A Two-Steps Next-Best-View Algorithm for Autonomous 3D Object Modeling by a Humanoid Robot

Torea Foissotte [1,2], Olivier Stasse [2], Adrien Escande [2], Pierre-Brice Wieber [3], Abderrahmane Kheddar [1,2]

[1] *CNRS-LIRMM, France*
[2] *CNRS/AIST JRL, Japan*
[3] *INRIA, France*

***Abstract*— A novel approach is presented which aims at building autonomously visual models of unknown objects, using a humanoid robot. Although good methods have been proposed for the specific problem of the next-best-view during the modeling and the recognition process; our approach differs as it takes advantage of humanoid specificities in terms of embedded vision sensor and redundant motion capabilities.**

In a previous work, another approach to this specific problem was presented which relies on a derivable formulation of the visual evaluation in order to integrate it with our posture generation method. However to get rid of some limitations we propose a new method, formulated using two steps: (i) an optimization algorithm without derivatives is used to find a camera pose which maximize the amount of unknown data visible, and (ii) a whole robot posture is generated by using a different optimization method where the computed camera pose is set as a constraint on the robot head.

I. Introduction

A. Context of the work

In this work, we're interested in object modeling with the purpose of allowing their robust detection and recognition. Three main problems need to be solved to ensure a successful modeling process: (i) object/environment distinction, (ii) object features processing and memorizing, and (iii) object manipulation or sensor movement so as to model the whole surface. Currently we are simplifying the first problem by putting the object on a known table in front of the robot. For the second problem, we take advantage of results from a previous work [1] using an occupancy grid and disparity maps obtained by stereo vision, coupled with SIFT [2] landmarks detection. This paper deals more particularly with the third problem by proposing an algorithm to generate successive postures, for a humanoid robot, in order to build the model of the object. For now, the manipulation of the object is not addressed.

B. Overview of related work

The planning of sensor positions in order to create an 3D model of an unknown object has been adressed specifically in the Next-Best-View (NBV) research field which is surveyed in these two autorithative papers: [3] and [4]. The most usual assumptions are that the depth range image is dense and accurate by using laser scanners or structured lighting, and that the camera position and orientation is correctly set and measured relatively to the object position and orientation. The object to analyze is also considered to be inside a sphere ([5], [6]) or on a turntable ([7], [8], [9]), i.e the sensor positioning space complexity to evaluate is reduced since its distance from the object center is fixed and its orientation is set toward the object center. The main aim of such works is to get an accurate 3D reconstruction of an object while reducing the number of viewpoints required.

In works related to object recognition [10], the problem of autonomously acquiring a model of the object is usually avoided as the modeling part is based on views taken manually by a human.

C. Contribution

In order to select a NBV pose for the humanoid robot, the amount of unknown data that is to be perceived needs to be quantified. Following the works of [5] and [6], our approach uses an occupancy grid and the space carving algorithm for this purpose. The object model is composed of perceived (known) voxels and occluded (unknown) voxels, and is updated using disparity maps obtained by stereo vision. Our algorithm is based on the evaluation of unknown data visible from a specific robot pose.

Though our modeling process also requires a NBV solution, it appears that working hypotheses are quite specific for a humanoid robot and thus our work differs in few important issues:

1) the limits of the sensor pose are constrained as it is embedded in a humanoid robot. Constraints such as self-collisions, collisions with the environment, joint limits, feet on the floor, and stability must be taken into account. We also need another constraint that keeps some landmarks visible from the cameras so as to correct positioning errors,
2) the sensor's result positions need not being further constrained to some precomputed discrete positions on a sphere surface, and its viewing direction is not forced toward a sphere center. Thus the algorithm can be used to model objects of different sizes and with more complex shapes,
3) an accurate 3D model of the object is not required. Our goal is to get a set of visual features around the object to allow its effective detection and recognition.

In [11], the object modeling was performed by generating postures with the robot head pose set as a constraint given by a human supervisor. In [12], A first attempt to complete this work by using visual cues to guide the modeling process automatically was proposed by using a formulation which can be directly integrated into our posture generator. Section II summarizes this previous approach with the main results and problems associated. Section III details our latest solution to generate a posture by using two distinct complementary steps. Section IV presents the test results for the new approach and section V concludes this paper.

II. $\mathcal{C}^1$ Function for Unknown quantification

A. Posture Generation

Our Posture Generator (PG), proposed as part of the work in [13] and [11], relies on FSQP, a gradient-based optimization method, to give a posture that minimizes an objective function while solving given constraints. In a previous work [12], we were interested in finding a $\mathcal{C}^1$ function for the quantification of unknown so as to include it in the PG.

B. Objective Function

A differentiable function to evaluate visual information was designed to be used as the objective function to minimize in the PG. In this approach, a voxel is considered as a sphere and thus its projection on the resulting image can be expressed as a 2D Gaussian function. The complete formulation of the objective function and its gradient have been described in [12].

C. $\mathcal{C}^1$ Function tests

An efficient and relatively fast convergence of the optimization method in order to generate a robot posture could not be achieved during our tests due to the presence of many local optima. These come from variations of low amplitude but high frequency in the function. This results in cases where the optimization algorithm cannot converge, or cases where it takes between 30 minutes and few hours to generate a single posture. We supposed that the problem resulted from our formulation as the function is sampled using the result image pixels location, and developed another approach to test this hypothesis.

D. Voxels as polygons

In this approach, voxels are represented by cubes, and voxels' faces projection on the camera image plane are represented by polygons which area can be computed analytically. Using such formulation, the amount of unknown visible is equal to the visible polygons' surface of unknown voxels. As this formulation does not rely on a threshold function nor any sampling, problems due to discretization are not present.

The results of this approach have been compared with the $\mathcal{C}^1$ function and a simple point-based rendering of the voxels using OpenGL, in Fig. 1. We also included a sampled version of this polygon approach, where polygons are displayed on the camera image and the unknown area visible corresponds to the number of corresponding pixels displayed. Evaluation results with the 4 methods are displayed for a small translation of the camera in front of a single unknown voxel.

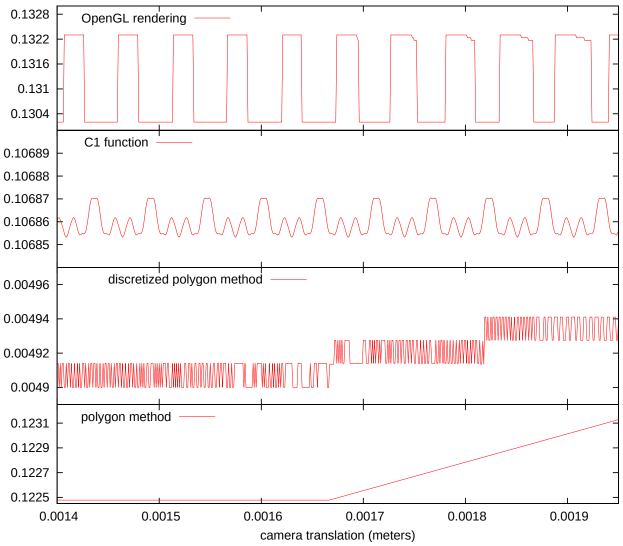

Fig. 1. Comparison of methods for the evaluation of an unknown voxel visibility relatively to the camera position

For our $\mathcal{C}^1$ function and the OpenGL method, the evaluation should be constant, as the distance between the camera and the voxel does not change, but oscillations appear, highlighting the problem with pixel-based approaches. The polygon approach gives results consistent with the expectation. Indeed the evaluation is constant when only one face of the cube occludes all others, then it increases linearly when a side face become also visible. The discretized polygon method would ideally stay constant then increase in a series of stair steps but the sampling introduces oscillations.

Though the main cause of limitations for our $\mathcal{C}^1$ function is understood, we could not find a way to modify the formulation in order to decrease the amount of local optima without increasing the computation time.

The polygon approach may look promising as an objective function, but the final formulation is changed depending on events and thus an approximation of its gradient is difficult to formulate. Moreover, as illustrated in the example in fig. 1, the gradient is not continuous everywhere.

III. Two steps NBV approach

In our particular problem, a proper formulation of an objective function for FSQP is difficult. In fact, traditional works in the NBV field reduce the problem's dimensionality and sample the configuration space in order to retrieve a solution in an acceptable amount of time without relying on the gradient.

To avoid previous problems encountered while taking into account the constraints related to the use of a humanoid, a novel solution to our Next-Best-View problem is introduced by decomposing it in two: first, find a camera position and orientation that maximizes the amount of unknown visible while solving specific constraints related to the robot head, then generate a posture for the robot using the PG.

We propose to solve the first step by using NEWUOA [14], a method that can find the minimum of a function by refining a quadratic approximation of it through a deterministic iterative sampling, and which can be used for non-differentiable functions. NEWUOA has the advantages of being fast, robust to noise, and allow us to keep the 6 degrees of freedom of the camera.

A. Evaluation of the camera pose

In this approach, the estimation of unknown data visible from a specific viewpoint can be computed by taking advantage of hardware acceleration, as a gradient is not required. Moreover oscillations of small amplitude have only a negligible influence on the convergence of NEWUOA. An OpenGL rendering of the occupancy grid was thus implemented by displaying non-empty voxels as cubes. The amount of unknown visible is then equal to the number of visible pixels belonging to "unknown" voxels.

B. Constraints on the camera pose

Though NEWUOA is supposed to be used for unconstrained optimization, some constraints on the camera pose need to be solved in order to be able to generate a posture with the PG from the resulting desired camera pose. The constraints on the camera position $\mathbf{C}$ and orientation $\mathbf{\Theta}c$ included in the evaluation function of the first step given to NEWUOA are:

$$\begin{cases} C_{zmin} \leq \mathbf{C}_z \leq C_{zmax} & (1) \\ d_{min} \leq d(\mathbf{C}, \mathbf{O}) & (2) \\ \Theta c_{xmin} \leq \mathbf{\Theta c}_x \leq \Theta c_{xmax} & (3) \\ \Theta c_{ymin} \leq \mathbf{\Theta c}_y \leq \Theta c_{ymax} & (4) \\ N_l \geq N_{lmin} & (5) \end{cases}$$

(1) limits the range of the camera height to what is accessible by the humanoid size and joints configuration. (2) imposes a minimum distance d_{min} between the robot camera and the closest non-empty voxel of the object $\mathbf{O}$. This corresponds to a requirement in order to generate the disparity map with the two cameras embedded in the robot head. There is no constraint on a maximum distance. (3) and (4) restricts the rotations on X and Y axes to ranges manually set according to the robot particularities. Finally (5) ensures that the number of landmarks currently visible N_l is greater than a chosen threshold N_{lmin}. By matching previous landmarks with those detected within the new viewpoint, it is possible to correct the odometry errors due to the movement of the humanoid and thus the position and orientation of the features detected all around the object, relatively to each other, can also be corrected.

C. Evaluation function formulation

In order to include the constraints into the function that NEWUOA evaluates, we formulate the interval constraints (1), (3) and (4), as:

$$K_v = (\alpha\, v - \mu)^p \tag{6}$$

where parameters α and μ are manually set to modulate, respectively, the interval center and width depending on the parameter v to constrain. v corresponds to the parameter $\mathbf{C}_z$, $\mathbf{\Theta c}_x$, or $\mathbf{\Theta c}_y$. p can be set to a large value so that the result is close to 0 inside the interval and increases quickly outside of it.

Following the same principle, the inequality constraint (2) related to the minimum distance between the camera and the object is formulated as:

$$K_d = \exp\left(\gamma\ (d_{min} - d(\mathbf{C}, \mathbf{O}))\right) \tag{7}$$

where γ parameter is set manually.

To test the landmark visibility constraint, we consider the number of pixels visible from voxels corresponding to each landmark. The surface visibility for a landmark *i* is computed relatively to its amount of pixels visible from the current viewpoint, pv_i, using a sigmoid function:

$$ls_i = \frac{1}{1 + exp\left(pmin_i - pv_i\right)} \tag{8}$$

The parameter $pmin_i$ is the minimum amount of pixels required to consider the landmark *i* visible, and its value depends on the original landmark size. We then compare the sum of all ls_i to an arbitrary defined threshold Nlm_{min}. When the threshold is reached, the constraint is formulated to encourage slightly the visibility of more landmarks:

$$K_l = -\eta\left(\left(\sum_{i=0}^{N} ls_i\right) - Nlm_{min}\right) \tag{9}$$

The η parameter can be small so that the minimization of other constraints and the maximization of unknown visible both have a greater priority than the increase of number of visible landmarks beyond the defined threshold. When the threshold is not reached, the configuration is penalized:

$$K_l = \left(\frac{\left(\sum_{i=0}^{N} ls_i\right) - Nlm_{min}}{Nlm_{min}}\right)^2 \tag{10}$$

The evaluation function used as input to NEWUOA is then:

$$f_e = \lambda_z K_{\mathbf{C}_z} + \lambda_x K_{\mathbf{\Theta c}_x} + \lambda_y K_{\mathbf{\Theta c}_y} + \lambda_d K_d + \lambda_l K_l - \lambda_n N_{up} \tag{11}$$

The λ parameters are fixed manually to modify the balance between the constraints. As N_{up}, the number of pixels corresponding to unknown voxels, depends on the image size, the value of the parameters used in the constraints formulation should be modulated accordingly.

D. NEWUOA configuration

NEWUOA seeks the minimum of f_e by approximating it with a quadratic model, inside a trust region. Thus an initial configuration is provided to the software which limits the initial sampling to a subspace according to a range given by the user. Nevertheless NEWUOA's complete search is not limited to the trust region and can test vectors outside depending on the quadratic approximation obtained.

Due to the constraints used and the objects analyzed, different cases can result in disjoint local minima in our evaluation function as can be seen in the example shown in

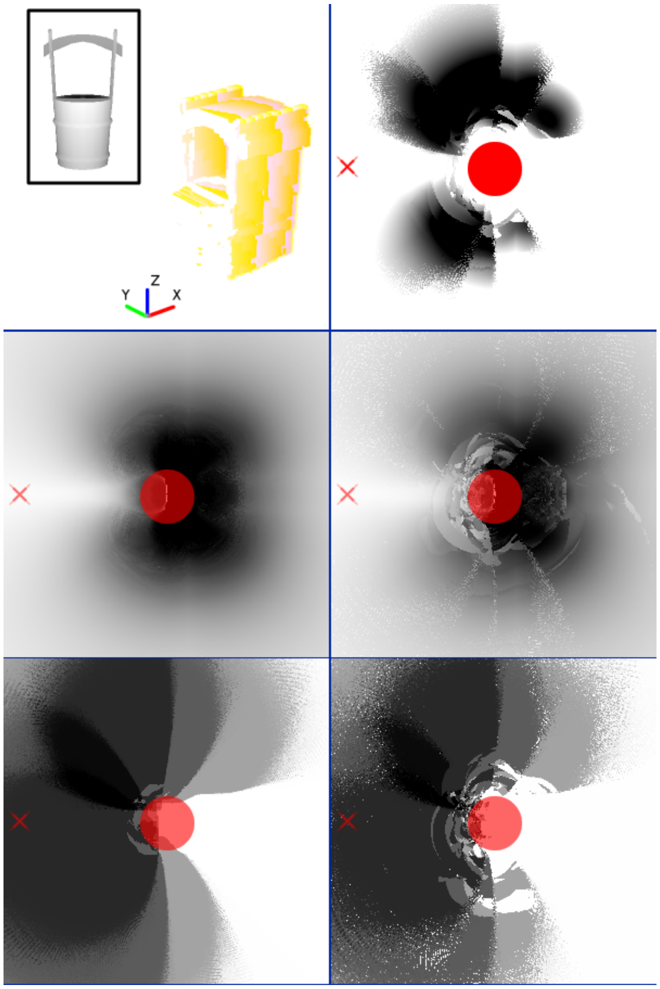

Fig. 2. Example of evaluation variations when moving the camera around an object carved once (top-left). The best orientation, i.e minimizing the evaluation function, is chosen. The red cross is the position from where the carving was done and the red circle represents the position of the object. Top-right: f_e. Center-left: f_e obtained when all parameters λ except λ_n are set to 0. Center-right: N_{up}. Bottom-left: f_e obtained when all parameters λ except λ_l are set to 0. Center-right: K_l.

Fig. 2. This figure shows some components of the evaluation function when the camera is moved in the XY plane around the carved object at a fixed height. Darker points correspond to better values. We can remark that using one constraint at a time (left-center and left-bottom images) to find the best orientation results in relatively smooth evaluation variations compared to the values obtained when all constraints are used (center and bottom images on the right). In such cases, the quadratic model cannot be pertinent if the trust region is too big.

In our actual implementation, NEWUOA is run once from a defined pose and run again iteratively by using its result configuration as a new starting pose. This is done until a chosen maximum number of iterations has been reached, or until the result pose is not better than the last starting one. A step of this iterative process is formulated as:

$$pose_k = Newuoa_k\left(pose_{k-1}\right) \tag{12}$$

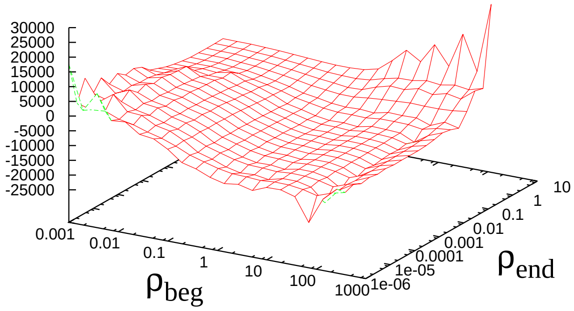

Fig. 3. Influence of trust region parameters on the evaluation of the pose obtained.

with k the iteration number of the NEWUOA algorithm from 1 to n, and $pose_{k-1}$ and $pose_k$ respectively the starting and found camera poses.

E. Second step: Posture Generator

Once an optimal camera pose has been found, the result is used as a constraint on the humanoid robot head in order to generate a whole-body posture that takes into account all other constraints such as stability, collisions, etc.

For this algorithm, the objective function for the PG is not necessary. Nevertheless it is possible to use it as an aesthetic criterion to place the robot posture close to a reference posture.

The starting robot pose is set using a pre-computed posture and a position deduced from the desired camera pose. In cases where the PG cannot converge, it can be launched again with a different pre-computed starting posture, or a different starting position.

IV. Simulations

A. NEWUOA tests for camera pose evaluation

We tested the influence of the trust region parameters on the optimal found with one iteration of NEWUOA. The parameter ρ_{beg} sets the maximum variation that can be taken by the camera pose parameters for the initial approximation, and the parameter ρ_{end} sets the accuracy of the optimum search. Tests were conducted by selecting a camera pose and by launching the optimization with different values for ρ_{beg} and ρ_{end}. This was repeated for 14 different objects with 3 different starting poses for each. Figure 3 presents the average of the results. The ρ parameters are multiplied by the object maximum size. Overall, better evaluated poses are obtained when ρ_{beg} is equal or superior to the object maximum size, and when ρ_{end} is relatively small.

The influence of the starting pose on the result was then tested by launching NEWUOA with different initial configurations. One of this test is illustrated in Fig. 4 where the camera is translated on the Y axis near the carved

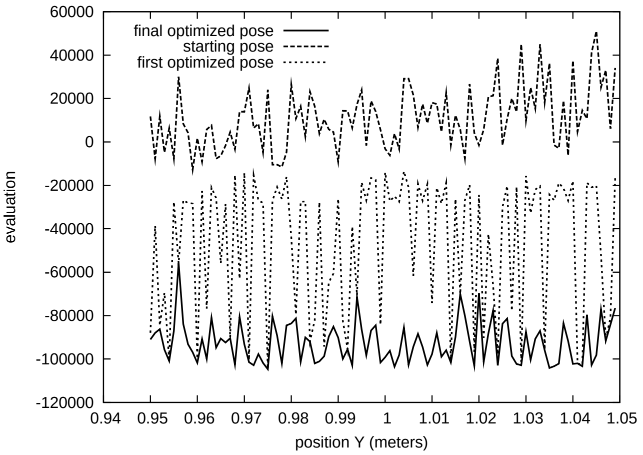

Fig. 4. Evaluation of the poses for $f_e(pose_s)$ and those found by our iterative optimization process $Newuoa_1\ (pose_s)$ and $Newuoa_n\ (pose_{n-1})$, depending on the initial starting position.

object shown in Fig. 2, with $\rho_{beg} = 0.4$ and $\rho_{end} = 10^{-5}$. Note that the evaluation of the unknown function, i.e the 'starting pose' curve, can change abruptly even with small variations of the pose. This highlights the complexity of our evaluation function which has a lot of local minima, and thus NEWUOA can generate relatively different quadratic approximations depending on the starting conditions. Though a single iteration of NEWUOA results in an improved pose, it is often stuck inside a local minima. Nevertheless, by using successive iterations, much better poses are usually reached. In fact, the camera can get moved up to 0.7 meters and rotated up to about 50 degrees in many final optimized poses around a small object, e.g. 0.4 meters long. In order to find a good pose, a big number of iterations is not necessary. In this test, the average number of iterations was 5 and the maximum number allowed, which was set to 10, was reached for only 2 percent of the tested initial poses.

During our tests, one iteration of NEWUOA takes between 1 and 3 seconds to find a minimum with an average computer. This is quick enough to apply our iterative method and select different initial starting poses in order to find a good Next-Best-View.

B. NEWUOA VS homogeneous sampling

We compared our method with a simple uniform sampling of the configuration space. This sampling is done around the last position where a space carving operation has been done. The number of samples as well as the limits of the area to test are defined manually for each of the 6 dimensions.

Not surprisingly, the uniform sampling can reach better pose using roughly the same number of sampled data. As noted earlier, depending on the object complexity, the NEWUOA search may find itself limited to the local minima close to the starting pose. Nevertheless, NEWUOA can find the local minima faster than a local uniform sampling, thus it is advantageous to mix the two methods: first, have a rough sampling of the areas of interest, then use NEWUOA to refine the search for the closest local minima.

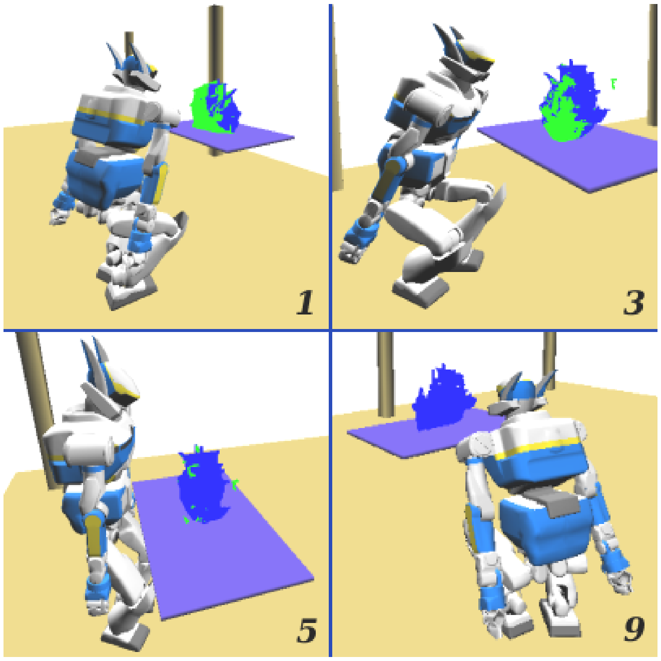

Fig. 5. Postures generated successively for the modeling of an unknown object

C. Modeling process simulation

The experimental setting is simulated by having a virtual 3D object perceived by a virtual camera. The modeling process loops through the following steps:

1) The disparity map is constructed using the object 3D information and is used to perform a space carving operation on the occupancy grid. Some known voxels are randomly selected to be considered as landmarks.
2) The NEWUOA routine is called in order to find an optimal camera pose by minimizing our evaluation function. We use a uniform sampling around the current position to select different starting poses from where our iterative search is launched.
3) When an optimal camera pose is found, it is sent to the PG in order to generate a whole-body posture.

Then we loop through all previously described steps until the amount of unknown voxels is below a specified threshold, or if it does not change after two space carving operations, i.e the unknown voxels cannot be perceived due to the constraints on the robot. Some of the 10 postures generated during a successful modeling process of a ship is illustrated in Fig. 5 with the updated occupancy grid at each step. The trust region parameters, ρ_{beg} and ρ_{end}, were set respectively to 0.4 and 10e-5. Other parameters settings are: $p = 3$, $\gamma = 20$, $Nlm_{min} = 5$, $\eta = 10^{-5}$, $\lambda_z = 200$, $\lambda_x = 80$, $\lambda_y = 80$, $\lambda_d = 100$, $\lambda_l = 10^5$ and $\lambda_n = 1$.

D. Pose generation

The second step of our Next-Best-View algorithm was tested by verifying that camera poses obtained in the first step

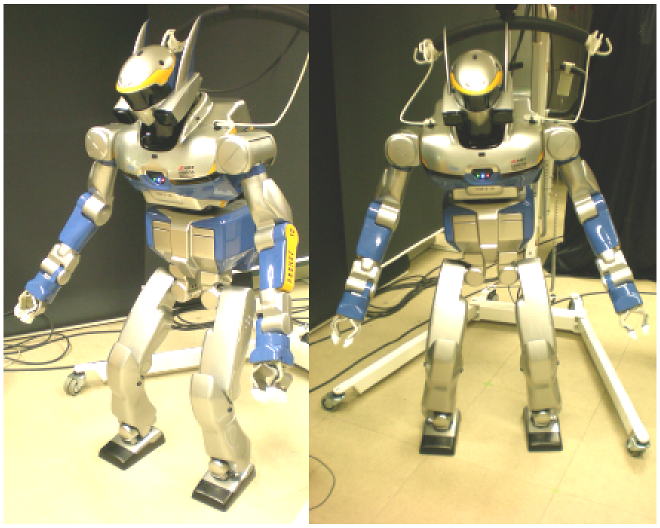

Fig. 6. Postures generated using our NBV algorithm

do not result in a constraint, on the robot head, impossible to satisfy when set in the PG with other constraints. Several camera poses were computed using different virtual objects with different states of space carving and the landmarks were randomly generated amongst the known voxels on the surface of the object.

The tests confirmed that the constraints set in the first step reduce the possible poses to what is achievable by the PG with our current settings. In our first simulations, we set the starting posture for the PG as a stand up position but found some cases where the posture could not be generated. This happens when the camera is set close to the minimum height limit. By using a squatting position as a starting posture, this convergence problem was not found afterwards.

Some of the whole-body postures obtained with the PG were played with OpenHRP and then on a real HRP-2 robot to ensure the stability constraint results in statically stable postures. Two of them are shown in fig. 6.

V. Conclusion

A new method to generate automatically postures for a humanoid robot depending on visual cues is presented. The postures are selected amongst the possible configurations allowed by stability, collisions, joint limitations and visual constraints, so as to complete the modeling of an unknown object using a minimum number of postures.

The presented method uses two different optimization methods, NEWUOA and FSQP, in order to get a reliable and fast generation of constrained posture, and thus solves the problems encountered with our previous approach.

Postures generated were checked to be free of self-collisions and statically stable on a real HRP-2 robot.

We are now planning to integrate our Next-Best-View algorithm with other works, focused on vision and motion planning tasks, in order to complete experimentally the autonomous modeling of the object.

Acknowledgment

This work is partially supported by grants from the ROBOT@CWE EU CEC project, Contract No. 34002 under the 6th Research program www.robot-at-cwe.eu.

The visualization of the experimental setup relied on the AMELIF framework presented in [15].

References

[1] F. Saidi, O. Stasse, K. Yokoi, and F. Kanehiro, "Online object search with a humanoid robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 1677–1682.

[2] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 4, pp. 91–110, 2004.

[3] K. Tarabanis, P. Allen, and R. Tsai, "A survey of sensor planning in computer vision," in *IEEE Transactions on Robotics and Automation*, 1995.

[4] W. Scott, G. Roth, and J. Rivest, "View planning for automated three-dimensional object reconstruction and inspection," *ACM Comput. Surv.*, 2003.

[5] C. Connolly, "The determination of next best views," in *IEEE International Conference on Robotics and Automation*, 1985.

[6] J. Banta, Y. Zhien, X. Wang, G. Zhang, M. Smith, and M. Abidi, "A best-nextview algorithm for three-dimensional scene reconstruction using range images," in *Proceedings SPIE*, 1995.

[7] J. Maver and R. Bajcsy, "Occlusions as a guide for planning the next view," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 1993.

[8] R. Pito, "A solution to the next best view problem for automated surface acquisition," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 1999.

[9] K. Yamazaki, M. Tomono, T. Tsubouchi, and S. Yuta, "3-d object modeling by a camera equipped on a mobile robot," in *IEEE ICRA Proceedings*, 2004.

[10] D. Lowe, "Local feature view clustering for 3d object recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[11] O. Stasse, D. Larlus, B. Lagarde, A. Escande, F. Saidi, A. Kheddar, K. Yokoi, and F. Jurie, "Towards autonomous object reconstruction for visual search by the humanoid robot hrp-2." in *IEEE RAS/RSJ Conference on Humanoids Robots, Pittsburg, USA, 30 Nov. - 2 Dec.*, 2007.

[12] T. Foissotte, O. Stasse, A. Escande, and A. Kheddar, "A next-best-view algorithm for autonomous 3d object modeling by a humanoid robot," in *IEEE RAS/RSJ Conference on Humanoids Robots, Daejeon, South Korea, 1-3 Dec.*, 2008.

[13] A. Escande, A. Kheddar, and S. Miossec, "Planning support contact-points for humanoid robots and experiments on hrp-2," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006, pp. 2974 – 2979.

[14] M. Powell, "The newuoa software for unconstrained optimization without derivatives," University of Cambridge, Tech. Rep. DAMTP Report 2004/NA05, 2004.

[15] P. Evrard, F. Keith, J.-R. Chardonnet, and A. Kheddar, "Framework for haptic interaction with virtual avatars," in *17th IEEE International Symposium on Robot and Human Interactive Communication (IEEE RO-MAN 2008)*, 2008.