Improved Particle Filtering for Pseudo-Uniform Belief Distributions in Robot Localisation

David Budden and Mikhail Prokopenko

Information and Communications Technologies Centre, Adaptive Systems
Commonwealth Scientific and Industrial Research Organisation (CSIRO)
PO Box 76, Epping, NSW 2121, Australia
{david.budden,mikhail.prokopenko}@csiro.au

Abstract. Self-localisation, or the process of an autonomous agent determining its own position and orientation within some local environment, is a critical task in modern robotics. Although this task may be formally defined as a simple transformation between local and global coordinate systems, the process of accurately and efficiently determining this transformation is a complex task. This is particularly the case in an environment where localisation must be inferred entirely from noisy visual data, such as the RoboCup robot soccer competitions. Although many effective probabilistic filters exist for solving this task in its general form, pseudo-uniform belief distributions (such as those arising from course-grain observations) exhibit properties allowing for further performance improvement. This paper explores the RoboCup 2D Simulation League as one such scenario, approximating the artificially constrained noise models as uniform to derive an improved particle filter for self-localisation. The developed system is demonstrated to yield from 38.2 to 201.3% reduction in localisation error, which is further shown as corresponding with a 6.4% improvement in goal difference across approximately 750 games.

Keywords: Robotics, localisation, particle filter, robot soccer.

1 Introduction

The RoboCup 2D Simulation League incorporates a number of critical challenges in the areas of artificial intelligence, machine learning and distributed computing [6,9]. These include, but are not limited to:

- Distributed client/server model, introducing the challenges of fragmented, localised and imprecise information (both in terms of noise and latency) about the environment [8].
- Asynchronous perception-action activity, and a limited window of opportunity to perform a desired action [5].
- No centralised controllers or central world model, resulting in a lack of global vision or localisation information [10,11].

S. Behnke et al. (Eds.): RoboCup 2013, LNAI 8371, pp. 385–395, 2014.
© Springer-Verlag Berlin Heidelberg 2014

This paper focuses on the final point; specifically, how to infer localisation information from noisy observations of the environment.

In its simplest form, the task of localisation (particularly self-localisation of an agent) can be viewed as a problem of coordinate transformation [13]; specifically, determining the transformation between the agent's local coordinate system and the environment's global coordinate system. Knowledge of this transformation allows the agent to consider global features (such as unique markers and goalposts) with reference to its own coordinate frame, facilitating navigation and execution of more complex behaviours.

Knowing the position and orientation of an agent is both sufficient and necessary for determining this coordinate transformation. As the RoboCup 2D Simulation League does not provide each agent with global vision information, the position and orientation must be inferred from observing unique and stationary *markers* within the environment. In a traditional robotics scenario, the agent employs physical sensors (such as cameras or range-finders) and computer vision techniques [3,4] to infer the relative position and orientation of such markers. These observations typically correspond with a well-defined Gaussian error, allowing for the effective applications of Kalman and particle filtering techniques [12,13].

Although the need for image processing is removed, the RoboCup 2D Simulation League introduces a number of unique challenges to the task of self-localisation. Particularly, the artificial observational noise introduced by the server is non-Gaussian (and approximately uniform within well-defined radial regions). The remainder of this paper presents a formal definition of the approximate positional error region given an arbitrary number of observations, and describes a modified particle filter that employs knowledge of the noise model to increase self-localisation accuracy. A linear approximation step is introduced to reduce the algorithm's time complexity, with the final performance results compared to the well-known agent2D implementation developed by Akiyama *et. al.* [1].

2 Particle Filter

The *particle filter* is a nonparametric implementation of the *Bayes filter*, where the posterior distribution $bel(x_t)$ is approximated by a set of random state samples (*particles*) drawn from this posterior [13]. Concretely, the set $\mathcal{X}_t$ of M particles are denoted

$$\mathcal{X}_t = \left\{ x_t^{[1]}, x_t^{[2]}, \ldots, x_t^{[M]} \right\},$$

where each particle $x_t^{[m]}$ $(1 \leq m \leq M)$ is an instantiation of the state at time t. Such particles are proportional to the Bayes filter posterior distribution

$$x_t^{[m]} \sim bel(x_t) = p(x_t | z_{1:t}, u_{1:t}),$$

where $z_{1:t}$ is the set of all previous *observations* and $u_{1:t}$ the set of all previous *actions*. Conceptually, the particle filter consists of the following steps [13]

Algorithm 1. `particleFilter`$(\mathcal{X}_{t-1}, u_t, z_t)$

$\overline{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$
for $m = 1 \ldots M$ **do**
  Sample $x_t^{[m]} \sim p(x_t | u_t, x_{t-1}^{[m]})$
  $w_t^{[m]} = p(z_t | x_t^{[m]})$
  $\overline{\mathcal{X}}_t = \overline{\mathcal{X}}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$
end for
for $m = 1 \ldots M$ **do**
  draw i with probability $\propto w_t^{[i]}$
  add $x_t^{[i]}$ to $\mathcal{X}_t$
end for
return $\mathcal{X}_t$

1. **Action update:** Sample M particles $\overline{\mathcal{X}}_t = \left\{ x_t^{[1]}, x_t^{[2]}, \ldots, x_t^{[M]} \right\}$ at random from the state transition distribution $p(x_t | z_{1:t}, u_{1:t})$. These particles form a nonparametric approximation of $\overline{bel}(x_t)$, where $\overline{bel}(x_t) = p(x_t | z_{1:t-1}, u_{1:t})$ is the *prediction* posterior before considering observation z_t.
2. **Calculate weights:** Calculate an *importance factor* (*weight*) $w_t^{[m]} = p(z_t | x_t^{[m]})$ for each particle $x_t^{[m]}$ given observation z_t. Add each weighted particle $\langle x_t^{[m]}, w_t^{[m]} \rangle$ to the temporary particle set $\bar{\mathcal{X}}_t$.
3. **Resample:** Draw M particles (with replacement) from $\overline{\mathcal{X}}_t$ and add to the particle set $\mathcal{X}_t$. The probability of drawing the particle $x_t^{[m]}$ is proportional to the corresponding weight $w_t^{[m]}$, and often results in the inclusion of duplicate particles (and consequent exclusion of particles with low importance weight).

Similarly to the Bayes filter, the particle filter recursively calculates the posterior $bel(x_t) \sim \mathcal{X}_t$ from the prior $bel(x_{t-1}) \sim \mathcal{X}_{t-1}$ by considering the observation z_t and action u_t at time t. A basic implementation `particleFilter`$(\mathcal{X}_{t-1}, u_t, z_t)$ of such a particle filter is defined in Algorithm 1.

3 Problem Model

As described in Sec 1, a lack of centralised controllers or a central world model within the RoboCup 2D simulation league results in a lack of global vision or localisation information. Agents are therefore required to self-localise by observing the positions of a number of uniquely identifiable markers, as illustrated in Fig. 1. The soccer server then introduces artificial noise to each observation. This section provides a concrete definition of this error model, as well as deriving the approximated self-localisation error that results from N marker observations.

3.1 Non-gaussian Observation Noise

As described in Sec. 1, an agent r in a RoboCup 2D Simulation League scenario self-localises by observing a number of fixed markers $m^{[n]}$, with the global

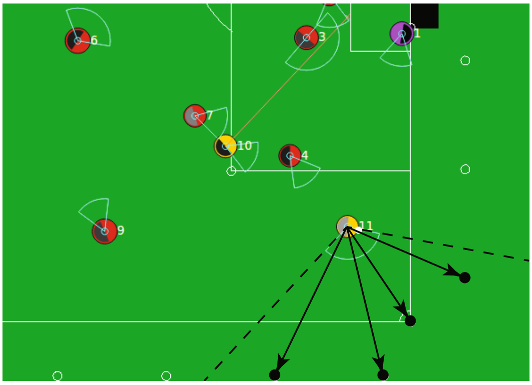


Fig. 1. Visualisation of an agent r making observations $r_m^{[n]}$ of markers $m^{[n]}$. The soccer server [6] introduces noise (as defined in (3.1)) to each observation, resulting in the self-localisation error model defined in Sec. 3.2.

position of each marker known a priori. If r were able to make noise-free measurements of the environment, a single observation of any such marker would result in zero localisation error. Instead, the soccer server [6] introduces artificial noise to each observation, such that the noise magnitude is proportional to the actual distance between agent and marker.

In a traditional localisation problem, such noise could be accurately modeled by a Gaussian distribution $\mathcal{N}(\mathbf{r}_m, \sigma)$ centered about the observed marker location $\mathbf{r}_m$, allowing for the implementation of well-known Kalman or particle filtering techniques [12,13]. The soccer server instead implements a more complex noise model $\mathcal{N}_s(d, \Delta_1, \Delta_2)$, involving the quantisation of exponential terms [6].

Concretely, Given the actual distance d of an agent from an observed marker, the noisy distance d' communicated by the server to the agent is defined as

$$\mathcal{N}_s : d \to d' = Q\left(e^{Q(\log(d), \Delta_1)}, \Delta_2\right),$$

where Δ_1 and Δ_2 represent the *quantisation step sizes* (set to 0.01 and 0.1 respectively), and the *quantisation function* Q is defined as

$$Q(x, y) = y \left\lceil \frac{x}{y} \right\rceil .$$

Although the nonparametric nature of a particle filter is suitable for such a distribution (by approximating the posterior distribution $bel(x_t)$ as a set of particles $\mathcal{X}_t$), further approximating this complex distribution as uniform within a radial region (corresponding with the agent's radial field of view[1]) allows for a simplified filter implementation. The remainder of this section derives a formal definition of the resultant self-localisation noise distribution.

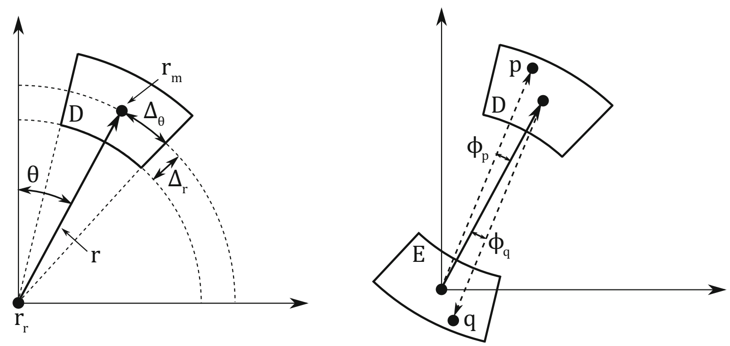
[1] The field of view of an agent r is chosen to be either π, $2\pi/3$ or $\pi/3$ radians, depending on the observed distance of the ball from r.

3.2 Self-localisation Error Model

One Observed Marker. The uncertainty in the position of a marker observed at $\mathbf{r}_m = (x_m, y_m)^\top$ resulting from observation $\mathbf{r}$ by an agent r at $\mathbf{r}_r = (x_r, y_r)^\top$ may be approximated as an annulus sector D of uniform probability, as illustrated in Fig. 2a. D may be concretely defined as

$$D = \left\{ \mathbf{p} \in \mathbb{R}^2 : \Big| ||\mathbf{p} - \mathbf{r}_r|| - ||\mathbf{r}_r - \mathbf{r}_m|| \Big| \leq \Delta_r \,, \quad |\phi_p| \leq \Delta_\theta \right\},$$

where Δ_θ and Δ_r parameterise the maximum angular and distance errors (as demonstrated in Fig. 2), and $\phi_p = \arccos\left(\frac{(\mathbf{r}_m - \mathbf{r}_r)\cdot(\mathbf{p}-\mathbf{r}_r)}{||\mathbf{r}_m - \mathbf{r}_r||\,||\mathbf{p}-\mathbf{r}_r||}\right)$. The uniform distribution D only has physical meaning when considering coordinates relative to the robot (as the exact position of the marker is known a priori). For the purpose of self-localisation, we are interested instead in the (closed) region E: the set of all possible locations $\mathbf{q}$ of the agent r resulting from observation $\mathbf{r}$.



(a) (b)

Fig. 2. Visualisation of the region D defining all possible locations of a marker m, given that an agent r at $\mathbf{r}_r$ observes m at $\mathbf{r}_m$ (a). As the position of m is known a priori, this observation results in the region E of all possible locations of the agent (b).

Fig. 2b illustrates the construction of the region E. As the position of the marker is known to be $\mathbf{r}_m$, it follows that a observation of the marker residing at $\mathbf{p} = (p_x, p_y)^\top \in D$ corresponds with an error of $\mathbf{p} - \mathbf{r}_m$. As the agent knows the absolute position of the marker a priori, it follows that positional self-belief may be defined as $\mathbf{q} = (q_x, q_y)^\top = \mathbf{r}_r - (\mathbf{p} - \mathbf{r}_m)$. It may be demonstrated that the equivalent mapping

$$T : \mathbf{p} \mapsto \mathbf{q} = (\mathbf{r}_r + \mathbf{r}_m) - \mathbf{p} \tag{1}$$

results in the image

$$E = \left\{ \mathbf{q} \in \mathbb{R}^2 : \mathbf{q}' = \mathcal{T}(\mathbf{r}_r)\mathcal{R}(\pi)\mathcal{T}(-\mathbf{r}_m)\mathbf{p}' \right\},$$

where $\mathbf{q}' = (q_x, q_y, 1)^\top$ and $\mathbf{p}' = (p_x, p_y, 1)^\top$ are the homogeneous representations of $\mathbf{p} \in D$ and $\mathbf{q} \in E$ respectively, $\mathcal{T}(\mathbf{v})_{3,3}$ is the homogenous translation matrix by 2-dimensional vector $\mathbf{v}$, and $\mathcal{R}(\theta)_{3,3}$ is the homogenous rotation matrix by angle θ[2]. Concretely,

$$\begin{aligned}
\mathbf{q}' = \left[\begin{array}{c}\mathbf{q}\\ \hline 1\end{array}\right] &= \mathcal{T}(\mathbf{r}_r)\mathcal{R}(\pi)\mathcal{T}(-\mathbf{r}_m)\mathbf{p}' \\
&= \left[\begin{array}{c|c}\mathbf{I}_2 & \mathbf{r}_r\\ \hline \mathbf{0} & 1\end{array}\right]\left[\begin{array}{c|c}-\mathbf{I}_2 & \mathbf{0}\\ \hline \mathbf{0} & 1\end{array}\right]\left[\begin{array}{c|c}\mathbf{I}_2 & -\mathbf{r}_m\\ \hline \mathbf{0} & 1\end{array}\right]\left[\begin{array}{c}\mathbf{p}\\ \hline 1\end{array}\right] \\
&= \left[\begin{array}{c}\mathbf{r_r} + \mathbf{r_m} - \mathbf{p}\\ \hline 1\end{array}\right] \\
\therefore \mathbf{q} &= (\mathbf{r}_r + \mathbf{r}_m) - \mathbf{p}\,,
\end{aligned}$$

where $\mathbf{I}_2$ is the 2-dimensional identity matrix. This result is equivalent to the mapping T defined in (1), confirming that E is the image of D under T. The region E, representing all possible locations of an agent r observing a single marker at $\mathbf{r}_m$ (as illustrated in Fig. 2), may finally be represented as

$$E = \left\{\mathbf{q} \in \mathbb{R}^2 : \Big| ||\mathbf{q} - \mathbf{r}_m|| - ||\mathbf{r}_r - \mathbf{r}_m|| \Big| \leq \Delta_r\,, \quad |\phi_q| \leq \Delta_\theta \right\}. \qquad (2)$$

This represents the intuitive definition of the region E; the region D rotated 180 degrees about the midpoint of $\mathbf{r}_r$ and $\mathbf{r}_m$.

Multiple Observed Markers. In a RoboCup 2D simulation league scenario, it is far more commonplace for an agent to observe not one, but N markers $m^{[n]}$ ($n \in [1, \cdots, N]$), each at a separate position known a priori. This results in N regions of uniform observation belief $D^{[1]}, \cdots, D^{[N]}$, which map to N corresponding regions of positional self-belief $E^{[1]}, \cdots, E^{[N]}$ as per Sec. 3.2. As the probability $p(\mathbf{q} \notin E^{[n]}) = 0$ ($\forall\, n \in [1, \cdots, N]$), it follows that the final region of self-belief E resulting from N such observations may be expressed as

$$E = \bigcap_{n=1}^{N} E^{[n]}, \quad E \neq \emptyset \qquad (3)$$

where $E^{[n]}$ is the region representing all possible locations of an agent r given an observation of marker $m^{[n]}$, defined as per (1).

4 Implementation

This section details the implementation of the Gliders2013 localisation system, based on the particle filter framework detailed in Sec. 2 and problem model of Sec. 3.

[2] Homogeneous representation of an $(n-1)$-dimensional transformation as an n-dimensional matrix allows for any affine transformation to be decomposed into the product of elementary shear, scaling, rotation and translation matrices [7].

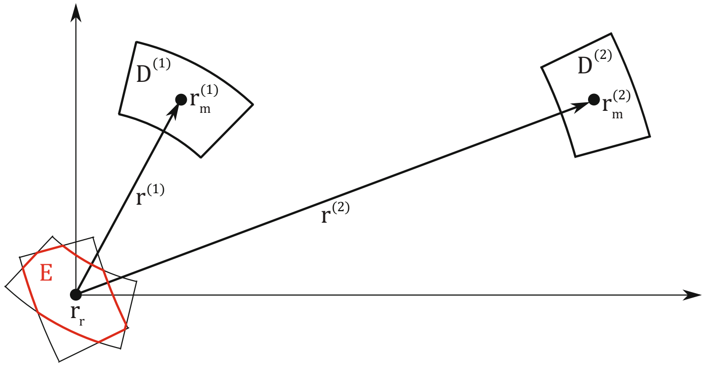


Fig. 3. Visualisation of the region E defining all possible locations of an agent r at $\mathbf{r}_r$, given that r observes markers $m^{[1]}$ and $m^{[2]}$ at $\mathbf{r}_m^{[1]}$ and $\mathbf{r}_m^{[2]}$ respectively

4.1 Action Update

As described in Sec. 2, the particle filter *action update* step involves sampling M particles $\overline{\mathcal{X}}_t = \left\{ x_t^{[1]}, x_t^{[2]}, \ldots, x_t^{[M]} \right\}$ at random from the state transition distribution $p(x_t|z_{1:t}, u_{1:t})$. In the Gliders2013 implementation, this nonparametric approximation of $\overline{bel}(x_t)$ is defined as

$$\overline{\mathcal{X}}_t = \left\{ x_t^{[m]} \in \mathbb{R}^2 : x_t^{[m]} = x_{t-1}^{[m]} + \mathbf{d}_{t-1} \quad \forall\, m \in [1, \cdots, M] \right\},$$

where $\mathbf{d}_{t-1}$ is the displacement resulting from the agent's action at time $t-1$, and the initial set of particles $\overline{\mathcal{X}}_0$ are chosen as M random positions satisfying a single observation of the nearest marker $m^{[n]}$.

4.2 Calculate Weights

As described in Sec. 2, the particle filter *calculate weights* step involves calculating an *importance factor* (*weight*) $w_t^{[m]} = p(z_t|x_t^{[m]})$ for each particle $x_t^{[m]}$ given observation z_t. As the problem model described in Sec. 3 involves only uniform probability distributions, these weights are simply defined as

$$w_t^{[m]} = \begin{cases} 1 & \text{if } x_t^{[m]} \in E_t \\ 0 & \text{otherwise} \end{cases}, \tag{4}$$

where E_t is the region E defined in (3), for observations of markers $m^{[n]}$ $(1 \leq m \leq M)$ at time t.

4.3 Resample

As described in Sec. 2, the particle filter *resample* step involves drawing M particles from $\overline{\mathcal{X}}_t$ to add to the particle set $\mathcal{X}_t$, with the probability of drawing the particle $x_t^{[m]}$ proportional to the corresponding weight $w_t^{[m]}$. As the particle weights defined in (4) take only binary values, it follows that any particle $x_t^{[m]} \notin E_t$ will never be drawn. Similarly, any particle $x_t^{[m]} \in E_t$ will be drawn with equal probability. Following this observation, the Gliders2013 implementation of the posterior $\mathcal{X}_t \sim bel(x_t)$ is defined as

$$\mathcal{X}_t = \overline{\mathcal{X}}_t \cap E_t \cup \mathcal{Y}_t, \quad (5)$$

where $\mathcal{Y}_t$ is a set of $|\overline{\mathcal{X}}_t - \mathcal{X}_t|$ points drawn at random from the uniform distribution E_t. This maintains a set of M particles between iterations, where each particle is a valid positional self-belief given all observations at time t[3].

5 Results

The performance of the presented Gliders2013 localisation system was evaluated against two metrics: the effect on self-localisation error, explicitly measurable by calculating the difference between server and self-belief player positions; and the effect on overall team performance, implicitly measured through the goal statistics of a statistically significant number of games.

Fig. 4 demonstrates the results for self-localisation error. Specifically, average self-localisation error (for all players) is plotted as a function of ball distance, for the previous agent2D [1] localisation system (blue) and introduced Gliders2013 particle filter system (red). Mean error across 10 games (60,000 cycles, or approximately 750,000 measurements) is presented as a solid line, with the corresponding standard deviation presented as a dashed line. As the x-axis represents the maximum distance from considered players to the ball, values approaching the right of the graph indicate the total mean, whereas values toward the left indicate the performance of players in critical positions (i.e. very close to the ball).

It can be seen from Fig. 4 that the Gliders2013 localisation system yields a mean self-localisation error reduction of 1.7 cm; an overall improvement of 38.2%. More significant is the improvement for players in close proximity to the ball, where localisation performance is especially critical. Concretely, self-localisation error decreases by 6.1 cm for players within 1 meter of the ball; an improvement of 201.3%.

Although these results appear significant as a percentage, a reduction of localisation error by centimeters may seem insignificant in the context of a full-sized soccer field. It is therefore critical that these results be demonstrated to correspond with a statistically significant improvement in game performance. To establish the extent of the overall team improvement brought upon by the increased localisation accuracy, multiple iterative experiments were carried out matching Gliders2012 up against teams not based on agent2D [1]. Among the

[3] Although this definition of the posterior $\mathcal{X}_t$ does not map exactly to the formal particle filter framework presented in Sec. 2, it is better suited to the scenario of uniform distributions of positional self-belief.

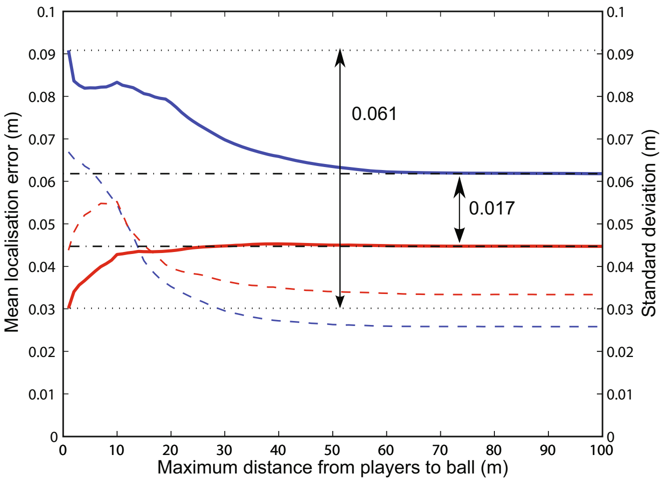


Fig. 4. Self-localisation error as a function of ball distance, for the previous agent2D [1] localisation system (blue), and introduced Gliders2013 particle filter system (red). Mean error is presented as a solid line, with standard deviation presented as a dashed line. An overall localisation improvement of 38.2% (or 1.7 cm) is evident, and a dramatic improvement of 201.3% (6.1 cm) for players within 1 meter of the ball (when accurate performance is most critical). Results presented are for 10 games (60,000 cycles, or $\sim$ 750,000 measurements) between Gliders2013 and WrightEagle.

latter class, WrightEagle [2] (the runner-up of RoboCup 2012) was selected. Approximately 750 games were conducted against this benchmark team in total, both for the baseline Gliders2012 team, and Gliders2012 incorporating the improved Gliders2013 localisation system (such that this system is the only variable between teams). These results are presented in Table 1.

Table 1. Game statistics between the Gliders2012 baseline team and WrightEagle [2] (the runner-up of RoboCup 2012), for both the old (agent2D [1]) and new (Gliders2013) localisation systems. As these results have been generated over approximately 750 games and the localisation system is the only variable between Gliders2012 teams, it can be confidently inferred that the 6.4% improvement in goal difference results directly from the 6.1 cm reduction in self-localisation error (for critically-positioned players; 1.7 cm overall).

	Old Localisation	**New Localisation**	**Improvement**
Goals Scored	0.36	0.41	14%
Goals Conceded	2.19	2.13	2.8%
Goal Difference	-1.83	-1.72	6.4%

6 Conclusion

Self-localisation is a complex yet critical task, both within the RoboCup 2D Simulation League, and the multi-billion dollar robotics industry at large. It allows an agent to infer its own position within some local environment, thus facilitating navigation around or interaction with obstacles, features and other agents. This paper has described a modified particle filter algorithm for non-Gaussian belief distributions, and demonstrated how approximating these distributions as uniform (given constrained noise models, such as those resulting from course-grain observations or artificial environments) facilitates reduction in self-localisation error. The proposed system yielded an improvement of 38.2% for agents within a RoboCup 2D Simulation League environment, which increased to 201.3% as agents approach performance-critical regions of the field. It was further demonstrated that this improvement results in a 14% increase in goals scored and 2.8% reduction in goals conceded for the Gliders2012 baseline team, evaluated across approximately 750 games against the WrightEagle team [2] (RoboCup 2012 runners-up).

Future research will focus on applying similar methodologies to ball and player localisation systems (including the incorporation and optimisation of team communication strategies), in an attempt to approach the performance leveraged by enabling full game state information for all player agents.

Acknowledgements. The authors would like to thank the University of Newcastle's NUbots RoboCup team in assisting with the preparation of this manuscript.

References

1. Akiyama, H.: Agent2D Base Code (2010), `http://www.rctools.sourceforge.jp`
2. Bai, A., Zhang, H., Lu, G., Jiang, M., Chen, X.: Gliders2012 wrighteagle 2d soccer simulation team description 2012. In: RoboCup 2012 Symposium and Competitions: Team Description Papers, Mexico City, Mexico (June 2012)
3. Budden, D., Fenn, S., Mendes, A., Chalup, S.: Evaluation of colour models for computer vision using cluster validation techniques. In: Chen, X., Stone, P., Sucar, L.E., van der Zant, T. (eds.) RoboCup 2012. LNCS (LNAI), vol. 7500, pp. 261–272. Springer, Heidelberg (2013)
4. Budden, D., Fenn, S., Walker, J., Mendes, A.: A novel approach to ball detection for humanoid robot soccer. In: Thielscher, M., Zhang, D. (eds.) AI 2012. LNCS, vol. 7691, pp. 827–838. Springer, Heidelberg (2012)
5. Butler, M., Prokopenko, M., Howard, T.: Flexible synchronisation within RoboCup environment: A comparative analysis. In: Stone, P., Balch, T., Kraetzschmar, G. (eds.) RoboCup 2000. LNCS (LNAI), vol. 2019, pp. 119–128. Springer, Heidelberg (2001)
6. Chen, M., Dorer, K., Foroughi, E., Heintz, F., Huang, Z., Kapetanakis, S., Kostiadis, K., Kummeneje, J., Murray, J., Noda, I., et al.: Robocup soccer server, Manual for Soccer Server Version 7 (2003)

7. Hill, F., Kelley, S.: Computer graphics: using openGL. Prentice Hall, Upper Saddle River (2001)
8. Noda, I., Stone, P.: The RoboCup Soccer Server and CMUnited Clients: Implemented Infrastructure for MAS Research. Autonomous Agents and Multi-Agent Systems 7(1-2), 101–120 (July-September)
9. Prokopenko, M., Obst, O., Wang, P., Held, J.: Gliders2012: Tactics with action-dependent evaluation functions (2012)
10. Prokopenko, M., Wang, P.: Evaluating team performance at the edge of chaos. In: Polani, D., Browning, B., Bonarini, A., Yoshida, K. (eds.) RoboCup 2003. LNCS (LNAI), vol. 3020, pp. 89–101. Springer, Heidelberg (2004)
11. Prokopenko, M., Wang, P.: Relating the entropy of joint beliefs to multi-agent coordination. In: Kaminka, G.A., Lima, P.U., Rojas, R. (eds.) RoboCup 2002. LNCS (LNAI), vol. 2752, pp. 367–374. Springer, Heidelberg (2003)
12. Russell, S., Norvig, P., Canny, J., Malik, J., Edwards, D.: Artificial intelligence: a modern approach, vol. 2. Prentice Hall, Englewood Cliffs (1995)
13. Thrun, S., Burgard, W., Fox, D.: Probabilistic robotics, vol. 1. MIT Press, Cambridge (2005)