International Journal of Humanoid Robotics
© World Scientific Publishing Company

# AUTONOMOUS 3D OBJECT MODELING BY A HUMANOID USING AN OPTIMIZATION-DRIVEN NEXT-BEST-VIEW FORMULATION

TOREA FOISSOTTE
*CNRS-LIRMM, France, CNRS-AIST JRL UMI 3218/CRT, Japan*

OLIVIER STASSE
*CNRS-LIRMM, France, CNRS-AIST JRL UMI 3218/CRT, Japan*

PIERRE-BRICE WIEBER
*INRIA - Rhone-Alpes, France*

ADRIEN ESCANDE
*CEA-LIST, France*

ABDERRAHMANE KHEDDAR
*CNRS-LIRMM, France, CNRS-AIST JRL UMI 3218/CRT, Japan*

Received Day Month Year
Revised Day Month Year
Accepted Day Month Year

An original method to build a visual model for unknown objects by a humanoid robot is proposed. The algorithm ensures successful autonomous realization of this goal by addressing the problem as an active coupling between computer vision and whole-body posture generation. The visual model is built through the repeated execution of two processes. The first one considers the current knowledge about the visual aspects and shape of the object to deduce a preferred viewpoint with the aim of reducing the uncertainty of the shape and appearance of the object. This is done while considering the constraints related to the embodiment of the vision sensors in the humanoid head. The second process generates a whole robot posture using the desired head pose while solving additional constraints such as collision avoidance and joint limitations. The main contribution of our approach relies on the use of different optimization algorithms to find an optimal viewpoint by including the humanoid specificities in terms of constraints, an embedded vision sensor, and redundant motion capabilities. This approach differs significantly from those of traditional works addressing the problem of autonomously building an object model.

*Keywords*: Object modeling; next-best-view; optimization; NEWUOA; humanoid; posture generation.

## 1. Introduction

This work focuses on the autonomous modeling of an unknown object in a known environment by a humanoid robot. Such functionality is expected to be helpful in enhancing the ability of multi-purpose robots to collaborate with human partners. For instance, visual models of new objects can be built and stored in a knowledge database autonomously. Additional properties and functionalities of the object models can then be added through interactions with human collaborators. The constraints of our system are then set according to the scenario illustrated in Fig. 1. A sensor system, consisting of a stereo rig, is embedded within the HRP-2 head. The cameras are used to perceive the unknown object from different viewpoints. The goal of our work is then to compute each viewpoint and the corresponding robot pose depending on the information available and several constraints related to vision, motion control and the robot body.

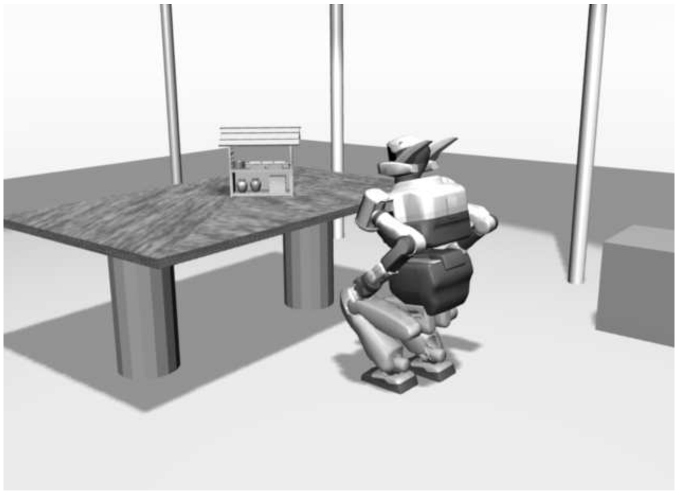

Fig. 1. Next-Best-View for a humanoid robot: finding the next posture to model an unknown object.

This work is part of the on-going ”treasure hunting” project [1] in our laboratory, the goal of which is the efficient construction of the model of an unknown object followed by its autonomous retrieval in an unknown, possibly cluttered, environment. The increased recent interest [2] in this type of problem is evidenced by the Semantic Robot Vision Challenge organized at CVPR.

The visual model should be sufficiently rich to allow not only the detection and recognition but also the manipulation of the object of interest. This is achieved by using two complementary modeling approaches. First, we use the stereo rig to obtain an approximation of the visible 3D surface of the object. This surface is

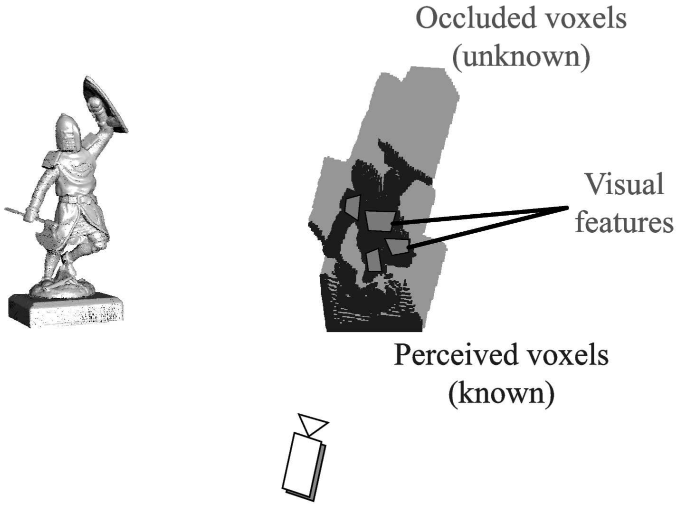

Fig. 2. Virtual model construction. (left) Original 3D model. (right) First update of the virtual model constructed using stereo vision and visual feature detection.

used to update an occupancy grid that records the current state of our knowledge about the 3D shape of the object. Second, visual features on the visible surface of the object are detected. Our algorithm does not rely on any one specific type of feature, and thus, several types of 2D features that are robust to slight modifications of the perception conditions can be used, such as SIFT [3], Affine-SIFT [8], or SURF [7]. These features can later be used for the fast detection and recognition of the object of interest, namely, when it will be placed in a different environment. An example of modeling is illustrated in Fig. 2.

The main originality of this work lies in the solution provided to generate viewing postures for a humanoid robot by considering its incremental knowledge of the environment and the visual properties of the object of interest. Two steps are required to achieve this result: (i) find a viewpoint that maximizes the amount of unknown data from the object that can be visible and (ii) generate a whole-body pose that is statically stable, collision-free and that sets the head position and orientation according to the desired camera pose. This solution is then integrated with other recent works on path planning and control to realistically simulate the modeling task using an HRP-2 robot model and the dynamic engine of OpenHRP-3.

The remainder of this paper is organized as follows. Section 2 presents some previous works that address similar problems. Section 3 presents an overview of our algorithm. Sections 4 and 5 respectively describe the first and second steps used to generate a robot posture. Finally, section 6 presents simulation results obtained using our method.

## 2. Related works

### 2.1. *Vision and viewpoint selection*

Many existing works aimed at computing a viewpoint according to some specific criteria focus on environment exploration [4]. Most such works considered wheeled robots; in fact, we found only one that considered the particularities of using humanoid robots [5].

With regard to computer vision, the problem of visual object recognition has been thoroughly addressed and has converged toward fast and robust methods that can be used in embedded systems [6]. Nevertheless, the modeling part usually relies on a supervised method where different views of an object are obtained manually by a human operator and then stored in a database available to the recognition algorithm.

In a rather unique work that tackled the problem of automatic modeling of an object by relying on vision [9], a mobile robot moves in a circle around an object in order to build its model. The algorithm uses visual feature detection and structure from motion to generate a rough 3D model. However, the motion around the object is planned offline depending on the distance and maximum size of the object but independently of its shape.

### 2.2. *Next-Best-View*

Many works have focused on the Next-Best-View (NBV) problem, i.e, the problem of planning successively different sensor poses in order to create the 3D model of an unknown object. Although we have extensively investigated currently available literature, we were unable to find any reports on autonomous object modeling by a humanoid robot.

The first well-known paper on NBV was by Connolly [10]; it described two algorithms to model an object with an octree structure in which voxels have one of three states: empty, occupied, or unseen. The first algorithm limits the sensor position to the surface of a sphere, the origin of which is the object center, and the sensor viewing direction is set toward this center. The surface of the sphere is sampled and the visibility of the unseen area is tested for all samples. This approach has subsequently been used in various works [11], although viewpoint selection is carried out according to different criteria based on the spatial distribution of the unseen areas. The second algorithm sacrifies precision in order to considerably reduce the computation time by directly generating a single next viewpoint pose according to the repartition of unseen voxels.

Other works [12] [13] consider an object placed on a turntable and restrict the sensor position to a cylindrical surface around the object, with the viewing direction set toward the object center. The algorithms mix a surface visibility criterion with an occluded surface visibility criterion in order to select the NBV position.

Other hypotheses and limitations of the main works in this field are detailed in two authoritative surveys [14] [15]. We note a common trend in the aim of the previous algorithms: to obtain an accurate 3D reconstruction of an object using voxels or polygons while reducing the number of viewpoints required. The typical assumptions made about a sensor 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 relative to the object position and orientation. In our case, however, such assumptions cannot be made when using an embedded stereo rig on an autonomously controlled humanoid robot. Depth perception through stereo vision can result in errors in the perceived 3D surface of the object and humanoid motion control can also result in errors in the perceived position and orientation of the robot with respect to the object. We thus have a limitation on the maximum accuracy that can be obtained for the 3D model of the object.

Another common hypothesis in previous works is that the object to be modeled is considered to be within a sphere or on a turntable, i.e, the sensor positioning space complexity to be evaluated is reduced to two dimensions because the sensor distance from the object center is fixed and its orientation is set toward the object center. This is useful for reducing the computation time required to obtain a solution; however it also restricts the variety of objects that can be modeled. For example, objects with complex concavities or oblong shapes may require the sensor to be closer to some specific parts and/or require having a viewing vector not targeting the object center. Furthermore, if we consider cluttered environments, obstacles may be placed at the specific distance from the object center where the sensor can be located. In such cases, important viewpoints may not be reached.

Because the hypotheses used in previous works impose some constraints on the sensor characteristics as well as on the object size and complexity that are not adequate with a humanoid, we design a novel NBV algorithm. Our method aims to overcome previous limitations while considering the specific constraints related to a humanoid robot.

## 3. Next-Best-View computation process

We first review the modeling process that is simulated and the related hypotheses that are made. At the beginning of the experiment, we consider that the humanoid robot faces an unknown object to be modeled. The approximate size and position of the object is known with sufficient precision so that a virtual occupation grid containing the entire object can be set at the proper location and with proper dimensions. We note that this assumption is not critical for our algorithm, and thus, it is possible to start with a specific grid size and position, and include additional occupancy grids in places where parts of the object are bigger than expected.

The environment is considered to be known and modeled to facilitate its visual discrimination with the unknown object and also to allow the walk planner to generate a collision-free motion trajectory between computed postures.

During our simulation process, the modeling algorithm is executed as follows:

(1) The depth map from the actual viewpoint is obtained.
(2) The filtered depth map is processed using a Sobel operator to create a normal map, following a common method used in computer graphics to perform bump mapping [16].
(3) Some landmarks are generated on the perceived envelop of the object and their normal vectors are computed.
(4) The voxels of the occupation grid are superimposed on the depth map by perspective projection to realize a space-carving operation. Thus, each voxel is set to one of the 3 states: empty, known (i.e, perceived using the camera), or unknown (i.e, occluded by known voxels or out of the field of vision). Actually, our algorithm only needs to consider the voxels on the model surface being built, i.e, known or unknown voxels that have at least one empty neighbor voxel.
(5) The resulting occupancy grid is provided as an input to our NBV algorithm, described in the following section 4. This algorithm searches for a target robot head pose by considering some related constraints.
(6) At this stage, we test the termination criterion. The modeling task stops in two cases: the model is considered finished when (i) the prediction of unknown to be perceived falls under a desired threshold value or (ii) this amount cannot be reduced after a pre-defined number of successive poses. In the second case, the model is considered incomplete but good viewpoints are considered to be out of reach from the robot because of collision risks, joint limitations, etc.
(7) A whole-body robot pose is then generated using our Posture Generator (PG) detailed in section 5.
(8) The robot moves from the actual posture to the generated one. The motion planning software component for the humanoid is built upon KineoWorks [17].
(9) Go back to (1).

## 4. Generation of desired viewpoint

The problem of finding an adequate viewpoint to complete the modeling of an object can be formulated as the minimization of a function $f$ that evaluates the unknown visible projected into the camera. Traditional works on NBV solve this problem by considering the camera configuration space as the input space of $f$. The problem's dimensionality is reduced and the configuration space is sampled in order to test a limited set of poses and find the best one in an acceptable amount of time. To do so, some assumptions are made on the size and complexity of the object to be modeled, and the environment is considered free of obstacles. However using latest hardware and optimization algorithms, it is possible to relax these assumptions while keeping a reasonable computation time. In order to broaden the types of object to be modeled while considering the constraints related to the use of a humanoid, a novel solution to the NBV problem is introduced by using the two steps illustrated in Fig. 3: first, we find a camera position and orientation that maximizes the amount of unknown visible while solving specific constraints related to the robot head; then, we generate a whole-body posture for the robot by considering the desired viewpoint as well as additional constraints related to the humanoid body.

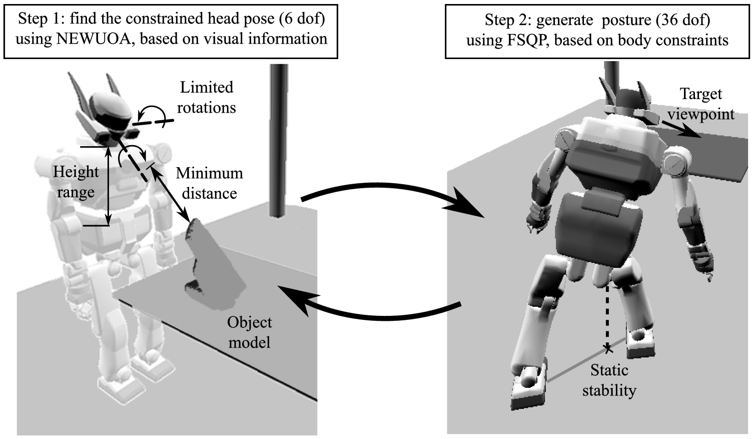
Step 1: find the constrained head pose (6 dof) using NEWUOA, based on visual information
Step 2: generate posture (36 dof) using FSQP, based on body constraints
Limited rotations
Height range
Minimum distance
Object model
Target viewpoint
Static stability

Fig. 3. Two steps to generate the next posture to update the object model.

We propose to solve the first step by using NEWUOA [18], a method that searches for the local minimum of a function by refining a local quadratic approximation through a deterministic iterative sampling, which can thus be used for non-differentiable functions. The sampled vectors at each step in the NEWUOA search process are selected according to the previous sampling results and the actual state of the quadratic approximation. A trust region must be defined using two radius parameters, $\rho_{beg}$ and $\rho_{end}$, and a given starting vector that will be the camera pose in our case. The trust region influences the sampling process but it does not limit it. Indeed, depending on the quadratic approximation found, vectors outside this region can be tested. NEWUOA has the advantages of being fast and robust to noise while allowing us to keep the 6 degrees of freedom required for the viewpoint.

### 4.1. *Evaluation of unknown visible*

In this approach, the estimation of unknown data visible from a specific viewpoint relies on the visualization of the current occupancy grid that is composed of voxels represented as colored cubes. The visualization of the grid from any specific viewpoint can be computed rapidly, typically in less than a few milliseconds, by taking advantage of current GPU acceleration capabilities. Although such a visualization results in a function that is not continuous and can present variations of small amplitude, these characteristics have a negligible influence on the convergence of NEWUOA when the trust region is sufficiently large. Therefore, we consider such an approximate representation to be a useful indicator to find a good viewpoint.

In this work, the amount of unknown visible, denoted by $N_{up}$, is set as the number of *unknown* voxels visible multiplied by the logarithm of the number of *unknown* voxels' pixels. This choice is justified by considering what happens if we focus only on either the number of *unknown* voxels visible or the number of *unknown* voxels' pixels. Because of the perspective projection, more voxels can be visible from a farther position, thus driving the robot far away from the object; however each voxel would then appear unnecessarily small and we obtain a poor precision for the 3D surface of the object. When considering only pixels, the robot tends to come as close as possible to the object in front of few *unknown* voxels. In this case, each viewpoint covers only a small fraction of the object, increasing the number of poses required to complete the modeling.

### 4.2. *Constraints on camera pose*

Although 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 computed viewpoint. The constraints on the camera position $\mathbf{C}$ and orientation $\psi_{\mathbf{c}}$ included in the evaluation function of the first step given to NEWUOA are

$$\begin{cases} C_{zmin} < \mathbf{C}_z < C_{zmax} & (1) \\ \forall Vi, d_{min} < d(\mathbf{C}, \mathbf{Vi_{center}}) & (2) \\ \psi_{\mathbf{c}xmin} < \psi_{\mathbf{c}x} < \psi_{\mathbf{c}xmax} & (3) \\ \psi_{\mathbf{c}ymin} < \psi_{\mathbf{c}y} < \psi_{\mathbf{c}ymax} & (4) \\ N_l > N_{lmin} & (5) \\ \forall i, \mathbf{C} \neq F_i \vee \psi_{\mathbf{c}} \neq Fr_i & (6) \end{cases}$$

The range of the camera height is limited by (1) to what is accessible by the humanoid size and its range of possible postures.

A minimum limit distance $d_{min}$ is required between the robot head and $\mathbf{Vi_{center}}$, the center of each voxel of the object, in order to efficiently use the stereo vision. This constraint is expressed in (2).

The rotations on the X (roll) and Y (pitch) axes are limited by (3) and (4) to ranges set according to the particularities of the robot.

Constraint (5) keeps a minimum number of landmarks, i.e, features that were detected in previous views, visible from the resulting viewpoint. These landmarks can be used to correct eventual positioning errors of the robot with respect to the object when it reaches the desired viewpoint. This is necessary to enhance the precision when updating the occupancy grid with a newly acquired 3D surface.

Finally, the particular constraint (6) ensures that the resulting pose will not be near previously found poses, with position $F_i$ and orientation $Fr_i$, that could not be reached by following the steps in the modeling process. It is also used to avoid positions in the environment where known obstacles are located. This constraint is necessary to ensure that the algorithm can converge toward a valid posture although some constraints are not expressed in the viewpoint search. For example, some obstacles in the environment may limit the possible motion trajectories of the robot while not visually occluding the view of the camera. In this case, the motion planner may fail to find a way between the current posture and the target one.

### 4.3. *Evaluation function formulation*

In order to include the constraints presented in 4.2 into the function that NEWUOA evaluates, we need to formulate and include them in a manner that does not strongly influence the visibility evaluation when the constraints are statisfied. On the other hand, when the constraints are violated, the function should increase significantly depending on its distance to the resolution of the violated constraints.

#### 4.3.1. *Interval*

The interval constraints (1), (3), and (4), are expressed as

$$K_v = (\alpha\, v - \mu)^p \tag{7}$$

where parameters $\alpha$ and $\mu$ are set according to the limits possible for the variable $N_{up}$. These are used to respectively modulate the interval center and the width depending on the parameter $v$ to be constrained, and thus, they are directly deduced from the specificities of the robot body. $v$ can correspond to the viewpoint height $\mathbf{C}_z$ and orientation angles $\psi_{\mathbf{c}_x}$ and $\psi_{\mathbf{c}_y}$. $p$ can be set to a large value, typically 4, so that the result is close to 0 within the interval and increases rapidly outside it. The parameters are computed in order to reach the maximum possible value of $N_{up}$ when we reach the point at which the constraint is not satisfied.

#### 4.3.2. *Minimum distance*

The inequality constraint (2) related to the minimum distance between the camera and the object is formulated as

$$K_d = \exp^r\left(\gamma\ (d_{min} - d(\mathbf{C}, V_{near}))\right) \tag{8}$$

where $\gamma$ and $r$ are parameters that are also set according to $N_{up}$, and $V_{near}$ is the closest voxel to the camera. The minimum distance $d_{min}$ is related to the specific geometric configuration of the stereo rig being used.

#### 4.3.3. *Landmark visibility*

For the landmark visibility constraint (5), the formulation relies both on the visible surface of landmarks and their normal vector. The surface visibility for each landmark $i$ is computed relative to its number of pixels visible from the current viewpoint $pv_i$ using a sigmoid function:

$$ls_i = \frac{1}{1 + exp\left(pmin_i - pv_i\right)} \tag{9}$$

The parameter $pmin_i$ is the minimum number of pixels required to consider the landmark $i$ to be visible, and its value depends on the original landmark size. The visibility of each landmark relative to its normal vector $\mathbf{Nl}_i$ and the current camera view direction vector $\mathbf{C}_{view}$ is expressed using another sigmoid function:

$$ln_i = \frac{1}{1 + exp\left(\beta\left(\left(\mathbf{C}_{view}.\mathbf{Nl}_i\right) + \phi\right)\right)} \tag{10}$$

where $\phi$ is related to the allowed range of angles and $\beta$ determine the slope of the sigmoid function.

The final visibility coefficient for each landmark is computed by multiplying $ls_i$ with $ln_i$. We set an arbitrarily defined minimum number of visible landmarks $Nlm_{min}$ that is compared with the obtained coefficients by

$$lv = \left(\sum_{i=0}^{N} ls_i.ln_i\right) - Nlm_{min} \tag{11}$$

The constraint for the evaluation function is defined in one of two ways depending on the sign of $lv$. Configurations maximizing $lv$ are slightly encouraged when it is positive:

$$K_l = -\eta\, lv \tag{12}$$

The parameter $\eta$ can be small such that the minimization of other constraints and the maximization of unknown visible both have a greater priority than the increase in the number of visible landmarks beyond the defined threshold.

In the other case, in which $lv \leq 0$, the configurations are greatly penalized:

$$K_l = 2\, I_p \left(\frac{lv}{Nlm_{min}}\right)^2 \tag{13}$$

The penalty is expressed in relation to the total number of pixels $I_p$ in the camera image.

### 4.3.4. *Forbidden poses*

The constraint to avoid unreachable postures (6) is simply formulated as a distance between the viewpoint considered and each of them:

$$K_f = \sum_{fp} exp\left(-\delta.D_{fp}\right) \tag{14}$$

where $D_i$ represents the sum of absolute differences between the values of the actual viewpoint and the unreachable pose $i$. The parameter $\delta$ corresponds to the sensitivity of the constraint.

4.3.5. *Viewpoint evaluation function*

The evaluation function, used as an input to the NEWUOA algorithm, includes all previously defined constraint formulations as well as the evaluation of visual data:

$$f_e = \lambda_z K_{\mathbf{C}_z} + \lambda_x K_{\psi_{\mathbf{c}_x}} + \lambda_y K_{\psi_{\mathbf{c}_y}} + \lambda_d K_d + \lambda_l K_l + \lambda_f K_f - N_{up} \tag{15}$$

where the $\lambda$ parameters are computed to modify the scale of each constraint in order to match the range of values that can be taken by the variable $N_{up}$.

### 4.4. *NEWUOA configuration*

NEWUOA is used to seek the minimum of $f_e$ by constructing a local approximation using a quadratic model, as illustrated in the left hand side graph of Fig. 4. Three parameters are used as an input to this optimization algorithm: an initial vector from where the search is started, a value $\rho_{beg}$ that delimits the trust region around the initial vector in order to build the initial quadratic approximation, and a desired accuracy value $\rho_{end}$ used as a stopping criterion. Because of the nature of the NBV problem and the constraints used, our objective function can present many local minima that are quite disjoint, as illustrated in the right hand side of Fig. 4. This figure shows the best results for $f_e$ obtained for the knight object carved once from Fig. 2, by constraining the camera at a fixed height and moving the sensor in the XY plane. For each sampled position, the orientation with the best result for the objective function is selected. Darker points correspond to better evaluations. The landmark visibility constraint formulation as well as the presence of possible self-occlusions can result in abrupt local variations in the objective function. The

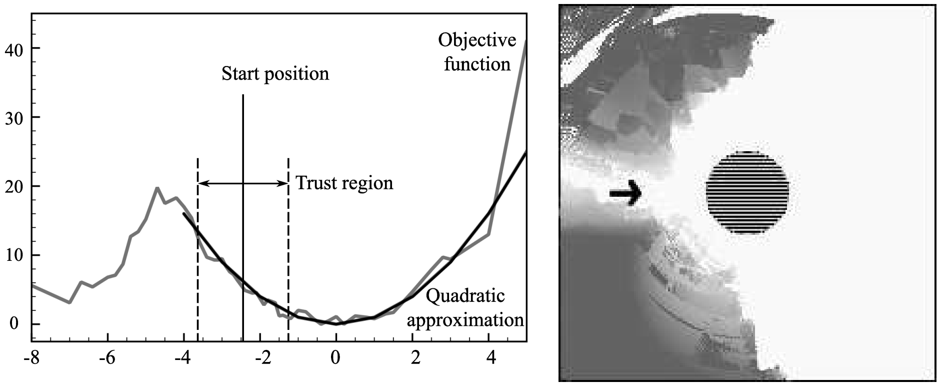

Fig. 4. NEWUOA evaluation. (left) NEWUOA method to find the minimum of a non-differentiable function. (right) Objective function results depending on the sensor position XY. The object location is displayed as the disk in the center and the previous perception pose is represented as an arrow.

minimum distance constraint also produces an important maximum in the middle of the search space around the object. Thus, the size of the trust region needs to be limited to some local space in order to compute an approximation that is pertinent enough.

### 4.5. *Viewpoint search process*

Because the quality of the results can depend greatly on the starting poses given, two additional techniques are implemented, as illustrated in Fig. 5. First, we run NEWUOA in an iterative manner, i.e, it is run once using a defined starting pose and again 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

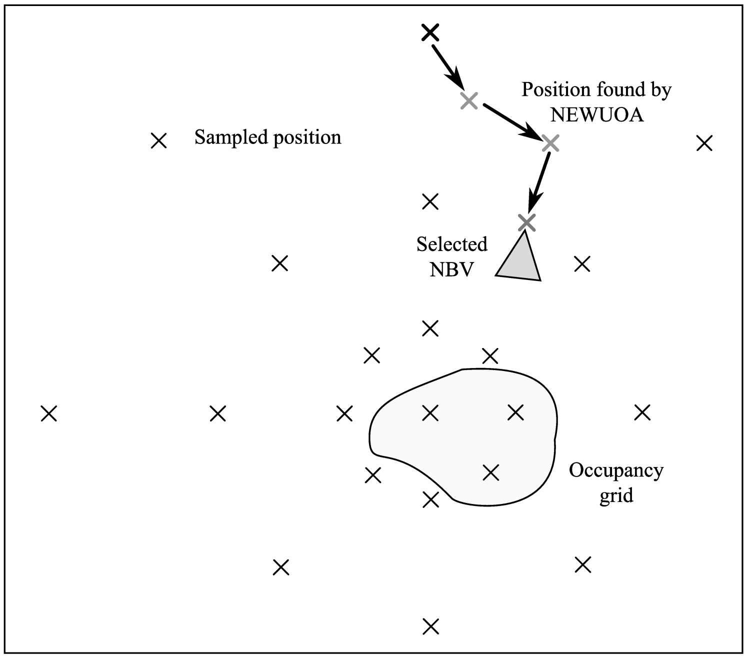

Fig. 5. Illustration of the NBV selection for the robot head.

$$pose_k = Newuoa_k\,(pose_{k-1}) \tag{16}$$

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

Second, we precompute a set of 3D starting positions around the object and launch the iterative process for each of them. The results of all optimizations are then compared to select the best camera pose. The sampled positions can be generated within the object to handle cases in which it has large empty spaces. For example, the algorithm can be applied to model both the inside and the outside of a house. The positions are distributed in the space relative to their distance from the object: the density decreases when moving away from the object because greater motions are required to obtain significant visual changes.

The desired six d.o.f camera pose can thus be obtained using a global three-dimensional sampling in conjunction with iterative local six-dimensional searches. This camera pose is then used as the starting point for the whole-body posture generation computed in the second step of our NBV algorithm presented in the following section.

## 5. Posture Generator

The Posture Generator (PG), presented as part of a work from Escande *et al.* [19], provides a whole-body posture for the robot in the second step of our NBV algorithm. The PG relies on feasible sequential quadratic programming (FSQP), a gradient-based optimization method, to provide a posture that minimizes an objective function while solving given constraints.

We note that a previous work [20] presented an attempt to directly include the evaluation of viewpoints as a $\mathcal{C}^1$ function that can be included in the PG. Such a solution would solve the NBV problem in one coherent step; however, although our analytical formulation results in a good evaluation, it has a relatively high computation cost and it also presents high variations in the gradient that result in convergence problems to generate a posture. Moreover, it is difficult to innclude additional vision constraints in such a formulation.

### 5.1. *Constraints on robot body*

Once an optimal camera pose has been found in the first step of our NBV algorithm, it is used as a constraint on the humanoid robot head in order to generate a whole-body posture that also considers other constraints: static stability, self-collision avoidance, collision avoidance with the environment, keeping the feet flat on the ground, and joint limitations. The posture generation problem is then written as

$$\min_{\mathbf{q}\in X} f_1(\mathbf{q}) \tag{17}$$

$$\begin{cases} \Theta_{\min} \leq \Theta \leq \Theta_{\max} & (18) \\ d\mathcal{C}_{min} \leq d(B_i(\mathbf{q}), B_j(\mathbf{q}))\ \forall(i,j) \in \mathcal{C} & (19) \\ F_l^z(\mathbf{q}) = F_r^z(\mathbf{q}) = 0 & (20) \\ d_{\min} \leq \parallel \mathbf{C}(\mathbf{q}) - \mathbf{x} \parallel^2 & (21) \\ dS_{min} \leq d(\mathbf{pc}, F_{seg}) & (22) \\ \mathbf{C} = \mathbf{C}_{s1} \wedge \psi_{\mathbf{c}} = \psi_{\mathbf{c}s1} & (23) \end{cases}$$

where $f_1$ is the objective function to be minimized, $\mathbf{q} = [r\ w\ \Theta]^\top$, $r$ is the position of the free-floating body, $w$ is its orientation, and $\Theta = \{\theta_0 \dots \theta_d\}$ denotes the robot's joints.

$X$ represents the set of constraints from (18) to (23).

$\Theta_{\min}$ and $\Theta_{\max}$ are two vectors that respectively represent the minimum and maximum limits for each joint.

$d(B_i(\mathbf{q}), B_j(\mathbf{q}))$ is the $\mathcal{C}^1$ distance between two bodies introduced by Escande *et al.* [21] and it must be greater than a precision value $d\mathcal{C}_{min}$. $\mathcal{C}$ is the set of collision pairs that are tracked to avoid non-desirable collisions and auto-collisions. We note that $B_i$ is not constrained to be a robot's body but it could be an object in the environment [22].

The constraint (20) ensures that the feet are on the ground by constraining the height value of the left foot $F_l$ and right foot $F_r$ to 0.

The constraint (21) ensures that the vision system distance to the object $\mathbf{x}$ is greater than a predefined value related to the stereo rig parameters used.

The static stability of the posture is set by (22), where $\mathbf{pc} = [c_x\ c_y]\top$ is the projection of the CoM on the floor, $F_{seg}$ is the segment between the feet centers, and $dS_{min}$ is a value that is sufficiently small such that the CoM cannot be projected outside the support polygon of the robot.

Finally, the last constraint sets the robot head according to the position $\mathbf{C}_{s1}$ and orientation $\psi_{\mathbf{c}s1}$ found in the first step.

### 5.2. *Posture computation*

For this algorithm, the objective function $f_1(\mathbf{q})$ for the PG is not necessary. Nevertheless, it can be used as an esthetic criterion to place the robot posture close to a reference posture.

Posture generation requires an initial posture and an initial free-flyer position and orientation in order to start the search. In this work, the starting posture is always set as a squatting posture because it is easier to reach more joint configurations from this posture. The starting free-flyer position and orientation are set accordingly to the desired viewpoint pose.

In cases in which the PG cannot converge, the goal camera pose is included in the list of forbidden poses that is used in constraint (6) described in section 4.2, and the first step is started again to find another viewpoint.

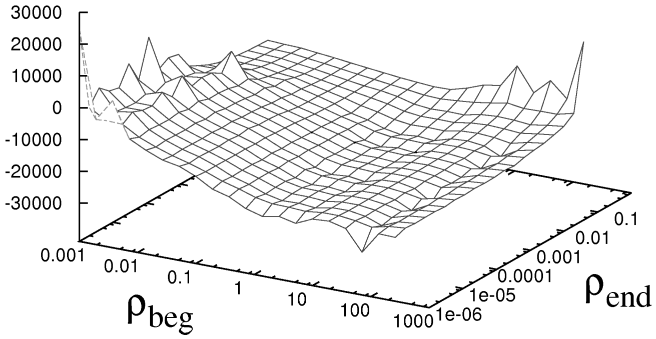

Fig. 6. Influence of NEWUOA trust region parameters on the search results. The $\rho$ parameters are multiplied by the maximum object size.

## 6. Simulation results

Because our solution differs significantly from previous NBV works in terms of the hypotheses, termination criteria, and constraints used, it is difficult to perform a meaningful comparison with existing methods. Therefore, we focus on the analysis of the simulation performance and behavior of our NBV algorithm.

### 6.1. *NEWUOA tests for camera pose evaluation*

Because of the nature of the NBV problem as well as the formulation of our objective function $f_e$, described in section 4.3, the initial conditions for a NEWUOA search can strongly influence the viewpoint found. We thus tested the variation of the results obtained with a NEWUOA search depending on the starting position and the trust region parameters.

Fig. 6 shows the average results for the viewpoint obtained depending on the $\rho$ parameters. As described previously in section 4.4, $\rho_{beg}$ sets the maximum variation that can be taken by the camera pose parameters for the initial quadratic approximation and $\rho_{end}$ sets the desired accuracy of the optimum search. The tests were conducted by selecting a camera pose around an object model and by launching the optimization with different values for $\rho_{beg}$ and $\rho_{end}$. This was repeated for 14 different objects with 3 different starting poses for each. Overall, the evaluated poses were better when $\rho_{beg}$ is similar to the maximum object size and when $\rho_{end}$ is smaller than one-hundredth this size.

Fig. 7 shows the influence of the starting pose on the viewpoints found by a single NEWUOA search and by an iterative NEWUOA search. This was tested by launching the search with different initial configurations, i.e, the camera is translated on the Y-axis in front of an object model. First, we 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

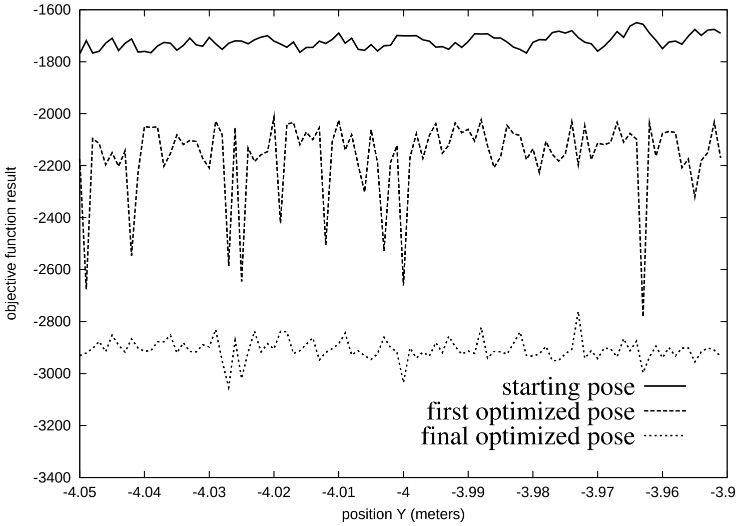

Fig. 7. Influence of the starting position on $f_e(pose_s)$ and the viewpoints found by our iterative optimization process, $Newuoa_1\,(pose_s)$ and $Newuoa_n\,(pose_{n-1})$.

function, as already discussed in 4.4, that has many local minima. Depending on the starting position, NEWUOA can thus generate relatively different quadratic approximations that will lead to the selection of different samples. This graph also highlights that although 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 viewpoints are reached with improved regularity. In fact, the camera can be moved by up to 0.7 m and rotated by up to approximately 50 degrees in many final optimized poses around a small object, e.g, a 0.4-m-long object. We note that in order to find a good pose, many successive search iterations are 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.

### 6.2. *NEWUOA VS. fixed sampling*

We compared the results obtained with a simple NEWUOA search against a pre-computed fixed sampling of the 6D viewpoint configuration space. This sampling is performed around the last position where a space-carving operation has been performed. The number of samples as well as the limits of the area to be tested are defined manually for each of the 6 dimensions.

Not surprisingly, the fixed sampling can result in viewpoints with similar or better results using roughly the same number of sampled vectors. As noted earlier, depending on various parameters such as the object complexity or the distribution of landmarks on the object surface, the NEWUOA search may find itself restricted to local minima close to the starting pose. Nevertheless, such local minima can be reached by NEWUOA using less samples than a fixed sampling of the local space. Thus, our search for a viewpoint presented in section 4.5 includes the two methods: first, carry out a rough sampling of positions in the areas of interest and then use NEWUOA to refine the search for the local minima at proximity.

### 6.3. *Computation time*

Each evaluation of a viewpoint relies on the OpenGL visualization of the occupancy grid that is loaded in the graphic card memory. The evaluation time is thus relatively small and remains of the order of $10^{-2}$ s although, of course, it can vary depending on the number of voxels in the model or the hardware specifications.

The search for the best viewpoint in the first step of the algorithm typically requires few thousands of evaluations. This depends on the number of sampled positions for the preliminary search and the input parameters for NEWUOA. During our tests, the first step could provide a solution within 10 s to 1 mn.

The second step can generate a posture in time of the order of $10^{-1}$ s if the starting conditions are relatively close to the solution and if there are no obstacles in the final location. In others cases, it can take up to a few seconds to obtain a solution or to abort the search.

### 6.4. *Pose generation*

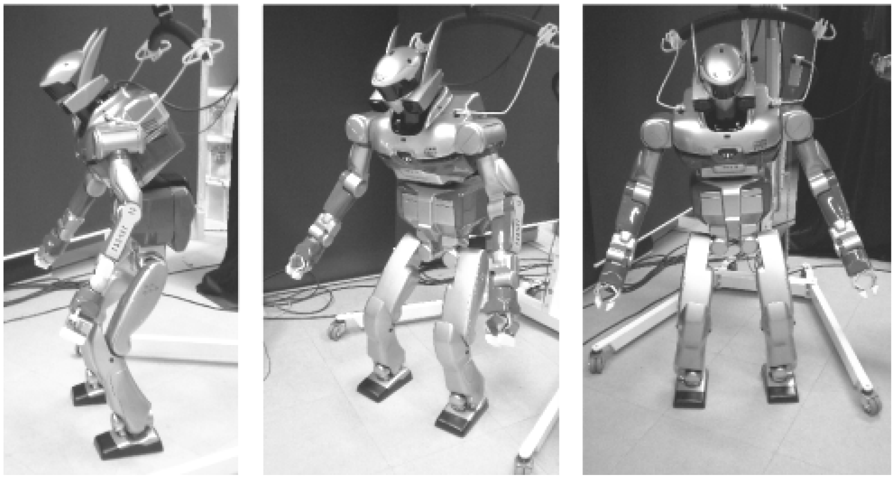

Fig. 8. Postures generated on an HRP-2 humanoid.

The second step of our NBV algorithm was tested by verifying that camera poses obtained in the first step do not result in a constraint on the robot head that is impossible to satisfy when set in the PG with other constraints. Several camera poses were computed using different virtual objects with different space-carving states and the landmarks were randomly generated amongst the known voxels on the surface of the object. Some of the generated postures, presented in Fig. 8, were tested on a real HRP-2 robot to ensure their stability and the avoidance of self-occlusions.

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. It should be noted that the posture used to initialize the PG has some influence on the convergence. Highly constrained postures such as squatting poses are often difficult to generate from the default standing posture; however the opposite process can be easily achieved. Thus, the starting posture in the PG is always initialized with a squatting posture.

### 6.5. *Modeling process simulation*

The experimental setting is simulated by having a virtual 3D object perceived by a virtual camera. Two examples of generated postures to complete the modeling process are presented in Fig. 9. IN each case, the first posture is set manually and the next ones are generated using our NBV algorithm. The trust region parameters $\rho_{beg}$ and $\rho_{end}$ were respectively set to 0.4 and 10e-5. Other parameters settings are $p = 6$, $\gamma = 20$, $d_{min} = 0.6$, $Nlm_{min} = 5$, $\eta = 1$, $\delta = 1$, $\lambda_z = 200$, $\lambda_x = 80$, $\lambda_y = 80$, $\lambda_d = 100$, $\lambda_l = 1$ and $\lambda_f = 1000$.

In the top part of Fig. 9, the similarity of the postures generated can be explained by considering the size of the object that is relatively large as compared to the robot size. Because of the small field of view of the cameras, the robot needs to be relatively far away from the object, and thus, the height of the cameras have little effect on the perception of the object.

In both examples, the posture generation method sets the arms in a common configuration because they are not used and have little to no influence on the resolution of the constraints.

The modeling process has been simulated several times with various complex shapes. We note that the number of poses necessary to construct the model depends on different parameters: the size and shape of the object, distribution of landmarks on the object surface, termination criterion, presence of obstacles in the environment around the object, etc. Typically, by using an uncluttered environment, a uniform distribution of landmarks on the object surface, and an object size between 40 cm and 3 m, we obtain an average of eight poses to obtain a model where the ratio of unknown to known voxels is below 5 percent.

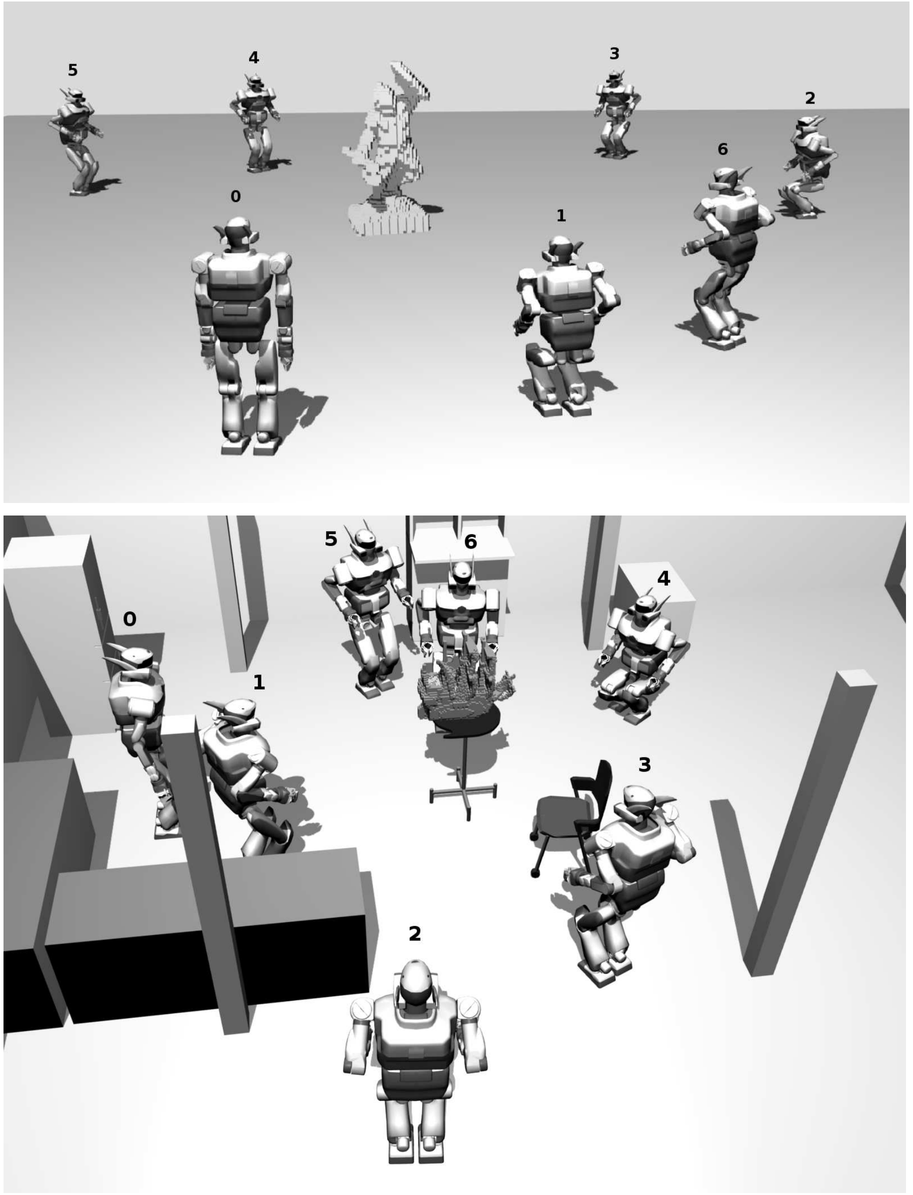

Fig. 9. Postures generated for the reconstruction of two objects. (top) 3-m-high object. (bottom) 70-cm-high object in a cluttered environment.

## 7. Conclusion

This work introduces a new method to automatically generate postures for a humanoid robot depending on visual cues. The algorithm presented differs significantly from previous Next-Best-View solutions in terms of the hypotheses and constraints involved.

The postures are selected amongst the possible configurations allowed by stability, collision, joint limitation and visual constraints, so as to complete the modeling of an unknown object while reducing the number of postures required. Complementary optimization methods, global sampling, NEWUOA, and FSQP, are used in our two-steps algorithm to generate each next-best-posture. The iterative NEWUOA search, coupled with a fixed sampling of the robot head configuration space, can efficiently deal with the noise and discontinuities of the viewpoint evaluation function to minimize. The Posture Generator can then rapidly find a posture satisfying all necessary constraints on the humanoid body. This approach was validated through simulation by successfully building models of various objects having complex shapes and in cluttered environment.

This work is being integrated with other components focused on vision, motion planning, and motion control tasks, in order to experimentally test the autonomous modeling of an object with an HRP-2 robot.

## 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 soldier 3D model used for the tests is provided courtesy of INRIA by the AIM@SHAPE Shape Repository (shapes.aim-at-shape.net).

The experimental setup was visualized using the AMELIF framework[23] and Blender, an open-source software for 3D content creation.

## References

1. O. Stasse, T. Foissotte and A. Kheddar, Treasure hunting for humanoids robot, in *IEEE RAS/RSJ International Conference on Humanoids Robots, Workshop on Cognitive Humanoid Vision* (Daejeon, South Korea, 2008).
2. D. Meger, P-E. Forssen, K. Lai, S. Helmer, S. McCann, T. Southey, M. Baumann, J. J. Little and D. G. Lowe, Curious George: An attentive semantic robot, *Robotics and Autonomous Systems* **56**(6) (2008) pp. 503–511.
3. D.G. Lowe, Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision* **60**(4) (2004) pp. 91–110.
4. J.M. Morel and G.Yu, ASIFT: A New Framework for Fully Affine Invariant Image Comparison, *SIAM Journal on Imaging Sciences* **2**(2) (2009), pp 438–469.
5. H. Bay, A. Ess, T. Tuytelaars and L. Van Gool, SURF: Speeded Up Robust Features, *Computer Vision and Image Understanding (CVIU)* **110**(3) (2008), pp 346–359.
6. J. Sanchiz and R. Fisher, A next-best-view algorithm for 3d scene recovery with 5 degrees of freedom, in *British Machine Vision Conference* (Nottingham, UK, 1999), pp. 163–172.
7. 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 (IROS)* (2007), pp. 1677–1682.

8. D.G. Lowe, Local feature view clustering for 3D object recognition, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2001), pp 682–688.
9. K. Yamazaki, M. Tomono, T. Tsubouchi and S. Yuta, 3-D object modeling by a camera equipped on a mobile robot, in *IEEE International Conference on Robotics and Automation (ICRA)* (2004), pp. 1399–1405.
10. C. Connolly, The determination of next best views, in *IEEE International Conference on Robotics and Automation (ICRA)* (1985), pp. 432–435.
11. J. E. Banta, L. R. Wong, C. Dumont and M. A. Abidi, A next-best-view system for autonomous 3-D object reconstruction, *IEEE Transactions on Systems, Man, and Cybernetics, Part A* **30**(5) (2000) pp. 589–598.
12. J. Maver and R. Bajcsy, Occlusions as a Guide for Planning the Next View, in *IEEE Transactions on Pattern Analysis and Machine Intelligence* **17**(5) (1993) pp. 417–433.
13. R. Pito, A Solution to the Next Best View Problem for Automated Surface Acquisition, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21**(10) (1999) pp. 1016–1030.
14. K.A. Tarabanis, P.K. Allen and R.Y. Tsai, A survey of sensor planning in computer vision, *IEEE Transactions on Robotics and Automation* **11**(1) (1995), pp. 86–104.
15. W.R. Scott, G. Roth and JF. Rivest, View planning for automated three-dimensional object reconstruction and inspection, *ACM Computing Surveys* **35**(1) (2003) pp. 64–96.
16. A. Hertzmann, Introduction to 3D Non-Photorealistic Rendering: Silhouettes and Outlines, in *SIGGRAPH 99 Course Notes* (1999).
17. J-P. Laumond, Kineo CAM: a success story of motion planning algorithms, *IEEE Robotics and Automation Magazine* **13**(2) (2006) pp. 90–93.
18. M.J.D. Powell, The NEWUOA software for unconstrained optimization without derivatives, in *DAMTP Report 2004/NA05* (University of Cambridge, England, 2004).
19. 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.
20. 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 International Conference on Humanoids Robots* (Daejeon, South Korea, 2008), pp. 333–338.
21. A. Escande, S. Miossec and A. Kheddar, Continuous gradient proximity distance for humanoids collision-free optimized postures, in *IEEE RAS/RSJ International Conference on Humanoid Robots* (Pittsburg, USA, 2007), pp. 188–195.
22. 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 International Conference on Humanoids Robots* (Pittsburg, USA, 2007), pp. 151–158.
23. P. Evrard, F. Keith, J-R. Chardonnet and A. Kheddar, Framework for Haptic Interaction with Virtual Avatars, in *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)* (Munich, Germany, 2008), pp. 15–20.

**Torea Foissotte** received his M.S. degree in Computer Science from the Ecole Polytechnique from the University of Tours, France, in 2002, and has since been involved in some computer-vision-related research projects at Postech University, South Korea, and AIST Tsukuba, Japan. He has also worked for a few years on various projects related to computer graphics and computer vision as a software engineer in 3D Incorporated, Japan. Since 2008, he is a PhD Candidate at the University of Montpellier 2, France, and he has joined the JRL in AIST Tsukuba as a research assistant. His interests include computer vision, machine learning, and robotics.

**Olivier Stasse** received the MSc degree in operations research (1996) and the PhD degree in intelligent systems (2000), both from University of Paris 6. He is assistant professor of Paris 13 His research interests include humanoids robots as well as distributed and real-time computing applied to vision problems for complex robotic systems. From 2000 and 2003, he was at the Laboratoire de Transport et Traitement de l'Information (L2TI), and then joined the SONY robotsoccer team of the Laboratoire de Robotique de Versailles (LRV). Since 2003, he has been a member of the Joint French-Japanese Robotics Laboratory (JRL) in a secondment position as a CNRS researcher. He is a member of the IEEE Robotics and Automation Society.

**Pierre-Brice Wieber** graduated from the Ecole Polytechnique, Paris, France, in 1996 and received his Ph.D. degree in Robotics from the Ecole des Mines de Paris, France, in 2000. Since 2001, he has been with the INRIA Rhone-Alpes in the BIPOP team. His research interests include the modeling and control of artificial walking.

**Adrien Escande** received the M.S. degree in 2005 from Ecole des Mines de Paris, Paris, France and the PhD degree in 2008 in robotics from Universite d'Evry Val-d'Essonne, Evry, France after spending three years in the Joint Japanese-French Robotics Laboratory (JRL) in Tsukuba, Japan. Since then, he has been working as a research scientist in CEA-LIST at Fontenay-aux-Roses, France. His current research interests include whole-body planning, control for humanoid robots, and mathematical optimization for robotics.

**Abderrahmane Kheddar** is currently Directeur de Recherche at CNRS and the Director of the CNRS-AIST JRL (Joint Robotics Laboratory), UMI 3218/CRT, he was (1998-2008) professor in computer science and control at the University of Evry and the head of the virtual reality and haptics group of the Laboratoire Systemes Complexes. He received a DEA (Master of science by research) and the PhD degree in robotics, both from the University Paris 6, France. His research interests include teleoperation and telerobotics, haptics (sensing and display), humanoids (contact planning, dynamic control), and electro-active polymers for haptic displays. He was a member of the Teleoperation Research Group under the French Nuclear Commissariat (CEA) and the French National Scientific Research Center (CNRS) auspices. He was the general chair of EuroHaptics 2006 which held in Paris. He is member of the EuroHaptics steering committee and served as a founding member and in the advisory board of the WorldHaptics IEEE TC chapter. He is in the editorial board of the IEEE Transactions on Haptics and the International Journal of Intelligent and Robotic Systems. He also served in the editorial board in many robotics conferences.