A Recursive Bayesian Filter for Landmark-Based Localisation of a Wheelchair Robot

Theodoros Theodoridis, Huosheng Hu, Klaus McDonald-Maier, and Dongbin Gu
School of Computer Science and Electronic Engineering
University of Essex, Wivenhoe Park, Colchester, CO4 3SQ, UK
{ttheod, hhu, kdm, dgu}@essex.ac.uk

***Abstract*—An odometry model, represented by a set of nodes (waypoints), is considered to be the infrastructure of any probabilistic-based localisation method. Gaussian and nonparametric filters utilise an odometry model to localise robots, while predictions are made by the filters to actively correct the robot's location and coordination. In this work, we present a recursive Bayesian filter for landmark recognition, which is used to verify the pose of a robotic wheelchair at a certain node location. The Bayesian rule in the proposed method does not incorporate a control action to rectify the robot's pose (passive localisation). The filter approximates the robot's pose based on a feature extraction sensor model. Features are extracted from local environmental regions (landmarks), and each landmark is assigned with a distinct posterior probability (signature), at each node location. A node is verified by the robot when the covariance between the posterior and prior probability falls bellow a threshold. We tested the proposed method in an indoor environment where accurate localisation results have been obtained. The experimentation demonstrated the robustness of the filter to work for passive localisation.**

I. Introduction

Gaussian and nonparametric filters have been used widely for robot localisation, for predicting the spatial location given a sensory observation. In its simplest form, localisation is primarily conducted via an odometry model. Odometry models utilise translational and rotational displacements so as to localise a robot within an array of predefined waypoints or nodes. The purpose of probabilistic filters is to aid the odometry model to make predictions by rectifying the robot's pose. Consequently, nodes can be reached by the robot more accurately while refining the drift error.

In this paper we present a recursive Bayesian filter for landmark recognition, which aims in assigning a distinct signature at each node location. The localisation method we propose is passive, which means that there is no control action suggested by the filter for pose rectification. However, the recursive update verifies the node location by updating the robot's pose $< x,\ y,\ \theta >$ with the current node location $< x_n,\ y_n >$. This happens when the signature of a landmark is being recognised as a feature $f \in F$. The following property describes this notion: iff $f_n \equiv f_{n-1} \in F = \{f_1,\ f_2,\ \ldots,\ f_N\} :< x,\ y > \leftarrow < x_n,\ y_n >$. More analytically, if the current feature f_n is equal to a model feature f_{n-1}, from a set of features F assigned for each node, then the odometry coordinates of the robot's pose are updated with the coordinates of the node.

II. Related Work

In Bayesian localisation, [1] showed a semantic representation for robot localisation. With a single camera and odometry information, spatial relations among object are created to conform a distance map. The map's metric data are estimated using visual pattern recognition. Further, localisation is performed by a Bayesian model using the topological-semantic distance map, which overall has demonstrated accurate localisation accuracy. A hierarchical vision-based Bayesian localisation model was introduced by [2], for the estimation of a set of colours. The model outperforms in two levels, where in the first, lighting conditions are estimated using a switching Kalman filter. In the second, a Bayesian model learns Gaussian priors from the robot's environment. In addition, a Rao-Blackwellised particle filter is employed to maintain a joint posterior of the robot's position.

Passive localisation does not incorporate control actions, suggested by a perception model to correct the robot's position. Previous research on passive robot localisation included the work of [3], who presented a passive localisation method applied on autonomous underwater vehicles embedded with periodic sonar transmission. The key idea is based on the delay time of the sonar beam using a single linear array. Effective estimation of the location has been shown with a simple array configuration. There are, however, more accurate localisation methods that are based on active localisation. In the work of [4], an active Markov localisation method is proposed where partial or full control applies actions to a robot. The method reduces the complexity of the localisation task by letting the robot to actively interact with its environment, even if the environment possesses a handful amount of features. The guiding principle of active localisation is to control a robot by minimising future expected uncertainty, using an entropy metric of future belief distributions. Similarly, [5] presented a fine-grained grid to approximate densities, which is able to globally localise the robot, and to recover from failures. Also, their method can robustly approximate models of the environment and noisy sensors. Eventually, a filtering technique allows the estimation of the robot's pose in overpopulated environments.

Recursive Bayesian filters motivated topics of robot prediction and tracking. In [6], a coordinated control technique based on recursive Bayesian filtering, incorporated with a unified sensor model and a unified objective function, has been used for tracking of multiple heterogeneous vehicles. A Bayesian search-and-tracking method was applied in a marine search-and-rescue scenario where heterogeneous vehicles performed multi-target tracking. A recursive Bayesian estimation on a bearings-only application presented by [7], showed how the incorporation of terrain information can improve estimation performance in target tracking using angle-only. They have solved the Bayesian estimation problem using a marginalised particle filter. An alternative work, which aims on the classification, presented by [8]. Similar to our model that is used mainly for the recognition of landmark features, a recursive Bayesian linear regression method was used for adaptive classification. A trajectory of non-stationary environments is traced to perform classification of benchmark face datasets. The methodology has shown to outperform support and relevance vector machines (SVM, RVM), and it is analogous to Bayesian SVMs.

In the wheelchair localisation literature, [9] is referred to a landmark tree model for self-localisation of an autonomous wheelchair. The method utilises an image retrieval technique and the Bayes rule to localise the wheelchair. In addition, a path planning algorithm is introduced, which exploits a tree-like structure to locate landmarks and destination locations. Landmarks are recognised through an image by extracting the shape and structure, and localisation is conducted by traversing within the tree nodes to elicit an optimal path. A probabilistic odometry (motion) model was introduced by [10], for an autonomous wheelchair. Their method constructs a set of frequency tables of the wheelchair's pose, stored in bins. A particle filter advises these tables to make predictions for localisation. The method is said to be efficient and easy to intergrade in any particle filter algorithm for real time localisation of robots and wheelchairs.

The rest of the paper is organised as follows. Section III presents three principal models. A recursive Bayesian model for landmark recognition, a feature extraction model, and an odometry model. The collaboration of all the three models constitute the paper's contribution. Section IV demonstrates the localisation results. The experiments carried out in the Essex robotics arena, which is an indoor environment. Ultimately, conclusions and future works are given in Section V.

III. Recursive Bayesian Landmark Model

Localisation is defined as the estimation of a robot's pose $< x^1,\ y,\ \theta >$ from a sensory observation. Position and orientation can be approximated given the map of the environment, and a set of sensors that access the environment [5]. In our study, instead of a map, localisation is conducted by providing a set of $< x_n,\ y_n >$ nodes n, through which the robot can access several locations of the environment. A node-location is verified by a recursive Bayesian landmark model, estimated using a Laser Range Finder (LRF) – the robot's sensory perception that "sees" the environment.

[1]For the reader's convenience, the observation state x_z has no relation with the odometry coordinate x, found in the robot's pose $< x,\ y,\ \theta >$.

A. Recursive Bayesian Model

The recursive Bayesian filter uses two assumptions for the recursive estimation [11]: (i) The states follow a first order Markov process $p(x_n|x_{n-1})$, (ii) The states are independent from the observations. In our model, each state is a signature observation z acquired by a range vector $\vec{r}$ via a LRF; thus, the states are defined as x_z. Consequently, the recursive Bayesian updating rule is described by the posterior $p(x_z|z_n)$ of Eq. 1. Over a number of recursive iterations, shown in Fig. 1(c), the posterior is updated by the product of the prior (current landmark observation) and the conditional model (initial landmark observations). The closer the prior distribution to the conditional model is, the higher the posterior becomes.

$$p(x_z|z_n) = \frac{p(x_z|z_{n-1})p(z_{1:n-1}|x_z)}{p(z_n|z_{1:n-1})} \quad (1)$$

· *Prior* The prior $p(x_z|z_{n-1})$ is the test instance, which is updated recursively with the posterior $p(x_z|z_n)$ (see Eq. 2). Fig. 1(a) depicts the prior model distribution of a landmark.

$$p(x_z|z_{n-1}) \leftarrow p(x_z|z_n) \quad (2)$$

· *Conditional* The conditional $p(z_{1:n-1}|x_z)$ defines the knowledge of the model. It is a set of measurement (range) distributions that reflect to a landmark's signature as Fig. 1(b) shows. These distributions are the training instances of the Bayesian model described by Eq. 3.

$$p(z_{1:n-1}|x_z) = p(z_1,\ z_2,\ \ldots,\ z_{1:n-1}|x_z) \quad (3)$$

· *Evidence* The evidence $p(z_n|z_{1:n-1})$ defines a normalisation factor of the measured data distributions:

$$p(z_n|z_{1:n-1}) = \int p(x_z)p(z_{1:n-1}|x_z)dx_z \quad (4)$$

where $p(x_z) = p(x_z|z_{n-1})$ as x_z and z_{n-1} are independent.

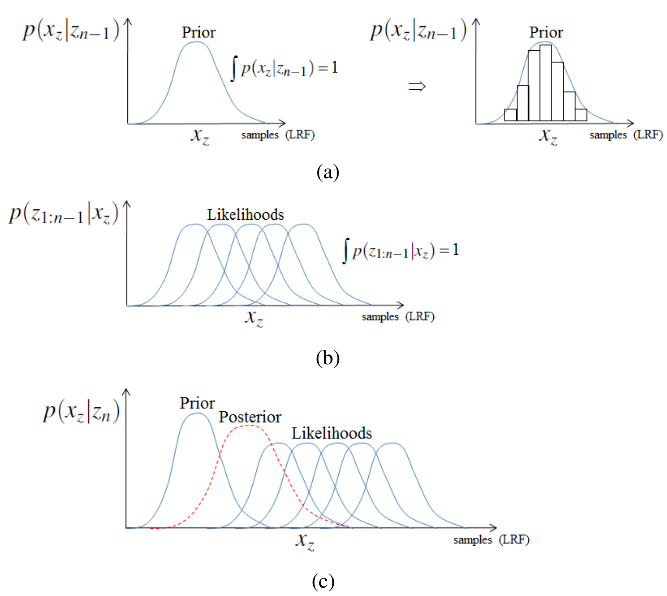

Fig. 1. (a) Prior distribution (current landmark observation), (b) Conditional distributions (initial landmark observation model), (c) Posterior distribution.

TABLE I
MULTI-NODE ODOMETRY MODEL.

turnTo($x_n,\ y_n$):	$\vartheta_n = \arctan 2(y,\ x) = 2 \cdot \arctan\left(\frac{y}{\sqrt{x^2+y^2}+x}\right)$ $\vartheta_n' = \frac{\vartheta_n \cdot 180}{\pi}$ $\delta_\theta \leftarrow (\vartheta_n' - \gamma_\theta) \leq \theta \leq (\vartheta_n' - \gamma_\theta)$	The x, y, and θ parameters are the wheelchair's odometry coordinates, and ϑ_n is the turn-to angle pointing to next node coordinates x_n, y_n. ϑ_n is then converted from radians to degrees. The rotational transition δ_θ is an angular control interval designated to stop the robot at ϑ_n, which points towards $< x_n,\ y_n >$, using the γ_θ threshold.
moveTo($x_n,\ y_n$):	$d_t = \sqrt{x^2+y^2}$ $d_c = \sqrt{(x-x_n)^2+(y-y_n)^2}$ $\delta_d \leftarrow (d_t - \gamma_d) \leq (d_t - d_c) \leq (d_t + \gamma_d)$	The parameters d_t and d_c are the total (t) and the current (c) Euclidean distances. The difference between the two distances aids to estimate the destination node $< x_n,\ y_n >$ as the robot moves. The translational transition δ_d is a linear control interval designated to stop the robot at $< x_n,\ y_n >$ using the γ_d threshold.
update( ):	$x = x_n,\ y = y_n$ $n = n_{n+1}$	In the update step, the odometry coordinates of the robot's pose $< x,\ y >$ are updated with current node coordinates $< x_n,\ y_n >$. Also, the node counter n is updated pointing to the next coordinate pair $< x_{n+1},\ y_{n+1} >$.

B. Feature Extraction Model

The recognition of a physical landmark, a feature, is being described by a collection of primitives. These primitives represent the identity of each distinct landmark using a set of vector triplets [12]. The set is a vector of three elements $[\vec{r}_t,\ \mathrm{x}_t,\ \gamma_t]^T$, where $\vec{r}_t$ is the LRF vector, x_t is the robot's pose, and γ_t is a recursive covariance signature, all updated at time t. The feature triplet is given analytically by Eq. 5.

$$\begin{bmatrix} \vec{r}_t \\ \mathrm{x}_t \\ \gamma_t \end{bmatrix} = \begin{bmatrix} z^{1:m} \\ < x, y, \theta > \\ Cov_f[p(x_z|z_{n-1}),\ p(z_{1:n-1}|x_z)] \end{bmatrix} \tag{5}$$

where $z^{1:m}$ is a measurement vector with m elements acquired by the LRF ($= 180$ samples). The pose $< x,\ y,\ \theta >$ defines the robot's location and coordination relative to a global coordinate frame. The covariance $Cov_{f_{1:n}}$, for feature f, extracts a signature between the posterior $p(x_z|z_{n-1})$ and the prior distribution $p(z_{1:n-1}|x_z)$. The covariance signature, given by Eq. 6, is being used as a threshold value γ for each landmark to verify a feature. When the wheelchair robot approaches a landmark, the covariance error between the prediction of the posterior distribution and prior observation diminishes. A covariance threshold γ then verifies whether a landmark f_n at a node $< x_n,\ y_n >$ is being recognised.

$$Cov_f[p(x_z|z_{n-1}),\ p(z_{1:n-1}|x_z)] = \sum_{i=1}^{S} \frac{[p(x_z|z_{n-1})_i - \overline{p(x_z|z_{n-1})}] \cdot [p(z_{1:n-1}|x_z)_i - \overline{p(z_{1:n-1}|x_z)}]}{S} \tag{6}$$

Heretofore, we have defined three parameters to characterise a landmark feature. Eq. 7 shows a more general form of the feature set that describes each distinct landmark.

$$f(z_t) = f_t^1,\ f_t^2,\ \ldots,\ f_t^n = \{ \begin{bmatrix} \vec{r}_t^1 \\ \mathrm{x}_t^1 \\ \gamma_t^1 \end{bmatrix}, \begin{bmatrix} \vec{r}_t^2 \\ \mathrm{x}_t^2 \\ \gamma_t^2 \end{bmatrix}, \ \ldots, \begin{bmatrix} \vec{r}_t^n \\ \mathrm{x}_t^n \\ \gamma_t^n \end{bmatrix} \} \tag{7}$$

C. Odometry Model

The odometry model takes into account the translational and rotational error during locomotion [10]. Table I summarises the node-based localisation using an odometry model. The whole localisation process is induced in three steps as shown in the table. The first step is the turnTo($x_n,\ y_n$), where the robot turns to the desired $< x_n,\ y_n >$ node location. Next is the moveTo($x_n,\ y_n$) step, where the robot moves towards the destination node $< x_n,\ y_n >$. In the update() step, the robot updates its current pose to the node's pose after reaching it.

Fig. 2 illustrates the multi-node odometry model. Several node locations can be reached by using translation and rotational kinematic transitions, δ_d and δ_θ respectively. In the odometry model we introduce, the parameter ρ represents a radius expanding from each node centre. Its purpose is to wide the spatial location of the nodes, and minimise the contingency of missing a node during localisation. In the figure, the feature vector triples are also shown at each node location. Here, nodes and feature vectors have the same index; it is therefore easy to validate a location using not only the pose, but also a set of features as described by the feature extraction model.

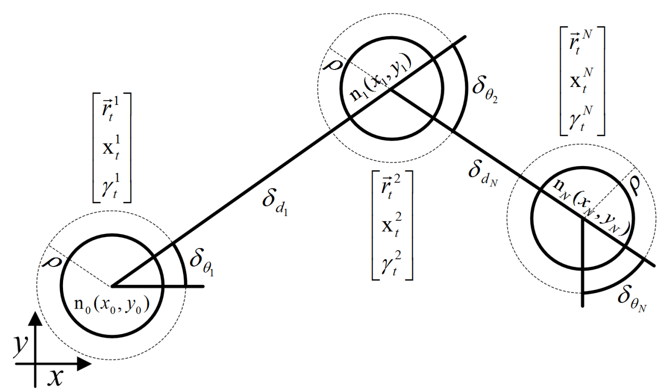

Fig. 2. Odometry model illustrating the linear δ_d and rotational δ_θ transitions, as well as the feature vectors acquired at each node n_i.

D. Passive Localisation Algorithm

In passive localisation the estimation of a location is determined by a series of sensory observations, while control commands with respect to linear or rotational displacements are not incorporated [4]. Our passive localisation model is referred as a position estimation technique rather than a position control one. Additionally, we have adopted the Markov assumption to carry out experiments on static environments, where the environment only affects immediately the sensor readings. Algo. 1 demonstrates the passive localisation process of our method, incorporating a multi-node odometry model, and a feature extraction model as shown earlier.

Algorithm 1 Passive localisation algorithm.

1: **for all** nodes $< x_n,\ y_n >$ **do**
2: $\quad \delta_{\theta n} = \text{turnTo}(x_n,\ y_n)$
3: $\quad$ **while** $\delta_{\theta n} \neq \theta$ **do**
4: $\quad\quad \text{turn}(\cdot)$
5: $\quad$ **end while**
6: $\quad \delta_{dn} = \text{moveTo}(x_n,\ y_n)$
7: $\quad$ **while** $\delta_{dn} \neq < x,\ y >$ **do**
8: $\quad\quad \text{move}(\cdot)$
9: $\quad\quad z^{1:m} = \vec{r}$
10: $\quad\quad \text{x} = < x,\ y,\ \theta >$
11: $\quad\quad \gamma = Cov\left[p(x_z|z_{n-1}),\ p(z_{1:n-1}|x_z)\right]$
12: $\quad\quad$ **if** $\delta_{dn} \cong < \text{x},\ \rho >$ and $\gamma < \gamma_n$ **then**
13: $\quad\quad\quad \text{stop}(\cdot)$
14: $\quad\quad\quad < x,\ y >= \delta_{dn}$
15: $\quad\quad$ **end if**
16: $\quad$ **end while**
17: **end for**

The whole localisation procedure is carried out using an odometry model, for transiting the wheelchair robot from node-to-node location. In Algo. 1, line 1 iterates through an array of nodes indicated as pairs of coordinates or poses $< x_n,\ y_n >$. Lines 2-5 perform a rotation pointing to $< x_n,\ y_n >$. $\delta_{\theta n}$, acquired from the turnTo() function (see Table I), holds the transition angle, and θ is the current robot's angle. After a successful rotation to $< x_n,\ y_n >$, lines 7-16 perform a translation towards $< x_n,\ y_n >$. δ_{dn}, acquired from the moveTo() function holds the $< x_n,\ y_n >$ node, whereas $< x,\ y >$ is the current robot's pose. The while statement in line 7 runs until the robot reaches the node location, which is the condition to be met. Lines 9-11 acquire the feature extraction triplet. The LRF vector $\vec{r}$, line 9, assigns in the observation z an array of samples m. Next, x is updated with the pose, and γ with the covariance threshold of the recursive Bayesian model. Lines 12-15 validate whether the next node δ_{dn} approaches the x pose and radius ρ, as well as whether the current covariance γ approaches the covariance of the next node γ_n. If this condition is met, the robot's pose $< x,\ y >$ is updated with the next node coordinate pair found in δ_{dn}, and the translational while statement breaks. Thereafter, the algorithm is repeated for the node location $< x_{n+1},\ y_{n+1} >$.

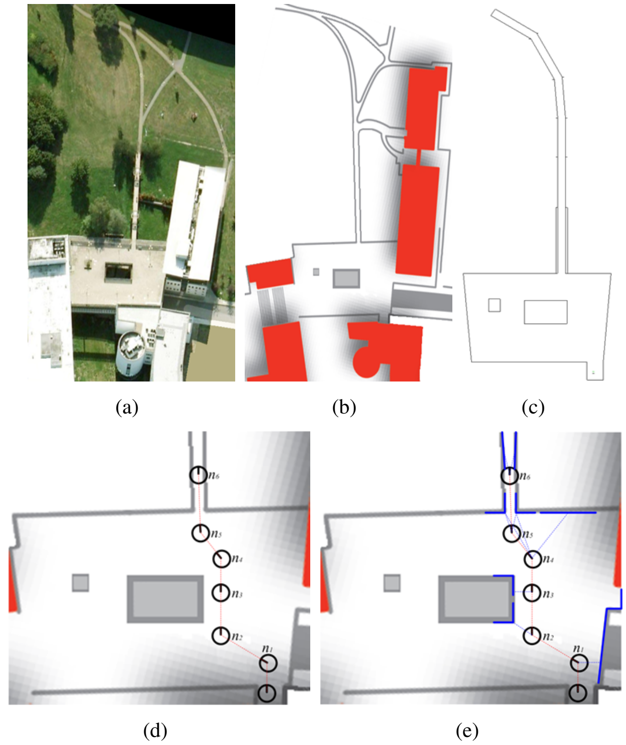

Fig. 3. The primary experimental environment. (a) Outdoor illustration, (b) Outdoor map, (c) Outdoor simulation. Indoor counterpart indicating the node and landmark locations. (d) Node locations, (e) Landmark locations.

IV. Experimental Results

The Essex robotic arena was the main experimental hall where the experimental procedure took place. The primary environmental setup illustrated in Fig. 3 was used for the conduction of the experiments. We have built an indoor counterpart of the outdoor environment shown in Figs. 3(d) and 3(e). Instead of a wheelchair, the experimental work was carried out using an Activmedia Pioneer robot for testing purposes. Later experiments will be based on using the actual Essex wheelchair.

A. Odometry Model Performance

We performed five experimental runs to test the recursive Bayesian landmark model in the counterpart (indoor) environment shown in Figs. 3(d) and 3(e). Table II contains the error performance acquired from each run and node location n. An error is estimated as the difference between an ideal node, acquired earlier, and the robot's current pose when reaching this node. The errors are subject of the recursive Bayesian landmark model, and the odometry model incorporated with the radius ρ. The average values in the table indicate that the maximum node error for a 5 run performance within 6 nodes was less than 260mm. The radius ρ was adjusted initially at 250mm meaning that each node can tolerate a 0.5m diameter to verify a successful location.

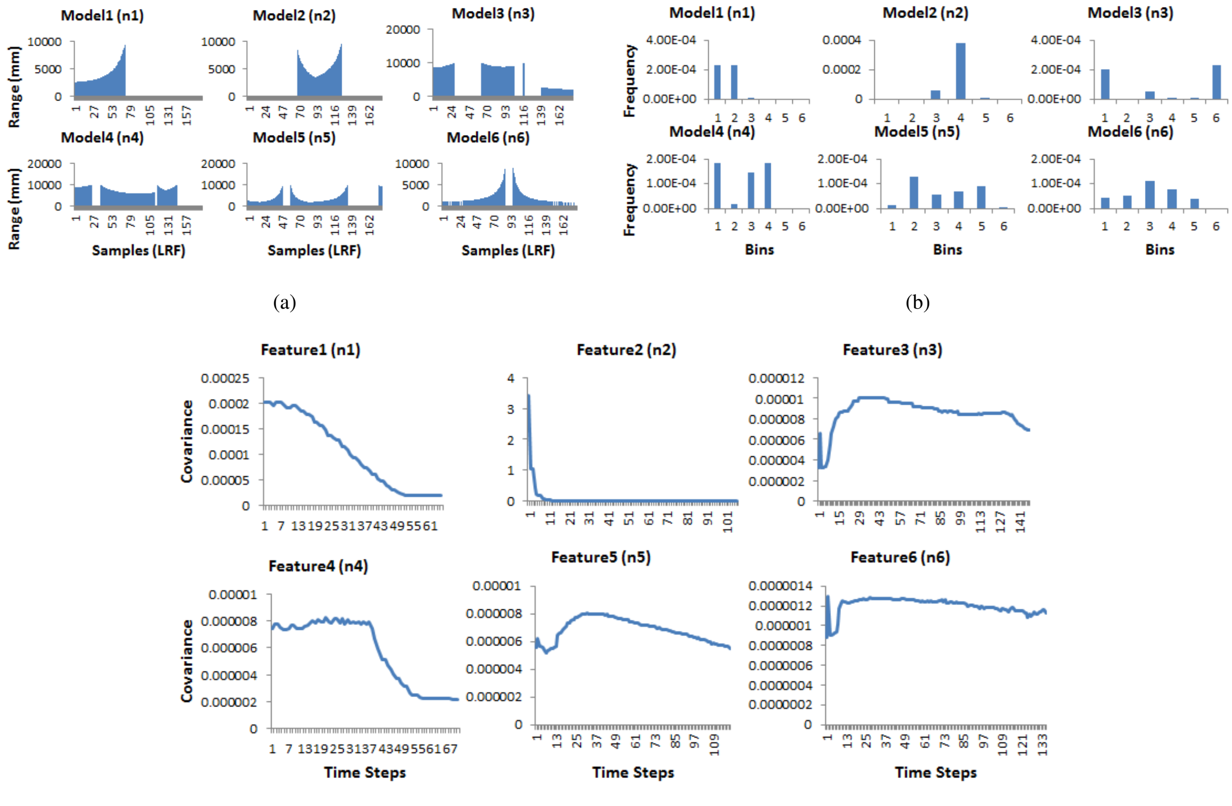

Fig. 4. Recursive Bayesian feature approximations. (a) Prior model distributions (raw), (a) Posterior model distributions (histogram), (c) Covariance error.

TABLE II
ERROR PERFORMANCE ACQUIRED AT NODE n FOR 5 RUNS. ERROR VALUES ARE IN MILLIMETERS.

Nodes	Runs 1	2	3	4	5	Average
1	244.0	217.3	240.1	251.3	241.1	238.76
2	261.0	244.2	230.1	264.3	254.5	250.82
3	236.9	241.4	254.0	264.3	253.3	249.98
4	246.1	268.1	264.9	263.4	263.6	261.22
5	242.4	237.7	248.5	254.7	267.3	250.12
6	252.8	241.7	247.1	254.2	260.1	251.18
Average	247.2	241.7	247.4	258.7	256.6	

B. Feature Model Performance

The landmark features have a distinctive distribution as Fig. 4(a) typifies. These are the prior models acquired at each node location n using a LRF. The priors are being given to the recursive Bayesian model not as raw range data, but as histograms. We have used a six-bin histogram for each model, which is said to speed up the online feature recognition process. The conditional distributions undergo as well the same process of converting the range distributions into histograms. Fig. 4(b) depicts the posterior histogram distributions estimated by the recursive Bayesian model for each node. Ultimately, we would like to use the recursive model to estimate the presence of a landmark. This happens as follows.

As shown earlier in Eq. 6, our feature model estimates the covariance between the posterior $p(x_z|z_{n-1})$ and the prior $p(z_{1:n-1}|x_z)$ distribution. Each node has been assigned with a distinct covariance threshold γ_n, signified as the landmark's signature. When the robot approaches the next node n the covariance error diminishes, and γ_n then verifies whether a landmark is relatively close. Fig 4(c) depicts this notion of a landmark being approached at each node. Actually, when the robot enters into a node's radius ρ, and approaches to the node's centre, the covariance error γ is being compared with the covariance threshold γ_n to verify the node location. The initial fluctuations shown in Fig. 4(c) are being filtered for the first 30 time steps, so as to avoid inaccurate landmark verification. A time step is equal to a 100ms delay required by the thread to run the localisation algorithm.

Overall, Fig. 4(c) demonstrates a robust feature recognition method with the covariance errors degrading significantly as the robot approaches to a landmark. This outcome finalises our passive localisation method using an odometry model, incorporated with a recursive Bayesian feature model for landmark recognition. Since the kinematics of a wheelchair is almost identical to the robot's kinematics, we believe that the overall experimental performance of a wheelchair should be relatively similar to the one demonstrated by the robot. In fact, wheelchairs appear to have more accurate wheel encoders, which means that their odometry is much more precise.

V. Conclusions and Future Work

In this work, an alternative passive localisation method has been introduced based on a probabilistic concept. We presented an node-based odometry model for localisation, and a recursive Bayesian filter for landmark recognition. The fusion of these two approaches constitutes the contribution of this work. A multi-node odometry model has been presented is Section III-C, consisting of a node array where each node represents a set of coordinates of a given location. For localisation, a node-to-node transition required a rotational and a translational displacement of the robot. A successful node location was verified by using a recursive Bayesian feature model (Section III-B), for the recognition of a landmark found at each node. In Section IV, we performed several runs in an indoor environment so as to show how effectively passive localisation can work. The overall localisation algorithm presented analytically in Section III-D, while its experimental performance was carried out using an ActivMedia Pioneer robot.

Despite the fact that the primary idea was to utilise a robotic wheelchair, this however remains a prospective application. Future works will focus on active localisation methods, incorporated with the feature extraction model presented in this paper. Such an architecture would have to introduce a posterior $p(\mathrm{x}|f(z),\ u) = p(\mathrm{x}|[\vec{r},\ \mathrm{x},\ \gamma]^T,\ u)$, which describes the probability of a robot being at pose x given a feature extraction vector $f(z)$, and an action u.

Acknowledgment

This research has been financially supported by the EU Interreg IV A 2 Mers Seas Zeen Cross-border Cooperation Programme - SYSIASS: Autonomous and Intelligent Healthcare System. More details can be found at the project's website http://www.sysiass.eu/. We would also like to thank Robin Dowling for the technical support.

References

[1] C. Yi, I. H. Suh, G. H. Lim, and B. U. Choi. Bayesian robot localization using spatial object contexts. In *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IROS'09, pages 3467–3473. IEEE Press, 2009.

[2] D. Schulz and D. Fox. Bayesian color estimation for adaptive vision-based robot localization. In *Dept. of CSE, Washington University, Seattle, WA, USA*, volume 2, pages 1884–1889. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'04), 2004.

[3] Z. Yang, W. Xu, Z. Xiao, and X. Pan. Passive localization of an autonomous underwater vehicle with periodic sonar signaling. In *Sydney OCEANS'10*, pages 1–4. IEEE, 2010.

[4] D. Fox, W. Burgard, and S. Thrun. Active markov localization for mobile robots. volume 25, pages 195–207, 1998.

[5] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11, 1999.

[6] T. Furukawa, F. Bourgault, B. Lavis, and H. F. Durrant-Whyte. Recursive bayesian search-and-tracking using coordinated uavs for lost targets. In *IEEE International Conference on Robotics and Automation (ICRA'06)*, pages 2521–2526. IEEE Press, 2006.

[7] R. Karlsson and F. Gustafsson. Recursive bayesian bstimation, bearings-only apllications. volume 152, pages 305–313. IEE, 2005.

[8] C. Jen-Tzung and C. Jung-Chun. Recursive bayesian linear regression for adaptive classification. *IEEE on Transaction on Signal Processing (TSP'09)*, 57(2):565–575, 2009.

[9] X. Zhao, X. Li, and T. Tan. A novel landmark tree based self-localization and path-planning method for an intelligent wheelchair. In *Proceedings of the 9th IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN'00)*, pages 84–89. Osaka, Japan. Piscataway (NJ): IEEE, 2000.

[10] T. Yaqub, M. J. Tordon, and J. Katupitiya. A procedure to make the probabilistic odometry motion model of an autonomous wheelchair. In *Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation*, pages 526–531. IEEE, 2006.

[11] C. Zhe. Bayesian filtering: From kalman filters to particle filters, and beyond. *Statistics*, pages 1–69, 2003.

[12] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.