

RESEARCH

Open Access

# 3D point cloud registration based on a purpose-designed similarity measure

Carlos Torre-Ferrero\*, José R Llata, Luciano Alonso, Sandra Robla and Esther G Sarabia

## Abstract

This article introduces a novel approach for finding a rigid transformation that coarsely aligns two 3D point clouds. The algorithm performs an iterative comparison between 2D descriptors by using a purpose-designed similarity measure in order to find correspondences between two 3D point clouds sensed from different positions of a free-form object. The descriptors (named with the acronym CIRCON) represent an ordered set of radial contours that are extracted around an interest-point within the point cloud. The search for correspondences is done iteratively, following a cell distribution that allows the algorithm to converge toward a candidate point. Using a single correspondence an initial estimation of the Euclidean transformation is computed and later refined by means of a multiresolution approach. This coarse alignment algorithm can be used for 3D modeling and object manipulation tasks such as "Bin Picking" when free-form objects are partially occluded or present symmetries.

**Keywords:** laser scanner, 3D point cloud, descriptor, similarity measure, coarse alignment, 3D registration

## 1. Introduction

The alignment of two point clouds is quite a frequent task, both in 3D modeling and in object recognition. Similarly, the need for automating certain applications, such as computer-aided manufacturing or bin-picking, has necessitated the use of 3D information about the parts being manipulated. This information can be sensed by 3D acquisition methods [1], such as laser scanners or time-of-flight cameras, which provide a range image for every different pose of the object.

Finding the rigid transformation producing a suitable alignment of the resulting point clouds, without having a previous estimate, is a problem that has been approached using different strategies [2,3]. Although no solution has prevailed as the most accepted, algorithms based on intrinsic properties have been more widely applied due to their generality. These algorithms extract shape descriptors [4-15], curves [16,17], structures [18,19], or graphs [20] from both point clouds (sometimes meshes are used instead) in order to compare them. If several correspondences are found, then a

coarse transformation that aligns them in a suitable way can be calculated.

On the other hand, the algorithms that use extrinsic properties will be subject to one important restriction: as they match properties that are relative to a coordinate system, the surfaces must be roughly aligned in order to establish point correspondences. Therefore, these algorithms (such as the ICP algorithm [21,22] and its variants [23]) are used to refine that initial transformation and obtain a more precise one. Since this refinement process has been successfully achieved, the most challenging part of the 3D point cloud alignment problem is to determine the rough initial transformation.

## 2. CIRCON descriptor

### 2.1. Introduction

After reviewing the state-of-the-art, the following problems were found in the most significant alignment algorithms [2,3]: lack of generality (they work well with objects of a given topology), excessive computation time, problems with symmetries, poor behavior when point clouds have low density and when they overlap each other in a small region, need for a method (usually based on robust statistical techniques) that discards false correspondences and obtains a valid correspondence group to obtain the Euclidean transformation. All the

\* Correspondence: carlos@teisa.unican.es  
Electronics Technology, Systems and Automation Engineering Department,  
University of Cantabria, Av. Los Castros s/n 39005 Santander (Cantabria),  
Spain

methods reviewed show, to a greater or lesser extent, at least two of these drawbacks.

The three characteristics to which we have given most importance when designing our alignment algorithm are it must have no restrictions regarding the type of objects that can be used, a good performance in the presence of symmetries (which is quite common in industrial components) and good behavior when the overlap and the density of the point clouds are low. However, we have also taken into account the other problems that may occur.

One of the main drawbacks observed after analysis of the commonly used descriptors in the state-of-the-art is that although many of them are based on geometric properties of the environment of the point-of-interest, the evaluation of their similarity does not have a direct relationship with the distance between the point clouds [4,8-15]. Moreover, since a good alignment is characterized by a small distance between corresponding points, it would be more convenient to use a descriptor that represents the geometry of the environment better. Furthermore, the descriptors analyzed need to find at least three good correspondences to determine an approximate Euclidean transformation.

Another drawback associated with some descriptors is that at the end of the local matching stage, a considerable percentage of false matches can be found. This is usually caused by a descriptor with low discriminating capacity and a choice of similarity measure that is not sufficiently appropriate.

In our opinion, for the correspondence search to be effective, the descriptor used by the coarse alignment algorithm should:

- Be based on the geometry in the environment of the points-of-interest.
- Be highly descriptive, so that the correspondences can be adequately discriminated and no false matches appear.
- Enable the use of a similarity measure based on distances between points of the cloud.
- Enable the calculation of a Euclidean transformation based on a single correspondence.
- Be useful for 3D modeling (alignment of two scans from two different views of the object) and for 3D object recognition (alignment of the point clouds in the scene and the model).

## 2.2. Descriptor construction

In order to obtain the descriptor associated with a particular point-of-interest in the cloud (let  ${}^w p_q$  be this point), it is necessary to express the cloud points in a local coordinate system centered on  ${}^w p_q$  and whose  $Z_q$ -axis is its

normal vector. The  $X_q$ -axis is chosen so that it is perpendicular to both the  $Y_w$ -axis of the reference coordinate system and the normal vector at the point-of-interest. Thus the  $Y_q$ -axis is determined by the cross product of unit vectors along the  $X_q$  and  $Z_q$  axes. This criterion establishes a unique reference for the angles of rotation about the  $Z_q$ -axis (i.e., above normal  $\bar{n}_q$ ), which will subsequently facilitate the calculation of the Euclidean transformation associated with that correspondence.

Once the cloud points are transformed to the local frame, the environment of the point-of-interest is considered to be divided into  $n_s$  sectors (whose angle is  $\rho_\theta$  radians), which are further divided radially into cells with length  $\rho_r$  mm (excluding the cell closest to the centre, "cell 0", which will be a sector with a radius  $0.5 \rho_r$  mm). The sectors are numbered clockwise starting with the sector that is centred on the  $X_q$ -axis ( $\theta_1 = 0$ ). Figure 1 shows, around the  $Z_q$ -axis, this sense of numbering and the nature of the cells for the  $i$ th sector.

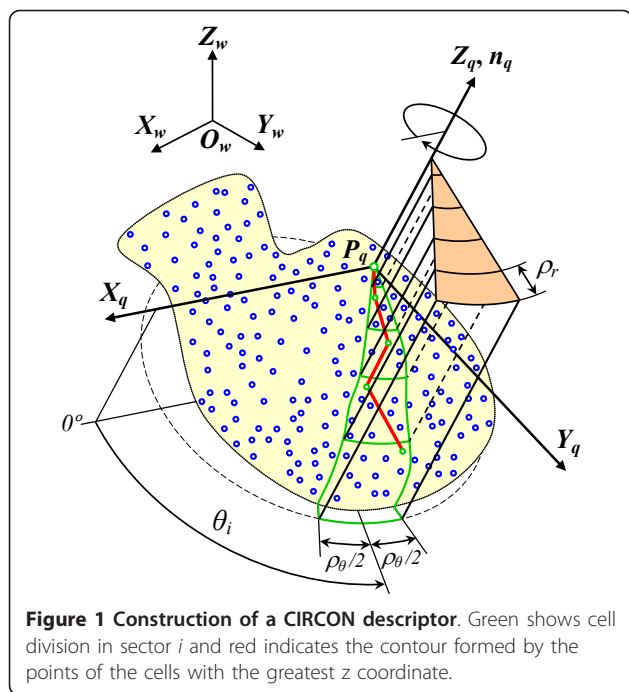
Taking into account this division of the point cloud into sectors and cells, a transformation based on cylindrical coordinates is applied in order to obtain, for any point  $p_d$  with coordinates  $(x_d, y_d, z_d)$  in the coordinate system with origin at  $p_q$ , the  $i$  index corresponding to the number of sector to which it belongs, the  $j$  index indicating the cell within that sector, and the height value associated with its coordinate  $z_d$ .

$$i = \left( \left\lfloor n_s - \frac{\tan^{-1} \left( \frac{y_d}{x_d} \right)}{\rho_\theta} \right\rfloor \bmod n_s \right) + 1 \quad (1)$$

$$j = \left\lfloor \frac{\sqrt{x_d^2 + y_d^2}}{\rho_r} \right\rfloor \quad (2)$$

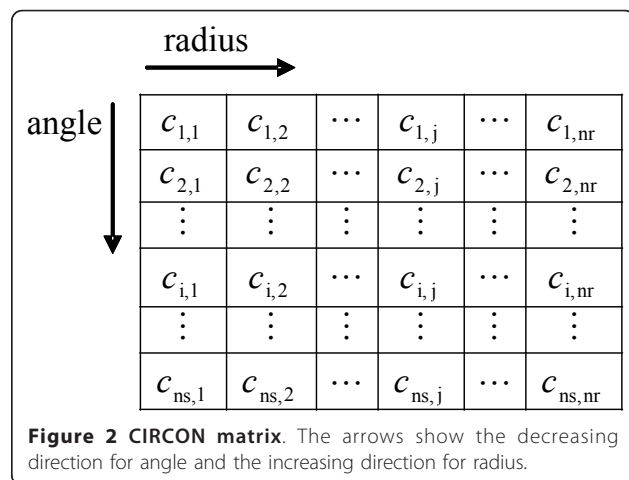
$$c_{i,j} = \left\lfloor \frac{z_d}{\rho_z} \right\rfloor \quad (3)$$

where  $\rho_\theta$  is the angular resolution,  $\rho_r$  the radial resolution, and  $\rho_z$  the height resolution. Note also that  $\lfloor x \rfloor$  is the nearest integer to  $x$ . Applying these three equations to all points of the cloud and maintaining the maximum values  $c_{i,j}$  for each cell (which is equivalent to saying that its  $z$  coordinate is maximum), we obtain a set of triples  $(i, j, c_{i,j})$  that uniquely correspond to each of the cells into which the environment of the point-of-interest has been divided. Therefore, the descriptor consists of a set of coded values that represent the contours described by the points with the maximum  $z$  coordinate of each of the cells into which each sector is divided. Figure 1 shows, for sector  $i$ , the contour described by points with the largest 'z' in the cells. As each pair of



indexes  $(i, j)$  has a single value of  $c_{i, j}$ , we can build a matrix  $C$  whose row index is the sector number,  $i$ , and whose column index is the number of cell,  $j$ , within that sector (see Figure 2). Note that the cells closest to the center (the origin of the local coordinate system) are not represented in this matrix since their value would always be close to zero.

Since the sequence should be closed because it describes the environment of the point-of-interest, the first and last rows must be considered adjacent, since their elements with the same column index correspond to adjacent cells. In other words, this descriptor has the property of being cyclical.



This matrix can also be viewed as an image (see Figure 3) whose pixels represent the values  $c_{i, j}$ , so that each of them has an associated colour (or a gray level). Hence, this descriptor can be considered as a cyclical image of the environment of a point-of-interest. Since each row represents a radial encoding of the contour described by the maximum heights of each of the cells in a sector, we will call this descriptor “CIRCON”, which is an acronym of “Cyclical Image of Radial CONtours”.

### 3. The proposed similarity measure

#### 3.1. Introduction

Unlike the similarity measures based on correlation coefficient (CC), mutual information (MI) [24,25], joint entropy [26,27], or others [28,29] that have been used by the most popular coarse alignment algorithms, such as spin images, this similarity measure is based on distance between pixels and takes into account the problems of occlusion that can appear in real situations that need 3D registration or object recognition.

This similarity measure gives weighting to both the overlap and the proximity of two point clouds. This enables the simultaneous evaluation of the geometric consistency of the correspondences. Although computational cost increases with the number of correspondences evaluated, the Euclidean transformation associated with this correspondence can be directly calculated, given that the rotation around the normal necessary to align the two point clouds is determined. Therefore, it will be possible to determine which correspondences give rise to the Euclidean transformation which fits best and to base the stopping criterion on their validation. In this way, the algorithm finalizes when the coarse transformation satisfies the end conditions imposed (detailed in the algorithm in Section 5.3), without necessity to evaluate all the points-of-interest selected in the two point clouds.

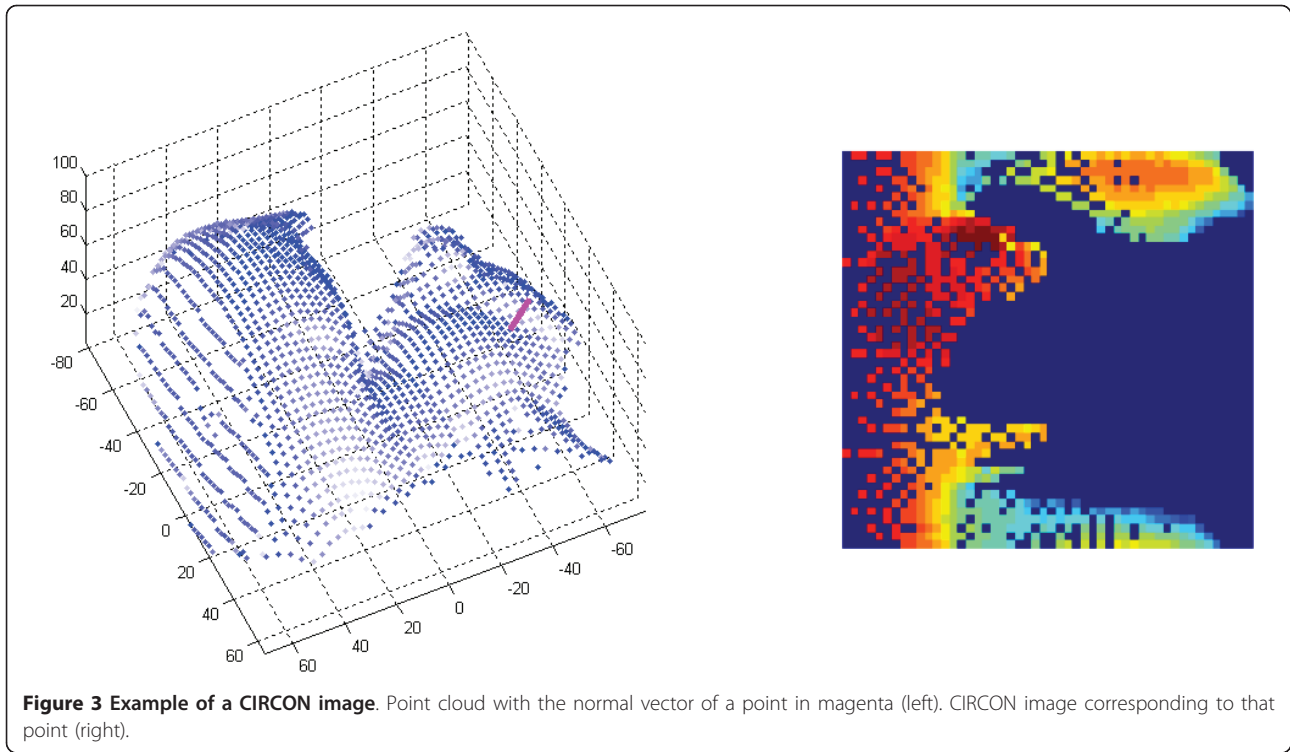
#### 3.2. Sets of pixels

Assuming two CIRCON descriptors  $A$  and  $B$  corresponding to two matched points; if both a pixel from  $A$ ,  $a_{ij}$ , and a pixel from  $B$ ,  $b_{ij}$ , represent a part of the point cloud, it will be considered that this pair of pixels, with indexes  $(i, j)$ , are overlapped.

Taking into account that the matrix elements not belonging to the point cloud will be considered computationally as ‘not-a-number’ (i.e.,  $NaN$ ), the following sets of pixels have been defined:

Intersection Set (overlapped pixels):

$$I_{AB} = \{(i, j) | (a_{ij} \neq NaN) \text{ and } (b_{ij} \neq NaN)\} \quad (4)$$



Exclusive-OR Set (non-overlapped pixels):

$$X_{AB} = \{(i, j) | (a_{ij} \neq NaN) \text{ xor } (b_{ij} \neq NaN)\} \quad (5)$$

Union Set (overlapped pixels and non-overlapped pixels):

$$U_{AB} = \{(i, j) | (a_{ij} \neq NaN) \text{ or } (b_{ij} \neq NaN)\} \quad (6)$$

These sets of pixels will be taken into account in order to define the similarity measure.

### 3.3. Area represented by each cell

The top-view area represented by each cell (see Figure 4) increases with the distance from the central point (the point-of-interest where the descriptor has been generated). Therefore, the size of this area will determine which points in the cloud correspond to a cell of the image matrix.

Accordingly, the pixels corresponding to a specific column  $j$  in the CIRCON image represent the points that have a radius between  $\rho_r(j-0.5)$  and  $\rho_r(j+0.5)$ . Therefore, the theoretical area corresponding to each one of these pixels will be the same,  $A_{pj}$ , and it will be given by the following expression:

$$A_{pj} = \frac{\pi}{n_s} \cdot ((\rho_r \cdot (j + 0.5))^2 - (\rho_r \cdot (j - 0.5))^2) \quad (7)$$

$n_s$  being the number of angular divisions.

Simplifying,

$$A_{pj} = \frac{2\pi}{n_s} \cdot \rho_r^2 \cdot j \quad (8)$$

### 3.4 Weight of the pixels

In the same way, the theoretical area corresponding to a pixel of the first column (equivalent to a cell of the first ring of the Figure 4) will be

$$A_{p1} = \frac{2\pi}{n_s} \cdot \rho_r^2 \quad (9)$$

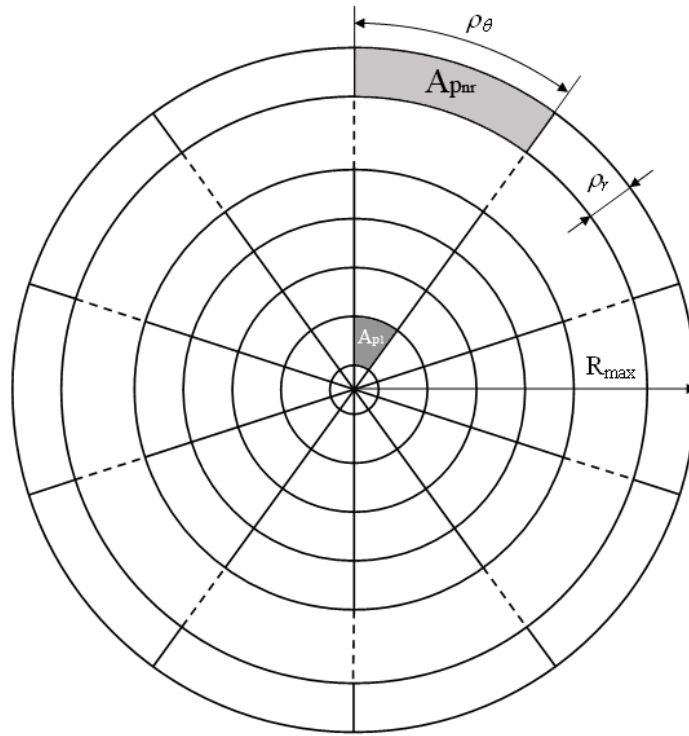
Therefore, the relationship of areas between a pixel in the  $j$ th column and the one in the first column will be as follows

$$\frac{A_{pj}}{A_{p1}} = j \quad (10)$$

Using this expression, the area represented by the pixels in different columns will be taken into account to correctly weight the contribution of each pixel to the average distance in the overlapped area.

### 3.5. Similarity measure expression

As was explained previously, the similarity measure selected will depend on the distance and the overlap among the CIRCON images.



**Figure 4** Area of the cells into which the environment of a point-of-interest is divided. These cells are represented by the pixels in the CIRCON image.

To calculate the average distance between the pixels of the two images, those of the overlapped area and those of the non-overlapped area will be considered separately.

Therefore, the average distance in the overlapped area will be defined using the following expression:

$$D_{ov} = \frac{\sum_{(i,j) \in I_{AB}} A_{pj} \cdot |a_{ij} - b_{ij}|}{\sum_{(i,j) \in I_{AB}} A_{pj}} = \frac{\sum_{(i,j) \in I_{AB}} j \cdot |a_{ij} - b_{ij}|}{\sum_{(i,j) \in I_{AB}} j} \quad (11)$$

And the overlap ratio is defined as

$$\sigma_{ov} = \frac{\sum_{(i,j) \in I_{AB}} j}{\sum_{(i,j) \in U_{AB}} j} \quad (12)$$

which expresses the relationship between the weighted number of overlapping pixels and the weighted total number of pixels pertaining to the object (overlapping and non-overlapping).

The similarity measure proposed considers both terms and it provides values between 0 and 1,

$$M_S = \frac{\sigma_{ov}}{(\rho \cdot D_{ov} + \lambda') + \sigma_{ov} \cdot (1 - \lambda')} \quad (13)$$

where  $\lambda'$  is defined as  $\lambda' = \rho \lambda$ ,  $\lambda$  being a parameter whose value represents the additional distance with which non-overlapping pixels are penalized. In contrast the parameter  $\rho$  modifies the relationship between the expected similarity value and the distance  $D_{ov}$  that produces it, when the overlap is 100%.

The values for these parameters used in our experiments are  $\rho = 1$  and  $\lambda = 1$ , which give a similarity value of 0.5 both when  $\sigma_{ov} = 0.5$  and  $D_{ov} = 0$  and when  $D_{ov} = 1$  and  $\sigma_{ov} = 1$ .

Since, given a matching pair, the point coordinates and their corresponding normal vectors are known for both points, only a free parameter is needed to compute the rigid transformation: the rotation around the normal. This can be easily calculated using CIRCON images since these are cyclical. By shifting the last row to the top for the first CIRCON image (from point cloud 1) and leaving the second one fixed (from point cloud 2), the similarity measure for a rotation  $\rho_\theta$  can be calculated. If the last two rows are shifted to the top, the equivalent rotation will be  $2\rho_\theta$ , and so on. This can be practically implemented by means of matrix blocks so that the similarity measures for all the shifts can be computed at the same time. Subsequently, the similarity value for a matching pair will be the maximum for all the possible rotations and it will be associated with an

angle  $k \cdot \rho_\theta$  ( $k$  being the number of row shifts). A preliminary analysis of this similarity measure can be found in [30].

#### 4. Coarse alignment algorithm

##### 4.1. Point-of-interest selection

First, it will be necessary to select a set of points-of-interest in both clouds,  $\{p_{1i}\}$  and  $\{p_{2i}\}$ . An important novelty is that the alignment algorithm does not need a thorough selection of the points-of-interest or a large number of them to obtain a proper alignment. Instead we have designed a simple algorithm that selects the points-of-interest by taking into account the particular characteristics of the descriptor proposed. First, normal vectors at each point are calculated by interpolation using the range image data. Applying basic morphological operations on the resulting image and the Laplace operator on the normal components (see Figure 5), points are extracted whose normal vectors are stable and also close to areas where the normal varies abruptly. This enables points-of-interest with a stable CIRCON descriptor to be selected that, in addition, represent close areas with relevant features.

The number of points extracted depends on the topology of the object, although a minimum distance between them is considered so that this number is not very high.

##### 4.2. Correspondence search algorithm

This search algorithm is the core of the coarse alignment algorithm. It is based on an iterative search for the greatest value of similarity measure using the array of cells,  $C_1$ , into which the environment of a point-of-interest in point cloud 1 is divided. The chosen stopping criterion ensures that this search is convergent, since the

environment where the correspondences are searched for is progressively reduced.

The algorithm will evaluate correspondences between different points of cloud 1 and a point-of-interest selected from cloud 2. The degree of validity of two matching points is to be determined based on the similarity between their CIRCON images:  $I_{1x}$  (image of a point  $P_{1x}$  in cloud 1) and  $I_2$  (target image). Since a single point,  $P_{1x}$ , is extracted from each cell of the array  $C_1$ , for each iteration, the algorithm performs as many similarity measure evaluations as the number of 'valid' cells in the distribution  $C_1$  around the point of the previous iteration. Note that the points whose CIRCON images obtain a low similarity value are stored in a list of non-valid indexes,  $ind_{nv}$ . Thus, a cell will be considered 'valid' when its ratio of non-valid points,  $r_{nv}$ , is less than a prefixed threshold  $\tau_{nv}$ . This allows to progressively reduce the number of cells to be checked.

Therefore, the similarity value returned by the algorithm,  $M_{Sc}$ , is the highest obtained in all the iterations until the stopping condition is met. The algorithm ends when an iteration uses a starting point whose distance to one of the previously used points is less than a preset  $\delta$  (in our implementation  $\delta = \rho_r/16$ ).

As will be explained in Section 5.3, the size of the descriptors used by this search algorithm depends on the resolution level in the main algorithm. Moreover, since one of its goals must be to avoid an incorrect alignment in the presence of occlusions and symmetries of the objects, the CIRCON images will represent the entire point clouds in order to increase their descriptiveness. However, depending on the application (e.g., mixed objects), the environment size of the descriptor can be varied.

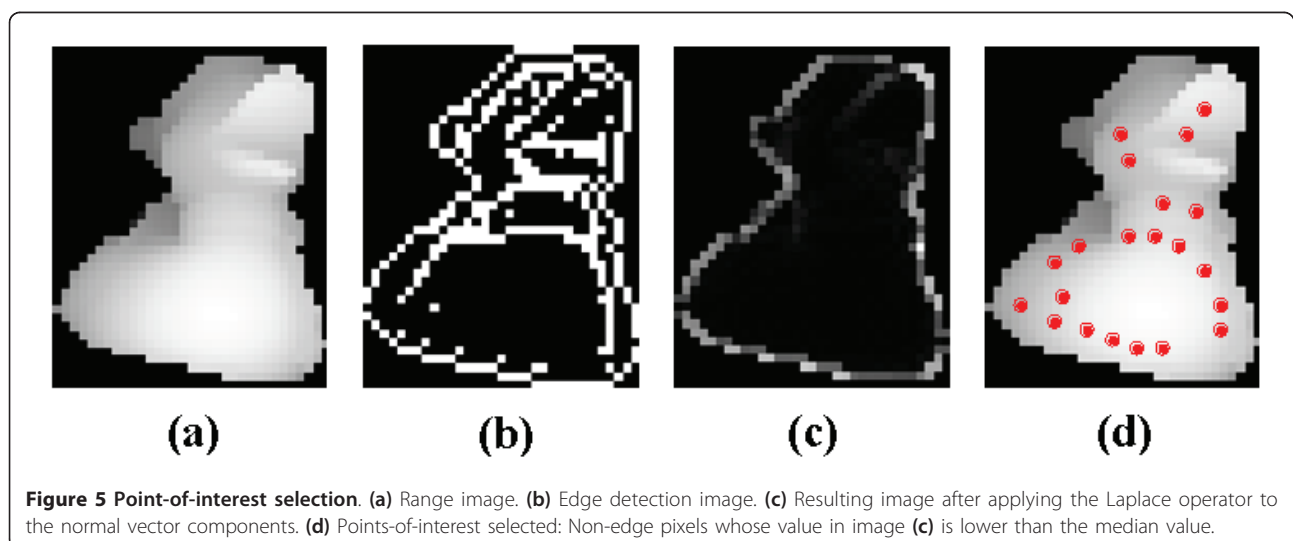


Figure 6 shows the steps that must be followed in order to perform this iterative search for correspondences.

#### 4.3. Main algorithm: selection of the most suitable transformation

Figure 7 shows all the steps that must be followed in order to obtain a coarse alignment between two point clouds. This algorithm uses the *Correspondence Search Algorithm* introduced in Section 5.2 in a multiresolution approach which refines the Euclidean transformation matrix.

For each interest-point chosen in cloud 2,  $P_{2y}$ , the search for correspondences by cells is established for  $n_v$  levels of resolution. The number of levels and the lowest

resolution must be determined through a compromise between computation time and accuracy of the point cloud alignment, which will depend on the application.

The *starting points* for the first level are the interest-points chosen in cloud 1,  $\{p_{1i}\}$ . This first level enables the discarding of those zones (cells) of the surroundings of the chosen point,  $P_{1x}$ , where, due to their low similarity, it is unlikely to find the desired correspondence. Once the *Correspondence Search Algorithm* has found an approximate correspondence,  $P_{1c}$ , for the first level, the resolution is increased and a new search around this new *starting point* is performed but with smaller cells. In this way, the search zone is reduced, which will be the object of progressive refinement for the next

---

#### Algorithm 1 Correspondence Search.

---

1. Initialization of variables and parameters.
  2. **while** the stopping criterion is not satisfied:
  3. The cells of  $C_1$  that contain some point are obtained.
  4. **for** each cell of  $C_1$ :
  5. **if**  $r_{nv} < \tau_{nv}$ 
    - (a) The point  $P_{1x}$  with normal  $n_{1x}$  closest to the center of the cell is chosen.
    - (b) Centered at the point  $P_{1x}$ , the *circon* image ( $I_{1x}$ ) and  $C_{1x}$  are obtained.
    - (c) The similarity value ( $M_{Sx}$ ) is calculated between  $I_{1x}$  and  $I_2$  (*target image*). The rotation index ( $k_{1x}$ ) around the normal  $n_{1x}$  is also obtained.
    - (d) **if**  $M_{Sx} > M_{Sit}$ , the information associated with that correspondence is saved:  $M_{Sit} = M_{Sx}$  ;  $I_{1it} = I_{1x}$  ;  $C_{1it} = C_{1x}$  ;  $k_{1it} = k_{1x}$  ; **end if**
  6. **end if**
  7. **end for**
  8.  $I_1 = I_{1it}$  and  $C_1 = C_{1it}$  (data are updated with the results of the iteration).
  9. The indexes of the points evaluated with low similarity are added to the list  $ind_{nv}$  which is necessary to obtain  $r_{nv}$ .
  10. The minimum distance,  $D_{min}$ , from the point  $P_{1it}$  to the points stored in previous iterations is calculated.
  11. The stopping criterion is evaluated:
    - (a) **if**  $D_{min} \leq \delta$ , the search is ended.
    - (b) **else**,  $P_{1it}$ ,  $M_{Sit}$  and  $k_{1it}$  are stored.
    - (c) **end if**
  12. **end while**
  13. The maximum similarity value that was stored ( $M_{Sc}$ ), its associated index ( $k_{1c}$ ), its corresponding point ( $P_{1c}$ ) and array of cells ( $C_{1c}$ ) are returned.
- 

Figure 6 Correspondence search algorithm.

---

**Algorithm 2** Coarse Alignment of two point clouds.

---

1. Points of interest in cloud 1  $\{p_{1i}\}$  and cloud 2  $\{p_{2i}\}$  are selected.
  2. **while** all the *target points*  $\{p_{2i}\}$  have not been checked:
  3. **while** all the *starting points*  $\{p_{1i}\}$  have not been checked:
  4. **for** the  $n_v$  resolution levels of the *circon* images:
  5. *Correspondence Search Algorithm* is applied to search in cloud 1 (starting with  $P_{1x}$ ) for the correspondence of a point of interest  $P_{2y}$ . The output data consist of the corresponding point  $P_{1c}$ , the array of cells  $C_{1c}$ , the rotation index around the normal,  $k_{1c}$ , and the similarity measure,  $M_{Sc}$ .
  6. **if**  $M_{Sc} < \tau_{Ms}(l)$ , Go to step 12; **else**  $\tau_{Ms}(l) = M_{Sc}$ , **end if**;
  7. **if** level  $l$  is the last one:
    - (a) The Euclidean transformation  $T_c$  associated with the correspondence is calculated.
    - (b) Two cells of  $C_{1c}$  that meet certain conditions of angle between normals and distance with respect to two other cells in cloud 2 are selected.
    - (c) Two correspondences between those cells are established:  $m \leftrightarrow n$ ,  $r \leftrightarrow s$
    - (d) Using the points  $P_{1c}$ ,  $P_{1m}$ ,  $P_{1r}$  from cloud 1 and the points  $P_{2y}$ ,  $P_{2n}$ ,  $P_{2s}$  from cloud 2, a fictitious correspondence  $t_1 \leftrightarrow t_2$  is created and its associated Euclidean transformation is calculated:  $T_f$ .
    - (e) The distances  $d_R$  and  $d_t$  between the transformations  $T_c$  and  $T_f$  are calculated.
    - (f) **if**  $(d_R < \tau_R)$  **and**  $(d_t < \tau_t)$ , The algorithm is ended. **end if**
  8. **end if**
  9.  $P_{1x} = P_{1c}$  (the first iteration for the next resolution level starts with the point that was found in the previous resolution level).
  10. The next resolution level is prepared:  $n_s = 2 \cdot n_s$ ;  $n_c = n_c/2$ ;  $k_{1c} = 2 \cdot k_{1c}$ .
  11. **end for**
  12.  $P_{1x}$ =Next point of  $\{p_{1i}\}$  (if this point has not been evaluated yet)
  13. **end while**
  14.  $P_{2y}$ =Next point of  $\{p_{2i}\}$
  15. **end while**
- 

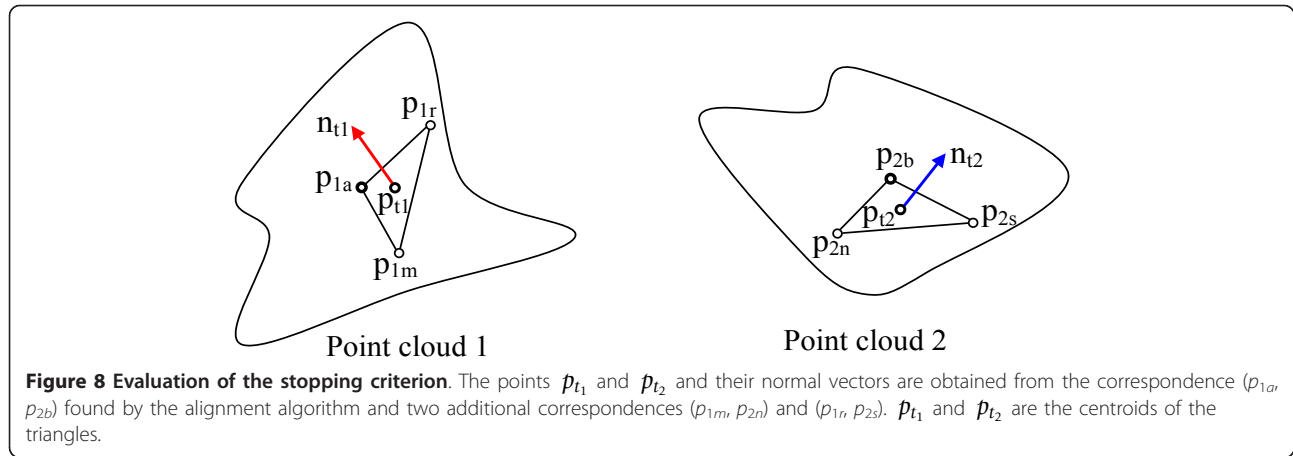
**Figure 7** Coarse alignment algorithm.

resolution levels (since only the first  $n_c$  columns of the array of cells  $C_{1c}$  are used for the correspondence search and this number is halved when the resolution is increased). When the convergence of the search associated with the last resolution level is achieved and the similarity value  $M_{Sc}$  of the resulting correspondence is greater than a value  $\tau_{Ms}(n_v)$ , its corresponding Euclidean transformation,  $T_c$ , is calculated using Equation (22).

We propose a stopping criterion for the alignment algorithm that also takes into account the characteristics

of the descriptor. It uses the correspondence found and two additional ones,  $(P_{1m}, P_{2n})$  and  $(P_{1r}, P_{2s})$ , to create a fictitious correspondence (see Figure 8) with which another transformation matrix,  $T_f$  is computed. Those additional points are chosen from corresponding cells in which both point clouds are distributed. The condition that must be satisfied for each of the two selected points in point cloud 1 is that the angle formed by their normal vectors and the normal at the point being evaluated should be very similar to the angle formed by their corresponding normal vectors in point cloud 2.





Then the matrix associated with the fictitious correspondence,  $T_f$ , is compared with that obtained by the algorithm,  $T_c$ . The execution is stopped if a distance measure for the rotation,  $d_R$ , and another for the translation,  $d_t$ , do not exceed their respective thresholds  $\tau_R$  and  $\tau_t$ .

Thus, the rotation distance  $d_R$  is obtained through the following expression:

$$d_R = \sqrt{\frac{1}{3}(\alpha_R^2 + \beta_R^2 + \gamma_R^2)} \quad (14)$$

where  $(\alpha_R, \beta_R, \gamma_R)$  are the ZYX Euler angles from the rotation matrix  $R_f^T \cdot R_c$ ;  $R_c$  being the rotation matrix obtained by the algorithm, and  $R_f$  the rotation matrix associated with the fictitious correspondence.

The translation distance  $d_t$  will be calculated as the RMS distance between the translation vectors,  $t_c$  and  $t_f$ , associated with both correspondences.

As will be shown in the results, the solution obtained by the algorithm can be sufficiently accurate for object manipulation tasks; however, the Euclidean transformation could be refined using the ICP algorithm by taking advantage of the data provided by our algorithm about the correspondences between the points in the two clouds.

#### 4.4. Calculation of the Euclidean transformation using a single correspondence

Once the correspondence with the highest similarity measure within the surroundings of the point-of-interest chosen for an iteration of the algorithm is found, the Euclidean transformation that coarsely aligns the point clouds can be calculated.

In the first place, it is necessary to express both point clouds within a frame of coordinates whose origin is each of the two points that have been matched and whose  $z$ -axis is aligned with the normal vectors at these points.

By convention, the  $x$ -axis of the new frame is perpendicular to the normal (new  $z$ -axis) and to the  $y$ -axis of the original frame  $W$ , since the CIRCON images were generated in this way.

Therefore, given an interest-point  ${}^W P_q$  with normal vector  $\bar{n}_q$  expressed in the frame  $W$ , the data will be rotated by using the following matrix:

$${}^q_W R = [\bar{x}_q \bar{n}_q \times \bar{x}_q \bar{n}_q]^T \quad (15)$$

where

$$\bar{x}_q = \frac{\hat{y}_W \times \bar{n}_q}{\|\hat{y}_W \times \bar{n}_q\|} \quad (16)$$

The translation vector is given by the coordinates of the point in the original frame. Therefore, the total Euclidean transformation will be given by the following expression:

$${}^q_W T = \begin{bmatrix} {}^q_W R & -{}^q_W R \cdot {}^W P_q \\ 0_{1 \times 3} & 1 \end{bmatrix}^{-1} \quad (17)$$

Given that the transformation matrix is  $4 \times 4$ , all the points in the following equations are expressed in homogeneous coordinates.

Suppose a correspondence between a point  ${}^{W1} P_\alpha$  of cloud 1 and a point  ${}^{W2} P_\beta$  of cloud 2. Let  ${}^\alpha_{W1} T$  be the transformation matrix that enables the expression within the interest-point frame of the coordinates of a point in the cloud 1 expressed in the original frame  $W_1$ . In the same way,  ${}^\beta_{W2} T$  permits a similar transformation for point cloud 2.

A point  ${}^{W1} P_i$  is expressed within the frame of point  ${}^{W1} P_\alpha$  in the following way:

$${}^\alpha P_i = {}^\alpha_{W1} T \cdot {}^{W1} P_i \quad (18)$$

In the same way, a point  ${}^{W_2}P_j$  in the frame centered on  ${}^{W_2}P_\beta$  will be:

$${}^\beta P_j = {}^\beta_{W_2} T \cdot {}^{W_2} P_j \quad (19)$$

As the points  ${}^{W_1}P_\alpha$  and  ${}^{W_2}P_\beta$  form a correspondence, the origins of coordinates and the  $z$ -axes of the new frames must be coincident. To align a point  ${}^\alpha P_i$  in point cloud 1 with its corresponding point  ${}^\beta P_j$  in cloud 2 it is necessary to rotate cloud 1 about the  $z$ -axis of the new frame by an angle of  $k \cdot \rho_\theta$  radians, where  $k$  is the number of rows the CIRCON image 1 was rotated to achieve the similarity value associated with this correspondence.

Therefore, to coarsely align the two points, the following must be fulfilled:

$${}^\beta P_j \approx R_z(k) \cdot {}^\alpha P_i \quad (20)$$

Substituting (18) and (19) into (20)

$${}^\beta_{W_2} T \cdot {}^{W_2} P_j \approx R_z(k) \cdot {}^\alpha_{W_1} T \cdot {}^{W_1} P_i \quad (21)$$

Consequently, the final Euclidean transformation  $T_c$  that coarsely aligns the two point clouds has the following expression:

$$T_c = {}^\beta_{W_2} T^{-1} \cdot R_z(k) \cdot {}^\alpha_{W_1} T \quad (22)$$

where

$$R_z(k) = \begin{pmatrix} \cos(\rho_\theta \cdot k) & \sin(\rho_\theta \cdot k) & 0 & 0 \\ -\sin(\rho_\theta \cdot k) & \cos(\rho_\theta \cdot k) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (23)$$

## 5. Results

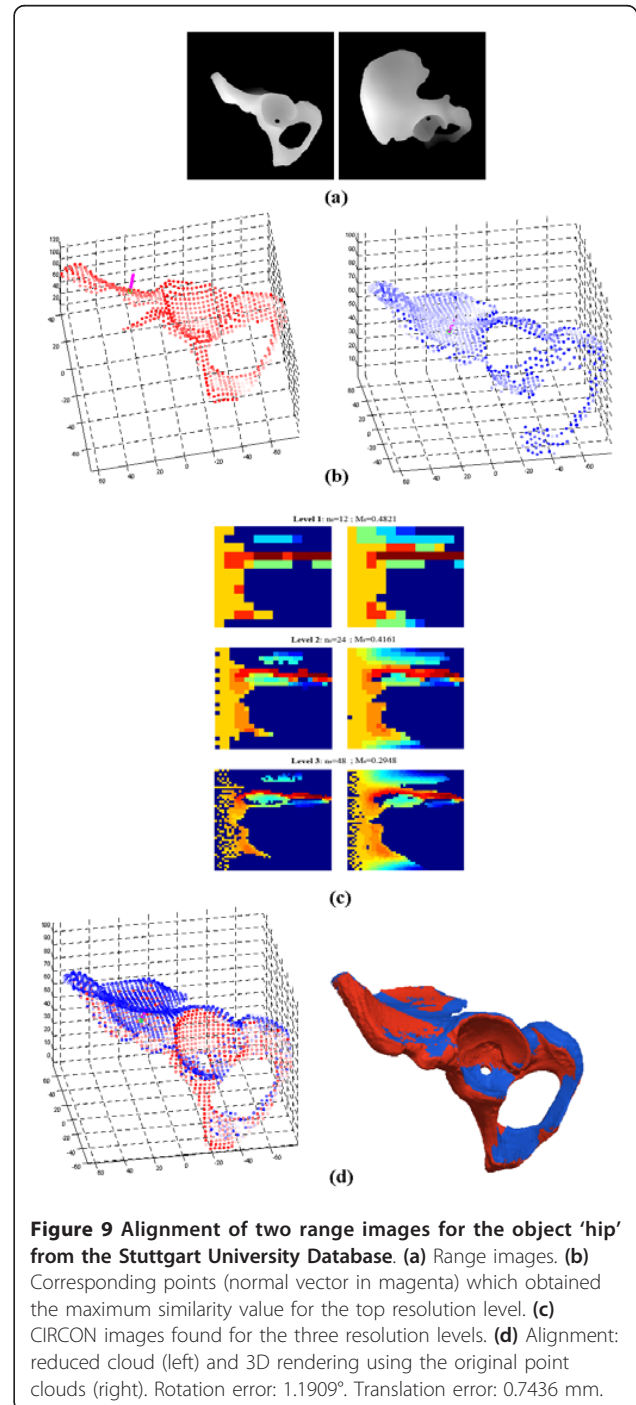
In order to enable the comparison of the proposed alignment algorithm with some of the existing ones we use some objects that were employed in different comparative studies [2,23,31].

As in the comparative study by Salvi et al. [2], an analysis of efficiency of the algorithm will not be carried out since, as discussed here, this is very implementation dependent (in our case *Matlab*<sup>®</sup> was used). For this reason, we assess the performance of the algorithm in terms of effectiveness by measuring the alignment error for different free-form objects.

However, as a reference, it can be said that when the point clouds have an overlap of more than 70%, the time spent by the alignment algorithm is, in most cases, less than 5 s, while for very low overlap percentages, that value can be exceeded. In this case the time increment is due, first, to the need to use more starting points for the algorithm in order to avoid false matches

and secondly because the choice of these points is not sufficiently suitable (given the simplicity of the points-of-interest selection algorithm), which implies, in both cases, an additional number of iterations.

Figure 9 shows an example of the results obtained for the alignment of two range images using the algorithm presented in the previous sections. To accelerate the



**Figure 9 Alignment of two range images for the object 'hip' from the Stuttgart University Database. (a)** Range images. **(b)** Corresponding points (normal vector in magenta) which obtained the maximum similarity value for the top resolution level. **(c)** CIRCON images found for the three resolution levels. **(d)** Alignment: reduced cloud (left) and 3D rendering using the original point clouds (right). Rotation error: 1.1909°. Translation error: 0.7436 mm.

evaluation of the correspondences, reduced point clouds have been used (see Figure 9b). To do this, a simple algorithm has been implemented which creates a grid of  $x$ - $y$  coordinates (with a predetermined spacing) and calculates the  $z$  coordinates through interpolation of the initial values of the surroundings of each new point. Obviously, this reduces the precision of the original point clouds; however, the results obtained for different objects (Figure 10) from the Stuttgart University Database [32] demonstrate the algorithm is robust to these alterations of the data.

The CIRCON images shown in Figure 9c correspond to the points in which the greatest similarity measure was obtained for the three resolution levels. The color map used to show these images was chosen with the aim of visualizing the similarity better.

The number of resolution levels used for the experiments was three with 12, 24, and 48 angular divisions. The number of search columns,  $n_c$ , was, respectively, 8, 4, and 2 so that the number of search cells was always 96. These values were chosen for the implementation of our algorithm by testing different combinations to align synthetic point clouds. It was noted that with less than ten angular divisions the algorithm was faster at the first level, but it favored the emergence of false correspondences, which increases the computation time of the next levels and can lead to incorrect final alignment. On the other hand, we observed that 48 angular divisions for the highest resolution level are sufficient to obtain

an acceptable approximate alignment. Although the maximum error on the rotation around the normal vector that could be committed is  $3.75^\circ$ , in practice the algorithm evaluates the similarity of so many correspondences with different orientations of the normal vector that usually the rotation error is under that value (as shown in the results).

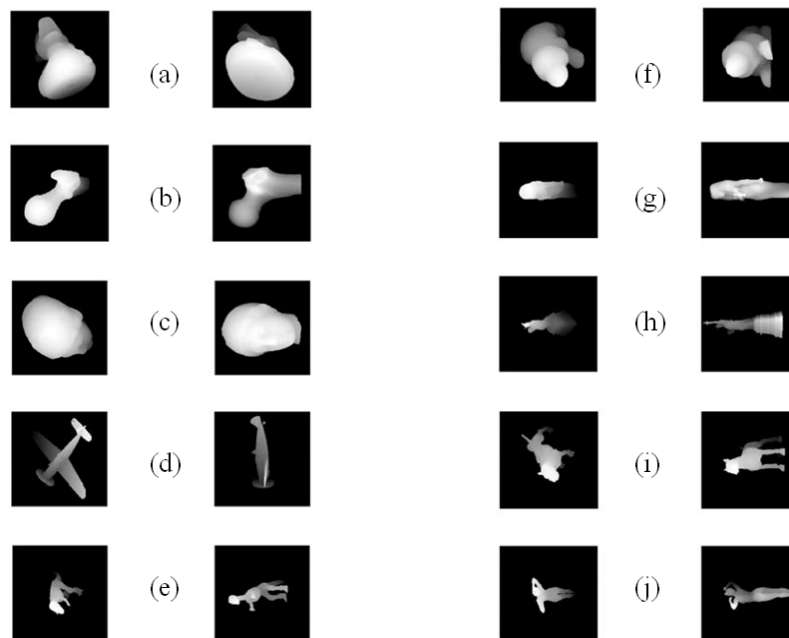
Figure 9d shows the coarse alignment obtained for the reduced point cloud and a 3D rendering when the transformation obtained by the algorithm is also applied to the original data.

The rotation and translation errors were computed using similar expressions to those introduced in Section 5.3. In this case the transformation matrix chosen for comparison was obtained by refinement using a variant of the ICP algorithm [24].

As can be observed in the caption of Figure 9 and in Table 1, the rotation errors are less than  $5^\circ$  and the translation errors are, in all cases, less than the resolution of the two reduced point clouds that are used by the coarse alignment algorithm.

This demonstrates that the algorithm is able to achieve a good performance despite using point clouds with few points (less than 6% of the original quantity) and different resolutions, which make it suitable for aligning point clouds with low density acquired by different devices.

The alignment algorithm was also evaluated using three different similarity measures for comparing the



**Figure 10** Ten pairs of range images for different objects from the Stuttgart University database: (a) Ducky, (b) Femur, (c) Igea, (d) Fighter, (e) Dino, (f) Mole, (g) Isis, (h) Liberty, (i) Pitbull, (j) Female.

**Table 1 Errors obtained by the coarse alignment algorithm for ten different objects**

	Reduced point cloud 1		Reduced point cloud 2		Rot. error (degrees)	Translation error (mm)
	Number of points (% of total)	Resolution (mm)	Number of point (% of total)	Resolution (mm)		
Ducky	859 (1.61%)	3.83	1142 (1.90%)	3.41	2.99	1.15
Femur	453 (1.54%)	3.75	497 (1.25%)	4.32	3.77	1.94
Igea	544 (0.93%)	4.67	419 (0.73%)	5.16	2.49	1.89
Fighter	635 (2.13%)	2.37	401 (2.67%)	2.20	1.50	0.47
Dino	528 (4.92%)	2.31	314 (2.45%)	2.87	4.09	1.00
Mole	673 (1.45%)	3.79	652 (1.65%)	3.41	2.34	1.06
Isis	543 (3.50%)	2.21	381 (1.47%)	3.37	3.62	2.06
Liberty	715 (5.27%)	1.93	428 (2.47%)	2.77	4.84	0.66
Pitbull	153 (0.83%)	4.99	108 (0.58%)	5.29	2.31	0.89
Female	549 (5.58%)	1.90	244 (1.67%)	3.38	3.54	0.62

descriptors: CC, MI, and the measure proposed (SM). This experiment aimed to demonstrate that the combination of descriptor and similarity measure proposed has a better performance under conditions of low density and overlap than the others. Using one single range image and removing different parts two new range images were created for different degrees of overlap. Previously, the resolution of the original range images was reduced by interpolation. Thus, the algorithm was evaluated starting from low resolution with the aim of obtaining an alignment with rotation and translation errors of less than 5° and 5 mm, respectively. If the algorithm were not capable of finding a good alignment, the resolution is increased successively having the original resolution as a limit. The number of points of the reduced range image (before removing parts) necessary for a good alignment is shown in Figure 11 for 12 range images corresponding to four different objects. These results show that the proposed algorithm and similarity measure have a good performance when both density and overlap are low, which is not possible with algorithms that use 2D or 3D descriptors [33], since they need a sufficient quantity of points in order to construct the descriptors.

Moreover, as is shown in Figure 12, the alignment algorithm can also be used to determine the pose of an object in a cluttered scene (problem that arises in bin-picking tasks). In this case, point cloud 1 represents the scene and point cloud 2 represents the object model. Thus, the radius of the point-of-interest environment must be limited to the maximum radius that can be obtained for the object (in Figure 12 this limit was fixed at 120 mm).

## 6. Conclusions










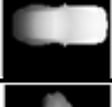
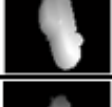

We have introduced a novel descriptor (CIRCON) which represents, through a cyclical image, the geometry

of the environment of a point-of-interest in the cloud. In order to construct the image matrix we distribute the points in sectors which, in turn, are subdivided into cells that have the same radial length. The values of the matrix elements represent the maximum z coordinate of the points contained in their corresponding cells. This represents an important difference with respect to other methods that use 2D histograms, such as spin images [8], which make them more vulnerable to the density changes of the point clouds (especially when their densities are significantly different).

We have also designed a novel similarity measure that takes into account both the distances between the pixels of the descriptors and their degree of overlap, which are not considered by other methods due to the particular characteristics of the descriptors. Furthermore, this similarity measure takes advantage of the cyclical nature of the descriptor to obtain, along with the similarity value, an index that represents the rotation around the normal at the point-of-interest. When the similarity of two descriptors is evaluated, this rotation index, the matched points and their normal vectors can be used to calculate a Euclidean transformation matrix; that is, the two point clouds can be aligned by determining one single correspondence.

Using this similarity measure, the descriptors can be compared without having to restrict the neighborhood of the point-of-interest, so the discriminating power could be increased in order to avoid problems of misalignment when the objects have symmetries or repeated regions (problems that are not well solved by other methods, such as spin images, as is explained in [2]).

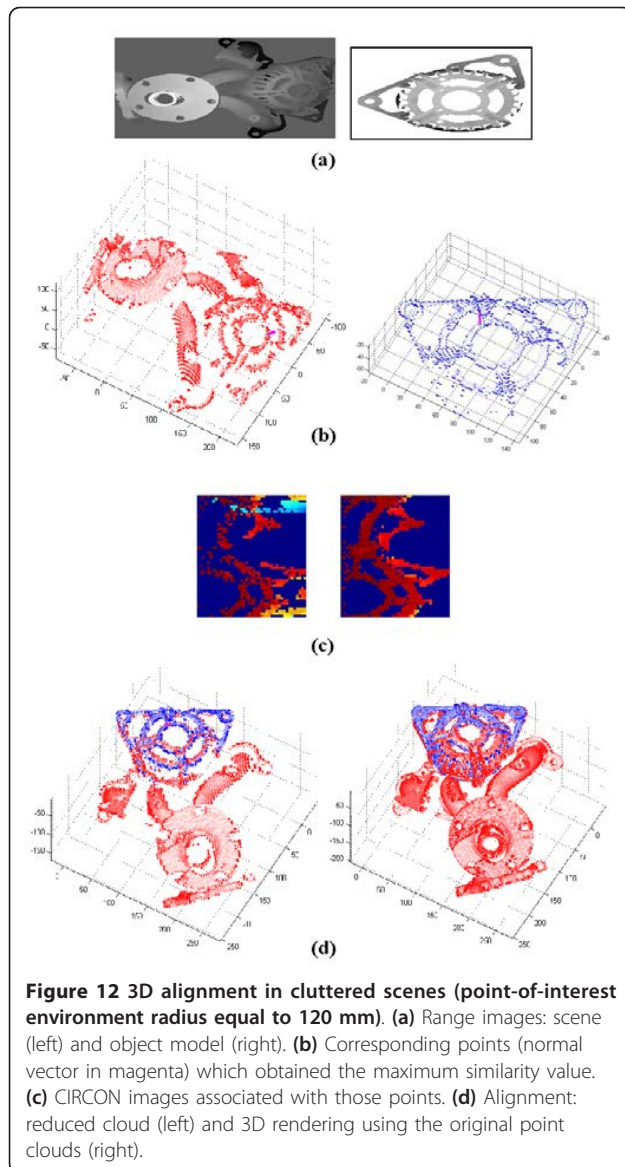
Based on this combination of descriptor and similarity measure we have designed a coarse alignment algorithm that eliminates the need to find a group of valid correspondences (which is necessary in most algorithms, including spin images [8]). One of the main advantages of this algorithm is that the stopping criterion is always

Range Image		Similarity Measure	Overlap between the point clouds				
			0.6	0.5	0.4	0.3	0.2
	Bunny 01	CC	105	105	105	Failure	Failure
		MI	197	105	105	681	Failure
		SM	105	105	105	197	327
	Bunny 02	CC	109	209	Failure	Failure	Failure
		MI	109	209	Failure	Failure	Failure
		SM	109	109	209	209	2161
	Bunny 03	CC	85	Failure	Failure	Failure	Failure
		MI	1528	1821	2141	Failure	Failure
		SM	85	167	85	274	583
	Ducky 01	CC	95	95	Failure	Failure	Failure
		MI	95	2979	Failure	Failure	Failure
		SM	187	95	95	610	1303
	Ducky 02	CC	128	128	Failure	Failure	Failure
		MI	244	244	244	244	2041
		SM	128	128	244	801	1060
	Ducky 03	CC	124	124	Failure	Failure	Failure
		MI	124	124	755	Failure	Failure
		SM	124	124	124	221	370
	Hip 01	CC	101	266	Failure	Failure	Failure
		MI	173	101	Failure	Failure	Failure
		SM	52	173	495	970	Failure
	Hip 02	CC	27	27	27	61	Failure
		MI	27	183	183	265	Failure
		SM	27	61	27	183	Failure
	Hip 03	CC	143	34	143	34	Failure
		MI	34	143	143	Failure	Failure
		SM	76	143	212	414	838
	Porsche 01	CC	325	436	Failure	Failure	Failure
		MI	140	78	Failure	Failure	Failure
		SM	78	78	140	Failure	Failure
	Porsche 02	CC	228	Failure	Failure	Failure	Failure
		MI	439	101	Failure	Failure	Failure
		SM	101	101	158	228	Failure
	Porsche 03	CC	41	41	Failure	Failure	Failure
		MI	267	41	114	Failure	Failure
		SM	41	114	114	267	267

**Figure 11** Points of the reduced cloud (before removing parts) that are necessary for a correct alignment using three different similarity measures and under low overlap conditions. CC, correlation coefficient; MI, mutual information; SM, proposed similarity measure.

evaluated when it finds a correspondence that exceeds the maximum similarity value found until that moment. Thus, if certain conditions are met, the algorithm ends without having to find additional correspondences.

The results show that the proposed algorithm is able to find a proper alignment despite using simple criteria for selecting the points-of-interest. However, in some cases these starting points are not the most appropriate and the



algorithm has to perform more iterations than necessary. As one of the advantages of our proposed algorithm is that it can end once it finds a correspondence that has high similarity and that meets the stopping criterion, if the points-of-interest are appropriately selected, it is very likely that the algorithm could end after the first iterations on the majority of occasions. Furthermore, if these key-points are obtained by new multiscale methods [34], the support sizes can be calculated for the descriptors in both point clouds and the alignment could be carried out, as in [35], using point clouds with different scale.

#### Acknowledgements

This study was carried out with the support of the Spanish CICYT project DPI2006-15313.

#### Competing interests

The authors declare that they have no competing interests.

Received: 9 March 2011 Accepted: 6 March 2012

Published: 6 March 2012

#### References

1. G Sansoni, M Trebeschi, F Docchio, State-of-the-art and applications of 3D imaging sensors in industry, cultural heritage, medicine, and criminal investigation. *Sensors*. **9**, 568–601 (2009). doi:10.3390/s90100568
2. J Salvi, C Matabosch, D Fofi, J Forest, A review of recent range image registration methods with accuracy evaluation. *Image Vis Comput*. **25**, 578–596 (2007). doi:10.1016/j.imavis.2006.05.012
3. BM Planitz, AJ Maeder, JA Williams, The correspondence framework for 3D surface matching algorithms. *Comput Vis Image Understand*. **97**, 347–383 (2005). doi:10.1016/j.cviu.2004.08.001
4. CS Chua, R Jarvis, Point signatures: A new representation for 3D object recognition. *Int J Comput Vis*. **25**, 63–85 (1997). doi:10.1023/A:1007981719186
5. N Gelfand, NJ Mitra, LJ Guibas, H Pottmann, Robust global registration, in *Symposium on Geometry Processing*, Vienna, Austria, pp. 197–206 (2005)
6. J Feldmar, N Ayache, Rigid, affine and locally affine registration of free-form surfaces. *Int J Comput Vis*. **18**, 99–119 (1996). doi:10.1007/BF00054998
7. G Barequet, M Sharir, Partial surface matching by using directed footprints, in *Proc 12th Annual Symp Computational Geometry*, Philadelphia, USA, pp. 409–410 (1996)
8. AE Johnson, M Hebert, Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Pattern Anal Mach Intell*. **21**, 433–449 (1999). doi:10.1109/34.765655
9. AP Ashbrook, RB Fisher, C Robertson, N Werghi, Aligning arbitrary surfaces using pairwise geometric histograms, in *Proc NMBIA98*, Glasgow, UK, pp. 103–108 (1998)
10. AP Ashbrook, RB Fisher, C Robertson, N Werghi, Finding surface correspondence for object recognition and registration using pairwise geometric histograms, in *Computer Vision-ECCV'98*, Freiburg, Germany, pp. 674–686 (1998)
11. SM Yamany, AA Farag, Surface signatures: an orientation independent free-form surface representation scheme for the purpose of objects registration and matching. *IEEE Pattern Anal Mach Intell*. **24**, 1105–1120 (2002). doi:10.1109/TPAMI.2002.1023806
12. T Masuda, Automatic registration of multiple range images by the local log-polar range images, in *Proc Third International Symposium on 3D Data Processing, Visualization, and Transmission*, Chapel Hill, USA, pp. 216–223 (2006)
13. M Körtgen, M Novotni, R Klein, 3D shape matching with 3D shape contexts, in *The 7th Central European Seminar on Computer Graphics*, Budmerice, Slovakia, pp. 1–12 (2003)
14. D Zhang, Harmonic shape images: a 3-D free-form surface representation and its application in surface matching, (Ph.D. dissertation, Carnegie Mellon University, 1999)
15. F Stein, G Medioni, Structural indexing: efficient 2D object recognition. *IEEE Trans Pattern Anal Mach Intell*. **14**, 1198–1204 (1992). doi:10.1109/34.177385
16. JV Wyngaerd, R Koch, M Proesmans, LV Gool, Invariant-based registration of surface patches, in *IEEE International Conference on Computer Vision*, vol. 1. Kerkira, Greece, pp. 301–306 (1999)
17. P Krsek, T Pajdla, V Hlavác, R Martin, Range image registration driven by a hierarchy of surfaces, in *22nd Workshop of the Austrian Association for Pattern Recognition*, Illmitz, Austria, pp. 175–183 (1998)
18. C Song Chen, Y Ping Hung, J Bo Cheng, RANSAC-based DARCES: a new approach to fast automatic registration of partially overlapping range images. *IEEE Trans Pattern Anal Mach Intell*. **21**, 1229–1234 (1999). doi:10.1109/34.809117
19. CS Chua, R Jarvis, 3D free-form surface registration and object recognition. *Int J Comput Vis*. **17**, 77–99 (1996). doi:10.1007/BF00127819
20. J Cheng, H Don, A graph matching approach to 3-D point correspondences. *Pattern Recogn Artif Intell*. **5**, 399–412 (1991). doi:10.1142/S0218001491000223
21. PJ Besl, HD McKay, A method for registration of 3-D shapes. *IEEE Pattern Anal Mach Intell*. **14**, 239–256 (1992). doi:10.1109/34.121791

22. Y Chen, G Medioni, Object modelling by registration of multiple range images. *Image Vis Comput.* **10**, 145–155 (1992). doi:10.1016/0262-8856(92)90066-C
23. S Rusinkiewicz, M Levoy, Efficient variants of the ICP algorithm, in *Proceedings of the Third Intl Conf on 3D Digital Imaging and Modeling*, Quebec City, Canada, pp. 145–152 (2001)
24. WM Wells III, P Viola, H Atsumi, S Nakajima, R Kikinis, Multi-modal volume registration by maximization of mutual information. *Med Image Anal.* **1**, 35–51 (1996). doi:10.1016/S1361-8415(01)80004-9
25. C Studholme, D Hill, D Hawkes, An overlap invariant entropy measure of 3D medical image alignment. *Pattern Recogn.* **32**, 71–86 (1999). doi:10.1016/S0031-3203(98)00091-0
26. A Collignon, F Maes, D Delaere, D Vandermeulen, P Suetens, G Marchal, Automated multi-modality image registration based on information theory, in *Proc of International Conference on Information Processing in Medical Imaging*, Ile de Berder, France, pp. 263–274 (1995)
27. C Studholme, DL Hill, DJ Hawkes, Multiresolution voxel similarity measures for MR-PET registration, in *Proc of International Conference on Information Processing in Medical Imaging*, Ile de Berder, France, pp. 287–298 (1995)
28. D Skerf, B Likar, F Pernus, A protocol for evaluation of similarity measures for rigid registration. *IEEE Trans Med Imag.* **25**, 779–791 (2006)
29. G Penney, J Weese, J Little, P Desmedt, D Hill, D Hawkes, A comparison of similarity measures for use in 2D-3D medical image registration. *IEEE Trans Med Imag.* **17**, 586–595 (1998). doi:10.1109/42.730403
30. C Torre Ferrero, J Llata, S Robla, E Sarabia, A similarity measure for 3D rigid registration of point clouds using image-based descriptors with low overlap. S3DV09, in *IEEE 12th International Conference on Computer Vision, ICCV Workshops 2009*, Kyoto, Japan, pp. 71–78 (2009)
31. T Zinsser, J Schmidt, H Niemann, A refined ICP algorithm for robust 3-D correspondence estimation, in *Proceedings of the International Conference on Image Processing*, Barcelona, Spain, pp. 695–698 (2003)
32. K Eisele, G Hetzel, Range image database, University of Stuttgart. <http://range.informatik.uni-stuttgart.de/htdocs/html/>
33. A Mian, M Bennamoun, RA Owens, A novel representation and feature matching algorithm for automatic pairwise registration of range images. *Int J Comput Vis.* **66**, 19–40 (2006). doi:10.1007/s11263-005-3221-0
34. A Mian, M Bennamoun, RA Owens, On the repeatability and quality of keypoints for local feature-based 3D object retrieval from cluttered scenes. *Int J Comput Vis.* **89**, 348–361 (2010). doi:10.1007/s11263-009-0296-z
35. J Novatnack, K Nishino, Scale-dependent/invariant local 3D shape descriptors for fully automatic registration of multiple sets of range images, in *Proceedings of the 10th European Conference on Computer Vision: Part III*, Marseille, France, pp. 440–453 (2008)

doi:10.1186/1687-6180-2012-57

**Cite this article as:** Torre-Ferrero et al.: 3D point cloud registration based on a purpose-designed similarity measure. *EURASIP Journal on Advances in Signal Processing* 2012 **2012**:57.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)

---