Structured light 3D free form recovering with sub-pixel precision

Fakhr-Eddine Ababsa, David Roussel, Malik Mallem

Structured Light 3D Free Form Recovering With Sub-pixel Precision

Fakhr-eddine Ababsa David Roussel Malik Mallem

Laboratoire Systèmes Complexes. CNRS FRE 2494
40, Rue du Pelvoux,CE1455-Courcouronnes,
91020 Evry Cedex France,
[ababsa|roussel|mallem]@iup.univ-evry.fr

Tel : (+33) 1 69 47 75 04
Fax : (+33) 1 69 47 75 99

Abstract

This paper deals with the 3D Free form object pose recovering problem that is present in several industrial applications, such as online quality control in production as well as image systems for assembly/welding, augmented reality and robotics. The method presented here uses a structured light based vision system to reconstruct several accurate local 3D patches of the objects. A robust sub-pixel method for image features detection has been developed in order to increase 3D reconstruction accuracy. The Curvature method is then used to compute a geometric invariant "footprint" to discriminate reconstructed patches, and so allows to match them with the object's model. Pose recovering is performed by using the prediction-verification hypotheses paradigm. Some experimental results are given to show the efficiency of the proposed solution when we apply it on a complex free form object.

Key words: Free form object, structured light, sub-pixel detection, 3D recovery, Matching, Geometric Invariant, recognition by indexation.

1 Introduction

The problem of 3D free form object pose recovering is an important research topic in computer vision. It consists in determining the position and orientation of the object with respect to a camera or a predefined frame of reference. Many industrial applications are confronted with this problem where it is necessary to know, with high precision, the object pose to successfully perform the required task. Thus, in Augmented Reality applications for instance, the accurate object pose recovering allows to superpose a virtual world on top of the real one correctly, whereas in Medical Imagery it is important for surgical planning.

Several approaches have been developed these last years to solve this problem, they can be subdivided into two categories (Cambpell, & Flynn, 2001):

1. approaches which use intensity imagery: like appearance-based methods (Murase, & Nayar, 1995). An appearance based system for 3D object recovering encodes individual object views as points in one or several multidimensional spaces. The bases for these spaces are obtained from a statistical analysis of the ensemble of training image sets. Recognition of an unknown view is typically performed by projecting that view into the parameter space(s) along the stored basis vectors and finding the nearest projected view of a training image. However, Appearance based approaches have some disadvantages such as the difficulties with segmenting the object from the background and dealing with occlusions.
2. approaches which use range images: 3D object recovering in range images has a longer history and systems developed for this purpose have demonstrated a large variety of methods. Besl (1990) reviewed the difficulties in matching free form objects in range data using only points. Such matching procedures like ICP (Besl, & McKay, 1992) have exponential computational complexity. Therefore, most researchers have developed techniques to reduce the amount of computation required. These methods often group corresponding model and scene point pairs into sets that can be used to determine object pose in the scene. For instance, Chua and Jarvis (1997) have formulated a new representation called “point signature” which encodes the

minimum distances of points on a 3D contour to a reference plane. The 3D contour is constructed by intersecting the surface with a sphere centered on the considered point. This provides a compact way to store information about the local structure of the surface around a point which is pose invariant. Their system shows quick recognition times for multi-object scenes using a data base of fifteen models with high accuracy. Johnson and Hebert (1999) also used point features for object recovering, they used "spin images" which could be considered as 2D histograms of the surface locations around a point. The spin image is generated using the normal to the selected point and rotating a cutting plane around the point using the normal as the rotation axis. As the plane spins around the point, the intersections between the plane and the surface are used to index a 2D histogram. The bins in the histogram represent the amount of surface a particular patch of the cutting plane intersects the object. Spin images are used to establish correspondence between scene and model points for 3D data. The correspondences are then grouped by model and checked for geometric consistency in order to compute a rotation and translation between model and scene points. The approach presented in this paper also uses such a local signature. It considers local patches extracted from the object's surface and determines, on the one hand, the distribution of angles between the surface patch normal and the normal vectors in the local neighborhood, and on the other hand, the surface patch local curvature. Such features are invariant with respect to rotation and translation and also discriminative for free form objects, their combination are used to perform matching between the scanned object and the object's model.

To successfully perform 3D object recovering using range images, 3D accurate measurements systems are needed (Rocchini, Cigoni, Montani, Pingi, & Scopigno, 2001). Many 3D sensing techniques have been developed these last few years. For instance, triangulation sensors are based on points correspondences between two or more points of view. In order to obtain 3D measurements, point correspondences have to be established, allowing 3D shape to be reconstructed. The family of

triangulating sensors can be further subdivided into *active* and *passive* triangulation systems. Active triangulation systems illuminates the scene with a controlled light (such as a laser) rather than relying on natural or uncontrolled lighting. On the other hand, a stereo camera set could be considered as the simplest example of passive triangulation. It uses two or more cameras to estimate stereo disparity between matched points and then recover depth. The difficulty is then to solve the correspondence problem between left and right views by means of image matching process which is generally a slow and complex process. In order to overcome this difficulty and to speed up the reconstruction process, one can use active triangulation systems which project a light pattern onto the object. The light patterns are distorted by the object surface and hence artificial features are then created on this surface (Valkenburg, & McIvor, 1998). These artificial features are observed by at least one camera and then more easily extracted from the image, matched with the undistorted pattern and used to reconstruct the object surface. Some of the most widely used active triangulation techniques are (Hu, & Stockman, 1989): Light dot range finders, light stripe range finders and light grid range finders. The achieved vision system to measure 3D object points consists of a manufactured laser grid projector and a CCD camera, this allows a local and coarse reconstruction of the object's surface without having to sweep across the object.

The remainder of this paper is organized as follows. In section 2, the built active vision system is presented. Thus, projector and camera models as well as their calibration procedure are described. The developed image processing technique to extract image features and the 3D reconstruction method are then presented. Section 3 explains the elaborated 3D object recovering method which is a Model-based method (Pennec, 1998), and so is based on 3D/3D matching between features extracted from the object model in the model database and features extracted from the scanned object. Section 4 shows experimental results and discusses the computational performance of the proposed techniques. Finally the conclusions are presented in Section 5.

2 Vision System

2.1 Vision Workbench and Constraints

Figure (1) shows the developed vision workbench, it is composed of a manufactured grid projector and a CCD camera. A regular grid of five horizontal and vertical lines is projected on the target object, intersect the surface of the objects and traces out a deformed grid. Such an encoded scene is then recorded by the camera. The projected grid is considered as a graph by considering only nodes (lines grid intersection points) and the topological relationship between them. The grid projector is mounted on rotating turret allowing several regions of the scene's objects to be scanned. The distance between the projector and the camera can be changed within the translation interval of 100mm to 500mm. Both projector and camera can translate along d_z direction which allows to translate the vision system closer to the worktable, where the object lies. The system features stepped motors in order to fully control the rotation and the translation of the vision system. Besides, The laser beam and the recovery process requires that scene objects should be considered as rigid, opaque, static and have piecewise smooth surfaces respectively.

2.2 System Calibration

The coordinate systems used in calibration procedure are illustrated in figure (2). The developed calibration method used is based on pin-hole optical models for both camera and projector.

2.2.1 Camera Calibration

Let *P(x,y,z)* be a 3D point in the world coordinate system (see figure 2), the relationship between P and its corresponding image point is expressed as (Faugeras, 1993):

$$\begin{bmatrix} s.u \\ s.v \\ s \end{bmatrix} = M_c \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{1}$$

where s is a scale factor and M_c is (3x4) calibration matrix.

Camera calibration involves estimating elements of the matrix M_c. By measuring enough known 3D points (126 points have been used) and their corresponding image points, M_c can be precisely determined

using a least-square solution to linear equation (1). Practically, the grid projector is used to generate 3D points as follows: first the vision system is placed at a distance d1 from (*xoy*) worktable plane to simulate a virtual vertical plane ($z = d_1$=100mm) in the world coordinate system, then the laser grid is projected in this plane for several rotation angles φ and θ which can be controlled with high precision (0.01 degree) by the rotating turret. Then, nodes coordinates are measured within world coordinate system. φ and θ are chosen in such a way that 3D points are isotropically scattered in the work space. This measuring process is repeated for several distances (d_2=300mm, d_3=600mm) in order to have a non coplanar set of 3D points.

2.2.2 Projector Calibration

As the grid projector can rotate along two axes φ and θ in the rotating turret coordinate system, a calibration method allowing to calibrate the projector for any known values of φ and θ have been developed. The projector is, first, calibrated in its initial position (φ=θ=0), the relationship between 3D grid node and their corresponding image grid point satisfies equation:

$$\begin{bmatrix} t.u_g \\ t.v_g \\ t \end{bmatrix} = M_p \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2}$$

where t is a scale factor, M_p is a (3x4) projector's calibration matrix.

In the same way as for the camera calibration, several 3D points are generated using the same method explained above, then least-square solution is used to linear equation (2) to estimate M_p elements.

When the projector rotates along (O_mX_m) or (O_mY_m), only extrinsic parameters (i.e. rotation and translation between the world coordinate system and a projector coordinate system) change. Thus, M_p can be decomposed in Intrinsic Parameter Matrix I_p(3x4) (i.e. projector internal parameters which doesn't change) and Extrinsic Parameters Matrix A_p(4x4) (Faugeras, 1993) such as:

$$M_p = I_p \times A_p \tag{3}$$

and

$$A_p = \begin{pmatrix} R_{3\times3} & T_{3\times1} \\ 0_{1\times3} & 1 \end{pmatrix} \tag{4}$$

where R_{3x3} and T_{3x1} are the rotation matrix and the translation vector respectively from the world coordinate system to the projector one.

When the turret rotates with angles φ and θ along (O_mX_m) and (O_mY_m) axes respectively, the rotation and translation matrices become:

$$R' = R_\varphi \times R_\theta \times R \tag{5}$$

$$T' = C_{m/w} \times \left(\left[R_\varphi \times R_\theta \right] \times O_{p/m} \right) \tag{6}$$

where $C_{m/w}$ expresses the transformation from the turret coordinate system to the world coordinates.

Equation (6) means that coordinates of the projector center are first computed in the turret coordinate system, then multiplied by $C_{m/w}$ matrix in order to obtain T' in the world coordinate frame.

Thus, the new extrinsic matrix A'_p expressed in world coordinate system can then be rewritten as:

$$A'_p = \begin{pmatrix} R'_{3\times3} & T'_{3\times1} \\ 0_{1\times3} & 1 \end{pmatrix} \tag{7}$$

Finally, the global transformation matrix M'_p of the projector on its current position is given by:

$$M'_p = I_p \times \left(A'_p \right)^{-1} \tag{8}$$

This equation allows to recalibrate the projector in any position without having to perform all the calibration procedure again.

2.3 Projected Image Grid Extraction

In this section, the algorithm to extract the projected image grid with sub-pixel accuracy, as well as the matching procedure between image nodes and grid nodes are described.

The image grid extraction procedure consists in two steps:

- Nodes detection.
- Grid reconstruction.

2.3.1 Nodes detection

Each node of the image grid is, locally, considered as a cross center. Thus, a model based method (Deriche, & Giraudon, 1993) is proposed in order to precisely locate the cross center in a gray level image. The proposed luminance model allows the modeling of any shaped cross (right or oblique cross – see figure 3), and is modeled with combination of gaussians by the following equation:

$$L(x,y)=a+b\cdot\left(e^{-c\cdot(x-d\cdot y-e)^2}+e^{-f\cdot(y-g\cdot x-h)^2}-e^{-c\cdot(x-d\cdot y-h)^2-f\cdot(y-g\cdot x-h)^2}\right) \tag{9}$$

where d, e, g and h determine the location parameters of the two edge lines of the cross primitive, and c, f determine the amplitude parameters.

This model applies to images containing only one cross, so single cross from the original image must first be localized into small “sub-images”.

The accurate node detection includes the following steps :

- Coarse detection of grid node using a skeleton method.
- Extract sub-image around the approximate cross center.
- The parameters of equation (9) are determined using a non linear optimization method in order to fit the luminance surface of the cross model.
- Accurate node position is then determined by the intersection of the estimated cross edge lines.

For each isolated cross sub-image *I(i,j)*, a Levenberg-Marquardt method is used to determine optimal parameters of the model by minimizing the following criterion:

$$c=\sum_{i=1}^{N_l}\sum_{j=1}^{N_c}\left[L(i,j)-I(i,j)\right]^2 \tag{10}$$

The parameters vector is defined as:

$$V=\begin{bmatrix} a & b & c & d & e & f & g & h \end{bmatrix} \tag{11}$$

In this method, vector V must be initialized, but one can notices that some parameters don't change much in the different cross sub-images, so they can be initialized with :

$$a=I_{min} \quad , \quad b=I_{max}-a \quad , \quad c=f=0.2 \tag{12}$$

where I_{min} and I_{max} are the minimal and the maximal gray level respectively of the current sub-image.

The remaining parameters related to edge line direction and thickness are initialized by using intersection points P_1, P_2, P_3 and P_4 (see figure 4) with the four blocks Fr_1, Fr_2, Fr_3 and Fr_4 in the skeletons sub-image,, such as :

$$\begin{array}{ll} e=P_1 & , \quad h=P_4 \\ d=\dfrac{\left\| \vec{P_1P_2} \right\|}{N_b} & , \quad g=\dfrac{\left\| \vec{P_4P_3} \right\|}{N_b} \end{array} \tag{13}$$

where N_b is blocks length (we used N_b=12).

Once the optimal parameters are determined, the coordinates of the cross center in the current sub-image are given by:

$$\begin{bmatrix} u_c \\ v_c \end{bmatrix} = \begin{bmatrix} 1 & -g \\ -d & 1 \end{bmatrix} . \begin{bmatrix} h \\ e \end{bmatrix} \tag{14}$$

Figure (5-a) shows an image of a projected grid on a vertical plane. Figure (5-b) illustrates a cross sub-image isolated by using the first coarse detection, and the lines corresponding to the accurate cross/node detection.

2.3.2 Grid reconstruction

The procedures described above give a list of nodes which must be organized to reconstruct the projected grid structure. We assume that ordering constraint is respected, which means that detected nodes in the "projected grid image" will appear in the same order as nodes in the "source grid". In practice this

constraint is always verified because scene objects must have piecewise smooth surfaces which presents smooth slope with respect to the grid nodes spacing.

To organize detected nodes, horizontal and vertical grid curves are first extracted from the binary skeleton image. The purpose is to extract each imaged grid curve, point per point, independently from the other curves. Then, only curves which contains grid nodes (figure 6) are selected and organized from left to right and from top to bottom. Finally, the projected grid structure is reconstructed by determining the intersection nodes between ordered horizontal and vertical node curves.

The elaborated method allows to reconstruct the projected grid even though all nodes may not be detected, one just has to find at least one node by horizontal and vertical node curves for the grid structure to be reconstructed.

Actually, projected grid reconstruction provides solution for the correspondence problem between projected grid and original one, which allows to achieve 3D points reconstruction.

2.4 3D Point Reconstruction

For each correspondence between projected grid node and original grid node, two set of equations are established from (1) and (2), such as:

$$\begin{aligned} u &= \frac{M_{c11}\cdot x + M_{c12}\cdot y + M_{c13}\cdot z + M_{c14}}{M_{c31}\cdot x + M_{c32}\cdot y + M_{c33}\cdot z + M_{c34}} \\ v &= \frac{M_{c21}\cdot X + M_{c22}\cdot Y + M_{c23}\cdot Z + M_{c24}}{M_{c31}\cdot X + M_{c32}\cdot Y + M_{c33}\cdot Z + M_{c34}} \\ u_g &= \frac{M_{p11}\cdot x + M_{p12}\cdot y + M_{p13}\cdot z + M_{p14}}{M_{p31}\cdot x + M_{p32}\cdot y + M_{p33}\cdot z + M_{p34}} \\ v_g &= \frac{M_{p21}\cdot x + M_{p22}\cdot y + M_{p23}\cdot z + M_{p24}}{M_{p31}\cdot x + M_{p32}\cdot y + M_{p33}\cdot z + M_{p34}} \end{aligned} \tag{15}$$

Coordinates x, y and z of each projected node P within the world coordinate system, are computed by solving this system of 4 linear equations. Once again, the least square method is used to solve this equations system.

2.5 Accuracy evaluation

A series of tests have been carried out to estimate the accuracy of the 3D coordinates computed with such a triangulation from the calibrated sensor parameters (Guisser, Payrissat, & Castan, 2000). In practice, the vision system accuracy is evaluated on a manufactured 3D object with different orientations (figure 7-a) as follows:

1. Project the grid onto the manufactured object so that the grid nodes are projected on the reference planes (P_1P_3) for the test 1 and (P_1P_2) for the test 2 (figure 7-b).
2. Compute the 3D coordinates of the projected grid nodes by triangulation, and use them to estimate the corresponding planes equations.
3. the 3D accuracy is evaluated by computing the distances between the estimated planes computed in (2) and their corresponding reference planes of the manufactured object.

Table (1) reports measurements given by the elaborated vision system. The mean error of reconstruction is about 0.9 mm for a projector/object distance of 2 m, and so implies a reconstruction accuracy lower to 1%.

3 3D Object Recovering

This section describes, first, the solution for the 3D free form object pose recovering problem, and second, how the developed vision system, presented above, is used to perform this recovering.

3.1 proposed approach

The main objective of 3D recovering is to find the geometric transformation (respectively translation + rotation) applied on a known object in the world coordinate system from one or several views of this object. This process is performed by matching features computed from the object model with features extracted from the 3D scanned object wherever it may be within the world coordinate system. Our matching method is based on the following concept: Since a free form object has a well defined surface normal which may be continuous almost everywhere except at peaks, sharp edges and cusps (Besl, &

Jain, 1985), one can consider some particular object regions called surface patches. Two features are then defined in order to characterize these local surface patches, the angles distribution and the surface patch curvature. Such features are invariant with respect to rotation and translation (Zisserman et al., 1995). Each surface patch in the object has its own angles distribution and curvature which could be identified even though the object is moved around (see figure 8). The combination of these two features make the matching process very reliable. The architecture of our 3D object recovering system is presented in figure (9).

3.2 Model Database

The surface patches of the free form objects used in this study are modeled with a triangular mesh obtained from scattered 3D points of the object measured with the vision system and expressed on the world coordinate system. For every patch vertex v_i, the distribution of angles between the normal to v_i and all normals to vertices in the neighborhood is computed. Furthermore, for each surface patch, gaussian curvature ($K=K_1*K_2$) and mean curvature ($H=(K_1+K_2)/2$) are computed compute to classify surface patch shape into eight basic categories (Durai, & Jain 1997) (see table 2), where K_1 and K_2 are the principal curvature characterizing the rate of change of surface patch orientation in the two extremal directions. In practice, a discrete curvature (Thirion, 1996) that applies directly to triangulated data is used. The discrete gaussian curvature K of a vertex is related to angles and faces that are connected to that vertex (see figure 10), and is given by:

$$K=\frac{2\pi-\sum_{i=1}^{k}\phi_i}{\frac{1}{3}A} \tag{16}$$

where $A=\sum_{i=1}^{k}S_i$ is a sum of each triangle's area, and ϕ_i denotes the angle at a vertex between the triangle edges.

Similarly the discrete mean curvature is given by:

$$H = \frac{\sum_{i=1}^{k} m(e_i)}{\frac{1}{3}A} \tag{17}$$

where e_i is an incident edge of a vertex and $m(e_i)$ is its curvature function defined by:

$$m(e_i) = \begin{cases} \gamma & if\ e_i\ is\ convex \\ 0 & if\ e_i\ is\ plane \\ -\gamma & if\ e_i\ is\ concave \end{cases} \tag{18}$$

angle γ is the angle between two adjacent surface normal (see figure 10)

Finally, a hash table (Lambdan, & Wolfson, 1988) is constructed to store the previously defined surface patches where the angular distributions and the patch curvatures are used as indexes and the information's stored are respectively the 3D coordinates and the normal vectors of the corresponding vertices within the world coordinate system.

3.3 3D/3D Matching and Verification

At recovering time, the vision system described in section (2) is used to extract several local surface patches of the object. Local regions presenting high curvature are chosen to be reconstructed first because in such regions angular distribution and curvatures are more discriminative and are relatively easy to identify in the hash table (Ababsa, Roussel, Mallem, & Didier, 2002). For each reconstructed surface patch, the angular distribution and curvatures are computed. These values are compared to the ones stored in the model data base, if a correspondence occurs, a surface patch recognition hypothesis is then generated. The verification module allows to discard false recognition hypotheses. For each generated hypotheses, the corresponding geometric transformation is computed, and then the model surface patches are back projected onto the image plane. Only hypotheses that projects closer to their corresponding image features are considered (Huttenlocher, & Ullman, 1990).

3.4 3D Localization

Let us consider a matching hypotheses case generated by the 3D/3D Matching module described above. Each correspondence between surface model patch and extracted one allows to identify a vertex and its normal when the object is in its new position (see figure 11). So 3D localization problem consists in finding the geometric transformation (translation *T* and rotation *R*) applied on the object initial position in the word coordinate system to bring it in the current position. Such a problem can be formulated as follows:

Let V_i be a 3D point of the object at its initial position (object model). Let $V_i^{'}$ be the point V_i displaced by the transformation we are looking for, and let $\vec{N}_i\left(N_{xi} \quad N_{yi} \quad N_{zi}\right)^T$ and $\vec{N}_i^{'}\left(N_{xi}^{'} \quad N_{yi}^{'} \quad N_{zi}^{'}\right)^T$ be the normal vectors to V_i and $V_i^{'}$ respectively. $\vec{N}_i^{'}$ coordinates are given in world coordinate system by:

$$\vec{N}_{i/R_w}^{'} = R \times \vec{N}_{i/R_w} \tag{19}$$

In order to determine *R*, we have to minimize the following criterion

$$C = \sum_{i=1}^{N_m} \left\| \vec{N}_i^{'} - R \times \vec{N}_i \right\|^2 \tag{20}$$

where N_m denotes the number of matched surface patches obtained by the 3D/3D matching module.

The rotation axis and angle representation $(\vec{n},\theta)$ are used to express the rotation matrix *R* as follows (Rodrigues Formula):

$$R = \begin{bmatrix} c+(1-c)n_x^2 & -n_z.s+(1-c)n_x.n_y & n_y.s+(1-c)n_x.n_z \\ n_z.s+(1-c).n_x.n_y & c+(1-c)n_y^2 & -n_x.s+(1-c)n_z.n_y \\ -n_y.s+(1-c)n_x.n_z & n_x.s+(1-c)n_z.n_y & c+(1-c)n_z^2 \end{bmatrix} \tag{21}$$

where $\vec{n}=\left[n_x \quad n_y \quad n_z\right]$, $\|\vec{n}\|=1$, $s=sin\theta$ and $c=cos\theta$

Considering the two following properties of a rotation: the scalar product and the vector module are both preserved by rotation ($\vec{n}\cdot\vec{N}_i=\vec{n}\cdot\vec{N}_i'$ and $\left\|\vec{N}_i\right\|=\left\|\vec{N}_i'\right\|$), we could build a 2xN_m linear equation system with 3 unknowns out of N_m matched features:

$$\begin{bmatrix} \vdots & \vdots & \vdots \\ 0 & -\left(N_{zi}+N_{zi}'\right) & N_{yi}+N_{yi}' \\ N_{zi}+N_{zi}' & 0 & -\left(N_{xi}+N_{xi}'\right) \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix} = \begin{bmatrix} \vdots \\ N_{xi}-N_{xi}' \\ N_{yi}-N_{yi}' \\ \vdots \end{bmatrix} \tag{22}$$

where,

$$\vec{\Omega}=tan\left(\frac{\theta}{2}\right)\vec{n} \tag{23}$$

At least two matched features are necessary to solve this system. The least square method is used to compute an approximate solution for this system, so that:

$$\begin{cases} \vec{n}=\dfrac{\vec{\Omega}}{\left\|\vec{\Omega}\right\|} \\ \theta=2\cdot arctan\left(\left\|\vec{\Omega}\right\|\right) \end{cases} \tag{24}$$

therefore, the rotation matrix R is computed using equation (21).

To determine the translation vector T, one can notice that the V_i' points coordinates are given in the world coordinate system by:

$$V_{i/R_w}'=R\times V_{i/R_w}+T \tag{25}$$

however, the coordinates of the matched vertices V_i and V_i' in the world coordinate system are known. So for each two matched vertices, the corresponding translation vector T_i is computed by:

$$T_i=V_{i/R_w}'-R_{opt}\times V_{i/R_w} \tag{26}$$

Finally, the translation vector is given by:

$$T=\frac{1}{N_m}\cdot\sum_{i=1}^{N_m}T_i \tag{27}$$

When the computed transformation is applied to the model vertices and the transformed vertices are back projected onto the image plane, a remaining error between image features and their corresponding image model features can be noted, this is due to the computation errors introduced by the data processing algorithms used in the several modules of the recovery system which are generally cumulative.

So in order to refine the computed transformation, an optimization procedure based on Levenberg-Marquardt method is used to minimize the mean 2D square distance between accurate nodes extracted from scene image and their corresponding image model vertices. Let D be the criterion to minimize, it is defined as:

$$D=\sum_{i=1}^{N_m}\left\|\hat{m}_i-m_i\right\|^2 \tag{28}$$

where $\hat{m}_i$ are the accurate nodes extracted from the object image and $m_i(u_i,v_i)$ are given by:

$$\begin{bmatrix} s.u_i \\ s.v_i \\ s \end{bmatrix}=Mc_{3x4}\cdot\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}\cdot V_i \tag{29}$$

The parameters vector P to estimate is then composed of the rotation and translation components. A unit quaternion Q is used to define rotation R. Q is given by it's four components as follows:

$$Q=\begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T \quad \text{where} \quad q_0^2+q_1^2+q_3^2+q_4^2=1$$

Using the unit quaternion Q allows to have an unique solution for the inverse rotation problem, and furthermore it doesn't present any singularity. Thus the vector P to estimate is composed of six parameters, the three components of the translation vector and three components of the unit quaternion Q:

$$P=\begin{bmatrix} q_1 & q_2 & q_3 & T_x & T_y & T_z \end{bmatrix}^T \tag{30}$$

A good initialization of the parameter vector P is necessary to ensure the convergence of the algorithm, so the transformation (R , T) obtained previously is used to initialize the vector P.

4 Experimental Results

In order to illustrate our 3D object recovering method, a particular example of a free form object has been considered, in this case the mask of lion (figure 12). The vision system is used to scan the object. Obtained 3D points are then locally meshed. For each surface patch, 25 vertices are reconstructed. A hash table is then constructed as explained in subsection (3.2).

Figure (13) depicts the intensity image of the object in its initial position. It also shows an example of two particular surface patches models used during the model database construction phase, figures (13-b) and (13-d) show angular distributions and surface curvature (gaussian + mean curvature) stored in the hash table. To perform localization of the object when it is moved around, at least two local regions of the object are scanned. The object surface patches are then constructed from extracted points and recognized in the hash table.

Figure (14), depicts the image of the object in its new unknown position. Regions to be extracted are emphasized by squares. In the same way, Figures (14-a), (14-b), (14-c) and (14-d) illustrate the extracted local surface patches and their angular distributions and surface curvatures. Using these features, the developed matching algorithm has well identify the extracted patches with model ones in the hash table.

Once the 3D/3D matching is done, matched features are fed to the 3D localization algorithm in order to find the current position of the object. Finally, the transformation computed above is used to initialize the refined localization of the object.

Figure (15) shows that when the refined transformation is applied to the model surface patches and re-projected, the obtained points onto the image fit closer to the extracted surface patches. This demonstrate the robustness and the pose accuracy of our method.

Table (3) illustrates the accuracy of the developed 3D recovering algorithm. It gives the mean square distance E_d between the projected vertices of the model surface patches and their corresponding nodes in the image, for the initial and the refine transformations, where E_d is defined as equation (28).

The proposed algorithm is compared to other approaches using real generated data. The methods tested here are those implemented by Shaheen, Mallem, and Chavand (2001), in this case the geometric three-segment method that gives an initial guess of the pose, the non-linear method based on the last pose found, the linear method based on the initial guess of the pose and using the rotation vector formalism, and the author's proposed algorithm called mixed method. Table 4 illustrates error statistics of the four methods (in pixels). The Obtained results demonstrate the important improvement provided by the refined localization to the object pose computation compared to the four methods.

Moreover, The proposed method can be applied even though the object is partially occluded because it is a local approach and it always has the possibility to extract visible region of the object.

5 Conclusion

In this paper a new solution to the 3D free form object recovering problem is presented. A vision system based on a manufactured grid projector and a CCD camera is conceived to achieve this solution. Calibration procedure of such a system is also studied. A robust sub-pixel model based approach to extract grid nodes from captured images had been developed and used in both camera calibration and 3D points reconstruction procedures. The results obtained from experimental tests demonstrates the accuracy of obtained 3D measurements (less than 1%). This result is crucial for the object pose recovering, since it allows to well reconstruct local surface patches and so to generate matching hypotheses more precisely. The possibility to control the projector’s orientation allows to scan particular local regions of the object and therefore extract 3D surface patches which are exploited to perform 3D object recovering. In the second part of this paper, the architecture of the developed object recovering system is presented in details. The construction of the object data base from 3D scattered points is described and the mathematical solution of the localization and refined localization problems are also given. The experimental results are very satisfying, the algorithm succeeds in determining the object pose by matching extracted surface patches with model ones using only their angular distributions and surface curvature. The obtained transformation is used to refine the object pose, this increases the robustness of

the algorithm to noise and computation errors. Experimental testing has shown that our proposed algorithm is generally more accurate than the classical methods. The image error obtained is lower to the hundredth of the pixel for a camera-object distance equal to 2 m for real images. Moreover, the proposed method can be applied even though the object is partially occluded. The current research efforts are focused on the generalization of the developed solution to find and identify multiple objects in scenes with any possibility of occlusion in order to increase its effectiveness in real environments. For such multiple object scenes large database studies need to be applied, therefore we are addressing the problem of accurately an quickly indexing the correct model. Success in this endeavor would create a complete 3D free form object recognition system.

References

Ababsa, F., Roussel, D., Mallem, M., & Didier, J.Y. (2002). 2D/3D automatic matching technique for 3D recovering of free form objects. *16th IAPR-IEEE International Conference on Pattern Recognition*. Quebec City, Canada. August 11-15, 2002, Vol. II, pp. 430-433.

Besl, P.J., & Jain, R. C. (1985). Three dimensional object recognition. *Computing Surveys* 17(1), 75-145.

Besl, P.J. (1990). The free form surface matching problem. (H. Freeman, ed.), *Machine Vision for Three Dimensional Scenes*, Academic Press, 25-71.

Besl, P.J., & McKay, N.D. (1992). A Method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239-256.

Cambpell, R., Flynn, P. (2001). A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding* 81(2). 166-210.

Chua, C. S., & Jarvis, R. (1997). Point signature : a new representation for 3D object recognition. *International Journal of Computer Vision* 17, 63-85.

Deriche, G.R., & Giraudon, G. (1993). A computational approach for corner and vertex Detection. *International Journal of Computer Vision* 10(2), 101-124.

Dorai, C., & Jain, A.K. (1997). COSMOS- a representation scheme for 3D free form objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(10), 1115-1130.

Faugeras, O. (1993). Three dimensional computer vision: a geometric viewpoint. MIT Press.

Guisser, L., Payrissat, R., & Castan, S (2000). PGSD: an accurate 3D vision system using a projected grid for surface descriptions. *Image and Vision Computing*. 18, 463-491.

Hu, G., & Stockman, G. (1989). 3D surface solution using structured light and constraint propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11(4), 390-402.

Huttenlocher, D. P., & Ullman, S. (1990). Recognizing solid objects by alignment with an image. *International Journal of Computer Vision* 5(2). 195-212.

Izquierdo, M.A.G., Sanchez, M.T., Ibanez, A., & Ullate, L.G. (1999). Sub-pixel measurements of 3D surfaces by laser scanning. *Sensors and Actuators Journal* 76. 1-8.

Johnson, A., & Hebert, M. (1999). Using spin-images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21(5), 433-449.

Lambdan, Y., & Wolfson, H. J. (1988). Geometric hashing : a general and efficient model-based recognition scheme. *IEEE Conference on Computer Vision* 1988 (pp. 238-289). Florida, USA

Murase, H., & Nayar, S. K., (1995). Visual learning and recognition of 3D objects from appearance. *International Journal of Computer Vision*, 14:5-24.

Nutbourne, A. W., & Martin, R. R. (1988). Differential geometry applied to curve and surface design. Ellis Horwood.

Pennec, X (1998). Toward a generic framework for recognition based on uncertain geometric features. *Journal of Computer vision research*. 58-87

Rocchini, C., Cigoni, P., Montani, C., Pingi, P., & Scopigno, R. (2001). A low cost 3D scanner based on structured light. *Eurographics Conference* (pp. 299-308). Interlaken, Switzerland.

Shaheen, M., Mallem, M., & Chavand, F. (2001). Visual command of a robot using 3D-scene reconstruction in an augmented reality system. *Control Engineering Practice*, 9:375-385.

Thirion, J. P. (1996). The extremal mesh and the understanding of 3D surfaces. *International Journal of* Computer Vision 19(2). 115-128.

Valkenburg, R.J., & McIvor, A.M. (1998). Accurate 3D measurement using structured light system. *Image and Vision Computing* 16(2), 99-110.

Zisserman, A., Forsyth, D., Mundy, J., Rothwell, C., Liu, J., & Pillow, N. (1995). 3D object recognition using invariance. *Artificial Intelligence* 78. 239-288.

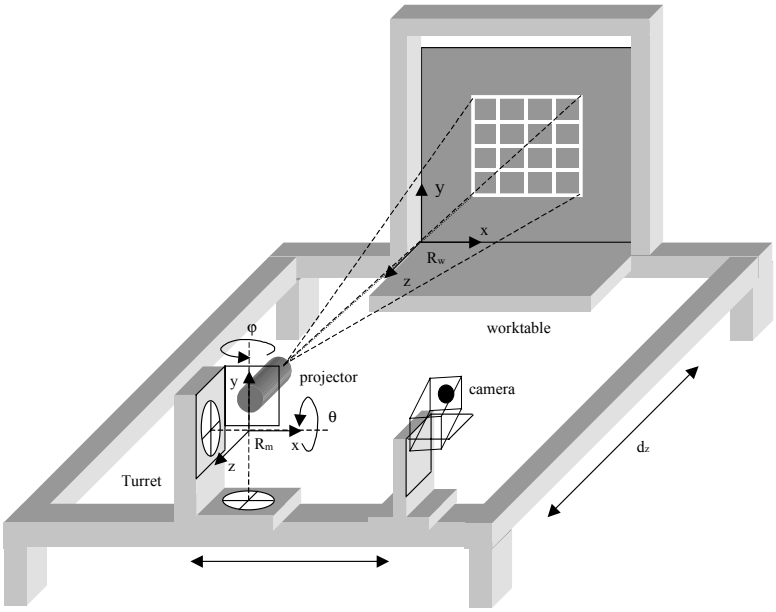


Figure 1: Vision workbench

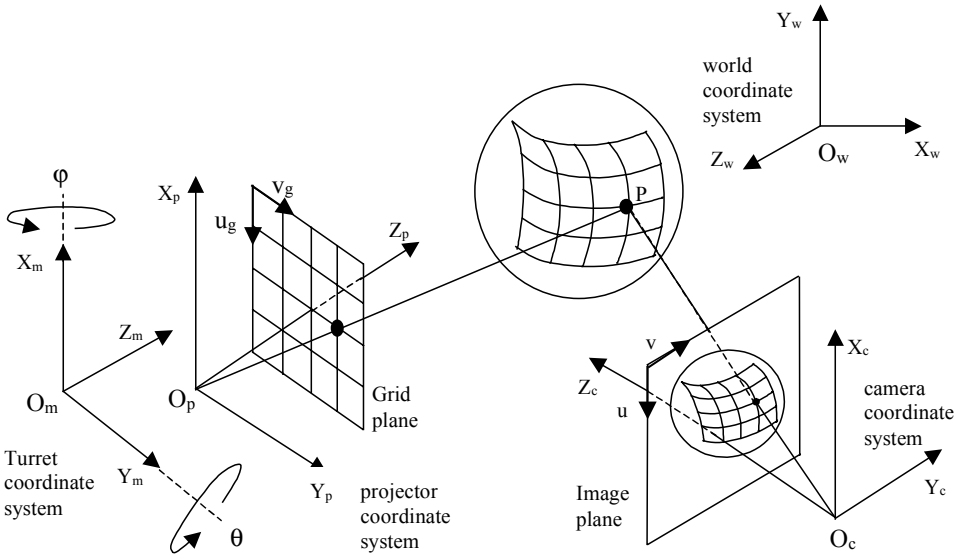
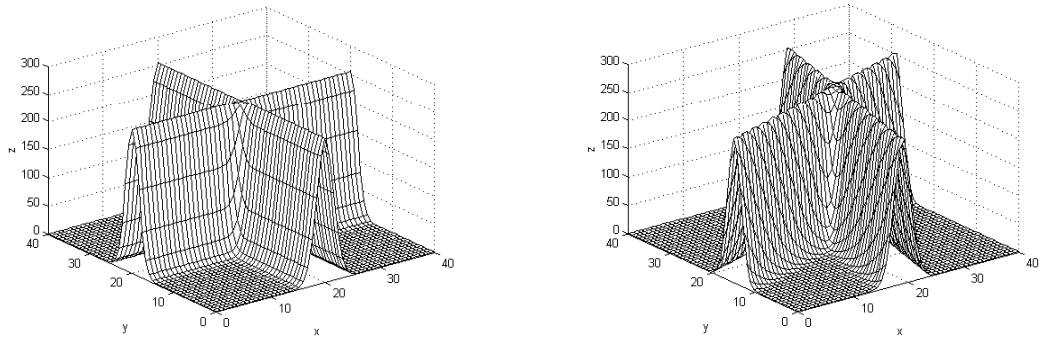


Figure 2: Coordinates system used for calibration and reconstruction



a=1 b=254 c=0.2 d=0
e=20 f=0.2 g=0 h=20
a) right cross

a=1 b=254 c=0.2 d=0.25
e=19 f=0.2 g=0.35 h=15
b) oblique cross

Figure 3: Cross model examples

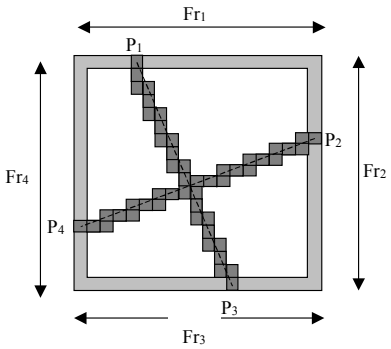
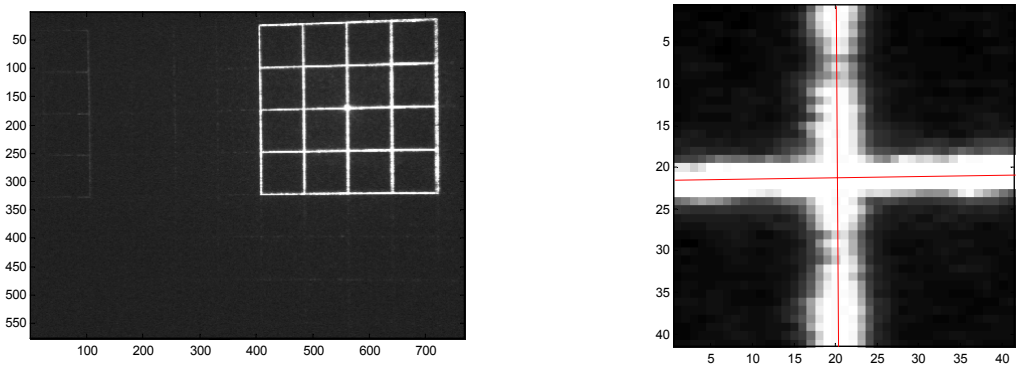


Figure 4: Approximate node sub-image



a) projected grid image

b) cross sub-image

Figure 5: Grid and cross image examples

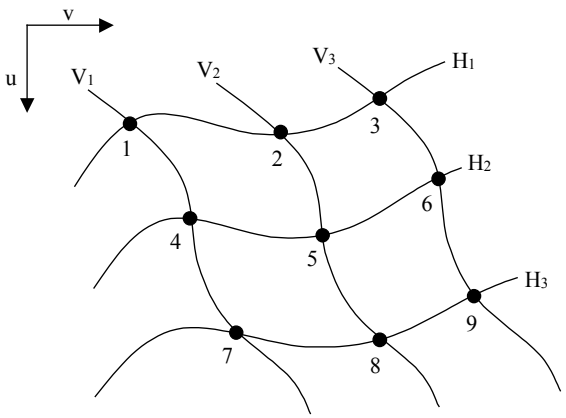
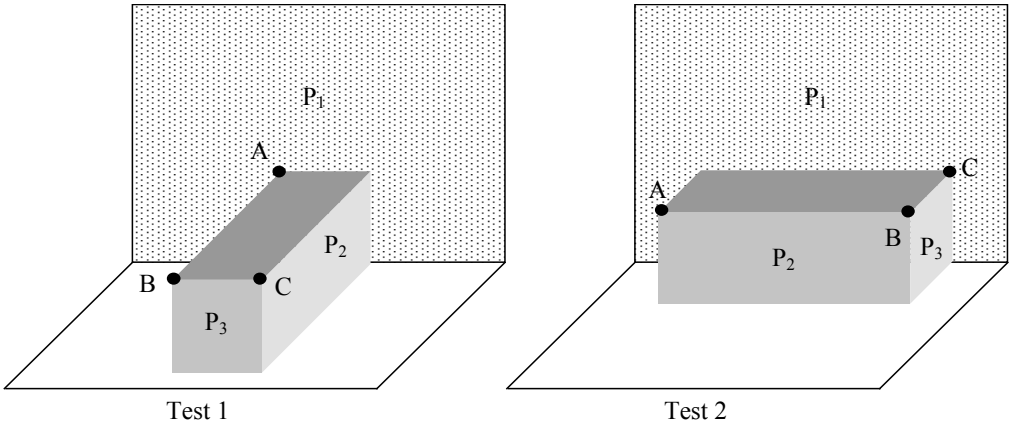
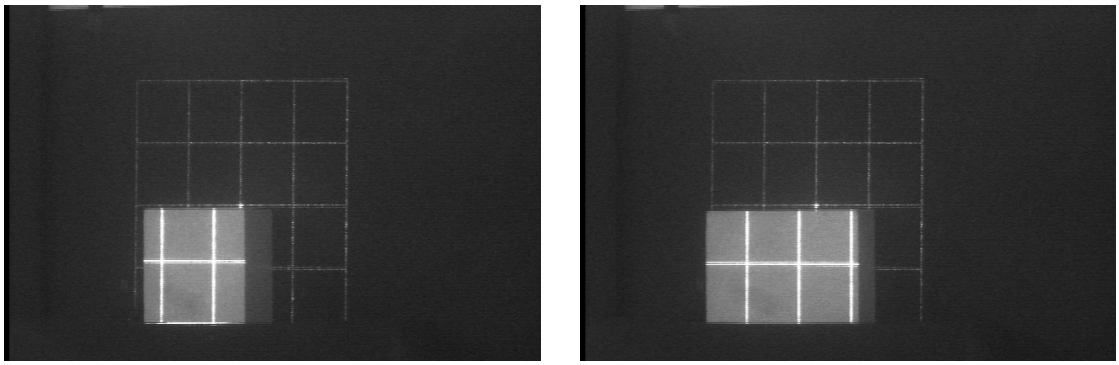


Figure 6: Projected grid node structure



(a)



(b)

Figure 7. Reconstruction accuracy : (a) manufactured object with different orientations. (b) projected grid images

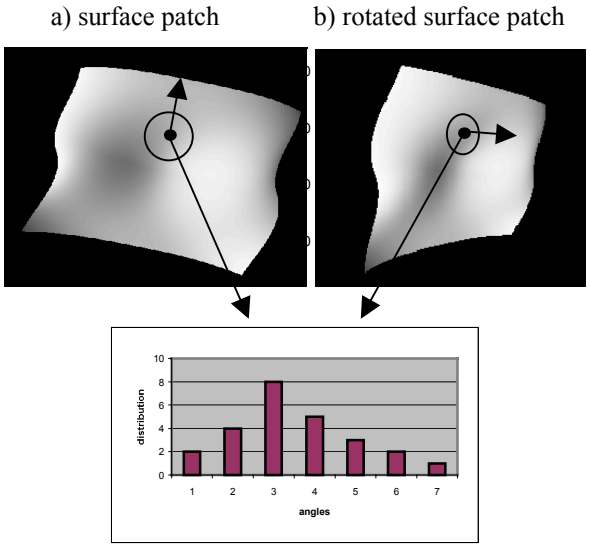


Figure 8: Principle of the surface patches matching

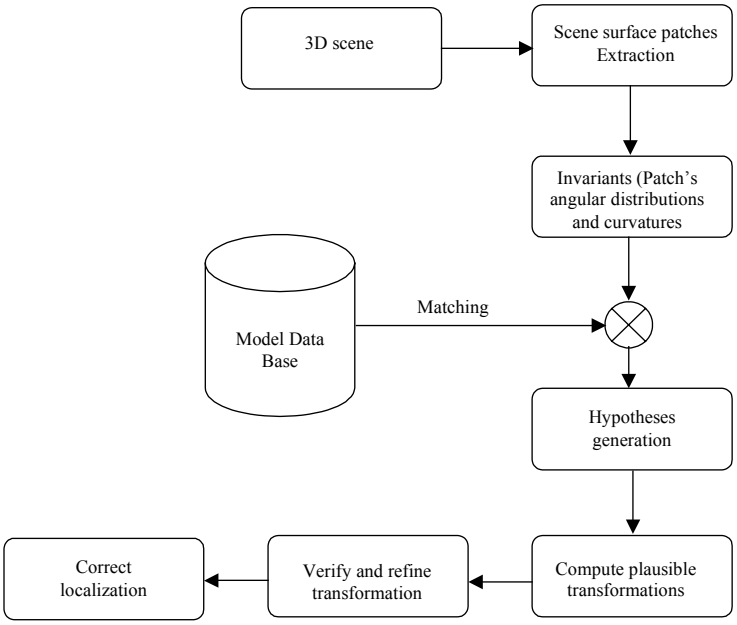
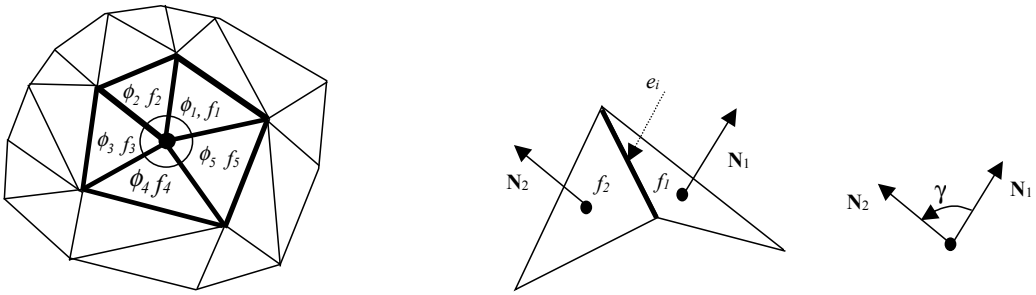


Figure 9: 3D object recovering system architecture



a) The gaussian curvature b) the mean curvature

Figure 10: The discrete curvature of a vertex

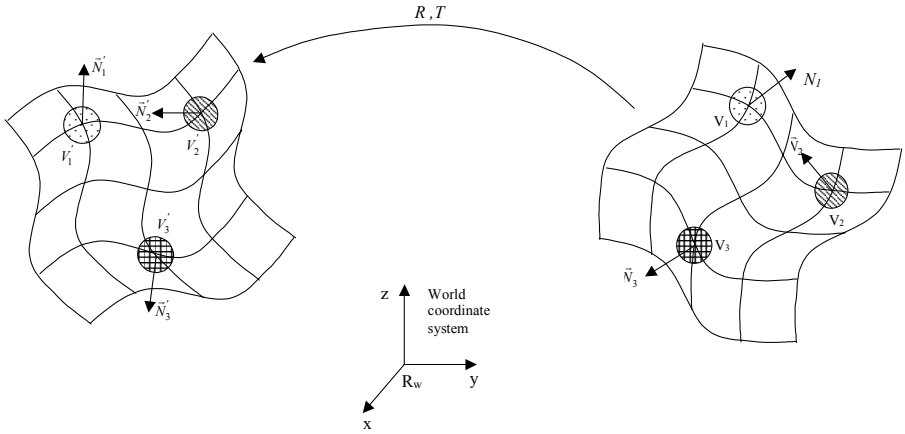


Figure 11: 3D localization

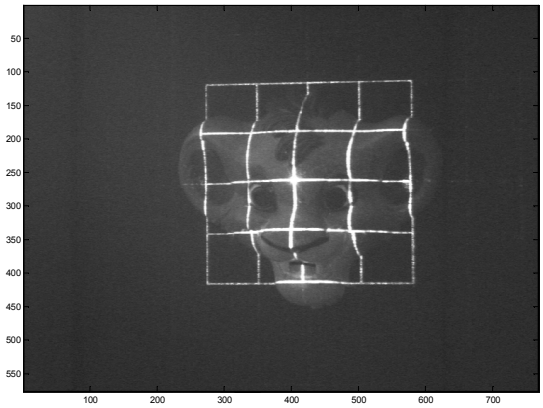


Figure 12: Projected grid on lion mask

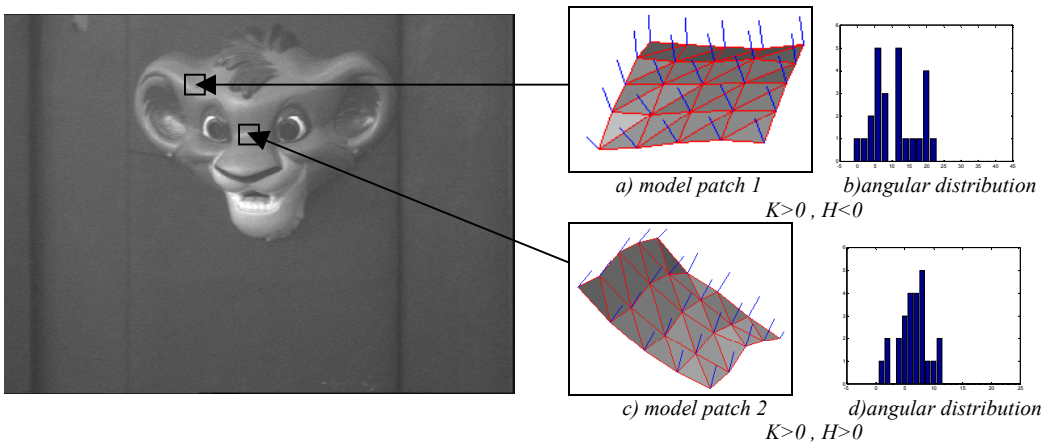


Figure 13: Surface model patches example

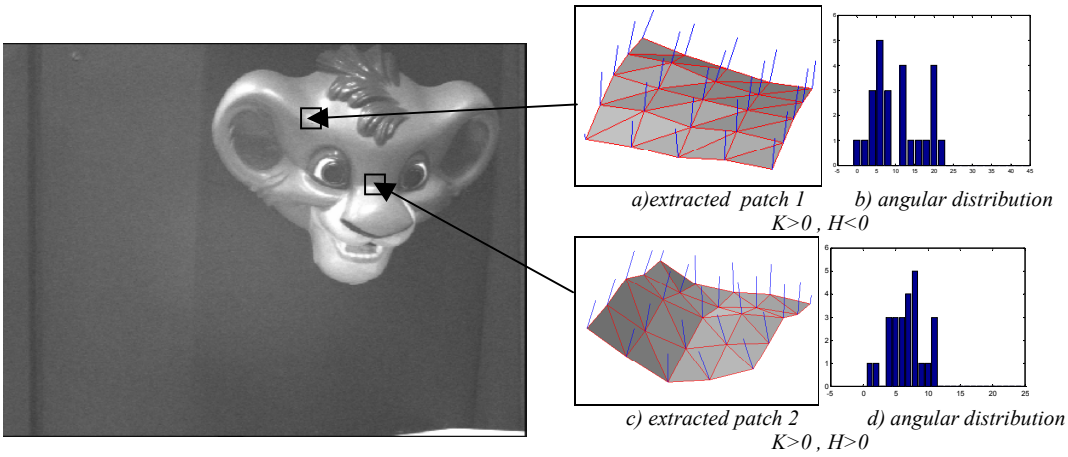


Figure 14: Object extracted patches

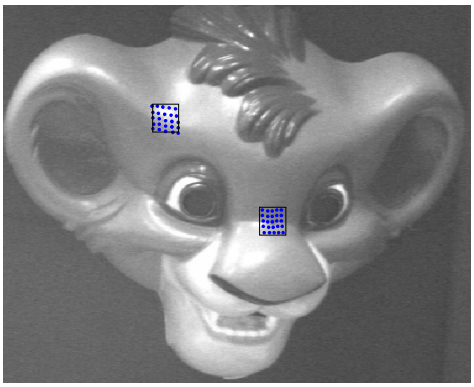


Figure 15: Object recovering patches

	Manufactured object (mm)	System measurement (mm)	residual error (mm)	Error %
Test 1				
Segment [A,B]	140	138.8	1.2	0.85
Test 2				
Segment [B,C]	90	89.4	0.6	0.66

Table 1 : Measurement Results given by the developed vision system

	H<0	**H=0**	**H>0**
K<0	Saddle Ridge	Minimal	Saddle Valley
K=0	Convex Cylinder	Plane	Concave Cylinder
K>0	Convex Ellipsoid	Impossible	Concave Ellipsoid

Table 2: Classification of surface patch shape using Mean H and Gaussian K curvature

	Transformation	**Refine Transformation**
Mean square error E_d (pixels)	3.736	0.274×10^{-2}

Table 3 : **Recovering errors**

	3-segment	**Non-linear**	**Linear**	**Mixed**
Mean square error E_d (pixels)	2.072	1.070	0.967	0.749

Table 4: Error statistics of the four methods (in pixels)