Book Title
Book Editors
IOS Press, 2003

Vision-based Virtual Information and Semi-autonomous Behaviours for a Humanoid Robot

Olivier* Stasse [a,1], Jean Semere [a] Neo Ee Sian [a] Takashi Yoshimi [b] and Kazuhito Yokoi* [a]

[a] *AIST/ISRI-CNRS/STIC, Joint Robotics Laboratory*
[b] *Intelligent Systems Research Institute*

Keywords. Semi-autonomous behaviours, vision based, humanoid robot

1. Introduction

This paper considers the problem of a humanoid robot evolving in an unknown environment semi-autonomously through a high-level human supervision. In this case the problems are to build a representation of the world, perceive objects, plan a path to explore the environment, and move accordingly. The sensing system consider here is a stereoscopic vision system.

Compare to wheeled robot where 2D representation [1] is often enough for evolving inside an environment, humanoid robot needs a 3D representation of the world. Indeed, they can step on and over obstacles [2] [3], go through narrow spaces and crawl [4]. Therefore in order to plan motions a 3D knowledge of the environment is needed. Gutmann in [5] suggest the construction of a 3D grid map to update a 2.5D navigation map on which an A* algorithm is performed. Kagami [6] used an online dense local 3D world reconstruction by merging depth maps, self localise using the RANSAC algorithm, and planned a trajectory on a projected 2.5 D map. In both case, interestingly, they are no justification on the choice of the destination.

One possibility is to use the unknown part of the environment to plan the next position of the robots as proposed by Sujan in [7]. This is especially useful for a full coverage of a closed area. But the most likely application is to be used in conjunction with a remote operator. In this case, a human is able to analyse the scene and to *guess* that a path is possible for the robot. Our goal is to enhance the information available for the remote operator, and to develop the robot's autonomy for an efficient collaborative work. In this

[1] Correspondence to: Olivier Stasse, Joint Japanese-French Robotics Laboratory,
Intelligent Systems Research Institute,
Centre Nationale de la Recherche Scientique
National Institute of Advanced Industrial Science and Technology,
AIST-Central 2 Umezono 1-1-1.
Tel.: +81 29 861 9244; Fax: +81 29 861 3443; E-mail: olivier.stasse@aist.go.jp.

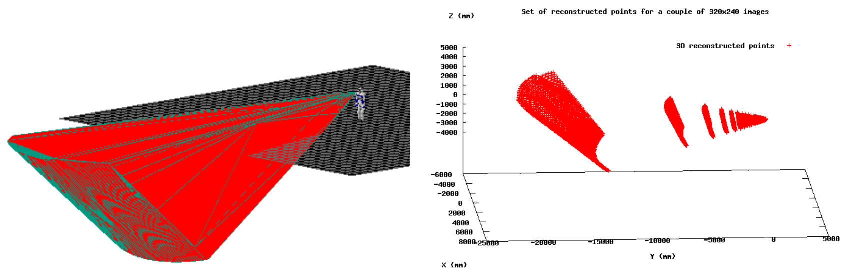

Figure 1. (left) A polytope describing the full range of a perception for a HRP2's stereoscopic system. (right) The set of reconstructed points for the polytope

context, a high level of local autonomy is very important because humanoid robots are almost impossible to manipulate at a low level: stability, motion generation are among the keys concepts which have to be solved for a proper teleoperation. Those points have been addressed in a previous work in our group by Sian in [8].

For a multiple purposes platform such as a humanoid, it is also important to be able to perceive precisely objects in order to localise and grasp them [8]. In order to meet both challenges, i.e. long range perception for map-building and short range for object perception several humanoids robots such as COG from MIT, DB from ATR, and several HRP-2 (DHRC, NAIST, JRL) have cameras with different field of views. In both situations the precision and the range of the field of view are important and should be presented to the user in a coherent manner. In this paper, we consider that a stereoscopic system is characterised by a polytope and a resolution function obtained by covariance analysis. The polytope representation gives a geometrical representation of the field of view and the depth-of-field. It is shown how those two information can be used to enhance the information presented to a user: by showing the 3D space where the robot can see in 3D, by showing the perception error of the 3D sensing, and by displaying the unknown space.

We also present a simple autonomous behaviour where the robot plan a trajectory for which the destination is specified by the user. In this context, we do not use the full capabilities of the robot, but follows the line of Gutmann [5], i.e. the configuration space is split in simple motions dynamically stable. The work presented here differs by the use of the walking pattern proposed by Kajita in [9]. The goal of this planner is to be fast, in order to be re-plan on-line if an obstacle appear in front of the robot. It can also be used as a high level planner to be used in conjunction with more sophisticated approaches. The remainder of this paper is organised as follow: The first section described the method to specify a stereoscopic system in order to display virtual informations to a remote operator. The second section described how the 3D representation of the world is build, and how unknown space is presented to the user. The semi-autonomous path planning behaviour is described in section 4, and is followed by the conclusion.

2. Characterisation of a stereoscopic system

2.1. Introduction

We consider a stereoscopic rig for which the two cameras $\mathcal{C}_r$ and $\mathcal{C}_l$ are model by two projective matrices $\mathbf{P}_r$ and $\mathbf{P}_l$. A 3D point noted $\mathbf{Q}$ is projected by those two matrices on two pixels $\mathbf{q}_r$ and $\mathbf{q}_l$. Considering that $\mathbf{q}_r$ and $\mathbf{q}_l$ are *measured*, and thus noisy, 3D triangulation consists in an estimation of $\{\mathbf{Q}_s\}$ the space where lies $\mathbf{Q}$. We are interested in representing the 3D space $\mathcal{P}$ where such estimation is possible. We propose to use a polytope representation such as

$$\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^4 : \mathbf{A}\mathbf{x} \geq 0\} \tag{1}$$

Considering that the stereoscopic rig has a position $\mathbf{C}$ and an orientation $\mathbf{R}$, let us define $\mathbf{H}$ is the homogeneous matrix representing the rigid transformation $\{\mathbf{R}, \mathbf{C}\}$. It is then very straightforward to test if a 3D point $\mathbf{Q}$ of the space might be perceived by the 3D sensor. Indeed if $\mathbf{AHQ} \geq 0$ then $\mathbf{Q}$ can be seen by the 3D sensor. However in order to make it computationally efficient, it is necessary to find a minimal description of $\mathcal{P}$ i.e. a minimal $\mathbf{A}$. This section describes how to compute such a minimal description. It also shows how to compute a polytope for a given resolution.

2.2. Building the polytope

To build the polytope we consider all the possible combinations between $\mathbf{q}_r$ and $\mathbf{q}_l$. Based on each pair, the corresponding $\mathbf{Q}$ is computed using the DLT algorithm has described in [10]. We recall the formulation of this system as it is used in the following of the paper:

$$\mathbf{M} = \begin{bmatrix} \frac{1}{\mathbf{p}_1^3\mathbf{Q}}(x_1\mathbf{p}_1^3 - \mathbf{p}_1^1)\mathbf{Q} \\ \frac{1}{\mathbf{p}_1^3\mathbf{Q}}(y_1\mathbf{p}_1^3 - \mathbf{p}_1^2)\mathbf{Q} \\ \frac{1}{\mathbf{p}_2^3\mathbf{Q}}(x_2\mathbf{p}_2^3 - \mathbf{p}_2^1)\mathbf{Q} \\ \frac{1}{\mathbf{p}_2^3\mathbf{Q}}(y_2\mathbf{p}_2^3 - \mathbf{p}_2^2)\mathbf{Q} \end{bmatrix} = \mathbf{WAQ} \tag{2}$$

where $\mathbf{p}_k^i$ the i-th row of the projective matrix $\mathbf{P}_k$, and $\mathbf{W}$ has only diagonal element given by the following vector $\{\frac{1}{\mathbf{p}_1^3\mathbf{Q}}, \frac{1}{\mathbf{p}_1^3\mathbf{Q}}, \frac{1}{\mathbf{p}_2^3\mathbf{Q}}, \frac{1}{\mathbf{p}_2^3\mathbf{Q}}\}$. $\mathbf{Q}$ is found by computing the least-square estimate of $(\mathbf{AX})^T(\mathbf{AX})$. This give us the set of possible 3D points which can be perceived by the stereoscopic system. The convex hull of this set is found using the Quickhull algorithm [11] implemented by the qhull library. The main problem with this algorithm is the degeneracy in the output of its result. As several points belongs only to one facet of the convex hull, the resulting polytope is not optimal. In order to find the minimal representation we used an implementation of the double description method provided by Fukuda [12].

2.3. Error of the 3D triangulation

For each 3D point the position uncertainty of $\mathbf{Q}$ is computed using covariance back-propagation as described in [10]. More precisely, the covariance matrix $\Sigma_{\mathbf{Q}}$ for $\mathbf{Q}$ is given by:

$$\Sigma_{\mathbf{Q}} = (\mathbf{J}^T \Sigma_{\mathbf{X}}^{-1} \mathbf{J})^+ \tag{3}$$

where $\mathbf{M}^+$ is the Moore-Penrose generalised inverse of matrix $\mathbf{M}$. The Jacobian $\mathbf{J}$ is given by

$$\mathbf{J} = \frac{\delta \mathbf{WAQ}}{\delta \mathbf{Q}} = \mathbf{WA} + \frac{\delta \mathbf{W}}{\delta \mathbf{Q}} \mathbf{AQ} \tag{4}$$

with $\frac{\delta \mathbf{W}}{\delta \mathbf{Q}}$ is a 4×4 matrix, and for which the diagonals elements are $\{-\frac{1}{\mathbf{p}_1^3 \mathbf{Q}^2}, -\frac{1}{\mathbf{p}_1^3 \mathbf{Q}^2}, -\frac{1}{\mathbf{p}_2^3 \mathbf{Q}^2}, -\frac{1}{\mathbf{p}_2^3 \mathbf{Q}^2}$, and all the other elements are 0.

Practically $\frac{\delta \mathbf{W}}{\delta \mathbf{Q}}$ is not computed because it is multiplied by $\mathbf{AQ}$ which should be equal to zero. As this reconstruction is done up to scale, also $\mathbf{J}$ has 4 rows, its rank is 3. This singularity is easily handled using a SVD decomposition. We call the resulting ellipsoid an *uncertain point*. This is used to update the occupancy grid.

2.4. Minimal representation

The minimal representation of the polytope consists in finding the matrix $\mathbf{A}$ with non redundancy. The results provided by the brute-force approach described in the previous paragraph is a polyhedra. The first step is to compute the half-space representation of this polyhedra. A half-space representation of a polyhedra is the name used in computational geometry for the equivalent set of inequalities.

Fukuda and others proposed to use the double description method to compute such minimal description. Other methods exist to compute the convex hull and the half-space representation of a polytope. As indicated previously we used the QuickHull algorithm to compute the convex hull of the polytope. Also this algorithm does not provide the minimal representation, it is very quick to use. It also provides a Delaunay triangulation useful to draw the polytope in a GUI. Other solutions exists such as: **lrs** [13] but this software accept only rational entries.

2.5. Results and comparison

Using the method described above, we computed the set of reconstructed points for two cameras of our humanoid each giving a 340x480 pixels image. The result is depicted in figure 1-right. It is very straightforward to notice that the precision of the reconstruction is decreasing at points far from the center of the cameras. Figure 2 depicts a scene where the humanoid robot is looking at close range to a complex object. The raw reconstructed surface is displayed in an OpenGL windows depicted in figure 3-left. The corresponding image with uncertainty is depicted in figure 3-right. We used a similar approach based on interval analysis [14] to design a pre-attentive behaviour where the robot autonomously find an object of interest inside the environment.

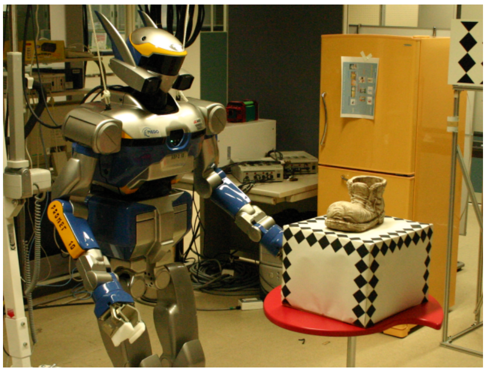

Figure 2. Unknown object to analyse

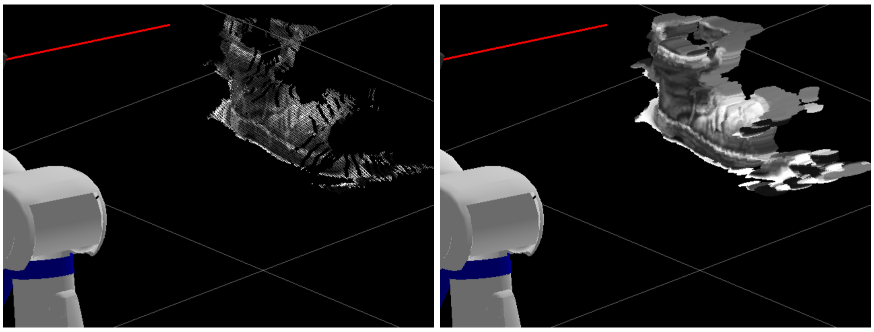

Figure 3. (left) Reconstructed surface without uncertainty ellipsoids (right) Reconstructed surface with uncertainty ellipsoids

The polytope representation computed by qhull gives 250 points. We used the Delaunay triangulation functionality to generate the surfaces needed for display as in figure 1-right. The use of cdd reduced the number of points to 132. Also it is interesting for display purposes, this representation is still too high to be used in vision computation. Indeed generally the two cameras projective matrices only are used to model a stereoscopic system. Indeed it is possible to test if a 3D point $\mathbf{Q}$ is seen by the stereoscopic system at a given location using the projective matrix of each cameras $\mathbf{P}_l$ and $\mathbf{P}_r$. If the projected points $\mathbf{q}_r$ and $\mathbf{q}_l$ are inside the image then the 3D point can be perceived in 3D. Thus this test can be done by 4 products of matrices (2 for changing the camera location, and 2 for projecting the point) and 8 conditions. Therefore any method dealing with such kind of problem must add some qualitative information. In this situation, the projective matrix does not give the depth limitation inherent to the discretization of the space. However, during disparity computation it is usual to put some limitation in the depth-of-field to limit the noise due to the background. This can be easily represented by putting a constraint on the distance of the point to the camera system.

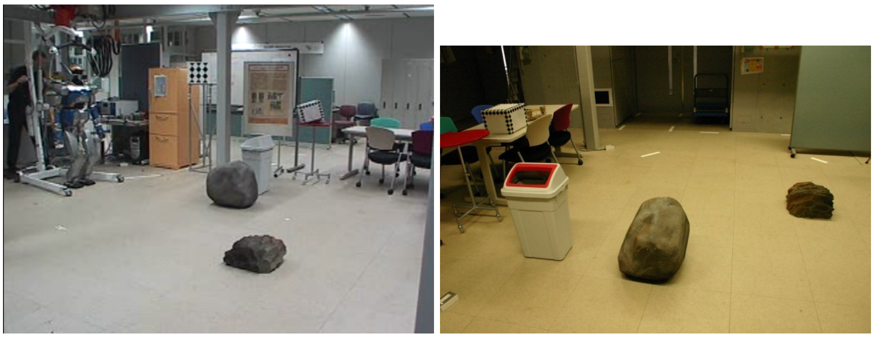

Figure 4. (left) Original scene: situation of the robot. (left) Original scene: view of the robot.

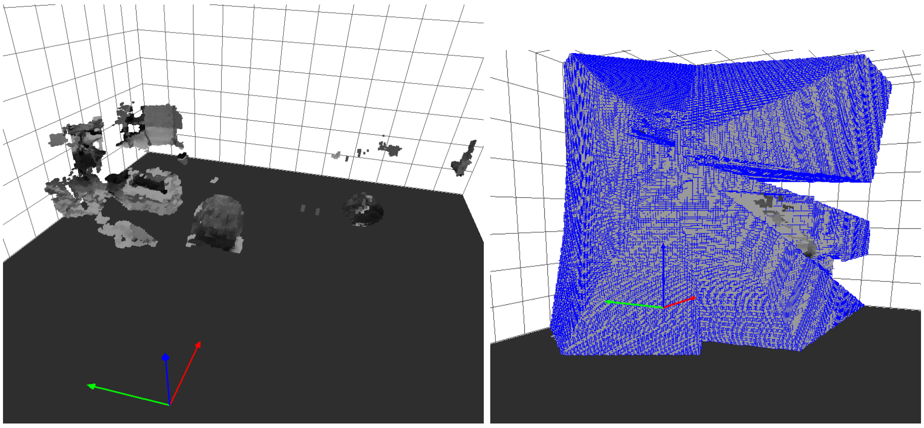

Figure 5. (left) Reconstruction of the scene in figure 4 (right) Space explored after 5 views.

3. Occupancy grid and explored space

The world is represented using an octree. This data structure, originally proposed by [15], recursively divided the 3D space. It stores the information detected by the stereoscopic system, the unseen space, and the empty space. The cells of the octree are updated following the usual occupancy grid algorithm as described in [1]. An example of such occupancy grid is given in figure 5-left. An information which is important for a remote operator and to develop an autonomous behaviour is the *explored space*. In both case, it can be used to decide where to look next. Thus for each view, each point $\mathbf{Q}$ of the occupancy grid has to be tested to decide if it has been seen or not. This is done using the projective matrices of the cameras. If the coordinates of the projection $\mathbf{q}_l = \mathbf{P}_l\mathbf{Q}$ are inside the images, the point is tested against the range-image. The range-image follows the coordinates of the left image and gives the corresponding 3D point $\mathbf{RI}(\mathbf{q}_l)$. If the $\mathbf{Q}$ is closer then it means that it does not exists anymore and its probability should be decrease. If $\mathbf{Q}$ is more distant then it cannot be perceived and stay unknown. The last case is where it correspond to the point perceived i.e. when $\mathbf{Q} = \mathbf{RI}(\mathbf{q}_l)$. This operation has been tested in the context of octree, and append to be quite time consuming (20 seconds in average for one view) partly because its implementation has not been optimised, and because several thousand of points are processed each time. Thus for unknown space, 3D occupancy grid as an array are used.

Figure 5-right depicts in blue the space explored by the humanoid-robot after 5 views. It can be seen that the perception of some elements in the scene hide some parts of the environment. Other views will be needed to cover those hidden places. It is possible to implement automatic Next Best View algorithm [7]. However as the function to minimise completely depends on the environment, the current common practice is too discretize the environment and chose the view which will minimise the motion to perform, and maximise the covered unknown area.

4. Path planning

This part presents a semi-autonomous behavior based on the visual information presented previously which allow to a remote operator to control the humanoid in a high-level manner. In this behavior, the aim was to have a very fast planner in order to be able to replanify between 2 images the path of the robot. This is specially useful for obstacle avoidance in dynamic environment. The main challenges are to planify the foot position and to take into account the subsequent trajectory of the waist. Kuffner in [16] used a Randomised Rapid Tree to generate a path and a dynamic filter to realize a stable trajectory. Chesnutt in [17] perform foot planning using an automata describing all the possible transition from one foot position to the other foot position while the robot is stable.

As we are using a HRP-2 humanoid robot with the initial foot planner and the stabiliser described in [18], the planner described here has been tailored for this setup. The approach described here is inspired from [5]. However, we do not consider in this paper 3D information. The main reason is that we do believe that in some situations, specifically where complex motion or even contacts with the environment have to be considered, a more complex planner is needed, and thus outside the scope of this paper. The main point of this planner is to provide to the exploration process a feasible path in real-time.

4.1. Pattern generator

The pattern generator provided by [18] allows three kinds of functionalities: Translational motion with change of orientation, motion on an arc, and next step position. Translational motion takes the targeted position and orientation related to the current position. Motion on an arc takes the center of the arc, and the targeted position.

In order to decrease the complexity of the search, following [5], the set of possible motions is split into 5 directions: forward, right, left, forward-right, forward-left. Because the cost of a path going backward is worst than a path which goes straight all the time, there is no backward motion. For right, left, forward-right and forward-left, they are several ways to realize the motion depending on the starting direction, and the final direction. Here we consider in general two different *modes* of walking: forward and sideway. Thus for each node of the grid, we consider a *configuration* being a pair of one mode and one direction. For each configuration a mask represents the 2D projection of the maximum space to be used by the robot in order to realize the motion. Those masks

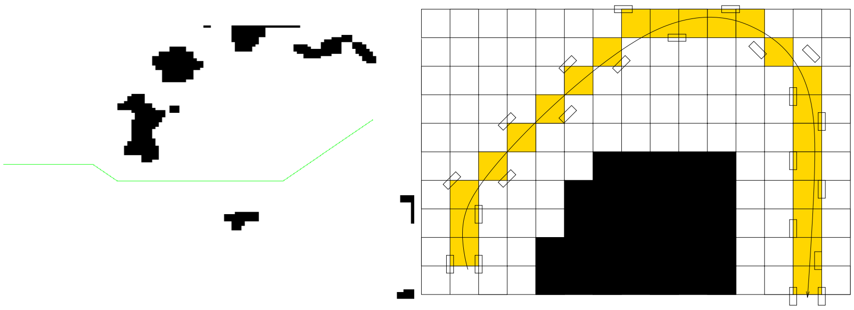

Figure 6. (left) Generated path by the planner described in the text. (right) Sequence of actions for the footstep positioning method

have been build by simulation and represent the boundary in which the motion can be realized.

*4.2. A**

The algorithm used to explore the 2D grid is A* with 4 kinds of heuristics. Three are very usual : a cost-from-start heuristic, a cost-to-goal heuristic, and an obstacle one. The fourth one is based on the path shape. It allows to favor the creation of straight trajectories.

4.2.1. Path execution

We developed a path execution system which either uses the basic functionalities of the original HRP-2, or which can generate foot positioning once the path is found. Because during the path generation the robot space considered is an upper bound, and because this is a 2D map, the steps can be positioned *after* finding the path. An extension of this planner would be to consider 3D and taking into account for instance stepping over obstacle motions. Once the steps have been planned as depicted in figure 6-right, a stack handler send the next step according to the delay imposed by the preview control described in [9]. This delay is the time needed by the robot to stop in a smooth manner, which is of two steps in HRP-2.

4.3. Experiences

For our experience a HRP-2 humanoid robot has been used. It has 30 DOFs, its height is 1m64, and weight 54 kilogrammes [19]. It has two CPU-board each embedding a Pentium III 1 Ghz. One is dedicated to control, while the other one is used for vision. The head of the robot contains 4 cameras. One has a 90 degrees field of view while each of the three others has a 25 degrees field of view. The robot is put in the situation depicted in figure 4-left. It build its own representation of the world by taking several views of the environment. The space coverage has been design using the 3D polytope associate with the 3D sensor system. More precisely, the field of view of the 3D sensor system has been found by searching the points with the minimum and maximum angular coordinates relative to the origin of the polytope frame. More precisely here, this origin is the center

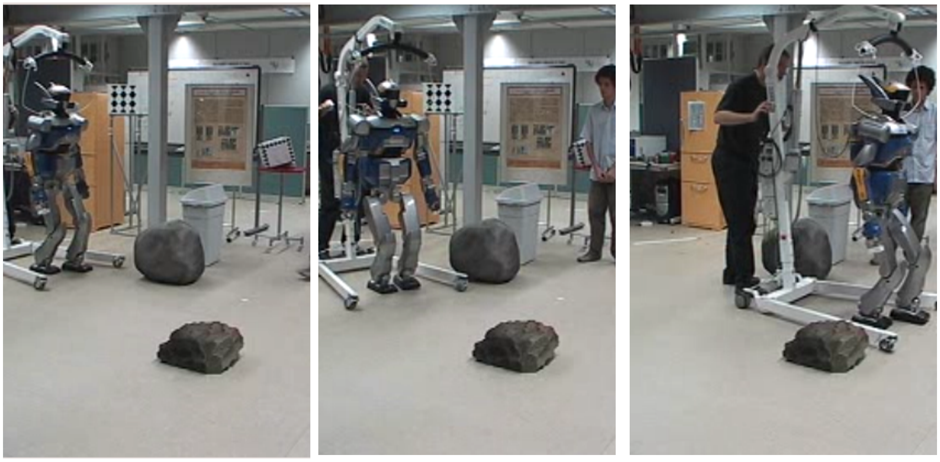

Figure 7. Execution of the generated path

of the baseline. Numerically we found that our robot has a stereoscopic field of view of 33^o degrees in the horizontal plan, and of 33 degrees in the sagital plan. The limitation of the robots head are $[-45; 45]$ for the horizontal plan and $[-30; 45]$ for the sagital plan. Therefore for a complete coverage of the space that can be seen by moving only the head, 9 different viewpoints are necessary. The result put inside the occupancy grid described previously is depicted in figure 4. From the occupancy grid, a 2D map is extracted by projecting all the obstacles on the floor. The remote operator choose the destination on the map represented in figure 6. The map, the starting point and the destination point are transmitted to the path planner and a path is generated as depicted in green in figure 6. The realization of the path is depicted in figure 7.

5. Conclusion

We have presented a set of tools which enhance the visual information reported to a remote operator. Those tools allow the operator to see where the 3D vision sensor of a humanoid robot can be applied. It allows also to visualise the precision of the 3D vision sensor by displaying the volume of integration inside an OpenGL window. In order also to improve the efficiency of the remote control of a humanoid robot, we have presented an autonomous behaviour where the robot create its own representation of a 3D world, and in which it can find a path.

References

[1] S. Thrun, "Learning occupancy grid maps with forward sensor models," *Autonomous Robots*, vol. 15, pp. 111–127, 2003.

[2] K. Okada, T. Ogura, A. Haneda, and M. Inaba, "Autonomous 3d walking system for a humanoid robot based on visual step recognition and 3d foot step planner," in *Proc. IEEE Int. Conf. on Robotics and Automations*, 2005, pp. 625–630.

[3] Y. Guan, K. Yokoi, and K. Tanie, "Feasibility: Can humanoid robots overcome given obstacles?" in *Proc. IEEE Int. Conf. on Robotics and Automations*, May 2005, pp. 1066–1071.

[4] F. Kanehiro, T. Yoshimi, S. Kajita, M. Morisawa, K. Fujiwara, K. Harada, K. Kaneko, H. Hirukawa, and F. Tomita, "Whole body locomotion planning of humanoid robots based on a 3d grid mapm," in *Proc. IEEE Int. Conf. on Robotics and Automations*, 2005, pp. 1084–1090.

[5] J.-S. Gutmann, M. Fukuchi, and M. Fujita, "Real-time path planning for humanoid robot navigation," in *International Joint Conference on Artifical Intelligence*, July 2005.

[6] S. Kagami, Y. Takaoka, Y. Kida, K. Nishiwaki, and T. Kanade, "Online dense local 3d world reconstruction from stereo image sequences," in *Int. IEEE Conf. on Intelligent Robots and Systems*, 2005, pp. 2999–3004.

[7] V. A. Sujan and S. Dubowsky, "Efficient information-based visual robotic mapping in unstructured environments," *The International Journal of Robotics Research*, vol. 4, no. 4, pp. 275–293, April 2005.

[8] N. E. Sian, K. Yokoi, S. Kajita, F. Kanehiro, and K. Tanie, "Whole body teleoperation of a humanoid robot -a method of integrating operator's intention and robot's autonomy." in *International Conference on Robotics and Automation*, 2003, pp. 1613–1619.

[9] 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," in *IEEE International Conference on Robotics and Automation, ICRA '03.*, vol. 2, sept 2003, pp. 1620–1626.

[10] R. Hartley and A. Zisserman, *Multiple View Geometry*. Cambridge University Press, 2003.

[11] C. B. Barber, D. P. Bobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software*, vol. 4, no. 22, pp. 469–483, December 1996, http://www.qhull.org.

[12] K. Fukuda and A. Prodon, "Double description method revisited," *Lecture Notes In Computer Science*, vol. 1120, pp. 91–111, 1995, selected papers from the 8th Franco-Japanese and 4th Franco-Chinese Conference on Combinatorics and Computer Science.

[13] D. Avis, "lrs: A revised implementation of the reverse search vertex enumeration algorithm," *Polytopes - Combinatorics and Computation, Ed. G. Kalai and G. Ziegler, Birkhauser-Verlag*, pp. 177–198, 2000.

[14] O. Stasse, B. Telle, and K. Yokoi, "3d segmentation using interval analysis and pre-attentive behavior for a humanoid robot," in *IEEE Int. Conf. on Robotics and Biomimetics*, 2005, pp. 284–289.

[15] C. I. Connolly, "The determination of next best views," in *International Conference on Robotics and Automation*, 1985, p. 432.

[16] J. J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, "Dynamically-stable motion planning for humanoid robots," *Autonomous Robots, Kluwer Academics publisher*, vol. 12, no. 1, pp. 105–118, January 2002.

[17] J. Chesnutt, M. Lau, G. Cheung, J. Kuffner, J. Hodgins, and T. Kanade, "Footstep planning for the honda asimo humanoid," in *International Conference on Robotics and Automation, Barcelona, Spain*, April 2005, pp. 631–636.

[18] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3d linear inverted pendulum mode : A simple modeling of a biped walking pattern generation," in *International Conference on Intelligent Robots and Systems, Maui, Hawaii, Usa*, November 2001, pp. 239–246.

[19] K.Kaneko, F.Kanehiro, S.Kajita, H.Hirukawa, T.Kawasaki, M.Hirata, K.Akachi, and T.Isozumi, "Humanoid robot hrp-2," in *Proceedings of the 2004 IEEE International Conference on Robotics & Automation*, vol. 2, 2004, pp. 1083–1090.