Towards Autonomous Object Reconstruction for Visual Search by the Humanoid Robot HRP-2

Olivier Stasse, Diane Larlus, Baptiste Lagarde, Adrien Escande, François Saidi, Abderrahmane Kheddar, Kazuhito Yokoi, Frédéric Jurie

▶ **To cite this version:**

Olivier Stasse, Diane Larlus, Baptiste Lagarde, Adrien Escande, François Saidi, et al.. Towards Autonomous Object Reconstruction for Visual Search by the Humanoid Robot HRP-2. Humanoids, Nov 2007, Pittsburgh, United States. pp.151-158, 10.1109/ICHR.2007.4813862 . inria-00548676

HAL Id: inria-00548676
https://inria.hal.science/inria-00548676

Submitted on 6 Jan 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Towards Autonomous Object Reconstruction for Visual Search by the Humanoid Robot HRP-2

O. Stasse[1], D. Larlus[3], B. Lagarde[1], A. Escande[1],
JRL CNRS[1], JRL AIST[2]
AIST-Central 2, Umezono 1-1-1, Tsubuka
Ibaraki 305-8568, Japan
Email: {first_name.family_name}@aist.go.jp

F. Saidi[1], A. Kheddar[1], K. Yokoi[2] and F. Jurie[3]
LEAR[3], INRIA Rhone-Alpes
655 avenue de l'Europe
38 334 Saint Ismier Cedex, France
Email: {diane.larlus,frederic.jurie}@inrialpes.fr

***Abstract*—This paper deals with the problem of object reconstruction for visual search by a humanoid robot. Three problems necessary to achieve the behavior autonomously are considered: full-body motion generation according to a camera pose, general object representation for visual recognition and pose estimation, and far-away visual detection of an object. First we deal with the problem of generating full body motion for a HRP-2 humanoid robot to achieve camera pose given by a Next Best View algorithm. We use an optimization based approach including self-collision avoidance. This is made possible by a body to body distance function having a continuous gradient. The second problem has received a lot of attention for several decades, and we present a solution based on 3D vision together with SIFTs descriptor, making use of the information available from the robot. It is shown in this paper that one of the major limitation of this model is the perception distance. Thus a new approach based on a generative object model is presented to cope with more difficult situations. It relies on a local representation which allows handling occlusion as well as large scale and pose variations.**

I. Introduction

This work is part of a project we called "Treasure Hunting".

A typical scenario would be to exhibit an object to the robot in order for it to build an internal representation. In a second step the object is placed somewhere in a given environment; the robot is then asked to seek for it and find it in an autonomous way.

We believe this behavior is fundamental for complex autonomous robotic systems aiming at helping humans in working place such as factories or offices. Our research addresses this problem in an integrated manner *i.e.* considering the capabilities and limitations of the robot and the interaction between object representations and its evaluation while searching. Part of the visual search behavior has been thoroughly described in Saidi *et al.* [1]; it assumes a system with several levels to recognize and process a given object. We formulate this assumption relying on two approaches:

- a 3-D SIFT-based representation of an object for construction and recognition: this is used for close distances, and
- a generative model approach for far-away object detection.

A posture generation devoted to this problem is proposed. Its role is to feed the recognition modules with necessary images to build the internal representation of a given object. The posture generation is optimized to take into account robot constraints while guided by specific requirements in building object representation.

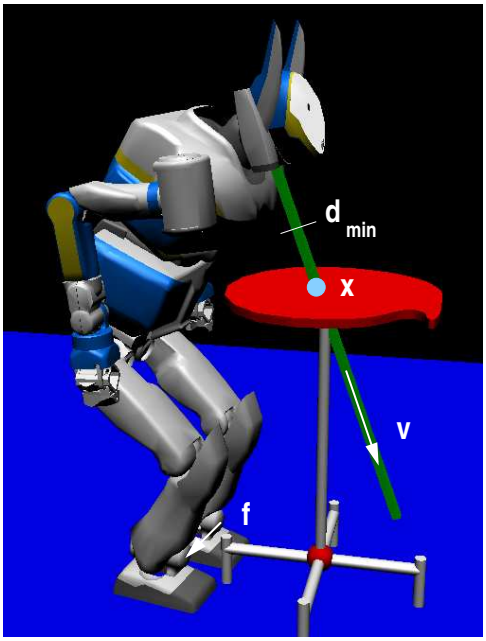

Fig. 1. Parameters considered to generate a pose for robot HRP-2, given point x and a direction $\mathbf{v}$ to look at. The rotation angle around $\mathbf{v}$ is free.

A. Overview of related work

1) Object learning and recognition on Humanoids: Fitzpatrick et al. [2] proposed to use active perception to have a humanoid robot learn a visual representation of the object. By light tapping, the humanoid robot Cog segments the boundaries of the object, and builds an orientation based representation of the object. This representation is invariant by scale and orientation; however it is not intended to handle 3D representation, and mostly targeted for closed-range interaction. Ude et al. [3] proposed to use a classification on a Gabor filter representation of objects. The classification is realized by a SVM classifier. It is also associated with detectors such as colors to elect candidates in the environment. Ude et al. uses

the wide view cameras of the DB humanoid robot to elect possible candidates, and brings them in the foveated images. Extending on this work, Welke et al. [4] uses a feed-forward neural network to learn the feature representations of an object model. There is however no autonomous mechanism to learn the visual features of an object as for Fitzpatrick. Of particular interest is the work of Taylor and Kleeman [5], where visual object models are build using a range laser stripe, and fit into geometric primitives; their robot was able to build a high-level representation of the scene through a graph, and then perform grasping in a cluttered environment.

2) Object detection: Object detection has receive a lot of attention during the last decades. Most of the methods use sliding windows techniques. A classifier is trained in order to decide if a region contains an object or not and is applied at all possible positions and scales. We can cite among others, the cascade of classifiers proposed by Viola and Jones [6]. These methods are known to be efficient but need a huge number of training images (few thousands) and require a very long training stage.

We decided to focus on a generative model based on local representation. Models considering images as collection of small patches have been used first for image retrieval and more recently for object classification. A popular technique is to use a quantification of these local representations into so called visual words (first introduced by [7]). These techniques can also be applied to the localization of objects in the image. They are particularly adapted when dealing with strong variations of object appearances and occlusions.

B. HRP-2 vision system

HRP-2 Number 10 has been modified to have three cameras with a narrow field-of-view (25 degrees), and a fourth camera with a large field-of-view (90 degrees). The large field-of-view camera has already been used for SLAM [8], whereas the narrow field of view is used for 3D edge-based object recognition [9]. Looking down to an object is quite challenging for such kind of robot, mostly because there are not so many features other than the object itself, even in the wide-lens camera. Secondly, the table we used has a round shape, which tends to foster bad localization of the corner features which slide along the border of the table. The information of the pattern generator is moreover not as good as in the usual case, because sideways motion causes much friction with the floor. Therefore they create a large drift when turning 360 degrees. It is thus not possible to use directly the information from our previously developed SLAM system to create an accurate model of an object, we rather used other information available to build the description of our object. In order to make the problem challenging compared to the previous well-known methods we used a dinosaur as the object to reconstruct. It is well textured, but its complex geometry makes 3D-edge based methods likely to fail, and manual modelling would be quite time-consuming.

II. Full Body Pose generation for a viewing direction

We describe the algorithm which generates the pose according to a viewpoint provided by a Next Best View (NBV). For the time being we assume a simple NBV algorithm that provides directions around a sphere bounding the object of interest. The problem is to find feasible poses considering: (i) the constraints due to object reconstruction such as the distance to the object, the viewing direction... and (ii) constraints inherent to the robot such as stability, joints limits, collision avoidance, etc. Our posture generation borrows from the one presented for the purpose of humanoid contact-support planning [10]. It appears that this posture generator can also be used for our vision reconstruction purposes, by reformulating parts of the optimization problem in a way it takes into account our application requirements.

A. Optimization problem

As depicted in figure 1 the inputs of the problem are a direction $\mathbf{v}$, a point $\mathbf{x}$, an interval $[d_{\min}, d_{\max}]$ and a 4×4 matrix $\mathbf{f}$. The output is a pose $\mathbf{q}$, for which the robot has its vision system frame aligned with $\mathbf{v}$, looking through $\mathbf{x}$. The vision system along the line should be at a distance between $d_{\min}$ and $d_{\max}$. Moreover the robot has to be statically stable, i.e. its CoM is projected into the support polygon, without any self-collision or collision with the environment. The matrix $\mathbf{f}$ specifies the pose of the right foot in the left foot reference frame. $\mathbf{q}$ is the pose of the free floating body, and the angular values. In Escande et al. [10], the objective function is used to obtain a more natural-looking posture by minimizing the distance between the joint angles and the middle of their limits. The stability is introduced as a set of inequalities computed from the convex hull of the foot. Depending on the starting point for difficult viewpoint, this might lead to solution where the CoM is on the limits of the polygon, cf figure 2. Therefore the objective function was changed to be the distance of the CoM from the center of the feet polygon. In order to solve the optimization problem, we used CFSQP [11] which can cope with linear and non-linear equalities and inequalities constraints

B. Constraints

1) Collisions and self-collisions avoidance: Collisions are introduced through non-linear inequalities using proximity distances between selected pairs of body. In order to compute a continuous gradient, Escande et al.[12] developed a strictly convex representation of the robot's bodies. The strict convexity ensures a continuity of the gradient for the distance between two such bodies. The strict convexity is obtained by constructing a boundary volume made of sphere and torus patches. Such construction for the table considered in this application is depicted in figure 3.

2) Feet positions: Instead of taking feet position as inputs of the problem as in [12], the relationship between the feet is considered instead through the matrix $\mathbf{f}$. Moreover instead of inserting the polytope related to the support polygon as a set

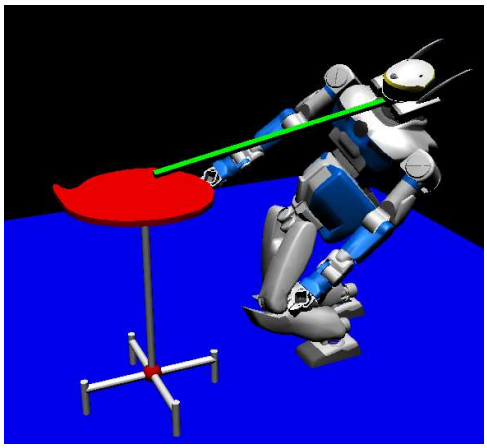

Fig. 2. Pose obtain when the CoM is introduced as a linear constraint and reaches the limit of the support polygon, and by using a minimization function which favor angular position in the middle of their limits.

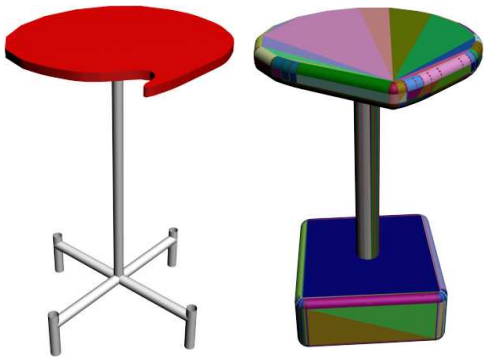

Fig. 3. The table around which the robot is turning, and its STP-BV representation.

of linear inequalities, it is introduced inside the minimization function. In order to have horizontal feet, their roll and pitch are constrained to be zero, through linear equalities. Implicitly, the support polygon can be freely translated and rotated around the yaw axis during the optimization process. This explains why some of the final postures have the feet not oriented towards the object.

3) Gaze direction: The gaze direction is set by linear equalities which force the vision system frame to be aligned with the vector $\mathbf{v}$. This let the head to rotate around this axis as it can be seen in figure 2. Finally we introduced a non linear constraint which is the distance between the point $\mathbf{x}$, where the object is supposed to be centered, and the vision system origin.

C. Experiments and discussion

Using the posture generator presented above we generated 8 postures with different elevation angles, assuming that the object would be on the red table, figure 3.

1) Kinematic limitations: The input parameters are depicted in table I. The most demanding postures are the ones when $\mathbf{v}$ is at the vertical direction. Indeed the cameras are rigidly mounted to the robot's head and look down with an

$v_\theta(deg)$	$\mathbf{f}(m, m, deg)$	$\mathbf{x_z}(m)$	$[d_{\min}, d_{\max}]$	Init. Pos.
0	(0.0, 0.2,30.0)	0.95	[0.5,2.0]	Squat
10	(0.0, 0.2,30.0)	0.95	[0.5,2.0]	Squat
20	(0.0, 0.2, 0.0)	0.95	[0.5,2.0]	Half-Sitting
30	(0.0, 0.2, 0.0)	0.95	[0.5,2.0]	Half-Sitting
40	(0.0, 0.2, 0.0)	0.85	[0.5,2.0]	Half-Sitting
50	(0.0, 0.2, 0.0)	0.75	[0.2,2.0]	Half-Sitting
60	(0.0, 0.2, 0.0)	0.75	[0.2,2.0]	Half-Sitting
70	(0.0, 0.2, 0.0)	0.75	[0.2,2.0]	Half-Sitting

TABLE I
INPUT PARAMETERS FOR EACH OF THE EIGHT POSE GENERATED.

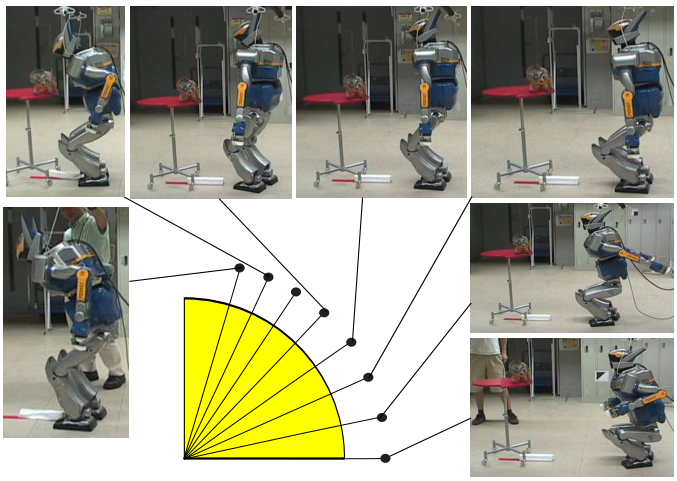

Fig. 4. The poses realized by HRP-2.

angle of 10 degrees. So when asking for 90 degrees, the robot head's has to perform 100 degrees, while avoiding the table. For this reason, the algorithm was not used to find solution over 70 degrees. Between 40 and 70 degrees the robot is asked to look at a point lower than the surface of the table. For 40 and 50 degrees the solution found where feasible and sufficiently far away from the object to be interesting. However considering 60 and 70 degrees, the position were so close to the object that it was practically useless. It shows nonetheless that for a different object, and a different table, the robot can reach such postures. Finally, the posture generated at 50 degrees makes the legs close to a singular position, and therefore is avoided practically. Most of those limitations are mainly due to the robot's kinematic limits and are quite natural. An extension of this work will use the information provided by FSQP on such posture to generate a new Next Best View.

2) Algorithmic limitations: The other demanding postures, but still feasible, are the ones where the vision system should face the object. The difficulty inherent to the optimization scheme to find a solution is the starting condition. In order to find a solution, we had to guide the optimization scheme by providing a squat position. Whereas for the other postures, we simply provided the neutral position of HRP-2's pattern generator. It might be possible to go under 0 degree if other part of the robot are in contact with the ground.

3) Experiments: The computed poses were tested on the humanoid HRP-2 and are depicted in figure 4. For the 8 positions of the robot provided by the posture generator, the appropriate step sequences were computed manually. The robot did perform the transition from the last position, provided by the pattern generator, to the pose by using inverse kinematics. There were no mechanism reinforcing the stability during this transition, but we plan to use the CoM's Jacobian to insure this part [13].

III. SIFT-BASED OBJECT RECONSTRUCTION

A. Object representation

Assuming we have a collection of images taken by the robot, this section focuses on the way the robot "learns" objects for close recognition and pose estimation.

Using its pattern generator, the humanoid can walk, squat, and bend over the object to sample many different views of it. A key step in the learning process is therefore the determination of the Rigid Motion between different views (figure 5). This enables the robot to merge the information it gathers into a single spatially consistent representation.

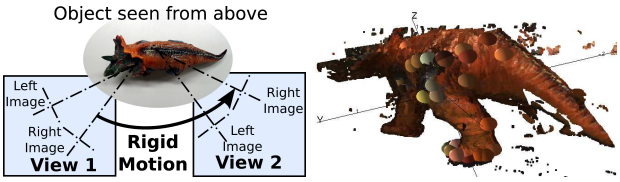

Fig. 5. Left : The data collected is two series of pictures taken by the robot's left and right video cameras from several different viewpoints. Right : The object representation obtained through the learning phase using 5 stereo views. Spheres represent the 3D Features which had played a role during rigid motion evaluation between two views.

To build the representation, we combine dense disparity maps [14] with UCLA's implementation [15] of Lowe's SIFT detector and descriptors [16]. The disparity maps (built using the stereoscopic system) provide us with partial surface information of the object, and the 3D position of detected SIFTs. The SIFT-descriptors are used to match points between the left-images of two stereoscopic views.

The SLAM system detailed in [8] could provide us with an estimate of the rigid motion. We therefore solve an optimization problem between two views. Instead of working in the projective geometry space, we use the Euclidean space for minimization: SIFT descriptors and scale combined with depth information provide matches between pairs of 3D points spotted in two stereoscopic views. The rigid motion evaluation is even simplified by estimating the translation using both clouds' Centers of Gravity, leaving only a rotation quaternion to determine. Thus the final function to minimize is:

$$f = \frac{\sum_{i=1}^{N} \parallel R(X_{1i}) - X_{2i} \parallel^2}{N_{Matches}} \qquad (1)$$

with $R(X_{1i}) = qX_{1i}\bar{q}$, X_{1i} being the point in view 1 and X_{2i} its match in view 2. $N_{Matches}$ is the number of pairs of points we work on (typically 30).

B. Filtering matches

Incorrect matches inevitably occur. In order to discard them, two methods are used: the disparity map, and a RANSAC-like method. The first method is quite natural as it consists in casting aside points for which the 3D location is unknown. Depth thresholds are also set during the disparity map computation: this avoids SIFTs from being detected in the background. The second method consists in checking the value of function f after the minimization process (equation 1). If it exceeds a given threshold, it means at least one match is wrong. Since correct matches all "agree" on the same rigid motion, the distance $\parallel R(X_{1i}) - X_{2i} \parallel^2$ is greater for the false ones. Using this fact, the worst matching feature is discarded, and minimization is performed once again. The process is repeated until f falls below the threshold.

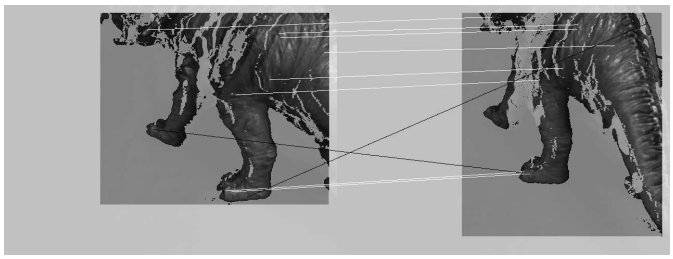

Fig. 6. Matches which are not consistent with the others can be spotted and cast aside.

As shown in figure 6, this is an efficient way of discarding inaccurate matches. Partial reconstruction results are given in figure 5.

C. Object recognition

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, using the same method than for pairs of views during the learning phase. Figures 7 and 8 illustrate those results. 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. Figures 9 and 10 show the last successful Object-Scene matches of both experiments, and tables in figure 11 show the associated results.

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.

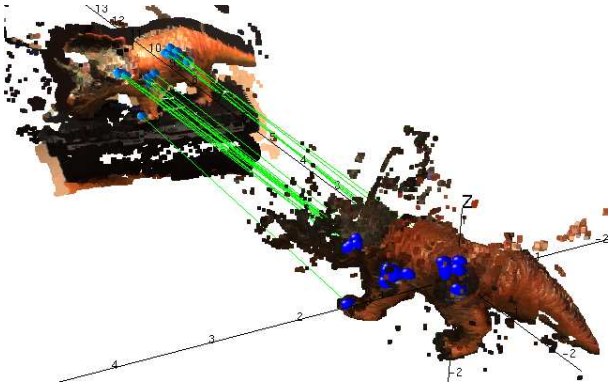

Fig. 7. A screenshot of the model successfully detected in the range map of the scenery. It contains over 6000 3D SIFT features, but only the best matches are represented.

Fig. 8. Left : The left eye's image of figure 7's scenery. Right : The pose of the object is successfully determined using Rigid Motion minimization.

IV. Seeing far away: a generative model based approach

As shown in the last section, the sift-based reconstruction method fails at detecting objects far away. The method presented in this section 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 [17] 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.

A. Visual Features

Images are represented by a set of n overlapping patches and a gradient map (see figure 12).

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 [16] 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 [18]. 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.

Fig. 9. Pose estimation on a uniform background is performed successfully at 1,7m with no occlusion.

Fig. 10. Pose estimation under severe clutter is performed successfully at 1,6m with no occlusion.

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.

B. 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.

Distance	Min Score	Nt	Nrm	Nf	Outcome
0.50 m	1.369E-003	30	20	651	OK
0.60 m	1.772E-003	30	17	654	OK
0.70 m	1.992E-003	30	19	705	OK
0.80 m	1.470E-003	30	19	618	OK
0.90 m	1.583E-003	30	21	571	OK
1.0 m	1.479E-002	30	18	470	OK
1.2 m	3.300E-003	30	19	377	OK
1.4 m	5.268E-003	23	14	273	OK
1.5 m	5.794E-003	29	20	246	OK
1.7 m	1.972E-002	13	5	203	OK
1.9 m	3.953E-002	10	9	175	TRANS
2.1 m	4.790E-002	5	4	145	TRANS
2.3 m	-	4	-	130	TRANS
2.5 m	-	3	-	115	TRANS
2.8 m	1.266E-001	4	4	103	TRANS
3.0 m	-	2	-	84	TRANS
3.5 m	-	2	-	65	TRANS

Distance	Min Score	Nt	Nrm	Nf	Outcome
1.0 m	9.588E-004	30	30	663	OK
1.2 m	2.721E-003	26	14	629	OK
1.4 m	2.225E-002	16	10	621	TRANS
1.6 m	2.669E-002	15	8	992	TRANS
1.8 m	-	30	0	1057	NOK
2.0 m	3.250E-002	9	5	1118	TRANS
2.2 m	-	3	-	1085	TRANS
2.5 m	-	5	-	995	NOK
2.7 m	-	3	-	1013	NOK
3.0 m	-	4	-	1215	NOK

Nf : Number of 3D Features in the scene
Nt : Initial number of matches
Nrm : Number of matches used to find the RM
Min Score : Final minimization score
OK : Pose successfully computed
TRANS : Only the translation was correct
NOK : Neither pose nor location was found

Fig. 11. Left: Recognition results on black background. Right: results in cluttered environment.

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:

$$\begin{aligned} p(\mathcal{P}|\Theta_k) &= p(w^{sift}, w^{color}, rgb, X, d|\Theta_k) \\ &= p(w^{sift}|\Theta_k)p(w^{color}|\Theta_k) \\ &\quad p(rgb|\Theta_k)p(X|\Theta_k)p(d|\Theta_k) \end{aligned} \quad (2)$$

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 IV-D).

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.

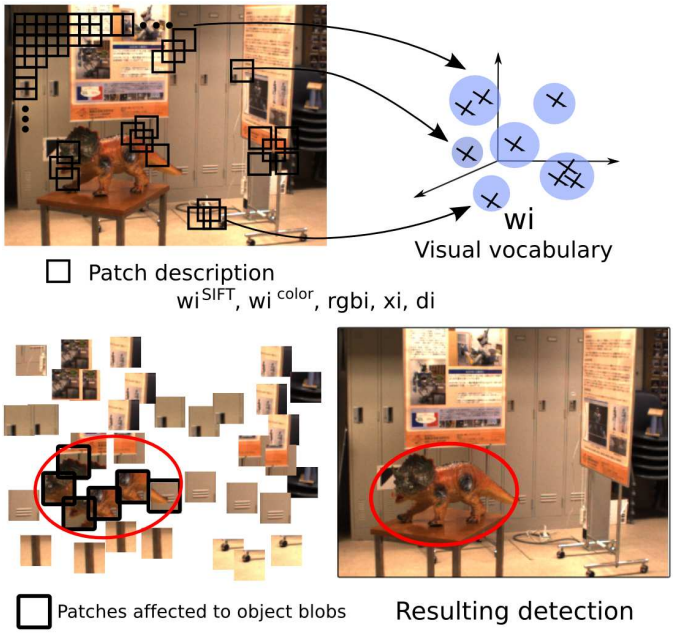

Fig. 12. First row: patches are extracted in a very dense manner. Each patch is associated to the closest visual word for sift and color descriptors, and then represented by the words indexes $(w_i^{SIFT}, w_i^{color})$, a RGB value (rgb_i), a position (x_i) and a depth (d_i) given by the disparity map. Second row: the model computes the best assignment of patches to object blobs or background and estimates to blobs positions.

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 $\mathcal{P}_i$

$$E_i = U_i + \gamma \sum_{j \in \mathcal{N}(i)} V_{i,j} \quad (3)$$

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.

C. 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 [19] (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 [17].

D. 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 (figure 5). The resulting dense disparity map provides local information that we use to create segmentation masks on positive images. This makes 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.

E. Experiments

To evaluate our detection method, we created a test set. It is composed of 118 images taken with the robot cameras. Most of them contains the object in very different conditions. We varied the distance to the object, the view point, the illumination, tried different backgrounds and realized occlusions. The remaining images do not contain the object.

1) Detection evaluation: Our model provides blob shape regions of the image which give approximate position of the object. In order to evaluate the detection performances, these blobs were transformed into bounding boxes describing a rectangular region of the image containing the object. These images were also hand annotated in order to produce a ground truth. A detection is considered as correct if the area of overlap between the predicted bounding box B_p and the ground truth bounding box B_{gt} exceed by a given threshold. We will use 50% by default. This is given by the formula

$$overlap = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} \quad (4)$$

The detection performance is evaluated using precision recall values. If we define N_d the number of detection obtained by our method, N_o the number of objects really present on the set of test images and N_c the number of correct detection, we can define the precision P and the recall R with the following formulas: $P = \frac{N_c}{N_d}$ and $R = \frac{N_c}{N_o}$. Multiple detections of the same object are considered as false detection.

	with depth info		without depth info	
tolerance area	precision	recall	precision	recall
0.2	0.81	0.64	0.65	0.61
0.3	0.73	0.57	0.50	0.48
0.4	0.60	0.48	0.39	0.37
0.5	0.41	0.32	0.26	0.24
0.6	0.15	0.12	0.17	0.16

TABLE II
PRECISION AND RECALL VALUES

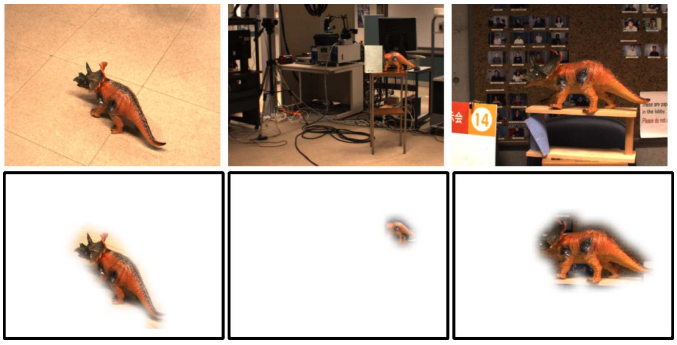

Fig. 14. The model gives a list of patches actually being components of the model. This produces segmentation masks.

In this part we would like to evaluate both the gain of using the depth information, and the overlap between detected bounding boxes and ground truth. Table II shows the precision and recall values for different thresholds. The two columns present precision and recall values for our method (left), and for the original one (right) which is not using any depth information.

First, we can clearly see the improvement obtained using the information given by the disparity map. The depth introduced within the model helps to estimate accurate boundaries of the object, and give good clues on the object expected size.

Looking at the precision values, we can see that on most of the cases (81%) the detected objects have at least a part in common with the original object.

Finally, the recall values indicate that 64% of the object were partially detected.

Figure 13 shows examples of correct detections (overlap greater than 50%) and false detections (overlap lower than 50%).

2) Qualitative segmentation evaluation: 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 14 provides segmentation masks in terms of probability maps of the object location on images where the detection succeeded.

V. CONCLUSION

As a conclusion, this work presents a step towards solving autonomously the object reconstruction for visual search. The 3 components we have described, once fully integrated, will

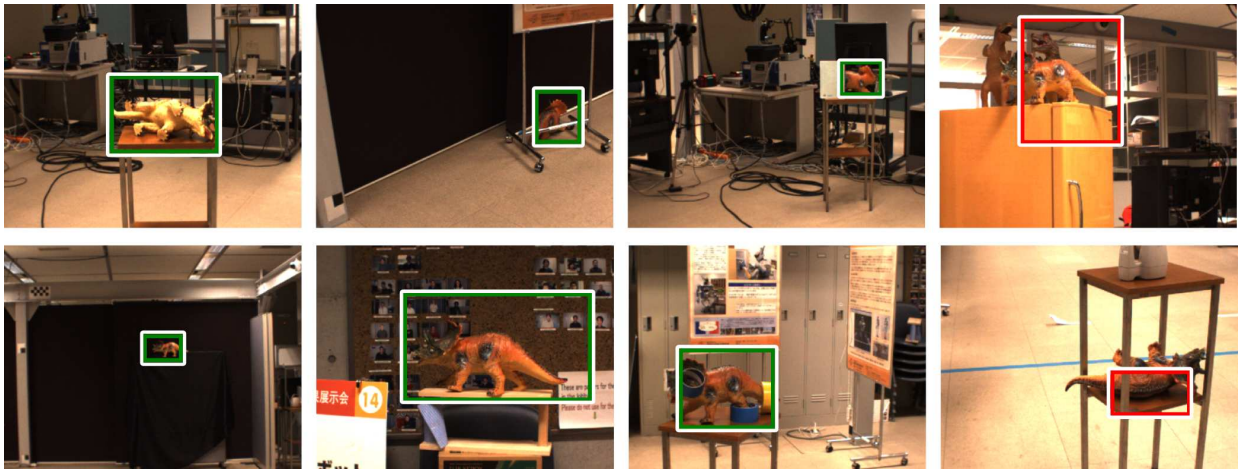

Fig. 13. Two first columns, examples of correct detections. Last column, examples of false detections.

allow a robotic behavior where the robot decides which information to learn and how to acquire it in order to recognize later the object even in challenging conditions.

We presented two different methods of object recognition. The first one relies on a 3D SIFT based representation, and is able to recognize and estimate the pose of a particular object. The second one is a detection method which is designed to provide hypothesis on the object position and scale in challenging conditions for the visual search system in order to reach conditions where the pose can be estimated.

We also presented a method for full body pose generation which is able to acquire some of the challenging images needed to build a robust representation of the object. Due to drift introduced by the rotation of the robot, at the time of this experiment the vision system was not properly centered to take correctly the picture. A correction of the head direction by detecting the location of the table should fixed this problem. Thus in this paper, the pictures used were taken from the robot by setting it manually in the appropriate situations. As we made no assumption on the object itself, the approach is generic and can be applied to any object.

The far away detection model is designed in a way that it can also be used for a multi-object framework, so our method can be easily extended to the recognition of several objects.

The authors would like the JSPS summer school program for partial funding of this work, and the European commission in the context of the ROBOT@CWE project.

REFERENCES

[1] F. Saidi, O. Stasse, and K. Yokoi, "A visual attention framework for search behavior by a humanoid robot," in *IEEE RAS/RSJ Conference on Humanoids Robots*, Genova, Italy, December 4-6 2006, pp. 346–351.

[2] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, and G. Sandini, "Learning about objects through action - initial steps towards artificial cognition," in *IEEE/RAS Int. Conf. on Robotics and Automation*, 2003, pp. 3140–3145.

[3] A. Ude, C. Gaskett, and G. Cheng, "Support vector machine and gabord kernels for object recognition on a humanoid with active foveated vision," in *IEEE/RSJ Int. Conf on Intelligent Robots and Systems*, 2004, pp. 668–673.

[4] K. Welke, E. Oztop, A. Ude, R. Dillmann, and G. Cheng, "Learning feature representations for an object recognition system," in *IEEE/RAS Int. Conf. on Humanoids Robots*, 2006, pp. 290–295.

[5] G. Taylor and L. Kleeman, *Visual Perception and Robotic Manipulation*, B. Siciliano, Ed. Springer, 2006.

[6] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.

[7] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *ECCV International Workshop on Statistical Learning in Computer Vision*, 2004.

[8] O. Stasse, A. Davison, R. Sellaouti, and K. Yokoi, "Real-time 3d slam for humanoid robot considering pattern generator information," in *International Conference on Intelligent Robots and Systems, IROS*, October 2006, pp. 348–355.

[9] 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.

[10] A. Escande, A. Kheddar, and S. Miossec, "Planning support contact-points for humanoid robots and experiments on hrp-2," in *Int. Conf. on Intelligent Robots and Systems, IROS*, 2006, pp. 2974–2979.

[11] C. Lawrence, J. L. Zhou, and A. L. Tits, *Users's Guide for CFSQP Version 2.5: A C Code for Solving (Large Scale) Constrained NonLinear (Minimax) Optimization Problems, Generating Iterates Satisfying All Inequality Constraints*, Electrical Engineering Dept. and Institute for Systems Research, University of Maryland, tR-94-16r1.

[12] A. Escande, S. Miossec, and A. Kheddar, "Strictly convex hulls for computing proximity distances having continuous gradients," *IEEE Transactions on Robotics*, p. (under review), 2007, http://staff.aist.go.jp/adrien.escande.

[13] T. Sugihara and Y. Nakamura., "Whole-body cooperative balancing of humanoid robot using cog jacobian," in *IEEE/RSJ International Conference on Intelligent Robotics and Systems*, 2002, pp. 2575–2580.

[14] D. Scharstein and R. Szeliski, "A taxomomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision, IJCV*, vol. 47, no. 1, pp. 7–42, avril 2002.

[15] A. Vedaldi and S. Soatto, "Local features, all grown up," *cvpr*, vol. 02, pp. 1753–1760, 2006.

[16] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[17] D. Larlus and F. Jurie, "Unifying appearance models and markov random fields for category level object segmentation," submitted to ICCV.

[18] J. van de Weijer and C. Cordelia Schmid, "Coloring local feature extraction," in *ECCV'06*, 2006, pp. 334–348.

[19] R. M. Neal, "Markov chain sampling methods for dirichlet process mixture models," University of Toronto, Tech. Rep. 9815, sep 1998.