Motion planning for humanoid robots in environments modeled by vision

Alireza Nakhaei, Florent Lamiraux

CNRS-LAAS, University of Toulouse, France $anakhaei@laas.fr, florent@laas.fr$

***Abstract*—The context of this work is vision-based motion planning for humanoid robots in an unknown environment. We present an efficient combination of on-line 3D environment modeling and motion planning methods for humanoid robots (e.g whole body motion and walking) in non-static environment.**

To construct the model of the environment, we rely on 3D occupancy grid that is updated incrementally by stereo vision. As the dimension of configuration space is high for humanoid robots, a roadmap based method is used for motion planning. However, as the environment is not static, it is necessary to update the roadmap after receiving new visual information. In other words, the nodes and edges which are in collision, based on the new update of the environment, must be erased.

Moreover, preliminary steps are necessary for considering the environment as a non-static model. As we construct the model incrementally by vision, several thousands points would appear in each update of the model. Therefore, updating the roadmap raises algorithmic complexity issues. Our approach is an extension of a recent idea to cope with the problem.

After presenting the approach, we implement our method by planning a collision-free motion in a 3D occupancy grid model generated by HRP2 based on stereo vision. As the robot navigates in the environment, it receives updated information through its on-board cameras and refreshes the 3D model of the environment incrementally. Conventionally, the 3D model can be composed of up to millions of voxel. If the statuses of some voxels change, our method uses these changes to update the last roadmap locally. This updated roadmap is then reused for further motion planning.

We evaluate our algorithm by measuring processing time and memory usage in each step and compare them with its descendant.

I. Introduction

In this paper, we present a vision-based motion planning strategy for a humanoid robot in an unknown environment. We have implemented an integrated stereo vision environment modeling and our new method for motion planning in non-static environments.

The goal is to enable a humanoid robot to integrate on-line environment modeling and motion planning in order to explore and interact with a changing environment.

A. Motivation

For increasing the application of robotics in research and industry, we need to develop more autonomous robotic systems which are able to sense and plan in real environments. For that, the robot must be able to interact with the world both in modeling the environment and in planning collision free motions in the model. Working in a real environment will make it essential to consider all changes in the workspace. However, motion planning in such changing environments becomes a much more challenging task. This gives rise to the need for an efficient algorithm capable of coping with changing environments.

B. Related works

a) Environment Modeling: Robotic mapping has been an active area in AI for a few decades. It addresses the problem of acquiring a spatial model of the workspace through available sensors on a robot. A significant amount of early works on mapping has been done in the past decades. We can distinguish two main approaches addressing slightly different problems:

1) sparse 3D mapping aims at building a geometrically coherent sparse map of characteristic features and at localizing the robot in this map. Recent work in this area include [11], [3].
2) dense SLAM aims at building a dense map of the environment integrating obstacles. These techniques are usually based on occupancy-grids. Pioneering work is described in [14]. Later work define a occupancy grid in a probabilistic framework [16], [2].

For more references and details about SLAM, we refer the reader to the following book [17]. As we need to plan motion in the model of the environment perceived by the robot, occupancy grid techniques better fit our requirements.

b) Motion planning in changing environments: A common task in robotics is to plan a collision free motion between initial and final configuration in an environment. Several approaches exist for solving such a common problem of motion planning. One of the popular techniques to solve this problem is to plan an optimal trajectory using Dijkstra algorithm or A* [15] through a uniformly discretized configuration space.

However, this approach is impractical for more complex configuration spaces, such as humanoid robots, in terms of both memory and computation time. Therefore, approaches such as Rapidly exploring Random Trees (RRTs) and Probabilistic Roadmaps (PRMs) have been widely-used in these domains [10], [8]. These approaches work by randomly sampling points in the continuous configuration space. The initial and final configurations are then connected via these samples. Many variants have been proposed in the literature. The best strategy usually depends on the type of problem and of the type of environment considered. [9] provide a comprehensive description of many methods for planning paths in a static environment.

To deal with changing environments, some approaches have been implemented in order to optimize the process of updating precomputed roadmaps at run-time [18], [13], [12]. The approach we propose in this paper to plan motions in changing environment is an extension of an existing approach using a mapping between workspace and configuration space [12].

In [12], Leven and Hutchinson begin by constructing a graph (roadmap) that represents the configuration space. In this first step, obstacles are not taken into account, only auto-collisions are considered. In a second step, they generate a representation that encodes a mapping from cells in the discretized workspace to nodes and edges of the roadmap. In the on-line planning stage, the planner first identifies the cells in the workspace that corresponds to obstacles and uses the encoding mapping to delete the corresponding nodes and edges from the roadmap. The remaining roadmap is used for the next motion planning step. This approach is a good step towards real-time motion planning in changing environments and copes with the associated problems efficiently. It was successfully applied to a robotic arm with 6 degrees of freedom.

However, applying this method to a humanoid robot in a much bigger workspace is not reasonable since the required memory grows quickly with the size of the workspace. In the approach we describe in this paper, we use the same basic idea of work space cell-decomposition, but we use cells of bigger size and we test again for collision the roadmap features lying in cells where obstacles have appeared.

C. Contribution

In this paper, we are interested in motion planning in environments where the position of obstacles is unknown. To construct the 3D model of the workspace, we create a 3D-occupancy grid map which is updated based on stereo-vision information. As the robot navigates in the environment, it successively takes stereo images and updates the 3D model. Creating a 3D model instead of 2D is essential for the humanoid robot to plan whole body motions.

The contribution of our approach is a new outlook to cell decomposing workspaces. In Leven and Hutchinson approach, a cell is the smallest volume of environment which can be considered as *free* or *occupied*. In order to have a good resolution of the workspace these cells should be small enough.

However, in our approach, we define a cell as a larger volume of environment composed of many small voxels. These voxels are the smallest identifiable volume of the environment. In case of changes in the environment, stereo vision provides information about the modified voxels . Then, cells which include modified voxels are detected and all nodes and edges associated with these cells are validated again. In case of failure in validating, the relevant nodes and edges are erased from the roadmap. As the environment is explored and modified, the roadmap grows incrementally.

To our knowledge, this work is one of the first to integrate 3D environment modeling and roadmap-based motion planning together.

The paper is organized as follows. In the following section we describe the method to model the environment. Section III presents our motion planning algorithm and our contribution. Finally, Section IV describes our experimental setup and the results of the tests that was conducted in a model of a real environment.

II. Environment modeling

A 3D occupancy grid map represents the environment in our approach. The occupancy grid representation employs a multidimensional tessellation of workspace into voxels where each voxels stores a probabilistic estimate of its state. In fact, environment is discretized to uniform cubes which are the smallest volume of our 3D model. The size of the cubes is selectable based on our desired resolution in the model. Based on the volume of the 3D model of environment, the model would be composed of up to millions voxels.

$$\mathcal{W} = \bigcup_{i=1}^{n} V_i, \forall i, j \in [1, n], V_i \cap V_j = \emptyset \tag{1}$$

where $\mathcal{W}$ is the workspace and V_i presents voxel i in the 3D model. The state variable associated with a voxel V is defined as a discrete random variable with three states *unknown*, *occupied* and *free*. Moreover, for determining the state of each , 3 probabilities are assigned to each one. $P[s(V_i) = O]$, $P[s(V_i) = F]$, $P[s(V_i) = U]$ are the probabilities of voxel i to be *occupied*, *free* and *unknown* respectively. The robot obtains information about its environment through its cameras and uses stereo vision method to get a range value r for a 3D point in the environment. A stochastic sensor model defined by a probability density function $p(r|z)$ is used to relate the reading r to a true parameter space range value z.

Based on the recent probability density, $P[s(V_i) = O]$, $P[s(V_i) = F]$, $P[s(V_i) = U]$ is modeled for each voxel. These probabilities are stored in G_i and will be updated as the robot gets the new information about the environment.

For certain application such as motion planning, it is necessary to assign a specific state to each voxel of the occupancy map. An optimal estimate of the state of a voxel is given by an extension of the maximum posteriori decision rule in [1]: Voxel i will be considered as:

- *Occupied* if $P[s(V_i) = O] > P[s(V_i) = U]$ and $P[s(V_i) = O] > P[s(V_i) = F]$,
- *Free* if $P[s(V_i) = F] > P[s(V_i) = U]$ and $P[s(V_i) = F] > P[s(V_i) = O]$,
- *Unknown* if $P[s(V_i) = U] > P[s(V_i) = O]$ and $P[s(V_i) = U] > P[s(V_i) = F]$.

For initializing the environment, as no information is available for the model, all voxels are considered as *unknown* and the probabilities are set as follows:

$$\begin{aligned} &\forall i \in \{1, .., n\} \rightarrow \\ &P_i(s(G_i) = O) = 1\%, \\ &P_i(s(G_i) = F) = 1\%, \\ &P_i(s(G_i) = U) = 98\%, \end{aligned} \tag{2}$$

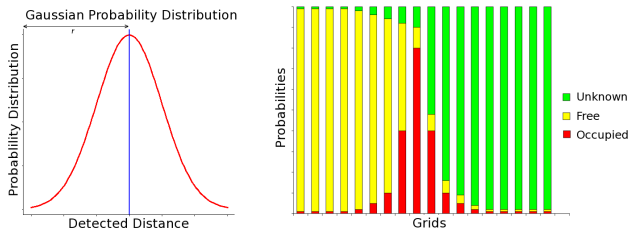

Fig. 1. Estimate the probabilities for each voxel in occupancy model from sensor data.

A. Updating 3D model

As it was explained, a sensor model is used to relate the reading r to a true parameter space range value z and to obtain the concerned probabilities for each voxels. In fact, in this approach, the cameras is modeled with simple Gaussian uncertainties in depth as follow:

$$p(r_j|z) = \frac{1}{\sqrt{2\pi}} exp(\frac{-(r_j - z)}{2\sigma^2}) \tag{3}$$

where j is the index of each 3D point of the stereo-image. Figure 1 illustrates a simple model which is used to obtain the probabilities.

To allow the incremental composition of sensory information, the sequential updating formulation of Bayes' theorem is used to determine the voxel probabilities [4]. Moreover, using these algorithms will keep the dynamic characteristic of the model of environment. In fact, in case of having an obstacle which enters in the model or exits from the model, the probabilities of the concerned voxels will change rapidly and their states will be modified. Extending the approach of [5] to the model, and considering the current probabilities of G_i as $P_i(s(G_i) = O \mid r_t)$, $P_i(s(G_i) = U \mid r_t)$,$P_i(s(G_i) = F \mid r_t)$ based on observations $\{r\}_t = \{r_1, r_2, ..., r_t\}$ and given a new observation as $\{r\}_{t+1}$, the updated estimate will be given as follows:

$$P[s(G_i) = O \mid \{r\}_{t+1}] =$$

$$\frac{P[r_{t+1} \mid s(G_i) = O].P[s(G_i) = O \mid \{r\}_t]}{\sum_{s(G_i)} P[r_{t+1} \mid s(G_i)].P[s(G_i) \mid \{r\}_t]} \tag{4}$$

At each step of updating the model, the current probabilities are obtained from the occupancy grid map and at the end of calculations, results are stored in the model. These stored probabilities are used as the initial probabilities of next update.

So, as the robot navigates in the workspace, it receives new information and generate the 3D model of the environment incrementally. This approach generates a non-static 3D model. Therefore, obstacles may appear or disappear in the model. Figure 2 illustrates an environment which is modeled by stereo vision. It should be mentioned that localization plays an important role in environment modeling. In fact, as the data obtained form sensors are read in the sensor coordinate frame, localization of sensors in the model is important for having a precise model.

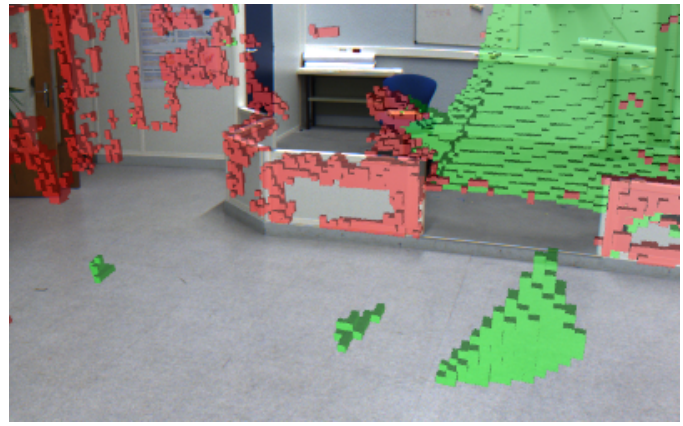

Fig. 2. The robotics lab in LAAS-CNRS which is modeled based on this approach. Red and green voxels (cube) represent occupied and unknown volume respectively in the model. The work space is modeled with 3D voxels of 5cm×5cm×5cm. Notice that this model is the result of several stereo-images taken from different points.

Fig. 3. The HRP2 humanoid robot.

At the following step, the generated model of environment is used for motion planning. The robot is allowed to move in the free workspace and incrementally improve the occupancy grid map and explore the unknown area.

III. Path planning

Most of the real life environments are non-static and for path planning in such environments, the key issue is how to deal with the changes in such models. As mentioned earlier, our approach of motion planning in changing environment is an extension of an existing method that uses a mapping between a cell-decomposed workspace and configuration space [12].

A. Constructing roadmap

Having the instant model of environment, we begin with creating a roadmap in the environment. Our planner uses all the internal constraints such as degree of freedom and kinematic limitations to create roadmap in the free zone of the 3D model. Nodes are generated by shooting random configuration in the C-space and acceptable nodes are connected to the roadmap by acceptable edge. We denote the roadmap as $\mathcal{R}(N, E)$ where N and E are the list of nodes and edges respectively. However, as the environment is not static, the free zone would be modified

as the robot received new information. So some nodes in N and some edges in E become invalid in the next instance of the environment model. Validating all nodes and edges with new features would be time-consuming. Our approach copes with this problem to prevent validating all nodes and edges.

B. Updating roadmap

As it was mentioned in the previous section, to prevent erasing and regenerating roadmap, it is necessary to update the roadmap after receiving new information from environment. The preliminary solution is validating all nodes and edges of the roadmap with the updated model . The nodes and edges which are in collision with the new model can be erased and the rest parts of roadmap can be used for planning a motion in the updated environment. Considering a cost for validating, it is inefficient to validate all the nodes and edges in a large environment because of a simple modification in the workspace. For example, consider a humanoid robot in a large environment composed of millions of voxels. Suppose that the status of one voxel changes from *free* to *occupied*. It is inefficient in term of time and cost to validate all nodes and edges of an existing roadmap. To prevent validating all nodes and edges, Leven, and Hutchinson [12] proposed a mapping between the workspace and configuration space. Based on this approach, work space is decomposed to uniform small cells and each node and edge are associated to one or various cells. However, cells are the smallest volumes of environment which would have the state of *occupied* or *free*. In case of changing the status of a cell from *free* to *occupied*, the associated nodes and edges are erased and the planner will use the rest of roadmap for future planning.

Our approach is an extension of this idea to cope with the problem. However, the important different is that in our approach, cells are not the smallest element of the environment and with this method, we would save a high amount of memory in dealing with the problem. Also, instead of erasing nodes and edges, our planner validates the nodes and edges concerned by the modified cells.

C. Work space cell-decomposition

A workspace cell-decomposition method is used in our approach. However, as opposed to [12], cells are not the smallest part of the environment. In fact, each cell is composed of up to thousand voxels and these voxels are the smallest element of the occupancy model of the environment. The concept of voxel is explained in section II.

$$\mathcal{W} = \bigcup_{i=1}^{m} C_i, \forall i, j \in [1, m], C_i \cap C_j = \emptyset \tag{5}$$

C_i presents a cell in the model and $\mathcal{W}$ is the workspace. In other hand,

$$C_i = \bigcup_{i=p}^{q} V_i, \forall i, j \in [p, g], V_i \cap V_j = \emptyset \tag{6}$$

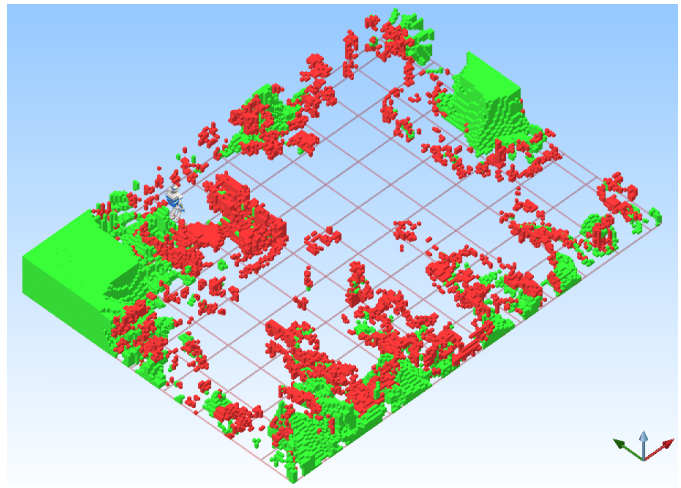

Fig. 4. Motion planning in 3D model of robotics lab in LAAS-CNRS. The size of workspace is 17m $\times$12.5m$\times$1.7m. The workspace is modeled with 10cm$\times$10cm$\times$10cm 3D *Voxels*. Also, workspace is decomposed to the 1.5m$\times$1.5$\times$1.5m *Cells* to implement the approach.

It should be mentioned that in this approach, cell-decomposition is not necessarily uniform in the workspace. Therefore, based on any preliminary information about the environment, the shape and size of cells would be different in a specific region to optimize the number of calculations.

Moreover, as in [12], a mapping between configuration space $\mathcal{C}$ and work space $\mathcal{W}$ is built to associate any configuration to a list of cells.

$$\phi(\mathbf{q}) = \{C_i \mid A(\mathbf{q}) \cap C_i \neq \emptyset\} \tag{7}$$

where $\mathbf{q}$ is configuration and $A(\mathbf{q})$ is a volume of environment which is occupied by robot in $\mathbf{q}$. In fact, thanks to this mapping, each $\mathbf{q}$ is associated to a list of cells which are in collision with the robot at this configuration. By using (7), a list of cells can be associated to any nodes and Edges.

On other hand, based on this approaches, a list of nodes and edges will be assigned to each cell. Therefore:

$$C_i = \{L_i^N, L_i^E\} \tag{8}$$

where, L_i^N is list of nodes which the robot in their configuration will be in collision with C_i. Also, L_i^E is list of edges which the robot is in collision with C_i if it tracks that edges.

D. Updating Cell

L_i^N and L_i^E are updated as the planner solves the problem and finds a path between initial and final configurations. To optimize the calculation time, cell lists are updated during execution of the generated path. Therefore, by using the updated roadmap, the concerned nodes and edges should be added to the associated lists of each cells C_i.

For updating L_i^N with a new node, the robot will be considered at the concerned $\mathbf{q}$ and the node will be added to L^N of cells which are in collision with robot. Updating the L^E is more complicated and time consuming. In fact, it concerned

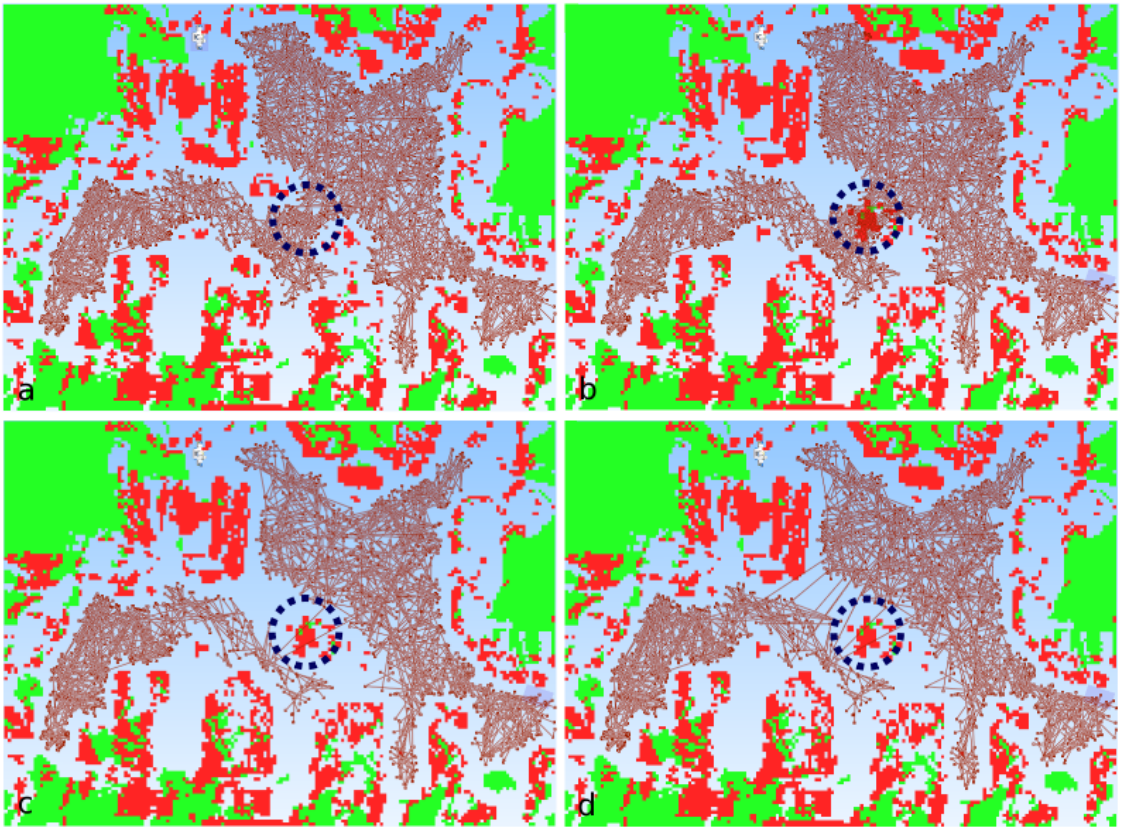

Fig. 5. Illustration of roadmap management during updates of workspace. Red and green voxels represents *occupied* and *unknown* zone respectively: a) A roadmap is illustrated in the top view of a 3D model of an environment which is generated by vision. b) The model is updated based on new information from camera. c) The roadmap is modified based on the updates in the 3D model by using our cell decomposition algorithm. Some nodes and edges are erased from the roadmap during the modifications. d) The modified roadmap is used for a motion planning process. Some nodes and edges are added to the roadmap.

to a mapping between an edge and a set of cells which are in collision with robot if the robot sweeps the concerned path. To update the L_i^E, the same algorithm as in [12] was used.

Figure 4 illustrates a 3D model which is decomposed to 1.5m cell to implement the approach.

IV. Experimental results

A. Exploring unknown environment

We experimented our approach in a real environment with HRP2. The robot is supposed to navigate in a 3D occupancy grid model of environment generated by vision. As mentioned in section II , localization is important for generating a precise model of environment. Localization errors become a critical issue when generating models of a large environment. So, a small volume of environment is allocated for exploring by HRP2.

The model of the environment is generated incrementally. At each step of updating the model, HRP2 is required to look around and take various photos. In generating a whole body motion for turning the head around and taking photos, a generalized inverse kinematic (IK) model was used as proposed in [19], [20]. Based on this method, such tasks as gaze control are treated and whole-body joint angles are computed to execute the motion. Meanwhile, several criteria such as manipulability, stability or joint limits are monitored and finally, the motion is executed. This method enables the robot to look around and take various photos. Then, the robot uses this photos to update the model of environments.

After updating the model, the robot is supposed to plan a path towards a predetermined position in the environment. As it was explained in section II , the model is composed of 3 types of voxels: *free*, *occupied* and *unknown*. Our planner consider the *occupied* voxels as obstacles and plan a path between initial and predetermined goal. Probably, a part of the computed path has intersection with the *unknown* voxels. So, the next step is computing the last configuration of the path which is still in the *free* zone of the model. To compute this configuration, in this step, the *unknown* are considered obstacle as well. Then, the path is discretized to successive configuration and the last valid configuration is selected. The robot follows the path to this configuration and again takes various photos to update the map and explore the remaining unknown zone to reach the predetermined goal.

We have 2 steps in planning a motion and navigating in the model. In the first step, our approach is applied to a motion planner of the bounding box of the humanoid robot as proposed in [20] and generates a path for the bonding box. In fact, this planner generate a path for a car-like

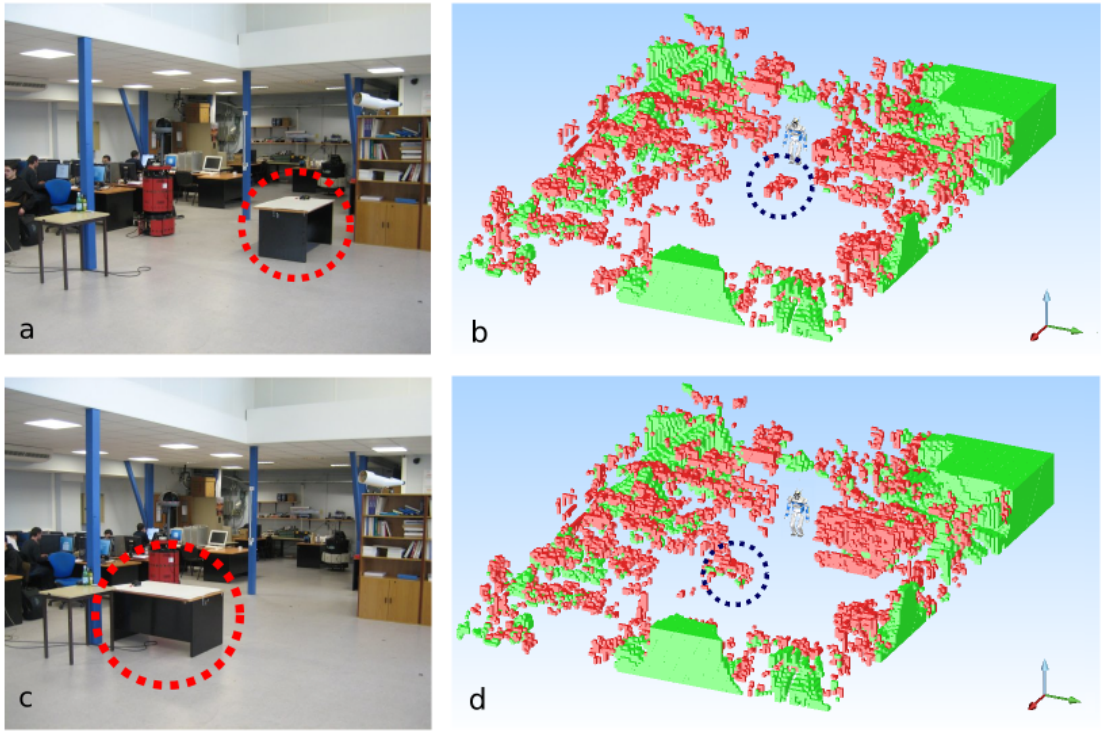

Fig. 6. Changes in the environment. HRP2 uses the cell decomposition method to validate some of the nodes and edges in the existing roadmap to for planning next motions

robot. In the second step, the planned collision-free path is transformed into dynamic humanoid locomotion. After the path is converted into footsteps, a walking pattern generator is applied to generate a dynamically stable walking motion using a method proposed by Kajita et al. [7] based on a preview controller for zero moment point (ZMP).

Based on this algorithm, HRP2 explores an unknown environment to reach a predetermined location. This location can be the position of an object which have specific color. Figure 7 illustrates the evolution of a 3D model of an unknown environment along with planning motion in the model.

B. Motion planning in changing environments

In order to validate our approach in motion planning in changing environment, we needed to build a large model of environment. However, we do not have yet a precise localization module on HRP2 robot. We therefore built the map using another mobile robot *Jido* [1] equipped with an accurate localization module. HRP2 then planned motions in this model.

Jido is a mobile manipulator which is equipped with a stereo camera and also a laser scanner based localization module. *Jido* was used in our experiments for taking several photos. These photos were used to generate the 3D occupancy model of the environment. The robotic lab of LAAS-CNRS is selected for conducting the tests as a large environment. 2 models of the lab were generated. The main difference between these 2 models is a table at the middle of lab which was displaced. Displacement of this table caused a part of *free* environment to become *occupied* and also, an *occupied* volume of environment to become *free*. Although the only difference in the 2 states of environment is the displacement of the table, there are much more differences between the two 3D models. In fact, based on any error of localization, some parts of model were modified. There are 7848 *occupied* voxel in the first model. There are 1810 voxels which have been *free* or *unknown* in the first model and they became *occupied* in the second model.

Figure 6 illustrates the 2 generated models which were used to test our approach in changing environments. Although environment modeling was not done on-line with HRP2, the approach is valid and it does not disturb the efficiency of the approach.

Moreover, instead of whole body motion planning in the changing environment, we used a path planning algorithm for the HRP2 bounding box as explained in the last section . Although we simplified the problem of motion planning from a robot with with 40 degree of freedom to a box with 3 degree of freedom, the approach and the results are valid for any roadmap based planner.

The allocated size of the environment in these models is 17m $\times$ 12.5m $\times$ 1.7m. The size of each voxel in the these 3D occupancy grid map was 10cm$\times$10cm $\times$10cm. It is simply obtained that the model is composed of 361250 voxels.

[1] http://www.laas.fr/robots/jido/data/en/jido.php

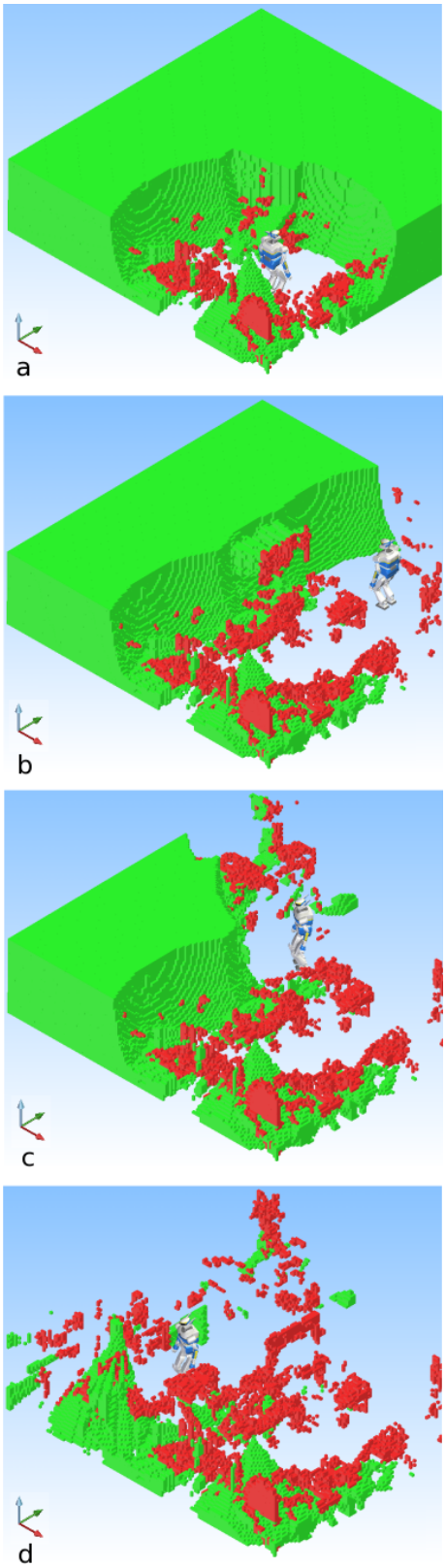

Fig. 7. Exploring an unknown environment incrementally. Red and green voxels represent occupied and unknown volume respectively in the model. The robot plans a path in non-*occupied*zone and then follows the generated path to the boundary of *unknown* zone.

TABLE I
TIME PERFORMANCE.

Test	Solving problem 1 [ms]	Updating Roadmap [ms]	Solving problem 2 [ms]
A	399563	0	314452
B	399563	5447	18716
C	399563	1936	18716

TABLE II
MEMORY PERFORMANCE.

Cell size [cm]	Number of voxel in each cell	Memory [kb]
10	1	264
50	125	132
100	1000	44
170	4913	32

We conducted 2 tests in the generated model to illustrate the efficiency of the our planner in terms of time and memory.

1) Time efficiency: In this scenario, we implement 3 types of tests to evaluate the efficiency of our algorithm in terms of time. The goal is measuring the time of planning motion between an initial and goal configurations in the 2 models.

In test A, we did not use our algorithm and the roadmap was erased after planning motion in the first environment. So, the time of planning a motion between 2 configurations was measured separately in the 2 models. In test B, we kept the roadmap after solving the motion planning problem in first model. However, after loading the second model, we validated all the nodes and edges of the roadmap with the new obstacles in the second model.

Finally, in test C, we used our approach to optimize the time of calculation. We used our approach and the planner validated locally just some of the nodes and edges of the roadmap. The size of cells in the last types of tests was set to 1.7m $\times$ 1.7m $\times$ 1.7m and clearly each cell is composed of 4913 voxels. Table I compares the average of calculation time in each stage of the conducted tests.

2) Memory efficiency: To illustrate the efficiency of our approach regarding the size of cells and memory usage, we conducted 4 types of tests for solving a same problem in the generated models. Exactly the same as last section, the first model of the environment was loaded in the planner. Then, the planner generated a roadmap and found a path between initial and final configurations. The generated roadmap was kept and modified after loading the second model of environment based on our approach. In the first types of tests, we used the smallest possible size for each cells. In this type of tests, the size of each cell was 0.1m $\times$ 0.1m $\times$ 0.1m and each cells is composed of just one voxel. We increased the size of cells in second, third and fourth types. Ultimately, in the fourth types of tests, the size of each cells was 1.7m $\times$ 1.7m $\times$ 1.7m and each cells composed of 4913 voxels. Table II compares the effect of cell size on memory usage in our approach.

Based on the results, it is clear that by using a bigger cells,

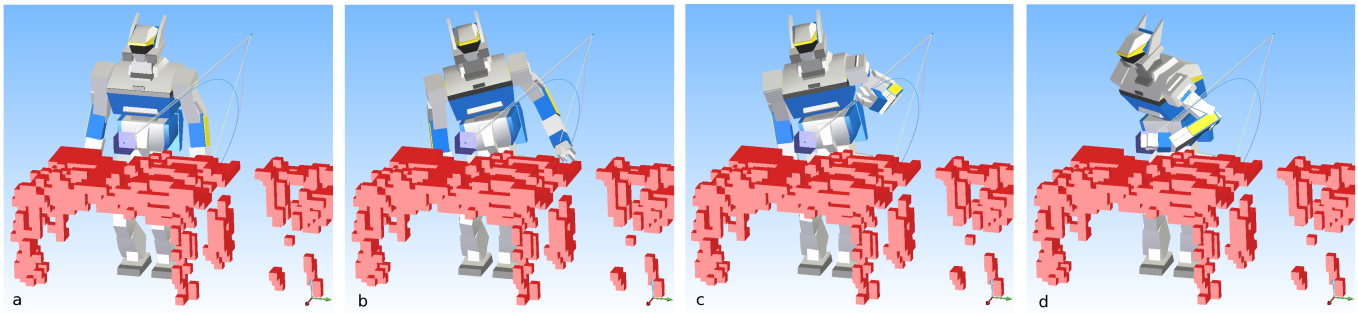

Fig. 8. HRP2 plans a whole body collision free motion to grasp an object in a generates 3D model. An occupancy model of a table which is generated by vision is used in motion planning.

we can save memory usage. Especially, in case of working with a robot with a high degree of freedom, cell size plays an important roll in memory usage. On other hand, as the size of cells increases, the number of nodes and edges in each cell increased. So, the planner consumes more time to validates the existing components(nodes and edges) of a modified cell.

V. Conclusions

We have presented an approach to deal with motion planning in unknown environments. In stage of environment modeling, we generated a 3D occupancy grid model by using stereo vision data. For motion planning in the changing environment, we extended an existing model and we improved its limitations in terms of memory. On the other hand, combining motion planing problems with a standard occupancy grid modeling of environment by stereo vision enable us to conduct an experiment in a real model of environment.

Moreover, the results illustrate the robustness of the approach in interacting with a large environment with thousands of voxels in term of modeling as well as updating roadmap based on our cell decomposition.

We believe that these approaches in terms of integrating the modeling and motion planning would be a step toward whole body path planning in a real environment for a humanoid robot.

As the perspective, to continue our work, we are going to couple our technique with path planning algorithms that produce whole-body motions such as grasping objects in 3D environment. As a primary step, the generated 3D model is coupled with a recent algorithms of local collision avoidance [6]. This integration enable HRP2 to plan a whole body motion in the generated 3D model of the environment. Figure 8 illustrates HRP2 who plan a collision free motion in the generated 3D model of environment to grasp an object on a table.

Furthermore, we are going to combine our algorithms with visual SLAM in order to make HRP2 able to explore large environments by itself.

References

[1] A. Bryson and Y. Ho. Applied Optimal Control. Waltham, 1969.

[2] D. Cole and P. Newman. Using laser range data for 3d slam in outdoor environments. In *IEEE International Conference on Robotics and Automation*, pages 1556–1563, Florida, May 2006.

[3] A. Davison, I. Reid, N. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE Transaction an pattern analysis and machine intelligence*, 29(6), June 2007.

[4] A. Elfes. *Occupancy Grids: A Probabilistic Framework for Robot Perception and Navigation*. PhD thesis, Carnegie Mellon University, 1989.

[5] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, 1989.

[6] F.Kanehiro, F.Lamiraux, O.Kanoun, E.Yoshida, and J.-P. Laumond. A Local Collision Avoidance Method for Non-strictly Convex Objects. In *Proceedings of 2008 Robotics: Science and Systems(submitted)*, 2008.

[7] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, 2, 2003.

[8] L. Kavraki, P. Svestka, J. Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high-dimensionalconfiguration spaces. *Robotics and Automation, IEEE Transactions on*, 12(4):566–580, 1996.

[9] S. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[10] S. LaValle and J. Kuffner Jr. Randomized Kinodynamic Planning. *The International Journal of Robotics Research*, 20(5):378, 2001.

[11] T. Lemaire, S. Lacroix, and J. Solá. A practical 3d bearing-only slam algorithm. Edmonton, Canada, Aug. 2005. IEEE/RSJ.

[12] P. Leven and S. Hutchinson. Toward Real-Time Path Planning in Changing Environments. *Algorithmic and Computational Robotics: New Directions: the Fourth Workshop on the Algorithmic Foundations of Robotics*, 2001.

[13] H. Liu, X. Deng, H. Zha, and D. Ding. A Path Planner in Changing Environments by Using WC Nodes Mapping Coupled with Lazy Edges Evaluation. *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 4078–4083, 2006.

[14] H. Moravec and A. Elfes. High resolution maps from wide angle sonar. *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, 2, 1985.

[15] N. Nilsson. *Principles of Artificial Intelligence*. Springer, 1982.

[16] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Probabilistic algorithms and the interactive museum tour-guide robot minerva. *IJRR*, 19(11):pp 972–999, 2000.

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

[18] J. van den Berg, D. Ferguson, and J. Kuffner. Anytime path planning and replanning in dynamic environments. *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2366–2371, 2006.

[19] E. Yoshida, O. Kanoun, C. Esteves, and J. Laumond. Task-driven Support Polygon Reshaping for Humanoids. *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 208–213, 2006.

[20] E. Yoshida, A. Mallet, F. L. O. Kanoun, O. Stasse, M. Poirier, P. F. Dominey, J.-P. Laumond, and K. Yokoi. Give me the purple ball – he said to hrp-2 n.14. In *IEEE RAS/RSJ Conference on Humanoids Robots, Pittsburg, USA, 30 Nov. - 2 Dec.*, page Oral presentation, 2007.