Monte Carlo Localization in 3D Maps Using Stereo Vision

Juan Manuel Sáez and Francisco Escolano

Robot Vision Group
Departamento de Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante
{jmsaez, sco}@dccia.ua.es
http://rvg.dccia.ua.es

Abstract. In this paper we present a Monte Carlo localization algorithm that exploits 3D information obtained by a trinocular stereo camera. First, we obtain a 3D map by estimating the optimal transformations between two consecutive views of the environment through the minimization of an energy function. Then, we use a particle-filter algorithm for addressing the localization in the map. For that purpose we define the likelihood of each sample as depending not only on the compatibility of its 3D percetion with that of the observation, but also depending on its compatibility in terms of visual appearance. Our experimental results show the success of the algorithm both in easy and quite ambiguous settings, and they also show the speed-up in convergence when visual appearance is added to depth information.

1 Introduction

Current approaches to solve the problem of localizing a robot with respect to a map that approximately describes the environment are widely based on sonar sensors [1] [2].Such a map is usually build from an occupation grid [3], that is, a bidimensional grid in which each cell contains the probability that its associated space is occupied by an obstacle. In order to obtain such a grid one needs to know robot's motion (odometry), but this information becomes more and more uncertain as the robot moves. Current techniques, like [2], relying on the EM algorithm [4] attempt to deal with such uncertainty and in most cases these approaches, like Monte Carlo methods [6] or bootstrap filters [7] are particular cases of the so called particle filters.

How to translate the latter approaches to deal with 3D maps? Moravec defined in [8] a method to build a 3D occupation grid (3D evidence grid) from several stereoscopic views and outlined the benefits of such research in other robotic problems like path planning and navigation. Current efforts in this area follow different directions. On one hand it is attempted to obtain the geometric primitives that describe the map through the Hough transform [9]: aligning these primitives and conveniently mapping their texture it is possible to build a poligonal model. On the other hand, stereo is exploited to build 2D long-range

sensors [10], that is only the Z component is considered. Other researchers use 3D information to build a topological map with landmarks like 3D corners [11]. In other works long-range laser scanners are used to obtain point clouds of the environment [14] [15].

In this contribution we exploit the mapping method presented in [12] to build a 3D occupation grid from a set of stereo views of the environment and we address the adaptation of particle filters to solve the task of localizing the robot with respect to that grid. As we also code appearance information in the grid it is possible to evaluate the contribution of visual appearance to the localization tasks. The paper is organized as follows. Section 2 describes the approach followed to obtain the 3D map. In section 3 we outline the elements of the Bayesian approach (posterior, likelihood) and present our particle-filter algorithm. In section 4 we show four representative experiments. Finally, section 5 contains our conclusions and future works.

Fig. 1. Digiclops camera and Pioneer mobile robot.

2 3D Maps of the environment

2.1 Observations and actions

Our observations are taken by a Digiclops trinocular stereo camera, with a resolution of 640x480 pixels and a frame rate of 14fps, mounted on a Pioneer mobile robot (see Figure 1). An observation v_t performed at instant t consists of k_t points of the 3D environment. For each one we register both their three spatial coordinates and their local appearance (grey level) in the left image (reference image):

$$v_t = \{p_1, p_2, ...p_{k_t}\}, p_i = (x_i, y_i, z_i, c_i). \quad (1)$$

Assuming a flat ground and also that the camera is always normal to the ground plane, its allowed motions are constrained to: translation along the X axis (horizontal), translation along the Z axis (depth) and rotation with respect to Y axis (vertical). Thus, robot's pose φ_t at instant t is defined by its coordinates

(x, z) in the XZ plane and the rotation α around Y, whereas a given robot action a_t is defined by the increments with respect to the previous pose:

$$\varphi_t = (x_t, z_t, \alpha_t), a_t = (\Delta x_t, \Delta z_t, \Delta \alpha_t). \tag{2}$$

Then, a robot trajectory (exploration) is defined by a sequence of $t-1$ actions, that is $A^{t-1} = \{a_1, a_2, \ldots, a_{t-1}\}$ and t associated observations $V^t = \{v_1, v_2, \ldots, v_t\}$.

2.2 Composing the 3D map

The 3D mapping process of a given environment consists of registering a set of observations V^t along a trajectory A^{t-1}. In order to integrate all the observations in the same 3D map we assume that the robot's initial pose, and thus the origin of the coordinate system of the map, is $\varphi_1 = (0, 0, 0)$. As this pose is associated to observation v_1, to map any observation v_k of the trajectory we need to know its pose, which may be obtained by accumulating all previous actions: $\varphi_k = \sum_{i=1}^{k-1} a_i$. Once the pose $\varphi_k = (x_k, z_k, \alpha_k)$ is estimated we multiply all points in v_k by the matrix 3:

$$T_{\varphi_k} = \begin{bmatrix} \cos(-\alpha_k) & 0 & \sin(-\alpha_k) & x_k \\ 0 & 1 & 0 & 0 \\ -\sin(-\alpha_k) & 0 & \cos(-\alpha_k) & z_k \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Integrating all observations over the same geometric space we obtain a first approximation to the map of the environment composed by a high-density 3D point cloud. This cloud is post-processed to remove replicated points (consider that each observation may produce 11,000 points) and also to discard outliers. Our map model is a geometric version of the Moravec's model [8].

We divide the bounding box of the point cloud $L = \{p_1, p_2, \ldots p_m\}$ in a 3D grid of voxels of constant size T_c (length of each edge of the cube). For each voxel enclosing a number of points greater than U_c we take a prototype (average) resulting a 3D matrix E in which we store the prototypes of each voxel in the grid.

In Figure 2 we show a map of our department. After integrating 187 observations we obtain a point cloud of 2,294,666 points. Setting $T_c = 8cm$ and $T_c = 15cm$ we obtain two maps of 25,637 and 19,809 prototypes respectively. In both cases $U_c = 3$. As stereo errors are not correlated in time, the integration process yields noise-free maps.

2.3 Action estimation

Action estimation is key both for map building and localization. Thus, we apply the energy minimization method described in [12]. Such a method searches the action that minimizes a given distance between two clouds of 3D points. Here, we

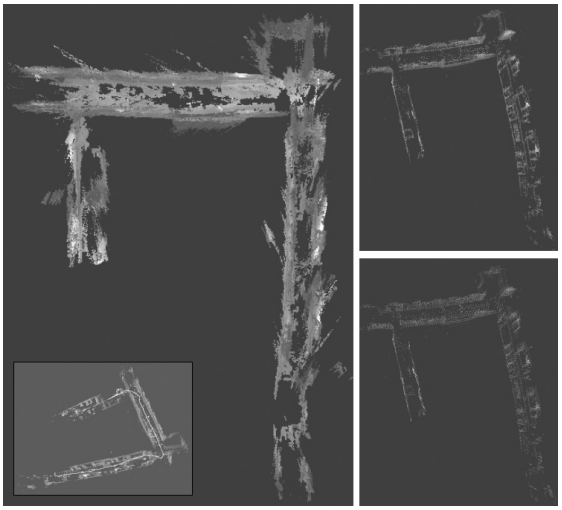

Fig. 2. Example of a map of our department. Left: point cloud after integrating all observations. Right: two maps with different thresholds.

highlight such a distance. Given the observations v_k and v_{k+1}, and the unknown action a_k, we define $\tilde{v}_k$ and $\tilde{v}_{k+1}$ as the observations mapped respectively to $\varphi_k = \sum_{i=1}^{k-1} a_i$ and $\varphi_{k+1} = \varphi_k + a_k$. Action a_k is the one that minimizes $D(\tilde{v}_k, \tilde{v}_{k+1})$, a distance $D(v_a, v_b)$ between two clouds of points v_a y v_b defined as follows:

$$D(v_a, v_b) = \frac{\sum_{i=1}^{K_a} D_{pp}(p_i, P(p_i, v_b))}{K_a}, \tag{4}$$

where $p_i \in v_a$ and $P(p_i, v_b)$ is the closest point to p_i in v_b, that is:

$$P(p_i, v_b) = \arg \min_{p_m \in v_b} \|(x_i, y_i, z_i) - (x_m, y_m, z_m)\|. \tag{5}$$

Finally, $D_{pp}(p_a, p_b)$ is the distance in terms of 3D coordinates and image appearance between points p_a and p_b:

$$D_{pp}(p_a, p_b) = \|(x_a, y_a, z_a) - (x_b, y_b, z_b)\| + \gamma |c_a - c_b|, \tag{6}$$

being γ a penalization constant defined so that both terms lie in the same range.

Given the latter distance, minimization is performed through Simulated Annealing [13] properly initialized: In order to reduce the number of iterations required to converge, we tend to start to search from the previous action. Moreover, as the cost of evaluating the distance is quadratic with the number of points we use a reduced version of the original clouds: We divide each depth image in cells of constant size and then we choose a prototype of each cell. The prototype is de disparity value d_C that minimizes de sum of differences between the disparities of the cell $C = \{d_1, d_2, \ldots d_n\}$:

$$d_C = \arg\min_{d \in C} \sum_{i=1}^{N} |d - d_i| \tag{7}$$

For instance, in Figure 3, we show the original 640x480 image, its associated depth image with 47,202 valid points, and the reduced 160x120 depth image with 1,520 valid points, assuming cells of 4x4 pixels.

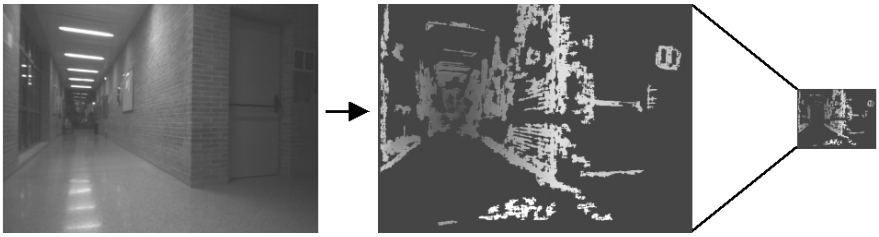

Fig. 3. Obtaining reduced views for action estimation.

In this work, the action estimation is a local process, that is prone to early erroneous estimations. We are currently investigating in globally consistent approaches.

3 Localization in the 3D map

3.1 Posterior term

Given a 3D map M obtained as explained in the latter section, we have adapted the CONDENSATION (CONDitional DENSity propagATION) filter proposed in [5] to the task of obtaining a sample-based estimation of the posterior probability density function $p(\varphi_t|V^t, A^{t-1})$ that measures the probability of the current pose φ_t given the sequence of observations $V^t = \{v_1, v_2 \dots v_t\}$ and actions $A^{t-1} = \{a_1, a_2 \dots a_{t-1}\}$ performed over M.

3.2 Likelihood term

The CONDENSATION algorithm consists of estimating the latter posterior through a sampling process. Each sample represents a localization/pose hypothesis φ_i, and we denote its likelihood given an observation v_j by $p(\varphi_i|v_j)$.

Given the distance between the observation v_j, mapped in the pose hypothesis φ_i, and the map M, that is $D(\tilde{v}_j, M)$, the likehood of such a pose is defined as the exponential expression:

$$p(\varphi_i|v_j) = e^{-\frac{D(\tilde{v}_j, M)}{\sigma^2}} \tag{8}$$

3.3 The CONDENSATION algorithm

The CONDENSATION algorithm encodes the current posterior probabilityof a pose $p(\varphi_t|V^t, A^{t-1})$ given t observations $V^t = \{v_1, v_2 \dots v_t\}$ and $t-1$ actions $A^{t-1} = \{a_1, a_2 \dots a_{t-1}\}$ as a set of N samples $(\varphi_1, \varphi_2 \dots \varphi_N)$ and their associated probabilities $(\omega_1, \omega_2, \dots \omega_N)$ attending to their likelihoods.

Initially the samples set is chosen attending to the prior distribution $p(\varphi_0)$. Then, the iteration associated to instant t consists of three steps: (1) compute the predicted pose of each sample given action a_{t-1}; (2) update the probabilities of each sample given the new pose and the current observation v_t; (3) build a new set considering the latter probabilities:

CONDENSATION Algorithm

Input: $M_{t-1} = \{(\varphi_{t-1}^1, \omega_{t-1}^1), (\varphi_{t-1}^2, \omega_{t-1}^2), \dots (\varphi_{t-1}^N, \omega_{t-1}^N)\}$
Output: $M_t = \{(\varphi_t^1, \omega_t^1), (\varphi_t^2, \omega_t^2), \dots (\varphi_t^N, \omega_t^N)\}$

1. **Prediction:** given action a_{t-1} the predicted pose for each sample $\varphi_{t-1}^i \in M_{t-1}$ is given by
$$\breve{\varphi}_t^i = \varphi_{t-1}^i + a_{t-1} + \epsilon, \;\; i = 1, 2, \dots N$$
where $\epsilon = (N(0, \sigma_x), N(0, \sigma_z), N(0, \sigma_\alpha))$.
2. **Update:** Given the observation v_t the new probability $\breve{\omega}_t^i$ for each sample $\breve{\varphi}_t^i \in \breve{M}_t$ is given by
$$\breve{\omega}_t^i = p(\breve{\varphi}_t^i|v_t), \;\; i = 1, 2, \dots N.$$
3. **Resampling:** build a new set of N samples resampling (with substitution) the set $\breve{M}_t$ in such a way that each sample $\breve{\varphi}_t^i$ is chosen with a probability proportional to $\breve{\omega}_t^i$:
$$(\varphi_t^i, \omega_t^i) \leftarrow \text{Sample from } \breve{M}_t, \;\; i = 1, 2, \dots N.$$
Then normalize the probabilities ω_t^i of the samples in M_t to satisfy $\sum_{i-1}^N \omega_t^i = 1$, that is
$$\omega_t^i \leftarrow \frac{\omega_t^i}{\sum_{j=1}^N \omega_t^j}, i = 1, 2, \dots N$$

Robot localization is seen as an iterative process along step-by-step exploration. Assuming that N is high enough to capture the true location of the robot, the algorithm tends to concentrate all samples around that location as the robot moves around following, in this case, a first-order Markov chain over the action space. We consider that the algorithm has converged when the dispersion $\psi(M_t)$ is below a give threshold U_d and the highest probability $\max(\omega_t^i)$ is greater than another threshold U_v (to deal with situations in which the initial sample is too sparse). We define dispersion $\psi(M_t)$ in terms of the averaged distance between the 2D coordinates of all pairs of samples in the set M_t:

$$\psi(M_t) = \sum_{\varphi_i \in M_t} \sum_{\varphi_j \in M_t} \frac{\|(x_t^i, z_t^i) - (x_t^j, z_t^j)\|}{N^2} \tag{9}$$

3.4 Process optimization

The bottleneck of the algorithm is the computation of the likelihood function for all the samples in the set. More precisely, the estimation of the closest prototype to each transformed point. In order to reduce the computation load, we build offline an extended map $\hat{E}$ in which each voxel stores the coordinates of the closer prototype (non-void cell) in the 3D matrix that registers M. Being $p_m = (x_m, y_m, z_m, 0)$ the minimal coordinates of all points in the map, the cell (i_a, j_a, k_a) in M associated to any point p_a is given by:

$$M(i_a, j_a, k_a) = (\lfloor \frac{x_a - x_m}{T_c} \rfloor, \lfloor \frac{y_a - y_m}{T_c} \rfloor, \lfloor \frac{z_a - z_m}{T_c} \rfloor) \tag{10}$$

M may be transformed into $\hat{E}$ though registering the closer prototype among its direct neighbors and then propagate such computation for each of them until all the space is covered. Then at each (i, j, k) we will have the closest prototype to the center $p_c(i, j, k) = (x_m + iT_c + \frac{T_c}{2}, y_m + jT_c + \frac{T_c}{2}, z_m + kT_c + \frac{T_c}{2})$ of that cell:

$$\hat{E}(i, j, k) = \arg \min_{p_r \in M} \|(x_r, y_r, z_r) - p_c(i, j, k)\|. \tag{11}$$

In Figure 4 we show a 2D sketch corresponding to four of the 79 iterations needed to extend the map in Figure 2 using $T_c = 15cm$ and $U_c = 3$.

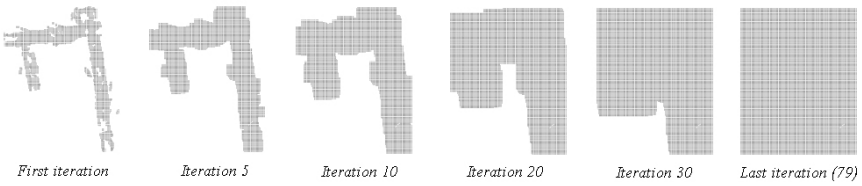

Fig. 4. Growing process for obtaining $\hat{E}$.

4 Experiments and validation

In this section we will show our five most representative experiments addressed to validate the method. In all cases we use the map in Figure 2 with $T_c = 15cm$ and $U_c = 3$.

Experiment 1: First of all, the robot explores a unambiguous part of the environment (the top right corner). Using 1500 samples for pose estimation, at the 7th iteration, $\varphi_* = (2.01m, -17.94m, 276.70°)$ is the sample with highest probability, being the real pose of the robot $\varphi_r = (2.05m, -17.97m, 277.66°)$. Each iteration consumes an averaged time of 2.55 secs in a Pentiun III 900Mhz.In Figure 5 we show several iterations of this experiment.

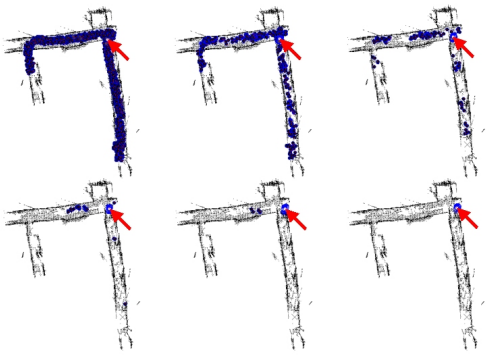

Fig. 5. Experiment 1: First 6 iterations of CONDENSATION (from left to right and from top to bottom) over a easy part. The arrow indicates the actual robot's position.

Experiment 2: Now, the robot explores an ambiguous part of the environment (the long corridor on the right). In Figure 6 we show several iterations of this experiment and in Figure 7 the evolution of the samples dispersion. We also are using 1500 samples for pose estimation. In the 9th iteration the sample with highest probability is $\varphi_* = (1.17m, -10.18m, 193.70°)$ being the real pose of the robot $\varphi_r = (1.14m, -10.21m, 193.78°)$.

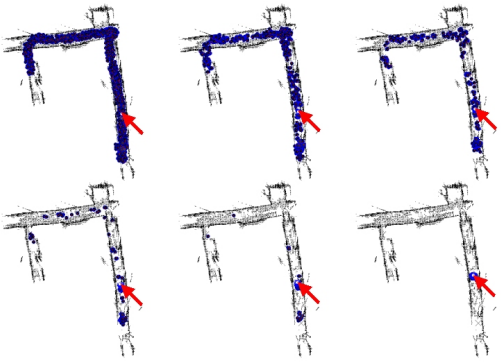

Fig. 6. Experiment 2: Iterations 1,2,3,4,5 y 9 of CONDENSATION (from left to right and from top to bottom) over an ambiguous trajectory.

Experiment 3: In order to evaluate the contribution of visual appearance we have repeated the latter experiment without considering that component ($\gamma = 0$). This results are in a lower convergence rate. Such a rate is even lower when we also discard the Y component (Figure 7).

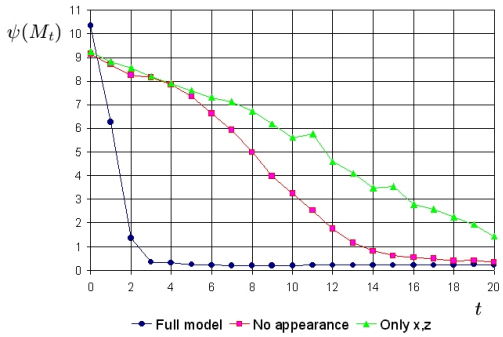

Fig. 7. Experiment 3: Evolution of the convergence rate for the complete algorithm, without appearance and without the Y component.

Experiment 4: In this case our purpose is to analyze the stability of the algorithm with respect to the number of samples considered. We repeat the second experiment but using only 500 samples, and the result is shown in Figure 8. The samples are finally clustered in an incorrect position, revealing the dependence of the approach on the number of samples.

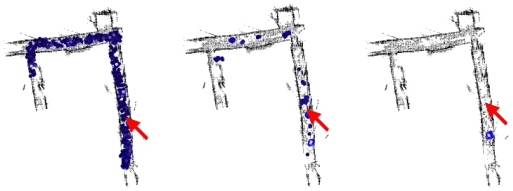

Fig. 8. Experiment 4: Iterations 1, 5 and 10 of the CONDENSATION algorithm with only 500 samples.

5 Conclusions and future work

In this paper we have adapted the CONDENSATION algorithm to the task of localizing a robot in a 3D map build by means of a stereo camera. We have designed a geometric map that encodes both 3D and appearance information and we have developed an auxiliar structure that contributes to reduce the temporal complexity of sampling. In our experiments we have evaluate the performance of the approach in real situations in which the map does not necessarily coincide with the environment and real perceptions may contain significant differences with respect to the data stored in the map.

Given this results, we are investigating both the definition and consideration of 3D landmarks and the use of more elaborated information of appearance like PCA or ICA models.

References

1. S. Thrun et al: Probabilistic Algorithms and the interactive museum tour-guide robot Minerva. International Journal of Robotics Research Vol 19 N11. November 2000.
2. D. Gallardo: Aplicación del muestreo bayesiano en robots móviles: estrategias para localización y extracción de mapas de entorno. Tesis doctoral. Universidad de Alicante, Junio de 1999.
3. F.Dieter, W. Burgard, S. Thrun: The dynamic window approach to collision avoidance. IEEE Robotics and Automation Magazine, 1997.
4. A. Dempster, A. Laird, D. Rubin: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B 39, 1 38, 1977.
5. M. Isard, A. Blake: Visual tracking by stochastic propagation of conditional density. European Conf. Computer Vision. Cambridge, England, Apr 1996.
6. D. Fox, W. Burgard, F.Dellaert, S. Thrun: Monte Carlo Localization: Efficient Position Estimation for Mobile Robots. Sixteenth National Conference on Artificial Intelligence (AAAI), Orlando, Florida, 1999.
7. D.Gallardo, F.Escolano, R.Rizo, O. Colomina, M. Cazorla: Estimación bayesiana de características en robots móviles mediante muestreo de la densidad a posteriori. I Congrés Catal d'Intel.ligncia Artificial. Tarragona, Octubre de 1998.
8. H.P. Moravec: Robot spatial perception by stereoscopic vision and 3D evidence grids. TR The Robotics Institute Carnegie Mellon University. Pittsburgh, Pennsylvania, 1996.
9. L. Iocchi, K. Konolige, M. Bajracharya: Visually realistic mapping of planar environment with stereo. Seventh International Symposium on Experimental Robotics (ISER'2000). Hawaii 2000.
10. D. Murray, J. Little: Using real-time stereo vision for mobile robot navigation. Computer Vision And Pattern Recognition (CVPR'98). Santa Barbara CA, June 1998.
11. S. Se, D. Lowe, J. Little: Vision-based mobile robot localization and mapping using scale-invariant features. IEEE International Conference on Robotics and Automation. Seoul, Korea May 2001.
12. J.M. Sáez, F. Escolano, E. Hernández: Reconstrucción de mapas 3D a partir de información estéreo utilizando un enfoque de minimización de energía. IX Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA 2001). Gijón, Noviembre 2001.
13. S. Kirkpatrick, C.D. Gellatt, M.P. Vecchi: Optimization by simulated annealing. Science, 220:671-680, 1983.
14. Y. Liu, R. Emery, D. Chakrabarti, W. Burgard, S. Thurn: Using EM to learn 3D models of indoor enviroments with mobile robots. Eighteenth International Conference on Machine Learning. Williams College, June 2001.
15. V. Sequeira, K.C. Ng, E. Wolfart, J.G.M Gonalves, D.C. Hogg: Automated 3D reconstruction of interiors with multiple scan-views. Electronic Imaging '99, IS&T/SPIE's 11th Annual Symposium. San Jose, California, USA, January 1999.