An ICP Inspired Inverse Sensor Model with Unknown Data Association

Peter Anderson, Youssef Hunter and Bernhard Hengst

***Abstract*— This paper introduces an Iterative Closest Point (ICP) inspired inverse sensor model for robot localisation given multiple simultaneous observations of aliased landmarks. Combined with a Kalman filter, the sensor model offers a robust alternative to maximum likelihood data association, or a computationally inexpensive alternative to a particle filter. The technique can also be used as a means for re-localising a kidnapped robot, or a sensor resetting method for a particle filter. In the RoboCup Standard Platform League, this sensor model is able to localise the robot from a single observation in 42% of field positions where multiple landmarks are visible.**

I. INTRODUCTION

Many robot localisation problems require a robot to localise itself in a 2D environment given a map of the environment and observations of various landmarks. The application of sensor models and recursive localisation filters to these problems is well documented [1]. In their simplest form, localisation filters assume that observable landmarks can be uniquely identified, and that there is one sensor update per time step.

These assumptions are often violated. In many practical applications, there can be multiple simultaneous observations during a time step, and landmark correspondences between observations and the map are often difficult to determine with certainty. This is known as the data association problem.

To address both of these issues, we propose a novel inverse sensor model inspired by the Iterative Closest Point (ICP) matching algorithm. The technique is a robust alternative to the maximum likelihood association of each observation separately. The method calculates a single robot pose estimate conditioned on the prior pose, a map of the environment, and a set of landmark observations with unknown associations. The resulting pose can either be used as a meta-measurement in state space coordinates for a Kalman filter, as a means for re-localisation in kidnapped robot scenarios, as a sensor resetting method for particle filters, or as a stand-alone localisation approach.

The technique was developed for the 2012 RoboCup Standard Platform League (SPL) robot soccer competition, which takes place on a field with many aliased landmarks, as illustrated in Figure 1. The field itself, consisting of white field markings and yellow goals, contains two axes of symmetry, meaning that all of the key field-features such as goal posts, field-line corners and field-line T-junctions are found in at least four locations on the field. Also, from most positions on the field multiple field-features will be observed

P. Anderson, Y. Hunter and B. Hengst are with the School of Computer Science and Engineering, University of New South Wales, UNSW, Sydney, 2052 AUSTRALIA pjan520@unsw.edu.au

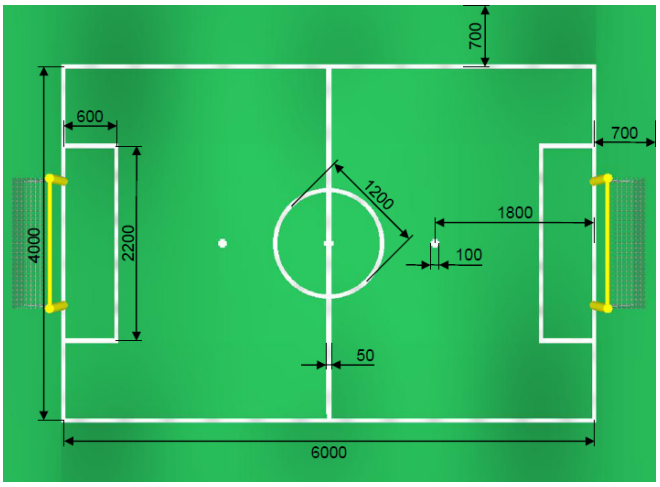

Fig. 1. Scale diagram of the RoboCup Standard Platform League field (measurements in millimetres). All key field-features are found in at least four separate locations on the field, and multiple field-features can be observed from most positions. Sourced from [2].

simultaneously. In our application, the sensor model was combined with a dual mode Kalman filter and used as an alternative to particle filter localisation, which can be limited by computational constraints set by the robot hardware. The robot used was an Aldebaran Nao v4 humanoid robot equipped with a 1.6 GHz Intel Atom processor and two 30 fps cameras.

Although the method was developed for RoboCup, it could be applied to any robot localisation problem where there are multiple simultanous observations of aliased landmarks. The remainder of this paper is organised as follows: Section II outlines related work, Section III describes the ICP-inspired sensor model, and Section IV presents experimental results.

II. BACKGROUND

To work effectively, localisation filters require well-specified sensor models to describe the process by which sensor measurements, such as landmark observations, are generated by the physical world. Inverse sensor models specify a distribution over possible robot poses conditioned on the observation and the map. In this paper we describe the ICP-inspired technique as an inverse sensor model, whilst acknowledging that the prior pose estimate is also required as an input.

When multiple landmark observations are made simultaneously, one course of action is to perform several consecutive sensor updates on the localisation filter. This approach suffers from a number of disadvantages. If the errors in the obser-

vations are correlated (for example, when the observations are all extracted from the one camera frame), the use of iterative filter updates results in a significant deterioration in localisation accuracy [3].

An alternative approach to dealing with multiple simultaneous observations is to perform a joint sensor update, as recommended by [3]. This approach requires that the observation covariance matrix accurately reflects the estimated level of error correlation between observations, and it does not solve the data association problem when landmarks are aliased.

A number of approaches to the data association problem have been formulated, including maximum likelihood landmark association, and Monte Carlo methods such as particle filters [1]. This paper outlines an alternative approach, in which observations are associated to the map in a hierarchical fashion, and an adaptation of the ICP algorithm is used to estimate the robot pose that minimises the squared positioning error over all observed features simultaneously.

By way of background, ICP is a widely used algorithm for geometric alignment of 2D or 3D point clouds when an initial estimate of the relative pose is known. Since the introduction of ICP by [4], many variants of the algorithm have been proposed, but in general the basic steps of the algorithm as outlined by [5] include:

1) **Selection** of some set of points in one or both point clouds.
2) **Matching** these points to samples in the other point cloud.
3) **Weighting** the corresponding pairs appropriately.
4) **Rejecting** certain pairs (optional).
5) **Assigning** an error metric based on the point pairs.
6) **Minimising** the error metric by finding a transformation between the two point clouds.
7) **Iterating** the above steps to further improve the alignment.

By representing both observed landmarks and the landmark map as 2D points, the ICP algorithm lends itself to the construction of an inverse sensor model, as described below.

III. INVERSE SENSOR MODEL

Given a set of observed non-unique landmarks in robot relative coordinates, a global landmark map, and a prior robot pose estimate, the ICP inspired sensor model calculates a robot pose estimate consistent with the observed landmarks.

In this application the robot pose is defined by its 2D planar coordinates x and y, along with its angular orientation θ, and the landmarks are SPL field-features outputted by a robot vision system that identifies field-lines (11), field-edges (4), field-line corners (8), field-line T junctions (6), goal posts (4), parallel lines (2) and the centre circle (1). The numbers in brackets indicate how many of each feature are found in the SPL field map.

If the application permits, the first step in the calculation of the robot pose observation should be to pragmatically pre-process landmark observations to restrict the number of possible map associations. In our application, this is achieved by examining the geometric relationship between field-features observed in the same frame. For example, if a goal post is observed next to a field-line T junction, the spatial relationship between the two features can be used to restrict the association of the T junction to one of two possible map features, rather than one of six.

After pre-processing, the key steps in the calculation of an updated pose observation include:

1) **Mapping** observed field-features from robot relative coordinates to field coordinates, using the prior iteration's robot pose estimate x_{k-1}, y_{k-1} and θ_{k-1}. In the first iteration this robot pose estimate is provided by the localisation filter, in subsequent iterations the computed pose is used.
2) **Hierarchically Matching** observed field-features with the map correspondence that maximises the data likelihood, starting with the most unique features in early iterations, and progressively including less unique features in subsequent iterations.
3) **Representing** matched field-features using 2D point pairs.
4) **Weighting** the corresponding pairs appropriately, given the confidence level of the observation.
5) **Assigning** a squared Euclidean distance error metric based on the distance between point pairs.
6) **Minimising** the error metric by finding a new robot pose estimate x_k, y_k and θ_k.
7) **Iterating** the above steps to further improve the robot pose and to allow any incorrectly associated features to be matched correctly.

There are many similarities between the algorithm described above and the ICP algorithm. Aspects of the algorithm are described in further detail below.

A. Hierarchical matching

During hierarchical matching (step 2), each observed field-feature in field coordinates is independently associated with the maximum likelihood map feature. However, only a subset of landmarks are included at each iteration, starting with the most unique landmark in the first iteration, and including the most unique remaining landmark at each subsequent iteration. This ensures that the association of the most heavily aliased landmarks (such as single field lines in our application) benefits from the improvement in the pose estimate provided by previous iterations, reducing the chance of incorrect association. In the RoboCup implementation, the following matching priority of field-features was used: Two goal posts, parallel lines, field-line corners, field-line T junctions, a single goal post, a centre circle, and lastly single field-lines.

It is important to note that this strategy is not just a greedy maximum likelihood approach, nor is it equivalent to iterative Kalman updates. Firstly, this is because (consistent with the ICP algorithm), all landmarks are re-associated at each iteration. This means that even if a landmark is incorrectly associated in an early iteration, there is still an opportunity for it to be re-associated correctly in subsequent iterations

as the pose estimate improves, assuming enough subsequent landmarks are correctly associated. If an incorrect association is corrected, there will be no negative impact on the accuracy of the localisation filter (which is not updated until the sensor model algorithm terminates). We have observed in testing that this correction happens quite frequently.

Secondly, if the individual landmark associations are internally inconsistent when the algorithm terminates, for example due to a false positive landmark identification, or an incorrect association, the error metric defined in step 6 will remain stubbornly high. In many cases this will allow false positives or incorrect associations to be detected. Greedy approaches and iterative Kalman updates do not have this ability.

In practise there are many camera frames in which no distinctive field-features can be seen, and the sensor model is forced to rely on a single-field line observation, as outlined in Table I. In this case, the sensor model relies heavily on the filter's prior robot pose to generate a new pose estimate, as indicated in Figure 7. Nevertheless, the hierarchical approach ensures that when a distinctive field-feature or a distinctive combination of features is seen, poor robot pose initialisation is unlikely to prevent the sensor model from converging to the correct robot pose, as demonstrated by the results in Section IV.

B. Representing and solving

Once observed field-features have been matched to the field map, matched field-features must be represented as 2D point pairs for the ICP based approach to be used (step 3). These points exist in the global field coordinate frame. In each point pair, one point represents the observed position of some field-feature, transformed from robot relative coordinates to field coordinates using the current robot pose estimate. The second point represents the true position of that field-feature on the map. In our application, compound features such as field-lines, T junctions and corners are represented with sets of point pairs.

In step 4, the weighting on each point pair is scaled according to the inverse distance between the robot and the observation. This puts less emphasis on the features that are further away and less precisely located than nearer features. As with ICP, it is also possible to weight point pairs according to the distance between the two points, which will tend to de-weight and reject false positive observations which do not converge to a true position on the map.

In steps 5 and 6, a new robot pose is estimated by rotating and translating the prior robot pose to approximately minimise the squared positioning error over all observed features simultaneously. Given a set of N source points $\{\vec{p}_1, \ldots, \vec{p}_N\}$ and N target points $\{\vec{q}_1, \ldots, \vec{q}_N\}$ with weights $\{w_1, \ldots, w_N\}$, the mean squared positioning error e is given as a function of rotation matrix $R(\delta\theta)$ and translation vector $\vec{t}$ by:

$$e(R, \vec{t}) = \frac{1}{N} \sum_{i=1}^{N} w_i ||\vec{q}_i - R\vec{p}_i - \vec{t}||^2 \tag{1}$$

To find an approximate minimum for this error function, the centroids $\bar{\vec{p}}$ and $\bar{\vec{q}}$ are deducted from both source and target point clouds (in order to separate rotation from translation). The rotation matrix R is approximated using a first order Taylor series approximation such that $sin(\delta\theta) \approx \delta\theta$ and $cos(\delta\theta) \approx 1$. Note that this approximation has little effect on the accuracy of the final pose since it is repeated at each iteration. Equating the first order partial derivatives of e with respect to t_x, t_y and $\delta\theta$ to zero gives the following matrix system for each point pair i:

$$A_i \begin{pmatrix} t_x \\ t_y \\ \delta\theta \end{pmatrix} = b_i \tag{2}$$

where

$$A_i = \begin{pmatrix} 2w_i & 0 & -2w_i p_{i,y} \\ 0 & 2w_i & 2w_i p_{i,x} \\ -2w_i p_{i,y} & 2w_i p_{i,x} & 2w_i p_{i,y}^2 + 2w_i p_{i,x}^2 \end{pmatrix} \tag{3}$$

$$b_i = \begin{pmatrix} 2w_i q_{i,x} - 2w_i p_{i,x} \\ 2w_i q_{i,y} - 2w_i p_{i,y} \\ -2w_i p_{i,y}(q_{i,x} - p_{i,x}) + 2w_i p_{i,x}(q_{i,y} - p_{i,y}) \end{pmatrix} \tag{4}$$

Since each point pair i contributes three equations, if $N > 1$, this system is overdetermined, and the least squares solution can be found using a singular value decomposition. The new estimate for the robot pose observation can then be calculated from the previous pose estimate as follows:

$$\begin{pmatrix} x_k \\ y_k \end{pmatrix} = T \begin{pmatrix} x_{k-1} \\ y_{k-1} \end{pmatrix} = R(\delta\theta)[\begin{pmatrix} x_{k-1} \\ y_{k-1} \end{pmatrix} - \bar{\vec{p}}] + \begin{pmatrix} t_x \\ t_y \end{pmatrix} + \bar{\vec{q}} \tag{5}$$

$$\begin{aligned} \theta_k = \theta_{k-1} &+ atan2(y_k - \hat{p}_{i,y}, x_k - \hat{p}_{i,x}) \\ &- atan2(y_{k-1} - p_{i,y}, x_{k-1} - p_{i,x}) \end{aligned} \tag{6}$$

where

$$\hat{\vec{p}}_i = T(\vec{p}_i) \tag{7}$$

and

$$i \in \{1, \ldots, N\} \tag{8}$$

Step 7, iteration, is continued until all observed features have been included in the process, and the squared positioning error $e(R, \vec{t})$ falls below a threshold or fails to improve, or a maximum number of iterations is reached (which is 12 in our case). At this point, if the mean squared error $e(R, \vec{t})$ is sufficiently small, the robot pose observation estimate x, y and θ can be provided to the localisation filter. If the position error is large, it may indicate that the observed combination of field-features is inconsistent, for example due to a false positive feature identification. It can also indicate that the algorithm became stuck in a local minima due to a poor initial pose, potentially indicating that the robot is lost.

C. Line observations

The sensor model as described treats all field-features as points. It does not deal well with observations of line features which leave the robot's location unconstrained in one direction. The use of a point representation will artificially constrain the robot's pose from sliding along line features to improve the alignment of other observations.

Since the field-line features on the SPL field are always aligned to the x or y axis of the field coordinate system, this problem can easily be improved by minimising the error function $e(R, \vec{t})$ with respect to just two of the three components of the change in the robot's pose. If the field-line parallels the x-axis, the correct minimisation is with respect to θ and t_x, and Equations 3 and 4 are replaced with Equations 9 and 10.

$$A_i = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2w_i & 2w_i p_{i,x} \\ 0 & 2w_i p_{i,x} & 2w_i p_{i,x}^2 \end{pmatrix} \begin{pmatrix} t_x \\ t_y \\ \delta\theta \end{pmatrix} \tag{9}$$

$$b_i = \begin{pmatrix} 0 \\ 2w_i q_{i,y} - 2w_i p_{i,y} \\ 2w_i p_{i,x}(q_{i,y} - p_{i,y}) \end{pmatrix} \tag{10}$$

If the field-line parallels the y-axis, the minimisation is with respect to θ and t_y, and Equations 3 and 4 are replaced with Equations 11 and 12.

$$A_i = \begin{pmatrix} 2w_i & 0 & -2w_i p_{i,y} \\ 0 & 0 & 0 \\ -2w_i p_{i,y} & 0 & 2w_i p_{i,y}^2 \end{pmatrix} \tag{11}$$

$$b_i = \begin{pmatrix} 2w_i q_{i,x} - 2w_i p_{i,x} \\ 0 \\ -2w_i p_{i,y}(q_{i,x} - p_{i,x}) \end{pmatrix} \tag{12}$$

IV. RESULTS

A. Examples

Figures 2 and 3 illustrate the performance of the sensor model when multiple field-features are observed simultaneously by a robot standing near the centre of the SPL field. Since bipedal robots frequently push and collide while playing soccer, causing large unanticipated heading changes, robustness of the sensor model to poor robot pose initialisation is an important feature in this application. In this example, even if the heading of the initial robot pose is extremely poor (plus or minus 90 degrees from the true heading), the sensor model still converges on the correct robot pose from approximately 50% of all possible field locations.

Since it is not possible to distinguish one end of the field from the other using only field-features, this result indicates that the sensor model can localise the robot from a single observation in this field location. If the sensor model is initialised with less heading error, the area of convergence to the correct pose increases to cover the entire field, as illustrated in Figure 3.

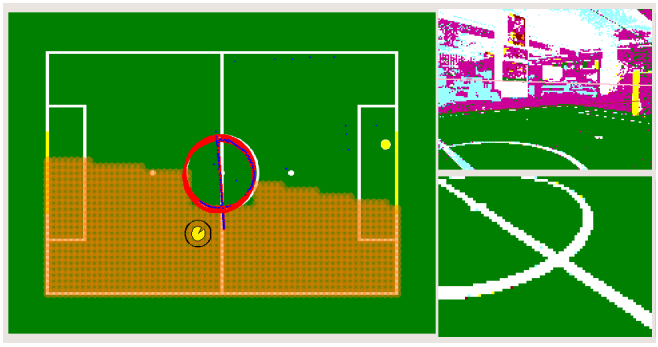

Fig. 2. Left: Sensor model area of convergence. Orange shading is used to indicate the set of initial robot poses that converge to the correct robot pose observation, shown as a yellow circle with direction cut-out, provided heading initialised within 90 degrees of the robot's true heading. Right: Colour classified source images from the robot's top and bottom cameras. Observations consist of a centre circle and a field line in the bottom camera, and a centre circle and goal post in the top camera.

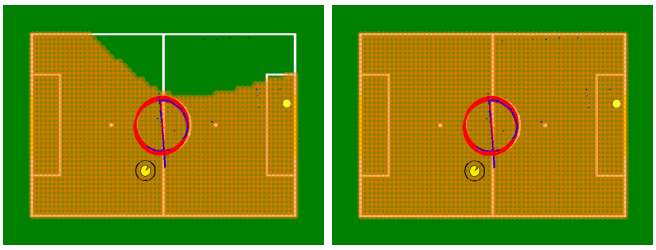

Fig. 3. Sensor model area of convergence after reducing the initial heading error to less than 60 degrees (left) and less than 45 degrees (right). These results indicate that when several field-features are observed, the sensor model is able to localise the robot from a single observation even if the robot pose initialisation is quite poor.

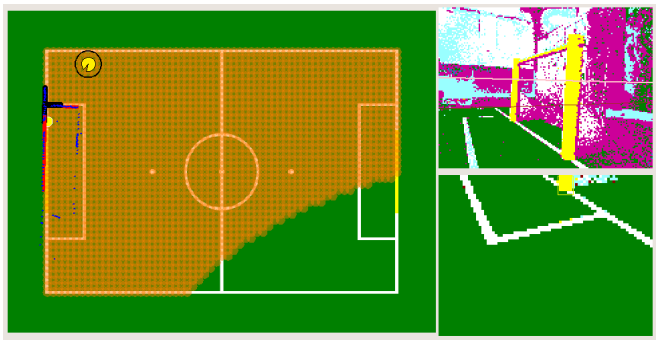

Fig. 4. Sensor model area of convergence given simultaneous observations of a field-line T junction and a goal post. The area of converge to the correct robot pose (shaded in orange) assumes the initial robot pose heading error is less than 45 degrees.

Figures 4, 5, 6 and 7 illustrate further examples of field-feature observations and the corresponding areas of convergence given 45 degrees initial heading error. In all results the sensor model observation is considered to be correct if it is within 10 cm and 10 degrees of the true robot pose, which in this application is sufficient to field a strong soccer team. In any case, the final positioning error of the sensor model is primarily determined by the accuracy of the field-feature observations, which are a function of the distance to the observation, the resolution of the camera image, and

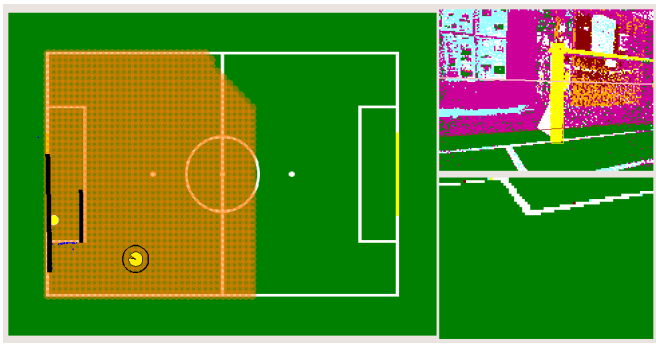

Fig. 5. Sensor model area of convergence given simultaneous observations of parallel field-lines and a goal post, assuming initial heading error of less than 45 degrees.

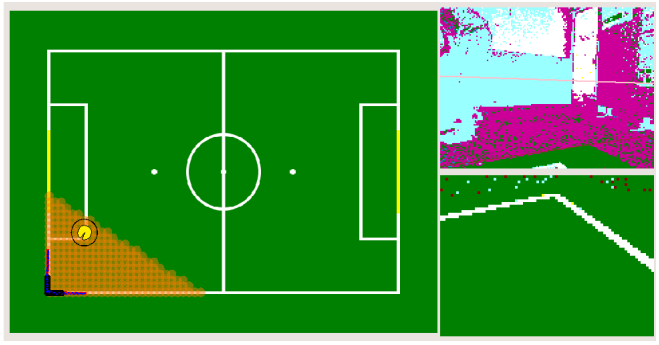

Fig. 6. Sensor model area of convergence given a single observation of a field-line corner, assuming initial heading error of less than 45 degrees.

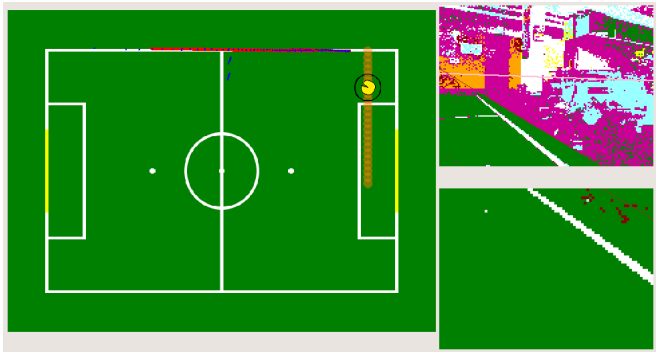

Fig. 7. Sensor model area of convergence given a single observation of a field-line, assuming initial heading error of less than 45 degrees. In this case the sensor model can only converge to the true robot pose if the component of the robot pose in the direction of the field-line is already correct.

errors in the robot's kinematic chain.

The area of convergence is largest when multiple field-features are observed or if the field-features seen are relatively unique. As shown in Figure 7, when only a single field-line is observed, the sensor model can only converge to the correct robot pose if the component of robot pose in the direction of the line is already correct.

B. Randomly Sampled Field Positions

To evaluate the effectiveness of the sensor model more formally, this section quantifies the sensor model's area of convergence to the correct robot pose, and its positioning error, at a range of different field locations. To represent the diversity of robot poses achieved during an SPL match in this test, 20 locations on the SPL field were chosen using uniform random sampling from the robot's x, y, θ field configuration space. The resulting test locations are shown in Figure 8.

TABLE I

INCIDENCE OF VARIOUS FIELD-FEATURE OBSERVATIONS IN 20 RANDOMLY SELECTED SPL FIELD POSITIONS.

Field-feature	Incidence
Single goal post	15%
Two goal posts	10%
Centre circle	20%
Field-line corner	5%
Field-line T junction	5%
Parallel field-lines	10%
Field-lines	95%

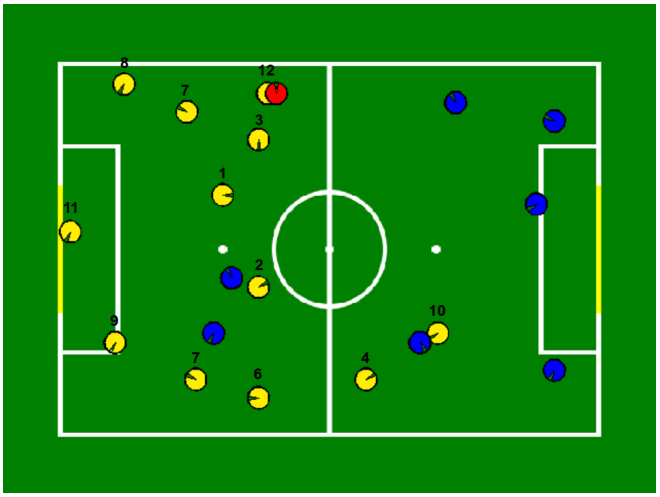

Fig. 8. Randomly determined robot poses used for sensor model evaluation. In positions marked with yellow, multiple field-features were visible. From blue positions, only a single field-line can be observed, and from the red position no field-features were visible. Position numbers correspond to Figure 9.

This data confirms that the general assumption of one sensor update per time step is frequently violated in RoboCup. In this data set of 20 randomly selected field positions, 5% of field positions yielded no useful observations (shown in red), 35% of positions yielded observations of a single field-line (shown in blue), and in 60% of positions multiple field-features could be observed (shown in yellow). The incidence of various types of field-feature observation in this dataset is presented in Table I. Note that the frequency of each observation owes more to the capabilities of the vision system than any inherent feature of the SPL field.

After discarding positions where only one or zero field-features could be observed, the performance of the sensor model in the remaining 12 positions with multiple field-feature observations was analysed. The area of convergence to the correct robot pose, given initial pose heading errors of plus and minus 15 degrees, 30 degrees, 45 degrees and 90 degrees respectively was used to evaluate the sensor model.

As previously discussed, since field-features cannot resolve one end of the field from the other, an area of convergence close to 50% with 90 degrees heading error indicates that the sensor model can localise the robot from a single observation. As indicated in Figure 9, this standard was achieved in 5 out of 12 (42%) of randomly determined field positions with multiple visible field-features, and 5 out of 20 (25%) of all field positions. The most informative field positions (positions 1 - 3) all contained observations of the centre circle.

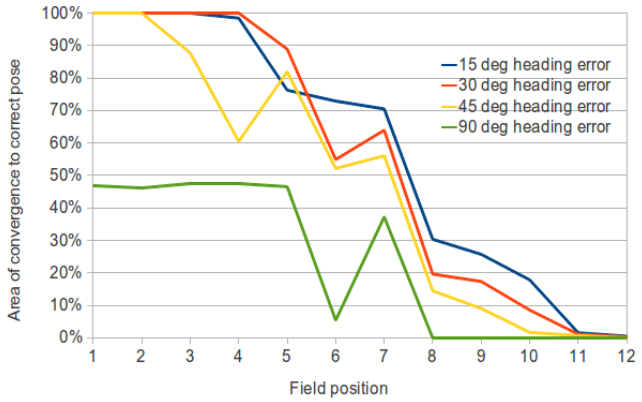

Fig. 9. Area of convergence to the correct robot pose over 12 randomly determined field positions with multiple visible field-features.

The mean positioning error of the sensor model across all trials in all 12 positions was 50 mm, and the mean execution time for the sensor model algorithm on the Nao robot was 3.2ms.

C. Kidnap Tests and RoboCup Performance

In the final test, the robustness of the sensor model was evaluated by combining it with a dual-mode Kalman filter and executing repeated kidnap tests. In this test, the Nao robot was programmed to walk to the kick-off position on the SPL field. It was then repeatedly kidnapped to various field locations and expected to return to the kick-off position. The robot was deemed to have arrived when it stopped walking and the time taken to localise and return to the kick-off position was recorded.

The four kidnap positions that were chosen are illustrated in Figure 10. Two trials were undertaken at each position. Position 1 is the kick-off position. Table II records the robot's final positioning error and the time taken in each trial.

As indicated by Table II, in every trial the robot was able to correctly return to the kick-off position. The mean final positioning error over the the 8 trials was 45 mm. A short uncut video illustrating a similar real-world experiment has been included as a supplement to this paper. It should be noted that the performance of the sensor model at the RoboCup competition, with spectators around the field and robots on the field, is similar to the results demonstrated in the video and in this paper. The sensor model is relatively robust to incorrect associations, as outlined in Section III, and the localisation system uses the sensor model error metric to reject camera frames containing false positive observations.

TABLE II

KIDNAP PERFORMANCE OF THE SENSOR MODEL.

Kidnap Position	Time Taken (s)	Positioning Error (mm)
2	12.9, 13.2	53, 25
3	32.1, 30.4	22, 95
4	16.9, 17.0	7, 28
5	16.6, 15.9	88, 42

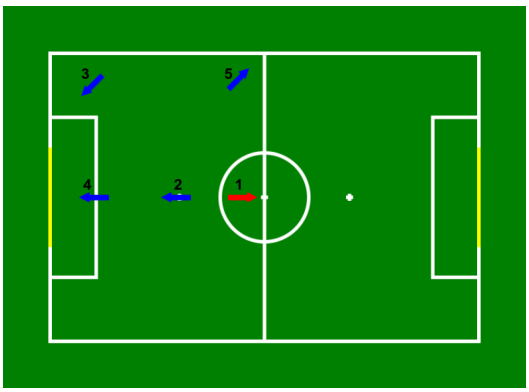

Fig. 10. Illustration of the positions used in the kidnap experiment. In each case the robot was required to return to the kick-off position labelled as position 1.

V. CONCLUSION

This paper has presented a novel ICP inspired sensor model applicable to a range of problems involving multiple simultaneous observations of aliased landmarks. The approach performs strongly in the RoboCup environment which is characterised by false positive observations and poor pose initialisation.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support of the School of Computer Science and Engineering at UNSW, past and present members of the rUNSWift RoboCup SPL team, and associated staff and students in the school's robotic laboratory.

REFERENCES

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[2] RoboCup SPL Technical Committee, "Robocup standard platform league (nao) rule book," Website: http://www.tzi.de/spl/pub/Website/Downloads/Rules2012.pdf, 2012.

[3] S. Tasse, M. Hofmann, and O. Urbann, "On Sensor Model Design Choices for Humanoid Robot Localization," in *RoboCup International Symposium 2012*, Mexico, June 2012.

[4] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2724–2729, 1991.

[5] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *3rd International Conference on 3D Digital Imaging and Modeling (3DIM 2001), 28 May - 1 June 2001, Quebec City, Canada*. IEEE Computer Society, 2001, pp. 145–152.