3D object recognition using spin-images for a humanoid stereoscopic vision system.

Olivier Stasse, Sylvain Dupitier and Kazuhito Yokoi
AIST/IS-CNRS/STIC Joint Japanese-French Robotics Laboratory (JRL)
Intelligent Systems Research Institute (IS),
National Institute of Advanced Industrial Science and Technology (AIST)
AIST Central 2, Umezono 1-1-1, Tsukuba, Ibaraki, 305-8568 Japan
{olivier.stasse,kazuhito.yokoi}@aist.go.jp

***Abstract*— This paper presents a 3D object recognition method based on spin-images for a humanoid robot having a stereoscopic vision system. Spin-images have been proposed to search CAD models database, and use 3D range informations. In this context, the use of a vision system is taken into account through a multi-resolution approach. A method for quickly computing multi-resolution and interpolating spin-images is proposed. The results on simulation and on real data are given, and show the effectiveness of this method.**

***Index Terms*— Spin-images, multi-resolution, 3D recognition, humanoid robot.**

I. Introduction

Efficient real-time tracking exists for collections of 2D views [1] [2]. However in a humanoid context, 3D geometrical information is important because the high redundancy of such robot allows several kinds of 3D postures. Moreover if the information is precise enough, it can also be used for grasping behaviour. Recent works on 3D object model building make possible a description based on geometrical features. Towards the design of a search engine for databases of CAD models, several 3D descriptors have been proposed to build signatures of 3D objects [3], [4], [5]. The recognition process proposed here is based on spin-images proposed initially by [3]. The main difference in the conventional work and this one lies on the targeted application and a search scheme based on multi-resolution spin images. Moreover the computation of the multi-resolution scheme is refined and allows a fast implementation.

The targeted application us a “Treasure hunting” behaviour on a HRP-2 humanoid robot [6]. This behaviour consists in two majors steps: first building an internal representation of an object unknown to the robot, second finding this object in an unknown environment. This behaviour is useful for a robot used in an industrial environment, or as an aid for elderly person. It may incrementally build its knowledge of its surrounding environment and the object it has to manipulate without any a-priori models. The time constraint is crucial, as a reasonable limit has to be set on the time an end user can wait the robot to achieve its mission. Finally the method to cope with the widest set of objects should rely on a limited set of assumptions.

The reminder of this paper is as follow: in section II the computation of spin images are introduced, section III details how the multi-resolution signature of objects is computed, section IV details the search process, finally section V presents the simulation and the experiments realized with the presented algorithm.

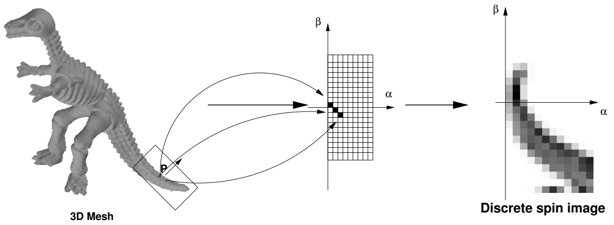

Fig. 1. Example of spin image computation.

II. Spin Images

A. Description

A spin-image can be seen as an image representing the distribution of the object’s density view from a particular point [3]. More precisely, it is assume that all the 3D data are given as a mesh $Mesh = V,E$ where V are the vertices and E the edges. Let’s consider a vertex $P \in V$. The spin image axis are the normal to the point P, and a perpendicular vector to this normal. The former one is called β, and the latter one α. The support region of a spin-image is a cylinder centred on P, and aligned around its normal. From this, each point of the model is assigned to a ring with a *height* along β, and a *radius* along α. An example of spin-images for a dinosaur model is given in Fig. 1.

They are two parameters of importance while using the spin-images: the size of the rings $(\delta\alpha, \delta\beta)$, and the boundaries of the spin-image $(\alpha_{max}, \beta_{max})$. The size of the rings is similar to a resolution parameter. The limitation $(\alpha_{max}, \beta_{max})$ allows to impose constraints between the points chosen for computing the spin-image P and other points of the model P'. This is particularly meaningful to take into account occlusion problem. In our implementation, two points should have less than 90 degrees between their normals. A greater value would implies that P' is occluded by some other points while P is facing the camera.

B. Normal computation

When computing spin-images, the normal computation should be as less sensitive as possible to noise. This is specially important for vision based informations where the noise might be significant. Following the tests done in [7], 8 methods have been tested: gravity center of the polynoms formed by neighbours of each point; inertia matrix; normal average of each face; normal average of faces formed by neighbour points only; normal average weighted by angle; normal average weighted by sine and edge length reciprocal; normal average weighted by areas of adjacent triangles; normal average weighted by edge length reciprocals; normal average weighted by square root of edge length reciprocals. Using the Stanford Bunny model, and adding a Gaussian noise of 20 percent from the average adjacent edge, the most stable method found was the gravity center of the polynoms formed by the neighbours of each point.

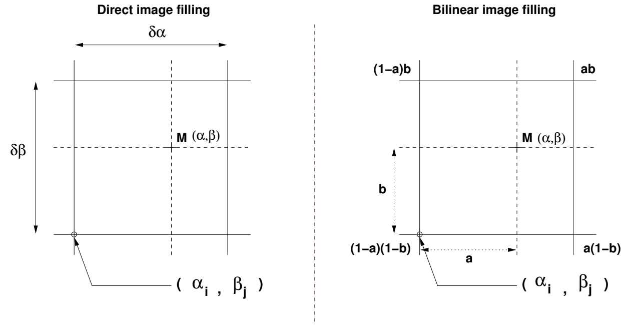

Fig. 2. Two ways to fill a spin-image: (a) direct way (b) bilinear interpolation.

C. Spin-image filling

Regarding the spin-image filling, Johnson propose two ways: either using a direct accumulation, or a bilinear interpolation. Those two methods are depicted in Fig. 2. M is the projection of a point $P' \in V$. The first solution relates $M = (\alpha, \beta)$ in surface (α_i, β_j)-(α_{i+1}, β_j)-$(\alpha_{i+1}, \beta_{j+1})$-$(\alpha_i, \beta_{j+1})$ to the point (α_i, β_j) regardless its position in the surface. This makes the spin-image sensitive to noise. Indeed if M is close to a boundary, it will involves important discrete modification. To solve this problem, a bilinear interpolation allows to smooth the effect of noise by sharing the density information among the 4 points connected to the surface. This is achieved by computing the distance of M to those 4 points, using two parameters (a, b) as depicted in Fig. 2. If the points are processed iteratively in the following $\{0, 1, ..., k, k+1, ...|V|-1\}$, then densities are updated as follows:

$$\begin{aligned} W_{i,j}(k+1) &= W_{i,j}(k) + (1-a)(1-b) \\ W_{i+1,j}(k+1) &= W_{i,j}(k) + a(1-b) \\ W_{i,j+1}(k+1) &= W_{i,j}(k) + (1-a)b \\ W_{i+1,j+1}(k+1) &= W_{i,j}(k) + ab \end{aligned}$$

where $a = (\alpha - \alpha_i)/\delta\alpha$ and $b = (\beta - \beta_j)/\delta\beta$. It is straightforward to check that for a point M the sum of each contribution is one. In the remainder of this paper, for sake of clarity the iteration number is implicit.

III. Multi-resolution

One of the most important feature needed in our case, is the possibility to perceive the object at different distances, and thus at different resolutions. This implies to build a multi-resolution signature of the object, and to be able to compute the resolution at which the object has been perceived. In the following, the finest spin-image $SI_{r_{max}}$ has the highest resolution which correspond to $(\frac{\delta\alpha}{2^{r_{max}}}, \frac{\delta\beta}{2^{r_{max}}})$, while the spin-image SI_k has a resolution $(\frac{\delta\alpha}{2^k}, \frac{\delta\beta}{2^k}) = (\delta\alpha_k, \delta\beta_k)$.

A. Computing resolution of an object

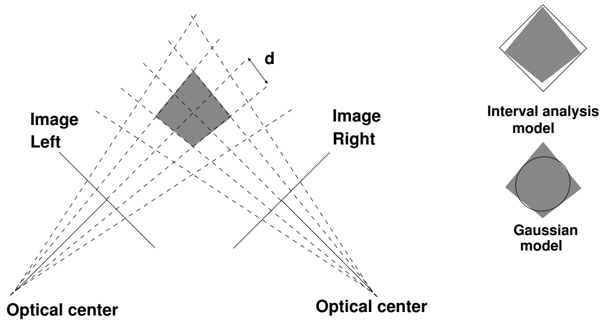

Fig. 3. Model induces by the surface nature of the pixels.

The resolution of the perceived object depends upon the stereoscopic system capabilities, the distance between the robot and the object, and the possible sub-sampling scheme during image processing. This error may also be induced by the segmentation used to match two points in the right and the left images, in our case a correlation. If the pixel is considered as a surface on the image plane, the stereoscopic vision system may be seen as a sensor which perceive 3D volumes. Those volumes are the intersection of the cones representing the surfaces on the image planes. A 2D representation is given in Fig. 3. They can be interpreted also as the location error of a 3D point. [8] and [9] proposed an ellipsoid based approximation of this volume, while [10] proposed a warranted bounding box using interval analysis. Both technics show the non-linearity of the uncertainty related to the reconstruction of a 3D point. However from those previous work, it is clear that the error estimation, and here the resolution, may be different for different parts of the object. While computing the signature, the resolution of the model is given by the average edge's length $L_{model} = \frac{1}{|E|}\sum_{e \in E} ||E||$ of its corresponding data. The number of multiple resolution m pictures can be deduced from the following relationship: $B_{model} = \frac{L_{model}}{2^m}$ where $B_{model} = min\{X_{max}, Y_{max}, Z_{max}\}$ and $\{X_{max}, Y_{max}, Z_{max}\}$ is the bounding box englobing the model. Thus in order to extract a global resolution from the scene, the average edge's length L_{scene} is also used. The resolution r is chosen in the signature such as:

$$min\{r \in \mathbb{N} | L_{scene} < 2^r L_{model}|\} \qquad (1)$$

B. Multi-resolution signature

The dyadic scheme consists in dividing by 2 each dimension of the spin image between two resolutions. Using the direct filling way, it is possible to compute, from the resolution r to $r+1$, the density of a point $M=(i,j)$ in SI_r by:

$$W^r_{(i,j)} = W^{r+1}_{(2j,2j)} + W^{r+1}_{(2i+1,2j)} + W^{r+1}_{(2i,2j+1)} + W^{r+1}_{(2i+1,2j+1)}$$

Using the bilinear interpolated image, the relationship between W^r and W^{r+1} is not so obvious. In Fig. 4, the points from resolution r and $r+1$ are depicted. Our goal is to find a relationship between the density $W^r_{(i,j)}$ and the densities $W^{r+1}_{(2i+k,2j+l)}$ for $k \in \{-2,-1,0,1,2\}$ and $l \in \{-2,-1,0,1,2\}$. The main question is how to share the information carried by the points which will disappear. In Fig. 4 let's consider N_4. As this point is not present in resolution $r+1$, its contribution has to be redistributed to the four adjacent points remaining at resolution r. However as the density of a point M depends upon its distance, if M was in $Q_{0,2}^{r\,(i,j)} = Q_2^{r+1\,(2i-1,2j-1)}$, then its contribution has already been partially taken into account by $N^{r+1}_{(2i,2j)}$, but not by $N^{r+1}_{(2i,2j-2)}$, $N^{r+1}_{(2i-2,2j-2)}$, and $N^{r+1}_{(2i-2,2j)}$. For this three points, an offset of $(\frac{\delta\alpha}{2^r}, \frac{\delta\beta}{2^r})$ has to be introduce while processing $N^r_{(i,j)}$.

We note $Q^{r\,(i,j)}$ the surface described by the points $N^r_{(i-1,j-1)}, N^r_{(i+1,j-1)}, N^r_{(i+1,j+1)}, N^r_{(i-1,j+1)}$. This surface can be cut in four quadrants $Q_l^{r\,(i,j)}$ $l \in \{0,1,2,3\}$ as depicted in Fig. 4. For convenience, and following those notations, those quadrants may also be divided by four and will be noted $Q_{l,k}^{r\,(i,j)}$ $k \in \{0,1,2,3\}$. One can notice that the same quadrant may have several notations depending of the reference point used. For instance $Q_2^{r\,(i,j)} = Q_0^{r\,(i+1,j+1)}$, or $Q_{0,2}^{r\,(i,j)} = Q_2^{r+1\,(2i-1,2j-1)}$.

The notation used for the variables (a,b) is now extended as they change according to the resolution. $a(M, N^r_{(i,j)})$ is the distance along α from $N^r_{(i,j)}$ to M. $b(M, N^r_{(i,j)})$ is the same along β. The relationship between those variables from one resolution to the next one is summarised in Tab. I.

TABLE I

COEFFICIENTS FOR COMPUTING THE MULTI-RESOLUTION BILINEAR INTERPOLATION

Areas	Distances
$Q_0^{r\,(i,j)}$	$a(M,N^r_{(i,j)}) = a(M,N^{r+1}_{(2i,2j)})$ $b(M,N^r_{(i,j)}) = b(M,N^{r+1}_{(2i,2j)})$
$Q_1^{r\,(i,j)}$	$a(M,N^r_{(i,j)}) = a(M,N^{r+1}_{(2i+1,2j)}) + \frac{\delta\alpha}{2^{r+1}}$ $b(M,N^r_{(i,j)}) = b(M,N^{r+1}_{(2i+1,2j)})$
$Q_2^{r\,(i,j)}$	$a(M,N^r_{(i,j)}) = a(M,N^{r+1}_{(2i+1,2j+1)}) + \frac{\delta\alpha}{2^{r+1}}$ $b(M,N^r_{(i,j)}) = b(M,N^{r+1}_{(2i+1,2j+1)}) + \frac{\delta\beta}{2^{r+1}}$
$Q_3^{r\,(i,j)}$	$a(M,N^r_{(i,j)}) = a(M,N^{r+1}_{(2i,2j+1)})$ $b(M,N^r_{(i,j)}) = b(M,N^{r+1}_{(2i,2j+1)}) + \frac{\delta\beta}{2^{r+1}}$

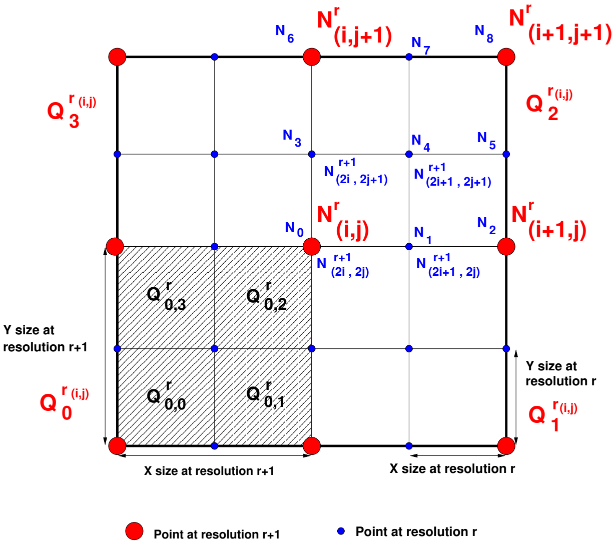

Fig. 4. Computing bilinear interpolated spin-images from one resolution to the other.

Lemma: Let's note $W^r_{(i,j)}(Q)$ the contribution of the quadrant Q for the density at point (i,j) of a spin image having a resolution r filled by bilinear interpolation. If $N_m \in \{N^{r+1}_{(2i+k,2j+l)}\}$ for $k \in \{0,1,2\}$ and $l \in \{0,1,2\}$, and $m = 3k+l$, then we have:

$$\begin{aligned}
W^r_{(i,j)}(Q_2^{r\,(i,j)}) &= \sum_{n=0}^{3}\sum_{m=0}^{8}(1-\frac{a_{N_m}}{\delta\alpha_r})(1-\frac{b_{N_m}}{\delta\beta_r})W^{r+1}_{N_m}(Q_{2,n}^{r\,(i,j)})\\
W^r_{(i+1,j)}(Q_3^{r\,(i+1,j)}) &= \sum_{n=0}^{3}\sum_{m=0}^{8}\frac{a_{N_m}}{\delta\alpha_r}(1-\frac{b_{N_m}}{\delta\beta_r})W^{r+1}_{N_m}(Q_{3,n}^{r\,(i,j)})\\
W^r_{(i,j+1)}(Q_1^{r\,(i,j+1)}) &= \sum_{n=0}^{3}\sum_{m=0}^{8}(1-\frac{a_{N_m}}{\delta\alpha_r})\frac{b_{N_m}}{\delta\beta_r}W^{r+1}_{N_m}(Q_{1,n}^{r\,(i,j)})\\
W^r_{(i+1,j+1)}(Q_0^{r\,(i,j)}) &= \sum_{n=0}^{3}\sum_{m=0}^{8}\frac{a_{N_m}}{\delta\alpha_r}\frac{b_{N_m}}{\delta\beta_r}W^{r+1}_{N_m}(Q_{0,n}^{r\,(i,j)})
\end{aligned} \tag{2}$$

with $a_{N_m} = a(M_m, N^r_{(i,j)})$, $b_{N_m} = b(N_m, N^r_{(i,j)})$, and $W^{r+1}_{N_m} = W^{r+1}_{N^{r+1}_{(2i+k,2j+l)}}$. Finally

$$W^r_{(i,j)} = \sum_{n=0}^{3} W^r_{(i,j)}(Q_n^{r\,(i,j)}) \tag{3}$$

Proof: We give here a partial proof to illustrate the general concept. Lets consider the point $M \in Q_{2,2}^{r\,(i,j)} = Q_2^{r+1\,(2i+1,2j+1)} = Q_2^{r+1\,N_4}$ at resolution $r+1$. The points N_4, N_5 and N_7 of the spin images mesh are considered. The contribution provided by M to each of those points is computed as follows:

$$\begin{aligned}
W^{r+1}_{N_4}(Q_2^{r+1\,N_4}) &= \sum_{M\in Q_2^{r+1\,N_4}} \frac{a(M,N_4)}{\delta\alpha_{r+1}}(1-\frac{b(M,N_4)}{\delta\beta_{r+1}})\\
W^{r+1}_{N_5}(Q_2^{r+1\,N_4}) &= \sum_{M\in Q_2^{r+1\,N_4}} (1-\frac{a(M,N_4)}{\delta\alpha_{r+1}})\frac{b(M,N_4)}{\delta\beta_{r+1}}\\
W^{r+1}_{N_7}(Q_2^{r+1\,N_4}) &= \sum_{M\in Q_2^{r+1\,N_4}} (1-\frac{a(M,N_4)}{\delta\alpha_{r+1}})(1-\frac{b(M,N_4)}{\delta\beta_{r+1}})\\
W^{r+1}_{N_8}(Q_2^{r+1\,N_4}) &= \sum_{M\in Q_2^{r+1\,N_4}} \frac{a(M,N_4)}{\delta\alpha_{r+1}}\frac{b(M,N_4)}{\delta\beta_{r+1}}
\end{aligned}$$

Now the same point $M \in Q_2^{r+1_{N_4}}$ at resolution r can be computed through bilinear interpolation filling. This may be written for $N^r_{(i,j)}$:

$$W^r_{(i,j)}(Q_2^{r+1_{N_4}}) = \sum_{M \in Q_2^{r+1_{N_4}}} (1 - \frac{a(M, N^r_{(i,j)})}{\delta\alpha_r}) \\ (1 - \frac{b(M, N^r_{(i,j)})}{\delta\beta_r}) \tag{4}$$

From Tab. I, and having $2\delta\alpha_{r+1} = \delta\alpha_r$ Eq. 4 can be rewritten:

$$\begin{aligned} W^r_{(i,j)}(Q_2^{r+1_{N_4}}) &= W^r_{(i,j)}(Q_{2,2}^{r(i,j)}) = \\ &= \sum_{M \in Q_2^{r+1_{N_4}}} (1 - \frac{a(M, N_4) + \delta\alpha_{r+1}}{2\delta\alpha_{r+1}}) \\ &\quad (1 - \frac{b(M, N_4) + \delta\alpha_{r+1}}{2\delta\beta_{r+1}}) \\ &= \sum_{M \in Q_2^{r+1_{N_4}}} \frac{1}{2}(1 - \frac{a(M, N_4)}{\delta\alpha_{r+1}}) \\ &\quad \frac{1}{2}(1 - \frac{b(M, N_4)}{\delta\beta_{r+1}}) = \frac{1}{4} W^{r+1}_{N_4}(Q_2^{r+1}) \end{aligned} \tag{5}$$

Using the same arguments, we can find:

$$\begin{aligned} W^r_{(i,j)}(Q_{2,0}^{r(i,j)}) &= W^{r+1}_{N_0}(Q_{2,0}^{r(i,j)}) + \frac{1}{2} W^{r+1}_{N_1}(Q_{2,0}^{r(i,j)}) \\ &\quad + \frac{1}{2} W^{r+1}_{N_3}(Q_{2,0}^{r(i,j)}) + \frac{1}{4} W^{r+1}_{N_4}(Q_{2,0}^{r(i,j)}) \\ W^r_{(i,j)}(Q_{2,1}^{r(i,j)}) &= \frac{1}{2} W^{r+1}_{N_1}(Q_{2,1}^{r(i,j)}) + \frac{1}{4} W^{r+1}_{N_4}(Q_{2,1}^{r(i,j)}) \\ W^r_{(i,j)}(Q_{2,3}^{r(i,j)}) &= \frac{1}{2} W^{r+1}_{N_3}(Q_{2,3}^{r(i,j)}) + \frac{1}{4} W^{r+1}_{N_4}(Q_{2,3}^{r(i,j)}) \end{aligned} \tag{6}$$

Thus

$$\begin{aligned} W^r_{(i,j)}(Q_2^{r(i,j)}) &= \sum_{n=0}^{3} W^r_{(i,j)}(Q_{2,n}^{r(i,j)}) \\ &= W^{r+1}_{N_0}(Q_{2,0}^{r(i,j)}) + \frac{1}{2} W^{r+1}_{N_1}(Q_{2,0}^{r(i,j)}) \\ &+ \frac{1}{2} W^{r+1}_{N_3}(Q_{2,0}^{r(i,j)}) + \frac{1}{4} W^{r+1}_{N_4}(Q_{2,0}^{r(i,j)}) \\ &+ \frac{1}{2} W^{r+1}_{N_1}(Q_{2,1}^{r(i,j)}) + \frac{1}{4} W^{r+1}_{N_4}(Q_{2,1}^{r(i,j)}) \\ &+ \frac{1}{2} W^{r+1}_{N_3}(Q_{2,3}^{r(i,j)}) + \frac{1}{4} W^{r+1}_{N_4}(Q_{2,3}^{r(i,j)}) \\ &= \sum_{n=0}^{3} \sum_{m=0}^{8} (1 - \frac{a_{N_m}}{\delta\alpha_r})(1 - \frac{b_{N_m}}{\delta\beta_r}) W^{r+1}_{N_m}(Q_{2,n}^{r(i,j)}) \end{aligned} \tag{7}$$

The same arguments holds for the other points and proof the lemma □.

The multi-resolution computation of the spin images is done first by computing the most precise spin-image through examination of every points. For each point of the spin image, four densities corresponding to each quadrant are stored. For lower resolution images, the density is computed using the position of the point regarding the quadrant considered and Eq. 2.

It should be stress here that in our current implementation, only the spin-images are submit to a multi-resolution scheme. In this first step, no sub-sampling of the mesh has been applied. Thus if the size of the spin-images decrease in this process, the number of points does not.

IV. Search process

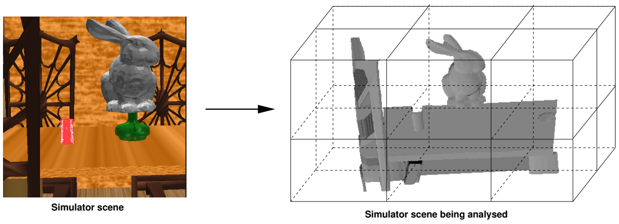

Fig. 5. A 3D mesh extracted from the Stanford Bunny flying in the OpenHRP simulator. The scene is cut according to the bounding box model.

The search process described here is based on a 3D mesh. This can be either a single view of the environment or an incrementally build representation. In our current implementation, it is a single view provided by the stereoscopic system. In the following, it is called the scene.
The scene is divided in sub-blocks. The sub-block size is given by the bounding box of the searched object as depicted in Fig. 5. On each of the sub-block the following algorithm is applied:

1) Select the best resolution according to the average edge-length;
2) Get the main rigid transformation which project the model into the scene;
3) Check if if the model is in the scene using the previously computed rigid-transformation. This provides a main correlation coefficient, and the position plus orientation in the scene of the seen object.

A. Selection of the best resolution

From section III, the object resolution is the average edge's length in the scene. Then the resolution for the model's spin-images is chosen according to Eq. 1. Two spin-images (p,q) with the same resolution are compared using the following correlation function as proposed in [3]:

$$R = \frac{N.\sum_{i=0}^{N} p_i.q_i - \sum_{i=0}^{N} p_i . \sum_{i=0}^{N} q_i}{\sqrt{N.\sum_{i=0}^{N} p_i^2 - (\sum_{i=0}^{N} p_i)^2} . \sqrt{N.\sum_{i=0}^{N} q_i^2 - (\sum_{i=0}^{N} q_i)^2}} \\ R \in [-1\,;\,1] \tag{8}$$

with N the number of non-empty points in spin-image of the scene. This correlation can be proven to be independent to the normalisation of a spin-image. Thus during the multi-resolution phase the spin-images are not normalised.

B. Rigid transformation evaluation

The main rigid transformation is obtained as follows: Some points are randomly selected in the scene. Their corresponding points in the model are searched by comparing

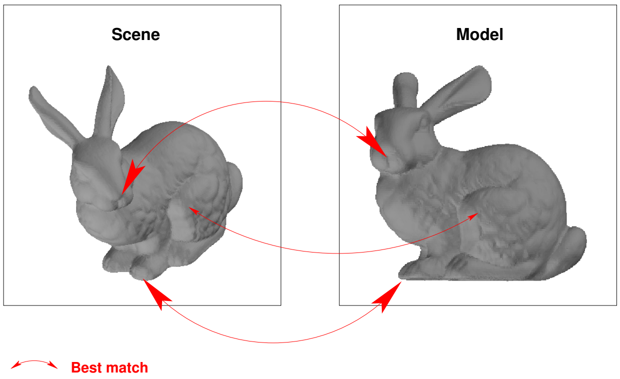

Fig. 6. Matching points.

their spin-image to all the model's spin-images as depicted in Fig. 6.

This gives a list L_C of matching points sorted by their correlation coefficients. To remove false matching, the last 20 % elements of L_C are discarded. From this, a list of rigid transformation L_{RT} is extracted by considering sets of 4 points in the list L_C as depicted in Fig. 7.

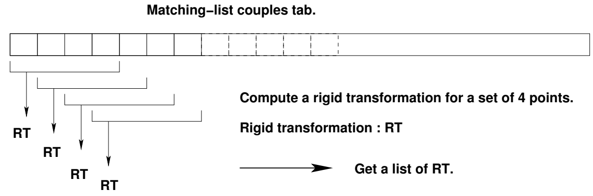
Matching–list couples tab.
Compute a rigid transformation for a set of 4 points.
Rigid transformation : RT
Get a list of RT.
RT
RT
RT
RT

Fig. 7. Sets of 4 points used to compute rigid transformation.

For each rigid transformation $e \in L_{RT}$, a mark is computed by considering all the couples of L_C. If e is the real rigid transformation, then it should project the maximum number of points from the scene to the model.

C. Final correlation coefficient

On order to verify the main rigid transformation, points of the model are chosen randomly and verified against the scene using the proposed main rigid transformation. The main correlation coefficient is the average of the 80 % best correlation coefficients.

This procedure is applied to each of the sub-space.

V. Experiments

A. Simulation

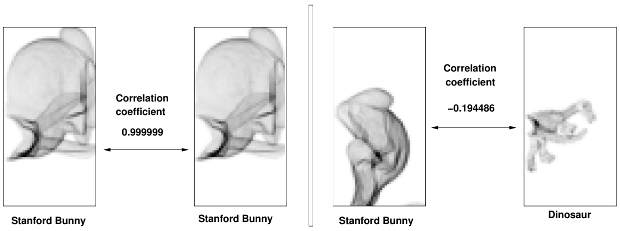

Fig. 8. Spin images comparison examples.

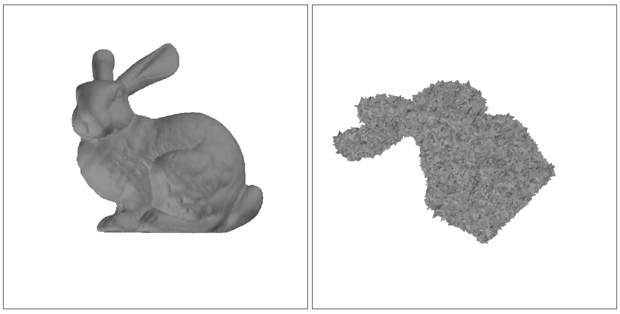

Fig. 9. The Stanford Bunny with a white noise based on the average edge length.

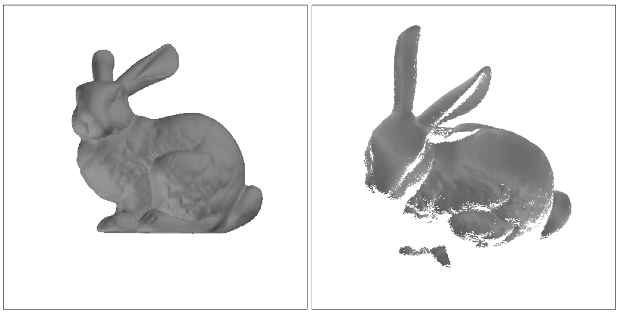

Fig. 10. The Stanford Bunny with self occlusion.

The previously described algorithm was tested on different situations to check its efficiency. First, a Stanford Bunny spin-image was tested against a spin-image of the dinosaur represented in Fig. 1. This intended to evaluate the correlation value against a very different spin-image. The returned correlation was -0.19.

Next, a 20 % white noise has been added to the Stanford Bunny after a rigid transformation including two rotations: 45 degrees around the X axis, and 90 degrees around the Y axis and no translation as depicted in Fig. 9. This noise was taken according to the average length of the connected edge for a point. The returned correlation was 0.91 and the rotation evaluated to 42 degrees around X, 92 degrees around Y and -1 around Z.

The third case intends to simulate a single view of the complete 3D model, and the subsequent self-occlusion as shown in Fig. 10. The associated rigid transformation has no rotation and no translation. The resulting main correlation coefficient was 0.22. From those simulation we can conclude that the search seems to be rotation invariant, robust against noise but is sensitive to occlusion. However the correlation coefficient is still higher when only partial informations are available, than with a complete different object. This provides a good candidate for the next view.

B. OpenHRP[11] simulator

In this context, the HRP-2 humanoid robot is simulated inside a house environment. The goal of this simulation was to try to cope with different objects present in the scene. In order to discard any perturbation from the occlusion, and the multi-resolution, the model used for the search process

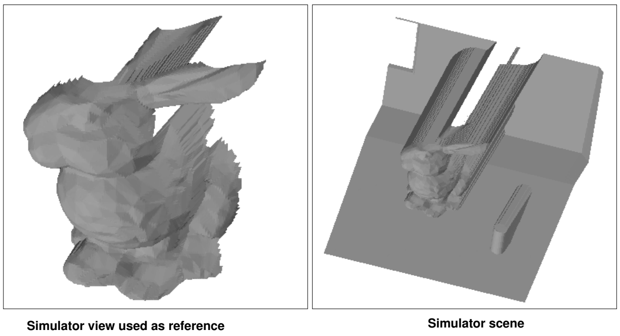

Fig. 11. Simulation using the OpenHRP simulator.

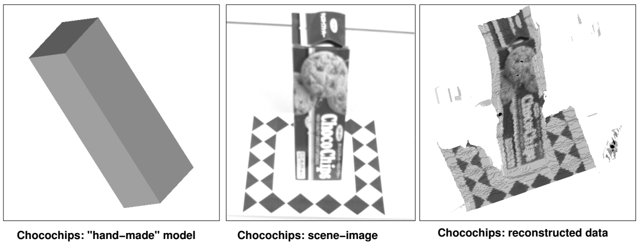

Fig. 12. Experiment on a single view of the HRP2 humanoid robot

was a view of the Stanford Bunny from the OpenHRP simulator.

This model is thereafter search inside a virtual house. The Stanford Bunny is above a table, behind chairs, and several objects are presents in the background, as depicted in Fig. 11 and Fig. 5. Using the previously described scheme, the model is found with a correlation coefficient close to 0.99. In this context, we can conclude that the other objects in the scene does not decrease the efficiency of the search.

C. *Real data*

The HRP-2 humanoid robot is equipped with a trinoptic vision system. In this particular case, only two cameras are used. Using a correlation method to match points between the left image and the right image, clouds of 3D points are computed using epipolar geometry. The implementation is a modified version of the VVV system [12]. The object used for this test is a box of cookies depicted in Fig. 12.(b). Its model, here hand-made, is represented in Fig. 12.(a). The reconstructed mesh is displayed in Fig. 12.(c). The recognition process returned a correlation coefficient equal to 0.234 which is similar to the result obtained through simulation.

D. *Computation time*

To build the Stanford Bunny model, it takes 6 minutes and 24 seconds for 34834 points. The recognition process takes 32 seconds for a scene, using 100 spin images to compute the rigid transformation. The recognition process applied to 8 scenes takes 2 minutes 19 seconds, using 50 spin images for the rigid transformation. Also our multi-resolution approach decreased the initial results obtained with this implementation, it is currently not sufficient for our targeted application. This implementation has not been optimised to take fully advantage of the newest Pentium capabilities. Moreover during the recognition process, as it has already been noted, if the size of the spin-images is decreasing in the multi-resolution scheme, it is not the case of the number of points. It has no impact on the efficiency of the recognition as the correlation is not sensitive to this problem. However, this is clearly time consuming. Two kinds of improvement are possible: using a compression scheme such as the Principal Component Analysis as proposed in [3], or a Wavelet based approach such as WaveMesh [13].

VI. Conclusion

A visual search process based on clouds of 3D points has been presented in this paper. It relies on a multi-resolution signature using spin-images as descriptors. A fast iterative algorithm has been proposed to compute efficiently lower resolution spin-images from the finest one. A first implementation and its applications to simulated and real data have been presented to validate the approach. It shows the process robust against noise, rotation invariant, able to cope with size, and still able to provide information when an occlusion occurs. Our future work is too improve the efficiency of this implementation, and applied it in the context of a “Treasure Hunting” behaviour.

References

[1] F. Jurie and M. Dhome, “Real time tracking of 3d objects : a robust approach,” *Pattern Recognition*, vol. 35, no. 2, pp. 317–328, 2002.

[2] A. Ude and C. Atkeson, “Probabilistic detection and tracking at high frame rates using affine warping,” in *Proceedings of the International Conference on Pattern Recognition, Quebec City, Canada*, august 2002.

[3] A. E. Johnson and M. Hebert, “Using spin images for efficient object recognition in cluttered 3d scenes,” 1999.

[4] M. Kazdhan, T. Funkhouser, and S. Rusinkiewicz, “Rotation invariant spherical harmonic representation of 3d shape descriptors,” 2003.

[5] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik, “Recognizing objects in range data using region and point descriptors,” 2004.

[6] K.Kaneko, F.Kanehiro, S.Kajita, H.Hirukawa, T.Kawasaki, M.Hirata, K.Akachi, and T.Isozumi, “Humanoid robot hrp-2,” in *Proceedings of the 2004 IEEE International Conference on Robotics & Automation*, 2004.

[7] S. Jin, R. R. Lewis, and D. West, “A comparison of algorithms for vertex normal computation,” 2003.

[8] H. Hirschmüller, P. R. Innocent, and J. M. Garibaldi, “Fast, unconstrained camera motion estimation from stereo without tracking and robust statistics,” in *7th International Conference on Control, Automation, Robotics and Vision, Singapore*, December 2002.

[9] N. Molton and M. Brady, “Practical structure and motion from stereo,” *International Journal of Computer Vision*, vol. 39, no. 1, August 2000.

[10] B. Telle, O. Stasse, T. Ueshiba, K. Yokoi, and F. Tomita, “3d boundaries partial representation of objects using interval analysis,” in *International Conference on Intelligent Robotics Systems and Systems (IROS), Sendai, Japan*, 2004.

[11] H. Hirukawa, F. Kanehiro, and S. Kajita, “Openhrp: Open architecture humanoid robotics platform,” in *Proceedings of the International Symposium of Robotics Research*, 2001.

[12] 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, January 2002.

[13] S. Valette and R. Prost, “Wavelet-based progressive compression scheme for triangle meshes: Wavemesh,” *IEEE Transaction on Visualization and Computer Graphics*, vol. 10, Number 2, 2004.