The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Probabilistic View Planner for 3D Modelling Indoor Environments

Efraín López-Damian, Gibran Etcheverry, Luis Enrique Sucar and Jesús López-Estrada

Abstract— **These days researchers are looking for design and development of autonomous robots. It is desirable that robots be capable to acquire the information they need to perform actions and make decisions. Any mobile or humanoid robot will need to construct a spatial representation or model of the surrounding environment that allows it to move and execute tasks with success. The reconstruction of an environment is an important and useful capability for these kind of robots. In order to construct a model, the robot needs to obtain information through a series of acquisitions from its sensors by solving occlusions. Therefore, an important issue is how to plan these robot placements (views) optimally, according to certain criteria for the purpose of reconstructing a complete model automatically. In this work we present a view planning algorithm to solve the problem of 3D modelling for indoor environments; the algorithm uses a volumetric representation as a reasoning domain. In this paper we propose the use of probability distribution functions as a model for the desirable behavior of the system, considering perception range data. The method uses a maximum a posteriori estimator to find the perception system parameters that defines the next best view position. We present results in simulation for a five degrees of freedom robot with a 3D range camera mounted on it to validate our approach.**

I. INTRODUCTION

Autonomous 3D environment modelling has an important place in service robotics (mobile or walking systems). It is desirable that a service robot could interact with the environment and displace itself in safety. The robot needs to acquire information in order to navigate and perform actions, so a 3D model has to be constructed. Vision offers a huge source of information for this task; however, it is impossible to obtain a good environment representation considering only one image. Several images are needed to reconstruct the environment and as a consequence an important problem to solve is how to plan the views where the robot should place its camera(s) to capture these images. View planning for three-dimensional environment reconstruction has to deal with the limited camera's field of view and with occlusions. The task is an iterative and incremental process, and at every iteration data are acquired through perceptive 3D sensors placed in the neighborhood of good positions. The data are registered with previous data images and finally fused with the current model. A view planning method for this modelling task must be executed in between every sensor acquisition; then, the problem to solve is to find the best sensor position depending on the already acquired data. This best next view is computed in general by the maximization of an utility function. Several criteria must be considered in this function: sensor geometry, equipment to move the sensor (mobile robot, user operator, PTZ platform,...) and the constraints linked to the task. This planning function must detect the end of the exploration and reconstruction task. The view planning problem is a difficult subject to deal with due to the need of knowledge of perception, computational geometry, robotics, mathematics and planning techniques. The existant methods differ according to the perception task (modelling or reconstruction, recognition, navigation,...) and the knowledge available for the system to perform the planning (partially known model or complete model of objects or environments). A view planner could be part of the process in the environment exploration, localization, mapping and object modelling tasks contributing to the robustness of the system.

E. López-Damian, G. Etcheverry and L. E. Sucar are with the Department of Computer Science, INAOE Puebla, Mexico (eldamian,getcheve,esucar)@ccc.inaoep.mx

J. López-Estrada is with the Faculty of Mathematics, UNAM, Mexico jele@matematicas.unam.mx

II. RELATED WORKS

In literature we find works where the view planning is a central part of the problem, among these we present some main works. that attack the problem of modelling objects or environments because they face almost the same difficulties and use similar techniques. In [1] the complexity is reduced constraining the sensor motion around a cylinder surface centered on the object, the sensor is oriented to the object. The algorithm described in [2] consists of two stages: the first stage applies a voting scheme considering occlusions edges; in the second one a hole filling procedure is executed. In [3], a simpler function is considered but for the first time they a quality notion is proposed. Visibility surfaces and accesible sensor volumes are used in [4] to place the sensor, the object is constructed using solid geometry functions.

Works [5][6] deal with the construction of a 2D model using laser sensors with only horizontal scanning. In [7] a geometric approach is presented, the algorithm computes visibility volumes from where occluded areas can be seen by the sensor. An algorithm based on the detection of occluded areas is described in [8]. The work in [9] uses the same approach proposed, but the occlusions are mapped into a graphics card. In [10] a voxel occupancy grid is used to place the sensor in free space, where certain thresholds are set as parameters to keep the sensor to a specific distance from the floor, ceiling and walls. Our research is based on the general framework described in the work of [11] and [12], where an optimization strategy is used to find a best view at every iteration, by obtaining data information from a scene volumetric representation (voxel map) during the process. The modelling task is assumed to be performed autonomously by a 3D sensor mounted in a pan-tilt mobile platform. The approach presented in this paper differs from the others works because we propose and develop a probabilistic formulation of the problem.

In the next section, an overview of the space representation is presented. Section IV describes our formulation and section V presents the view planning algorithm. Then, in section VI we show some experimental results of the planner using synthetic data acquired from a CAD model of a built environment. Finally, the papers ends with the conclusions and future works in section VII.

III. SPACE REASONING

A volumetric representation (voxel map) of the environment is used by our algorithm. A voxel map is a three-dimensional voxel matrix; a voxel is a small cube of the space. The size of voxels define the resolution of the voxel map. The sensor resolution must be compatible with the voxel map resolution to model the environment properly. For each voxel, we know its location in the voxel map but this data is insufficient. In order to use this voxel map as a reasoning domain, we associate information: a label that has a color associated with it for display purposes, the number of acquired points, the short distance to the sensor and the average surface normal (just for some labels). This surface normal is computed with the average of all the occupied voxels that conforms the surface.

A. Voxel labels

The labels that we use are as follows:

- *Unknown*: a voxel in an unexplored area. At the begining all the voxels are labeled in this manner.
- *Occupied*: a voxel where points are found or acquired. This voxel belongs to the environment surface.
- *Empty*: a voxel in a perceived area but that there are no points acquired in this position. The optique sensor rays have crossed this voxel without touching any surface.
- *Occluded*: a voxel in a see area but unaccessible because it is behind a voxel labeled as occupied.
- *Occplane*: a voxel that was labeled as occluded but adjacent to any of the six faces of an empty voxel.
- *Border*: a voxel that was labeled as occupied but adjacent to any of the six faces of an unknown voxel.

B. Optimal View Function

Once we have defined the space reasoning through the voxel map, the exploration for modelling an indoors environment is intended to be performed by mobile robot moving on the floor (x_r, y_r), a sensor is mounted in a positioning system with three degrees of freedom, an elevation z_s and orientation (θ_s, ϕ_s) of the sensor to acquire an optimal view. So, a viewpoint is defined by five parameters. For the moment the other sensor parameters are constants. In order to find the best view, we define a function that models the interest of a possible robot/sensor placement. This view function must satisfies some criteria:

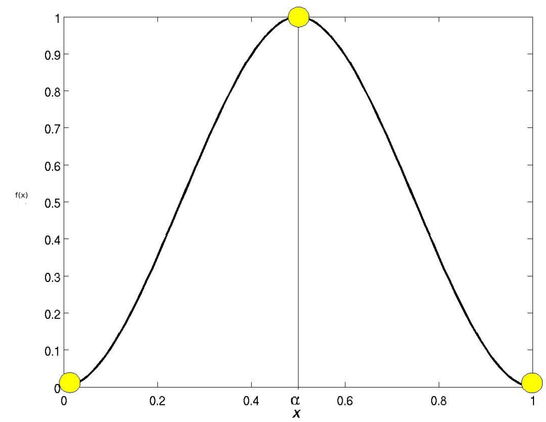


Fig. 1. Desirable behavior for the view function

- The view field must have a certain rate of overlapping with the previous view fields to make the registration process easier.
- Solve occlusions areas: the sensor placement selected must allow to see occluded surfaces behind occupied voxels.
- Perceive new unknown areas.
- Improve if possible occupied voxels quality.

For a given sensor placement, the value of view function is based on the voxels percentages (x) present in the view except for empty voxels and the acquisition quality. To compute these percentages, the view planner simulates the capture of the range image considering the sensor place to evaluate, the sensor parameters and the current state of the voxel map. The sensor simulates the acquisition of a range image by means of a ray tracing algorithm. The range of the rays are limited by the boundaries of the voxel map, resulting in a finite number of voxels of each type. Then, the percentage of a voxel type is computed as the amount of voxels of this type (label) divided by the total number of voxels in the image. For a given voxel type, the view function $f_v(x)$ must be optimal $f_v(x) = 1$ for a desire percentage (optimal) $x = \alpha$ (input parameters) and minimal $f(x) = 0$ when $x = 0$ (none voxel of this type is seen) or $x = 1$ (only this kind of voxels are seen). Fig. 1 shows the desirable behavior of the view function.

1) Quality factor: Quality is defined by the angle δ of incidence between the sensor optical axis and the normal surface. We can consider the quality as constraint factor K in the sensor orientation. Then, in order to improve the quality of the model, it is preferable to place the sensor orthogonal to the environment surface.

$$K = \frac{\sum_{i=1}^{np} cos(\delta_i)}{np} \tag{1}$$

where np is the number of voxels of some kind seen from the current sensor placement. We can estimate the quality that a unknown voxel could have considering the quality of the neighboring seen surfaces (occupied voxels).

2) Global View function: The global view function is the sum of view functions f_{v_i} for each type of voxel times the quality factors for some kind of voxels. For the planner we have define the following global view function:

$$\begin{aligned} F_{v_G}(x) = f_{v_{occupied}} + f_{v_{occplane}} * K_{occplane} \\ + f_{v_{unknown}} * K_{occupied} + f_{v_{border}} \end{aligned} \tag{2}$$

IV. PROBABILITY FORMULATION

Bayesian estimation gives us the possibility of employing a distribution function instead of a polynomial function of utility [12], in order to estimate the parameter x of a conditional distribution $P(\Gamma, x)$ of $\mathbf{x}$ for each voxel percentage configuration of the space, where $x = \alpha$ (optimal) and Γ as the five parameters viewpont group measurements. Thus, the sum of the four voxels parameters distributions, as stated on our space reasoning configuration for voxel labeling, can be computed. In this manner, the *best viewpoint position* γ_i can be obtained.

The prior information we have already established about each x distribution through the *best percentages* view of the voxels in the configuration space, leads us to the problem of *estimating* the x's values in terms of the environment acquisition position Γ. In other terms, it is possible to *predict* the configuration input parameters x's.

A. Observation Model

The probability conditional density function $p(x|\Gamma)$ measures the likelihood, by means of the Bayes theorem, of the input parameter x captured with respect to the *voxels percentages* acquisition configuration, based on a given position $\Gamma = (\gamma_1, \ldots, \gamma_n)$.

B. Bayes Theorem

The probability conditional density function $p(x|\Gamma)$ (x predicted) is defined as:

$$p(x|\Gamma) = \frac{p(\Gamma|x), p(x)}{p(\Gamma)} \tag{3}$$

where $p(\Gamma) = \int_0^1 p(\Gamma|x)\, p(x)\, \mathrm{d}x$ serves as a normalization factor; $p(x|\Gamma)$, $p(\Gamma|x)$ and $p(x)$ are the posterior, likelihood, and prior distributions, respectively.

C. Probability Density Function

In this section we show how the probability distribution function *beta* can be used as a model for the optimal behavior of our system to find the next best view.

In order to best describe each of the voxels expected function values, it is required to consider a probability density function which presents a maximum value for a range of possible values within the interval $0 < x < 1$. The function that satisfies this requirement is the *beta density function*.

The beta density function, taking it as the prior density when estimating the success probability of a binomial distribution, yields as the posterior one, another beta density function [13].

D. Beta Density

If we consider the probability of finding, from n-trials, k-times the x's *best percentage* on viewpoint γ_i, we have the binomial law:

$$p(\Gamma|x) = x^k (1-x)^{n-k} \tag{4}$$

and replacing it into (3),

$$p(x|\Gamma) = \frac{x^k (1-x)^{n-k}\, p(x)}{\int_0^1 x^k (1-x)^{n-k} p(x)\, \mathrm{d}x} \tag{5}$$

In this manner, it is possible to predict the probability of finding the expected voxel percentage x within $0 < x < 1$, by choosing a smooth density function like the beta one. As $p(x)$ is smooth, the product $p(x)$ by (4) is concentrated near at the value k/n, which it is the same that we consider $p(x) = 1$, [13].

Then, we take the identity

$$\int_0^1 x^k (1-x)^{n-k}\, \mathrm{d}x = \frac{k!(n-k)!}{(n+1)!} \tag{6}$$

we find the *beta density function*, by substituting (6) in (5)

$$p(x|\Gamma) = \frac{(n+1)!}{k!(n-k)!} x^k (1-x)^{n-k} \quad 0 < x < 1 \tag{7}$$

We can produce different shapes of the function depending on optimal voxel percentages, see Fig. 2. This cannot be done with another kind of function (e.g. gaussian function). It is assumed that $n > 1$, $k \geq 1$, and $n > k$ as minimal and natural constraints of the probability formulation. In Fig. 2, appears three beta functions that represent the behavior of the optimal percentages of voxels we used in our experiments. For occupied voxels we use $\alpha = 0.2$ (20% of image voxel) and we set the beta parameters k=1 and n=5, for unknown and occplane voxels, $\alpha = 0.375$ with k=1 and n=2.67; finally, for border voxel we are interested to obtain an $\alpha = 0.05$, so k=1 and n=20.

E. Maximum estimator a posteriori

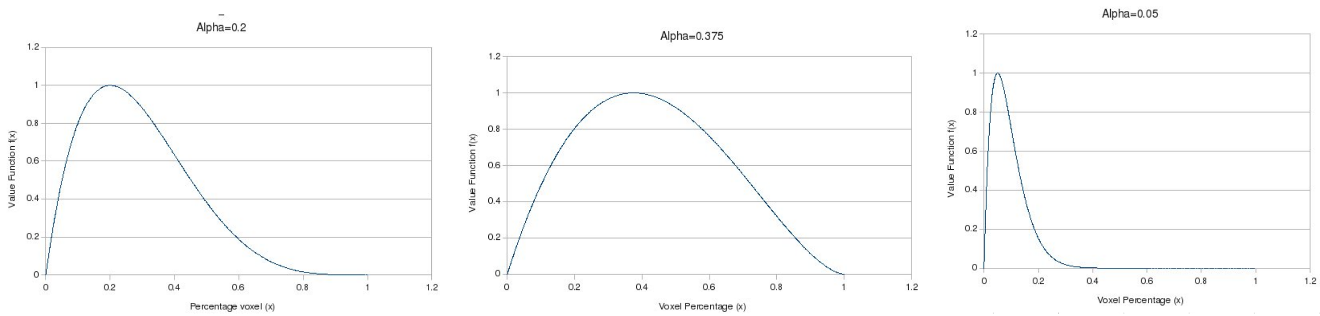
This method gives the estimate of the value of the unknown parameter that maximizes the conditional probability function, which in our case is:

$$\hat{x} = \underset{x}{argmax}\; p(x|\Gamma) \tag{8}$$

Last expression is derived from (3), where $p(\Gamma)$ is independent of x. In general, we don't have previous knowledge of x's percentage viewpoint; hence, $p(\Gamma)$ is considered uniform and lets us finally with the expression

$$p(x|\Gamma) = p(\Gamma|x)\, p(x) \tag{9}$$

which is implemented to obtain (8).



(a) α=0.2 (b) α=0.375 (c) α=0.05

Fig. 2. Probability Density Function for voxel percentages: occupied (left), unknown and occplane (center), and border (right). The x-axis of diagrams are voxel percentages x and y-axis is the function value.

V. VIEW PLANNING ALGORITHM

The algorithm starts by taking a first range image of the environment from a known free-space sensor position in the voxel map. The points acquired from the range image are converted to the coordinate system of the voxel map generating a partial model of the environment by labelling the voxels. Then, the estimator uses a simplex method [14] for (x, y, z) parameters and a tessellated sphere for (θ, ϕ) orientation parameters to construct the conditional probability domain given the current voxel map to generate candidate views. Every view must satisfy the free-space constraint, the view position space must have enough empty voxels to place the sensor, for this a bounding box enclosures the whole robot, if the free-space is equal of bigger than this bounding box the candidate view is valid. The beta function gives the conditional probability for each kind of voxel given the current sensor position. The estimation step outputs the next best view. Next step consists to move the range camera (robot) to this position and we proceed to capture a range image to update voxel map and obtain a more complete model. The process is repeated until at least one of the ending criteria are achieved: there are no occplane voxels in the voxel map, the estimator does not find a next view, meaning the tested views does not provide new information or the model is complete (Ψ). We summarize our iterative method in algorithm 1.

Algorithm 1: View Planner Algorithm for 3D Indoor Environment Modelling

Data: Initial Robot Position$(x_0, y_0, z_0, \theta_0, \phi_0)$, α_i
Result: 3D model
Image_Capture Im;
Update_Voxel_Map $(VM) \leftarrow Im$;
while not *StopCondition (Criteria Ψ)* **do**
    Next_Best_View$\leftarrow$ Maximum_Estimator(α_i);
    Positioning_Robot-Sensor;
    Image_Capture;
    Update Voxel_Map $(VM) \leftarrow Im$;
end

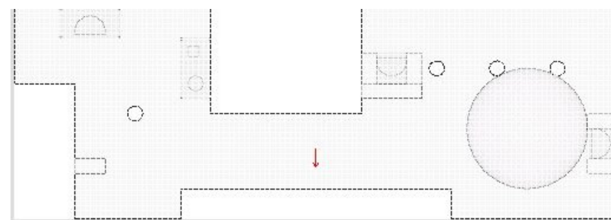
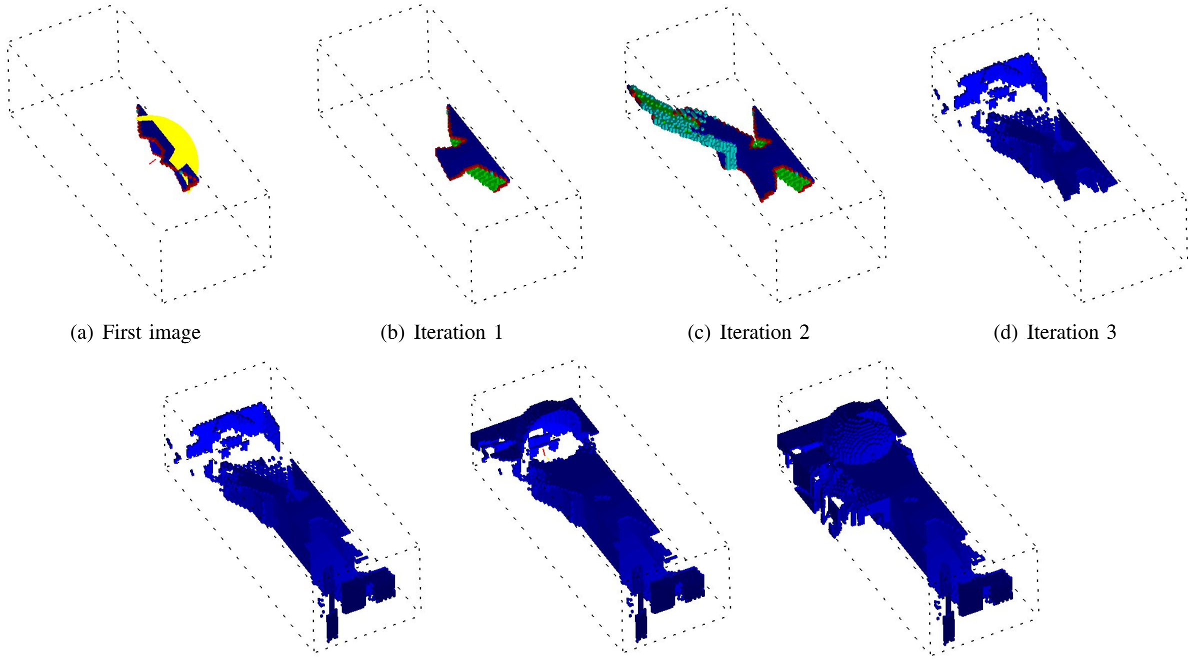


Fig. 3. Top view of the environment model

VI. RESULTS

Our algorithm was tested in simulation, the range camera was simulated and the objects were taken from virtual models. The voxel map dimensions are 52 x 40 x 138 voxels and the voxel size is 15 cm. The simulated range camera returns a range image similar to a real range camera. The system was implemented in C, the machine used for the tests was a MacBook 2.4Gz and 2 GB RAM memory. In Fig. 3 we show the synthetic model of the environment. In Fig. 4 we present six iterations of the view planner, as we saw in Algorithm 1, initially the sensor capture a first image in the central corridor facing the wall, then the planner computes views where only the orientation parameters change due to the limited free space (empty voxels) in the environment, where the walls, ceiling and floor can be reconstructed. In iteration 7 the sensor can move to the left side of the environment, and in the acquisition 9, the cupola and his surroundings can be modeled.

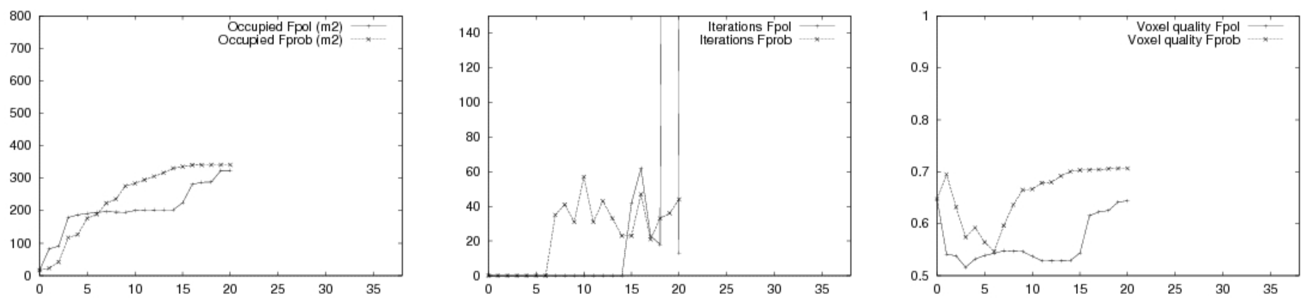
The Fig. 5 shows the comparison diagrams of the two different view planners. Two view functions are compared, the global utility function based on probability density functions and a second often used utility function based on a third-degree polynomial functions [12]. Fig. 5(a) represents the area filled by the occupied voxels, we can see that the probabilistic approach grows quicker than the polynomial function, allowing to perceive more occupied area after 20 views. In Fig. 5(b) shows the iterations the simplex algorithm make to compute the next view. We can see that when the robot is moving for both approaches, the number of iterations to compute the next view generally is bigger in the



(a) First image (b) Iteration 1 (c) Iteration 2 (d) Iteration 3

(e) Iteration 5 (f) Iteration 7 (g) Iteration 9

Fig. 4. Series of view acquisition planned by the probabilistic view planner, in the first views we show different types of voxels and the last images only the occupied voxels



(a) Occupied voxels area (b) Iterations per view (c) Voxel quality

Fig. 5. Comparison of probabilistic versus polynomial approaches, results after 20 views

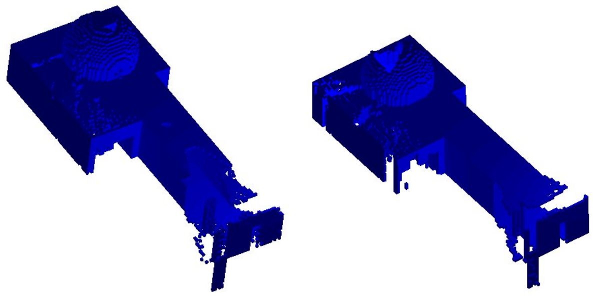
polynomial approach, increasing significantly in some views the computation time for the polynomial case. The Fig. 5(c) shows the mean quality on the occupied voxels, again we can see that the probabilistic planner has a better performance and on every sensor position, it finds a view that improves the quality value. Hence, we can say it satisfies better the criteria imposed for the optimal view. The table I shows the computing time for some views of the process, for both functions this time is similar when only orientation motions are performed, in the probabilistic approach the robot starts to move before the polynomial method does, making that the searching method takes more iterations to find the next view. We are not considering the sensor acquisition time for an image neither the time for updating the voxel map. When the planner finds a view where the robot has to travel the computing time increases for both cases (rows 3 and 4 in table), but this increment is bigger for the polynomial. The total time for the probabilistic planner is smaller.

We have executed both view planners and in Fig. 6, we present the environment model after 20 planned views; we can observe that the image is consistent with previous results. The model constructed using the probability function and estimator has more occupied area, resulting in a more complete environment.

TABLE I

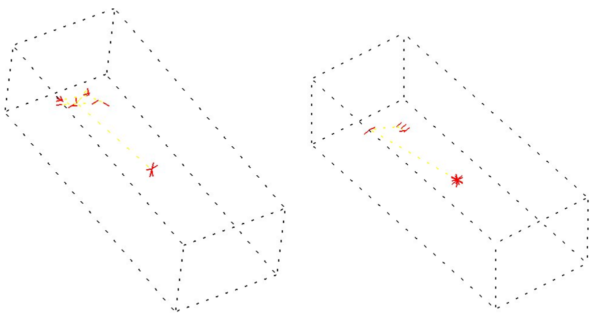
RECONSTRUCTION TIME (VIEWS)

View No.	Time(s) Polynomial F.	Time(s) Probability F.
1	0.000012	0.000009
2	0.000025	0.000018
19	0.034686	0.007475
20	0.034898	0.008089
Total	0.075761	0.060319



(a) Probability Function (b) Polynomial Function

Fig. 6. Environment model after 20 iterations of the view planner



(a) Probability Function (b) Polynomial Function

Fig. 7. Path followed by the sensor

Fig. 7 presents the sensor placements from which the views were acquired. We can see that the polynomial function expends more views to model the corridor and goes to the left side of the environment after around 10 iterations. The probability function in view 10, has a bigger part reconstructed. Finally, in Fig. 8 shows the complete model of the environment, validating our approach.

VII. CONCLUSIONS AND FUTURE WORKS

We have presented a view planning algorithm that determines the next best view to reconstruct any arbitrary 3D environment. Based on a voxel map representation, to position a three degrees of freedom range sensor mounted in a mobile robot. The planner has been evaluated in simulation reconstructing a complex synthetic model environment. The results show a good quality in the reconstructed model, with savings in computation time. The main contribution is that we approach the view problem from a probabilistic point of view, transforming the often used polynomial utility function for a probability beta density function and computing the next best view using a maximum a posteriori estimator. Given the promising results obtained in simulation, our future work is to test it in an experimental mobile robot with a stereo vision system. We will add a motion factor to the global utility function to select views that do the robot travel the shortest possible distance, besides it, by using a motion planner, we can not only be able to find free view robot placements, but paths between two consecutive views.

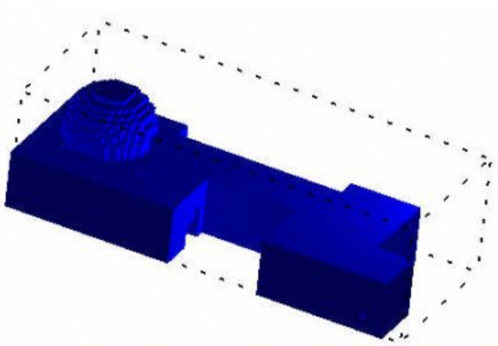


Fig. 8. Complete model of the environment by the probabilistic planner after 42 views

VIII. ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of the National Council of Science and Technology (CONACyT) and INAOE, Michel Devy (LAAS-CNRS) and M. Lozano to provide us with some libraries and CAD model for the experiments.

REFERENCES

[1] R. Pito, "A solution to the next best view problem for automated surface acquisition," *IEEE Transactions on Pattern Analysis and Machine Learning*, vol. 21, no. 10, pp. 1016–1030, 1999.

[2] M. García and S. Velázquez, "A two-stage algorithm for planning the next view from range images," in *In Proc 6th British Machine Vision Conference*, 1998, pp. 720–729.

[3] N. A. Massios and R. B. Fisher, "A best next view selection algorithm incorporating a quality criterion," in *British Machine Vision Conference BMVC98, Southampton*, 1998, pp. 780–789.

[4] P. Allen and M. Reed, "3-d modeling from range imagery : An incremental method with a planning component," *Image and Vision Computing*, vol. 17, pp. 99–111, 1999.

[5] J. colly, M. Devy, M. Ghallab, and L. Pampagnin, "Sensory control for 3d environment modelling," in *International Advanced Robotics Programme, First Workshop on multi-sensor fusion and environment modelling*, 1989, pp. 1–14.

[6] R. Murrieta-Cid, B. Tovar, C. Esteves, and D. Perez, "Robot motion planning under visibility constraints," in *International Symposium on Intelligent Robotic Systems*, 2001.

[7] K. Tarabanis, R. Tsai, and A. Kaul, "Computing occlusion-free viewpoints," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 3, pp. 279–292, 1996.

[8] V. Sequeira, J. Goncalves, and M. Ribeiro, "3d reconstruction of indoor environments," in *IEEE International Conference on Image Processing*, 1996, pp. 405–408.

[9] K. Klein and V. Sequeira, "The view-cube : an efficient method of view planning for 3d modelling from range data," in *Fifth IEEE Workshop on Applications of Computer Vision*, 2000.

[10] B. Null and E. Sinzinger, "Next best view algorithms for interior and exterior model acquisition," in *Advances in Visual Computing*, vol. 4292/2006. Springer Berlin / Heidelberg, 2006, pp. II: 668–677.

[11] J. Sanchiz and R. Fisher, "A next-best-view algorithm for 3d scene recovery with 5 degrees of freedom," in *British machine vision conference, Nottingham, UK*, 1999, pp. 163–172. [Online]. Available: citeseer.ist.psu.edu/sanchiz99nextbestview.html

[12] M. Lozano-Albalate, M. Devy, and J. Sanchiz-Martí, "Perception planning for an exploration task of a 3d environment," in *ICPR '02: Proceedings of the 16 th international conference on pattern recognition (ICPR'02)*, vol. 3. Washington, DC, USA: IEEE Computer Society, 2002, pp. 704– 707.

[13] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 1991.

[14] J. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, pp. 308–313, 1965.