Treasure hunting for humanoids robot

O. Stasse [1], T. Foissotte [1], D. Larlus [2], A. Kheddar [1], K. Yokoi [1]

[1] *JRL, CNRS/AIST, ISR, Tsukuba Japan*
[2] *INRIA, Grenoble, France*
{olivier.stasse, torea.foissotte, abderrahmane.kheddar, kazuhito.yokoi}@aist.go.jp
{diane.larlus}@inrialpes.fr

***Abstract*— This paper intends to describe the current status of our group in trying to make a humanoid robot autonomously build an internal representation of an object, and later on to find it in an unknown environment. This problem is named "treasure hunting". In both cases, the main difficulty is to be able to find the next best position of the vision sensor in order to realize the behavior while taking care of the robots limitation. We briefly describe the models and the processes we are currently investigating in building this overall behavior. Along the description we stress the current key problems faced while trying to solve this problem.**

I. Introduction

A. Context of the work

The works presented in this paper are parts of an ongoing project called 'treasure hunting', where the robot should retrieve **autonomously** an object in an unknown environment [1] based on a model that it **autonomously** build and stored [2] during a previous phase. This work takes its foundation upon a previous work [3] where some parts not described here (i.e. the software structure and the motion generation) were detailed more specifically. In this paper, we describe more specifically the two problems of object visual model construction and visual search. In the first case, a new optimization problem and its resolution are introduced. This allows to find a pose for a HRP-2 humanoid robot which maximize the unknown surface perceived with the robot's stereoscopic while coping with all the humanoid robot constraints. In the second case, we investiguate the problem related to visual recognition and research strategies when dealing with the images taken in the first phase.

II. Object Visual Model building

A. Overview of related work

Many existing works focus on the environment exploration [4] or object recognition problems [5]. The modeling part usually relies on a supervised method where different views of an object are taken manually by a human and served as an input to the algorithm. A number of works are dedicated to planning of sensor positions in order to create a 3D model of an unknown object, see for example [6], [7] or [8]. Hypothesis and limits of such works are detailed in these two surveys: [9] and [10]. 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 or on a turntable, 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 is to get an accurate 3D reconstruction of an object, using voxels or polygons, while reducing the number of viewpoints required.

B. Contribution

Though our modeling process also requires a Next-Best-View (NBV) solution, it appears that working hypothesis are quite specific for a humanoid robot. Our approach looks similar to the works of [11], or [6], as we also rely on an occupancy grid and a space carving method, but it still differs in few important ways:

1) the limits of the sensor pose are constrained due to it being 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 positionning errors.
2) the sensor possible positions are not constrained to some precomputed discrete positions on a sphere surface, and its viewing direction is not forced toward a sphere center
3) an accurate 3D model of the object is not required. Our goal is to get a set of SIFT around the object to allow its effective detection and recognition.

In [2], 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 the posture generator proposed in [2]

III. Two steps NBV approach

Traditional works in the NBV field reduce the problem dimensionality and sample the configuration space in order to retrieve a solution in an acceptable amount of time without relying on the gradient.

In order 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 [13], a method that can find a function minimum by refining a quadratic approximation of the function through a deterministic iterative sampling, and which can be used for non-derivable functions. The sampling positions at each step in the iteration process are selected according to the previous sampling results and the state of the actual quadratic approximation. Moreover they are limited to vectors inside a trust region, which is defined relatively to two radius parameters: ρ_{beg} and ρ_{end}, and a given starting vector, which will be the camera pose in our case. NEWUOA has the advantages of being fast and robust to noise while allowing 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 negligeable influence on the convergence of NEWUOA. An OpenGL rendering of the occupancy grid was thus implemented by displaying voxels as cubes whose color corresponds to one of the two possible states: "known" and "unknown". The amount of unknown visible, noted N_{up}, is then equal to the number of pixels of the color related to "unknown" state present in the framebuffer. For such purpose, voxels' normals and lighting functionalities of OpenGL are not used, which allows to speed up the computation. Further optimization can be achieved by storing voxels data in the graphic card memory.

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:

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

(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 center of the occupancy grid. This corresponds to a requirement in order to generate the disparity map with the two cameras embedded in the robot head. (3) and (4) restricts the rotations on X and Y axises to ranges manually set according to the robot particularities. Finally (5) constraint keeps a minimum number of landmarks, i.e. features that were detected in previous views, visible from the resulting viewpoint. By matching them with features 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.

The landmark visibility constraint is currently implemented by assigning a unique color to each landmark and setting all corresponding voxels to the appropriate color. If there is a sufficient amount of pixels with a specific landmark color, then the landmark can be considered as visible.

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 can correspond 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 beween the camera and the object is formulated as:

$$K_d = \exp^r(\gamma\ (d_{min} - d(\mathbf{C}, \mathbf{og_{center}}))) \tag{7}$$

where γ and r parameters are set manually.

For the landmark visibility constraint, the formulation depends on two cases. When N_l is greater or equal to N_{lmin}, configurations maximizing N_l are slightly encouraged :

$$K_l = \beta\ (N_{lmin} - N_l) \tag{8}$$

The β parameter influences how important is the maximization of N_l in the optimization process. Its value should be low enough so that the maximization of N_{up} stays the main priority of the algorithm.

In the other case, configurations where N_l is less than N_{lmin} are greatly penalized:

$$K_l = \exp(\delta\ (N_{lmin} - N_l)) + \delta\, d(\mathbf{C}, \mathbf{C}^p) \tag{9}$$

The penality is expressed through the δ value and gets larger when the camera moves away from previously validated position by using $d(\mathbf{C}, \mathbf{C}^p)$, the distance between the actual camera position, $\mathbf{C}$, and the closest of the previous camera positions where landmarks have been detected, $\mathbf{C}^p$.

The evaluation function used as input to the NEWUOA algorithm is then:

$$f_e = \lambda_z K_{\mathbf{C}_z} + \lambda_x K_{\mathbf{\Theta c}_x} + \lambda_y K_{\mathbf{\Theta c}_y} + \lambda_d K_d + \lambda_l K_l - \lambda_n N_{up} \tag{10}$$

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

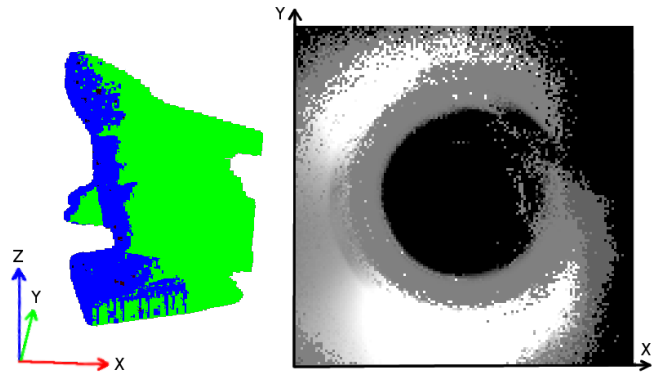

Fig. 1. (right) Best constrained visibility of unknown obtained depending on fixed camera XY positions around a carved object (left). Clearer color mean better results.

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 sampling to a subspace according to a range given by the user. Due to the constraints used, many different cases can result in local minimums in our evaluation function that are quite disjoint as can be seen in the example shown in Fig. 1. This figure shows the best results for f_e obtained for a soldier statue carved once and using different sampled values of $\mathbf{C}_x$ and $\mathbf{C}_y$. Darker points means worse evaluations. Known voxels are represented as blue on the displayed object, and unknown voxels in green. We can observe discontinuities in the evaluation results due principally to the distance to the object constraint, e.g the black zone in the center of the image, and the landmark visibility constraint, e.g the black zones on the right and on the top-left corner. In such cases, the quadratic model cannot be pertinent if the trust region is too big.

In our actual implementation, the optimization process is biased by setting the starting pose of the camera to a pose deduced from previous ones, and by limiting the trust region size. When an optimum is found, NEWUOA is run again by using the 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 starting one. Another way to improve the results is to choose a set of possible starting poses around the object and launching the optimization process for each of them. Results of all optimizations are then compared to select the best camera pose.

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.

The head posture is fixed by setting the embedded left camera, with position $\mathbf{h}_p$ and coordinate sytem vectors $(\mathbf{h}_i, \mathbf{h}_j, \mathbf{h}_k)$, to the desired pose, with position $\mathbf{C}$ and coordinate sytem vectors $(\mathbf{C}_i, \mathbf{C}_j, \mathbf{C}_k)$, using:

$$\begin{cases} ((\mathbf{C}+\mathbf{C}_i)-\mathbf{h}_p).\mathbf{C}_j = 0 & (11)\\ ((\mathbf{C}+\mathbf{C}_i)-\mathbf{h}_p).\mathbf{C}_k = 0 & (12)\\ \mathbf{h}_k.\mathbf{C}_j = 0 & (13)\\ \mathbf{h}_k.\mathbf{C}_k \geq 0 & (14)\\ \mathbf{h}_i.\mathbf{C}_j = 0 & (15)\\ \mathbf{h}_i.\mathbf{C}_k = 0 & (16)\\ \mathbf{h}_i.\mathbf{C}_i \geq 0 & (17)\\ d(\mathbf{C},\mathbf{h}_p) \leq \epsilon_d & (18) \end{cases}$$

For this algorithm, the objective function for the PG is not necessary. Nevertheless we use it as an esthetic criterion to place the robot posture close to a reference posture where joints are set to one quarter of their possible range from their minimum limit. The associated non-linear optimization problem to is solved using FSQP as described in more detail in [2].

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

We tested the influence of the trust region parameters on the optimal found with NEWUOA. The parameter ρ_{beg} sets the maximum variation that can be taken by the camera pose parameters, 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 process with different values for ρ_{beg} and ρ_{end}. This was repeated several times with different poses in order to check if some specific values result in a convergence of NEWUOA toward a better pose in most cases.

During our tests, it generally took NEWUOA between 1 and 3 seconds to find a minimum with an average computer. This is quick enough to select and test different starting poses in order to find a good Next-Best-View.

A. Modeling process simulation

We implemented the first step of our algorithm using a C version of the original FORTRAN code published by Powell [13]. The experimental setting is simulated by having a virtual 3D object perceived by a virtual camera. The modeling process loop the following steps:

1) The disparity map is constructed using the object 3D informations 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 then called in order to find an optimal camera pose by minimizing our evaluation function. In our current implementation, the starting camera pose for NEWUOA is selected by rotating the previous camera orientation 90 degrees on the Z axis

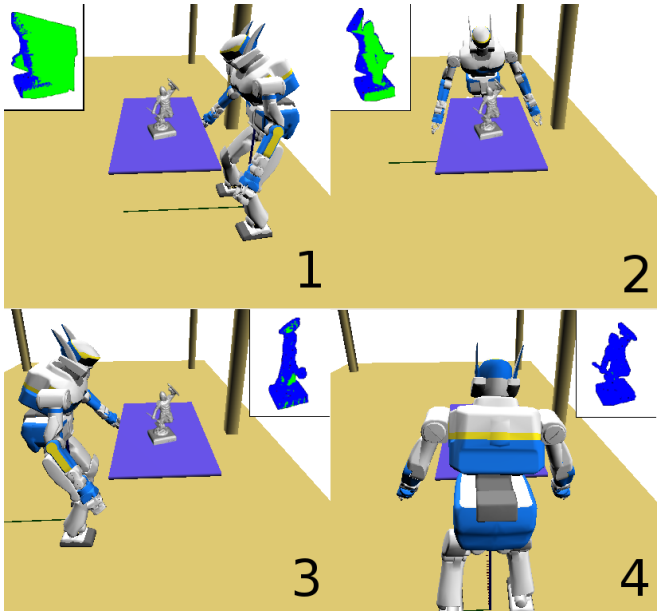

Fig. 2. Postures generated successively for the modeling of an unknown object

and by positioning the camera at the same distance from the object than the previous pose while the new view vector is pointing at the occupancy grid center. Other starting poses are generated by using a fixed sampling of 3D positions centered on the computed one. The view vectors are set toward the occupancy grid center for each starting pose.

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 a space carving operation, i.e the unknown voxels cannot be perceived due to the constraints on the robot. An example of postures generated during a successful modeling process is illustrated in Fig. 2 with the updated occupancy grid at each step.

B. Pose generation

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

During our tests, we could confirm 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 standup 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.

Fig. 3. Postures generated using our NBV algorithm

Some of the whole-body postures obtained with the PG were played with OpenHRP2 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. 3.

V. Conclusion and Future Work

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. An extended version of this work has been submitted at ICRA 2009 [14].

The next target for our work is to use a planner which is not based on a A^* as proposed by Kuffner [15] and Bourgeot [16]. Indeed such kind of planner usually create an unwelcome stepping because the robot has its action set decreased artificially to facilitate the path search resolution. In the case of the visual search, as described in paragraph VI-G, this increase very much the realization time of the behavior.

VI. Object Visual Search

The object visual search has recently regain some interest especially with new contest such as the Semantic Robot Challenge [17] which interestingly is taking place in the Computer Vision community. The 2007 contest winners described their architecture in [18][19].

A. Visual recognition

1) SIFT model based approach: The object model used by the robot consists of all the 3D features that had been spotted during the learning phase, moved to a unique frame of reference. What follows explains how such a representation is used for object recognition.

First, feature detection is run on the scenery. The resulting features are then matched between the scene and the object,

in the same way as it had been done for pairs of views during the learning phase. Rigid motion evaluation is then performed with unlikely matches cast aside.

The results for close-up scenes (up to 1 meter) are excellent, but worsen when the distance increases. In order to measure the influence of distance on this algorithm, object detection was run from many distances, in two different experiments: with the object alone on a black background, and in a heavily cluttered environment.

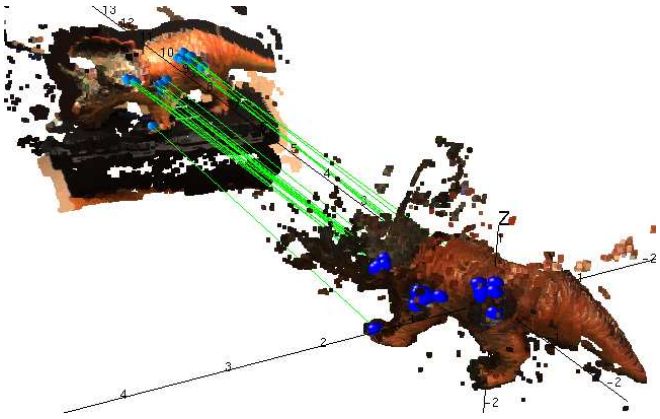

Fig. 4. A screenshot of the model successfully detected in the range map of the scenery. It contains over 6000 3D SIFT features, but only the best matches are represented.

Fig. 5. Left : The left eye's image of figure 4's scenery. Right : The pose of the object is successfully determined using Rigid Motion minimization.

Beyond 2 meters, the object can still be located in the scene's 3D map, but the pose estimation fails. This is due to the position error of the disparity map's 3D points. As specified [2], our approach uses geometric information to check the geometrical relationship between the landmarks. Thus we do not have problem related to the rotation of the features as in a monocular approach [18] provided than enough landmarks are detected during the visual model construction.

B. Seeing far away: a generative model based approach

As the sift-based reconstruction method fails at detecting objects far away, a method [2] was presented which aims at providing coherent hypothesis of the object position and scale in the robot field of view. It can detect object in challenging conditions, such as difficult viewpoints, small scale, extreme illumination conditions and occlusions. This hypothesis can be used as an input for the visual search when the 3D object reconstruction fails. It is an extension of the method of [20] and uses additional information coming from the robot to guide the model estimation process. In particular we will use both the left and right images of the robot cameras to compute dense disparity maps and then use the resulting depth information as an extra component of the model.

C. Visual Features

Images are represented by a set of n overlapping patches and a gradient map (see figure 6).

Overlapping visual patches. Patches, denoted $\mathcal{P}_i, i \in \{1, \ldots, n\}$, are sets of pixels belonging to square image regions. Five different characteristics are computed from each patch.

First of all, a visual codebook is obtained by k-means clustering SIFT [21] based representations of the patches. Then, each patch $\mathcal{P}_i$ is associated to the closest codeword. The assigned codeword is denoted w_i^{sift}; this is the first characteristic. We also produce visual words based on color information by clustering color descriptors [22]. The patch $\mathcal{P}_i$ is also characterized by its closest color codebook word w_i^{color}. A RGB value is computed by averaging over pixels extracted in the center of the patch. This 3D-vector is denoted rgb_i. We also consider the coordinates of the patch center $X_i = (x_i, y_i)$ in the image. Finally, the dense disparity map provides an estimation of the depth d_i of the patch.

Gradient Map. In addition to this patch based characteristics computation, we also extract a gradient map $\mathcal{G}(x, y)$, that consists of the strength of the gradient at each (x, y) pixel location.

In the end, the gradient map $\mathcal{G}(x, y)$ and the characteristics of the n overlapping patches $\mathcal{P}_i$: $\{w_i^{sift}, w_i^{color}, rgb_i, X_i, d_i\}$, $i \in \{1 \cdots n\}$ compose all the information we use to describe an image.

D. Model description

The strength of our model lies in the combination of two (different but) complementary components: (i) a blob based generative model using visual words for its good object localization properties, and (ii) a MRF (Markov Random Field) structure which provides a coherent field of labels following object boundaries.

1) A blob-based generative model: We consider that an image is made of "blobs", and that each blob generates some patches with its own model. Intuitively, if an image contains three objects, we may have three blobs, one over each object region. Each blob is thus responsible for generating a set of patches the appearance of which corresponds to the object category.

The generation of a patch requires to a) select a blob, and b) generate a patch with the patch model *specific* to that blob.

The blob generation is assumed to follow a Dirichlet process. The Dirichlet process exhibits a self-reinforcing property: the more often a given value has been sampled in the past, the more likely it is to be sampled again. This means that each newly generated patch can either belong to an existing image blob B_k or start a new region.

We characterize each blob $B_k, 1 \leq k \leq K$, with a set of random variables: $\Theta_k = \{\mu_k, \Sigma_k, C_k, l_k, N_k, S_k\}$. μ_k, Σ_k are respectively the mean and the covariance matrix describing the geometric shape of the blob, l_k is the blob label (object category), C_k is a Gaussian mixture model representing the colors of the blob, N_k is the number of patches generated by the blob, S_k is the scale of the blob which is closely related to the distance between the object and the camera.

We characterize each patch $\mathcal{P}_i$ by its features $(w_i^{sift}, w_i^{color}, rgb_i, X_i, d_i)$.

The probability of generating a patch, given that it is generated by the blob B_k of parameters Θ_k: $p(\mathcal{P}|\Theta_k)$ is made of 5 distinct parts, as the model assumes that patch position and scale, color and appearance are independent for a given blob. The probability for a blob B_k to have generated patch $\mathcal{P}$ thus consists of five terms:

$$\begin{aligned} p(\mathcal{P}|\Theta_k) &= p(w^{sift}, w^{color}, rgb, X, d|\Theta_k) \\ &= p(w^{sift}|\Theta_k)p(w^{color}|\Theta_k) \\ &\quad p(rgb|\Theta_k)p(X|\Theta_k)p(d|\Theta_k) \end{aligned} \tag{19}$$

The position X of a patch is chosen according to a normal distribution of parameters μ_k and Σ_k for object blobs. It is uniform for background blobs.

We assume that background and object blobs have a Gaussian Mixture color model. The patch depth is closely related to the blob size. Finally, the probability of the SIFT and color codewords only depend on the class label. These distributions encode object appearance information and are responsible for the recognition ability of our model. They are learned using training images in a way described later on (section VI-F).

2) A MRF structured field of blob assignment: A MRF of blob assignment regularizes the assignment of neighboring patches and also aligns borders between the object and the background with natural image contrast and with strong depth changes. This field is defined over a grid (8-connectivity), nodes correspond to patch centers.

This component basically defines a Gibbs energy that is used to compute conditional probability of patch assignment. This energy has a model fitting term based on the blob representation previously defined as well as neighboring constraint terms for spatial regularization.

The total energy E of the field is the sum of local energies E_i defined for each patch $\mathcal{P}_i$

$$E_i = U_i + \gamma \sum_{j \in \mathcal{N}(i)} V_{i,j} \tag{20}$$

where $\mathcal{N}(i)$ represents the 8 neighbors of $\mathcal{P}_i$, γ balances the proportion of the two terms. Let b_i be the blob assignement index of patch $\mathcal{P}_i$. $U_i = -\log p(b_i|\mathcal{P}_i, N_{1:K}, \Theta_{b_i})$ is a potential that measures the coherence between the patch and the blob model, and $p(b_i|\mathcal{P}_i, N_{1:K}, \Theta_{b_i})$ is the probability of the blob assignment knowing the patch and the parameters of all the blobs. It stems from the model presented in the last section and makes the link between the two components of the model. $V_{i,j}$ is a potential that measures similarity between two patches $\mathcal{P}_i$ and $\mathcal{P}_j$. It enforces local coherence of the object/background labels, via constraints on the similarity of neighboring patch labels. These contraints are computed using the gradient map $\mathcal{G}$ and the distance between depth values of neighboring patches. It encourages cuts along high image gradients and depth discontinuity.

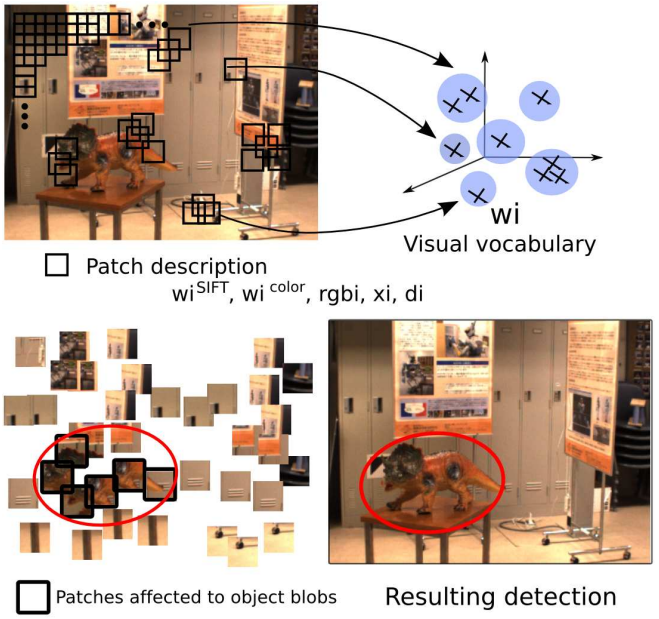

Fig. 6. First row: patches are extracted in a very dense manner. Each patch is associated to the closest visual word for sift and color descriptors, and then represented by the words indexes $(w_i^{SIFT}, w_i^{color})$, a RGB value (rgb_i), a position (x_i) and a depth (d_i) given by the disparity map. Second row: the model computes the best assignment of patches to object blobs or background and estimates to blobs positions.

E. Model Estimation

Now that the model has been defined, its parameters have to be estimated for each image to produce object/background blobs labels (l_i) and patch assignments to blobs (b_i). The model is estimated by a Gibbs sampling algorithm [23] (specific case of Markov Chain Monte Carlo (MCMC) method). A Gibbs sampler generates an instance of parameter values from the distribution of each variable in turn, conditional on the current values of the other variables. More details on the model estimation could be found in [20].

F. Learning an object appearance

In order to learn the object appearance information, examples of images containing the object are fed to the robot. Once again, these learning images are stereoscopic views, taken from several viewpoints. The resulting dense disparity map provides local information that we use to create segmentation masks on positive images. This makes

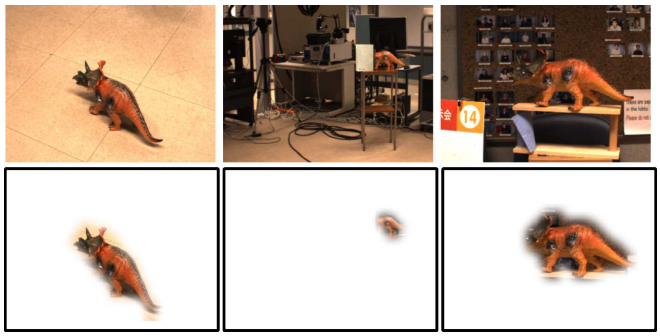

Fig. 7. The model gives a list of patches actually being components of the model. This produces segmentation masks.

the estimation of object model more accurate by knowing exactly which part of the image belongs to the object and which does not.

We also use a set of negative images (*ie* not containing the object) provided by the robot camera while moving in its environment.

Descriptors (SIFT + color) are extracted on local regions exactly as described for the test images. These descriptors are used first to create visual words by a quantification process, and then to compute the probability for each visual word to be observed as a component of an object blob or not. These probability distributions ($p(w^{sift}|\Theta_k)$ and $p(w^{color}|\Theta_k)$) are stemed from an occurrence histogram obtained by a counting process.

The model also provides the list of patches belonging to a particular object instance. The patches correspond to sets of pixels belonging to their support. Using the information on all patches containing a given pixel, we can create a segmentation of the object. Figure 7 provides segmentation masks in terms of probability maps of the object location on images where the detection succeeded.

G. Global strategy

In this section we present a high-level behavior which relies on all the previously presented functionnalities to reason and take autonomously a decision in order to find an object. Our main contribution is to introduce the constraints related to the walking algorithm and the recognition system into one entity called the *visibility map* to reduce the space of the sensor configuration.

1) Introduction: Sensor planning to find a known object in an unknown environment using vision with a mobile platform is an old problem which received a lot of attention during the 80s. Most of this work relied on the use of a range finder coupled with a camera, whereas the object model was either a polyhedron, a 3-edge based representation or a voxel grid description. Even so the recognition process is still valid, and used in recent humanoid applications by Neo [24] to achieve autonomous behavior, it is interesting to revisit this problem in the context of humanoids. Indeed new available hardware such as multi-core CPUs make efficient implementation of such algorithms possible. However as this problem is NP-complete, simplified models are still necessary to simplify

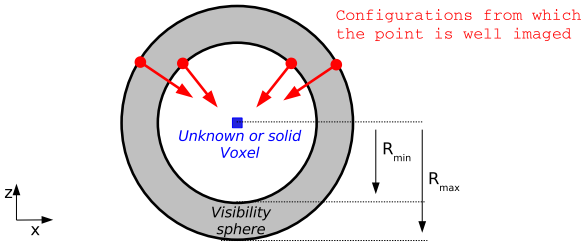

Fig. 8. Visibility sphere for a given 3D point.

the search. Finally, the motion capabilities of humanoid robots, instead of the classical 2D representation of the search space, requires the adoption of a 3D representation.

2) The simplified model: In this work we consider mainly the problem of finding the best next camera pose to search an object in an unknown environment. Here the camera pose is given by a 3D position plus an orientation provided by pan and tilt values, which give us a five dimensional space. In order to simplify the problem we [1] take into account two considerations: the robot's motion capabilities, the recognition system's characteristics. Depending on the task different recognitions can be used, as we have at our disposal either a 3D-edge model [25] or a Spin-Image [26].

The first consideration allows to limit the domain of the pan and tilt values according to the joint limits. Moreover if we consider only the case where the robot walks[1], the vertical axis can be deduced from the constraint on the CoM. The second consideration implies to use a model of the recognition system. But in addition to the classical statistical model, we consider that they are practical bounding values $(R_{\min}, R_{\max})$ for which the recognition system is able to work. From this additional information found experimentally and which vary according to the object we proposed the concept of *visibility map*.

3) The visibility map: In order to explain what visibility map is, we shall introduce the concept of *visibility sphere*. Let us assume that we have a partial representation of the world using a voxel grid representation. A visibility sphere is the set of poses for which an unknown or solid point of the voxel grid is seen within the perception interval $(R_{\min}, R_{\max})$. A *visibility map* can then be defined by the intersection between the constraints related to the robot's motions and the visibility spheres centered on each voxel of the world map. When the robot walks, the height of the camera is fixed, so the *visibility map* is a plane going through the head of the robot.

4) Planning: Once the visibility map has been made, the next step is to chose the best candidate location to search the object while taking account three quantities: a cost for motion, the new information and the detection probability. The motion cost (MC) is an approximation of the cost to reach a particular pose. It is based on the Chamfer distance and a specific weighting of each DOF. The new information (NI) is quantified by projecting the environment grid onto the camera pose candidate. It also includes a likelihood of occlusion. This part, which is the most costly, can be easily

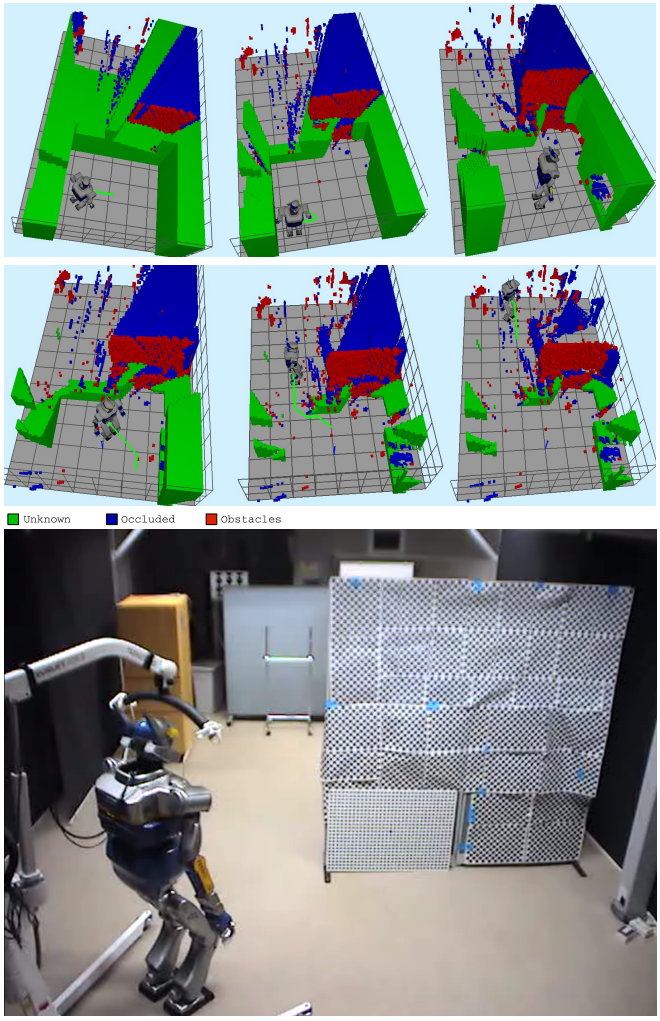

Fig. 9. 3D reconstruction and the real environment in which the robot evolves.

paralleled using multi-core architecture, or even with a GPU. Finally the detection probability (DP) for any given voxel is built upon the probability that this voxel belongs to the target, and the resolution at which it is perceived. Those three quantities are combined together in the rating function:

$$RF = \alpha_{DP}.DP + \alpha_{NI}.NI - \alpha_{MC}.MC \qquad (21)$$

The weights α balance the contributions between a wide exploration of the environment and a deep search of each potential target. A detailed explanation of this approach can be found in [1].

5) Integration: It is important to notice that the key to reduce the search space is the concept of the visibility map which includes the constraints related to the walking algorithm. Here more particulary this constraint is the height of the CoM. Moreover this module is at the highest level of abstraction and relies totally on the other modules to perform the full behavior as depicted in figure 9. Once the next best view is decided, it uses the path planner to feed the motion generator with appropriate foot steps and posture. The reasonning is performed on a visual reconstruction of the world .

VII. Conclusion

We have presented our current status in trying to have a robot building automously a representation of an object and finding it back in an unknown environment. In our approach we have try so far to make as few assumptions as possible, but to use all the knowledge available on the robot and its control structure. We still have some problems related to the drift of the robot while realizing this complex overall behavior partly due to the poor quality of the A^* planner, and because of the inherent floor reaction when trying to perform complex actions. Our current work is trying to address those issues.

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 tests is provided courtesy of INRIA by the AIM@SHAPE Shape Repository `shapes.aim-at-shape.net`.

The visualization of the experimental setup relied on the AMELIF framework presented in [27].

References

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

[2] 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.

[3] O. Stasse, B. Verrelst, A. Davison, N. Mansard, F. Saidi, B. Vanderborght, C. Esteves, and K. Yokoi, "Integrating walking and vision to increase humanoid autonomy," *International Journal of Humanoid Robotics,special issue on Cognitive Humanoid Robots*, vol. 5, no. 2, 2008.

[4] J. Sanchiz and R. Fisher, "A next-best-view algorithm for 3d scene recovery with 5 degrees of freedom," in *British Machine Vision Conference*, 1999.

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

[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] R. Pito, "A solution to the next best view problem for automated surface acquisition," *Ieee Transactions on Pattern Analysis and Machine Intelligence*, 1999.

[8] 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.

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

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

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

[12] T. Foissotte, O. Stasse, A. Escande, and A. Kheddar, "Towards a next-best-view algorithm for autonomous 3d object modeling by a humanoid robot," in *The 26th Annual Conf. of the Robotics Society of Japan*, 2008.

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

[14] T. Foissotte, O. Stasse, P. Wieber, A. Escande, and A. Kheddar, "A two-steps next-best-view algorithm for autonomous 3d object modeling by a humanoid robot," in *ICRA*, 2009, p. submitted.
[15] P. Michel, J. Chestnutt, J. J. J. Kuffner, and T. Kanade, "Vision-guided humanoid footstep planning for dynamic environments," in *Proc. IEEE/RAS Int. Conf. on Humanoid Robotics (Humanoids'05)*, 2005, pp. pages 13–18.
[16] J.-M. Bourgeot, N. Cislo, and B. Espiau, "Path-planning and tracking in a 3D complex environment for an anthropomorphic biped robot," in *IEEE/RSJ IROS*, 2002, pp. 2509–2514.
[17] "The semantic robot challenge." [Online]. Available: http://www.semantic-robot-vision-challenge.org/
[18] P.-E. Forssén, D. Meger, K. Lai, S. Helmer, J. Little, and D. G. Lowe, "Informed visual search: Combining attention and object recognition," in *ICRA*, 2008, pp. 935–942.
[19] D. Meger, P.-E. Forssén, 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 Journal*, vol. 56, no. 6, pp. 503–511, June 2008.
[20] D. Larlus and F. Jurie, "Combining appearance models and markov random fields for category level object segmentation," in *IEEE Conference on Computer Vision & Pattern Recognition*, 2008.
[21] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 4, pp. 91–110, 2004.
[22] J. van de Weijer and C. Schmid, "Coloring local feature extraction," in *ECCV'06*, 2006, pp. 334–348.
[23] R. M. Neal, "Markov chain sampling methods for dirichlet process mixture models," University of Toronto, Tech. Rep. 9815, sep 1998.
[24] E. S. Neo, K. Yokoi, S. Kajita, F. Kanehiro, and K. Tanie, "A switching command-based whole-body operation method for humanoid robots," *IEEE/ASME Transactions on Mechatronics*, vol. 10, no. 5, pp. 546–559, 2005.
[25] Y. Sumi, Y. Kawai, T.Yoshimi, and T. Tomita, "3d object recognition in cluttered environments by segment-based stereo vision," *International Journal of Computer Vision*, vol. 6, pp. 5–23, January 2002.
[26] O. Stasse, S. Dupitier, and K. Yokoi, "3d object recognition using spin-images for a humanoid stereoscopic vision system," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, Beijing, China*, October 9-15 2006, pp. 2955–2960.
[27] 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.