and minor axes of the measurement distribution. This creates a Gaussian approximation to localization error. The standard deviations of the observations can then be "grown" to accommodate positional error by merging it with the position distribution (re-centered at the measurement distribution's mean). Preliminary calibration of the Cye odometry indicates that approximation as a Gaussian distribution is likely reasonable.

Several additional assumptions were introduced for simplicity in experimentation and camera calibration. First, the camera parameter calibration assumes that objects are at a known height from the ground plane; unknown objects are therefore assumed to be on the ground plane. This reduces the transformation from three dimensions to two. This is not a highly restrictive limitation, as common obstacles, agents, landmarks (etc) in environments are generally on the ground plane. Second, objects of interest are assumed to be unique in order to avoid the necessity of solving the association problem.

5. Robotic Hardware Platforms

The robots used in these experiments are designed to be an inexpensive, mechanically homogenous, multi-purpose platform (Figure 6). They are constructed entirely from parts that are commercially available and inexpensive.

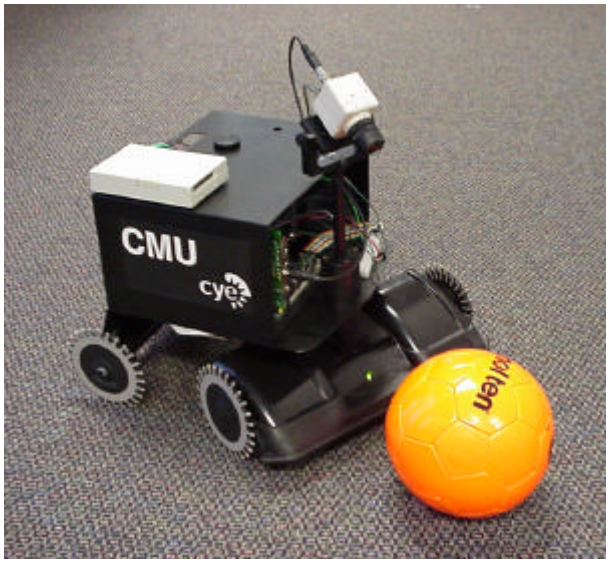

Figure 6. Enhanced Cye Robot platform [20] with soccer ball. The NTSC video camera and wireless network box are visible on top of the robot.

5.1. Cye Robot

The robot platform is based on the Cye robot, an inexpensive platform commercially available from Probotics [20]. This platform consists of a drive section and a trailer. The drive section uses differential drive with two wheels; the wheels are spiked for accurate odometry in x, y, and q. The drive section is additionally equipped with a motor current sensor to detect head-on collisions. The trailer is passive with fixed wheels, similar to those of the drive section. On-board the Cye is a motor controller processor. In our current implementation, this processor is responsible for generating the direct motor commands. The processor has capabilities for some higher level tasks, but these are not used.

5.2. Additional Hardware

An additional Pentium 300 MHz computer, running Linux, runs all control software and image processing. High level commands generated on-board are communicated to the Cye computer via a serial link or a wireless network connection. An NTSC video camera is mounted on the drive section of the Cye, fixed to the Cye's orientation. The camera provides wide field-of-view (~100 degrees) images that are captured at variable resolution in YUV color space (Y = luminance, U = red-Y, V = blue-Y). A video capture card, 256 MB of RAM, and a 2 GB hard drive are also present. A battery is connected for seamless change between external power/charging and battery power for autonomous operations.

5.3. Software

The algorithms to generate these high level commands and image processing are implemented in C and Java, using *TeamBots* software architecture developed at Carnegie Mellon University and Georgia Tech [2]. The image processing software, *CMVision* (also from Carnegie Mellon University), performs color analysis and color blob detection and merging [5]. *CMVision* is equipped with a graphical interface for color calibration and can detect up to 32 colors in YUV space. The cycle time of the control system, including vision processing and robot commands, is approximately 90-100 milliseconds.

For use on the hardware platform, camera calibration was conducted at two levels. First, *Flatfish*, a tool developed by Hans Moravec of Carnegie Mellon University, determined the parameters describing the aberrations of the lens. These parameters map from pixel location to points in three-space [16].

A second calibration step is used to characterize systemic errors. Targets are placed at a set of fixed distances and angles relative to the robot and the distance and angle calculated by the vision system is

recorded. Comparing measured distance versus actual distance provides a mean bias as a function of measurement distance. After correcting measurements for this bias, proportional errors are determined. Observed errors are small, typically less than 0.5 percent. Accuracy is greater in the local horizontal (left-right) dimension than in the local depth dimension.

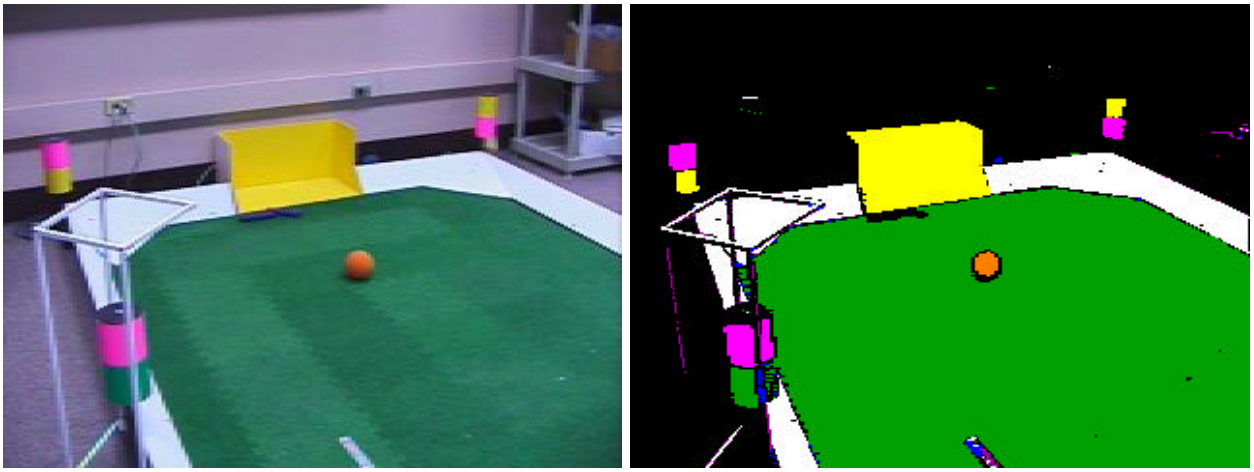

Figure 7. The *CMVision* color image processing [5]. *Left:* The raw video image of a soccer field, showing the ball and yellow goal as well as markers. *Right:* The results of color classification and blob detection on the same video frame. The yellow goal, white walls, green surface, and orange ball are clearly discernable.

A histogram of these sensing errors is provided in Figure 8. The figure indicates that the corrected distances are distributed approximately normally about the actual distance. From these errors, a standard deviation in distance can be directly determined. A similar process was completed for angle, though no bias correction was conducted. These deviations are used as parameters of the observation distributions.

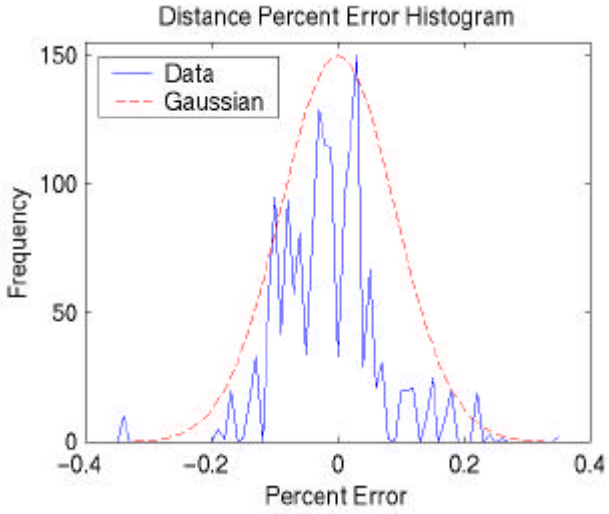

Figure 8. The histogram of distance error as a percent of distance is approximately Gaussian.

5.4. Communication

At each cycle, robots broadcast the location of any visible targets to all teammates. While robots are not synchronized, only data that has been received since the last cycle is considered. At most, this data is 120 milliseconds old. In most cases, this time difference results in negligible differences (much smaller than the standard deviations). In the rare instances when differences are large, merging is not permitted.

6. Validation on a Robot Team

6.1. Experimental Setup

To test the approach, an experiment was devised in which three robots track an object through several sequential points in the environment. In this way, the accuracy of single-robot measurements can be directly compared to the accuracy obtained by combining data from two and three robots. An illustration of this experimental setup is shown in Figure 9; each robot is 1.5 meters from the origin of the coordinate frame.

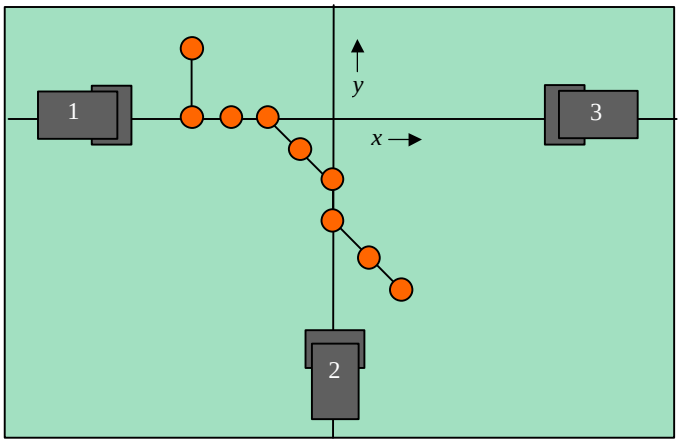

Figure 9. Experimental setup for validation. The line represents the ball trajectory with ball locations marked as circles. Dotted lines represent the global coordinate frame.

During the experiment, the ball was placed at a series of known discrete points along a trajectory. At each point, the location of the ball as measured by each robot was recorded. These individual observations were merged in pairs as well as merging all three individual observations.

6.2. Experimental Results

The experimental results are shown graphically. In Figures 10-12, the trajectories seen by individual robots are compared to the actual path of the target. It should be noted that the error increases as the target distance increases and as the target angle increases, as expected from the camera calibration.

In Figures 13-15, the estimates resulting from the merging of two robots' observations is similarly compared for each pair of robots. The estimate resulting from the merging of all three robots' measurements is shown in Figure 16.

In Figure 17, trajectory errors point-by-point and mean error are shown and compared for all single-robot, two-robot, and three-robot measurements.

In the merging process, we see improvement for several reasons. First, a more accurate data point from a closer robot will tend to counteract the larger errors on data points that are further away (or at higher angles). Additionally, the slightly better accuracy in the local horizontal direction allows robots roughly perpendicular to each other to provide additional information where it is most needed. Particularly note the significant improvement on the high errors on the top left data point for Robot 2 and Robot 3; this error is nearly eliminated when combined with Robot 1's data.

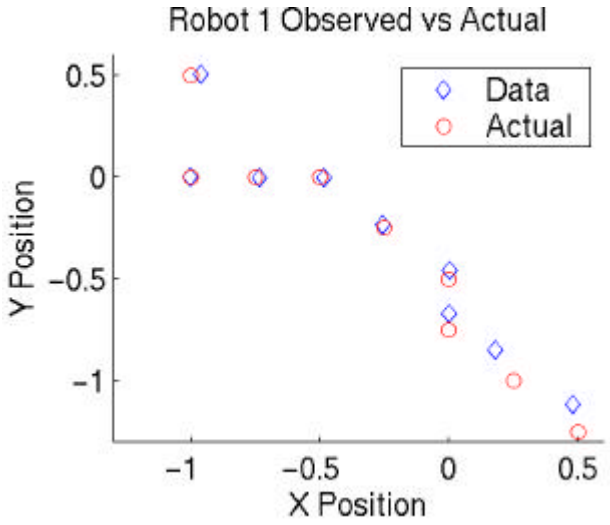

Figure 10. Data of Robot 1 versus actual ball locations.

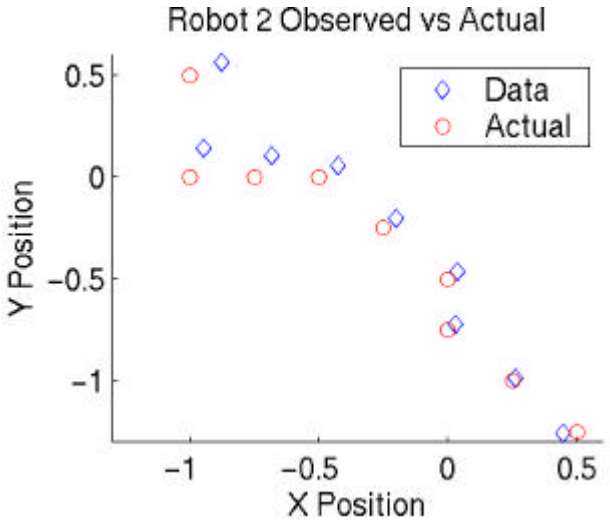

Figure 11. Data of Robot 2 versus actual ball locations.

Figure 12. Data of Robot 3 versus actual ball locations.

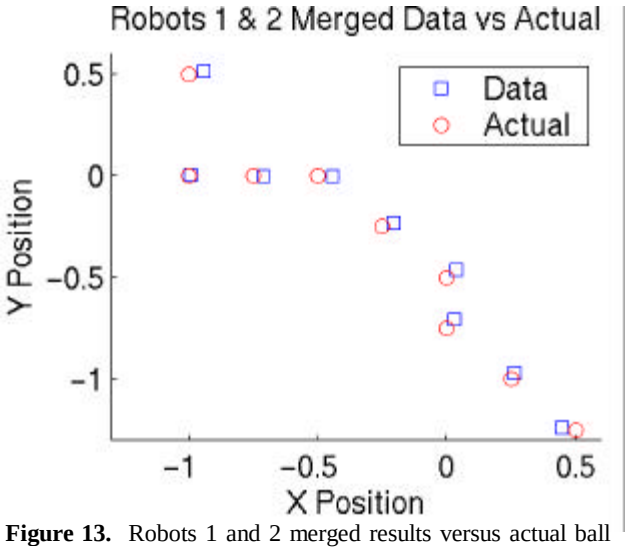

Figure 13. Robots 1 and 2 merged results versus actual ball locations.

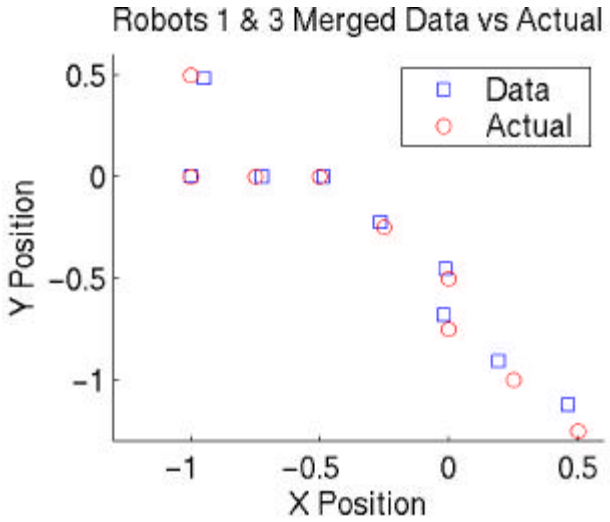

Figure 14. Robots 1 and 3 merged results versus actual ball locations.

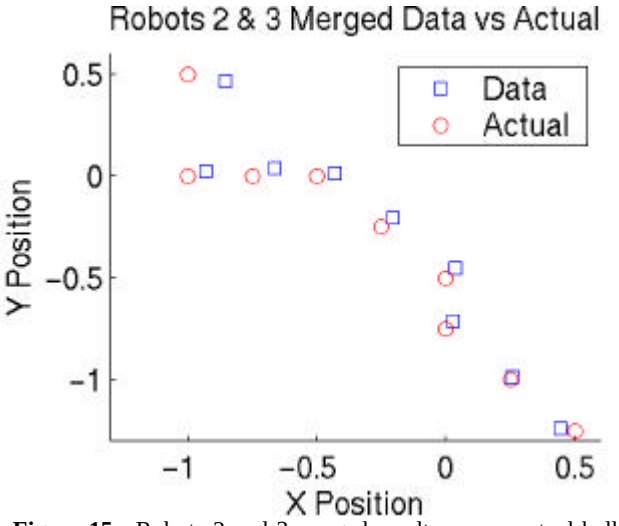

Figure 15. Robots 2 and 3 merged results versus actual ball locations.

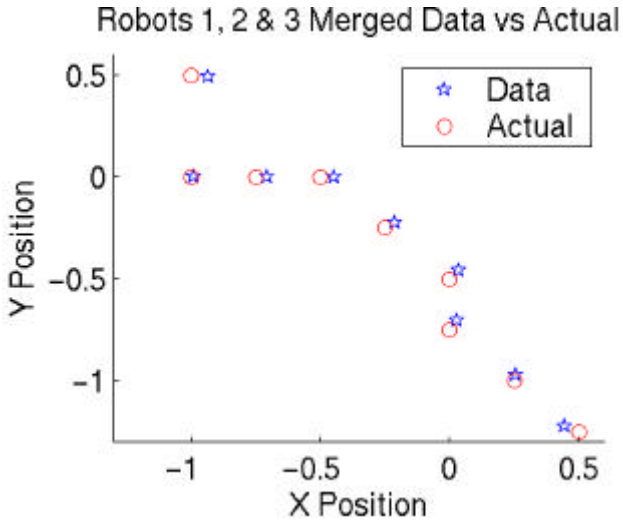

Figure 16. Results of merging observations of robots 1, 2, and 3 versus actual ball locations.

While individual trajectories are sometimes accurate at single points (in fact, occasionally more accurate that the combined information), the consistency of accuracy present in the combined results is absent in the single-robot trajectories. The high-error outliers prevalent in the single-robot measurements are far more rare in the merged data; at each level of merging the errors become more clustered near the origin, as demonstrated in Figure 17. This trend is clearly characterized by plotting the mean error of single-robot observations, two-robot observations, and three-robot observations, as shown in Figure 18.

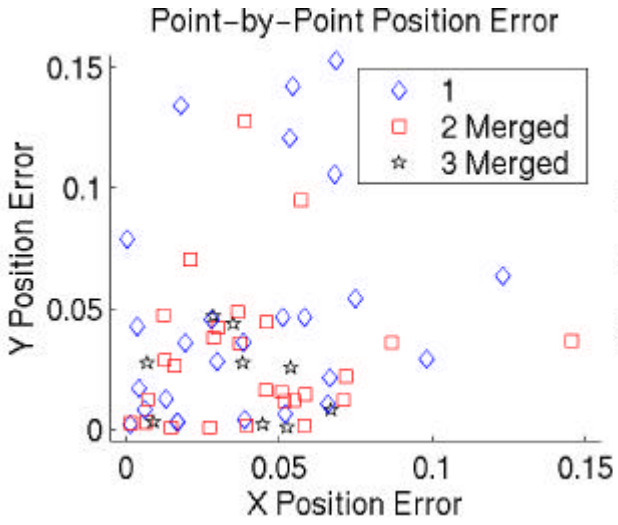

Figure 17. The x-y error at each point is shown. Each successive merging leads to tighter clustering of errors around the origin.

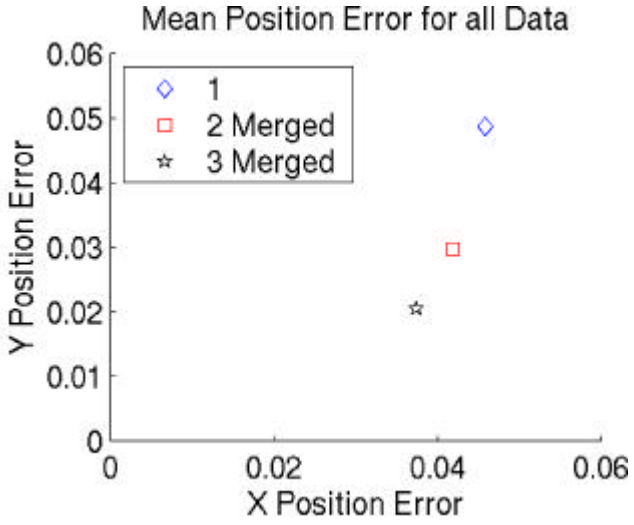

Figure 18. The average x-y error for each type of merging (none, 2-point, 3-point) are shown. The average decreases with each subsequent merging.

It should be noted that the improvement in *x* is of a smaller degree than the improvement in *y*. While this is one experiment and may not be typical, this is likely to be a result of the specific experiment configuration. In this case, only one robot is aligned along the y-axis to improve accuracy along the *x*-axis, whereas two robots have better accuracy along the *y*-axis. This imbalance may have resulted in this discrepancy.

7. Sample Applications And Results

7.1. Location and Retrieval of Unseen Targets

This sample application demonstrates the ability to increase the effective field of view of agents. In this experiment, one robot is positioned so that it can see, but not reach, the target object. The other robot cannot initially see the target object, even with camera panning, but the path to the object is unobstructed (Figure 19). This experiment was run approximately 10 times, and was considered to be a success if the target stopped within (or partly within) the goal zone.

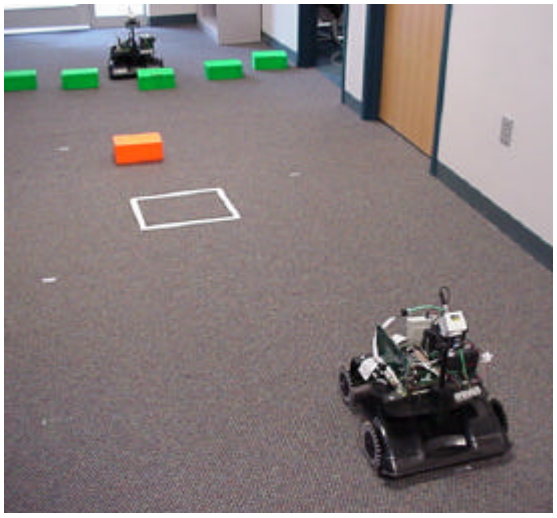

Figure 19. Unseen target location and pushing experiment setup. Robot 1 (front) cannot see target at starting location, even with camera pan, but is free to move. Robot 2 (back) can see the target, but cannot reach it. The goal is shown as a white square.

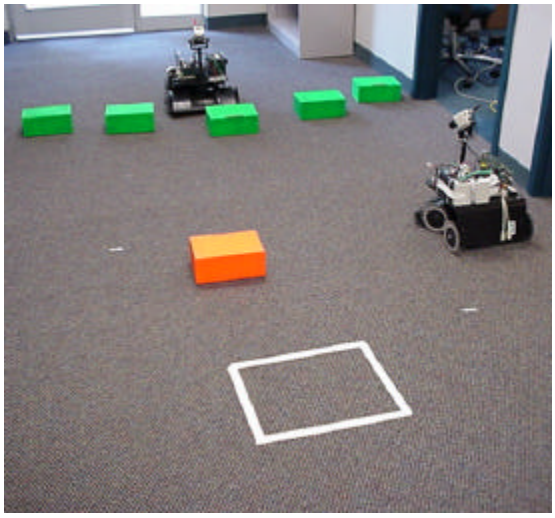

Figure 20. Robot 1 proceeds directly to target without searching, using the target location provided by Robot 2.

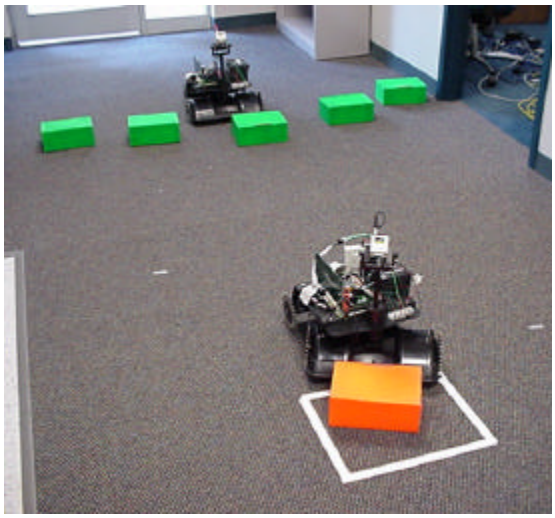

Figure 21. Robot 1 successfully pushes the target to the goal using a merged position estimate for the target.

By sharing visual information, the unobstructed robot immediately obtains a target location without requiring random search (Figure 20). The robot is able to successfully locate the object using information provided exclusively by the second robot. Once the object is located, it effectively reaches and

manipulates the target using the merged position provided by both robots (figure 21). Success was achieved in all but one experiment, in which the robot got stuck next to a wall and became disoriented. Due to the small distances traveled from precisely known starting positions, the assumptions on robot localization and coordinate frames hold. The target was never allowed to leave the ground plane.

7.2. Blind Robot Target Tracking

In this experiment, three robots were positioned around a target area. A target was moved throughout the target area, and all three robots tracked the ball using the position obtained by merging all three observations. One of the robots was subsequently blindfolded by covering the camera, for example with a box (Figure 22). The selected target area was small to minimize the individual errors of the robots' measurements. The robots are positioned so as to provide complementary viewpoints (along different axes) on the target area. For convenience, these robots were positioned at relative headings of 90 degrees.

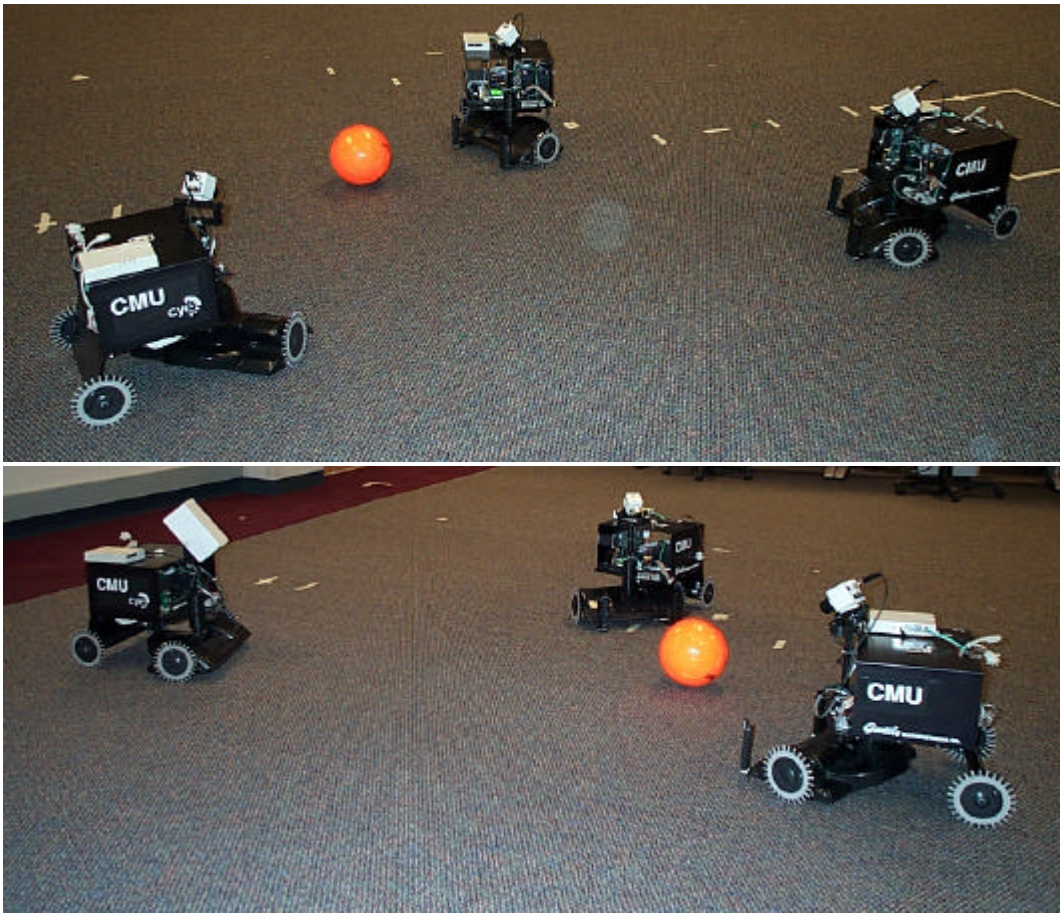

Figure 22. Blind robot ball tracking experiment setup. *Above:* In this experiment, three robots track the ball with at a location generated by combining all three observations. *Below:* In this experiment, a robot is blinded (for example, a box covers the left robot's camera) and it still can successfully track the ball with combined data from the remaining two. In both cases, all the robots directly face the ball at all times.

The ability of the blinded robot to track the ball was not substantially diminished by using the merged position from the other two. The robots were able to track the target, even at higher speeds, in most cases and always quickly recovered the object when lost. Even when the target traveled within the line of sight of a single robot (with diminished accuracy in this dimension), the additional point of view could make up for this accuracy. As robot positions are precisely known and fixed, the assumptions on robot localization and coordinate frames also hold here.

7.3. Robot Soccer

This approach to distributed sensing was applied to the CMU Hammerheads middle-sized robot soccer team for RoboCup 2000 (Figures 23 and 24). At each cycle, robots transmitted the position of the ball, if visible, so that it could be combined with all other current observations. Widely conflicting observations (for example, those with means differing by at least 2 standard deviations) were not merged. This eliminates confusion resulting from false targets (or more generally, multiple targets) and data collected in incompatible coordinate frames. This, in theory, would provide robots far from the ball with more accurate positions and allow robots that could not see the ball to quickly locate it.

Figure 23. A test run of robot soccer. Attacker (right) attempts to score on blue goal defender (left).

The soccer team robots were localized entirely by on-board odometry. This odometry drifts over time, leading to differences in coordinate frames and a violation of our assumptions. As a result, the primary impact of distributed sensing was to provide a starting point for robots to locate the ball when it had moved out of view. This use of shared information did allow robots to more quickly locate a lost ball during competition when it was not entirely obstructed. Despite the discrepancy in coordinate frames, the coordinate frames were generally coherent enough (due to slow drift and frequent resets) that when a robot looked at a potential target position, the ball became visible within the camera's wide field of view.

Figure 24. A game at RoboCup 2000, Melbourne, Australia. The CMU goalie is defending against CS Freiburg attacker. CS Freiburg went on to become the 2000 middle-size champions.

8. Conclusions

We present a method for using a simple representation of two-dimensional Gaussian distributions to fuse target position estimates from two or more robot agents. This approach is based on accepted Kalman filter theory and implements real-time sensor data fusion on a reactive multi-robot system for several different applications. The successful ability to fuse these statistical measurements and the ability to receive position estimates on targets not visible allows our robots to quickly acquire targets and to more accurately estimate object position. While this work is used vision for sensing, the approach can be applied to any sensor or suite of sensors which can be modeled by approximately Gaussian distributions.

This approach to distributed sensing and information sharing is very promising based on the applications presented here: unseen target location, accurate target acquisition and manipulation, and robot soccer. However, several extensions of this work are necessary for practically implementing this method of distributed sensing and information sharing. Even in well-localized systems, disparity between coordinate frames can arise. Such disparity must be accommodated and/or corrected. Autonomously determining the relative transformation between coordinate frames using sensors will be investigated. Additionally, well-localized robots still maintain some uncertainty in position, as computed by the localization. The accommodation of robot positional uncertainty will be incorporated into the target position distributions, as described in section 4. The limiting assumption that objects are unique must also be addressed if this method is to be applicable to general problems. Lastly, it may be possible to remove the restriction that objects of unknown size are on the ground plane; this involves a different method of transforming pixel location into world coordinates and requires further research.

References

[1] W. Anderson and R. Duffin. Journal of Mathematical Analysis Applications, 26:576, 1969.

[2] T. Balch. TeamBots. http://www.teambots.org, 2000.

[3] S. Blackman and R. Popoli. **Design and Analysis of Modern Tracking Systems.** Artech House, Boston, USA. 1999.

[4] R. G. Brown and B.R. Donald. "Mobile robot self-localization without explicit landmarks." Algorithmica, 26(3/4):515-559, 2000.

[5] J. Bruce. "CMVision Realtime Color Vision." http://parrotfish.coral.cs.cmu.edu/fastvision.

[6] J. Brusey, A. Jennings, M. Makies, C. Keen, A. Kendall, L. Padgham, and D. Singh. "RMIT Raiders." In: Veloso, Pagello, and Kitano, eds. RoboCup-99: Robot Soccer World Cup III. Springer-Verlag, pages 741-744, 2000.

[7] A. Cai, T. Fukuda, and F. Arai. "Information Sharing among Multiple Robots for Cooperation in Cellular Robotic System." In: Proceedings of IROS 97, pages 1768-1773, 1997.

[8] A. Cai, T. Fukuda, F. Arai, and H. Ishihara. "Cooperative Path Planning and Navigation Based on Distributed Sensing." In: Proceedings of the 1996 IEEE International Conference on Robotics and Automation, pages 2079-2084, 1996.

[9] P. Costa, A. Moreira, A. Sousa, P. Marques, P. Costa, and A. Matos. "5dpo-2000 Team Description." In Veloso, Pagello, and Kitano, eds. RoboCup-99: Robot Soccer World Cup III. Springer-Verlag, pages 754-757, 2000.

[10] G. Dissanayake, H. Durant-Whyte, and T. Bailey. "A Computationally Efficient Solution to the Simultaneous Localisation and Map Building (SLAM) Problem." In: Proceedings of the 2000 IEEE International Conference on Robotics and Automation, pages 1009-1014, 2000.

[11] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. "Collaborative Multi-Robot Localization." In: Autonomous Robots on Heterogeneous Multi-Robot Systems, Special Issue, 8(3), 2000.

[12] J.-S. Gutmann, W. Hatzack, I. Herrmann, B. Nebel, F. Rittinger, A. Topor, and T. Weigel. "Reliable self-localization, multirobot sensor integration, accurate path-planning and basic soccer skills: playing an effective game of robotic soccer." In: Proceedings of the Ninth International Conference on Advanced Robotics, pages 289-296, 1999.

[13] J.-S. Gutmann, W. Hatzack, I. Herrmann, B. Nebel, F. Rittinger, A. Topor, T. Weigel, and B. Welsch. "The CS Freiburg Robotic Soccer Team: Reliable Self-Localization, Multirobot Sensor Integration, and Basic Soccer Skills." In: Asada and Kitano, eds. RoboCup-98: Robot Soccer World Cup II. Springer-Verlag, pages 93-108, 1999.

[14] R. Hanek, T. Schmitt, M. Klupsch, and S. Buck. "From Multiple Images to a Consistent View." To appear in: Stone, Balch, and Kraetzschmar (eds) RoboCup-2000: Robot Soccer World Cup IV, Springer-Verlag, 2001.

[15] T. D. Larsen, K. L. Hansen, N. A. Andersen, O. and Ravn. "Design of Kalman filters for mobile robots; evaluation of the kinematic and odometric approach." In: Proceedings of the 1999 IEEE International Conference on Control Applications, Vol. 2, pages 1021-1026, 1999.

[16] H. Moravec. "Robust Navigation by Probabilistic Volumetric Sensing." http://www.ri.cmu.edu/~hpm/project.archive/robot.papers/2000/ARPA.MARS.reports.00/Report.0001.html.

[17] L. Moreno, J. M. Armingol, A. de la Escalera, and M. A. Salichs. "Global integration of ultrasonic sensors information in mobile robot localization." In: Proceedings of the Ninth International Conference on Advanced Robotics, pages 283-288, 1999.

[18] D. Nardi, G. Adorni, A. Bonarini, A. Chella, G. Clemente, E. Pagello, and M. Piaggio. "ART99 – Azzurra Robot Team." In: Veloso, Pagello, and Kitano, eds. RoboCup-99: Robot Soccer World Cup III. Springer-Verlag, pages 695-698, 2000.

[19] G. Petryk and M. Buehler. "Robust estimation of pre-contact object trajectories." In: Robot Control 1997, Vol. 2, pages 793-799, 1997.
[20] Protobics. "Cye, the first personal robot." http://www.probotics.com/
[21] D. Rembold, U. Zimmermann, T. Langle, and H. Worn. "Detection and handling of moving objects." In: Proceedings of the 24th Annual Conference of the IEEE Ind. Electron. Soc., IECON '98, Vol. 3, pages 1332-1337, 1998.
[22] J. Z. Sasiadek and P. Hartana. "Sensor data fusion using Kalman filter." In: Proceedings of the Third International Conference on Information Fusion, Vol.2, pages 19-25, 2000.
[23] R. C. Smith and P. Cheeseman. "On the Representation and Estimation of Spatial Uncertainty." In: The International Journal of Robotics Research, 5(4):56-68, 1986.
[24] C. Y. Tang, Y. P. Hung, S. W. Shih, and Z. Chen. "A feature-based tracker for multiple object tracking." In: Proceedings of the National Science Council, Republic of China, Part A, Vol. 23, No. 1, pages 151-168, 1999.
[25] H. Wang, C. S. Chua, and C. T. Sim. "Real-time object tracking from corners." Robotica, 16(1):109-116, 1999.

Appendix: Derivation of Merging Equations

We wish to demonstrate that the normalized joint probability distribution of two independent Gaussian distributions is another Gaussian distribution with mean and covariance as in equations 4 and 5.

Given two Gaussian distributions:

$$G_1 = k_1 e^{\left[-\frac{1}{2}\left(X-\bar{X}_1\right)^T C_1^{-1}\left(X-\bar{X}_1\right)\right]} \quad \text{Eq. A1}$$

$$G_2 = k_2 e^{\left[-\frac{1}{2}\left(X-\bar{X}_2\right)^T C_2^{-1}\left(X-\bar{X}_2\right)\right]} \quad \text{Eq. A2}$$

where $\bar{X}$ represents the mean of the distribution, the joint probability distribution can be computed as the product of the two individual distributions.

Let us assume that the resulting distribution is Gaussian in the form

$$G_3 = k_3 e^{\left[-\frac{1}{2}\left(X-\bar{X}_3\right)^T C_3^{-1}\left(X-\bar{X}_3\right)\right]} \quad \text{Eq. A3}$$

The resulting coefficient will be the product of the individual coefficients and a normalizing factor to make the probability sum to 1. This coefficient does not appear in the exponent with the mean and covariance; thus, this coefficient is not required to determine these properties and is hereafter ignored.

Merged Covariance Matrix: Equation 4

In multiplying exponentials, the resulting exponent is equal to the sum of the individual exponents. Computing this sum, looking only at the exponent and setting equal to the resulting exponent gives

$$-\frac{1}{2}\left(X-\bar{X}_3\right)^T C_1^{-1}\left(X-\bar{X}_3\right) = -\frac{1}{2}\left(X-\bar{X}_1\right)^T C_1^{-1}\left(X-\bar{X}_1\right) - \frac{1}{2}\left(X-\bar{X}_2\right)^T C_2^{-1}\left(X-\bar{X}_2\right) \quad \text{Eq. A4}$$

Dividing through by the coefficient, expanding each side

$$X^T C_3^{-1} X - X^T C_3^{-1}\bar{X}_3 - \bar{X}_3^T C_3^{-1} X + \bar{X}_3^T C_3^{-1}\bar{X}_3 = X^T C_1^{-1} X - X^T C_1^{-1}\bar{X}_1 - \bar{X}_1^T C_1^{-1} X + \bar{X}_1^T C_1^{-1}\bar{X}_1 + X^T C_2^{-1} X - X^T C_2^{-1}\bar{X}_2 - \bar{X}_2^T C_2^{-1} X + \bar{X}_2^T C_2^{-1}\bar{X}_2$$

Collecting terms on the right hand side leads to

$$X^T C_3^{-1} X - X^T C_3^{-1}\bar{X}_3 - \bar{X}_3^T C_3^{-1} X + \bar{X}_3^T C_3^{-1}\bar{X}_3 = X^T\left(C_1^{-1}+C_2^{-1}\right)X - X^T\left(C_1^{-1}\bar{X}_1 + C_2^{-1}\bar{X}_2\right) - \left(\bar{X}_1^T C_1^{-1} + \bar{X}_2^T C_2^{-1}\right)X + \bar{X}_1^T C_1^{-1}\bar{X}_1 + \bar{X}_2^T C_2^{-1}\bar{X}_2 \quad \text{Eq. A5}$$

Equating like terms on each side leads to four equations. The first of these is obtained by equating the terms of the form $X^T C^{-1} X$.

$$X^T C_3^{-1} X = X^T\left(C_1^{-1}+C_2^{-1}\right)X \quad \text{Eq. A6}$$

Multiplying Equation A6 by the appropriate inverses, as shown,

$$\left(X^T\right)^{-1} X^T C_3^{-1} X X^{-1} = \left(X^T\right)^{-1} X^T\left(C_1^{-1}+C_2^{-1}\right)X X^{-1}$$

leads to

$$C_3^{-1} = C_1^{-1} + C_2^{-1} \qquad \text{Eq. A7}$$

Pre-multiplying or post-multiplying each term on the right hand side by an equivalent of the identity matrix, *I*, and regrouping terms, we get

$$\begin{aligned} C_3^{-1} &= \left(C_2^{-1}C_2\right)C_1^{-1} + C_2^{-1}\left(C_1C_1^{-1}\right) \\ &= C_2^{-1}C_2C_1^{-1} + C_2^{-1}C_1C_1^{-1} \\ &= C_2^{-1}\left(C_2C_1^{-1} + C_1C_1^{-1}\right) \\ &= C_2^{-1}(C_2 + C_1)C_1^{-1} \end{aligned} \qquad \text{Eq. A8}$$

It should be noted that the order of C_1 and C_2 in Equation A6 are interchangeable (addition being commutative), and this leads to an alternate, but equivalent form:

$$\begin{aligned} C_3^{-1} &= C_2^{-1} + C_1^{-1} \\ &= C_1^{-1}C_1C_2^{-1} + C_1^{-1}C_2C_2^{-1} \\ &= C_1^{-1}(C_1 + C_2)C_2^{-1} \end{aligned} \qquad \text{Eq. A8b}$$

Now, in product form, we can invert Equation A8 by iteratively use the relationship $(AB)^{-1} = B^{-1}A^{-1}$.

$$\begin{aligned} C_3 &= \left[C_2^{-1}(C_2 + C_1)C_1^{-1}\right]^{-1} \\ &= \left(C_1^{-1}\right)^{-1}\left[C_2^{-1}(C_2 + C_1)\right]^{-1} \\ &= C_1(C_2 + C_1)^{-1}\left(C_2^{-1}\right)^{-1} \\ &= C_1(C_2 + C_1)^{-1}C_2 \end{aligned} \qquad \text{Eq. A9}$$

Further manipulation leads us to the desired result. First, C_1 is added and subtracted to the right hand side, and it is simplified using equivalence with the identity matrix.

$$\begin{aligned} C_3 &= C_1(C_2 + C_1)^{-1}C_2 \\ &= C_1(C_2 + C_1)^{-1}(C_2 + C_1 - C_1) \\ &= C_1\left[(C_2 + C_1)^{-1}(C_2 + C_1 - C_1)\right] \\ &= C_1\left[(C_2 + C_1)^{-1}(C_2 + C_1) - (C_2 + C_1)^{-1}C_1\right] \\ &= C_1\left[I - (C_2 + C_1)^{-1}C_1\right] \\ &= C_1 - C_1(C_1 + C_2)^{-1}C_1 \end{aligned} \qquad \text{Eq. A10}$$

as in Equation 4.

Performing this same manipulation on the alternate form, Equation A7b, leads to a symmetric result.

$$C_3 = C_2(C_1 + C_2)^{-1}C_1 \qquad \text{Eq. A9b}$$

$$= C_2 - C_2(C_1 + C_2)^{-1}C_2 \qquad \text{Eq. A10b}$$

Merged Mean: Equation 5

The equations obtained by equating terms of the form $X^T C^{-1} X_n$ or of the form $X_n{}^T C^{-1} X$ in Equation A5 are:

$$X^T C_3^{-1} \bar{X}_3 = X^T \left(C_1^{-1} \bar{X}_1 + C_2^{-1} \bar{X}_2 \right) \quad \text{Eq. A11}$$

$$\bar{X}_3^T C_3^{-1} X = \left(\bar{X}_1^T C_1^{-1} + \bar{X}_2^T C_2^{-1} \right) X \quad \text{Eq. A12}$$

Due the symmetry of covariance matrices, as demonstrated by Equation 1b, these two equations are symmetric and equivalent, so we will only look at the first, Equation A10.

$$\left(X^T \right)^{-1} X^T C_3^{-1} \bar{X}_3 = \left(X^T \right)^{-1} X^T \left(C_1^{-1} \bar{X}_1 + C_2^{-1} \bar{X}_2 \right)$$

$$C_3^{-1} \bar{X}_3 = C_1^{-1} \bar{X}_1 + C_2^{-1} \bar{X}_2$$

$$C_3 C_3^{-1} \bar{X}_3 = C_3 C_1^{-1} \bar{X}_1 + C_3 C_2^{-1} \bar{X}_2$$

$$\bar{X}_3 = C_3 C_1^{-1} \bar{X}_1 + C_3 C_2^{-1} \bar{X}_2 \quad \text{Eq. A13}$$

First, we begin by multiplying by the appropriate inverses to eliminate the X and C_3^{-1}.

Now we rewrite C_3 in terms of C_1 and C_2 as provided by equations A8 and A8b. We use the two different forms for convenience, to eliminate the inverses.

$$\begin{aligned} \bar{X}_3 &= C_3 C_1^{-1} \bar{X}_1 + C_3 C_2^{-1} \bar{X}_2 \\ &= C_2 (C_1 + C_2)^{-1} C_1 C_1^{-1} \bar{X}_1 + C_1 (C_1 + C_2)^{-1} C_2 C_2^{-1} \bar{X}_2 \\ &= C_2 (C_1 + C_2)^{-1} \bar{X}_1 + C_1 (C_1 + C_2)^{-1} \bar{X}_2 \end{aligned} \quad \text{Eq. A14}$$

Next, we add and subtract the same term and rearrange to obtain

$$\begin{aligned} \bar{X}_3 &= C_2 (C_1 + C_2)^{-1} \bar{X}_1 + C_1 (C_1 + C_2)^{-1} \bar{X}_2 + C_1 (C_1 + C_2)^{-1} \bar{X}_1 - C_1 (C_1 + C_2)^{-1} \bar{X}_1 \\ &= \left[C_1 (C_1 + C_2)^{-1} + C_2 (C_1 + C_2)^{-1} \right] \bar{X}_1 + C_1 (C_1 + C_2)^{-1} \bar{X}_2 - C_1 (C_1 + C_2)^{-1} \bar{X}_1 \\ &= (C_1 + C_2)(C_1 + C_2)^{-1} \bar{X}_1 + C_1 (C_1 + C_2)^{-1} \left(\bar{X}_2 - \bar{X}_1 \right) \\ &= \bar{X}_1 + C_1 (C_1 + C_2)^{-1} \left(\bar{X}_2 - \bar{X}_1 \right) \end{aligned} \quad \text{Eq. A15}$$

as in Equation 5. This verifies the merged mean equation. By plugging these results into the original product equation, the equivalence of the result can be verified, demonstrating that the result is indeed a Gaussian distribution of the expected form.

Rotation of Covariance Matrix: Equation 3

The final derivation required is the relationship in Equation 3.

From the matrix form of the two-dimensional Gaussian distribution (Equation 1), we look at the exponent: $x^T C^{-1} x$. In order to use this formulation, we must begin in the local coordinate frame (designated L). In this frame, the major and minor standard deviations align with the local x and y axes.

If we wish to transform into a new frame, aligned with the global axes, each coordinate must be transformed, in this case with a rotation. Given that the angle from the local coordinate frame to the global one is *?*, in order to rotate back to the global coordinate frame, a rotation of $-?$ must be done. In order to do this, we also make use of the relationship $(AB)^T = B^T A^T$.

The rotation of coordinates

$$x^T C^{-1} x \quad \Rightarrow \quad x^T R^T(-q) C_L{}^{-1} R(-q) x \qquad \text{Eq. A16}$$

implies that the inverse of covariance matrix, C, can be written as:

$$C^{-1} = R^T(-q) C_L{}^{-1} R(-q) = R^T C_L{}^{-1} R \qquad \text{Eq. A17}$$

We compute this inverse using the by iteratively using the relationship $(AB)^{-1} = B^{-1}A^{-1}$:

$$\begin{aligned} C &= \left(R^T C_L{}^{-1} R\right)^{-1} \\ &= \left(C_L{}^{-1} R\right)^{-1} (R^T)^{-1} \\ &= (R)^{-1} (C_L{}^{-1})^{-1} (R^T)^{-1} \\ &= R^{-1} C_L (R^T)^{-1} \end{aligned} \qquad \text{Eq. A18}$$

Next, we demonstrate that $R^{-1}=R^T$.

$$R^T = \begin{bmatrix} \cos(-q) & \sin(-q) \\ -\sin(-q) & \cos(-q) \end{bmatrix} \qquad \text{Eq. A19}$$

$$\begin{aligned} R^{-1} &= \begin{bmatrix} \cos(-q) & -\sin(-q) \\ \sin(-q) & \cos(-q) \end{bmatrix}^{-1} \\ &= \frac{1}{\cos^2(-q) + \sin^2(-q)} \begin{bmatrix} \cos(-q) & \sin(-q) \\ -\sin(-q) & \cos(-q) \end{bmatrix} \\ &= \frac{1}{1} \begin{bmatrix} \cos(-q) & \sin(-q) \\ -\sin(-q) & \cos(-q) \end{bmatrix} \\ &= \begin{bmatrix} \cos(-q) & \sin(-q) \\ -\sin(-q) & \cos(-q) \end{bmatrix} \end{aligned} \qquad \text{Eq. A19}$$

$$R^T = R^{-1} \qquad \text{Eq. A20}$$

We plug this result into Equation A18 and get

$$\begin{aligned} C &= R^{-1} C_L (R^T)^{-1} \\ &= R^T C_L (R^{-1})^{-1} \\ &= R^T C_L R \end{aligned}$$

which verifies Equation 3.