A Coarse-to-Fine Algorithm for 3D Registration based on Wavelet Decomposition

C. TORRE-FERRERO, S. ROBLA, E.G. SARABIA, J.R. LLATA
Electronics Technology, Systems and Automation Engineering Department
University of Cantabria
Av. Los Castros s/n. 39005 Santander
SPAIN
[carlos, srobla, esther, llata]@teisa.unican.es

Abstract: - This paper introduces a coarse-to-fine algorithm for 3D registration that uses an alternative 3D Shape Representation: CIRCON (Cylindrical Image of Radial CONtours). Two point clouds can be aligned using their corresponding *circon* images for searching for correspondences. In order to accelerate this search, a coarse-to-fine approach based on wavelet image decomposition will be applied to the *circon* images. This permits to reduce, for each resolution level, the amount of indexes to be evaluated. The information obtained by the surface matching process is then used for calculating a coarse transformation (translation and rotation) between both surfaces; in such a way that two point clouds whose relative pose is unknown can be aligned.

Key-Words: - 3D registration, surface alignment, shape representation, coarse transformation, object recognition.

1 Introduction

In recent years, several areas of computer vision have been dedicated to finding a solution to problems of industrial scope. As a consequence, certain applications, such as computer-aided manufacturing or bin-picking, have made necessary the use of 3D information of the parts in order to be automated. If this 3D information is acquired by using a laser scanner or similar methods, a range image is obtained for every scan. Therefore, it will be indispensable to have at our disposal a set of range images in order to build a complete 3D model of a part.

If no information about the Euclidean transformation that relates the range images is available, a method for recovering it will be needed. This problem of 3D registration problem has given rise to many surface matching algorithms that use either intrinsic or extrinsic surface properties in order to find a correspondence between two point clouds whose relative pose is unknown.

The intrinsic properties relate the surface to itself whereas the extrinsic ones relate it to the coordinate frame in which the surface is expressed. Both types of properties give rise to different approaches.

On the one hand, the extrinsic algorithms, such as ICP and its variants [1],[2], can obtain a very accurate alignment but this depends on a coarse initial transformation where the algorithm is initialized. On the other hand, the intrinsic algorithms can obtain the Euclidean transformation that aligns the two point clouds with no need for previous coarse alignment. These surface matching algorithms depends on the chosen characteristics (such as spin images, geometric histogram, surface signatures, harmonic shape images, etc). All these methods extract the 3D shape descriptors from both surfaces in order to compare them. If many correspondences are found, then a coarse transformation that aligns them in a proper way can be calculated.

In the following sections a coarse-to-fine algorithm for surface matching is presented (Section 3 and 4) that uses an alternative 3D shape representation (Section 2). The results provided by this process allow obtaining directly a coarse transformation between the two surfaces (Section 5).

2 CIRCON: 3D Shape Representation

2.1 3D Shape Representations

Diverse types of coordinates are used by some well-known shape representations in different ways in order to index different geometric properties. Next, a concise description of the most important ones is made. A more complete analysis can be found in [7].

2.1.1 Spin Images

This representation extracts features based on horizontal and perpendicular distances from regions around an interest-point. These features are represented using smoothed 2D histograms known as spin-images [3].

2.1.2 Geometric histograms

The angles between normals and perpendicular distances are extracted as features from the regions.

These features are accumulated in smoothed 2D geometric histograms [4].

2.1.3 Surface signatures

This representation uses the whole surface around the interest-point. Distance and angular features are extracted from the region and are represented using 2D signatures [5].

2.1.4 Harmonic shape image

Features based on curvature are extracted from the regions to create 2D representations known as harmonic shape images [6].

2.2 Cylindrical coordinates

Our 3D shape representation is based on cylindrical coordinates, hence a brief review of their expressions may be necessary.

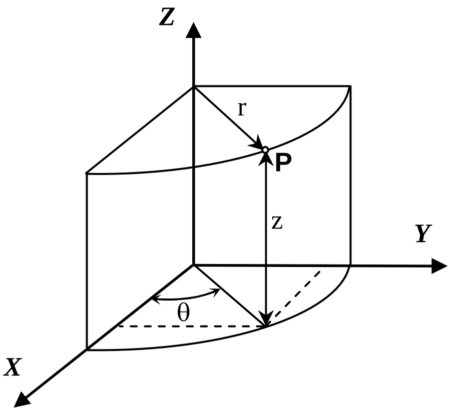


Fig. 1. Cylindrical coordinates.

As it can be inferred from Fig.1, cylindrical coordinates (r, θ, z) are related to Cartesian coordinates (x,y,z) by,

$$r = \sqrt{x^2 + y^2} \tag{1}$$

$$\theta = \tan^{-1}\left(\frac{y}{x}\right) \tag{2}$$

$$z = z \, . \tag{3}$$

Note that (2) is undefined at r=0.

2.3 Description

As is shown in Fig.2, the generation process of this 3D shape representation can be visualized as a plane that is rotated around the surface normal at one point P_0 of the object, producing a slice for each angle θ_i being considered. The contours of these slices are coded and stored in a matrix.

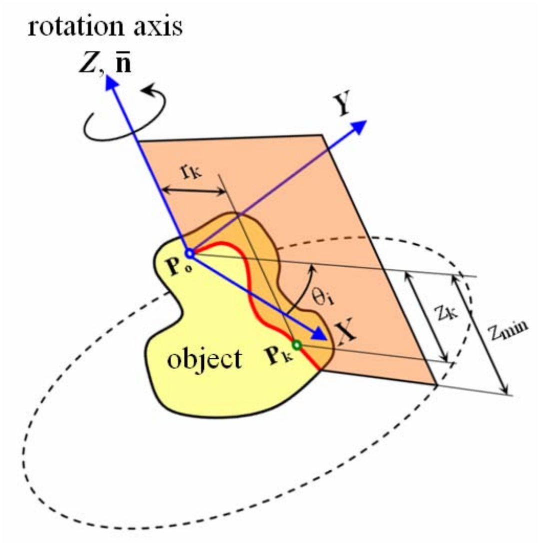


Fig.2. Generation of the 3D shape representation.

The first step is referring all the points in the point cloud to a coordinate frame whose origin is situated in P_0.

If a transformation based on cylindrical coordinates is defined, given a 3D point P_k belonging to a *slice i* (angle θ_i) and whose Cartesian coordinates are (x_k, y_k, z_k), these could be transformed by using the following expressions:

$$i = round\left(\frac{-\tan^{-1}\left(\frac{y_k}{x_k}\right)}{\rho_\theta} + 1\right) \tag{4}$$

$$j = round\left(\frac{\sqrt{x_k{}^2 + y_k{}^2}}{\rho_r} + 1\right) \tag{5}$$

$$c_{i,j} = round\left(\frac{z_k}{\rho_z}\right) \tag{6}$$

where ρ_θ is the angular resolution, ρ_r the radial resolution and ρ_z the height resolution; $c_{i,j}$ is the matrix element that belongs to i row and j column.

$$\begin{array}{c|cccccc} & \text{radius} \rightarrow & & & & & \\ \text{angle} \downarrow & c_{1,1} & c_{1,2} & \cdots & c_{1,j} & \cdots & c_{1,nr} \\ & c_{2,1} & c_{2,2} & \cdots & c_{2,j} & \cdots & c_{2,nr} \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & c_{i,1} & c_{i,2} & \cdots & c_{i,j} & \cdots & c_{i,nr} \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & c_{ns,1} & c_{ns,2} & \cdots & c_{ns,j} & \cdots & c_{ns,nr} \end{array}$$

Fig. 3. CIRCON matrix. The arrows shown in the figure indicate the increasing direction for angle and radius.

By applying this transformation to all the *slices*, the results can be stored in a matrix, as is shown in Fig.3. This matrix can be visualized as an image (see Fig.4), where the row indexes indicate the number of angular section (slice) whereas the columns indexes code the radius and the image grey levels code the heights.

The matrix will have $n_S=2\pi/\rho_\theta$ rows, one for each prefixed rotation angle.

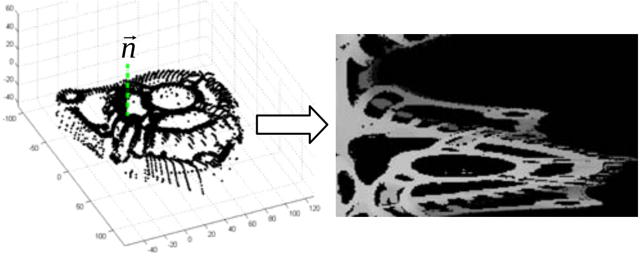


Fig. 4. CIRCON of an alternator cover (ρ_θ=2.5º, ρ_r=ρ_z=0.5mm).

2.4 Properties

An example of the resultant image of applying this transformation to the 3D points belonging to an arbitrary pose of an alternator cover is shown in Fig.4. The 3D data shown were obtained by the 3D vision system presented in [8], whereas the resolutions being used can be seen in the footnote.

It is interesting to note that the last and the first rows of the image keep continuity in the grey levels respectively. This is logical since the sequence of slices is closed. Hence, this is a kind of cylindrical image. Moreover, every row represents the object contour belonging to that radial plane (slice).

Consequently, the name we have chosen for this 3D shape representation is CIRCON (Cylindrical Image of Radial CONtours).

This 3D shape representation has the following main properties:

-- it is object-centred.
-- it can be used for global or local representation.
-- it is robust to clutter and occlusion.
-- it compress the 3D shape, making it more manageable.
-- the shape of the object can be recovered (though sampled).

A more detailed description can be found in [8].

3 Coarse-to-fine Algorithm Based on Wavelet Decomposition

3.1 Surface Matching

The main stage that has to be accomplished before calculating the transformation that aligns the two point clouds is the search for correspondences between them, namely, the surface matching.

Since the goal of this process is to obtain the best matching between two point clouds S_A and S_B corresponding to two different views of the same object, a global matching is performed, i.e., the *circon* images are generated using all the points on both point clouds. If the goal were not 3D registration but 3D object recognition in the presence of clutter, the matching should be local.

The results obtained by this process are used later for calculating a coarse transformation that aligns both surfaces.

The main drawback of the basic algorithm described in [9] was the difficulty to reduce the amount of indexes (i_o,j_o) to be tested using the basic block diagram of the surface matching process shown in Fig. 5.

Given two *circon* images, *A* and *B*, corresponding to different scans of the same object; in order to search correspondences between the two point clouds, the image *A* must be transformed by using a pair of indexes (i_o,j_o). The result of this transformation (see Subsection 4.1.) is another image, denoted by $A(i_o,j_o)$, that will be compared to *B* by means of a similarity measure (see Subsection 4.2.). The resultant values M_s will be used later for choosing the best correspondence between the two point clouds.

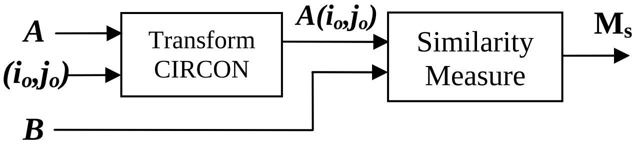


Fig. 5. Basic block diagram of the surface matching process.

3.2 Wavelet Decomposition

The approach presented in this paper uses wavelet decomposition in order to reduce progressively, in a coarse-to-fine way, the amount of indexes to be verified.

This method consists basically of decomposing the two *circon* images by using the Discrete Wavelet Transformation (DWT) [10].

This two-dimensional transformation leads to a decomposition of approximation coefficients at level *L* in four components: the approximation at level *L*+1, and the details in three orientations (horizontal, vertical, and diagonal). In turn, the approximation A_{L+1} can be decomposed in another four components, as is shown in Fig. 6.

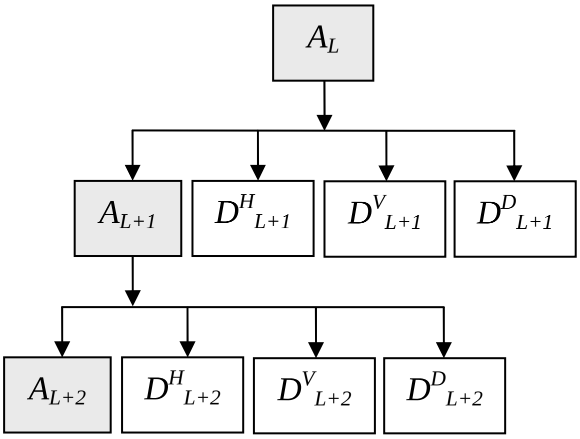


Fig. 6. Image decomposition using Discrete Wavelet Transformation.

The wavelet family that has been chosen for our experiments is the Haar wavelets [11], since these produced the best results for the surface matching algorithm which is explained in the next paragraph.

3.3 A Coarse-to-fine Approach

Given two views of the same object, whose initial misalignment is shown in Fig. 7, this coarse-to-fine algorithm tries to reduce the amount of information to be tested by using a multi-level processing. Firstly, two *circon* images (*A*, *B*) are generated in the closest point to the centroid of the point clouds (see Fig. 8). Following that, the two images obtained are decomposed into *N* levels and the *N* approximations A_L and B_L are taken to be processed as shown in Fig.9. The upper views of their corresponding 3D shapes are shown in Fig.10 (note that shape recovery is one of the properties of the *circon* images).

Next, correspondences are evaluated for the level-*N* images using the block diagram shown in Fig. 4 for every pixel in the image A_N that belongs to the object represented. The resultant similarity values permit to discard those correspondences (and hence, their associated Euclidean transformations) that do not reach a minimum of similarity.

In turn, the same evaluation will be repeated for the level-(*N-1*) images but it will be only applied to the indexes (i_o,j_o) that correspond to the pixels that have not been eliminated in the previous step. In such a way that, in every step, the possible correspondences are reduced progressively (see Fig. 11). Finally, a small group of correspondences will have been obtained; these will be verified in order to know which of them produces the best alignment between the two point clouds S_A and S_B (Subsection 4.4).

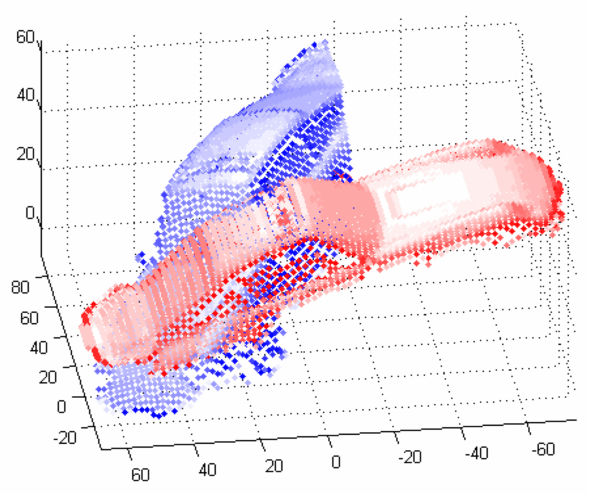
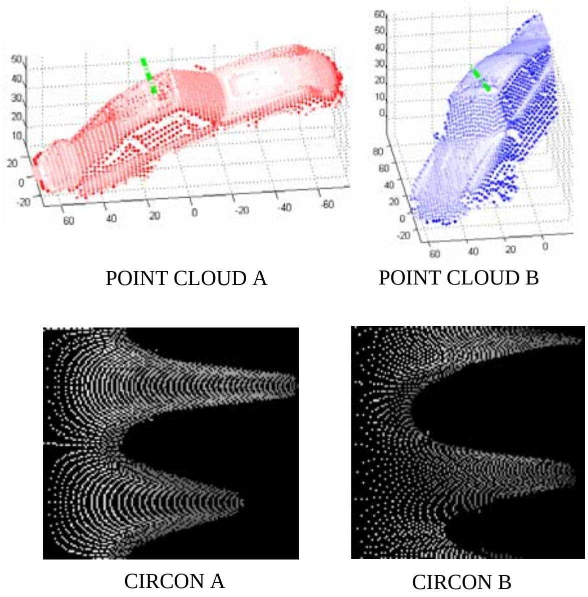


Fig. 7. Initial misalignment.



POINT CLOUD A

POINT CLOUD B

CIRCON A

CIRCON B

Fig. 8. CIRCON images corresponding to the two scans shown of the same object.

4 Details of the Algorithm

4.1 Transformation of a *circon* image

In order to find correspondences between the point clouds, a transformation of one of the two corresponding *circon* images has to be performed. This transformation modifies the original *circon*, producing a new one that will be compared with the other *circon* image.

Suppose that, as is shown in Fig. 5, *circon A* is to be transformed. If two indexes (i_o,j_o) are provided, this transformation modifies all the matrix elements and their original positions by centering the *circon* in the 3D point corresponding to the pixel with row index i_o and column index j_o in the image.

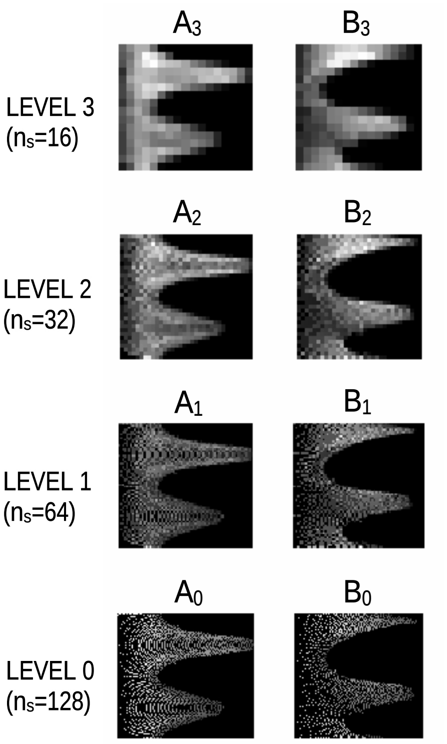


Fig. 9. Wavelet Approximations for three resolution levels of the two *circon* images. The two last images correspond to the original ones.

The transformation comprises three steps. Firstly, the following expressions are applied to all the pairs (*i*,*j*) in the matrix.

$$x'_{ij} = \rho_r \cdot \left(j \cdot \cos(\rho_\theta \cdot (i-1)) - j_o \cdot \cos(\rho_\theta \cdot (i_o - 1)) \right) \quad (7)$$

$$y'_{ij} = -\rho_r \cdot \left(j \cdot \sin(\rho_\theta \cdot (i-1)) + j_o \cdot \sin(\rho_\theta \cdot (i_o - 1)) \right) \quad (8)$$

The new position (*i'*,*j'*) of the pixel a_{ij} will be calculated by (1) and (2), whereas the pixel value by,

$$c_{i'j'} = c_{ij} - c_{i_o j_o} \quad (9)$$

The next step will be to refer the new *circon* to the surface normal at the 3D point where it is centered now. The normal is calculated by fitting a plane to the 3D points obtained from the neighboring pixels.

Hence, the data from the matrix will be rotated by using the following matrix:

$$R_n = \left[\bar{x}_n \quad \bar{n} \times \bar{x}_n \quad \bar{n} \right]^{\mathrm{T}}, \quad (10)$$

where

$$\bar{x}_n = \frac{\left[0 \quad 1 \quad 0\right]^{\mathrm{T}} \times \bar{n}}{\left\| \left[0 \quad 1 \quad 0\right]^{\mathrm{T}} \times \bar{n} \right\|}. \quad (11)$$

Finally, the last k rows will be moved to the beginning of the matrix. This produces, since the image is cylindrical, a counter-clockwise rotation about the surface normal of $\rho_z \cdot k$ radians.

The resultant matrix of applying these three steps will be denoted by $A(i_o,j_o)$.

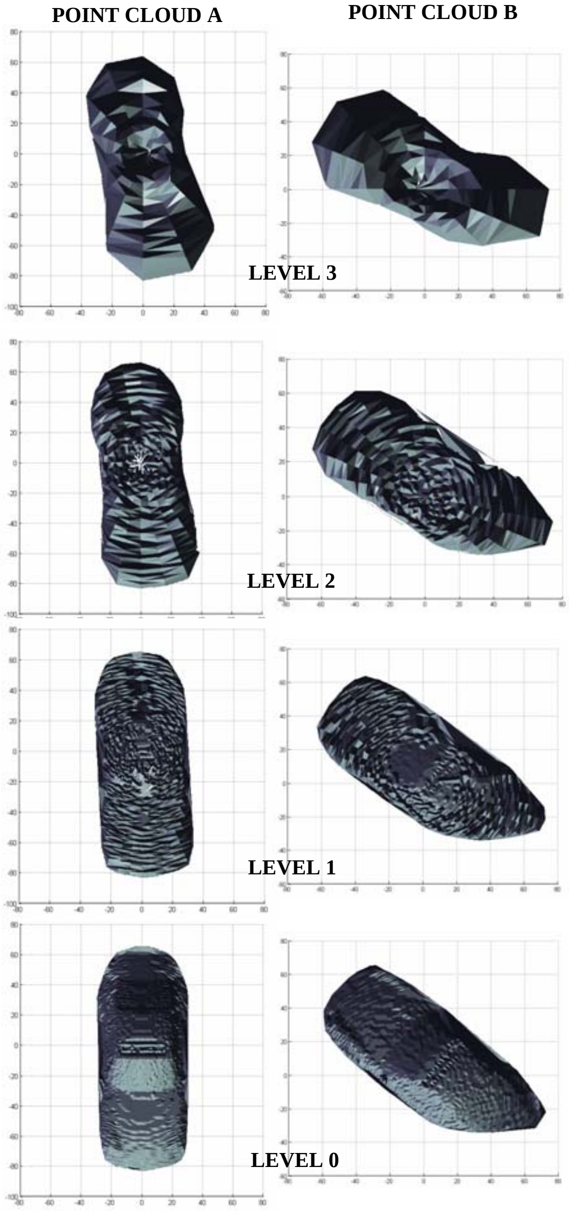


Fig. 10. Upper views of the 3D reconstructions corresponding to the images shown in Fig.9.

4.2 Similarity Measure

Before any calculation is made, the minimum values of both matrices ($A(i_o,j_o)$ and *B*) have to be equal.

Let the minimum of both matrices be

$$m = \min\left(A(i_o, j_o), B \right) \quad (12)$$

Hence, if a_{ij}=*min(A)* then a_{ij}=*m*. In the same way, if b_{ij}=*min(B)* then b_{ij}=*m*.

Then, the similarity measure is defined as

$$M_s = \frac{\sum_{i}^{n_s} \sum_{j}^{n_r} \left(a_{ij} - \bar{A} \right)\left(b_{ij} - \bar{B} \right) \cdot w_{ij}}{\sqrt{\left(\sum_{i}^{n_s} \sum_{j}^{n_r} \left(a_{ij} - \bar{A} \right)^2 \cdot w_{ij} \right)\left(\sum_{i}^{n_s} \sum_{j}^{n_r} \left(b_{ij} - \bar{B} \right)^2 \cdot w_{ij} \right)}} \quad (13)$$

where the mean values $\overline{A}$ and $\overline{B}$ are given by

$$\overline{A} = \frac{1}{n_s \cdot n_r} \sum_i^{n_s} \sum_j^{n_r} a_{ij} \tag{14}$$

and

$$\overline{B} = \frac{1}{n_s \cdot n_r} \sum_i^{n_s} \sum_j^{n_r} b_{ij} \tag{15}$$

This is a weighted correlation coefficient. The weights are assigned to be the column index of the pixel being evaluated in both images. Therefore, both a pixel a_{ij} and b_{ij} will have a weight $w_{ij}=j$. In this way, the pixels far from the central point are better weighted, since they represent a larger area of the object. However, if $a_{ij}=m$ and $b_{ij}=m$ then $w_{ij}=0$. Furthermore, if $a_{ij}=m$ and $b_{ij}\neq m$ (or vice versa), then

$$w_{ij} = j \cdot \left(1 - \frac{\sum_{(i,j)\in I} w_{ij}}{\sum_{(i,j)\in U} w_{ij}} \right) \tag{16}$$

where I and U are respectively the intersect set and the union set of indexes defined as

$$I = \left\{ (i,j) \middle| \left(a_{ij} \neq m\right) \text{and} \left(b_{ij} \neq m\right) \right\} \tag{17}$$

$$U = \left\{ (i,j) \middle| \left(a_{ij} \neq m\right) \text{or} \left(b_{ij} \neq m\right) \right\} \tag{18}$$

4.3 Removal of non-valid correspondences

Once all the object indexes have been evaluated for the lowest resolution level, there will be a similarity value M_s and a self-rotation parameter k for each possible correspondence. Therefore, these results can be displayed as images, as is shown in Fig. 11.

The indexes whose similarity value does not overcome a certain threshold τ_L will not be evaluated for the following resolution level.

Since approximations of the original images are used, this threshold will have a different value for each level L according to the expression:

$$\tau_L = (\eta_1)^L \cdot M_{S(\max)} \tag{19}$$

where η_1 is the percentage of $M_{S(max)}$ corresponding to the level 1 (in our experiments η_1=0.95).

However, when the resolution is low it is possible that some good correspondences are eliminated. In order to preserve their corresponding indexes a lower threshold τ_L' is applied to the neighbours of the pixels that have overcome the previous threshold. If their similarity value is bigger than this new threshold, these pixels will be also considered for the next resolution level.

The expression is the same as (19), but in this case the value of η_1' is lower (in our experiments, η_1'=0.8).

Finally, the indexes that have been chosen are converted to the new resolution level. Note that the number of rows and columns depends on the resolution level.

As can be seen in Fig. 11, the amount of indexes to be verified is reduced progressively. This permits to accelerate the search for correspondences.

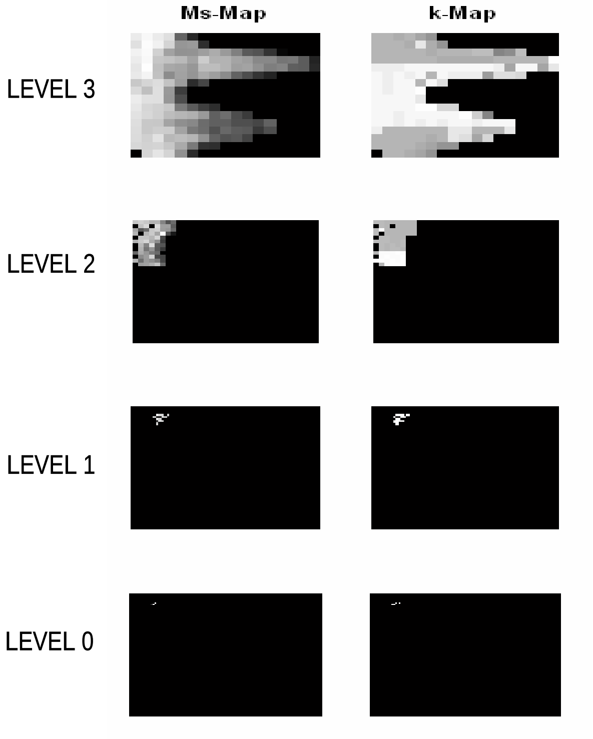


Fig. 11. Maps of similarity values and self-rotation indexes.

4.4 Verification of the best matches

Once the indexes corresponding to the last resolution level have been evaluated, the ten (at the most) transformed *circon* images with the highest similarity measures are chosen for verification. An example is shown in Fig. 12. The first image corresponds to the original *circon A*, whereas the ten next ones are its transformed *circon* images *A(i$_o$,j$_o$)* with the highest M_s. The last one corresponds to *circon B*, the image that they were compared with.

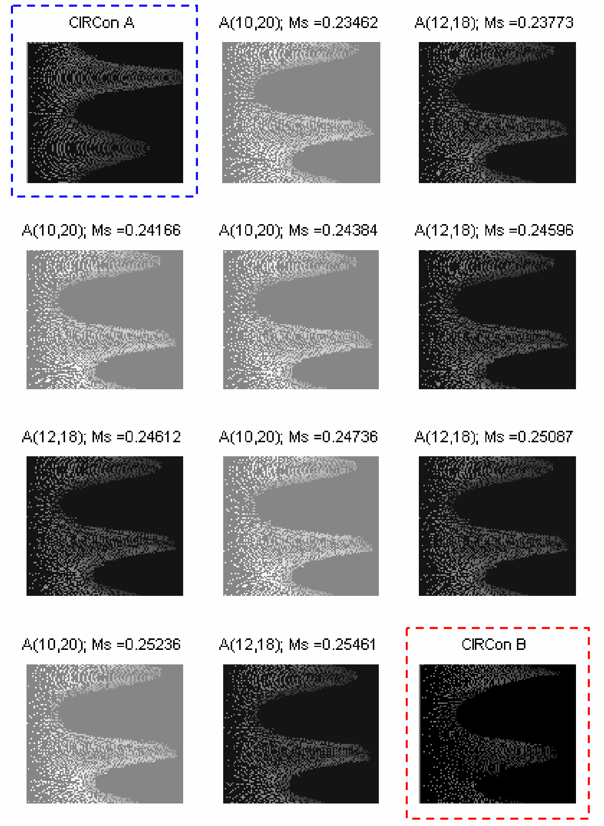


Fig. 12. The ten circon images with the highest similarity

The following expression is evaluated for each of these ten images:

$$M_d = \frac{tol}{\sum_i^{n_s}\sum_j^{n_r} w_{ij}} \left(\sum_i^{n_s}\sum_j^{n_r} \frac{w_{ij}}{\max\left\{\left|a_{ij} - b_{ij}\right| - d_w,\ tol\right\}} \right) \tag{20}$$

This allows to verify if the distance between both surfaces is sufficiently small and to determine which of the *circon* images provides the best alignment.

The values of M_d are comprised between 0 and 1. If both images were exactly equal, then M_d=1.

Note that the pairs of pixels whose distances are smaller than d_w+*tol* will have a weight equal to *1/tol*, whereas the weights for the rest will be inversely proportional to the distance between both pixels. Accordingly, this formula gives a higher importance to the pixels whose distance is very small. In the verification results shown in Table 1, d_w=1 and *tol*=0.1.

Table 1
VERIFICATION RESULTS

circon	*Ms*	*Md*
A(10,20,43)	0.2346	0.0027
A(12,18,43)	0.2377	0.0484
A(10,20,46)	0.2417	0.0027
A(10,20,47)	0.2438	0.0027
A(12,18,47)	0.2460	0.0573
A(12,18,44)	0.2461	0.0510
A(10,20,45)	0.2474	0.0027
A(12,18,45)	0.2509	0.0491
A(10,20,44)	0.2524	0.0027
A(12,18,46)	0.2546	0.0550

The *circon* with the highest M_d will be considered that produces the best alignment between the two surfaces S_A and S_B. In Table 1, A(12,18,47) has M_d=0.0573, i.e., the best alignment is produced when the matrix indexes are i_s=12, j_s=18 and the self-rotation index is k_s=47.

5 3D Registration

One interesting property of this 3D representation is that, once the best correspondence is determined, the 3D transformation between both surfaces depends on the indexes (i_s,j_s) and the number of rows k_s that were shifted to maximize the similarity value.

5.1 Euclidean Transformation

The two following expressions allow calculating the transformation (translation vector and rotation matrix) that aligns both point clouds S_A and S_B.

Translation vector:

$$t_C = \begin{pmatrix} \rho_r \cdot j_s \cdot \cos(-\rho_\theta \cdot (i_s - 1)) \\ \rho_r \cdot j_s \cdot \sin(-\rho_\theta \cdot (i_s - 1)) \\ \rho_z \cdot a_{ij} \end{pmatrix} \tag{21}$$

Rotation matrix:

$$R_C = R_Z(k_s) \cdot R_n \tag{22}$$

where R_n is the matrix obtained in (10) and,

$$R_Z(k_s) = \begin{pmatrix} \cos(\rho_\theta \cdot k_s) & \sin(\rho_\theta \cdot k_s) & 0 \\ -\sin(\rho_\theta \cdot k_s) & \cos(\rho_\theta \cdot k_s) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{23}$$

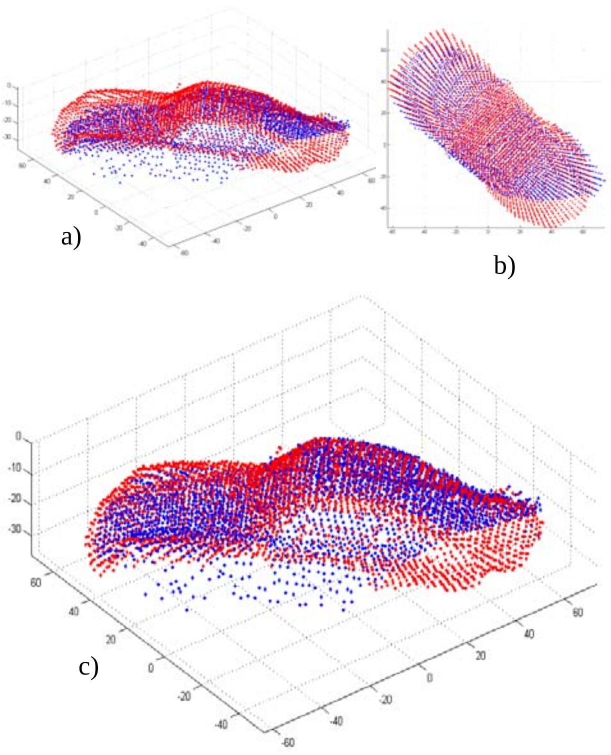


Fig. 13. Refinement of the Euclidean transformation.

5.2 Refinement of the Transformation

As a consequence of the rough calculation of the surface normals and also due to the wavelet decomposition of the *circon* images, after applying the Euclidean transformation calculated, the 3D point clouds are not sufficiently close from each other (see Fig. 13a). For that reason, it will be necessary to refine the Euclidean transformation obtained in the previous subsection.

However, the 3D point clouds are quite similar if seen from above (Fig. 13b), since the resultant images found by transformations have a high similarity value and hence, the percentage of overlap will be also high. Therefore there will be a sufficient number of correspondences to calculate the correction of the Euclidean transformation. The translation vector and the rotation matrix will be obtained by fitting a plane to the 3D point cloud whose z-coordinates are the differences between the z-coordinates corresponding to the overlapped pixels in the image (see Fig.14).

The plane equation will be calculated by Principal Component Analysis; thus, the normal vector and the origin distance *d* are obtained. Therefore, the correction of the translation vector Δ*t* will have the following expression:

$$\Delta t = \begin{bmatrix} 0 \\ 0 \\ d \end{bmatrix} \qquad (24)$$

And the correction of the rotation matrix Δ*R* will be calculated using (10) with the normal vector obtained.

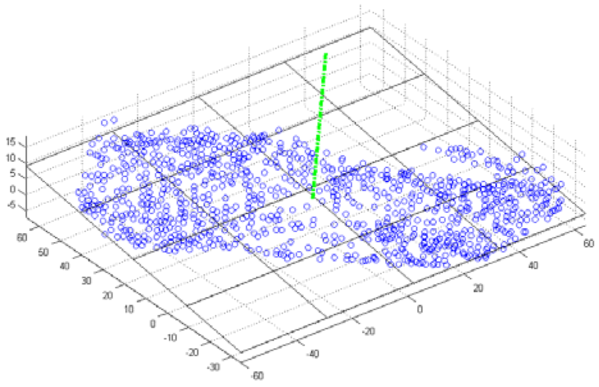


Fig. 14. Plane fitted for calculating the correction of the Euclidean transformation.

Consequently, the corrected translation vector is as follows:

$$t_C = t_C + R_C^{-1} \cdot \Delta t \qquad (25)$$

And the rotation matrix:

$$R_C = \Delta R \cdot R_C \qquad (26)$$

The results of these corrections are shown in Fig. 13c. However, all the 3D points shown in Fig.13 are expressed in the coordinate frame centered at the interest-points used to calculate the *circon* images (see Fig. 8). Therefore, it is necessary to consider these additional transformations in order to obtain the final alignment between the initial point clouds shown in Fig.7.

Let t_A and R_A be the translation vector and rotation matrix that allow expressing in the interest-point frame the coordinates of a point in S_A expressed in the original frame. In the same way, t_B and R_B permit a similar transformation for the point cloud S_B.

Consequently, the final Euclidean transformation *T* that aligns roughly both point clouds has the following expression:

$$T = \begin{bmatrix} R_B & -R_B \cdot t_B \\ 0_{1\times3} & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} R_C & -R_C \cdot t_C \\ 0_{1\times3} & 1 \end{bmatrix} \cdot \begin{bmatrix} R_A & -R_A \cdot t_A \\ 0_{1\times3} & 1 \end{bmatrix} \qquad (27)$$

As a result, given a point P_{Ak} belonging to the point cloud S_A, its new coordinates will be calculated as,

$$P'_{Ak} = T_{3\times4} \cdot P_{Ak} \tag{28}$$

where T_{3x4} is the matrix formed by the three first rows of T.

As is shown in Fig.15, the coarse transformation obtained is sufficiently good and it can be used as an initial estimate by the ICP algorithm [1],[2] if a finer alignment is required.

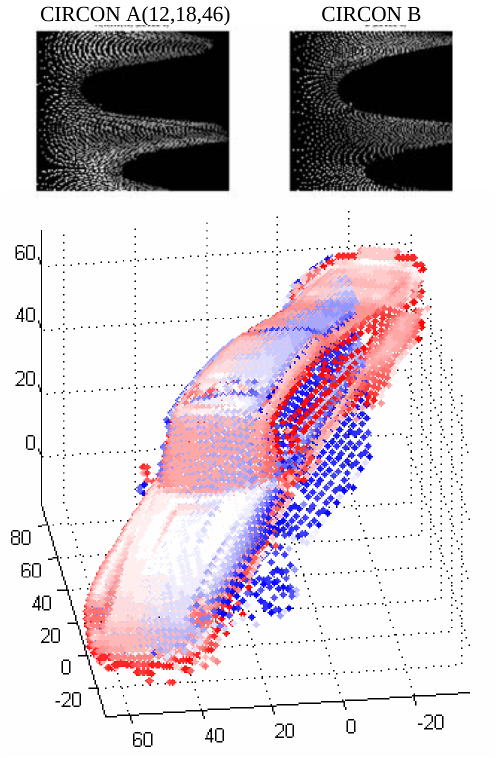


Fig. 15. Final alignment

6 Results

In addition to the final alignment shown in Fig.15, another results obtained for different objects are shown in Fig.16. The three first objects come from the Range Image Database of the Stuttgart University [12] whereas the last object was sensed by the 3D vision system presented in [8].

This vision system uses a variant of the well-known active method. As is shown in Fig.17, two CCD cameras are arranged fixed on opposite sides of the scene while a rotary platform turns the object. For each angle previously fixed, the laser projector illuminates an object *slice*. Subsequently, the two images acquired by the cameras are processed by a computer in order to make a 3D reconstruction of that slice (radial contour) just before the platform moves to the next position. At the end of the acquisition process a 3D point cloud is provided by this 3D vision system along with its corresponding *circon* image.

As is shown in Fig.16, the results obtained for all these objects can be considered acceptable since a coarse alignment was found; its corresponding Euclidean transformation can be used to initialize the ICP algorithm in order to refine this alignment.

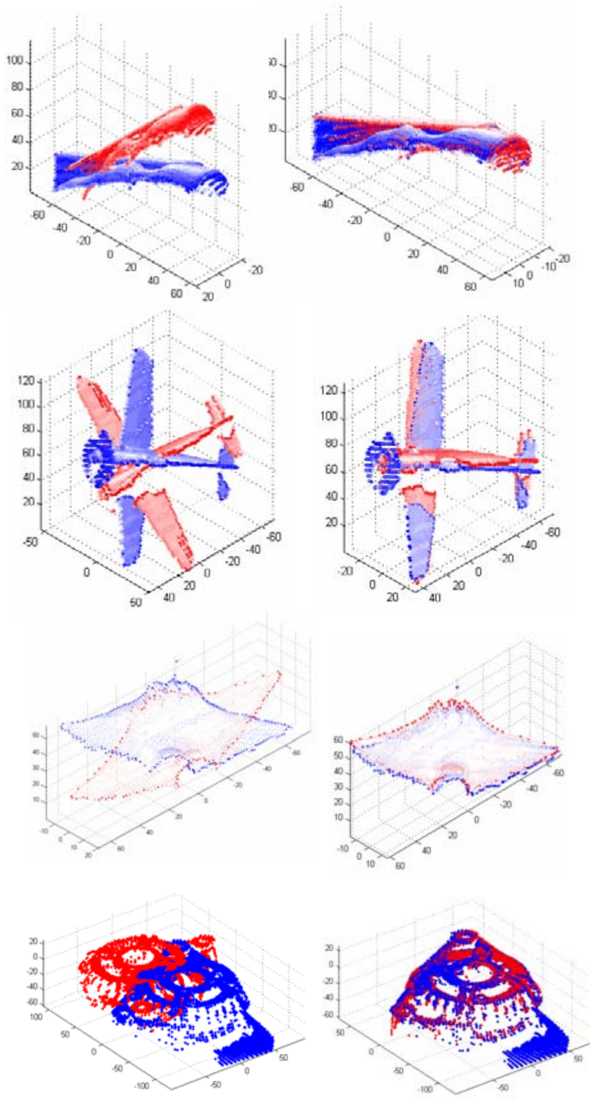


Fig. 16. Results obtained for different objects.

7 Conclusion

A coarse-to-fine algorithm for 3D registration has been presented in this paper. This algorithm makes use of a multilevel approach based on wavelet decomposition in order to search for correspondences between two point clouds belonging to the same object. This approach permits to accelerate the 3D surface matching process.

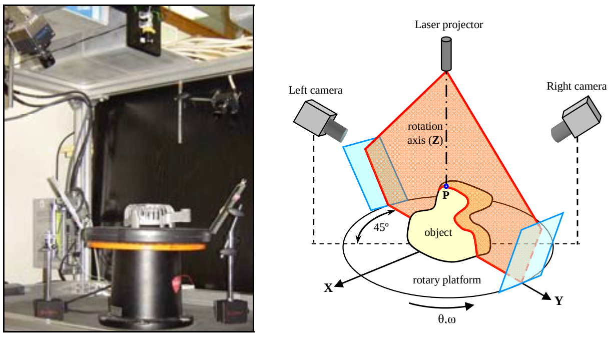


Fig. 17. 3D Vision System: real photo and scheme.

Although, as is shown in Fig. 16, the algorithm works well in different objects, some problems can be found when the surrounding of the point selected for building the *circon* image is too abrupt. This can have a significant influence on the calculation of the normal vector at that point and hence, on the search for correspondences.

At the present time we are working on making a better selection of the initial point in both point clouds. Besides, a new algorithm for 3D object recognition is being developed based on a similar approach.

8 Acknowledgement

This work has been carried out under the support of the CICYT Spanish project DPI2006-15313.

References:

[1] P.J. Besl, N.D. McKay, "A method of registration of 3-D shapes", *IEEE Trans. Pattern Anal. Mach. Intell.* 14 (2) (1992) 239–256.

[2] Y. Chen, G. Medioni, "Object modeling by registration of multiple range images", *Image Vision Comput.* 10 (3) (1992) 145–155.

[3] A. Johnson, M. Hebert, "Surface registration by matching oriented points", in: *IEEE Proc. Int. Conf. on Recent Advances in 3-D Digital Imaging and Modeling*, 1997, pp. 121-128.

[4] A. Ashbrook, R. Fisher, N. Werghi, C. Robertson, "Aligning arbitrary surfaces using pairwise geometric histograms", in: *Proc. Noblesse Workshop on Non-linear Model Based Image Analysis*, Glasgow, Scotland, 1998, pp. 103–108.

[5] S.M. Yamany, A.A. Farag, "Free-form surface registration using surface signatures", in: *Int. Conf. on Computer Vision*, vol. 2, 1999, pp. 1098–1104.

[6] D. Zhang, "Harmonic Shape Images: A 3D free-form surface representation and its applications in surface matching", Ph.D. thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, USA (November 1999).M. Young, *The Technical Writers Handbook.* Mill Valley, CA: University Science, 1989.

[7] B.M. Planitz, A.J. Maeder, J.A. Williams, "The Correspondence Framework for 3D Surface Matching Algorithms", *Computer Vision and Image Understanding*, Volume 97, Issue 3 , March 2005, pp 347-383.

[8] C. Torre-Ferrero, S. Robla, E.G. Sarabia, J.R. Llata, "CIRCON: An Altenative 3D Shape Representation for Surface Matching", *IECON'06, The 32nd Annual Conference of the IEEE Industrial Electronics Society*, Paris, France, 2006.

[9] C. Torre-Ferrero, S. Robla, E.G. Sarabia, J.R. Llata, "3D Registration by Using an Altenative 3D Shape Representation", *ISCGAV'07, The 7th WSEAS International Conference on Signal Processing, Computational Geometry and Artificial Vision*, Vouliagmeni, Greece, 2007.

[10] Mallat, S. (1989), "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Pattern Anal. and Machine Intell.*, vol. 11, no. 7, 1989, pp. 674-693.

[11] Meyer, Y. (1990), "Ondelettes et opérateurs", Tome 1, Hermann Ed. (English translation: Wavelets and operators, Cambridge Univ. Press. 1993.)

[12] K. Eisele, G. Hetzel, Range Image Database, 'http://range.informatik.uni-stuttgart.de/htdocs/html/', University of Stuttgart.