Pose Uncertainty in Occupancy Grids through Monte Carlo Integration

Daniek Joubert
Electronic Systems Lab
Electrical and Electronic Engineering
Stellenbosch University, South Africa
Email: daniekj@sun.ac.za

Willie Brink
Applied Mathematics
Department of Mathematical Sciences
Stellenbosch University, South Africa
Email: wbrink@sun.ac.za

Ben Herbst
Applied Mathematics
Department of Mathematical Sciences
Stellenbosch University, South Africa
Email: herbst@sun.ac.za

***Abstract*—We consider the dense mapping problem where a mobile robot must combine range measurements into a consistent world-centric map. If the range sensor is mounted on the robot, as is usually the case, some form of SLAM must be implemented in order to estimate the robot's pose (position and orientation) at every time step. Such estimates are typically characterized by uncertainty, and for safe navigation it can be important for the map to reflect the extent of those uncertainties. We present a simple and computationally tractable means of incorporating the pose distribution returned by SLAM directly into an occupancy grid map. We also indicate how our mechanism for handling pose uncertainty fits naturally into an existing adaptive grid mapping algorithm, which is more memory efficient, and offer some improvements to that algorithm. We demonstrate the effectiveness and benefits of our approach using simulated as well as real-world data.**

I. Introduction

Dense and consistent mapping of a mobile robot's environment is vital for collision-free path planning and safe navigation. Measurements obtained from a range sensor mounted on the robot provide information on the structure of the visible environment, but are relative to the robot's unknown pose (location and orientation). In order to combine these range measurements into a world-centric map, some form of simultaneous localization and mapping (SLAM) that estimates the robot's pose at every time step is necessary. However, since landmark measurements and robot motion are inherently noisy, the pose estimates are typically characterized by uncertainty. When building a map it is essential to deal with the uncertainties in pose estimates and range measurements in a principled manner to avoid overconfidence in the map.

Occupancy grids are well-suited for the task of maintaining uncertainty in the map – as new measurements become available over time – and this makes them enormously popular [1]. However, occupancy grids are usually employed to solve the 'mapping with known poses' problem where an exact pose is available at every time step. In practice, where a SLAM system provides the pose distribution, the standard approach for the dense mapping stage seems to be to take the most likely pose from that distribution [2] and discard any further information on the uncertainty in such an estimate. We address this issue and propose a computationally tractable means of incorporating pose uncertainty into occupancy grid maps.

Efforts have been made to improve pose estimation prior to mapping, by decreasing the associated uncertainty [3]. However, the problem of directly incorporating pose uncertainty into the occupancy grid mapping process has received little attention. A notable exception is the work of O'Callaghan et al. [4], [5]. Gaussian processes are used to incorporate the effects of pose uncertainty, leading to a continuous functional representation of map properties (such as occupancy) from which a grid map can be extracted. Our work differs from theirs in the sense that we retain the simplicity of the traditional occupancy grid paradigm and alter only the way in which cell occupancy probabilities are updated. Merali and Barfoot [6] also deal with uncertainty in occupancy grid maps, by relaxing the assumptions of independence between grid cells and independence between measurements taken over time, but they do not specifically focus on the issue of pose uncertainty.

For the purposes of this paper we shall assume that a SLAM system is in place and outputs at every time step a distribution describing the robot's estimated pose. In Section II we outline the SLAM problem and mention two commonly used solutions. These illustrate two fundamental ways of describing pose uncertainty: with parameters of a closed-form probability density function or with a set of weighted particles.

The idea then is to combine dense range measurements taken over time into a consistent map. An estimate of the robot's pose is available, so those measurements can be transformed to global map coordinates. An occupancy grid is used to represent the map, and a brief description of the traditional algorithm is given in Section III. We consider a way of modelling range measurement uncertainty in Section IV and address the issue of pose uncertainty in Section V, where the occupancy grid update equation is altered to include weighted samples from the pose distribution.

An issue in occupancy grids that is becoming increasingly relevant, particularly with the development of more affordable real-time 3D scanning technologies and the need for larger maps, is that of high memory usage. In Section VI we consider the adaptive grid mapping method of Einhorn et al. [7], which selects and updates local resolution based on the measurements, and argue how it integrates easily with our handling of pose uncertainty.

We test our methods, both in simulation and with real data, and report on results in Section VII. We find that our inclusion of pose uncertainty does not reduce accuracy, compared to output from the traditional algorithm, but rather increases information in the map.

II. Pose Uncertainty from SLAM

Online SLAM uses control inputs and sensor readings to estimate at every time step t the joint posterior probability of the robot's pose x_t and a map m. The map constructed by SLAM is typically sparse, in the form of position coordinates of salient landmarks, and not particularly suitable for path planning or obstacle avoidance. Dense mapping can be performed with measurements from additional sensors such as laser range finders or stereo cameras. These sensors are usually mounted on the robot and, in order to combine such measurements coherently into a world-centric map, the dense mapping process must rely on the localization output from some type of SLAM system.

In our mapping algorithm we incorporate the pose estimates from SLAM and, importantly, also consider the issue of uncertainty in these estimates. We proceed by first providing a brief description of two fundamentally different forms in which SLAM may return this information. Thrun et al. [2] and Durrant-Whyte and Bailey [8] give more in-depth descriptions of the SLAM problem in general, and the two systems mentioned here.

A. Pose uncertainty from EKF-SLAM

It is quite common to assume that the noise in both landmark measurements and robot motion is Gaussian, in order to employ the extended Kalman filter (EKF) for estimating robot pose and landmark locations. A mean state vector and associated covariance matrix are updated at every time step and together define the joint posterior over the robot's pose variables (position and orientation) and landmark positions. In order to extract the pose estimate from this joint distribution one simply marginalizes out the landmark variables. The resulting distribution remains Gaussian [9].

We note that this is an example of the case where the pose uncertainty is represented as parameters of a known closed-form distribution.

B. Pose uncertainty from FastSLAM

In order to circumvent the sometimes restrictive assumption of Gaussian noise, the FastSLAM algorithm of Montemerlo et al. [10] utilizes a Rao-Blackwellized particle filter for its estimation. In this formulation the robot pose variables are represented by particles, and therefore need not assume any particular distribution, while landmarks are estimated by separate low-dimensional EKFs. At every time step the set of particles (either before or after the customary resampling step in the particle filter), along with their importance weights, serve as an approximation to the pose distribution.

Such a set of weighted particles that approximates a distribution is the second fundamental form in which the pose uncertainty can be represented.

III. Traditional Occupancy Grid Mapping

The occupancy grid mapping algorithm, first introduced by Moravec and Elfes [11], discretizes the area to be mapped into a regular grid of cells (squares in 2D or cubes in 3D). Ultimately, every cell must be labelled as either free or occupied. However, since measurements of the environment can be noisy and the exact pose of the robot on which the sensor is mounted can be uncertain, a hard labelling is risky. Instead it seems more appropriate to maintain probabilities of cells being occupied, and herein lies the power of occupancy grids.

If m_i is the event that cell i is occupied, the aim is to determine the joint posterior probability $p(m_{1:N}|z_{1:t}, x_{1:t})$. Here $m_{1:N}$ denotes the intersection of $m_1, m_2, \ldots, m_N$, with N the number of cells, and $z_{1:t}$ and $x_{1:t}$ are all the range measurements and robot poses up to time t. Computing this joint probability is intractable, and a standard remedy is to assume conditional independence between cells [2] so that

$$p(m_{1:N}|z_{1:t}, x_{1:t}) = \prod_{i=1}^{N} p(m_i|z_{1:t}, x_{1:t}). \tag{1}$$

Once the measurements relative to the robot have been transformed to global map coordinates we can omit the pose itself from the estimation. Furthermore, by applying Bayes' rule and assuming conditional independence between measurements at different times, we have

$$p(m_i|z_{1:t}) = \frac{p(m_i|z_t)\, p(z_t)\, p(m_i|z_{1:t-1})}{p(m_i)\, p(z_t|z_{1:t-1})}, \tag{2}$$

and, in terms of log odds ratios,

$$\begin{aligned} &\log \frac{p(m_i|z_{1:t})}{p(m_i^c|z_{1:t})} \\ &\quad = \log \frac{p(m_i|z_t)}{p(m_i^c|z_t)} + \log \frac{p(m_i|z_{1:t-1})}{p(m_i^c|z_{1:t-1})} - \log \frac{p(m_i)}{p(m_i^c)}, \end{aligned} \tag{3}$$

where m_i^c denotes the complement of m_i. This expression is the standard occupancy grid update equation. The log odds ratio of a particular cell at time t is found simply by adding the log odds ratio arising from the new measurement z_t to the log odds ratio obtained at the previous time step and subtracting a prior. This prior log odds ratio is set to 0, i.e. $p(m_i) = \frac{1}{2}$, if no prior information about the environment is available.

The first two terms on the right-hand side of (3) require probabilities of the form $p(m_i|z)$. This is called the inverse sensor model, and we provide details of ours in the next section.

IV. Measurement Uncertainty

The occupancy grid algorithm assigns and maintains a probability for every cell, based on measurements captured at discrete time steps. A range measurement consists of one or multiple rays emanating from the sensor, each associated with a distance to the first observed obstacle along that ray. All cells intersected by a ray must be updated according to some function of that measured distance, and the rules specifying

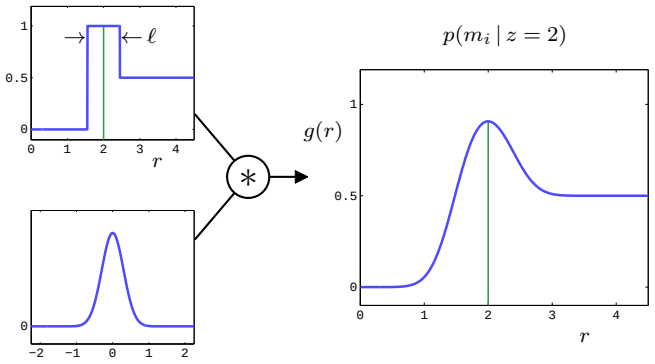


Fig. 1. The ideal sensor model (top left) is convolved with a Gaussian noise model, with $\sigma = 0.3$, to produce our Gaussian inverse sensor model (right). This model corresponds to a range measurement of $z = 2$.

these updates are known as sensor models. For the derivation of our inverse sensor model we focus on sensors with fairly narrow beams, so that the uncertainty in angle is negligible compared to the uncertainty in range.

If the sensor returns perfect noise-free measurements we can employ an ideal sensor model such as the one shown in Fig. 1 (top left). This example corresponds to a range measurement of $z = 2$. Cells closer than 2 units from the sensor are updated with a probability value of 0 (corresponding to free), cells that are around 2 units far with a value of 1 (occupied), and those further than 2 units away are updated with a probability value of $\frac{1}{2}$ which corresponds to 'unknown'. The width of the peak (ℓ in Fig. 1) indicates the range of cell distances that are deemed occupied by the current measurement and can be set to the diagonal of a grid cell.

If we want to assume that the measurement is corrupted by additive Gaussian noise, say centred around the recorded distance, we need to convolve the ideal sensor model with a Gaussian function. If z is the measured distance and the noise variance is σ^2 (which can be a function of z), our Gaussian inverse sensor model is given explicitly as

$$g(r) = \begin{cases} 0, & r \in (-\infty, z - \frac{\ell}{2}), \\ -\frac{1}{2}A + \frac{1}{2}B, & r \in [z - \frac{\ell}{2}, z + \frac{\ell}{2}), \\ -\frac{1}{4}A + \frac{1}{2}B - \frac{1}{4}C, & r \in [z + \frac{\ell}{2}, \infty), \end{cases} \tag{4}$$

where $A = \mathrm{erf}\big(-z/(\sqrt{2}\sigma)\big)$, $B = \mathrm{erf}\big((r - 2z + \frac{\ell}{2})/(\sqrt{2}\sigma)\big)$ and $C = \mathrm{erf}\big((r - 2z - \frac{\ell}{2})/(\sqrt{2}\sigma)\big)$. The function g is obtained by convolving the ideal sensor model and a zero-mean Gaussian with variance σ^2. As mentioned, the variance of the Gaussian noise model can depend on the measured distance, and this dependence in turn is dictated by the type of sensor used. The uncertainty in range measurements returned by a stereo camera sensor, for example, grows quadratically with distance and σ is often chosen as $z^2/(bf)$ where b is the baseline and f the focal length [12].

The rest of the discussion is not dependent on a specific choice of inverse sensor model. In our experiments, however, we employ the Gaussian model in (4).

V. Pose Uncertainty

Occupancy grid mapping typically functions under the assumption that the pose of the robot is known exactly, at every time step, so that measurements relative to the sensor are easily transformed to global coordinates before being incorporated into the map. The problem with this assumption is that, if the robot's pose has large uncertainty, the resulting map will not reflect it.

We deal with this problem by sampling from the pose distribution (returned by the SLAM system), transforming the measurement to map coordinates using every sample individually and, finally, updating the map by summing over the samples.

A. Sampling from the pose distribution

We consider two basic forms in which a typical SLAM system might return the pose distribution, namely closed-form (such as the multivariate Gaussian returned by EKF-SLAM) and a sample-based approximation (such as the weighted particles returned by FastSLAM).

If the pose distribution is known in closed form we can sample directly from it using the inverse transform method [13]. In some cases, particularly when the distribution is Gaussian, the number of samples can be calculated from a required level of representativeness [14]. If the pose distribution is already approximated by a set of weighted samples, as is the case when a particle filter is employed for SLAM, there is no need for further sampling and we can proceed.

B. Integrating the samples

Let us now assume that the pose distribution at time t is described by a set of samples

$$x_t = \left\{ x_t^{[1]}, x_t^{[2]}, \ldots, x_t^{[M]} \right\} \tag{5}$$

and corresponding weights (summing to 1)

$$w_t = \left\{ w_t^{[1]}, w_t^{[2]}, \ldots, w_t^{[M]} \right\}. \tag{6}$$

Using each pose sample we transform the measurement at time t to global coordinates, to obtain a set of M measurements

$$z_t = \left\{ z_t^{[1]}, z_t^{[2]}, \ldots, z_t^{[M]} \right\}. \tag{7}$$

The log odds ratio with which to update cell i is now approximated using the set of transformed measurements z_t and weights w_t. If y is a random variable defined as

$$y = \log \frac{p(m_i|z_t)}{p(m_i^c|z_t)}, \tag{8}$$

its expected value can be estimated as

$$\mathbb{E}[y] = \int p(y)\, y\, dy \;\approx\; \sum_{j=1}^{M} w_t^{[j]} \log \frac{p(m_i|z_t^{[j]})}{p(m_i^c|z_t^{[j]})}. \tag{9}$$

This Monte Carlo approximation replaces the first term on the right-hand side of (3). We note that it will converge to the true expected value as $M \to \infty$ (according to the strong law of large numbers).

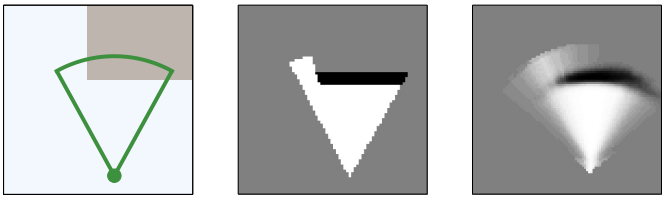


Fig. 2. An example to illustrate the effect of incorporating pose uncertainty: a simulated 2D environment and field of view (left); the occupancy grid resulting from using the exact pose and an ideal sensor model (middle); and the occupancy grid after the incorporation of uncertainty in robot position and orientation (right).

We update a particular cell's value by computing a log odds ratio for every sample using the inverse sensor model, multiplying that ratio by the sample weight, taking the sum over all samples, and then adding the result to the cell's currently stored value (and subtracting the prior, if applicable).

We illustrate the effect of this strategy by means of a simulated example in Fig. 2. The robot is placed in a simple 2D environment and a range measurement is captured. Uncertainty in pose is modelled with a 3-dimensional Gaussian distribution (for position and orientation), with a randomly generated covariance matrix. Samples are drawn from this distribution and a map is constructed. In order to depict occupancy probabilities we use grey levels ranging from white (free space) to black (occupied space).

VI. Adaptive Occupancy Grid Mapping

An important consideration in the design of an occupancy grid is that of cell size. The map must be able to represent a certain level of detail, necessitating a fine resolution, but memory and computational limitations often hamper such a requirement. The use of an adaptive grid, that changes its resolution locally based on the measurements received, can be hugely effective at alleviating this conflict. In this section we explain how our mechanism for handling pose uncertainty is extended to adaptive grids.

An adaptive grid map consists of an array of squares or cubes with different side lengths. A standard way of representing such maps is by means of region trees — quadtrees for 2D maps and octrees for 3D maps. By allowing nodes on different levels of the tree to be active, a map of varying resolution is obtained.

The tree is typically initialized at some level. As measurements are received over time, cells are checked and, if deemed necessary, recursively split for higher resolution in a local region or merged for lower resolution. We focus on the splitting and merging rules of Einhorn et al. [7], as our incorporation of pose uncertainty fits naturally into their method, and offer some improvements.

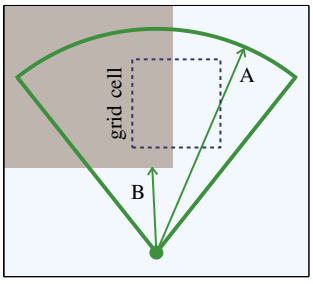
A. Splitting a cell

A cell should be split if its current resolution is inadequate to model the environment as observed by the sensor. That is, we need to split a cell if it receives conflicting information regarding its occupancy. Einhorn et al. [7] keep track of the number of hits (measurement rays indicating an obstacle within the cell boundaries) and misses (rays that pass through the cell before hitting an obstacle). The distribution of hits and misses is compared to an expected distribution, that accounts for sensor noise, by means of a χ^2-test. If the test fails, the cell is believed to be partially occupied and subjected to a split. The process is continued recursively until no more splits are performed or until a predefined finest resolution is reached. Once a cell is split, the newly created children inherit their parent's occupancy probability from the previous time step.

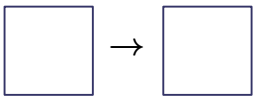
We note that this test alone may give rise to errors in the map. Consider the example in Fig. 3. The dashed lines in (a) represent a grid cell for which only misses are recorded (those rays that hit an obstacle before reaching the cell are ignored) so that the cell is believed to be entirely free of obstacles (b). However, this current measurement indicates that only part of the cell is free and the rest unknown, as shown in (c), necessitating a split. It is therefore clear that rays hitting obstacles before reaching the cell in question must be taken into consideration.

We adapt the algorithm of Einhorn et al. by counting not only hits and misses but also unknowns (rays that hit obstacles before reaching a particular cell). The distribution of these three observables is then compared to expected distributions, using a χ^2-test with one extra degree of freedom, and if any significant deviation is found the cell is split. Both the hit and miss counts are carried over to the next time step, while the number of unknowns is discarded as it need not influence a cell that becomes visible in a later view.

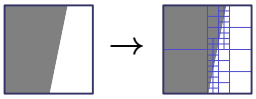
This technique for increasing local resolution is easily extended to incorporate our handling of pose uncertainty. As explained in Section V, we have at every time step a set of measurements with accompanying weights. For every cell we count hits, misses and unknowns for every individual measurement in this set, multiply those counts by the corresponding sample weight and sum over the samples.



(a) ray A passes through a grid cell, while B stops before reaching it



(b) cell is believed to be entirely free and should not be split



(c) cell is in fact measured as partially free and must be split

Fig. 3. In adaptive occupancy grids we count hits and misses for every cell. This example shows that these counts alone can lead to errors. No hits are recorded for the grid cell in (a), and it is therefore believed to be entirely unoccupied (b). However, by taking into consideration those rays that hit obstacles before reaching the cell, we realize that the measurement indicates the state to be only partially unoccupied, and otherwise unknown, so the cell must be split (c).

B. Merging cells

Once all necessary splits have been done and every active node's probability value has been updated, we check to see if any cells can be merged (or pruned from the tree). Following the work of Einhorn et al. [7], children with a common parent are merged if they satisfy any of the following criteria:

- the standard deviation of their probability values is less than some predefined threshold and their mean is within a threshold distance of 0 (certainly free) or 1 (certainly occupied);
- all probability values are greater than a predefined value;
- all probability values are less than a predefined value.

Merging is performed recursively as long as the criteria are met. After a merge, the newly activated cell must receive a probability value. For this we assign the mean of the merged cells' log odds ratios. In doing so little information is lost since we merge siblings only when they have similar values or are almost certainly occupied or free.

VII. Experimental Results

In this section we present some results using simulated as well as real-world data. Since our mapping algorithm requires a pose distribution at every time step, we must have a SLAM system running in the background. It is important to note that we cannot measure the performance of the occupancy grid mapping algorithm and that of the SLAM system separately, as both influence the map. However, for the purposes of our experiments, we shall assume that the SLAM system does model the pose distributions realistically.

A. Simulated data

We make use of the simulation environment of Brink et al. [15] that was developed originally as a testing platform for FastSLAM with stereo vision. A 2D environment consisting of straight-edge obstacles is created and landmarks are defined. A simulated robot moves through this environment by following set waypoints and gathers measurements (stereo image coordinates) of the landmarks. Noise is added to these measurements and also to the control commands. At every time step the SLAM system takes these inputs and generates a pose distribution. Furthermore, we equip the robot with a simulated laser scanner that captures range measurements for our occupancy grid mapping.

Every cell in an obtained occupancy grid can be classified as either occupied or free, by imposing some threshold, for comparison to the true environment. We can then measure false positive and true positive rates, and by varying the threshold we generate a receiver operating characteristic (ROC) curve.

In Fig. 4 (a) we compare the ROC curve obtained from our inclusion of pose uncertainty ('with pose unc.') to the ROC curve of a map generated from only the most likely pose at every time step ('w/o pose unc.'). The performance of our algorithm is comparable to traditional occupancy grid mapping, which implies that by incorporating pose uncertainty to the map little (if any) accuracy is lost while information regarding that uncertainty is gained.

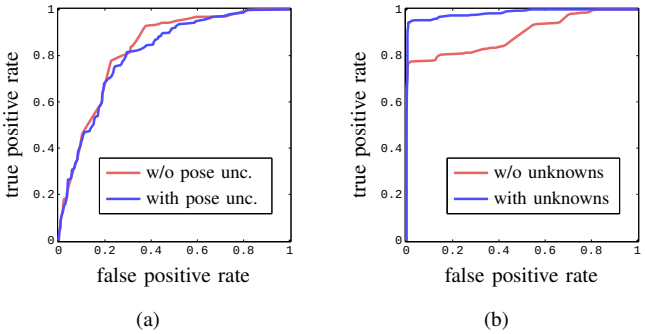


Fig. 4. ROC curves depicting (a) the performance of traditional occupancy grid mapping and our inclusion of pose uncertainty, measured against the true map; and (b) the performance of adaptive mapping with and without counting unknowns, measured against a fine regular grid.

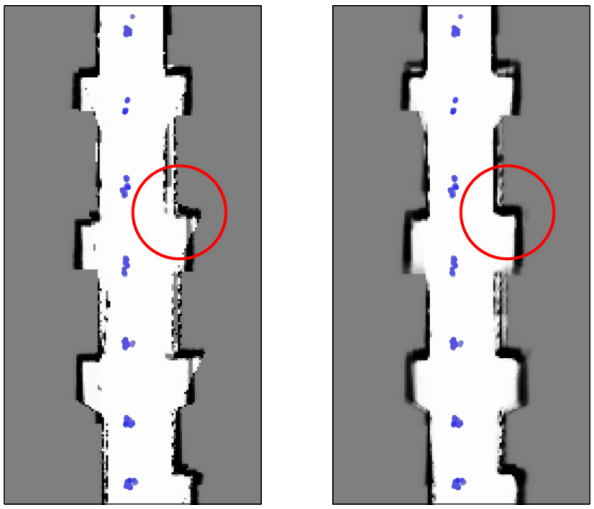
Next we consider the performance of our adaptation to the adaptive occupancy grid mapping algorithm of Einhorn et al. [7]. The aim of an adaptive grid is to replicate its regular (fine resolution) counterpart, using fewer cells. We therefore measure false positives and true positives against a regular grid map, rather than the true environment, and obtain the ROC curves shown in Fig. 4 (b). We see that our adaptation ('with unknowns') does indeed outperform the existing algorithm ('w/o unknowns'), as expected.

B. Real-world data

In order to test our method on real-world data we make use of the Bicocca indoor set from the Rawseeds Project [16]. A wheeled robot equipped with various sensors is driven through fairly narrow corridors with office doors on their sides. Recorded odometry data and tracked stereo image features provide input for our FastSLAM system, which we base on an implementation by Brink et al. [15], that generates pose distributions. Note that, in the absence of loop closure and since no absolute location information is retrievable from tracked stereo image features, the uncertainty in pose is expected to increase over time. At every time step we take the estimated pose distribution, which is given as a set of particles and associated importance weights, and range data from the robot's forward facing laser scanner to construct a 2D occupancy grid map.

In Fig. 5 we compare results from the traditional occupancy grid algorithm, that simply takes a maximum likelihood sample from the pose distribution as input, with our algorithm that incorporates information from all the particles at every time step (in these depictions the robot moves upwards). We show close-ups of the maps to better demonstrate the behaviour of the two algorithms. Our incorporation of pose uncertainty leads to a map with an overall smoother appearance and, more importantly, it seems that the effects of inaccurately estimated poses are lessened.

The result in Fig. 5 (a) illuminates a problem often encountered when a maximum likelihood pose is drawn from a set of particles, for use in mapping. As the importance weights of particles are updated over time, based on the



(a) without pose uncertainty (b) with pose uncertainty

Fig. 5. Close-ups of occupancy grids generated by (a) the traditional algorithm and (b) our algorithm that incorporates pose uncertainty. The pose distribution at every time step, in the form of particles estimated by FastSLAM, is shown as a collection of blue dots. The circled areas emphasize how the incorporation of pose uncertainty can mitigate inaccuracies in the map.

observations, the index associated with maximum weight may jump around and cause a track of the most likely particles to appear erratic. While a weighted average of the particles at every time step might produce a smoother track, and therefore seem more appropriate, its likelihood can be low (given the pose distribution).

VIII. Conclusion

We considered the problem of dense mapping with range measurements captured over time by a mobile robot. We presented a means of incorporating the uncertainty associated with the robot's pose, as estimated by a SLAM system, into occupancy grid maps. Our solution requires samples from the pose distribution. If a particle filter is used for estimation, as is the case in FastSLAM, this distribution is already approximated by samples. Alternatively, if the pose distribution is given in closed-form as is the case when EKF-SLAM is employed, we generate samples from it. The occupancy grid update equation is then altered to include these samples through Monte Carlo integration.

We also showed how the efficient adaptive grid mapping algorithm of Einhorn et al. [7] extends easily to include our handling of pose uncertainty. We identified a potential shortcoming in how cells are selected for splitting, to increase resolution locally, and offered a remedy. For every cell we count not only hits and misses but also unknowns (rays hitting obstacles before reaching the particular cell). In doing so a cell that is only partially observed by the current measurement can be recognized and split if necessary.

Simulations suggest that our incorporation of pose uncertainty is effective, giving a more informative map, without leading to a loss in accuracy. Our adjustment to the adaptive grid algorithm significantly improves the ability of the adaptive grid to mimic its regular (fine resolution) counterpart.

For future work we hope to investigate the possibility of incorporating a Gaussian pose distribution by blurring the grid map with an appropriate kernel. This idea is not that straightforward to implement in practice, however, due to difficulties with the dimensions of the kernel that correspond to the robot's bearing.

We set out to validate the theoretical arguments behind our approach, but there is still room for improving the implementation. In fact, the new update equation can lead to an implementation structure slightly different from the traditional occupancy grid algorithm. For every cell a single assignment can be made from all the measurement beams that intersect the cell across all pose samples. This structure would lend itself to effective parallelization for real-time applications.

Acknowledgments

The authors thank the Harry Crossley Foundation and TATA Africa for funding this work.

References

[1] S. Thrun, *Robotic mapping: a survey*, Exploring Artificial Intelligence in the New Millennium, G. Lakemeyer and B. Nebel (editors), Morgan Kaufmann, 2002.

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

[3] E. Ivanjko and I. Petrović, *Extended Kalman filter based mobile robot pose tracking using occupancy grid maps*, IEEE Mediterranean Electrotechnical Conference, pp. 311–314, 2004.

[4] S. O'Callaghan, F. Ramos and H. Durrand-Whyte, *Contextual occupancy maps incorporating sensor and location uncertainty*, IEEE International Conference on Robotics and Automation, 3478–3485, 2010.

[5] S. O'Callaghan and F. Ramos, *Gaussian process occupancy maps*, International Journal of Robotics Research, vol. 31, no. 1, pp. 42–62, 2012.

[6] R. Merali and T. Barfoot, *Patch map: a benchmark for occupancy grid algorithm evaluation*, IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3481–3488, 2012.

[7] E. Einhorn, C. Schröter and H. Gross, *Finding the adequate resolution for grid mapping – cell size locally adapting on-the-fly*, IEEE International Conference on Robotics and Automation, pp. 1843–1848, 2011.

[8] H. Durrant-Whyte and T. Bailey, *Simultaneous localization and mapping (SLAM): part I*, IEEE Robotics and Automation Magazine, vol. 13, no. 2, pp. 99–110, 2006.

[9] C. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[10] M. Montemerlo, S. Thrun, D. Koller and B. Wegbreit, *FastSLAM: a factored solution to the simultaneous localization and mapping problem*, AAAI National Conference on Artificial Intelligence, pp. 593–598, 2002.

[11] H. Moravec and A. Elfes, *High resolution maps from wide angle sonar*, IEEE International Conference on Robotics and Automation, pp. 116–121, 1985.

[12] F. Andert, *Drawing stereo disparity images into occupancy grids: measurement model and fast implementation*, IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5191–5197, 2009.

[13] E. Rubinstein and D. Kroese, *Simulation and the Monte Carlo Method*, John Wiley & Sons, 2008.

[14] M. Ugarte, A. Militino and A. Arnholt, *Probability and Statistics with R*, CRC Press, 2008.

[15] W. Brink, C. van Daalen and W. Brink, *FastSLAM with stereo vision*, 23rd Annual Symposium of the Pattern Recognition Association of South Africa, pp. 24–30, 2012.

[16] A. Bonarini, W. Burgard, G. Fontana, M. Matteucci, D. Sorrenti and J. Tardos, *RAWSEEDS: robotics advancement through web-publishing of sensorial and elaborated extensive data sets*, Proceedings of IROS'06 Workshop on Benchmarks in Robotics Research, 2006.