

# Multi-Observation Sensor Resetting Localization with Ambiguous Landmarks

**Brian Coltin and Manuela Veloso**

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

## Abstract

Successful approaches to the robot localization problem include Monte Carlo particle filters, which estimate non-parametric localization belief distributions. However, particle filters fare poorly at determining the robot's position without a good initial hypothesis. This problem has been addressed for robots that sense visual landmarks with sensor resetting, by performing sensor-based resampling when the robot is lost. For robots that make sparse, ambiguous and noisy observations, standard sensor resetting places new location hypotheses across a wide region, in positions that may be inconsistent with previous observations. We propose Multi-Observation Sensor Resetting, where observations from multiple frames are merged to generate new hypotheses more effectively. We demonstrate experimentally in the robot soccer domain on the NAO humanoid robots that Multi-Observation Sensor Resetting converges more efficiently to the robot's true position than standard sensor resetting, and is more robust to systematic vision errors.

## 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 to interact intelligently with the physical world fundamentally depends on its ability to self-localize, or determine its own pose relative to the environment. The robot must localize quickly in response to any real-time constraints tied to the successful completion of its task, and also robustly, in the presence of noisy, ambiguous, and even incorrect sensing.

The localization problem has been extensively studied, and a prevalent solution is the use of Monte Carlo particle filters, where a set of "particles" model multiple 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 implemented for robots with various types of sensory inputs, including 2D planar LIDAR scans, 3D point clouds, visual information, and even the signal strength of WiFi access points. Different sensor

modalities offer different challenges and advantages. We are interested specifically in localizing with visual landmarks.

Localizing based on visual landmarks is in some ways a more challenging problem than localizing with point cloud sensors. The first challenge is simply the quantity of data available. LIDAR sensors return hundreds of distances in every scan, but cameras have limited fields of view and landmarks are more sparse, so the robot may only be able to detect one or two landmarks in a single frame. Additionally, with cameras where the pan and tilt angles are controllable, the sensory input is active rather than passive, and the primary task of the robot is not to localize but to perform some task; hence, camera usage must be shared between localizing and other tasks.

The second challenge is that systems which detect visual landmarks are imprecise and prone to errors. In recognizing visual landmarks, false positives, false negatives and errors in location are highly probable. Localization algorithms must be robust to these errors.

Despite these challenges, the difficulty of the localization problem with visual landmarks is mitigated by the fact that individual observations are much more informative than a single laser reading. By sensing two unambiguous visual landmarks, an estimate of the robot's position can be triangulated. Sensor resetting localization inserts additional hypotheses generated from sensing when the robot is uncertain of its position (Lenser and Veloso 2000). Sensor resetting implementations must perform a delicate balancing act between exploration and exploitation: with too much sensor resetting, the particles and the robot's position will jump quickly and abruptly in response to noise and vision errors, but with too little, the particle filter will not converge to the robot's true position in a timely fashion. However, sensor resetting is less effective when landmarks are *ambiguous* and observations are less informative.

We postulate that standard sensor resetting errs too strongly on the side of exploration due to the way that new hypotheses are generated based on every set of observations, including observations of only a single landmark. Standard sensor resetting only considers the observations from a single camera frame; however, more informed pose hypotheses can be generated by considering multiple observations across multiple frames. We propose Multi-Observation Sensor Resetting (MOSR) localization as an extension to stan-

standard sensor resetting. MOSR localization converges more quickly and accurately by relaxing the assumption that observations from separate frames of the camera are independent, and by using multiple observations to generate fewer but more informed new hypotheses for sensor resetting. In addition to speedy convergence times, MOSR localization is more robust to certain types of systematic errors in vision, including false positives.

In this paper, we first provide a brief overview of related work in self-localization with a focus on the use of visual landmarks, followed by a description of a complete algorithm for self-localization. We then present in detail the algorithm for Multi-Observation Sensor Resetting Localization. Finally, we extensively demonstrate the effectiveness of the algorithm experimentally in the domain of RoboCup Standard Platform Soccer on the NAO humanoid robots.

## Background

Particle filters for mobile robot localization maintain a probability distribution of the robot’s pose  $l$  given sensory inputs  $s$  and control actions  $u$ , that include the motion of the robot and camera. The probability distribution is represented as a set of particles, or location hypotheses  $p_i$  with weights  $w_i$ . The weights represent the likelihood that the robot is in the associated pose (Dellaert et al. 1999).

With every observation  $s$  and control action  $u$ , the particles are updated in a three step process:

1. **Predict Step.** The particles move based on the motion model of the robot. The motion of the particles is sampled from a distribution representing the uncertainty of the motion model.
2. **Update Step.** The weights  $w_i$  of the particles are multiplied by  $p(s|l_i)$ , or the likelihood of making the observed sensor readings given that the robot is in pose  $p_i$ . The computation of  $p(s|l)$  is referred to as the *sensor model*.
3. **Resample Step.** New particles are chosen probabilistically by sampling according to the weights of the old particles, so that more hypotheses may be modelled in the areas of greatest likelihood.

Particle filters are able to model multi-modal distributions in a computationally inexpensive manner. However, with a limited number of particles the entire configuration space of the robot cannot be covered.

When the observations  $s$  are visual landmarks, as in the RoboCup domain,  $p(l|s)$  can be computed directly from observations in addition to  $p(s|l)$ . Sensor resetting localization (SRL) extends standard particle filters by using  $p(l|s)$  to place new hypotheses directly at likely locations of the robot. Each particle is replaced with a particle generated directly from sensing with a probability  $p_{reset}$  inversely proportional to the total weight of the particles after the resampling step (see Algorithm 1). 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’ locations are unlikely and they are chosen anew from  $p(l|s)$  (Lenser and Veloso 2000).

Resampling from  $p(l|s)$  depends on the distance and relative angle to the landmarks from the robot. If only a single

---

**Algorithm 1** `localize( $p, w, s, u$ )`: Sensor Resetting Localization and Adaptive-MCL.  $k, \nu, \alpha_l$  and  $\alpha_s$  are constants.  $N$  is the number of particles,  $p_i$  are the particle poses, and  $w_i$  are the particle weights.

---

```

for  $i = 1$  to  $N$  do
   $p_i \leftarrow$  motion_predict( $p_i, u$ )
   $w_i \leftarrow$  vision_update( $p_i, w_i, s$ )
end for
 $p^{old} \leftarrow p, w^{old} \leftarrow w$ 
 $\bar{w} \leftarrow \sum w_i$ 
if Standard Sensor Resetting then
   $p_{reset} \leftarrow (1 - \frac{\bar{w}}{kN})$ 
else if Adaptive-MCL then
   $\bar{w}_l \leftarrow \bar{w}_l + \alpha_l(\bar{w} - \bar{w}_l)$ 
   $\bar{w}_s \leftarrow \bar{w}_s + \alpha_s(\bar{w} - \bar{w}_s)$ 
   $p_{reset} \leftarrow \max\{0, 1 - \nu \frac{\bar{w}_s}{\bar{w}_l}\}$ 
end if
for  $i = 1$  to  $N$  do
  if random() <  $p_{reset}$  then
     $p_i \leftarrow$  sensor_reset( $s$ )
  else
     $(p_i, w_i) \leftarrow$  sample  $p_i^{old}$  from  $p^{old}$  w/ prob.  $\propto w_i^{old}$ 
  end if
end for

```

---

landmark is visible, the most likely locations are spread in a circle around that landmark at the observed distance and angle to the landmark. If two or more landmarks are visible simultaneously, sensor resetting is performed at the intersection of the two circles. Sampling from  $p(l|s)$  accounts for noise in the vision process, so the resampled particles are not all at the same location (Lenser and Veloso 2000).

One shortcoming of sensor resetting is the way  $p_{reset}$  is chosen. 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 position and vision 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,  $\bar{w}_l$  and  $\bar{w}_s$ , and mitigates this effect (see Algorithm 1) (Gutmann and Fox 2002).

Another extension to MCL specific to visual landmarks is the use of negative information. For point cloud sensor models, if the laser hits no obstacle this information is used to compute  $p(s|l)$ , but for visual landmarks only positive observations are generally used. Negative information gained from a lack of expected observations may be included in the sensor model (Hoffman et al. 2005).

A final challenge of localizing with visual landmarks is that perception is active rather than passive—the robot decides *what* to look at. In robot soccer, the robot must simultaneously localize and track the ball. 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 (Coltin et al. 2010). When

observing landmarks, RoboCup teams commonly use fixed scanning head motions or stare at each in a sequence of landmarks. Another approach is to make the observations that are expected to reduce the entropy the most in the underlying particle distribution for localization (Seekircher, Laue, and Röfer 2010).

The leading teams in the RoboCup Standard Platform League currently use variants of Sensor Resetting Localization (Burchardt, Laue, and Röfer 2010) (Hester and Stone 2008) or a combination of a Kalman filter and particle filter (Ratter et al. 2010). These implementations all include ambiguous observations in the sensor model  $p(s|l)$ , but typically only make limited use of ambiguous landmarks for sensor resetting (i.e., only resetting from goal posts).

### Localization in RoboCup SPL Soccer

Before presenting the Multi-Observation Sensor Resetting algorithm, we discuss the problem domain of RoboCup Standard Platform League robot soccer.

The SPL plays using the NAO humanoid robots. The NAO senses with two cameras: one on its forehead and one in its chin, although only one camera may be used at a time. The field of view of the camera is very limited (see Figure 1). The NAO can freely move its head to look at landmarks.

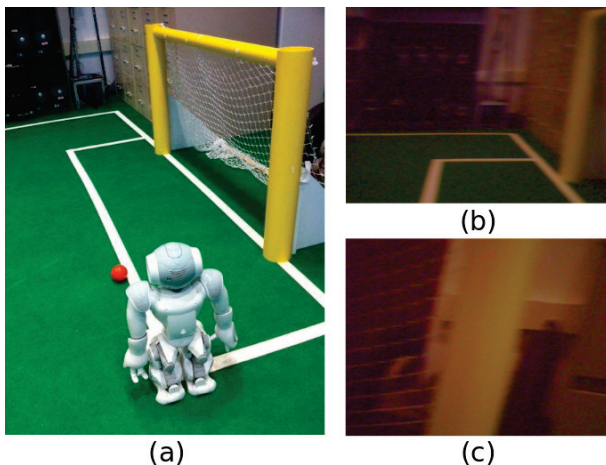


Figure 1: (a) The NAO humanoid robot stands on the field near the yellow goal. (b, c) Two images are shown from the robot’s camera with the head at different angles. The field of view is limited, and the robot cannot see the top bar to determine whether it sees a left or right post.

The NAOs play soccer on a 6m x 4m playing field. The visual landmarks on the field include goal posts, corners, the center circle and lines. Our vision system 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 “unknown” goal post.

The remaining landmarks observed on the field include the white lines marking the borders of the field, the center

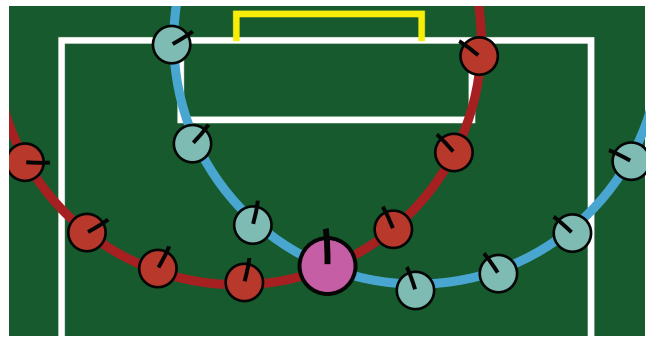


Figure 2: The circles surrounding the two goal posts indicate the possible robot positions given observations of the left and right goal posts (or one observation of an “unknown” goal post). Possible robot poses from sensor resetting are drawn on the circles. The larger pose at the circles’ intersection represents the hypothesis generated by sensor resetting from both goal posts.

line and circle, and the goal boxes. Our vision system detects line segments, as well as the intersection of line segments at corners and the center circle. There are three types of corners: ‘L’ corners, ‘T’ corners, and ‘X’ corners. A total of eight ‘L’ corners mark the corners of the field and goal boxes, six ‘T’ corners mark the intersections of the field border with the goal boxes and center line, and two ‘X’ corners denote the penalty kickoff points on both halves of the field.

It should be noted that the majority of the field markers are *ambiguous*, and if observed, could actually correspond to multiple locations. A detected unknown goal post could be one of two landmarks, a detected corner refers to between two and eight landmarks, and an observed line could be paired with nearly any line segment on the field.

Our team’s localization algorithm uses the Adaptive-MCL variant of Sensor Resetting Localization shown in Algorithm 1. It uses several other extensions to MCL, including low variance resampling, the use of negative information and a variable number of particles depending on uncertainty (Liemhetcharat, Coltin, and Veloso 2010).

In single frame sensor resetting localization, the `sensor_reset` function only places hypotheses near some of the least ambiguous landmarks, goal posts. If one unambiguous goal post is seen, the possible locations for the robot form a circle around that goal post, and locations are selected uniformly at random for the new hypotheses. For an unknown goal post, a random post is selected to place the new hypothesis around (see Figure 2). In every case, noise is added to the observations before generating a new sensor resetting hypothesis. The noise is proportional to the expected observation noise in the sensor model. The addition of noise encourages diversity of particles by not placing them all in the same location or the exact same distance and angle from a goal post.

This form of sensor resetting raises a few concerns:

1. **Spread of Particles.** Standard sensor resetting based on a single landmark spreads the distribution of particles over



a wide region surrounding the landmark, which may be inconsistent with previous observations.

2. **Ambiguous Landmarks.** Standard sensor resetting does not generate hypotheses based on corner observations.
3. **False Positives.** Standard sensor resetting is sensitive to false positives from vision. It deletes old hypotheses and places particles around incorrectly observed landmarks.

Multi-Observation Sensor Resetting addresses each issue.

## Multi-Observation Sensor Resetting

We introduce Multi-Observation Sensor Resetting, which is based on Algorithm 1 and uses the notion of a *scan*. A scan is a sequence of visual landmarks detected over a period of time, the duration of which is determined by some arbitrary function. During a scan, the robot may move its camera to observe a sequence of landmarks. In MOSR, no sensor resetting is performed while a scan is in progress, but the robot records observations made in a list, *obs*. The list *obs* includes everything except lines: goal posts (left, right, or ambiguous), corners, and circles. As the robot moves, these observations are updated based on odometry. Then, when the scan completes, an extra phase of the localization algorithm is performed that includes both a vision update and resample step. Sensor resetting is performed in this phase using *all* of the observations from the scan (see Algorithm 2). MOSR relaxes the assumption in standard sensor resetting that observations are independent to generate better hypotheses. The computation of  $p_{reset}$  is identical to Adaptive-MCL.

---

**Algorithm 2**  $\text{mosr}(p, w, obs)$ : The sensor resetting phase of Multi-Observation Sensor Resetting Localization, performed after a scan completes.

---

```

for  $i = 1$  to  $N$  do
   $w_i \leftarrow \text{vision\_update}(p_i, w_i, obs)$ 
end for
for  $i = 1$  to  $N$  do
  if  $\text{random}() < p_{reset}$  then
     $p_i \leftarrow \text{sensor\_reset}(obs)$ 
  else
     $(p_i, w_i) \leftarrow \text{sample } p_i^{old} \text{ from } p^{old} \text{ w/ prob. } \propto w_i^{old}$ 
  end if
end for

```

---

Since there are typically more than two landmarks in *obs* observed over the course of a scan, a new algorithm for `sensor_reset` is needed to reflect  $p(l|s)$  given multiple observations (see Algorithm 3). When a new particle is generated with sensor resetting at the end of a scan, we select two observations uniformly at random from *obs*. A more informed selection method may be employed, such as favoring two observations of different landmarks. Noise is added to the selected observations based on the sensor model. For ambiguous observations such as corners or unknown goal posts, we associate the observation with a random global position of a landmark from the same class (e.g., L corner, yellow goal post, center circle).

We then attempt to triangulate the robot’s position based on the selected observations and their associated landmarks by finding the intersection of the circles that are the observed distances from the landmarks. If non-matching landmarks were chosen, the triangulation fails. We also may check that the predicted position is consistent with other observations before accepting it. In our experiments we used the identity function for verification and still achieved good results, but this can be improved by choosing other verification functions. If the hypothesis is invalid, we select two new observations and associated global landmark positions and try again. If after a fixed number of attempts we do not succeed, we give up and revert to resampling from the particles. Additionally, if only a single landmark is seen in the scan, we fall back to placing the observations at random around that landmark as in the original sensor resetting algorithm.

---

**Algorithm 3**  $\text{sensor\_reset}(obs)$ : Generating new hypotheses based on sensing for MOSR Localization.

---

```

for  $i = 1$  to  $K$  do
  for  $j = 1$  to  $2$  do
     $o_j \leftarrow \text{random observation from } obs$ 
     $\text{distance}(o_j) \leftarrow \mathcal{N}(\text{distance}(o_j), \sigma_x^2)$ 
     $\text{angle}(o_j) \leftarrow \mathcal{N}(\text{angle}(o_j), \sigma_\theta^2)$ 
     $l_j \leftarrow \text{random global landmark of same class as } o_j$ 
  end for
   $p \leftarrow \text{triangulate}(o_1, l_1, o_2, l_2)$ 
  if  $\text{verify}(p, obs)$  then
    return  $p$ 
  end if
end for
return failure

```

---

In our system for robot soccer, the 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. The robot observes landmarks with three different types of scans:

1. A **horizontal** scan, where the robots 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 location 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 that would most reduce entropy (Seekircher, Laue, and Röfer 2010).

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 position so it knows where the landmarks are. Thus, we initially use the horizontal scan to roughly determine the robot’s position and then switch to one of the other scans. The robot may move while scanning.

MOSR localization has several advantages over standard localization. First, MOSR generates fewer new hypotheses, but the hypotheses are more informed. This helps MOSR to maintain good estimates by spreading the particles less and

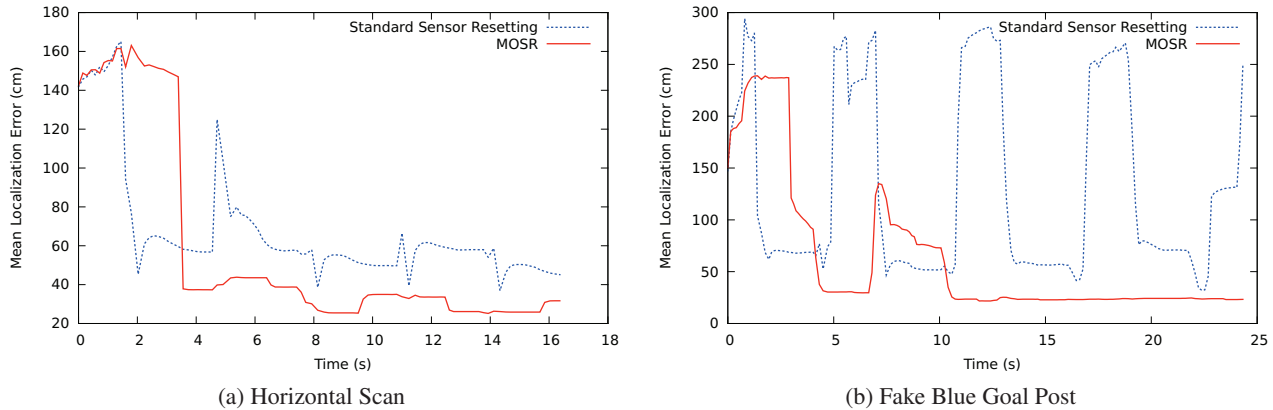


Figure 3: Localization error from (a) repeatedly horizontally scanning in place on a standard field, and (b) with an extra, “fake” blue goal post. In the second graph, the spikes in standard sensor resetting error correspond to the sighting of the blue goal post.

enables faster convergence. In addition, MOSR improves robustness to false positives in vision, gained both from performing less resampling and requiring a pair of consistent observations to reset based on. Next, we demonstrate all of these advantages experimentally.

### Experimental Results

To test Multi-Observation Sensor Resetting, we compared it directly with standard sensor resetting using Adaptive-MCL. The MOSR localization implementation was identical to the standard sensor resetting implementation in every respect except for when sensor resetting is used. The standard sensor resetting algorithm also used the new `sensor_reset` function created for MOSR, which is identical to the old function for one or two observed landmarks.

For each experiment, a NAO robot ran on half of the soccer field. A pattern was attached to the robot’s head and monitored by an overhead camera using SSL-Vision (Zickler et al. 2009). The robot’s state and the position information from SSL-Vision were recorded in a log file. 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 and the standard deviation of the particles from the final pose were computed. The experiment was run with several robot configurations and behaviors.

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 3a).

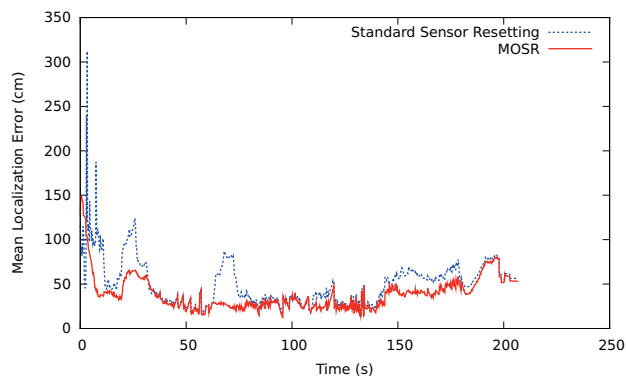
Next, we chose to simulate the “blue jeans problem” in the SPL. Blue jeans worn by spectators may be consistently misidentified as goal posts if no heuristics are used to discard them. We use the same experimental setup as before, scanning in place with a horizontal scan, but place an actual

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 3b). MOSR does not generate new hypotheses using the goal post because it requires two observations to reset from, and the location of the fake goal post is inconsistent with the two yellow posts.

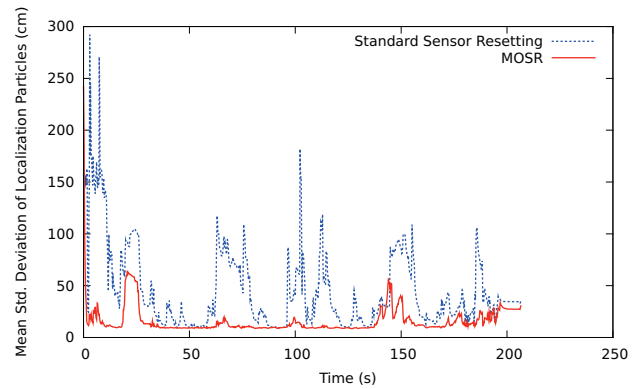
We also test Multi-Observation sensor resetting while the robot is in motion, both while constantly performing the landmark scan and while performing the entropy-based scan, with the horizontal scan as a fallback when the robot is lost. The robot repeatedly chose a random location on one half of the field and moved to it. The results are shown in Figure 4 for both the landmark scan and the entropy-based scan. MOSR converges to the neighborhood of the robot’s true position faster and tends to remain closer to the true position than standard sensor resetting. Furthermore, the particles representing the distribution of locations have a smaller variance with MOSR, since fewer hypotheses invalidated by other nearby observations are generated.

### Conclusion

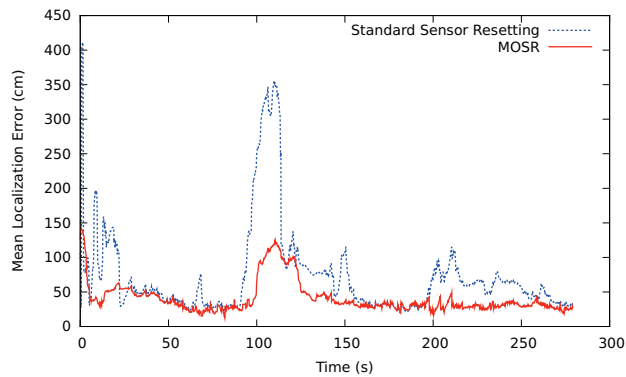
Mobile robots localize using visual information in a variety of applications. With limited fields of view, robots may have only partial or limited perception of visual landmarks in a single frame. We introduced MOSR localization, which generates new hypotheses directly from sensing only at the end of a scan, based on multiple visual frames, to resolve the ambiguity of a single landmark. MOSR converges faster to the robot’s true position than standard sensor resetting and is less prone to jumping away from the robot’s true position based on erroneous observations. We demonstrated MOSR’s effectiveness experimentally in the robot soccer domain, but it is applicable to any system where a robot localizes based on visual landmarks.



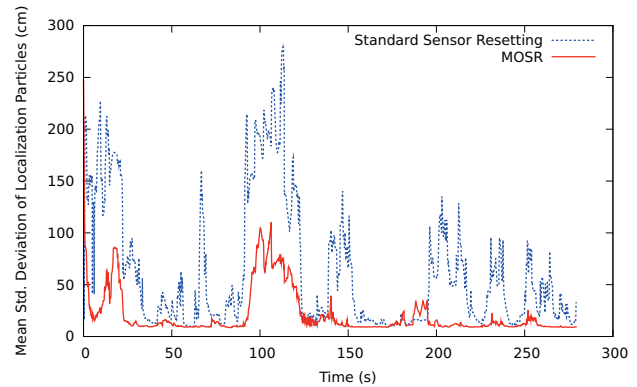
(a) Landmark Scan Error



(b) Landmark Scan Standard Deviation



(c) Entropy Scan Error



(d) Entropy Scan Standard Deviation

Figure 4: Mean localization error and the mean standard deviation of the particle distribution for both the landmark scan and entropy-based scan while the robot is in motion. At the large spike in the error for standard sensor resetting using the entropy-based scan, a false positive was detected outside the field.

## Acknowledgments

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

- Burchardt, A.; Laue, T.; and Röfer, T. 2010. Optimizing particle filter parameters for self-localization. In *RoboCup 2010: Robot Soccer World Cup XIV*, Lecture Notes in AI.
- Coltin, B.; Liemhetcharat, S.; Mericli, C.; Tay, J.; and Veloso, M. 2010. Multi-humanoid world modeling in standard platform robot soccer. In *Proc. of Humanoids*.
- Dellaert, F.; Fox, D.; Burgard, W.; and Thrun, S. 1999. Monte Carlo localization for mobile robots. In *ICRA*.
- Gutmann, J.-S., and Fox, D. 2002. An experimental comparison of localization methods continued. In *Proc. of IROS*.
- Hester, T., and Stone, P. 2008. Negative information and line observations for Monte Carlo localization. In *ICRA*.
- Hoffman, J.; Spranger, M.; Gohring, D.; and Jungel, M. 2005. Making use of what you don't see: negative information in Markov localization. In *Proc. of IROS*.
- Lenser, S., and Veloso, M. 2000. Sensor resetting localization for poorly modelled mobile robots. In *Proc. of ICRA*.
- Liemhetcharat, S.; Coltin, B.; and Veloso, M. 2010. Vision-based cognition of a humanoid robot in standard platform robot soccer. In *Proc. of the 5th Workshop on Humanoid Soccer, Humanoids 2010*.
- Ratter, A.; Hengst, B.; Hall, B.; White, B.; Vance, B.; Sammut, C.; Claridge, D.; Nguyen, H.; Ashar, J.; Pagnucco, M.; Robinson, S.; and Zhu, Y. 2010. rUNSwift team report 2010.
- Seekircher, A.; Laue, T.; and Röfer, T. 2010. Entropy-based active vision for a humanoid soccer robot. In *RoboCup 2010: Robot Soccer World Cup XIV*, Lecture Notes in AI.
- Zickler, S.; Laue, T.; Birbach, O.; Wongphati, M.; and Veloso, M. 2009. SSL-Vision: The shared vision system for the RoboCup small size league. In *Proc. of the RoboCup Symposium*.