

Multi-Observation Sensor Resetting Localization with Ambiguous Landmarks

Brian Coltin and Manuela Veloso

the date of receipt and acceptance should be inserted later

Abstract Successful approaches to the robot localization problem include particle filters, which estimate non-parametric localization belief distributions. Particle filters are successful at tracking a robot’s pose, although they fare poorly at determining the robot’s global pose. The global localization problem has been addressed for robots that sense unambiguous visual landmarks with sensor resetting, by performing sensor-based resampling when the robot is lost. Unfortunately, for robots that make sparse, ambiguous and noisy observations, standard sensor resetting places new pose hypotheses across a wide region, in poses that may be inconsistent with previous observations. We introduce Multi-Observation Sensor Resetting (MOSR) to address the localization problem with sparse, ambiguous and noisy observations. MOSR merges observations from multiple frames to generate new hypotheses more effectively. We demonstrate experimentally on the NAO humanoid robots that Multi-Observation Sensor Resetting converges more efficiently to the robot’s true pose than standard sensor resetting, and is more robust to systematic vision errors.

1 Introduction

Whether a robot is driving through city streets, navigating the corridors of buildings, laboring on the floor of a factory, or playing a game of soccer, the ability of the robot

Brian Coltin and Manuela Veloso
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
{bcoltin, veloso}@cs.cmu.edu



Fig. 1: In the RoboCup SPL, NAO humanoid robots compete at soccer on a 4m X 6m field, with color coded goal posts and field lines which the robots use to localize.

to interact intelligently with the physical world fundamentally depends on its ability to self-localize, or determine its own pose relative to the environment. We are particularly interested in tasks where the robot must localize quickly in response to real-time constraints, and also robustly, in the presence of noisy, ambiguous, and even incorrect sensing.

Our motivation stems primarily from the RoboCup Standard Platform League (SPL), in which the NAO humanoid robots must localize in order to play soccer using visual landmarks, namely the goal posts and the markings on the field (see Figure 1). Critically, these landmarks are *ambiguous*: an observed “L”-shaped corner, for example, could correspond to any of eight such markings on the field. Additionally, color-segmentation vision algorithms will often detect false positives from either objects on the field or objects outside of the field, and localization algorithms must be robust to these errors.

Although this work is inspired by the RoboCup SPL, the problem of localizing based on ambiguous landmarks and with false positives is far from specific to this domain. Our

algorithms are robust to false positives, and apply to any domain in which the robot observes multiple, potentially ambiguous landmarks. For example, imagine a robot that navigates the halls of a building. It can detect hallway intersections, and visually observe doors. Or, imagine a robot that observes buildings on city streets. The robot sees several chain restaurants and a coffee shop. Alone, each piece of information is ambiguous, but in *combination*, the robot can determine its pose.

The localization problem has been extensively studied, and one common solution is the use of Monte Carlo Localization (MCL), where a set of particles models multiple pose hypotheses. These particles are updated based on both a model of the robot’s motion and a sensor model. The sensor model computes the likelihood of possible robot poses given sensory data. Particle filters have been widely used for robots with diverse sensory inputs, including 2D planar LIDAR scans (e.g., Dellaert et al, 1999; Adams et al, 2004), 3D point clouds (e.g., Levinson et al, 2007; Biswas and Veloso, 2012), stereo vision (e.g., Porta et al, 2005; Elinas and Little, 2005), visual information (e.g., Lenser et al, 2001; Vlassis et al, 2002; Wolf et al, 2005; Andreasson et al, 2005), range-only measurements (e.g., Kantor and Singh, 2002), and the signal strength of WiFi access points (e.g., Biswas and Veloso, 2010). Different sensor modalities offer different challenges and advantages.

One major weakness of MCL, however, is that due to practical limits on the number of particles maintained, the full pose probability distribution cannot be modeled. If no particles are in the area of the robot’s true pose, standard MCL may take a long time to converge to the robot’s true pose: hence, it requires a good initial estimate of the robot’s location. Sensor Resetting Localization (SRL) (Lenser and Veloso, 2000), an extension to MCL, addresses this *kidnapped robot* problem (Engelson and McDermott, 1992) by inserting additional hypotheses generated from sensing when the robot is uncertain of its position.

SRL is effective at both local position tracking and global position estimation. However, it still does have a few shortcomings:

1. **Exploration vs. Exploitation.** SRL favors exploration by generating observations from single camera images, spread across a large region, which increases the likelihood of localization converging to an incorrect location.
2. **Ambiguous Landmarks.** SRL does not generate hypotheses based on ambiguous observations as they could correspond to many landmarks. Thus SRL ignores potentially useful information.
3. **False Positives.** SRL is sensitive to false positives from vision, as it generates more new hypotheses from observations that contradict the current state estimate.

Multi-Observation Sensor Resetting (MOSR) localization, a new sensor resetting algorithm, addresses each of these

issues (Coltin and Veloso, 2011). MOSR localization converges quickly and accurately by using multiple observations across multiple camera frames to generate fewer but more informed new hypotheses for sensor resetting. In addition to speedy convergence times, MOSR localization is robust to false positives. In this article, we introduce a RANSAC-like approach for MOSR to robustly select samples for sensor resetting, and present extensive experiments demonstrating MOSR’s effectiveness.

In this article, we first put our work into context with an overview of related work on self-localization. Next, we present the complete algorithm for Multi-Observation Sensor Resetting Localization. Finally, we extensively demonstrate the effectiveness of the algorithm experimentally on the RoboCup Standard Platform League field with the NAO humanoid robots, which provides a challenging scenario with multiple ambiguous landmarks, detected by the NAO’s limited field of view.

2 Background and Related Work

Let $x_t \in \mathcal{R}^d$, y_t and u_t represent the robot’s d -dimensional pose, sensor observations, and control input at time t , respectively. Then let $Y_t = \{y_1, \dots, y_t\}$ and $U_t = \{u_1, \dots, u_t\}$ be the history of observations and controls. The goal of the localization problem is to determine the robot’s current pose, x_t , typically in order to perform some location-dependent task. Due to noise in sensing and motion, x_t cannot be computed with certainty. Instead, we model the pose *belief* (also called *posterior*) $bel(x_t) = p(x_t|U_t, Y_t)$, as the probability distribution over the robot’s pose given its sensing and control inputs.

However, the true posterior is typically intractable to compute. Instead, most localization approaches rely on the *Markov assumption*: that the robot’s history of observations and sensing can be ignored, and that the robot’s pose belief $bel(x_t)$ can be recursively computed with only $bel(x_{t-1})$, y_t , and u_t . The belief is then updated using the equation

$$bel(x_t) = kp(y_t|x_t) \int p(x_t|x_{t-1}, u_t)bel(x_{t-1})dx_{t-1}$$

where k is a normalizing constant. In this formulation, $p(y|x)$ is the sensor model, the probability of a set of sensory observations given the robot’s pose, and $p(x_t|x_{t-1}, u_t)$ is the motion model of the robot’s motion based on its control inputs.

The localization problem is often divided into two sub-problems: *local position tracking* and *global position estimation* (Dellaert et al, 1999). Given an initial pose estimate, a local position tracker maintains an accurate estimate of the robot’s position. However, if the robot becomes lost or does not know its initial position estimate, the local position tracker may take a long time to recover, if it can recover at

all. Algorithms designed for global position estimation, on the other hand, determine a coarse estimate of the robot's position without the need for a prior.

A variety of approaches have been proposed to solve the global localization problem. One early approach coregistered successive observations on an occupancy grid (Elfes, 1989). In a later approach, the robot's state space was discretized and a probability maintained that the robot was in each cell (Burgard et al, 1996). This approach can find the robot's global position, but only coarsely unless a highly dense discretization is used, which requires high processing time. Discretizations have also been used in combination with fuzzy logic in Fuzzy Markov localization (Buschka et al, 2000).

Other approaches are specifically tailored to the local position tracking problem. One of the earliest and most successful approaches is a non-linear version of the Kalman filter (Kalman et al, 1960), such as the Extended Kalman Filter (EKF), which robustly and reliably tracks a robot's position given an initial estimate (Leonard and Durrant-Whyte, 1991). However, Kalman filters only represent uni-modal distributions, while the actual probably distribution is often multi-modal. This is especially the case when the environment contains ambiguous landmarks, and an observation indicates only that the robot is in one of several symmetric locations. Several extensions to Kalman filters have been proposed to address this problem, including schemes that use multiple EKFs (Jensfelt and Kristensen, 2001; Quinlan and Middleton, 2010) or combine multiple EKFs with Fuzzy Markov localization (Martín et al, 2007).

2.1 Monte Carlo Localization

A more recent approach to the local position tracking problem is Monte Carlo Localization (MCL), in which a multi-modal particle filter maintains the belief of the robot's pose $bel(x_t)$, represented as a set of weighted particles, pose hypotheses p_i^t with weights w_i^t . The weights represent the likelihood that the robot is in the associated pose (Dellaert et al, 1999).

With every observation y_t and control action u_t , the particles and the weights are updated. The most common update algorithm is sampling / importance resampling (Gordon et al, 1993), although other approaches, such as the auxiliary particle filter (Pitt and Shephard, 1999; Vlassis et al, 2002), exist. Sampling / importance resampling is a three step process:

1. **Predict Step.** The particles move based on a sampling from the motion model of the robot, $p(p_i^t | p_i^{t-1}, u_t)$.
2. **Update Step.** The weight $w_i^t = w_i^{t-1} p(y_t | p_i^t)$ is updated by the sensor model, the likelihood of making the observed sensor readings given the robot's pose.

3. **Resample Step.** New particles are chosen probabilistically, where particle p_i is chosen (with replacement) with probability

$$\frac{w_i}{\sum w_i}.$$

With resampling, more particles are placed in regions of higher likelihood. The additional particles will spread out due to sampling from the motion model in the predict step, creating a more diverse particle spread in regions of higher likelihood and leading to a more precise estimate of the robot's true pose.

At each timestep, a single pose is typically selected as the robot's estimated pose, although other robot behaviors may consider multiple particles and the uncertainty in the robot's pose.¹

This formulation of MCL has a major flaw in the case of ambiguous or noisy observations, due to the nature of the resampling step. If the robot continues to acquire ambiguous or noisy observations which do not distinguish one hypothesis from another, in the long run the resampling step will cause all but one of the hypotheses to die out, leading to a reduction of diversity. To address this, the MCL resampling step is better performed with low variance resampling rather than simply drawing with replacement (Rekleitis, 2004). Other researchers have developed clustered particle filters to preserve particles for multiple likely hypotheses caused by ambiguous landmarks (Milstein et al, 2002).

Many extensions to MCL have been introduced for better local tracking. Monte Carlo Markov Chains (MCMCs) (Metropolis et al, 1953; Hastings, 1970) and the Hybrid Monte Carlo (HMC) filter (Duane et al, 1987; Choo and Fleet, 2001) both refine particles by using the gradient of the full posterior, $\frac{d}{dx} p(x_t | Y_t, U_t)$, and hence require fewer particles. However, the gradient is typically not possible to compute in practical applications, and the steps required to compute the MCMC and HMC are computationally expensive. Corrective Gradient Refinement (CGR) (Biswas et al, 2011) also refines samples locally, but with estimates of the gradient of the observation model rather than the full posterior, which can be computed efficiently.

Particle filters model multi-modal distributions in a computationally inexpensive manner. However, a limited number of particles cannot sample the entire configuration space of the robot. So MCL by itself may fail to solve the global localization problem without sufficient particles.

One way to partially mitigate this problem is to increase the number of particles with the uncertainty of the belief. If the uncertainty is high, more particles are introduced to cover a wider area, and if the robot's pose is more certain, fewer particles are used to reduce computational requirements (Fox, 2001). Another technique that may help is to

¹ See (Rekleitis, 2004) for a detailed tutorial on implementing particle filters in practice.

use a more highly peaked sensor model during local position tracking and a smooth likelihood function during global localization (Pfaff et al, 2006). Then some particles are sampled randomly from the entire space, and the position will eventually converge. However, a large number of particles is still required for effective performance.

2.2 Sensor Resetting Localization

Particle filters are effective at local position tracking but fare poorly at global localization. Typically a fixed percentage of particles is drawn at random from the environment, but this will either take significant time to converge or require an unmanageably large number of particles for large environments. However, if in addition to computing $p(y|x)$, we can compute $p(x|y)$ directly from observations, we can solve the global localization problem. Sensor Resetting Localization (SRL) extends standard particle filters by using $p(x|y)$ to place new hypotheses directly at likely poses of the robot (Lenser and Veloso, 2000). Each particle is replaced with a particle generated directly from sensing with probability

$$p_{reset} = 1 - \frac{\sum w_i}{kN},$$

where k is a constant and N is the number of particles. So if the total weight is high, the particles are already in a likely configuration and little sensor resetting is performed. If the total weight is low, the particles' poses are unlikely and they are chosen anew from $p(x|y)$ (Lenser and Veloso, 2000). Sensor resetting has been deployed for a number of domains and sensors, including to localize urban cars with the help of GPS (Levinson et al, 2007), based on features extracted from camera images (Menegatti et al, 2004), based on WiFi signal strength (Biswas and Veloso, 2010), and based on detecting visual landmarks (Lenser and Veloso, 2000; Liemhetcharat et al, 2010). The ideas from sensor resetting have also been applied to localize with Kalman filters (Jochmann et al, 2012).

Sensor resetting localization is designed to solve the kidnapped robot problem. However, by choosing p_{reset} based only on the likelihood of the current observations given the current particles, p_{reset} is extremely sensitive to noisy observations and false positives. If the particles have converged to the robot's true pose and the vision module detects a false positive, p_{reset} will become high and large numbers of particles will be replaced based on the false observation. Adaptive-MCL instead chooses p_{reset} based on smoothed estimates of the observation likelihood, and somewhat mitigates this effect by rejecting some temporary outliers (Gutmann and Fox, 2002). Other researchers have introduced heuristics for selecting p_{reset} that bias the algorithm away from exploration and towards exploitation (Marchetti et al, 2007). The MOSR algorithm that we introduce further eliminates the

effect of false positives while maintaining fast convergence times.

A second problem with sensor resetting localization is that it assumes observations are unambiguous. Ambiguous observations cannot be used effectively since SRL uses observations from only a single step in computing $p(x|y)$. Upon observing an ambiguous landmark, SRL may place new particles based on all possible matchings to landmarks. However, this removes particles that could be tracking the true pose and increases the likelihood that local position tracking will fail. This problem is addressed in part by keeping a running history of observations, and merging observations into estimates of the landmarks' positions, incorporating robot motion (Sridharan et al, 2005). Sensor resetting is then performed using triangulation with two or three of the merged landmark estimates. This approach remains sensitive to false positives, and is intended for *unambiguous* landmarks to which the relative angle and distance may be known, but not the global angle. By keeping a running history, triangulation can be used to determine a unique robot pose, even if only one landmark was observed in a given visual frame. This approach does not address ambiguous landmarks which could be at multiple locations in the environment. Other research has considered using a changing observation model based on an explicit probabilistic model of which set of landmarks is being observed (Özkucur and Akn, 2011). However, this increases the size of the state space that needs to be covered by the particle filter, increasing the necessary number of particles, and likewise remains sensitive to false positives. MOSR makes use of ambiguous landmarks by sampling from $p(x|O)$ where $O \subseteq Y$ instead of from $p(x|y)$.

2.3 Localization in the RoboCup Standard Platform League

Sensor resetting was first introduced in the context of the RoboCup Standard Platform League (SPL) (Lenser and Veloso, 2000), in which the Sony AIBOs competed on a field with six unique, unambiguous landmarks and color-coded goals. Since landmarks are detected with color-segmented vision, the robots are particularly prone to erroneous or even false measurements. Upon detecting one landmark, $p(x|y)$ places particles at random in a circle around that landmark, since the distance to the landmark is known, but the global angle is not. If two landmarks are detected in a single frame the pose is triangulated (Lenser and Veloso, 2000). As the league progressed, teams continued to incorporate more information into their sensor models, including negative information (not seeing a landmark) (Hoffman et al, 2005; Odakura et al, 2009), and lines and corners on the field (Röfer and Jungel, 2003; Schulz et al, 2011).

At the same time, the number of unique landmarks on the field has steadily decreased as localization algorithms

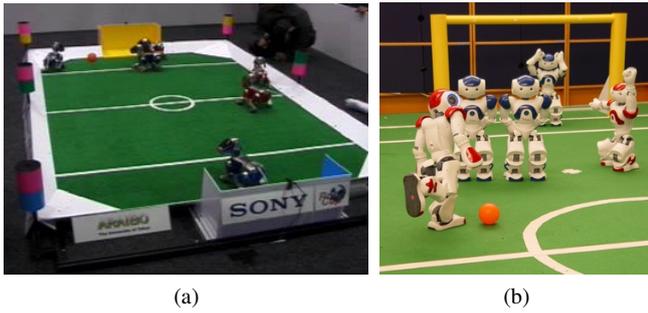


Fig. 2: (a) In 2000, the SPL played soccer with the AIBOs on a field with six unique color coded beacons on the sidelines. (b) The league has now moved to the NAO humanoid robots on a larger field without beacons.

have improved. In 2008 the RoboCup SPL switched from the AIBOs to the NAO humanoid robots (Iocchi et al, 2009). The field has no beacons on the sidelines, and only contains the field lines and corners, which are highly ambiguous, and the color coded goals (see Figure 2). When close to the goal, the robot cannot see the top goal bar, and the goal posts are also ambiguous.

Teams in the RoboCup Standard Platform League currently use variants of SRL (Burchardt et al, 2011; Hester and Stone, 2008; Kaplan et al, 2006), Kalman filters (Whe- lan et al, 2011), or a combination of the two (Ratter et al, 2010; Jochmann et al, 2012). These algorithms mainly include ambiguous observations in the sensor model $p(y|x)$, but only make limited use of ambiguous landmarks for sensor resetting (i.e., only resetting from goal posts).

2.4 Active Localization

A final challenge of localizing with visual landmarks is incorporating *active* perception into localization—the robot can decide *what* to look at. Researchers have addressed this problem for choosing a location to explore for grid-based localization methods (Fox et al, 1998) and for multiple Kalman filters (Jensfelt and Kristensen, 2001), with selecting a target for a stereo camera (Porta et al, 2005) or tiltable laser (Kümmerle et al, 2008), and even to select actions for localizing a robot with a bump sensor (Erickson et al, 2008).

In robot soccer, robots must simultaneously localize and track the ball. The robot may even have multiple hypotheses of the ball’s location to track (Rybski and Veloso, 2009), some acquired from shared information (Vail and Veloso, 2003). Heuristics based on the time since the ball or landmarks were seen and the uncertainty of localization are often used to determine whether to look at the ball or at landmarks, in order to acquire the perception necessary to actuate, to maintain a model of the world, and to localize (Win-

ner and Veloso, 2000; Roth et al, 2003; Coltin et al, 2010). When observing landmarks, RoboCup teams commonly use fixed head scanning motions or stare at each in a sequence of landmarks. Another approach is to make the observations expected to reduce the entropy the most in the underlying localization particle distribution (Seekircher et al, 2011).

3 Multi-Observation Sensor Resetting

We have mentioned that SRL has shortcomings stemming from the fact that SRL generates hypotheses from $p(x|y)$, where y is the observation from a *single* visual frame. SRL considers only the most recent observation y , but ignores every other frame in the history of observations Y .

We introduce the Multi-Observation Sensor Resetting (MOSR) algorithm, based on SRL. Rather than placing new particles by sampling from $p(x|y)$, MOSR samples from $p(x|O)$, where $O \subseteq Y$. Algorithm 1 presents the MOSR algorithm, which takes the set of particles and their weights, observations, and controls as input.

Algorithm 1 $\text{mosr}(p, w, y, u)$: The MOSR localization algorithm for a single scan $(T, T + \Delta T)$. ν , α_l and α_s are constants controlling the number of particles sensor resetting is applied to. N is the number of particles, p_i are the particle poses, w_i are the particle weights, y is the observation, and u is the control.

```

1:  $O \leftarrow \emptyset$ 
2: for  $t = T$  to  $T + \Delta T$  do
3:   for  $i = 1$  to  $N$  do
4:      $p_i \leftarrow \text{motion\_predict}(p_i, u_t)$ 
5:      $w_i \leftarrow \text{vision\_update}(p_i, w_i, y_t)$ 
6:   end for
7:    $p^{old} \leftarrow p, w^{old} \leftarrow w$ 
8:    $\bar{w} \leftarrow \sum w_i$ 
9:    $\bar{w}_l \leftarrow \bar{w}_l + \alpha_l (\bar{w} - \bar{w}_l)$ 
10:   $\bar{w}_s \leftarrow \bar{w}_s + \alpha_s (\bar{w} - \bar{w}_s)$ 
11:   $p_{reset} \leftarrow \max\{0, 1 - \nu \frac{\bar{w}_s}{\bar{w}_l}\}$ 
12:  for  $i = 1$  to  $N$  do
13:     $(p_i, w_i) \leftarrow \text{sample } p_i^{old} \text{ from } p^{old} \text{ w/ prob. } \propto w_i^{old}$ 
14:  end for
15:   $O \leftarrow \text{odometry\_update}(O, u_t)$ 
16:   $O \leftarrow O \cup \{y\}$ 
17: end for
18: for  $i = 1$  to  $N$  do
19:   if  $\text{random}() < p_{reset}$  then
20:      $p_i \leftarrow \text{mo\_hypothesis}(O)$ 
21:   else
22:      $(p_i, w_i) \leftarrow \text{sample } p_i^{old} \text{ from } p^{old} \text{ w/ prob. } \propto w_i^{old}$ 
23:   end if
24: end for

```

O is the set of observations made during a *scan*, which is an interval of time that may be delineated by a fixed interval, a set number of observations, or the robot’s behavior. Instead of performing sensor resetting after every frame,

as in SRL, MOSR performs sensor resetting only after a scan completes. By considering part of the history of observations, MOSR places fewer, more accurate hypotheses that are consistent with multiple observations. MOSR disambiguates multiple ambiguous observations, and effectively filters out false positives.

MOSR is most effective if $p(x|O)$ is highly peaked, meaning that the robot observes a set of disambiguating observations during the scan. In practice, the duration of a scan is tightly coupled with the robot’s behaviors and actions. The scan may continue while the robot moves to *actively perceive* multiple landmarks. Alternatively, if the robot is required to focus on its task, the scan may end when either a fixed number of observations is made or a set time elapses. Each scan should detect sufficient observations to disambiguate the robot’s pose. Additional observations add redundancy to reduce the effect of errors and false positives, but come at a cost in the robot’s time.

In Algorithm 1, as the robot senses the world, it applies predict (line 4), update (line 5), and resampling (lines 12-14) steps identical to standard MCL. The algorithm also updates the old observations in O with odometry information (line 15) to be relative to the robot’s current pose, and adds new observations to O (line 16). We assume that the odometry error accumulated during a scan is small enough to be ignored. The effect of odometry error is mitigated by the use of multiple observations, including more recent ones.

When a scan completes, an extra iteration of the resampling step is performed (lines 18-24). Sensor resetting is performed in this phase using *all* of the observations from the scan (lines 19-20), sampling from $p(x|O)$, not $p(x|y)$. The value of p_{reset} is computed as in Adaptive-MCL (Gutmann and Fox, 2002) (lines 7-11).

We expect the robot to add more than two observations to O to over the course of a scan, so a new algorithm is required to sample from $p(x|O)$. Algorithm 2 introduces the function `mo_hypothesis` which generates a pose hypothesis from multiple observations, using a method similar to RANSAC (Fischler and Bolles, 1981). The algorithm has four steps:

1. *Sample Observations*, line 3. Sample a subset D of observations from O that is only *finitely* ambiguous, meaning that the observations generate a finite number of possible pose hypotheses.
2. *Generate Hypotheses*, line 4. Generate a set H containing all the (finitely many) poses consistent with the observations in D .
3. *Acceptance Test*, lines 5-10. Test each hypothesis $h \in H$ against *every* observation in O to determine if they are compatible. An observation $o \in O$ is considered compatible with an observation if the sensor model $p(o|h)$ exceeds a threshold (e.g., a set percentile of the sensor noise model). Record the fraction of compatible obser-

Algorithm 2 `mo_hypothesis(O)`: Generate a pose hypothesis based on sensing of multiple observations.

```

1: for  $i = 1$  to  $K$  do
2:    $V \leftarrow \emptyset$ 
3:    $D \leftarrow$  random finitely disambiguating observations from  $O$ 
4:    $H \leftarrow$  generate_hypotheses( $D$ )
5:   for  $h \in H$  do
6:      $r_h \leftarrow$  acceptance_rate( $h, O$ )
7:     if  $r_h \geq MIN\_ACCEPTANCE\_RATE$  then
8:        $V \leftarrow V \cup h$ 
9:     end if
10:  end for
11:  if  $|V| > 0$  then
12:    return sample  $h$  from  $V$  w. prob/ prop. to  $r_h$ 
13:  end if
14: end for
15: return failure

```

ations, and throw out hypotheses for which the fraction of compatible observations falls below a threshold.² If no hypotheses are valid, return to step 1 and select a different subset D up to K times before declaring failure.

4. *Hypothesis Selection*, lines 11-13. Finally, choose a valid hypothesis with probability proportional to the fraction of observations that hypothesis agreed with. In the case that the multiple observations observed in the scan are still ambiguous, this step generates particles for all of the valid hypotheses.

MOSR uses a randomly sampled subset of the observations in O to generate hypotheses, but it uses *every* observation made during the scan to confirm each hypothesis’ validity. It is this acceptance test which empowers MOSR’s resilience to false positives.

MOSR addresses each of the issues previously discussed as limitations of standard sensor resetting:

1. **Exploration vs. Exploitation.** MOSR only samples at the end of a scan, generating fewer but more informed hypotheses. MOSR’s directed exploration enables further exploitation of strong hypotheses.
2. **Ambiguous Landmarks.** MOSR generates hypotheses from ambiguous observations.
3. **False Positives.** MOSR’s acceptance test throws out inconsistent hypotheses, making it robust to false positives.

Next, we discuss as an example how MOSR is applied to the domain of a RoboCup Standard Platform League soccer field.

² This test is sometimes called the “individual compatibility test” as it does not consider joint associations between observations, and may accept two mutually contradictory observations. A joint compatibility test could be conducted instead at additional computational cost (Neira and Tardós, 2001).

4 MOSR for RoboCup SPL Soccer

We deployed MOSR localization on the NAO robots on the RoboCup Standard Platform League (SPL) soccer field. In this section, we discuss the specifics of MOSR localization as it is applied to the landmarks of the SPL soccer field.

4.1 The SPL Setup

The RoboCup SPL plays on the Aldebaran NAO humanoid robots. The NAO senses with two cameras: one on its forehead and one in its chin, although so far only one camera may be used at a time. The field of view of the camera is very limited, as exemplified in Figure 3. The NAO can freely turn its head to look at landmarks.

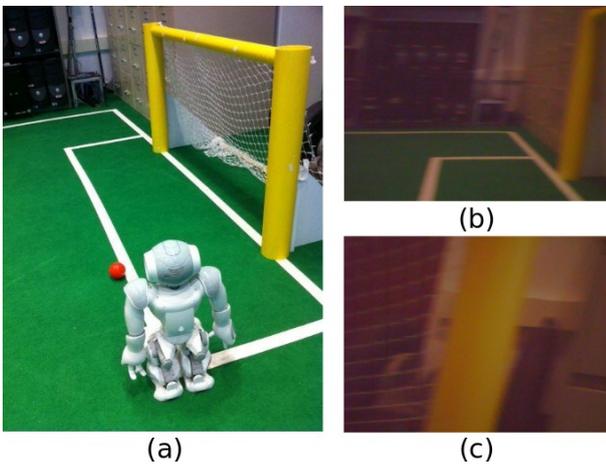


Fig. 3: (a) The NAO humanoid robot stands on the field near the yellow goal. (b, c) Two images from the robot's camera with the head at different angles. The field of view is limited, and the robot cannot see the top bar to determine whether it sees a left or a right post.

The NAOs play soccer on a playing field of fixed size (see Figure 4). The visual landmarks on the field include goal posts, corners, the center circle and lines. Our robot vision system, CMVision (Bruce et al, 2000) can distinguish between yellow and blue goal posts for each team, as well as distinguish between the left and right goal posts if the robot sees the top bar of the goal. However, if the robot does not see the top bar of the goal, the post cannot be identified as on the left or right side, and it is classified as an ambiguous unknown goal post.

The remaining landmarks observed on the field include the white lines marking the borders of the field, the center line and circle, and the goal boxes. Our vision system detects line segments, as well as the intersection of line seg-

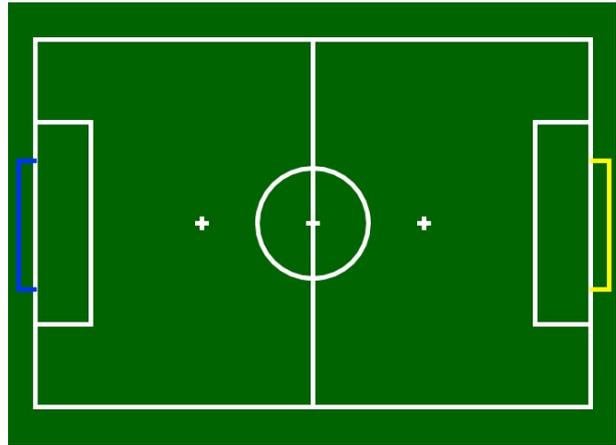


Fig. 4: The field that the SPL is played on. Robots observe two color-coded goals, and ambiguous field lines and corners.

ments at corners and the center circle. There are three types of corners:

- 8 'L' corners mark the corners of the field and goal boxes,
- 6 'T' corners mark the intersections of the field border with the goal boxes and center line, and
- 2 'X' corners denote the penalty kickoff points on both halves of the field.

Critically, the majority of the field markers are *ambiguous* and could actually correspond to multiple landmarks. A detected unknown goal post could be one of two landmarks (namely, the left or right post), a detected corner refers to between 2 and 8 landmarks, and an observed line could be paired with nearly any line segment on the field.

4.2 Monte Carlo Localization for RoboCup SPL

Our team's previous localization algorithm used Sensor Resetting Localization (Liemhetcharat et al, 2010). It also uses several other extensions to MCL, including low variance resampling. We localize with 50 particles, which we have found in practice allows the filter to localize successfully when run at 30 Hz.

In practice, robot soccer behaviors typically play soccer using a single pose estimate rather than the full probability distribution modeled by the particle filter. The localization module outputs a final pose for the use of the behaviors by first selecting the highest weighted particle within a fixed neighborhood of our previous pose estimate. This helps prevent the pose estimate from jumping across the field based on individual observations. To compute the final pose, we take the weighted mean of particles within a set radius of the selected particle. If the weight of all such particles is smaller than a fixed threshold, meaning the robot's pose is

very uncertain, we instead begin with the particle of highest global weight, allowing the robot’s pose estimate to jump (Liemhetcharat et al, 2010).

Our motion model uses the standard technique of sampling from a Gaussian for both translation and angular odometry. As a humanoid robot with biped motion, the NAO’s odometry is exceptionally poor. Furthermore, there are significant per-robot differences in how each robot moves. The robots are not capable of localizing for any significant time purely based on odometry. The use of visual observations is essential to accurately localize.

The sensor model, $p(y|x)$, weights each particle based on the likelihood of observing the landmarks y from the pose x . We compute $p(y|x)$ as a product of each individual landmark’s observation likelihood.

$$p(y|x) = \prod_{y_i \in y} p(y_i|x)$$

To compute $p(y_i|x)$ for ambiguous landmarks, we must first *match* the observation y_i to a landmark on the field. For ambiguous goal posts, we compute the observation likelihood of both goal posts, and take whichever post (left or right) is more likely.

But for lines and corners, we would need to compute many likelihoods, one for each line on the field. This is computationally expensive, especially for a common operation that must be performed every single frame on every single particle. Instead, we use a decision tree to match corner and line observations to specific corner or line landmarks, and only compute a single likelihood function.

Each individual landmark observation’s likelihood is a product of Gaussians: one for the observed distance to the robot, one for the angle to the robot, and, for lines and corners, one for the angle of the corner or line relative to the robot. Let d be the observed distance to the landmark y_i , θ the relative angle to the landmark, and ϕ , for lines and corners, the angle of the landmark relative to the robot. If μ_d , μ_θ , and μ_ϕ give the expected pose of the matching landmark, then

$$p(y_i|x) = f(d; \mu_d, \sigma_d^2(d)) f(\theta; \mu_\theta, \sigma_\theta^2(d)) f(\phi; \mu_\phi, \sigma_\phi^2(d))$$

where $f(x; \mu, \sigma^2)$ is the probability density function of a Gaussian distribution. For goal posts and the center circle, the term involving ϕ is omitted because the vision system does not detect the orientation of these landmarks.

The variances of the normal distributions are linear functions of the distance to the landmark: more distant landmarks give less accurate measurements and so the sensor model expects a higher variance. The change in variance is large for d , since the accuracy of distance measurements decrease drastically with distance as pixellation effects increase, but minor with θ and ϕ .

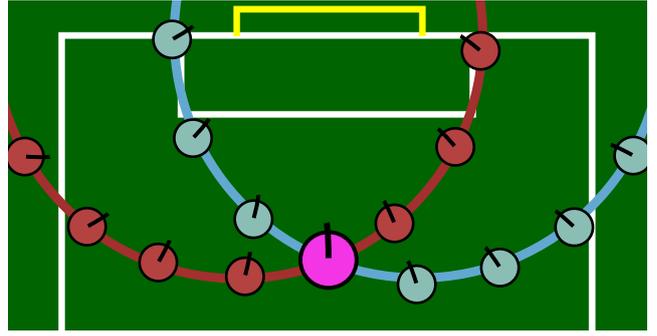


Fig. 5: The circles surrounding the two goal posts indicate the possible robot poses given observations of the left and right goal posts (or one observation of an “unknown” goal post). Possible robot poses from sensor resetting are drawn on the circles. The larger pose at the circles’ intersection represents the hypothesis generated by sensor resetting from both goal posts.

4.3 Standard Sensor Resetting in the RoboCup SPL

In standard sensor resetting localization as it is commonly applied to robot soccer, the sensor resetting only places hypotheses based on goal posts, the least ambiguous landmarks. Corners and lines are not used. If one unambiguous goal post is seen, the possible poses for the robot form a circle around that goal post, and new pose hypotheses are selected uniformly at random. For an unknown goal post where the top bar is not visible, a random post is selected to place the new hypothesis around. If two goal posts are seen in a single frame, the robot’s pose is triangulated (see Figure 5).

Noise is added to the observations before generating a new hypothesis. The noise is proportional to the expected observation noise in the sensor model, and increases with distance. The addition of noise encourages diversity of particles by placing them in slightly different poses.

Triangulating the robot’s pose from two goal posts seems straightforward, but how this is done is important. Let l_1 and l_2 be the global positions of the two goal posts, d_1 and d_2 be the observed distance to the posts, and θ_1 and θ_2 be the observed angle from the robot to the posts (see Figure 6a). We solve the following equations for the pose p of the robot:

$$l_1 = \begin{bmatrix} p_x \\ p_y \end{bmatrix} + d_1 \begin{bmatrix} \sin p_\theta + \theta_1 \\ \cos p_\theta + \theta_1 \end{bmatrix}$$

$$l_2 = \begin{bmatrix} p_x \\ p_y \end{bmatrix} + d_2 \begin{bmatrix} \sin p_\theta + \theta_2 \\ \cos p_\theta + \theta_2 \end{bmatrix}$$

Note that we have four equations and three unknowns, an overconstrained system.

The simplest way to solve this system of equations is to find the intersection of the circles around the goal posts with radii d_1 and d_2 , place the robot at the intersection of the circles, and match up the robot’s angle with one of the goal

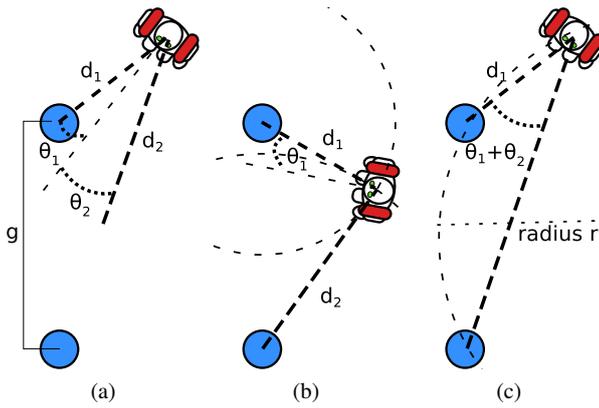


Fig. 6: (a) The robot observes the two blue goal posts at angles θ_1 and θ_2 , and distances d_1 and d_2 . Distance d_2 is inaccurate due to lighting changes. (b) Sensor resetting based on distances places the robot in the wrong pose relative to the posts, at the intersection of the two circles. (c) Sensor resetting from angular observations places the hypothesis in a position to make a valid shot on goal, on the circle of radius r .

posts (see Figure 6b). With this method, the distance to the goal posts matches the observations, but the angle to one of the goal posts will be incorrect. This is problematic, because the angle measurements are accurate, but the distance measurements are sensitive to pixellation and color calibration issues and are much less precise. In certain cases, this may lead to incorrect hypotheses which cause the robot to miss the goal when shooting.

Instead, we first solve so that the angles to the goal posts are correct. It follows from the law of sines that the robot falls on the circumcircle of radius $r = g/(2 \sin \theta)$ including the two goal post observations, where g is the length of the goal. The hypothesis is then placed at the intersection of this circumcircle and the circle around the closer goal post with the radius as the observed distance (see Figure 6c). With this method, the robot is positioned to have the correct angle to both goal posts, which means it will shoot in the direction of the goal, even if its distance is incorrect.

4.4 MOSR for the RoboCup SPL Soccer Field

Two steps of the MOSR algorithm are domain-specific: selecting disambiguating observations, and generating hypotheses from these observations. For robot soccer, we consider two types of disambiguating observations: a single corner observation, which corresponds to no more than 8 corner landmarks, or 2 observations of goal posts or the center circle. In the latter case, we ensure with the use of thresholds that we do not select two observations of the same landmark.

Lines observations are not added to O to generate new hypotheses, but are included in the sensor model.

Given an observation of a corner at distance d , angle to the robot θ , and orientation ϕ , paired with a matching field landmark (c_x, c_y, c_θ) , the generated hypothesis (p_x, p_y, p_θ) is given by the system of equations

$$\begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} c_x \\ c_y \end{bmatrix} + d \begin{bmatrix} \cos \phi \\ \sin \phi \end{bmatrix}$$

$$p_\theta = c_\theta + \phi + \pi + \theta.$$

For two disambiguating landmarks (goal posts and/or the center circle) we generate a hypothesis in the same way as standard sensor resetting, maintaining the angles to the landmarks. We again add noise to the observations before generating hypotheses to encourage diversity.

4.5 Active Vision in SPL Soccer

Our robots alternate between looking at the ball and looking at landmarks on the field to localize, depending on the state of the game and the robot's uncertainty. We have introduced three different types of scans:

1. A **horizontal** scan, where the robot moves its head from side to side to observe the goal posts.
2. A **landmark** scan, where the robot forms a list of every landmark that should be visible from its estimated current pose and looks at each in turn.
3. An **entropy**-based scan, similar to the landmark scan, but the robot only looks at the three landmarks expected to reduce the entropy of the particles the most (Seekircher et al, 2011). Looking at three landmarks is typically sufficient for MOSR to disambiguate the robot's pose. Additionally, non-targeted landmarks are often detected during the scan as well.

The landmark scan and entropy-based scan are faster and more informative since the robot looks directly at landmarks, but they assume that the robot already has some idea of its pose so it knows where the landmarks are. Thus, we initially use the horizontal scan to roughly determine the robot's pose and then switch to one of the other scans. The robot may move while scanning.

We prefer that the robot looks at the ball as much as possible so as not to lose sight of it. However, the robot should ideally be well-localized when it arrives at the ball, so it can simply kick immediately without scanning, before the opponents come and block the shot.

Our approach is to have localization report one of three states to the robot's behaviors, indicators of increasing severity: first, whether localization is confident that it is *Localized*; second, whether it is *Suspicious* of its own correctness;

and third, whether it is *Lost*. If *Suspicious*, the robot behaviors will perform a scan of the landmarks as soon as possible. If *Lost*, the robot halts its current behavior and searches for landmarks.

Figure 7 illustrates the finite state machine transitions between these three states. Since the robot’s odometry is so poor, localization becomes *Suspicious* after traveling a fixed distance: either moving 2m or turning 2π radians, whichever comes first. Upon completing a MOSR scan where the robot made at least five observations and where the final pose passes the MOSR acceptance test, the state transitions to *Localized*. The NAO becomes *Lost* whenever the variance of the particle filter’s particles exceeds a threshold.

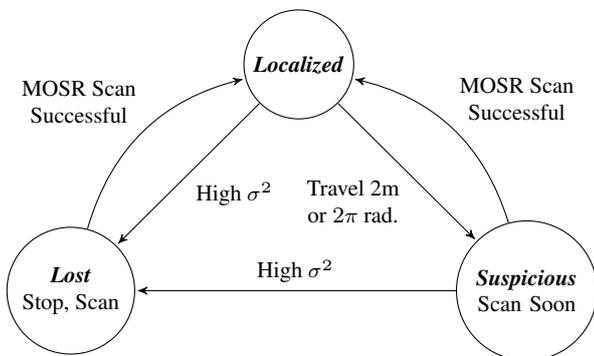


Fig. 7: The finite state machine for transitioning among three localization behavioral states, *Localized*, *Suspicious*, and *Lost*, as a function of the variance of the localization particles σ^2 , the robot’s odometry information, and the occurrence of successful MOSR scans.

5 Experimental Results

To test MOSR, we compared it directly with Sensor Resetting Localization (Lenser and Veloso, 2000). The MOSR implementation is identical to the SRL implementation in every respect except for when and how sensor resetting is performed. SRL uses Adaptive-MCL’s method of choosing the probability of sensor resetting to reduce the effect of false positives (Gutmann and Fox, 2002).

We performed two sets of experiments to validate the effectiveness of MOSR. Namely, in the first set of experiments, we measured the localization accuracy over time as the robot moved. In the second set, we studied the robot’s effectiveness at a task which is highly dependent on localization: moving to a specific position.

5.1 MOSR Localization Over Time

For the first set of experiments, a NAO robot moved on half of the soccer field. A pattern attached to the robot’s head was monitored by an overhead camera using SSL-Vision (Zickler et al, 2010). The robot’s state and the pose information from SSL-Vision were recorded in a log file for ground truth. Then, both localization algorithms were run on the log file a thousand times, and for each frame the average error of the final localization pose output by the localization module and the standard deviation of the particles from this final pose were computed.

For the first experiment, the robot was placed on the ‘X’ corner facing the yellow goal and continuously performed a horizontal scan. The particle filter was initialized with the particles spread throughout the field uniformly at random. In this experiment, the error from standard sensor resetting drops earlier when sensor resetting occurs around single posts, but after the scan completes, MOSR localization’s error drops even lower and remains there until standard sensor resetting eventually begins to catch up (see Figure 8). MOSR takes slightly longer to reach initial convergence since it waits for a scan to complete before performing sensor resetting.

Next, we chose to simulate the “blue jeans problem” in the SPL. Blue jeans worn by spectators may be consistently misidentified as goal posts if no heuristics are used to discard them. We use the same experimental setup as before, scanning in place with a horizontal scan, but place an actual blue goal post on the side of the field to introduce “false” positives into vision. Standard sensor resetting jumps particles to the other end of the field whenever it sees the blue goal post. MOSR initially transfers some weight to the other side of the field after seeing a blue goal post, but after the initial hypotheses die out, the blue goal post does not cause localization to jump (see Figure 9). MOSR does not generate new hypotheses using the goal post because it requires multiple observations to reset from, and the location of the fake goal post is inconsistent with the observations of the two yellow posts.

We also tested Multi-Observation Sensor Resetting while the robot is in motion, both while constantly performing the landmark scan and while performing the entropy-based scan, with the horizontal scan as a fallback when the robot is lost. The robot repeatedly chose a random location on one half of the field and moved to it. Figure 10 shows the results for the landmark scan, and Figure 11 shows the results for the entropy-based scan. MOSR converges to the neighborhood of the robot’s true pose faster and tends to remain closer to the true pose than standard sensor resetting. Furthermore, the particles representing the distribution of poses have a smaller variance with MOSR, since fewer hypotheses invalidated by other nearby observations are generated.

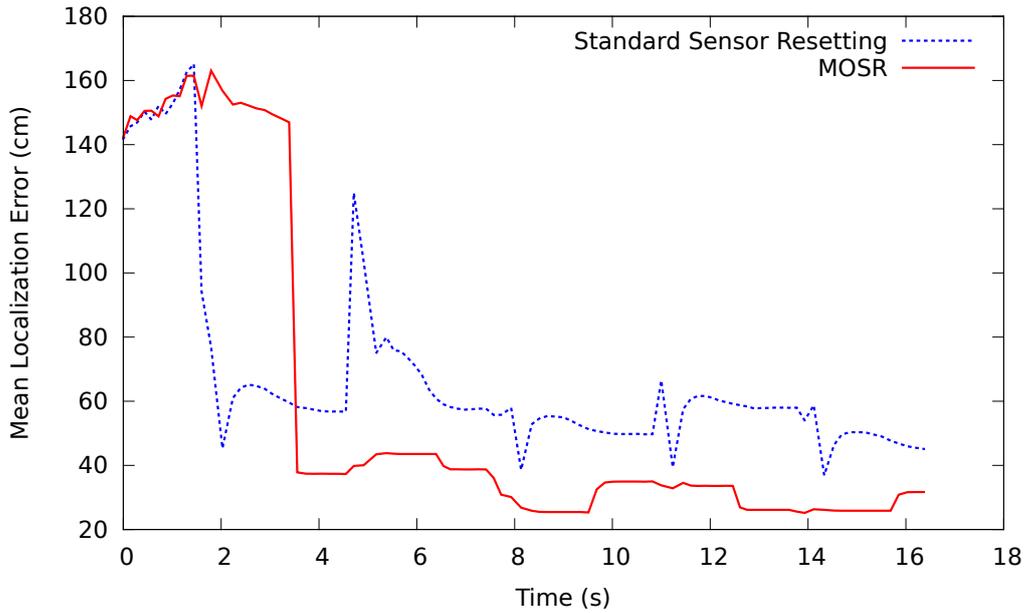


Fig. 8: Localization error from repeatedly horizontally scanning while standing still on a standard field.

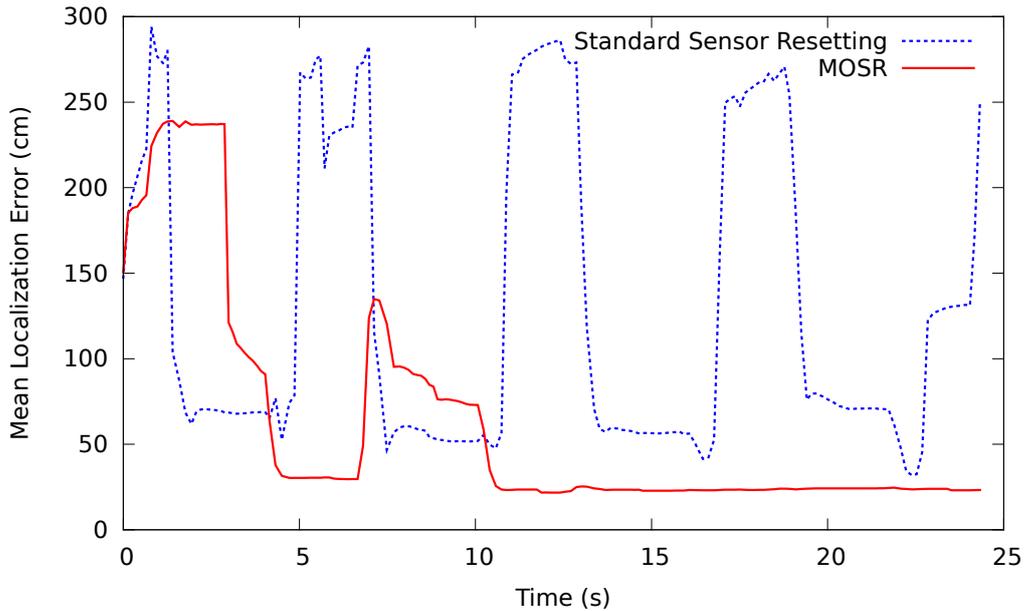


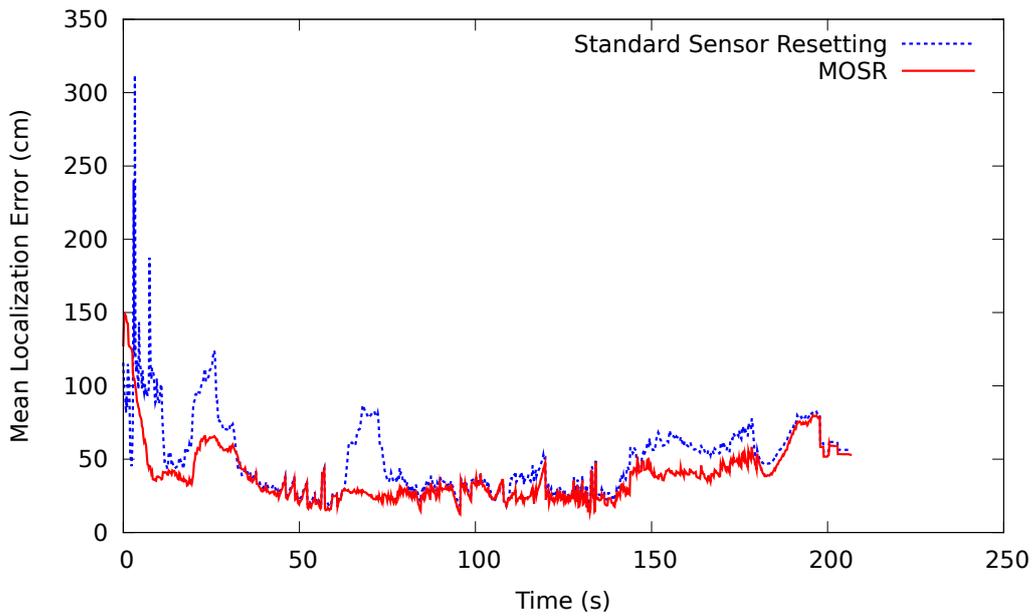
Fig. 9: Localization error with an extra, “fake” blue goal post detected at times $t = 5, 11, 17,$ and 23 , as indicated by the spikes in standard sensor resetting error.

5.2 Task-Focused MOSR Localization

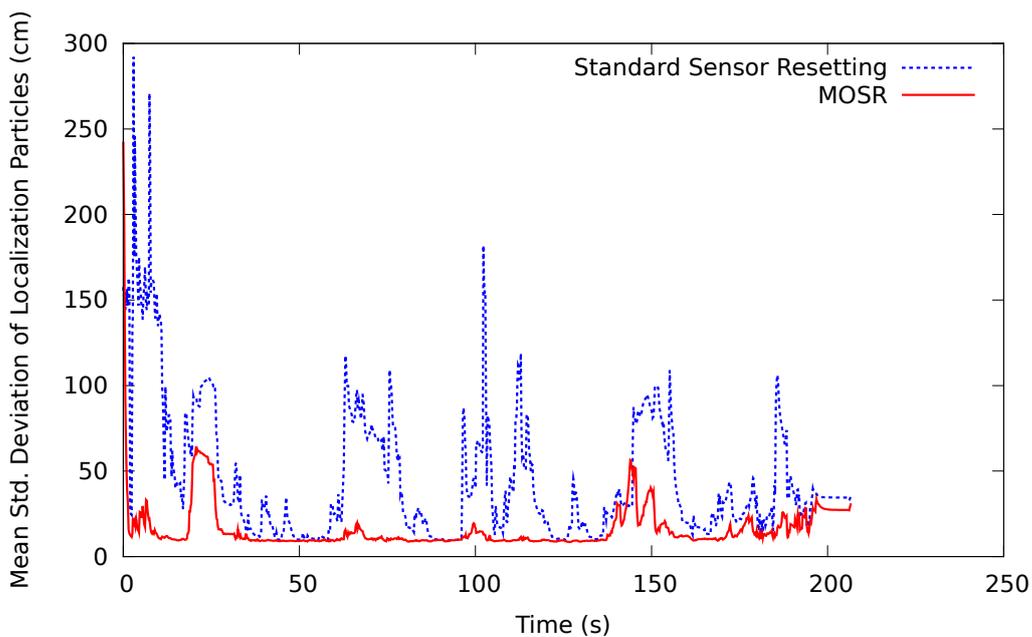
For the second set of experiments, we tested MOSR’s effectiveness in scenarios similar to what would be encountered in games of robot soccer. Rather than monitoring the accuracy of localization as the robot moves, we instead determine the robot’s effectiveness at reaching a target posi-

tion quickly and accurately with different localization algorithms.

For each experiment, we repeat ten trials in which the robot heads from a fixed starting pose to a destination pose. The particle filters begin initialized uniformly at random. Upon reaching the destination, the robot waits three seconds to make sure its position has converged, and declares that it has arrived at its destination. We measure the time the robot



(a) Landmark Scan Error



(b) Landmark Scan Standard Deviation

Fig. 10: Mean localization error and the mean standard deviation of the particle distribution for the landmark scan while the robot is in motion.

takes to reach the destination, along with the final error in angle and distance. If the robot either leaves the field (in which case it would be penalized during an actual game) or takes longer than three minutes to reach the destination, the trial is marked as a failure. Unless otherwise stated, for the experiments in this section the robot uses the entropy-based scan.

We tested the localization algorithms on two scenarios:

Scenario 1 The robot heads from the side of the field to the center of the goal box facing downfield (see Figure 12). This is the action a goalie must take in the game to return to guarding the goal after it has been penalized, and is particularly difficult when the robot is close to the goal and the objects it can see in its field of view are limited.

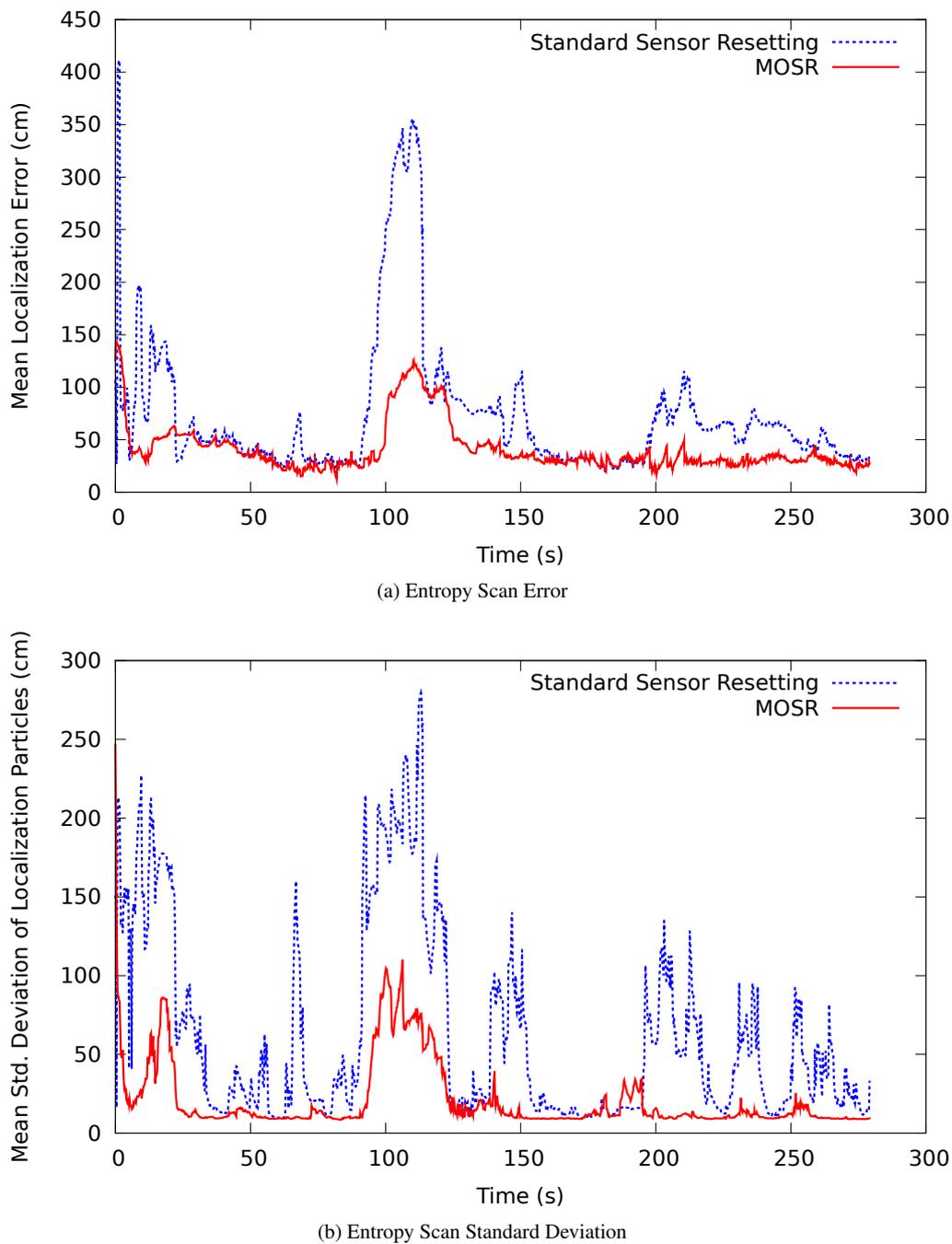


Fig. 11: Mean localization error and the mean standard deviation of the particle distribution for the entropy-based scan while the robot is in motion. At the large spike in the error for standard sensor resetting, a false positive was detected outside the field.

It cannot see the crossbar from the goalie box to determine which goal posts it detects, so all observations are ambiguous.

Scenario 2 The robot moves from a corner of the field to the edge of the center circle (see Figure 12), an action the robot must take at the start of each half of the game

to move to its initial position. This scenario is difficult at the beginning, when the robot can only see a single (ambiguous) nearby goal post. These experiments test the effectiveness of the localization algorithms for scenarios that occur in an actual game, and measure the end result

of the robot’s behavior rather than directly measuring the accuracy of localization.

Using these scenarios, we compare MOSR to SRL and MCL, compare active vision algorithms, and show MOSR’s effectiveness in response to false positives from vision. Furthermore, we demonstrate MOSR’s ability to localize in different field layouts and environments.

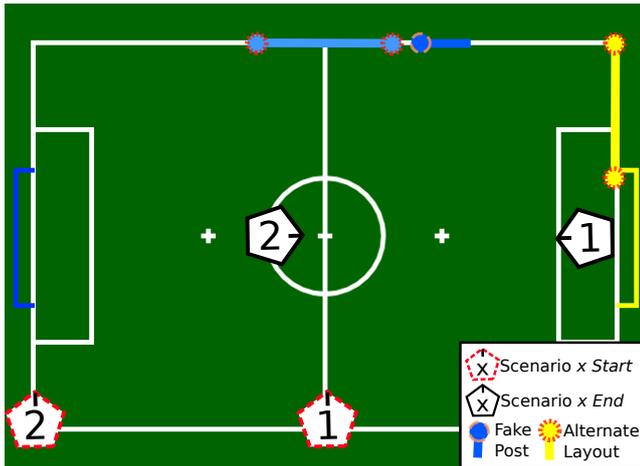


Fig. 12: The experimental setup, showing starting and ending poses for the two scenarios, the position of the fake goal post, and the alternate field layout.

5.2.1 Comparing MOSR, SRL, and MCL

For the first experiment, we compared Multi Observation Sensor Resetting to both standard sensor resetting and standard Monte Carlo Localization using the entropy-based scan. The implementations were identical, aside from how and if sensor resetting is performed. The standard sensor resetting algorithm sampled from $p(x|y)$ for every frame with observations, and standard MCL never did. Standard MCL chose 5% of the particles uniformly at random from the entire field on every frame where a landmark was detected, so that the particle filter would be able to eventually converge and solve the kidnapped robot problem.

Table 1 presents the results. For both scenarios, standard MCL performed poorly, succeeding in under half of the trials. In Scenario 1, four of the seven failures were due to leaving the field, and three were due to wandering for more than three minutes without reaching the destination. This scenario was particularly challenging because directly in front of the goal, which is the final destination, the robot cannot see the goal’s crossbar. The robot thus cannot determine whether it sees a left or right goal post, and all its observations are ambiguous. MCL may converge to an incorrect pose that agrees with the one post the robot can see,

Scenario 1				
Method	Failures	Error, cm ($\mu \pm \sigma$)	Error, $^\circ$ ($\mu \pm \sigma$)	Time, s ($\mu \pm \sigma$)
MCL	7 / 10	19.7 \pm 19.6	26.7 \pm 19.1	126 \pm 57
SRL	4 / 10	29.5 \pm 38.5	15 \pm 13.8	53 \pm 20
MOSR	0 / 10	17.4 \pm 7.6	5.1 \pm 3.7	32 \pm 3
Scenario 2				
MCL	5 / 10	34.6 \pm 41.6	15.2 \pm 6.6	65 \pm 26
SRL	0 / 10	27.4 \pm 27.4	9.8 \pm 11.3	53 \pm 25
MOSR	0 / 10	10.4 \pm 5.3	6.7 \pm 3.8	32 \pm 7

Table 1: MOSR, SRL and Standard MCL results for two scenarios. Mean errors and times include only successful trials.

and then either walk to the other side of the goal or walk inside of the goal, leaving the field.

For Scenario 2, the robot initially only sees an ambiguous blue goal post and two distant yellow posts (which are weighted low in the sensor model due to their distance). For four of the trials, the robot converged to the wrong pose initially from the ambiguous blue goal post, and proceeded to leave the field. When standard MCL did complete its task, it was largely successful at arriving in the correct pose. However, because of increased convergence time and hesitation, standard MCL took significantly longer than SRL or MOSR to arrive.

SRL succeeded every time at Scenario 2. Sampling from $p(x|y)$ allowed SRL to focus more particles in the area made feasible by the ambiguous goal post. The incorrect hypotheses were then eliminated by observing the yellow posts. For Scenario 1, however, SRL failed four times by leaving the field. As with MCL, this occurred because the robot only sighted ambiguous observations near the goal. SRL would place hypotheses that assumed the robot saw either the left goal post or the right post, and occasionally the robot would converge to the wrong pose when only one of the posts was in its visual range. The robot would then either wander back and forth in front of the goal once or twice before correcting and heading to the correct position, or leave the field before it could do so. When SRL did succeed for Scenario 1, the robot finished its task significantly faster than with MCL, in part due to a speedy initial convergence and less hesitation at the goal itself. For Scenario 2, SRL was slightly faster as well.

MOSR has little difficulty dealing with the ambiguous landmarks by the goal in Scenario 1, since it uses ambiguous goal posts and corners to place new particles only in the neighborhood of poses supported by multiple observations. For Scenario 2, MOSR quickly converged to the robot’s pose after a single scan of the ambiguous blue goal posts and the yellow goal, and proceeded to the kickoff position. MOSR succeeded in every trial, and the robot, on average, arrived at the final pose in nearly half the time with MOSR that it took with standard SRL. Furthermore, the variance of the er-

ror and arrival time were significantly reduced with MOSR, indicating that the algorithm is more consistent.

5.2.2 Comparing Active Localization Methods

In this experiment, we aimed to test the importance of active localization methods and their effect on task performance. We tested both having the robot repeatedly perform a side to side scan, and looking at all landmarks predicted to be visible, as in previous experiments. Table 2 shows the results of these tests with MOSR localization in Scenario 1, and reprints the results from the previous test which used the entropy-based scan.

Scan	Failures	Error, cm ($\mu \pm \sigma$)	Error, ° ($\mu \pm \sigma$)	Time, s ($\mu \pm \sigma$)
Horizontal	3 / 10	20.7 \pm 14.8	29.0 \pm 22.8	37 \pm 10
Landmark	0 / 10	21.9 \pm 8.3	12.4 \pm 11.9	31 \pm 3
Entropy	0 / 10	17.4 \pm 7.6	5.1 \pm 3.7	32 \pm 3

Table 2: Active Localization Methods for Scenario 1

The horizontal scan failed three of the ten trials due to leaving the field (each time, the robot ran into a goal post and fell). When the algorithm did succeed, there were three trials with angular error greater than 40°, and another trial with displacement greater than 50 cm. This indicates that actively perceiving objects, particularly the corners (which the side-to-side scan does not detect) is important for localization.

The landmark and entropy scans succeeded in reaching the destination every time. Furthermore, there was little difference in the error or arrival time for these two active vision methods. In this particular case, looking at the object expected to decrease entropy the most gives a negligible improvement over looking at every visible landmark in sequence. However, we do not expect this to hold in the general case.

5.2.3 Localizing with False Positives

For the next experiment, we examined how localization algorithms fare in the presence of false positives from vision. These are common when, for example, someone with blue jeans stands by the side of the field and is detected as a blue goal post. For this experiment, we removed the blue goal from the field and place one post by the side of the field on the other half, and covered up the second post (see Figure 12). We tested both MOSR and SRL for Scenario 1 with this setup. Table 3 shows the results.

SRL failed in seven out of ten trials: six due to leaving the field, and one due to taking more than three minutes. When the robot saw the blue goal post, sensor resetting would make the pose estimate jump to the wrong position.

Method	Failures	Error, cm ($\mu \pm \sigma$)	Error, ° ($\mu \pm \sigma$)	Time, s ($\mu \pm \sigma$)
SRL	7 / 10	17.3 \pm 4.7	5.7 \pm 3.7	115 \pm 3
MOSR	0 / 10	21.9 \pm 8.3	12.4 \pm 11.9	31 \pm 3

Table 3: Localization Methods for Scenario 1 with False Goal Posts

Upon seeing the yellow goal posts again, the robot would correct itself. However, the robot would hesitate, moving back and forth, and tended to eventually leave the field. In the three trials that succeeded, the robot happened to approach from an angle such that it did not see the blue goal post upon arriving at the final position. In these cases, the task took nearly four times as long as it did with MOSR.

MOSR succeeded every time, and the effects of the false goal post were hardly noticeable. There was no significant difference in the time it took the robot to arrive at the destination with and without the additional goal post. This is because MOSR only performs sensor resetting based on a landmark if that landmark is in agreement with the other landmarks the robot sees. So the blue goal post was used to update the weights of the particles, but is effectively filtered out when performing sensor resetting by all the observations it conflicts with.

5.2.4 Localizing with Another Field Layout

For the final experiment, to demonstrate the general applicability of MOSR aside from this specific domain, we changed the layout of the field. The blue goal was moved to the sideline at midfield, and the yellow goal was shifted to the corner of the field. The robot’s field map was updated to account for these changes. Table 4 shows the results.

Failures	Error, cm ($\mu \pm \sigma$)	Error, ° ($\mu \pm \sigma$)	Time, s ($\mu \pm \sigma$)
1 / 10	14.4 \pm 6.4	10.1 \pm 12.3	38 \pm 5

Table 4: MOSR Localization, Scenario 1, Alternate Field Layout

The robot successfully arrived at its destination in nine out of ten trials. It took slightly longer with the alternate field layout as well. This is because, if the robot happens to turn left from its initial position for whatever reason, the robot cannot see the yellow goal and has no landmark to correct itself with (the robot only sees the corners if it knows its position, otherwise if it doesn’t see the goals it repeatedly performs the side-to-side scan). For the time the robot did not succeed, it ended up facing towards the left, most likely due to poor odometry. It continued walking in that direction without detecting any landmarks, and eventually left the

field. The robot successfully completed 9 out of 10 tasks in a previously untested field layout.

5.3 MOSR's Computational Cost

To compare the computational cost of MOSR with that of SRL, we conducted twenty trial runs of Scenario 2 in simulation on an Intel 2.53 GHz i5 CPU. We recorded the time spent for the localization algorithm that runs every frame (Algorithm 1, lines 3-16) and the time spent in each sensor resetting phase (Algorithm 1, lines 18-24) separately. Table 5 presents the results.

Algorithm	Mean Time (ms)	Max Time (ms)
SRL	0.224 ± 0.172	1.372
MOSR, w/o SR	0.053 ± 0.034	0.363
MOSR, SR Only	0.758 ± 1.156	6.45
MOSR, Both Phases	0.070 ± 0.211	6.517

Table 5: Per Frame Computation Times for SRL and MOSR Phases

Both algorithms are fast enough to run in real-time on the robot (and likely have room for further optimization). An average frame of MOSR runs in approximately a third the time of SRL, since MOSR does not need to perform sensor resetting every frame. However, when MOSR does perform its sensor resetting phase, there is a large spike in computing time. The robot is still able to localize at full framerates without delay, and has computation time remaining for other perception and planning tasks.

6 Conclusion

We have introduced MOSR localization, which generates new localization hypotheses from multiple visual observations collected during a scan. MOSR localization converges quickly and accurately by generating fewer but more informed new hypotheses for sensor resetting from multiple observations. By generating hypotheses from multiple observations, MOSR is able to make use of ambiguous observations, and is robust to false positives. We demonstrated MOSR's effectiveness experimentally in the robot soccer domain.

MOSR is applicable to any system where a robot needs to localize based on ambiguous landmarks. To extend MOSR to additional domains, MOSR requires a domain-specific scanning behavior to seek out landmark observations, a domain-specific function to select sets of finitely ambiguous observations, and a domain-specific function to generate hypotheses from finitely ambiguous observation sets. Potential future work on MOSR includes further optimization of MOSR's

parameters (particularly the number of particles selected for sensor resetting), and further study to guide the selection of finitely disambiguating sets of observations in other domains.

Acknowledgments

The authors thank the other members of the CMurfs robot soccer team: Somchaya Liemhetcharat, Junyun Tay and Cetin Meriçli, for their contributions in developing the complete RoboCup system used for testing. Special thanks also go to Francisco Martin and Joydeep Biswas for their help in setting up SSL-Vision.

This research was partially sponsored by the Office of Naval Research under grant number N00014-09-1-1031. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

References

- Adams M, Zhang S, Xie L (2004) Particle filter based outdoor robot localization using natural features extracted from laser scanners. In: Proceedings of the International Conference on Robotics and Automation (ICRA), IEEE, vol 2, pp 1493–1498
- Andreasson H, Treptow A, Duckett T (2005) Localization for mobile robots using panoramic vision, local features and particle filter. In: Proceedings of the International Conference on Robotics and Automation (ICRA), IEEE, pp 3348–3353
- Biswas J, Veloso M (2010) Wifi localization and navigation for autonomous indoor mobile robots. In: Proceedings of the International Conference on Robotics and Automation (ICRA), IEEE, pp 4379–4384
- Biswas J, Veloso M (2012) Depth camera based indoor mobile robot localization and navigation. In: Proceedings of the International Conference on Robotics and Automation (ICRA), IEEE, pp 1697–1702
- Biswas J, Coltin B, Veloso M (2011) Corrective gradient refinement for mobile robot localization. In: Proceedings of the International Conference on Intelligent Robots and Systems (IROS), IEEE, pp 73–78
- Bruce J, Balch T, Veloso M (2000) Fast and inexpensive color image segmentation for interactive robots. In: Proceedings of the International Conference on Intelligent Robots and Systems (IROS), IEEE, vol 3, pp 2061–2066
- Burchardt A, Laue T, Röfer T (2011) Optimizing particle filter parameters for self-localization. In: RoboCup 2010: Robot Soccer World Cup XIV, Springer, pp 145–156

- Burgard W, Fox D, Hennig D, Schmidt T (1996) Estimating the absolute position of a mobile robot using position probability grids. In: Proceedings of the National Conference on Artificial Intelligence (AAAI), pp 896–901
- Buschka P, Saffiotti A, Wasik Z (2000) Fuzzy landmark-based localization for a legged robot. In: Proceedings of the International Conference on Intelligent Robots and Systems (IROS), IEEE, vol 2, pp 1205–1210
- Choo K, Fleet D (2001) People tracking using hybrid monte carlo filtering. In: Proceedings of the International Conference on Computer Vision (ICCV), IEEE, vol 2, pp 321–328
- Coltin B, Veloso M (2011) Multi-observation sensor resetting localization with ambiguous landmarks. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), pp 1462–1467
- Coltin B, Liemhetcharat S, Meriçli C, Tay J, Veloso M (2010) Multi-humanoid world modeling in standard platform robot soccer. In: Proceedings of the International Conference on Humanoid Robots (Humanoids), IEEE, pp 424–429
- Dellaert F, Fox D, Burgard W, Thrun S (1999) Monte Carlo localization for mobile robots. In: Proceedings of the International Conference on Robotics and Automation (ICRA), IEEE, vol 2, pp 1322–1328
- Duane S, Kennedy A, Pendleton B, Roweth D (1987) Hybrid Monte Carlo. *Physics letters B* 195(2):216–222
- Elfes A (1989) Using occupancy grids for mobile robot perception and navigation. *Computer* 22(6):46–57
- Elinas P, Little J (2005) σ MCL: Monte-Carlo localization for mobile robots with stereo vision. In: Proceedings of the Robotics: Science and Systems Conference (RSS), pp 373–380
- Engelson S, McDermott D (1992) Error correction in mobile robot map learning. In: Proceedings of the International Conference on Robotics and Automation (ICRA), IEEE, pp 2555–2560
- Erickson L, Knuth J, O’Kane J, LaValle S (2008) Probabilistic localization with a blind robot. In: Proceedings of the International Conference on Robotics and Automation (ICRA), IEEE, pp 1821–1827
- Fischler M, Bolles R (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 24(6):381–395
- Fox D (2001) KLD-sampling: Adaptive particle filters and mobile robot localization. *Advances in Neural Information Processing Systems (NIPS)* pp 26–32
- Fox D, Burgard W, Thrun S (1998) Active Markov localization for mobile robots. *Robotics and Autonomous Systems* 25(3):195–207
- Gordon N, Salmond D, Smith A (1993) Novel approach to nonlinear/non-gaussian bayesian state estimation. In: Proceedings From Radar and Signal Processing, IET, vol 140, pp 107–113
- Gutmann JS, Fox D (2002) An experimental comparison of localization methods continued. In: Proceedings of the International Conference on Intelligent Robots and Systems (IROS), IEEE, vol 1, pp 454–459
- Hastings W (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57(1):97–109
- Hester T, Stone P (2008) Negative information and line observations for Monte Carlo localization. In: Proceedings of the International Conference on Robotics and Automation (ICRA), IEEE, pp 2764–2769
- Hoffman J, Spranger M, Gohring D, Jungel M (2005) Making use of what you don’t see: negative information in Markov localization. In: Proceedings of the International Conference on Intelligent Robots and Systems (IROS), IEEE, pp 2947–2952
- Iocchi L, Matsubara H, Weitzenfeld A, Zhou C (2009) RoboCup 2008: Robot Soccer World Cup XII, vol 5399. Springer
- Jensfelt P, Kristensen S (2001) Active global localization for a mobile robot using multiple hypothesis tracking. *Transactions on Robotics and Automation* 17(5):748–760
- Jochmann G, Kerner S, Tasse S, Urbann O (2012) Efficient multi-hypotheses unscented Kalman filtering for robust localization. In: RoboCup 2011: Robot Soccer World Cup XV, Springer, pp 222–233
- Kalman RE, et al (1960) A new approach to linear filtering and prediction problems. *Journal of basic Engineering* 82(1):35–45
- Kantor G, Singh S (2002) Preliminary results in range-only localization and mapping. In: Proceedings of the International Conference on Robotics and Automation (ICRA), IEEE, vol 2, pp 1818–1823
- Kaplan K, Celik B, Meriçli T, Meriçli C, Akin H (2006) Practical extensions to vision-based Monte Carlo localization methods for robot soccer domain. In: RoboCup 2005: Robot Soccer World Cup IX, Springer, pp 624–631
- Kümmerle R, Triebel R, Pfaff P, Burgard W (2008) Monte carlo localization in outdoor terrains using multilevel surface maps. *Journal of Field Robotics* 25(6-7):346–359
- Lenser S, Veloso M (2000) Sensor resetting localization for poorly modelled mobile robots. In: Proceedings of the International Conference on Robotics and Automation (ICRA), IEEE, vol 2, pp 1225–1232
- Lenser S, Bruce J, Veloso M (2001) Cmpack: A complete software system for autonomous legged soccer robots. In: Proceedings of the International Conference on Autonomous Agents, ACM, pp 204–211
- Leonard J, Durrant-Whyte H (1991) Mobile robot localization by tracking geometric beacons. *Transactions on Robotics and Automation* 7(3):376–382

- Levinson J, Montemerlo M, Thrun S (2007) Map-based precision vehicle localization in urban environments. In: Proceedings of the Robotics: Science and Systems Conference (RSS)
- Liemhetcharat S, Coltin B, Veloso M (2010) Vision-based cognition of a humanoid robot in standard platform robot soccer. In: Proceedings of the Workshop on Humanoid Soccer, Humanoids 2010
- Marchetti L, Grisetti G, Iocchi L (2007) A comparative analysis of particle filter based localization methods. In: RoboCup 2006: Robot Soccer World Cup X, Springer, pp 442–449
- Martín F, Matellán V, Barrera P, Cañas J (2007) Localization of legged robots combining a fuzzy-Markov method and a population of extended Kalman filters. *Robotics and Autonomous Systems* 55(12):870–880
- Menegatti E, Zoccarato M, Pagello E, Ishiguro H (2004) Image-based monte carlo localisation with omnidirectional images. *Robotics and Autonomous Systems* 48(1):17–30
- Metropolis N, Rosenbluth A, Rosenbluth M, Teller A, Teller E (1953) Equation of state calculations by fast computing machines. *The journal of chemical physics* 21:1087
- Milstein A, Sánchez J, Williamson E (2002) Robust global localization using clustered particle filtering. In: Proceedings of the National Conference on Artificial Intelligence (AAAI), pp 581–586
- Neira J, Tardós JD (2001) Data association in stochastic mapping using the joint compatibility test. *Transactions on Robotics and Automation* 17(6):890–897
- Odakura V, Sacchi R, Ramisa A, Bianchi R, Costa A (2009) The use of negative detection in cooperative localization in a team of four-legged robots. *Anais do Simpósio Brasileiro de Automação Inteligente*
- Özkucur N, Akn H (2011) Localization with non-unique landmark observations. In: RoboCup 2010: Robot Soccer World Cup XIV, vol 6556, Springer, pp 72–81
- Pfaff P, Burgard W, Fox D (2006) Robust Monte-Carlo localization using adaptive likelihood models. In: European Robotics Symposium 2006, Springer, pp 181–194
- Pitt M, Shephard N (1999) Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association* pp 590–599
- Porta J, Verbeek J, Kröse B (2005) Active appearance-based robot localization using stereo vision. *Autonomous Robots* 18(1):59–80
- Quinlan M, Middleton R (2010) Multiple model Kalman filters: a localization technique for RoboCup soccer. In: RoboCup 2009: Robot Soccer World Cup XIII, Springer, pp 276–287
- Ratter A, Hengst B, Hall B, White B, Vance B, Sammut C, Claridge D, Nguyen H, Ashar J, Pagnucco M, Robinson S, Zhu Y (2010) rUNSwift team report 2010
- Rekleitis I (2004) A particle filter tutorial for mobile robot localization. Technical Report TR-CIM-04-02, Centre for Intelligent Machines, McGill University
- Röfer T, Jungel M (2003) Vision-based fast and reactive Monte-Carlo localization. In: Proceedings of the International Conference on Robotics and Automation (ICRA), IEEE, vol 1, pp 856–861
- Roth M, Vail D, Veloso M (2003) A real-time world model for multi-robot teams with high-latency communication. In: Proceedings of the International Conference on Intelligent Robots and Systems (IROS), IEEE, vol 3, pp 2494–2499
- Rybski P, Veloso M (2009) Prioritized multihypothesis tracking by a robot with limited sensing. *EURASIP Journal on Advances in Signal Processing* 2009:138–154
- Schulz H, Liu W, Stückler J, Behnke S (2011) Utilizing the structure of field lines for efficient soccer robot localization. In: RoboCup 2010: Robot Soccer World Cup XIV, Springer, pp 397–408
- Seekircher A, Laue T, Röfer T (2011) Entropy-based active vision for a humanoid soccer robot. In: RoboCup 2010: Robot Soccer World Cup XIV, Springer, pp 1–12
- Sridharan M, Kuhlmann G, Stone P (2005) Practical vision-based Monte Carlo localization on a legged robot. In: Proceedings of the International Conference on Robotics and Automation (ICRA), IEEE, pp 3366–3371
- Vail D, Veloso M (2003) Dynamic multi-robot coordination. In: *Multi-Robot Systems: From Swarms to Intelligent Automata*, Volume II, Kluwer Academic Publishers, pp 87–100
- Vlassis N, Terwijn B, Krose B (2002) Auxiliary particle filter robot localization from high-dimensional sensor observations. In: Proceedings of the International Conference on Robotics and Automation (ICRA), IEEE, vol 1, pp 7–12
- Whelan T, Stdli S, McDonald J, Middleton R (2011) Efficient localization for robot soccer using pattern matching. In: Proceedings of the International ISO LA Workshop on Software Aspects of Robotic Systems
- Winner E, Veloso M (2000) Multi-fidelity robotic behaviors: Acting with variable state information. In: Proceedings of the National Conference on Artificial Intelligence (AAAI), pp 872–878
- Wolf J, Burgard W, Burkhardt H (2005) Robust vision-based localization by combining an image-retrieval system with monte carlo localization. *Transactions on Robotics* 21(2):208–216
- Zickler S, Laue T, Birbach O, Wongphati M, Veloso M (2010) SSL-Vision: The shared vision system for the RoboCup small size league. In: RoboCup 2009: Robot Soccer World Cup XIII, Springer, pp 425–436