A GRID-BASED PROPOSAL FOR EFFICIENT GLOBAL LOCALISATION OF MOBILE ROBOTS

Man Yin Yee, Jaco Vermaak

Cambridge University Engineering Department
Cambridge, CB2 1PZ, U.K.

ABSTRACT

In this paper we present an extension to Monte Carlo Localisation (MCL) to solve the global localisation problem. This extension is in the form of an efficient data-dependent proposal that can be used both for initialisation and re-initialisation after tracking failure or robot kidnapping. The proposal is a Gaussian mixture over a fixed grid of locations, each of which has a sensor structure similar to that of the robot. The robot measurements are matched to these structures to give the best-match orientation for each grid point. The mixture components are then centred on the grid locations and best-match orientations, with the component weights proportional to the best-match likelihoods. Empirical results illustrate that our MCL approach is more computationally efficient than standard MCL, and demonstrates faster recovery from localisation failures.

1. INTRODUCTION

Mobile robot localisation involves the estimation of the pose (position and orientation) of a robot based on its sensor measurements and a map of the environment. There are two main problems under mobile robot localisation: position tracking and global localisation. In position tracking, the initial state of the robot is known and the problem is to keep track of the robot over time and compensate for incremental errors in the robot state. Global localisation is more challenging as it addresses the problem when the initial state of the robot is unknown. This encompasses the kidnapped robot problem [1], where a well-localised robot is moved to some other location without notification.

Probabilistic approaches have been recognised to be one of the best strategies to provide efficient and real-time solutions for the robot localisation problem. Recently, the most popular of these has been Monte Carlo Localisation (MCL) [2, 3], which is an application of Sequential Monte Carlo (SMC) methods [6, 7]in the context of mobile robot localisation. It represents the distribution of the robot state with a weighted set of samples that is recursively updated as more measurements arrive. Its popularity stems the fact that it is able to deal with non-linearities, and represent non-Gaussian and multi-modal distributions.

A naive extension to MCL to deal with global localisation involves adding a small proportion of uniform samples during tracking [2]. However, since the samples are generated without knowledge of the data this approach tends to be very inefficient. Recently, a more efficient method, mixture-MCL, was proposed [4]. In addition to the standard MCL sampling process, mixture-MCL guesses poses based on the most recent sensor measurements. A kd-tree is learned to allow fast sampling from the proposal distribution. The main disadvantage of mixture-MCL is the computational expense required to build and update the tree. Other approaches include adapting the sample size during localisation [2, 5].

In this paper we propose an extension to MCL that solves the global localisation problem. Our approach is to construct an efficient data-based proposal that can be used for both initialisation and re-initialisation. We do so by constructing a grid over the environment, where each grid point has a sensor structure similar to that of the robot. We then match the robot measurements to the grid points, and determine, for each grid point, the orientation of the best match and the corresponding likelihood. This matching can be done very efficiently, since the expected measurements for each grid point need only be computed once. Grid points close to the true robot pose will tend to have a relatively large associated likelihood for the best match. Using this principle, we define the grid-based proposal as a Gaussian mixture over the grid points, with one component for each grid point. The component is centred on the grid point location and best-match orientation, with its weight proportional to the best-match likelihood. Typically, only a small number of components will contribute to the bulk of the probability mass, and only these are included in the final mixture.

For initialisation we sample from this grid-based proposal to generate the initial sample set. Since this approach uses the information in the measurements, far fewer samples are required compared to schemes that initialise uniformly. During tracking we combine the grid-based proposal with the robot dynamics in a mixture fashion to act as the proposal for new robot poses. We dynamically adjust the weighting of this mixture to reflect our confidence in the estimation accuracy. If the robot is confident in its pose estimate the weight for the grid-based component is low and samples are mostly generated from the dynamic model. If, in contrast, the confidence in the pose estimate is low, the weighting is increased, and more samples are generated from the grid-based proposal. The use of an efficient data-based proposal and an online adaptive scheme for the weighting means that far fewer samples are required to achieve the same estimation accuracy as naive extensions to MCL.

The remainder of the paper is organised as follows. In Section 2 we describe the models for the robot and its range-finding sensors. Section 3 describes the MCL framework. We derive our grid-based proposal in Section 4, and illustrate its performance in Section 5. Finally, we conclude with a summary in Section 6.

2. ROBOT MODEL

We consider a generic mobile robot, equipped with range-finding sensors placed evenly on the robot. We will denote the robot state at time t by $\mathbf{x}_t = (x_t, y_t, \theta_t)$, with (x_t, y_t) the robot location, and θ_t its orientation. We will denote the model for the robot dynamics by $p_d(\mathbf{x}_t|\mathbf{x}_{t-1}, u_t)$, where u_t is the control command. This model encapsulates the robot kinematics and any uncertainty in the modelling and state evolution. To accommodate the fact that the robot may be moved, or that track may be lost so that re-initialisation is necessary, we augment the dynamics with a uniform component, *i.e.*

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}, u_t) = (1-\alpha)p_d(\mathbf{x}_t|\mathbf{x}_{t-1}, u_t) + \alpha U(\mathbf{x}_t), \quad (1)$$

where $U(\mathbf{x}_t)$ is the uniform distribution over the environment, and $0 < \alpha < 1$ is its weight in the dynamics.

The robot sensors are short range range-finding sensors. They have a maximum range of 300 cm, a diffraction angle of 20°, and a noise level of 5%. Each sensor measures the distance to the closest obstacle within the detection range. We will denote the range measurement for the j-th sensor at time t by $r_{j,t}$, and denote by $\mathbf{y}_t$ the collection of all the range measurements. For the observation likelihood $p(\mathbf{y}_t|\mathbf{x}_t)$ we adopt a model similar to the one in [1], *i.e.*

$$p(\mathbf{y}_t|\mathbf{x}_t) = \prod_j p(r_{j,t}|\mathbf{x}_t). \quad (2)$$

Thus, the sensors measurements are assumed to be independent conditional on the robot pose. For each sensor the observation model can be expressed as $p(r_{j,t}|\mathbf{x}_t) = p(r_{j,t}|r_{j,\mathbf{x}_t})$, where $r_{j,\mathbf{x}_t}$ is the expected range given the robot pose and environment map. This model is a mixture of three components: a Gaussian component centred at the expected range that models the event of detecting a known obstacle, an exponential component that models maximum-range measurements and the event that a sensor fails to detect an obstacle, and a uniform component that models unexpected readings caused by unknown obstacles.

3. MCL LOCALISATION

The objective of robot localisation is to estimate the posterior distribution of the robot state given all the available measurements $p(\mathbf{x}_t|\mathbf{y}_{1:t})$, also known as the *belief state*. Using Bayes' rule this distribution can be recursively updated according to

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \propto p(\mathbf{y}_t|\mathbf{x}_t) \int p(\mathbf{x}_t|\mathbf{x}_{t-1}, u_t) p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1}. \quad (3)$$

The recursion is initialised with the initial state distribution $p_0(\mathbf{x}_0)$, which is normally assumed to be uniform.

The recursion in (3) does not lead to closed-form expressions for the models of interest in robot localisation, and approximate numerical techniques are required. Of these MCL have proved to be the most successful, since it its able to deal with non-linearities, non-Gaussianities and multi-modality. It represents the belief state by a weighted set of samples that is recursively updated as new measurements arrive. If the sample set $\{w_{t-1}^{(n)}, \mathbf{x}_{t-1}^{(n)}\}_{i=1}^N$ approximates the belief state at the previous time step, MCL generates new samples for the current time step by simulating from a suitably defined proposal distribution, *i.e.*

$$\mathbf{x}_t^{(n)} \sim q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(n)}, u_t, \mathbf{y}_t), \quad (4)$$

and then updating the importance weights to

$$w_t^{(n)} \propto w_{t-1}^{(n)} \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(n)}) p(\mathbf{x}_t^{(n)}|\mathbf{x}_{t-1}^{(n)}, u_t)}{q(\mathbf{x}_t^{(n)}|\mathbf{x}_{t-1}^{(n)}, u_t, \mathbf{y}_t)}. \quad (5)$$

The sample set $\{w_t^{(n)}, \mathbf{x}_t^{(n)}\}_{i=1}^N$ is then a valid approximation of the new belief state. Resampling needs to be performed from time to time to avoid degeneracy of the sample-based representation. For a full discussion of these issues, see *e.g.* [6, 7].

The success of MCL hinges on the design of the proposal distribution. Standard MCL sets the proposal equal to the robot dynamics, so that the new weights becomes proportional to the sample likelihoods. This takes no account of the new measurements, and leads to tracking failure when the robot is kidnapped. Mixture-MCL alleviates this problem by defining the proposal as a mixture of the dynamics and a component that is an approximation to the inverted likelihood. The approximation is based on a kd-tree that is learned from the measurements. In the next section we present an alternative data-based proposal that requires no learning.

4. AN EFFICIENT GRID-BASED PROPOSAL

Similar to mixture-MCL, we define the proposal as a mixture of the robot dynamics and a data-dependent component, *i.e.*

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}, u_t, \mathbf{y}_t) = (1-\phi)p(\mathbf{x}_t|\mathbf{x}_{t-1}, u_t) + \phi q(\mathbf{x}_t|\mathbf{y}_t), \tag{6}$$

where the mixture coefficient $0 \leq \phi \leq 1$ is allowed to vary with time. Our aim is to design the data-dependent component so that it is an efficient and reasonably accurate approximation of $p(\mathbf{x}_t|\mathbf{y}_t) \propto p(\mathbf{y}_t|\mathbf{x}_t)p_0(\mathbf{x}_t)$. We do so by constructing a grid over the environment, where each grid point has a sensor structure similar to that of the robot. We adopted a regular grid, with the grid point separation determined by Monte Carlo tests to compute the error statistics as a function of separation. During tracking we match the measurements $\mathbf{y}_t$ to the grid points, and determine, for each grid point, the orientation of the best match and the corresponding likelihood. Grid points close to the true robot pose will tend to have a relatively large associated likelihood for the best match. Using this principle, we define the data-dependent proposal as a Gaussian mixture over the grid points, with one component for each grid point, *i.e.*

$$q(\mathbf{x}_t|\mathbf{y}_t) = \sum_{i=1}^{G} \beta_i \mathcal{N}(\mathbf{x}_t|\boldsymbol{\mu}_i, \mathbf{\Lambda}), \tag{7}$$

where G is the number of grid points. The component for the i-th grid point is centred on the grid point location (x_i^G, y_i^G) and the best-match orientation θ_i^*, so that the mean is given by $\boldsymbol{\mu}_i = (x_i^G, y_i^G, \theta_i^*)$. The isotropic covariance $\mathbf{\Lambda}$ is the same for all grid points, and is set proportional to the distance between the grid points. Finally, the mixture weight is set proportional to the best-match likelihood, *i.e.* $\beta_i \propto p(\mathbf{y}_t|\boldsymbol{\mu}_i)$. Typically, only a small number of components will contribute to the bulk of the probability mass, and in practice only these are included in the final mixture.

The proposal in (6) is useful for both initialisation and re-initialisation after tracking failure or robot kidnapping. For initialisation we set $\phi = 1$, *i.e.* samples are generated only from the data-dependent component. Since constant re-initialisation may prove to be detrimental to performance, as well as computationally expensive, we want to perform it only when the confidence in the robot pose estimate is low. We achieve this by monitoring the moving average of the maximum likelihood over the samples, and allow re-initialisation only if this quantity falls below a predetermined threshold. The weight for the re-initialisation component is then set to be proportional to the extent by which this quantity falls below the threshold.

Since we allow frequent re-initialisation the samples at any time may not be from a single mode. Thus, care has to be taken when computing estimates of the robot pose, such as the weighted sample average. Subsequent to initialisation we do not compute pose estimates until the samples collapse to a single cluster, *i.e.* the diagonal of the rectangle that encloses all the samples is smaller than a threshold. Upon re-initialisation, the newly generated samples may not be consistent with the original cluster set, thus showing a multi-modal distribution. Therefore, these samples are not used in pose estimation, but only those from the robot dynamics component of the proposal. The samples that originated from re-initialisation are not used in pose estimation until all the samples have again collapsed to a single cluster.

5. EXPERIMENTAL RESULTS

Our proposed MCL with re-initialisation algorithm is evaluated in the context of indoor mobile robot localisation using Matlab simulations. For this purpose robot trajectory data along with sensor measurements have been generated from the model described in Section 2.

5.1. Global Localisation

In this experiment the global localisation and tracking abilities of our MCL algorithm is evaluated and compared against standard MCL. Figure 1 (right) shows the simulation environment with the paths we will use to assess the performance.

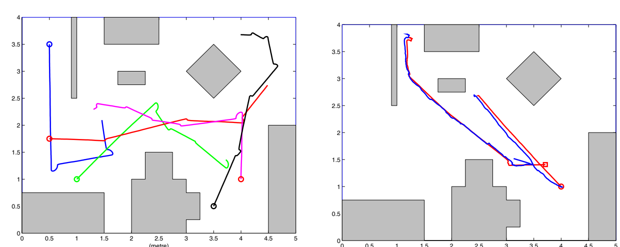

Fig. 1. Paths for global localisation (left) and an example of kidnapping (right).

Figure 2 shows the performance results for standard MCL and MCL with re-initialisation as a function of sample size. The results were obtained by averaging over 100 runs for each of the paths in Figure 1, each with different simulated sensor measurements. The loss rate is defined as the percentage of time during which the algorithm has lost track, *i.e.* when the estimated robot location deviates by more than 50 cm from the true location. The position estimation error is only computed during periods of tracking. Including estimation errors during periods of track loss tends to overstate the error and does not reflect the tracking ability of the algorithm.

MCL with re-initialisation achieves consistently low estimation errors and loss rates that are virtually independent

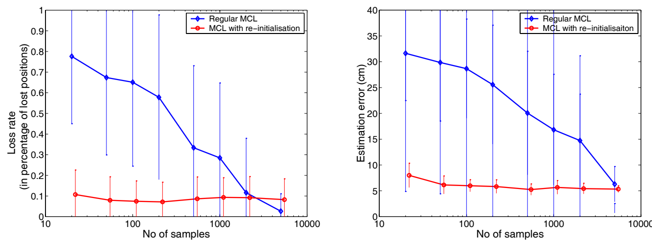

Fig. 2. Loss rate (left) and position estimation error (right) as a function of sample size for standard MCL (blue diamonds) and MCL with re-initialisation (red circles).

of the sample size. For sample sizes as low as 100 the loss rate is as much as 60% smaller than that for standard MCL. This is due to the efficiency of the data-dependent proposal. For very large sample sizes MCL with re-initialisation is outperformed by standard MCL. This is consistent with the result in [4], where experiments have shown that the optimal sample size is between 1000 and 5000.

5.2. Kidnapped Robot Problem

Figure 3 shows results for the kidnapped robot problem similar to those for global localisation in Figure 2. These results demonstrate the superiority of MCL with re-initialisation to standard MCL and MCL with added random samples [2]. The trajectories in this case each included one kidnapping event. One such example is given on the right in Figure 1. As illustrated in Figure 4, the algorithm is able to detect the kidnapping event, and the data-dependent proposal successfully re-initialises the samples on the new robot location. This result is achieved with only 100 samples.

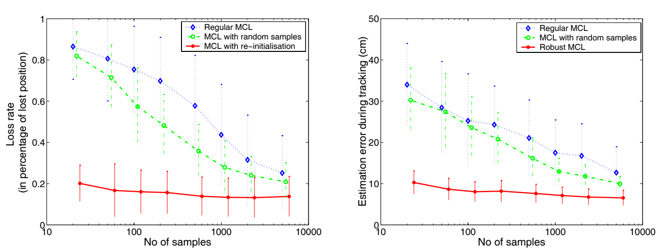

Fig. 3. Loss rate (left) and position estimation error (right) as a function of sample size for standard MCL (blue diamonds), MCL with added random samples (green circles) and MCL with re-initialisation (red dots).

6. CONCLUSIONS

We have presented an efficient extension to MCL to solve the global localisation problem. Samples for (re)-initialisation are generated from a data-dependent grid-based proposal

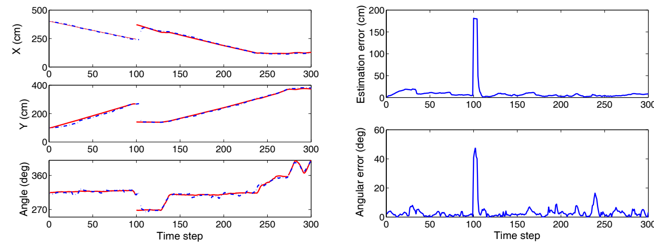

Fig. 4. True (solid red) and estimated (dashed blue) robot pose (left) and location and orientation error (right) as a function of time for the kidnapped robot example in Figure 1.

that requires only information about the sensor structure of the robot. Experimental results show that our approach can increase both localisation accuracy and efficiency when compared to standard MCL. Moreover, our method demonstrates a significantly faster recovery from robot kidnapping compared to MCL with added random samples. These results were achieved with relatively small sample sizes, illustrating the efficiency and robustness of the data-dependent proposal.

7. REFERENCES

[1] D. Fox, W. Burgard, and S. Thrun, "Markov localisation for mobile robots in dynamic environments," *Journal of Artificial Intelligence Research*, vol. 11, pp. 391–427, 1999.

[2] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte Carlo localisation: efficient position estimation for mobile robots," in *Proc. of American Association for Artificial Intelligence*, 1999.

[3] W. Burgard F. Dellaert, D. Fox and S. Thrun, "Monte Carlo localisation for mobile robots," in *Proc. of IEEE / RSJ International Conference on Intelligent Robots and Systems*, 1999.

[4] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localisation for mobile robots," *Artifical Intelligence*, vol. 128, no. 1-2, pp. 99–141, 2001.

[5] D. Fox, "Adapting the sample size in particle filters through KLD-sampling," *International Journal of Robotics Research*, vol. 22, no. 12, pp. 985–1003, 2003.

[6] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, pp. 197–208, 2000.

[7] A. Doucet, J. F. G. de Freitas, and N. J. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*, Springer-Verlag, New York, 2001.