

Multi-Robot Information Sharing for Complementing Limited Perception: A Case Study of Moving Ball Interception

Richard Wang, Manuela Veloso, and Srinivasan Seshan

Abstract—

Poor sensor data because of uncertainty and hardware limitations results in a robot misinterpreting the state of its surrounding environment, leading to bad decisions and eventually failure to successfully perform its desired tasks. These limitations can be overcome if a teammate robot with a better view shares its visual information. Our work aims to investigate why current approaches fail to effectively use teammate sensor data, propose an alternative where a teammate helps to better capture the state of the environment, and demonstrate that the robot can make better decisions when a teammate shares its perceptual data. Raw teammate sensor data is not meaningful unless provided a relative, geometric transform to place this data within another robot’s own egocentric coordinates. There are few approaches that are able to discover this relative localization accurately in sparse environments while remaining computationally light. Our approach addresses these limitations by accumulating correspondence matches of objects over time from the overlapping views of two stationary robots to compute an accurate relative localization. We evaluate the benefits of teammate sensor data used with our computed relative localization with a challenging, time critical task where the robot’s cameras alone are lacking. Our empirical results with two coordinating robots indicates that our approach is able to successfully take advantage of teammate robots with a better view within the challenging physical and hardware constraints of our robots.

I. INTRODUCTION

Estimating the state of the surrounding environment from its sensor data is essential for a robot to react appropriately to perform difficult open loop tasks. Unfortunately, the robot cannot always make good decisions given that its cameras have hardware limitations resulting in limited sensor range, field of view, and resolution in addition to the reality that its camera will not always be in advantageous positions or angles to perceive all objects of interest within its environment. Another teammate robot with a better view can help to augment a robot’s own limited sensors but requires knowing the relative transform between the two robots to effectively use this valuable sensor data. It is also valuable to take into consideration the time delay of teammate observations so that observations by both robots do not conflict.

Relative localization is difficult to perform in real time using robots with standard HD cameras and limited processing power. Most techniques extract either computationally expensive features or known objects from perception sensor data. A robot can either match these against a teammate robot’s sensor data to directly compute their relative transform or against a known map to compute the robot’s global localization. Computing the relative transform between two robots by instantaneously matching features or objects from

overlapping views fails in sparse environments when presented with an insufficient number of robust correspondence points to match on. An alternative is for both robots to individually compute their global position on the known map and then share their globally transformed sensor data. Either robot can then take this global sensor data and transform it back into its own relative coordinates to use. This sensor data has now been passed through two transforms based on noisy global localization estimates by two different robots, potentially magnifying any errors such that the transformed teammate’s sensor data cannot be relied upon for accurate estimates.

Prior to using teammate sensor data with our approach, the two robots must have been stationary with overlapping views sufficiently long to capture correspondence points from a moving object to compute their relative localization. It is the hope that this approach can be extended in the future so that their relative transform can be discovered so quickly that the robots appear as if only momentarily stationary before being able to fully take advantage of a teammate’s sensor readings. In our approach, relative localization is computed by directly matching sensor data from two robots for increased accuracy within the limitations of a sparse environment and limited processing. Reduced processing requirements are achieved by extracting known objects instead of image features. A robot can capture some correspondence points from the few identifiable objects in a sparse environment but simply lacks a sufficient number to instantaneously compute a good relative localization. Our approach creates a rich environment over time by composing a unique set of correspondence points from these sparse environments until a sufficient number are accumulated to both compute their relative localization and overcome sensor noise.

We evaluate our computed relative transform by presenting a robot with the challenging, time critical task of altering the path of a ball with its feet, where its own camera sensor data is often insufficient for reacting correctly in time. The robot must capture the state of the environment accurately to determine where and when it should actuate but also early enough to overcome the robot’s limited physical walking speed. Our empirical results indicate that our computed relative localization increases the ability of the robot to perform this task, especially once the time delay of teammate sensor data is taken into consideration.

II. RELATED WORK

While many robots rely on standard HD cameras and are found in sparse environments, we choose to focus on the Nao

humanoid robots used in the RoboCup Standard Platform League (SPL) given that all teams are constrained to the same set of robot hardware and competition means that the robot’s abilities are constantly pushed to their limits. From our knowledge of the league, teammate observations cannot be relied upon for tasks where this data must be accurately transformed into another robot’s egocentric coordinates. We focus on the task of altering the path of a ball with a robot’s feet because it seems to be physically possible but sensor challenges make this difficult.

Altering the path of the ball has been successful with goalkeeper diving since 2009; however, performing this task with a robot’s feet requires more accurate sensor data. Diving covers a wide area very quickly to circumvent the robot’s slow walk but comes with the risk of damaging the robot and also incapacitates the robot until it can stand up again. Diving is successfully performed by capturing ball trajectory estimates with both interpolation from consecutive frames [10] as well as particle filters [7]. Compared to diving, a robot needs more accurate and earlier ball trajectory estimates since its feet cover much less area than its body and a walk is very slow compared to a nearly instantaneous dive. The increased accuracy requirements of ball trajectories will require much more accurate sensor data that is difficult to capture by the robot on its own.

In the RoboCup Standard Platform League (SPL), teammate sensor data is typically only relied upon for tasks where global positions are important, like which part of the field to look for the ball. For tasks requiring accurate relative positions, like approaching a ball, a robot ends up relying on its own sensor readings because teammate sensor data is unreliable in the robot’s own egocentric coordinates [1]. Current techniques are unreliable because the relative transform is computed indirectly from global localization estimates by each robot. Recent global localization work on cooperative world modeling [9] performs UKF-SLAM on both static and dynamic objects [2] and has the ability to globally localize within 5 cm. Even with such accurate global localization, the resulting relative transform by current techniques magnify any errors resulting from each robot’s individually computed global localization.

Work from the RoboCup Middle Size League (MSL) has studied the analogous problem of sensor fusion that attempts to estimate a single global estimate of an object from the individual global estimates by multiple robots on a soccer field. Robots in this league, custom designed with distinct advantages of faster movement and omni-directional cameras, impressively compute global localization with errors of only 2.5 cm [5]. Despite these advantages over the Nao’s, sensor fusion results in errors of 20 cm [3]. The increase in error over global localization is a result of attempting to combine conflicting globally transformed sensor data computed from each robot’s noisy global localization estimates. Transforming global estimates from sensor fusion into a robot’s egocentric coordinates can only increase errors because of the additional transformation from one’s own imperfect global localization estimate. Nevertheless, this is

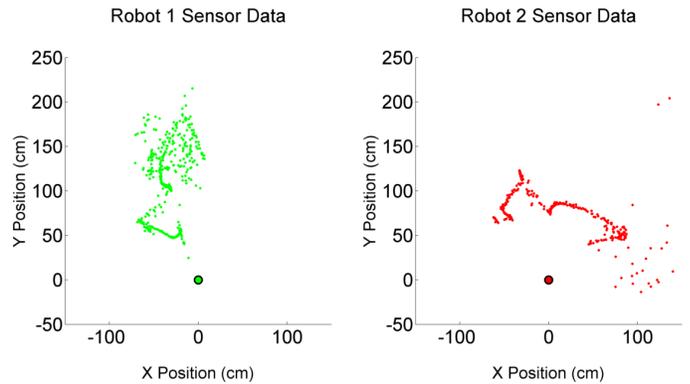


Fig. 1. Position estimates of the same unique object over time by two robots located at different positions in their own egocentric coordinates centered at the origin.

common practice because one can instantaneously use sensor data from another robot but ends up being used only as a last resort. Taking these difficulties into consideration, it appears that accurate incorporation of teammate sensor data requires direct estimates of relative localization as opposed to indirectly through global localization.

III. RELATIVE LOCALIZATION IN SPARSE ENVIRONMENTS WITH LIMITED COMPUTE

The accuracy of relative localization is directly related to the accuracy of the correspondence points used to compute it. We assume that there are insufficient identifiable objects within the overlapping views of both robots to instantaneously compute their relative localization but some objects can and will move to different positions in the environment. Our approach creates correspondence points of an object rich environment by composing these points from sparse object environments over time. Evidenced in Figure 1, two robots with the ability to identify only a single object in the environment can take advantage of position estimates over time to suggest a reasonable relative transform. The challenge of composing object poor environments is that it must result in a set of diverse correspondence points, a minimum of eight with perfect sensor data [4]. Sensor data noise also means that a relative transform estimate benefits as more diverse correspondences points are accumulated. The challenge is to carefully choose the correspondence points from object poor environments over time.

Intuitively, a stationary object no matter how long we observe it provides only a single unique correspondence point. Thus, capturing a diverse set of these points requires identifying the state of the objects across many frames over time. It would be difficult to completely model every potential object state but the algorithm has the option to accumulate correspondences only when it feels confident about correctly capturing the state of the object. An added benefit once the state of the object has been identified is the simultaneous filtering of noise from sensor data. The relative localization of the two robots can be computed once a sufficient number of correspondence points have been

accumulated and then either robot can use sensor readings from the other.

Capturing correspondences by identifying, filtering, and accumulating correspondences followed by computation of the relative localization is explained in Section IV. We then measure the added benefit of our computed relative localization for ball trajectory estimates followed by the effect of teammate sensor data in performing the challenging task of altering the trajectory of a ball’s path with its feet in Section V.

IV. DISCOVERING RELATIVE TRANSFORM OVER TIME

The key intuition is to create an object rich environment from the composition of object poor environments. This is possible because we assume that the robots remain stationary with overlapping views so that their relative transform remains constant while we accumulate correspondences from the object poor environments. Accumulating good correspondence points from both robots’ sensor data requires that we identify the state of the object, carefully pick a diverse set, and finally eliminate sensor noise.

A. Identifying Known Object States with Confidence

Our work attempts to identify objects when they are either stationary or moving with fixed velocity since these two states are common to many objects although another set of object states could easily work with our approach. These states are identified with a Switching Kalman Filter (SKF) [6], a Markov Model composed of several Kalman Filter models. We have a Kalman Filter that models the object while stationary and another while moving with fixed velocity. The SKF maintains the likelihood of each Kalman Filter model based on measurement sensor data and we interpret the most likely model as the current state of the object. An added benefit of the Kalman Filter is that its output already attempts to filter out sensor noise.

We will take advantage of the identified states by the SKF to ensure that both robots confidently agree on the state of the object, collect a diverse set of correspondence points, and provide the Kalman Filters sufficient opportunity to filter out noise as shown in Listing 1. At a minimum, sensor data from both robots should agree on the same state of the object as noise and false positives may result in very poor correspondences. Another challenge is that the SKF is unaware of every potential state of the object. Our experience finds that when the SKF lacks a sufficient model, the SKF fails to settle on a single state over a sequence of frames. As a result, we are confident of a good potential correspondence point when both robots agree on the state of the object for a consecutive threshold number of frames with the added benefit of simultaneously providing the Kalman Filter ample sensor data to filter out noise. A naive approach to accumulating unique correspondences is to require state changes in between captures of correspondence points. The result of this process is a set of hopefully very accurate

```
function SelectCorrespondences(Obs1, Obs2)
  if (Dist(Obs1) < GOODSENSORRANGE)
    ObjState1, ObjKFPosEst1 = SKF(Obs1);
  if (Dist(Obs2) < GOODSENSORRANGE)
    ObjState2, ObjKFPosEst2 = SKF(Obs2);

  if (ObjState1 == ObjState2) { AgreedStateCnt++; }
  else { AgreedStateCnt = 0; }

  if (AgreedStateCnt > CONFIDENCETHRESH)
    Correspondences.add(ObjKFPosEst1, ObjKFPosEst2);
```

Listing 1. Selecting correspondence points from Kalman Filter position estimates after both robots agree on state of object exceeding a confidence threshold.

and diverse correspondence points for computing the relative localization.

B. Computing the Relative Localization

Provided the matched correspondence points, relative localization is computed with the shape matching algorithm of Procrustes Analysis [8] because it is fast to compute and correspondences are already matched. Given an initial coarse relative transform estimate, the algorithm computes a rotation that minimizes the distance error followed by a translation that together is our best estimate of the relative transform between the two robots. The rotation is computed by translating both sets of correspondences to their origin, scaled to 1, and then rotated to find the angle that minimizes the sum of square distance between both robots. The resulting translation centers the rotated points of one robot around the other robot’s correspondence points. The two robots with their relative transform can now freely use each other’s sensor data as long as they remain stationary.

V. EVALUATING BENEFITS OF TEAMMATE SENSOR DATA

We wish to observe that teammate sensor data is the pivotal factor contributing to an increase in a robot’s success when performing a difficult task. We choose to evaluate the task of altering the path of a ball moving towards a robot using its feet. We have found that Nao humanoid robots benefit by taking advantage of teammate sensor data and additionally benefits when considering the time delay in processing the teammate’s observations. We describe the evaluation task presented to the robot, show how teammate observations are beneficial with our computed relative localization, and the increased success rate when performing a difficult task with teammate observations.

A. Developing an Evaluation Scenario

To evaluate our computation of relative localization, a ball is rolled down a ramp towards the intercepting robot as depicted in Figure 2(a) with an abstract representation in the robot’s own egocentric coordinates and in Figure 2(b) with an actual overhead view. In the robot’s egocentric coordinates, the robot itself is located at (0, 0) and the ball is originally observed on the right of the robot at (190, 0). The ramp is raised to three different heights to present the robot with

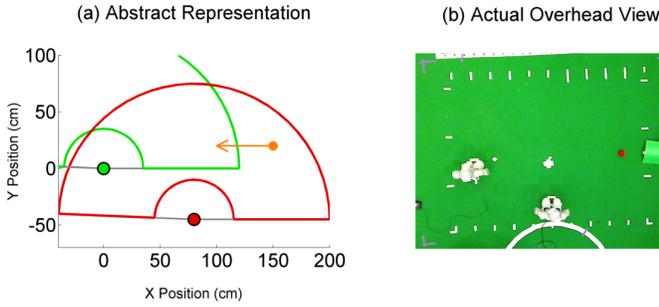


Fig. 2. Depiction of our evaluation scenario with an (a) abstract representation and (b) actual overhead view. The forward facing 180° field of view is shown between each robot’s semicircles in (a).

different ball speeds: Slow, Medium, and Fast. The robot must estimate ball trajectory, project an interception point directly in front or in back, and then walk to this point in time to influence the motion of the ball with its feet. The Nao’s have an approximately 180° forward facing horizontal field of view as depicted within the two semicircles, where objects too close or too far are difficult to capture. This is a difficult task for the robot because the ball approaches almost directly from its right, which is close to the limit of its perceivable range. The teammate robot will be placed with an approximate relative position of (80, -45) and angle of 0° with an excellent view of the entire path of the ball. We will demonstrate that taking advantage of teammate sensor data is pivotal in helping the robot to better accomplish this task.

B. Evaluating Benefits of Teammate Observations

We present the robot with real sensor data for which we know the ground truth and compare the robot’s estimates of the state of the ball with and without using its teammate’s sensor data. We expect that teammate sensor data is able to better capture the state of the object.

Figure 3 depicts ball trajectory estimates of an actual instance of our evaluation scenario relying on each robot’s individual sensor data in (a) and (b) and then combined sensor data in (c), all overlaid with the actual ground truth ball trajectory in blue. The robot on its own in Figure 3(a) reveals that its raw sensor data in black is noisy when far away and fails to perceive the ball when close due to its cameras being occluded by its own shoulder. To project when and where to alter the path of the ball, the estimated ball trajectory by the robot on its own in green is short lived and provides only a few frames for the robot to capture an accurate ball trajectory and then make a good decision. This places an enormous burden on the SKF to capture accurate trajectory estimates from little evidence; however, it can be aided by a teammate robot with a much better view.

Our algorithm computes the teammate robot to be at a relative location of (82.3654, -44.9545) and angle of 2.89° during the calibration step. The transformed teammate sensor data is shown in Figure 3(b), where it is evident that the teammate captures many more raw ball observations during the entire path of the ball. In fact, unapparent from the

```

function AlterBallPath(Obs1, Obs2)
  BestObs = RobotClosestObs(Obs1, Obs2);
  if (BestObs == Obs2)
    BestObs = BallTransform(Obs2);

  BallState, BallPos, BallVel = SKF(BestObs);

  if BallState == MOVING {
    BallIntcptTime, BallIntcptDist =
      ProjectIntercept(BallPos, BallVel);
    RobotIntcptTime = ComputeTimeToWalk(BallIntcptDist);
    DiffIntcptTime = RobotIntcptTime - BallIntcptTime;

    if LittleTimeToReact(DiffIntcptTime)
      Walk(BallIntcptDist);
  }

```

Listing 2. Intercepting robot uses observations from robot currently closest to the ball, projects the estimated trajectory of ball, then decides when and where to walk to in order to alter the path of the ball.

intercepting robot’s observations is a slight unevenness in the floor slightly influencing the ball’s trajectory more in the y-direction as reflected in the ground truth path. The teammate captures a much more complete and accurate set of trajectory estimates as shown in red. Suppose teammate sensor data had been transformed using our desired relative localization of (80, -45) and 0° angle. We can see in Figure 3(d) and (e) that the intercepting robot would then have an estimate of the state of the ball that poorly follows from the ground truth path. Clearly, teammate sensor data transformed with our computed relative localization helps the intercepting robot to better capture the state of its surrounding environment for making better informed decisions.

With a relative transform, the intercepting robot has the benefit of using both its own and its teammate’s sensor data. Listing 2 shows the decision process made by the intercepting robot. From our experience, ball position estimates are more accurate the smaller its distance is to the robot’s camera so the intercepting robot chooses to use the closest observed ball position by either robot. Ball trajectory is estimated with a SKF providing ball position and velocity estimates. It is then straightforward to project the interception point and decide how the robot should react. We do not allow the robot to use any more sensor data once it decides to actuate. As a result, the robot walks the entire distance that it projects the moment it decides to actuate so that we can indirectly observe its belief of the state of the ball at the time. Not all tasks allow robots the luxury to reverse their initial decisions so it seems important to evaluate how effectively the intercepting robot is able to make its initial decision. As a result, the robot benefits by waiting as long as possible to both capture better trajectory estimates and account for any unevenness in the floor that may change slightly the ball’s trajectory. Once there is little time remaining to react, the robot actuates and hopefully alters the path of the ball. We expect through many trials that sensor data that is able to better capture the state of the environment will be more successful in performing this task.

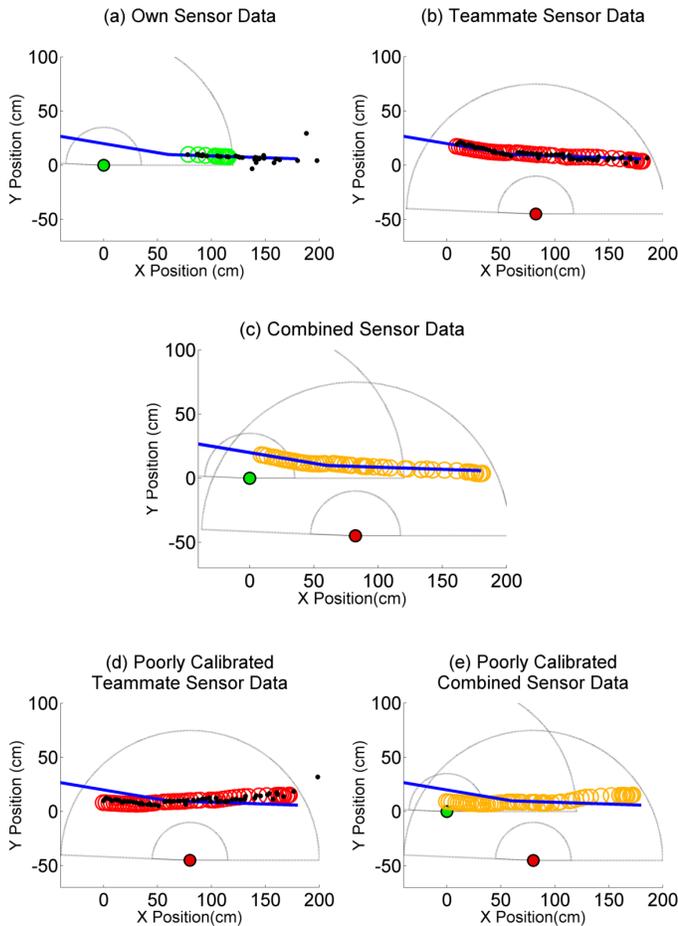


Fig. 3. Raw sensor data in black of the same rolling ball by (a) the intercepting robot and (b) the teammate robot. Ground truth path of the ball is shown in blue. Each estimated trajectory is the center of each circle depicted by (a) the intercepting robot shown in green, (b) the teammate robot shown in red, and (c) combined observations by both robots in orange. Effects of a poor relative localization in (d) and (e) provide the intercepting robot a misleading estimate of the state of the ball.

C. Using Teammate Observations to Perform a Task

To perceive any difference in the ability of the robot to make good decisions based on its own sensor data and combined sensor data, we measure the success rate in performing multiple instances of our evaluation task. The intercepting robot is presented with three different ball speeds to see if there are any differences in success rate as faster moving balls demand even more accurate trajectory estimates with less sensor data to work with. Each ball speed type using a particular set of sensor data is evaluated 25 times each for a total of 225 instances of the task performed. While we attempt to have identical execution, factors like uneven floors results in the ball crossing robot within a range of ± 20 cm requires that the robot correctly captures the state of the ball. We have a very strict criteria for an instance of the task to be considered a success. Figure 5 is a successful instance of a ball moving at medium speed with the robot successfully altering the path of the ball and demonstrating a good projected interception point. Figure 6 is considered a

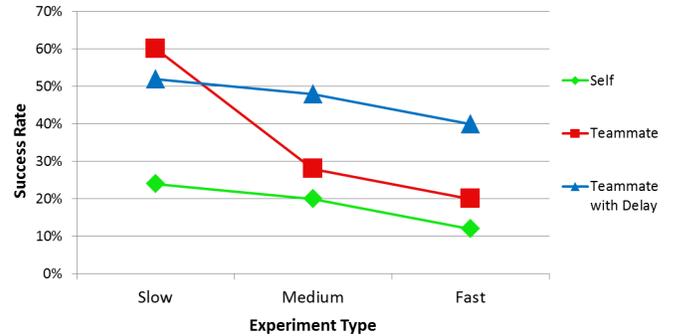


Fig. 4. Comparing the success rate in successfully performing the evaluation task using different sets of sensor data with different ball speeds.

failure despite altering the path of a ball moving at slow speed because the point at which it stops reveals a poor projected ball interception point when it made its decision.

In Figure 4, it is evident that the success rate using only the intercepting robot's sensor data in green is dominated by nearly 10% when also taking advantage of teammate sensor data in red. For the slow moving ball, teammate sensor data has an impressive success rate of nearly 60%. When the time delay of processing teammate sensor data is also considered, the success rate dominates the intercepting robot using only its own sensors by nearly 30%. These trends suggest that teammate sensor data, especially when used effectively, helps a robot to better capture the state of the environment leading to better decisions and ultimately allow the robot to more successfully perform difficult tasks.

D. Accounting for the Time Delay of Teammate Sensor Data

When the intercepting robot receives teammate sensor data, it was taken from a different time than the robot's own current observations. A number of factors contribute to this delay like network communication, processing, etc. The wireless network over a single router introduces almost negligible overheads of 3-5 ms. The stunning difference in success rates by considering delays in Figure 4 is the result of considering the total delay by the time teammate observations are actually processed by the intercepting robot. Our estimates of processing delay from our robot's control loop in addition to network communication delay of approximately 100 ms. This is a reflection of the limited compute of the Nao humanoids but does not indicate that the robot cannot work with this limitation. By recognizing the delay in processing of teammate sensor data, the intercepting robot can overcome this limitation by either delaying processing of its own observations or projecting past teammate sensor data forward in time by this delay. We choose to project teammate sensor data forward using trajectory estimates by our SKF. The success rate in Figure 4 when considering delay, especially with fast ball speeds, suggests that processing delay is incredibly important for successfully performing difficult tasks by effectively taking advantage of teammates with better views.

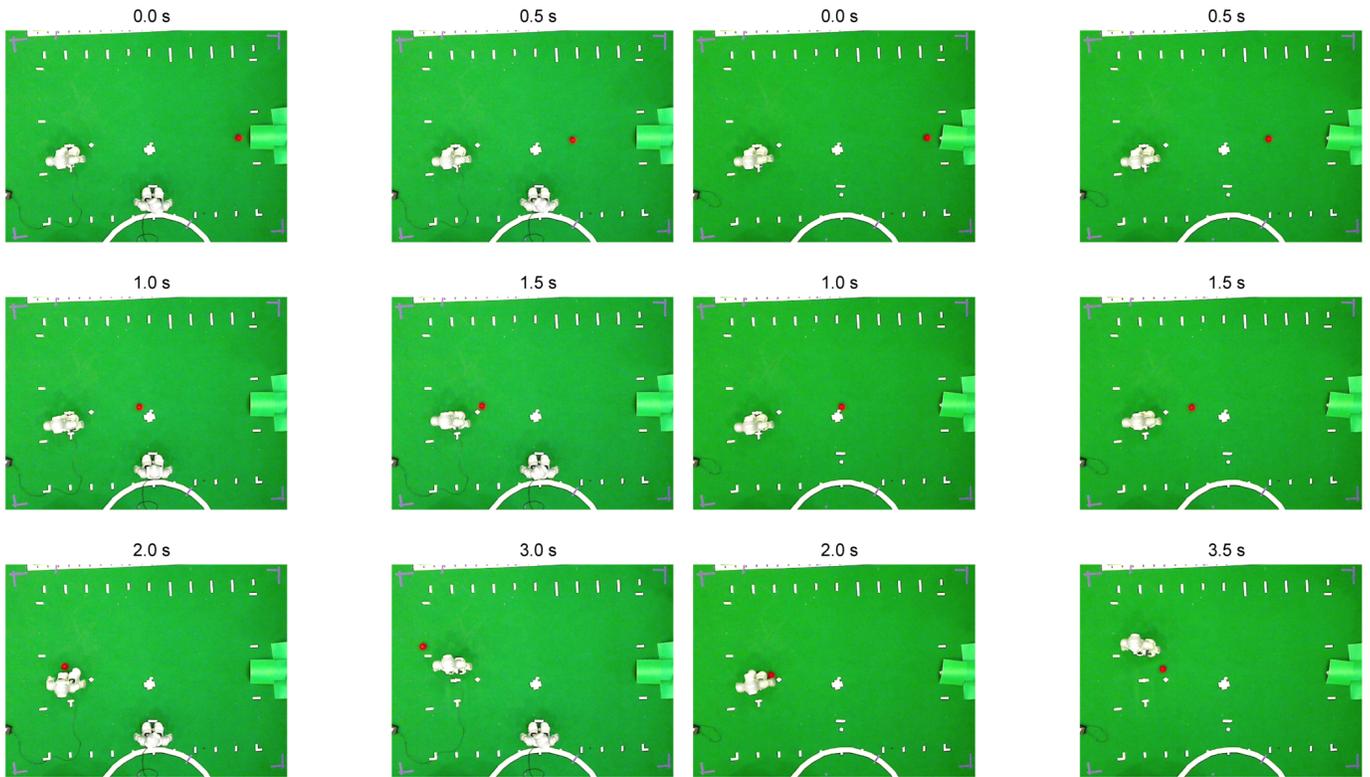


Fig. 5. Robot successfully altering path of ball at medium speed with its feet using teammate sensor data including delay. Initial decision to actuate occurs at 1.0 s. Robot's right foot kicks ball immediately before 2.0 s. Robot's body settles at 3.0 s indicating good projection of ball trajectory.

Fig. 6. Robot altering path of ball at slow speed with its feet using only its own sensor data but fails to demonstrate accurate projection of ball interception point is considered a failure in performing the evaluation task. Robot actuates just after 1.0 s and ball just happens to luckily hit robot's foot at 2.0 s. Robot settles at 3.5 s indicating poor projection of ball trajectory.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose an algorithm for computing the relative transform between overlapping views of two robots over time in sparse environments with limited compute. Relative localization is computed by identifying, matching, and capturing robust, confident correspondence points from state estimates of an object by both robots. Our results show effective use of teammates with a better view is pivotal for increasing the success rate of performing a difficult, time critical task, especially once the time delay of teammate observations is accounted for. This increased success rate is the result of the robot being able to better capture the state of its surrounding environment leading to better decisions by taking advantage of teammate sensor data.

Discovering the relative localization should be a nearly seamless process in the operation of both robots. The time to compute relative localization can be reduced in the future by better capturing correspondence points from objects in motion and working with robots in motion. It would also be beneficial for two robots without overlapping views to share their observations indirectly through the direct overlapping views of other robots. Automatic estimation of the delay in using teammate sensor data would be important to most effectively use these observations.

REFERENCES

- [1] J. Ashar, D. Claridge, B. Hall, B. Hengst, H. Nguyen, M. Pagnucco, A. Ratter, S. Robinson, C. Sammut, B. Vance, et al. Robocup standard platform league-runswift 2010. In *Australasian Conference on Robotics and Automation*, 2010.
- [2] S. Czarnetzki and C. Rohde. Handling heterogeneous information sources for multi-robot sensor fusion. In *Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 133–138. IEEE, 2010.
- [3] A. Ferrein, L. Hermanns, and G. Lakemeyer. Comparing sensor fusion techniques for ball position estimation. *RoboCup 2005: Robot soccer world cup IX*, pages 154–165, 2006.
- [4] Richard I. Hartley. Estimation of relative camera positions for uncalibrated cameras. In Giulio Sandini, editor, *ECCV*, volume 588 of *Lecture Notes in Computer Science*, pages 579–587. Springer, 1992.
- [5] M. Lauer, S. Lange, and M. Riedmiller. Calculating the perfect match: an efficient and accurate approach for robot self-localization. *Robocup 2005: Robot soccer world cup IX*, pages 142–153, 2006.
- [6] K.P. Murphy. Switching kalman filters. *Dept. of Computer Science, University of California, Berkeley, Tech. Rep.*, 1998.
- [7] Thomas Röfer, Tim Laue, Colin Graf, Tobias Kastner, Alexander Fabisch, and Christian Thedieck. B-Human Team Description for RoboCup 2010. In *RoboCup 2010: Robot Soccer World Cup XIV Preproceedings*, Singapore, 2010. RoboCup Federation.
- [8] A. Ross. Procrustes analysis. *University of South Carolina*, 2004.
- [9] S. Tasse, S. Kerner, O. Urbann, M. Hofmann, and I. Schwarz. Nao devils team report 2011.
- [10] T. Wilken, M. Missura, and S. Behnke. Designing falling motions for a humanoid soccer goalie. In *Proc. of the 4th Workshop on Humanoid Soccer Robots, Int. Conf. on Humanoid Robots*, 2009.