Universidad de Cantabria

Departamento de Tecnología Electrónica,
Ingeniería de Sistemas y Automática

Tesis Doctoral

CONTRIBUCIONES AL ALINEAMIENTO DE NUBES DE PUNTOS 3D PARA SU USO EN APLICACIONES DE CAPTURA ROBOTIZADA DE OBJETOS

Carlos Torre Ferrero
Ingeniero Industrial
2010

CONTRIBUTIONS TO THE REGISTRATION OF 3D POINT CLOUDS FOR USE IN APPLICATIONS OF ROBOTIC MANIPULATION OF OBJECTS (SUMMARIZED VERSION)

Carlos Torre Ferrero

Acknowledgements

I would like to express my deep gratitude to:

- Dr. José Ramón Llata, my supervisor, for their support, motivation and assistance during the duration of this research work.
- Dr. Peter Sturm for giving me the opportunity to work with the Perception Group of the INRIA Rhône-Alpes in Grenoble (France) and for his guidance in the beginning of this thesis.
- The support of the PROFIT Spanish project FIT-020100-2003-715 and the CICYT Spanish project DPI2006-15313.
- My work colleagues for their collaboration and good work atmosphere.
- The Department Secretariat staff for their valuable work.
- And overall, my family and friends for their continuous support, affection and absolute confidence in me.

Contents

List of Figures

Chapter 1

Introduction

1.1 Problem statement and motivation

The main motivation of this thesis is to contribute to finding a more general solution to the problem of locating objects in robotic capture applications. A solution will be sought that can be applied with as few restrictions as possible. This will focus our work on those parts that involve more difficulty, such as object recognition and pose determination, using only acquisition methods which provide information about the scene in the form of 3D point clouds.

The data acquisition methods of three-dimensional object surfaces have improved considerably in the last two decades, so it is becoming more frequent to use point clouds obtained by laser scanners or time-of-flight cameras. However, their use entails certain disadvantages that result in having to have a method to identify the different parts that can divide the point cloud and to determine the orientation of each of these parts by comparison with a known model . There are also problems when it is desired to scan the entire shape of an object (which is known as 3D modelling), as it will be necessary to make several passes with the scanner that subsequently have to be referred to a common coordinate system. In either case, you must have an algorithm for finding correspondences between the two point clouds in order to calculate the transformation that can align them. In our case, since we consider that objects are not deformable, this transformation will be rigid, i.e., it should consist of only rotation and translation.

Our goal is to develop a coarse alignment algorithm that has a wide scope (i.e., not restricted to a particular topology of objects), that does not require high accuracy and density of point clouds and that allows an acceptable solution to be obtained even under low overlap between the point clouds.

1.2 Document overview

This document, which is a summarized version of the thesis document written in Spanish, is organized in the following chapters:

- Chapter 2: A brief review of the state of the art is made. In the complete thesis document an exhaustive one is provided.
- Chapter 3: We propose a novel descriptor that can be applied to 3D point cloud alignment and different tasks in 3D computer vision.
- Chapter 4: A novel similarity measure is proposed. We describe the fundamentals of this measure and how it can be applied to the search for correspondences.
- Chapter 5: A correspondence search algorithm is proposed and described by using an example.
- Chapter 6: A coarse alignment algorithm, which combines the descriptor and the similarity presented respectively in Chapter 3 and Chapter 4, is proposed and described.
- Chapter 7: A version of the algorithm in Chapter 6 that is based on wavelet decomposition is briefly described.
- Chapter 8: We propose a 3D vision system based on the descriptor presented in Chapter 3.
- Chapter 9: We propose an adaptation of the ICP algorithm that uses a cell distribution of the point clouds that is provided by the coarse alignment algorithm.

- Chapter 10: Results.
- Chapter 11: Conclusion and Future work.

Chapter 2

State of the art

2.1 Introduction

In this chapter we review the most significant aspects of 3D point cloud alignment. Though a description of the main acquisition methods of 3D point clouds is not provided in this summarized version, this is included in the complete thesis document.

2.2 3D Point cloud alignment

In general, there are two different types of 3D alignment: pairwise registration and multiview registration [15, 30, 20, 34]. The first type attempts to find a transformation that aligns two scanned partial views or a scanned partial view with a 3D model of the object. In contrast, the second type will aim to create a three-dimensional model of an object by processing groups of views (usually all scans).

This thesis focuses on 3D alignment algorithms between pairs of views, as they have a direct application to problems of pose determination and object recognition.

The process of alignment of two partial views can be divided into two distinct parts: correspondence and alignment. The first part consists in identifying regions of the two point clouds that represent the same physical region of the scanned object, while the second part, alignment, attempts to find the rigid transformation that minimizes the distance between those corresponding regions.

Most existing algorithms attempt to analyze the surfaces in order to find a set of points that will be used to calculate a transformation that aligns them. Therefore, the degree of validity of these correspondence hypotheses will affect the quality of the alignment.

Alignment algorithms can be classified according to how they perform the correspondence between the surfaces. Although there are different ways of carrying it out, the most widespread method consists in matching points with similar surface properties. However, the strategy to follow will be conditioned by the type of surface property that is considered in each case. This will classify the properties into two basic types: intrinsic and extrinsic.

Intrinsic properties are related exclusively to the local information from the surface, that is, they are properties that relate the surface to itself. In contrast, extrinsic properties will relate the surface to the coordinate system in which it is expressed and, therefore, they will be modified when this coordinate system changes. An example of intrinsic property is the distance between two surface points (which is invariant in rigid transformations), while the plane tangent to one of these points is an example of an extrinsic property (it varies when the surface is expressed in a different coordinate system).

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, local alignment algorithms use these properties to refine that initial transformation and obtain a more precise one.

Given the above, it can be concluded that the most challenging part of the 3D point cloud alignment problem is to determine this rough initial transformation, since its refinement process has been properly resolved [5, 7]. A detailed review of the most efficient variants of the ICP algorithm is provided in [31].

2.3 Coarse Alignment Algorithms

This section gives a brief overview of the most relevant coarse alignment methods, basing their classification on the strategy followed to solve the correspondence problem. This classification comprises two groups: those based on descriptors and those based on curves, graphs and structures. In a third group we include methods which cannot be classified in the other two categories. More complete reviews can be found in [32, 29].

2.3.1 Based on descriptors

A descriptor should represent the local surface around a point of interest on a surface (or a point cloud) and possess certain invariant characteristics. The invariance is a very important feature in a descriptor, as this should not change when a transformation is applied to the points.

Next, we list the descriptors used by the most relevant coarse alignment algorithms, which are described in detail in the thesis document: principal curvature [12], point signatures [10], footprints [4], spin images [16, 17], geometric histograms [2, 3], surface signatures [44], harmonic shape images [45], 3D shape contexts [22], log local-polar range images [27], splash [36] and volume integral [13].

2.3.2 Based on correspondence of curves, structures and graphs

These algorithms, instead of basing the search for correspondences on descriptors, will search for curves with different topology, geometric structures, or graphs that should be repeated in both. Thus, in the thesis document, the following methods are explained: bitangent curves [43], intrinsic curves [21], RANSAC-based DARCES [35], intrinsic point matching [9] and graph matching [8].

2.3.3 Other methods

In the thesis document, we also review some methods that cannot be included in the previous categories. These are the following: spherical attribute images [14], correspondence based on mean field theory [1] and correspondence based on genetic algorithms [18].

Chapter 3

CIRCON Descriptor

3.1 Introduction

One of the main drawbacks that we 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 point-of-interest environment, the evaluation of their similarity does not have a direct relationship with the distance between the point clouds. Moreover, since a good alignment is characterized by a small distance between corresponding points, it would be more convenient, in our opinion, to use a descriptor that better represents the geometry of the environment. 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 is found. This is usually caused by a descriptor with low discriminant capacity and the choice of a similarity measure that is not sufficiently appropriate.

In our opinion, the main features that a descriptor should have for a correct correspondence search are:

- Based on the geometry in the environment of the points of interest.
- Highly descriptive, so that the correspondences can be adequately discriminated and no false matches appear.
- It must allow the use of a similarity measure based on distances between

points of the cloud.

- It must allow the calculation of a Euclidean transformation based on a single correspondence.
- Useful for 3D modelling (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).

3.2 Descriptor Construction

3.2.1 Local Coordinate System

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 vector 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.

3.2.2 Cyclical Image of Radial Contours

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 ρ_θ radians), which are further divided radially into cells with length ρ_r mm (excluding the cell closest to the centre, ”cell 0”, which will be a sector with a radius $0.5 \cdot \rho_r$ mm). The sectors are numbered clockwise starting with the sector that is centred on the X_q-axis ($\theta_1 = 0$). Figure 3.2 shows, around the Z_q-axis, this sense of numbering and how the cells would be 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,

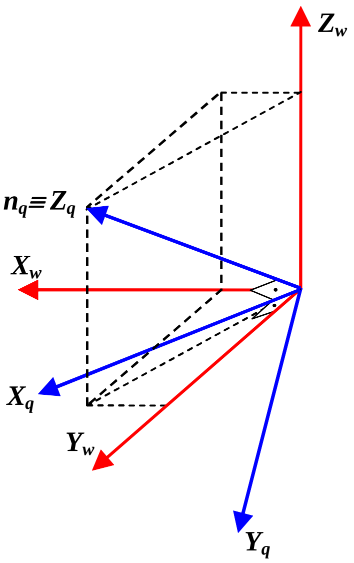


Figure 3.1: Orientation of the local frame axes with respect to those of the global frame W.

for any point p_d with coordinates $(x_d,\, y_d,\, z_d)$ in the coordinate system with origin in p_q, the i index corresponding to the number of sector to which it belongs, the j index indicating the number of cells within that sector, and the height value associated with its coordinate z_d.

$$i = \left(\left\lfloor n_s - \frac{\tan^{-1}\left(\frac{y_d}{x_d}\right)}{\rho_\theta} \right\rceil mod\, n_s \right) + 1 \tag{3.1}$$

$$j = \left\lfloor \frac{\sqrt{x_d^2 + y_d^2}}{\rho_r} \right\rceil \tag{3.2}$$

$$c_{i,j} = \left\lfloor \frac{z_d}{\rho_z} \right\rceil \tag{3.3}$$

Where ρ_θ is the angular resolution, ρ_r the radial resolution and ρ_z the height resolution. Note also that $\lfloor x \rceil$ is the nearest integer to x.

Applying these three equations to all points of the cloud and maintaining with the values $c_{i,j}$ that are maximum for each cell (which is equivalent to

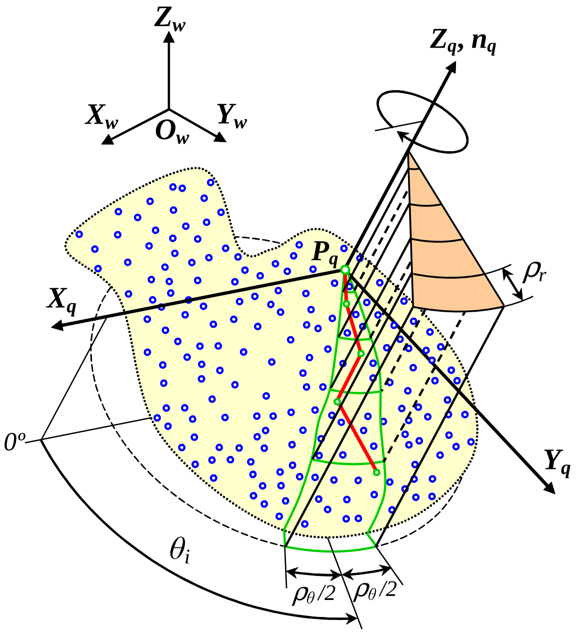


Figure 3.2: Construction of a CIRCON descriptor. Green shows cell division in sector i and red indicates the contour formed by the points of the cells with the greatest z coordinate.

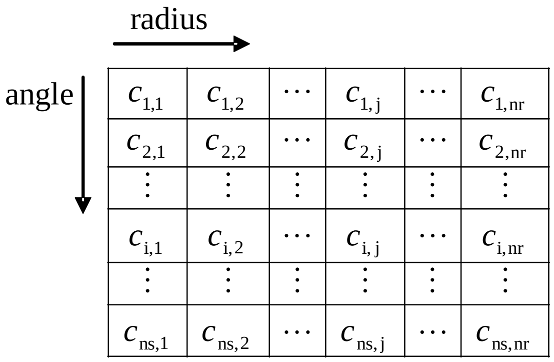
saying that its coordinate z is maximum) we obtain a set of triples $(i,\ j,\ c_{i,j})$ that uniquely correspond to each of the cells into which the environment of the point of interest has been divided.

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 3.2 shows, for sector i, the contour described by points with the biggest '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 which has as row index the number of sector, i , and as column index the number of cell, j, within that sector.

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 3.4) whose pixels



$c_{1,1}$	$c_{1,2}$	$\cdots$	$c_{1,j}$	$\cdots$	$c_{1,nr}$
$c_{2,1}$	$c_{2,2}$	$\cdots$	$c_{2,j}$	$\cdots$	$c_{2,nr}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$c_{i,1}$	$c_{i,2}$	$\cdots$	$c_{i,j}$	$\cdots$	$c_{i,nr}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$c_{ns,1}$	$c_{ns,2}$	$\cdots$	$c_{ns,j}$	$\cdots$	$c_{ns,nr}$

Figure 3.3: CIRCON matrix. The arrows show the decreasing direction for angle and the increasing direction for radius.

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".

3.3 Area represented by the descriptor pixels

In order to build a CIRCON descriptor it is necessary to firstly define the radial resolution ρ_r and the angular resolution ρ_θ with which we will sample the environment of a particular point of interest. These parameters determine the number of rows n_r and the number of columns, n_s, that the matrix of the descriptor will have.

For a single cell in the first ring (one pixel in the first column of the descriptor matrix):

$$A_{P_1} = \frac{A_{anillo\,1}}{n_s} = \frac{\pi \cdot (1.5 \cdot \rho_r)^2 - \pi \cdot (0.5 \cdot \rho_r)^2}{n_s} = \frac{2\pi \cdot \rho_r^2}{n_s} \tag{3.4}$$

If this expression is generalized to a cell between the radii $(j-0.5)\cdot\rho_r$ and

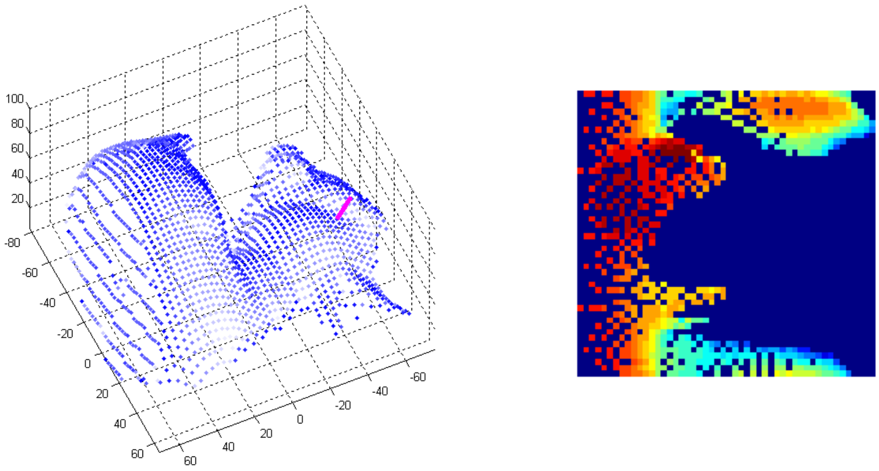


Figure 3.4: Example of a CIRCON image. Point cloud with the normal vector of a point in magenta (left). CIRCON image corresponding to that point (right).

$(j + 0.5)\cdot\rho_r$(corresponding to a pixel in column j of the matrix), the area can be calculated as:

$$A_{P_j} = j \cdot A_{P_1} \tag{3.5}$$

That is, the area representing a pixel in column j of the matrix descriptor is j times the area of a pixel in the first column.

3.4 Applications

The descriptor that we have presented in this chapter can be used for different applications in the field of 3D computer vision which are detailed below.

3.4.1 3D Data Compression

It can also be noted that, when the cloud points are transformed in order to obtain the CIRCON matrix, a compression of the original data is performed. Though this is a lossy compression, the three-dimensional shape of the cloud will be preserved. That is, the transformation to obtain the CIRCON descriptor

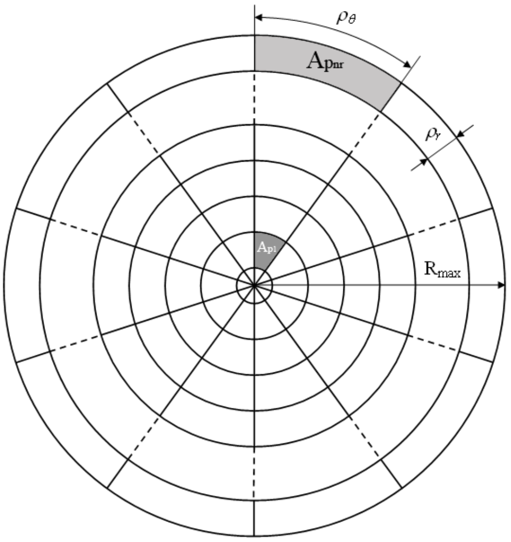


Figure 3.5: Area represented by the pixels in the CIRCON image.

can be undone and an approximation to the original 3D shape is obtained. This property, which is very useful when comparing two descriptors, is not supported by spin images neither other similar descriptors.

3.4.2 3D Registration

As discussed in the State of the Art, 3D registration of two point clouds is usually performed by searching for local correspondences by comparison of intrinsic descriptors. In this thesis, as is explained in Chapter 5, CIRCON descriptors are used to find the corresponding points needed to calculate a rigid transformation that aligns the two point clouds.

3.4.3 Object Recognition

The CIRCON descriptor can also be used for tasks involving object recognition and pose estimation. It will also be useful in ”Bin-Picking” tasks, since its characteristics, in combination with the similarity measure presented in Chapter 4, make it appropriate in situations where the point cloud represents only a small part of the object.

Since the distance between two adjacent elements belonging to the same column of the matrix depends on the radio, those points that are far from the point of interest will be represented less accurately than those which are closer. This provides an advantage in combating problems of occlusions and clutter (mixed objects) since the information will be denser in the environment close to the point of interest, while more distant points (those with more probability of belonging to another object, if they are mixed) have less importance in the descriptor.

Chapter 4

Similarity Measure

As noted in Section 3.1, one of the main features that is demanded on the descriptor is its suitability for being used along with a similarity measure based on distances between points of the cloud. Another requirement is that it facilitates a correct distinction of the correspondences evaluated in order to avoid the appearance of false matches. This problem usually occurs to a greater extent when using similarity measures such as correlation, mutual information [42, 25, 37] and joint entropy [11, 6] (to name some of the most widely used in 3D registration) in the presence of low overlap between the point clouds. As these measures are usually applied to the overlapping area, it is more likely to have more points with similar environments if this common area is small. To avoid such false matches it would be more appropriate to also take into account non-overlapping areas between the two descriptors.

On the basis of the above requirements, we designed a similarity measure based on differences between pixel values of CIRCON images, considering overlapping and non-overlapping areas.

In addition, to try to overcome the drawbacks of the selection of false matches, the CIRCON images will represent the full extent of the point clouds, thus increasing its descriptive power. Though in practice the "reach" of a descriptor could be reduced, it is desirable in order to demonstrate that the similarity measure is capable, even in the presence of low overlap between descriptors, of finding a correspondence that produces an acceptable alignment of the point clouds.

4.1 Pixel Classification

We will make a distinction between *'object'* and *'non-object'* pixels when two CIRCON images are compared to evaluate their similarity.

Pixels will be considered as *'object'* when they correspond to cells that contain at least one point in the cloud. Therefore, the remaining image pixels, whose corresponding cells do not contain any point, are called *'non-object'* pixels. These pixels are computationally considered as a 'not-a-number' value (*NaN*) as it is not possible to calculate a c_{ij} value for them.

On the basis of the previous definitions, the pairs of pixels in two CIRCON descriptors can be classified into the following groups:

- Intersection Set (overlapping pixels):

$$I_{AB} = \{\ (i,j)\ \mid (a_{ij} \neq NaN)\ and\ (b_{ij} \neq NaN)\ \} \tag{4.1}$$

- Exclusive-OR Set (non-overlapping pixels):

$$X_{AB} = \{\ (i,j)\ \mid (a_{ij} \neq NaN)\ xor\ (b_{ij} \neq NaN)\ \} \tag{4.2}$$

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

$$U_{AB} = \{\ (i,j)\ \mid (a_{ij} \neq NaN)\ or\ (b_{ij} \neq NaN)\ \} \tag{4.3}$$

4.2 Similarity measure proposed

In order to obtain a measure based on distance we consider separately the pixels of the overlapping and the non-overlapping areas.

The distance of each pair of overlapping pixels will be weighted by the theoretical area they represent (Equation 3.5). So, the average distance for those pixels is defined by the following expression:

$$D_{ov} = \frac{\sum\limits_{(i,j)\in I_{AB}} j \cdot |a_{ij} - b_{ij}|}{\sum\limits_{(i,j)\in I_{AB}} j} \tag{4.4}$$

On the other hand, since it is not possible to calculate a distance for the corresponding pixels in the non-overlapping areas (as one of the two pixels does not represent the object), a distance $D_{ov} + \lambda$ (con $\lambda > 0$) will be assigned to that pair of pixels.

If the overlap ratio, σ_{ov}, is defined as:

$$\sigma_{ov} = \frac{\sum\limits_{(i,j)\in I_{AB}} j}{\sum\limits_{(i,j)\in U_{AB}} j} \tag{4.5}$$

Then, we will use the following expression that considers, at the same time, the average distance between the overlapping pixels and the overlap ratio between the descriptors.

$$D_S = \frac{D_{ov}}{\sigma_{ov}} + \lambda \cdot \left(\frac{1}{\sigma_{ov}} - 1 \right) \tag{4.6}$$

Note that the value assigned to the parameter λ allows the contribution of the non-overlapping areas to this measure to be controlled.

Although the estimated similarity measure D_S could be used directly, it is often more desirable to have a normalized measure that provides values close to one for values of D_S near zero and near zero when D_S is very large.

To normalize the similarity measure to values between 0 and 1, we will use the following expression:

$$M_S = \frac{1}{\rho \cdot D_S + 1} \tag{4.7}$$

where ρ is a parameter that modifies the relationship between D_S and M_S.

If we define the parameter λ' as $\lambda' = \rho.\lambda$, the expression for M_S will be as follows,

$$M_S = \frac{\sigma_{ov}}{(\rho \cdot D_{ov} + \lambda') + \sigma_{ov} \cdot (1 - \lambda')} \tag{4.8}$$

This similarity measure will be used to compare descriptors in order to find correspondences between the point clouds. A more detailed explanation is provided in [38].

4.3 Rotation matrix associated with the similarity measure

Let us suppose that a correspondence $a \leftrightarrow b$ has been evaluated, which implies that the corresponding points, ${}^{w_1}p_a$ and ${}^{w_2}p_b$, and their normal vectors, ${}^{w_1}n_a$ and ${}^{w_2}n_b$, are known (the left superindex indicates the coordinate system in which they are expressed).

As seen in the previous section, when the similarity value $M_{S\,max}$ between two descriptors is obtained, the rotation index k_a (associated with the correspondence $a \leftrightarrow b$) is also determined. Therefore, by using this index, the corresponding rotation matrix can be obtained, which will represent a rotation of $-k_a \cdot \rho_\theta$ radians around the Z_a axis of the local coordinate system :

$$R_{Z_a}\left(-k_a \cdot \rho_\theta\right) = \begin{bmatrix} \cos\left(k_a \cdot \rho_\theta\right) & \sin\left(k_a \cdot \rho_\theta\right) & 0 \\ -\sin\left(k_a \cdot \rho_\theta\right) & \cos\left(k_a \cdot \rho_\theta\right) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.9}$$

Its associated transformation matrix being:

$$T_{Z_a}\left(-k_a \cdot \rho_\theta\right) = \begin{bmatrix} R_{Z_a}(-k_a \cdot \rho_\theta) & 0_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix} \tag{4.10}$$

Chapter 5

Correspondence Search

5.1 Introduction

We have designed an algorithm that searches for the rough correspondence of a point of interest from one of the two point clouds. This algorithm evaluates the similarity between the descriptor associated with that point of interest and the descriptors that can be extracted around a point of interest in the other point cloud.

Before applying the correspondence search algorithm it will be desirable to reduce the number of points (if this is excessive) and select a set of points of interest in the resulting cloud. These two operations can significantly reduce the computation time with little decrease in effectiveness.

We propose an algorithm which reduces the amount of information in the point cloud that uses the range image in which the point cloud is expressed and then, which performs a data interpolation.

Although there are different methods for selecting points of interest in a cloud, in our case we have designed one that takes into account the particular characteristics of the CIRCON descriptor. Through a series of operations on the reduced range image, points will be selected in which the CIRCON descriptor is potentially more stable and has more discriminating power.

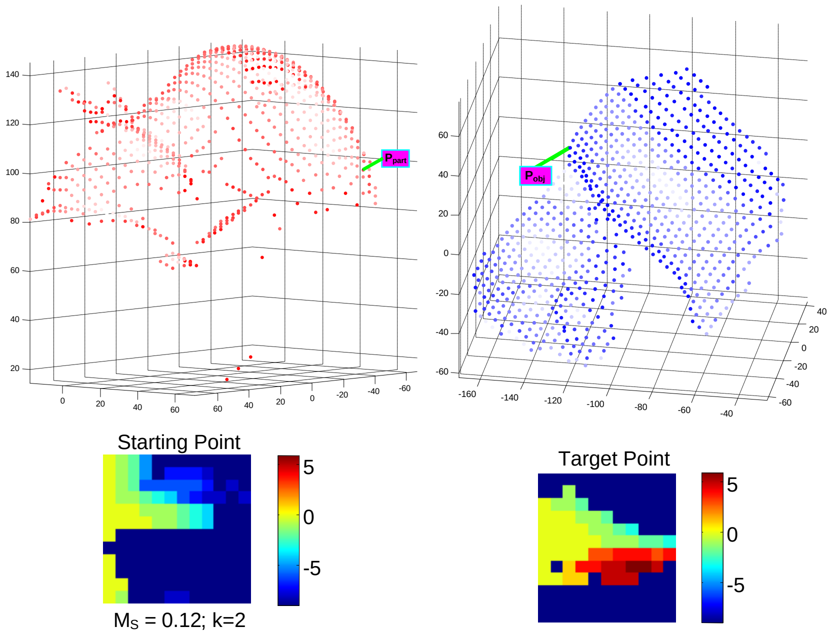


Figure 5.1: To find the correspondence of a *target point* in cloud 2 (top right) the search algorithm starts at a point of interest in cloud 1 (top left). CIRCON images corresponding to these two points are shown below along with their similarity value and rotation index k.

5.2 Correspondence Search Algorithm

This search algorithm is the core of the coarse alignment algorithm. It is based on an iterative search of the greatest value of similarity measure using cells into which the environment of a starting point of interest 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 the 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).

Figure 5.1 shows a *starting point* in cloud 1 (top left) which is where the search begins, and a *target point* in cloud 2 (top right), the aim being to find the point in cloud 1 corresponding to that *target point*. The maximum similarity between them (with a value of 0.12) occurs, in this case, when the rows of the CIRCON image at the *starting point* are rotated clockwise two positions. Since, in this example, the number of rows of the descriptors is $n_s = 12$, an index $k = 2$ implies a clockwise rotation of 60 degrees around the normal at the *starting point*. To make the comparison easier for the reader's eyes, the CIRCON images shown in Figures 5.1 and 5.2 have been rotated the number of positions indicated by the index k.

Figure 5.2 shows the results of the three iterations performed by the search algorithm until the stop condition was found. For each iteration, the algorithm performs as many similarity measure evaluations as the number of valid cells in the distribution around the point of the previous iteration (one point is extracted from each cell). In this case, the stop occurs because the result obtained in iteration 3 coincides with the result of iteration 1, which implies that, as the conditions in both iterations are identical, it would again repeat the same sequence of results.

The similarity value that the algorithm returned in this case was $M_{Sc} = 0.44$ (with an index $k_{1c} = 8$), which is the highest obtained in the three iterations. Therefore, the point obtained in the second iteration will be the approximate correspondence, p_{1c}, of the *target point* in cloud 2 (see Figure 5.1).

In summary, this correspondence search algorithm starts with a selected point in cloud 1, p_{1x}, with the aim of finding the correspondence of a *target point* in cloud 2 of which its CIRCON image, I_2, is known. After successive iterations where point p_{1x} is modified using the cell distribution around the point of the previous iteration, and once the stopping condition is satisfied, the result obtained is the point p_{1c} with the highest similarity value (M_{Sc}) of all iterations. k_{1c} is also obtained (rotation index around the normal at p_{1c}).

The algorithm ends when an iteration uses a starting point whose distance to one of the previously used points is less than a preset δ (in our implementation $\delta = \rho_r/16$).

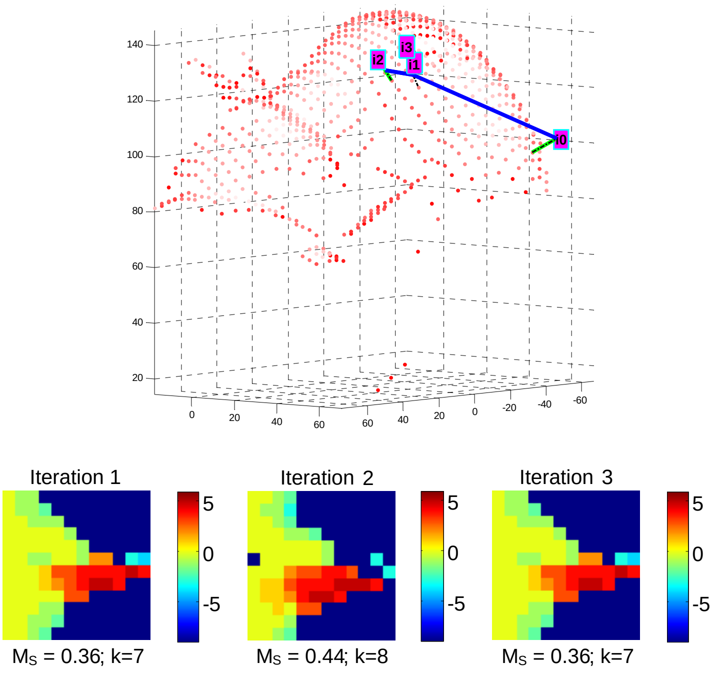


Figure 5.2: Example of the path described by the correspondences with the maximum similarity in each iteration (above) until convergence of the algorithm occurs. CIRCON images of the three iterations are shown below along with their similarity values and indices k.

5.3 Transformation Matrix associated with one correspondence

When the correspondence search algorithm finds a correspondence $a \leftrightarrow b$, a rough transformation matrix that aligns the two clouds of points can be calculated using the following expression:

$$ {}^{w_2}_{b\leftrightarrow a}T_{w_1} = \begin{bmatrix} {}^{w_2}_{b\leftrightarrow a}R_{w_1} & {}^{w_2}_{b\leftrightarrow a}p_{w_1 org} \\ 0_{1\times 3} & 1 \end{bmatrix} \tag{5.1} $$

where,

$$ {}^{w_2}_{b\leftrightarrow a}R_{w_1} = {}^{w_2}R_b \cdot R_{Z_a} \cdot {}^{w_1}R_a^T \tag{5.2} $$

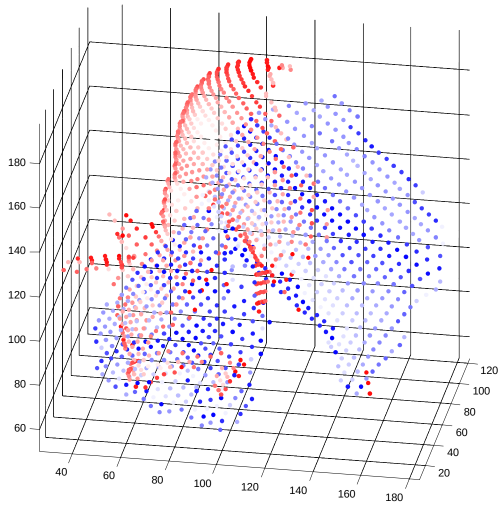


Figure 5.3: Rough alignment obtained from the correspondence shown in Figure 5.2. The two point clouds are expressed in the original coordinate system of point cloud 2.

and

$$ {}^{w_2}_{b\leftrightarrow a}p_{w_1 org} = {}^{w_2}p_b - {}^{w_2}_{b\leftrightarrow a}R_{w_1} \cdot {}^{w_1}p_a \tag{5.3} $$

As is shown, the transformation matrix associated with a correspondence $a \leftrightarrow b$ depends only on the matching points ${}^{w_1}p_a$ and ${}^{w_2}p_b$, on the normal vectors (since ${}^{w_1}R_a$ and ${}^{w_2}R_b$ were constructed entirely with their components) and on the index k_a which expresses the rotation around the normal in ${}^{w_1}p_a$ (since the matrix R_{Z_a} is calculated using k_a).

Applying Equation 5.1 to the correspondence shown in Figure 5.2, we obtain the rough alignment shown in Figure 5.3.

Chapter 6

Coarse Alignment Algorithm

6.1 Introduction

The algorithm presented in Section 5.2 begins with a single point of interest in cloud 1 which is modified iteratively to find the approximate correspondence of a point of interest in cloud 2. However, when two point clouds with a small common part (low overlap) are to be aligned, it may be that, if both are partial, a selected point in cloud 2 has no corresponding point in cloud 1. Therefore, the alignment algorithm should select a number of points of interest in cloud 2, $\{p_{2i}\}$, so that there is a greater likelihood of finding valid correspondences. For each of those points, which we call *'target points'*, several searches for its corresponding point in cloud 1 will be carried out using different levels of resolution. These searches (following the algorithm described in Chapter 5) should begin, for the first level, with a set of points of interest, $\{p_{1i}\}$, previously selected in cloud 1.

6.2 Algorithm description

The basic strategy of the alignment algorithm is to accelerate the search for correspondences by using multiple resolution, which will subsequently enable areas to be discarded when it is not probable to find the correspondence of the *target point* p_{2y}.

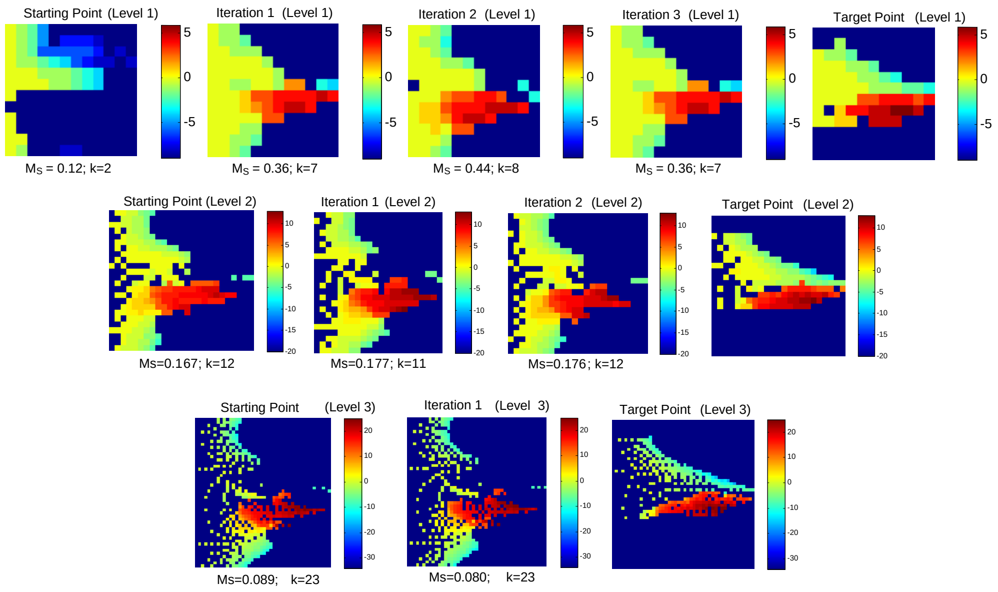


Figure 6.1: Example of refinement of the correspondence at three resolution levels. The target images are shown on the right.

Once the Correspondence Search Algorithm has found an approximate correspondence for the first level, the resolution is increased and a new search around this new *starting point* is performed but with smaller cells.

When the convergence of the search associated with the last resolution level is achieved and the similarity value of the resulting correspondence is greater than a value $\tau_{M_S}(n_v)$, its corresponding Euclidean transformation is calculated (using Equation 5.1) and its validity is checked. If the result of this check is positive, the rough alignment of the two point clouds is determined by this transformation.

Figure 6.1 shows how to obtain the correspondence of the same *target point* as in Figure 5.1 through a progressive refinement at three resolution levels. As resolution increases the search is refined and the number of iterations reduced.

Algorithm 6.1 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** $M_{Sc} < \tau_{M_S}(l)$, Go to step 12; **else** $\tau_{M_S}(l) = M_{Sc}$, **end if;**
      7. **if** level $l$ is the last one:
        (a) The Euclidean transformation ${}^{w_2}_{y\leftrightarrow c}T_{w_1}$ 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 $p_{2y}$, $p_{2n}$ and $p_{2s}$ from the cloud 2, a fictitious correspondence $t_1 \leftrightarrow t_2$ is created and its associated Euclidean transformation is calculated: ${}^{w_2}_{t_1\leftrightarrow t_2}T_{w_1}$.
        (e) The distances $d_R$ and $d_t$ between the transformations ${}^{w_2}_{y\leftrightarrow c}T_{w_1}$ and ${}^{w_2}_{t_1\leftrightarrow t_2}T_{w_1}$ are calculated.
        (f) **if** $(d_R < \tau_R)$ **and** $(d_t < \tau_t)$**,** The algorithm is ended. **end if**
      8. **end if**
      9. $p_{1x} = p_{1c}$
      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 already been evaluated)
  13. **end while**
  14. $p_{2y}$=Next point of $\{p_{2i}\}$
15. **end while**
```

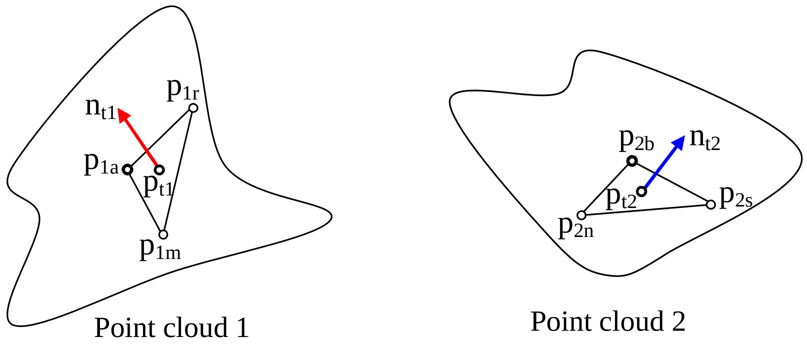


Figure 6.2: Evaluation of the stopping criterion. The points p_{t_1} and p_{t_2}, and their normal vectors, are obtained from the correspondence $(a \leftrightarrow b)$ found by the alignment algorithm and two additional correspondences $(m \leftrightarrow n$ and $r \leftrightarrow s)$. The new correspondence $t_1 \leftrightarrow t_2$ will be used to calculate a more robust Euclidean transformation.

6.3 Evaluation of the stopping criterion

When the algorithm has reached the final resolution level and has surpassed the corresponding threshold $\tau_{M_S}(n_v)$ (which is the maximum similarity value found so far), it will carry out the evaluation of the stopping criterion of the algorithm.

Figure 6.2 shows how the fictitious correspondence $t_1 \leftrightarrow t_2$ is obtained. Then, a Euclidean transformation ${}^{w_2}_{t_2\leftrightarrow t_1}T_{w_1}$ is calculated using Equation 5.1.

We will decide if the algorithm must end or continue based on the difference between the transformations ${}^{w_2}_{b\leftrightarrow a}T_{w_1}$ and ${}^{w_2}_{t_2\leftrightarrow t_1}T_{w_1}$. To do this we use a measure for the rotation and another for the translation.

Figure 6.3 shows the rough alignment obtained by the algorithm for the correspondence found in the example in Figure 6.1 after checking the validity with $\tau_R = 4^{\text{o}}$ y $\tau_t = 2\,mm$. The result of applying the same Euclidean transformation to the original point cloud is also shown.

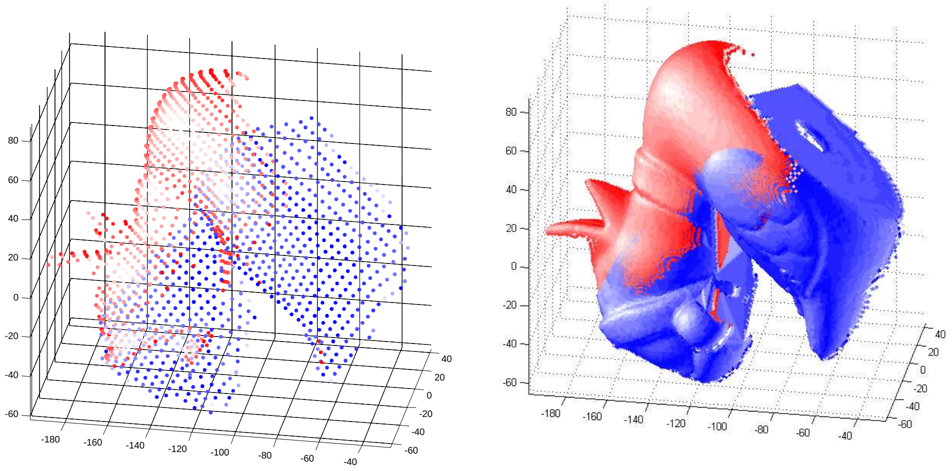


Figure 6.3: Coarse alignment of the reduced point clouds using the Euclidean transformation obtained by the algorithm (left). Coarse alignment of the original point clouds (right).

Chapter 7

Coarse Alignment Algorithm based on Wavelet Decomposition

In this chapter we describe an algorithm for 3D coarse alignment that uses a strategy similar to that presented in Chapter 6, but in this case, descriptors are generated from the simplified 3D shape (see Section 3.4.1 and [40]) obtained from a *basic descriptor* (this must be built using the environment of the closest point to the centre of mass of the point cloud). Then, the search for correspondences is based on wavelet decomposition [26] of these *basic descriptors*. This speeds up the search by reducing, for each resolution level, the number of descriptors that must be evaluated. Finally, the information obtained by this process is used for the calculation of a Euclidean transformation (translation and rotation) that allows a coarse alignment of the two point clouds to be obtained. An interesting characteristic of this algorithm is that this Euclidean transformation depends only on indices i_S and j_S (corresponding to the row and column of the *basic descriptor* that determine the position of the point whose associated descriptor provided the highest similarity value) and the number of rows, k_S, that were shifted to calculate that similarity value.

A more detailed explanation can be found in [41].

Chapter 8

Vision System based on CIRCON descriptor

The 3D vision system proposed obtains a three-dimensional reconstruction of an object placed on a rotating platform and also generates a CIRCON descriptor that will facilitate the alignment of the point cloud with one from another scan or with the complete object model.

This vision system uses a variant of the active triangulation method. As shown in Figure 8.1, its structure is composed of two cameras on either side of the platform that will remain fixed while rotating the object. For each predetermined angle, the laser projector illuminates an object contour. Then, the two images acquired by the cameras are processed by a computer to obtain the three-dimensional reconstruction of the contour just before the platform is moved to the next position.

As both the laser plane and the cameras are calibrated, the points which are only seen by one of the two cameras can be obtained by active triangulation. Furthermore, in order to avoid some parts not being illuminated due to occlusions caused by the object itself, two mirrors are placed for reprojecting the laser and so those regions are illuminated.

The points acquired for each contour can be stored following the matrix structure of the CIRCON descriptor. Note that, because the CIRCON image is built with coded radial contours, two rows of the matrix are filled (the rows

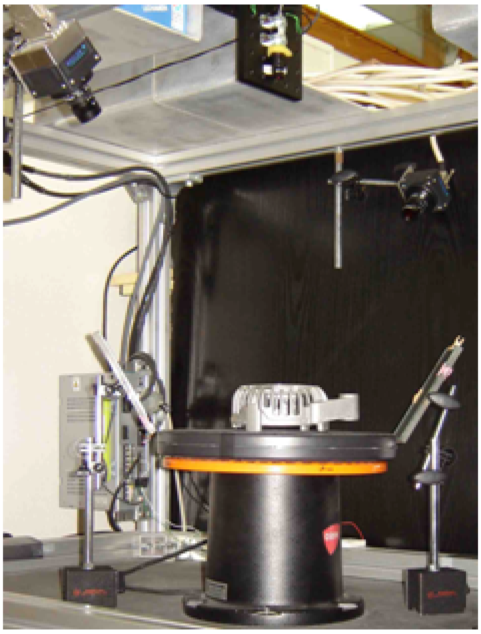


Figure 8.1: 3D vision system proposed.

i and $i + N$), for each reconstructed contour i (where N is the number of predetermined angles of rotation and equal to $n_s/2$).

The descriptor obtained can be used directly for tasks that involve both the recognition of the object placed on the platform and the estimation of its pose. Besides, the vision system presented could be integrated with a robotic manipulator by mounting the cameras and the laser on it, so the wrist rotation can be used to rotate them at the same time while the objects remain stationary on a flat surface or stacked in a container. This setup could be useful for automatic object capture, so, if the rotation axis points toward one of the object points, the resulting CIRCON descriptor will be centred on this point and it may be compared with that of the model to determine the position and orientation of the object.

A more detailed description of this 3D vision system can be found in [39].

Chapter 9

Adaptation of ICP Algorithm

As discussed in the State of the Art, the fine alignment algorithms need an initial Euclidean transformation that roughly aligns the two point clouds. Using this starting alignment, correspondences between points are established (based mainly in proximity) and they are used to calculate a new transformation matrix that will be refined iteratively.

In this case, the ICP algorithm will be adapted to take advantage of the data set that is provided by the 3D alignment algorithm described in Chapter 6. The Euclidean transformation found is used to initialize the ICP algorithm, while the arrays, C_1 and C_2, containing the indices of the points of each cloud distributed in cells around a central point, will be used to establish the correspondences.

By assigning to each correspondence a weight w_k which depends on its distance from the central point of the CIRCON image, more distant correspondences will be taken account to a greater extent in the calculation of the Euclidean transformation. In this way we try to prevent alignment error being greater at these points. In order to calculate the transformation matrix, we use an adaptation of the closed-form solution presented by Maurer in [28].

Another way to avoid this problem is to use one or two additional cell distributions at points of interest that are sufficiently far away from the one obtained by the coarse alignment algorithm. This allows different correspondences to be obtained which will mean the transformation is less affected by errors from the most distant correspondences.

Chapter 10

Results

10.1 Introduction

The experiments that were carried out, which are presented in this chapter, have two main aims:

- To show that the similarity measure is able to select the best correspondence for different degrees of overlap when the descriptors are built using the points of the reduced partial clouds (Section 10.2).
- To apply the alignment algorithm to point clouds from different sources and show its robustness in finding a coarse rigid transformation (Section 10.3).

In all the experiments we use three resolution levels with 12, 24 and 48 angular divisions, since this allows a balance between a suitable description of the point-of-interest environment and a low computational requirement.

10.2 Evaluation of the similarity measure

In this experiment we compare the 3D alignment results obtained with the similarity measure presented in Chapter 4 with those obtained using two commonly used similarity measures for 3D alignment algorithms: correlation coefficient and mutual information.

The objective of this experiment is to characterize how the algorithm, described in Chapter 6, behaves with each of these three measures of similarity

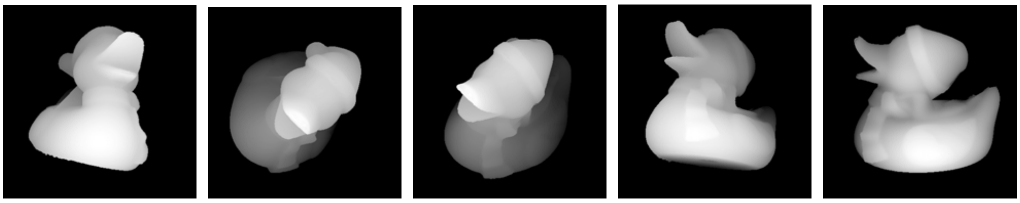


Figure 10.1: Range images of the object 'ducky' used to evaluate the similarity measure.

when the percentage of overlap of the two point clouds is changed.

To make this comparison we use synthetic range images (obtained from the database in [19]) which are modified to obtain two partial point clouds that are not aligned and have a certain percentage of overlap. Figure 10.1 shows the range images used for the object 'ducky'.

By cropping the range images we obtained different degrees of overlap and then, one of the two resulting point clouds is rotated and translated. The algorithm must find the transformation matrix that was used.

We test the algorithm performance for five range images from five different objects using three similarity measures: the one proposed, mutual information and correlation coefficient. Then, the degree of overlap is varied and the number of points of the cloud (before cropping) is increased until the rotation error is less than 5 degrees and the translation error is less than 5 mm. We begin using a reduced range image whose rows or columns are 15 and then, if the error requirements are not met, we use a reduced range image with five more rows or columns and so on until the rotation and translation errors are smaller than the preset errors. We establish a maximum size of 80 rows or columns for the reduced range images. If this limit is reached we consider that the algorithm fails because it does not find a proper alignment using a reasonable size for the reduced range images.

Figure 10.2 shows, for the object 'ducky', how the similarity proposed is able to find a proper alignment for the range images 'ducky01', 'ducky02' and 'ducky03' with an overlap ratio equal to 0.2. However, using mutual information as the similarity measure, the algorithm only finds a good solution for the range image 'ducky03' which uses more points than the one that our measure finds. Moreover, using the correlation coefficient, the algorithm starts to fail when the overlap ratio is less than 0.5.

Note that, using the similarity measure proposed with an overlap ratio of 0.3, a proper alignment was found for the five range images, while only one was found with the mutual information and none with the correlation coefficient.

Note also that, only one alignment with the similarity measure proposed ('ducky05' with overlap 0.3), needs a reduced cloud with a number of points greater than 1500. However, for this range image, the other two similarity measures, MI and CC, only find a good solution for an overlap of 0.5 and 0.6 respectively.

If the ovelap ratio is less than 0.2 the algorithm fails in all cases with the requirements we have fixed.

10.3 Point cloud alignment

The 3D alignment algorithm presented in Chapter 6 was tested with different types of point clouds: synthetic data and real data (from a laser scanner and from a time-of-flight camera) in order to demonstrate its performance in both cases. The first type of data is used to compare the algorithm with other methods and the second one to show how it performs in real world conditions.

10.3.1 Synthetic point clouds

In order to compare the proposed alignment algorithm with some of the existing ones we use some objects that were employed in comparative studies such as [32, 46, 31]).

The data used correspond to partial clouds of different objects from the database of range images of the University of Stuttgart [19]. The errors that appear in the footnotes of the figures correspond to the rotation and translation errors of the rough transformation with respect to a Euclidean transformation previously obtained with the adaptation of the ICP algorithm (since it provides a fine alignment of the two point clouds).

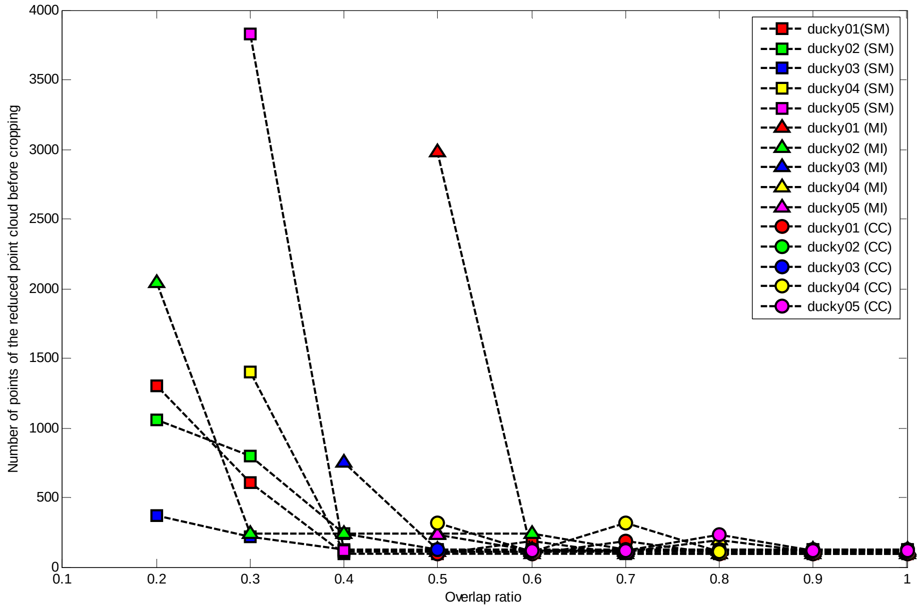
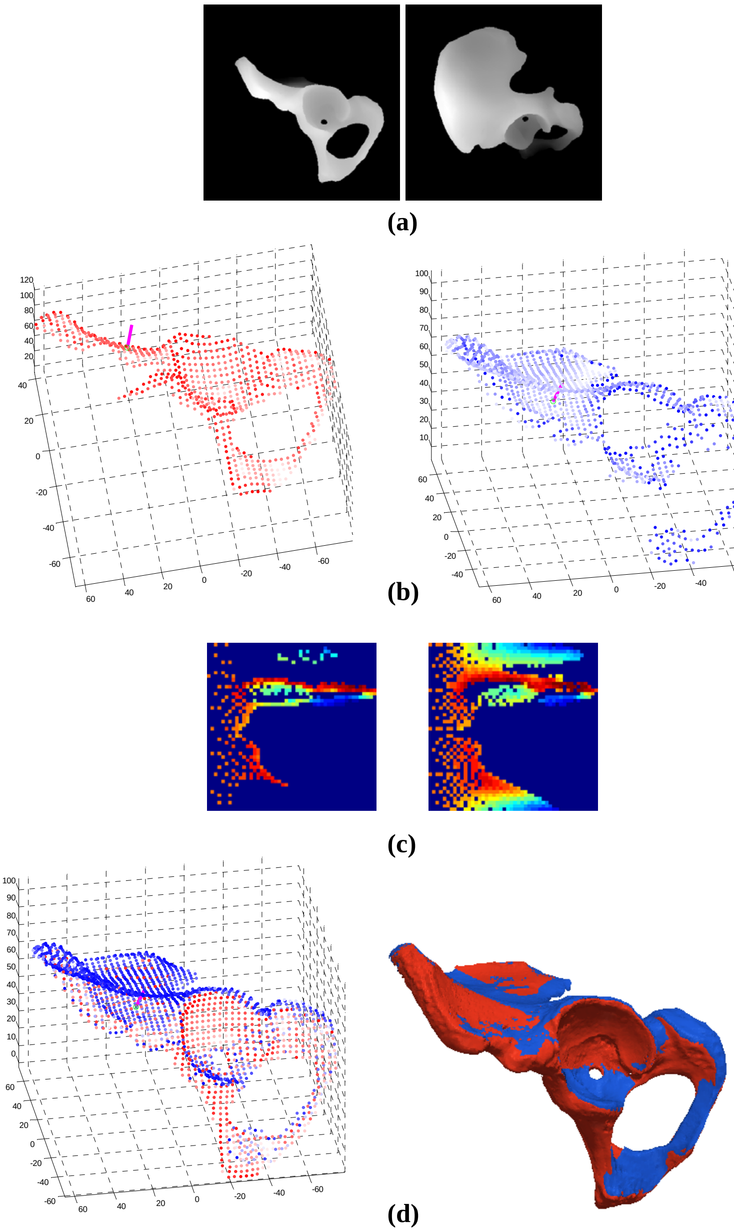


Figure 10.2: Algorithm performance with three similarity measures and five range images from the object 'ducky': the one proposed (SM), mutual information (MI) and correlation coefficient (CC). The degree of overlap is varied and the number of points in the cloud (before cropping) is increased until the rotation error is less than 5 degrees and the translation error is less than 5 mm.

Figure 10.3 shows an example of the results obtained by the coarse alignment algorithm proposed in this thesis with one of the test objects. In this case, the range images have low overlap but the two descriptors with the highest similarity value provide a quite good alignment.

Figure 10.4 shows an example for one of the objects used in the comparative study in [32]. The spin images provided, in that case, a bad alignment for the same object.



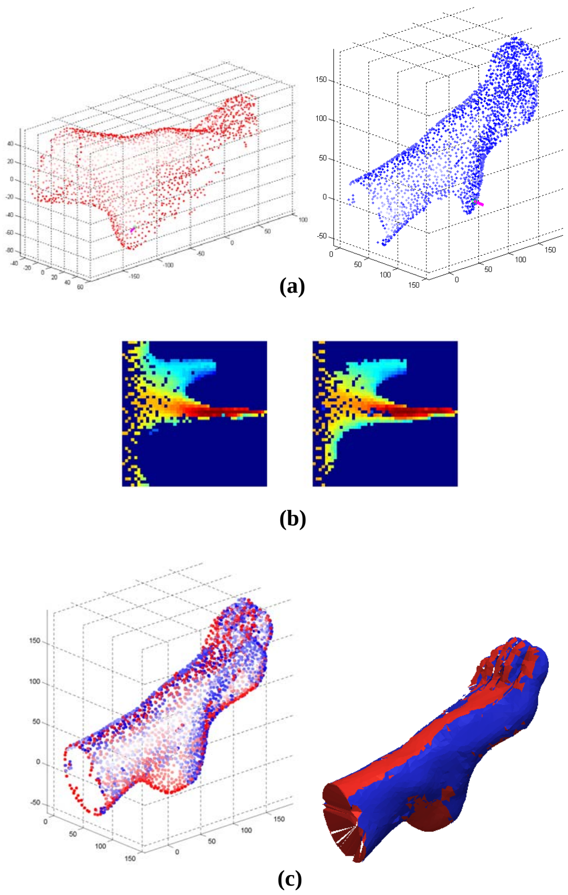
(a)

(b)

(c)

(d)

Figure 10.3: Alignment of two range images for the object 'hip'. **(a)** Range images. **(b)** Corresponding points (normal vector in magenta) which obtained the maximum similarity value. **(c)** CIRCON images associated with those points. **(d)** Alignment: reduced cloud (left) and 3D rendering using the original point clouds (right). Rotation error: 1.1909º. Translation error: 0.7436 mm.



(a)

(b)

(c)

Figure 10.4: Alignment obtained by the algorithm for the object 'Foot'. **(a)** Corresponding points (normal vector in magenta) that obtained the maximum similarity value. **(b)** CIRCON images associated with those points. **(c)** Alignment: reduced cloud (left) and 3D rendering using the original point clouds (right). Rotation error: 0.3668º. Translation error: 0.6111 mm.

10.3.2 Point clouds from laser scanners

The alignment algorithm was tested with point clouds taken from the laser triangulation scanner shown in Figure 10.5. Figure 10.6 shows an example of the alignment obtained for two range images acquired with this device.

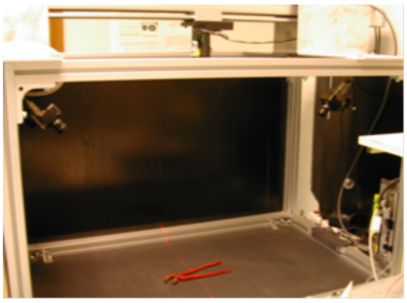
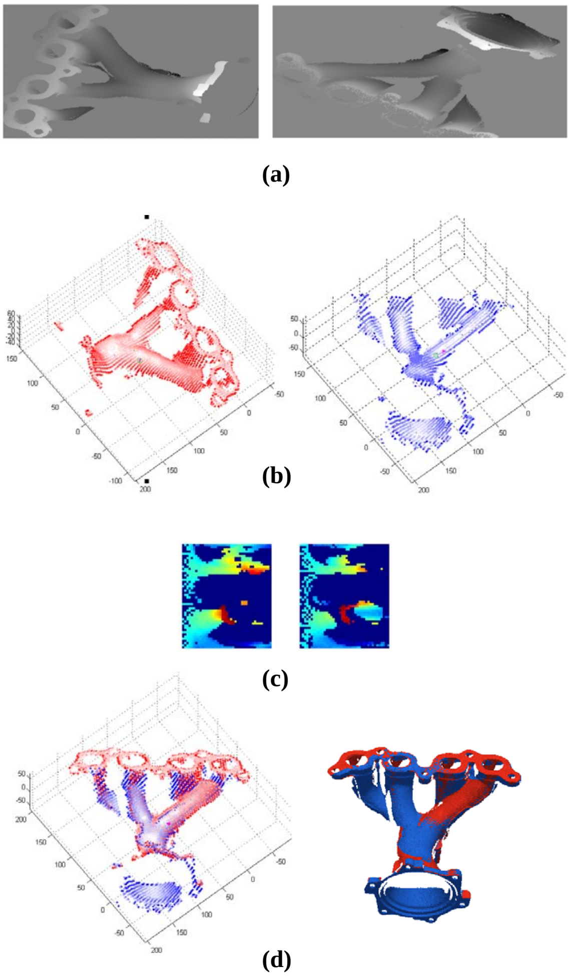


Figure 10.5: Laser scanner used to acquire the range images.



(a)

(b)

(c)

(d)

Figure 10.6: Alignment of two range images from a laser scanner. **(a)** Range images. **(b)** Corresponding points (normal vector in magenta) which obtained the maximum similarity value. **(c)** CIRCON images associated with those points. **(d)** Alignment: reduced cloud (left) and 3D rendering using the original point clouds (right). Rotation error: 1.7427º. Translation error: 0.7638 mm.

10.3.3 Point clouds from time-of-flight cameras

Another way to obtain three-dimensional information is through time-of-flight cameras. In this case we used the model SR4000 of MESA Imaging AG (see Figure 10.7), which provides range images of 176 × 144 pixels almost in real time. This camera has a measuring range that goes from 0.8m to 5m with an absolute accuracy of ±10mm, which can result in substantial variation of the position of the points on the object surfaces. However we will use objects that have dimensions exceeding 20cm in length, so that the lack of resolution and precision of the camera is compensated by a sufficient number of surface points. Figures 10.8 and 10.9 show how the algorithm is capable of aligning point clouds from TOF cameras.

Figure 10.7: TOF Camera used in our experiments.

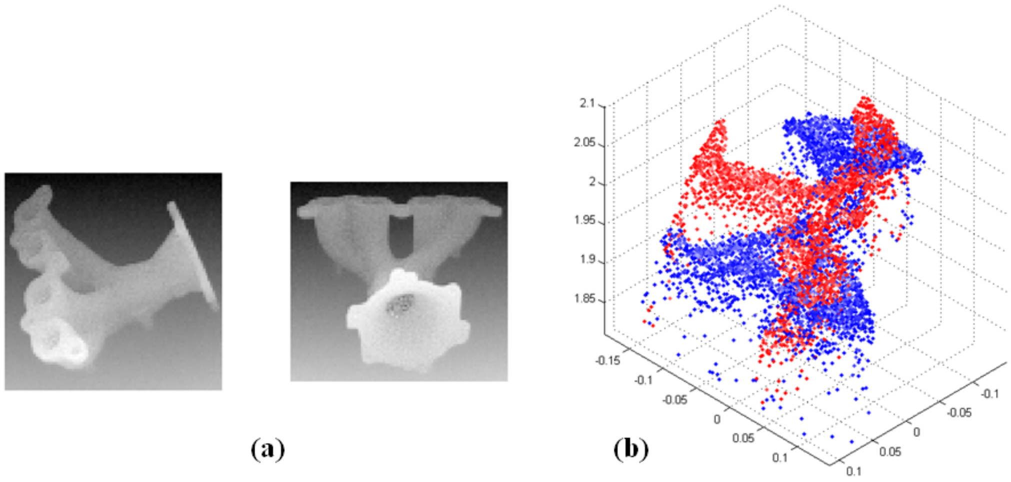


Figure 10.8: (a) Range images acquired by the TOF Camera shown in Figure 10.7. (b) Initial misalignment between the two corresponding point clouds.

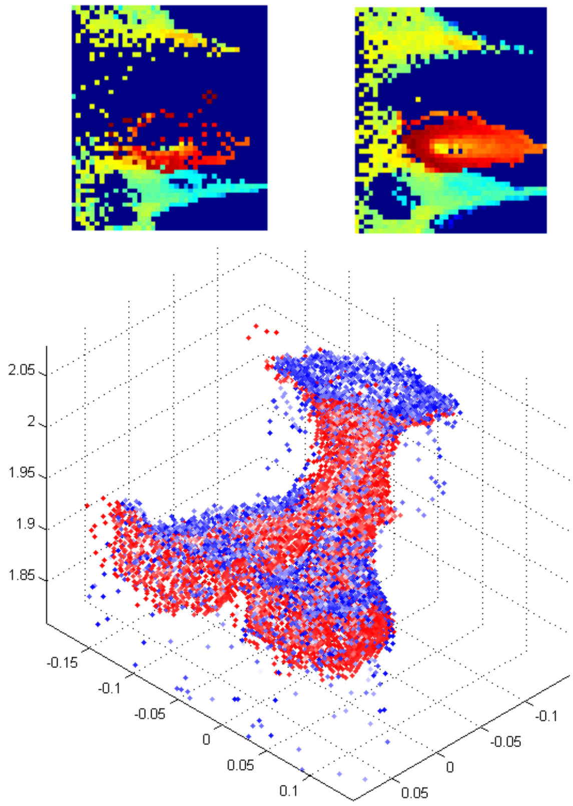


Figure 10.9: Similar CIRCON images that were found in the point clouds shown in Figure 10.8 (above). Coarse alignment obtained by the algorithm (below).

10.4 3D alignment using point clouds from different devices

The algorithm was also tested by aligning range images obtained from different devices: a TOF camera and a laser scanner. An example is shown in Figure 10.10. In spite of the low density and precision of the point cloud from the TOF camera, a quite acceptable alignment is obtained, since the algorithm finds a correspondence with two similar CIRCON images.

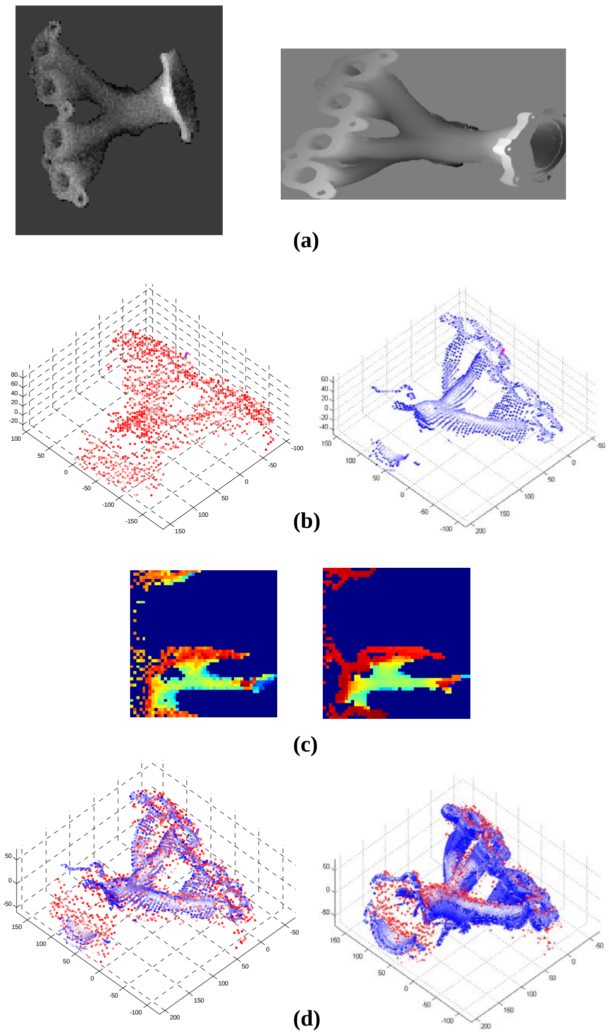
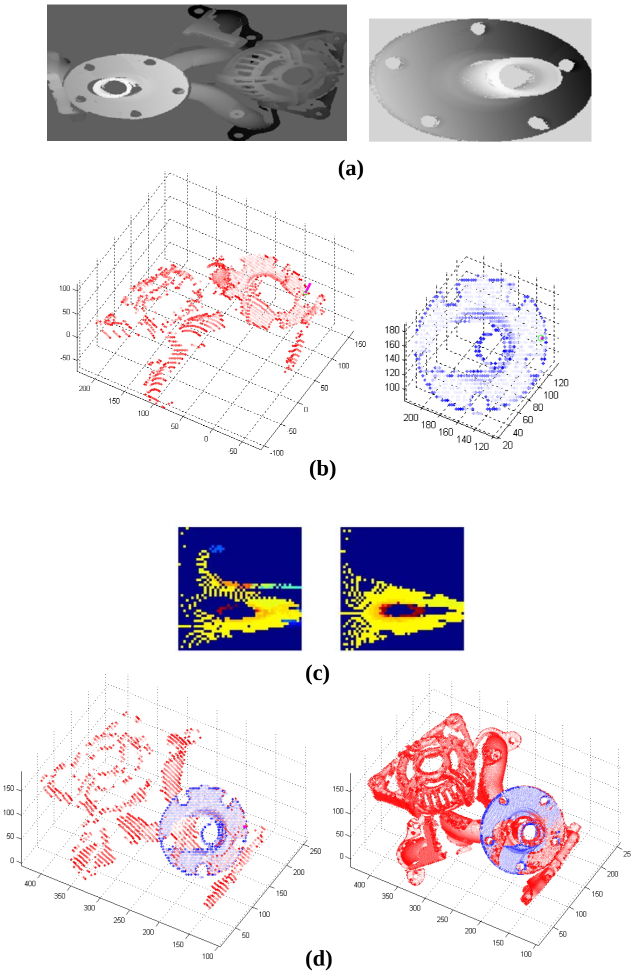


Figure 10.10: 3D alignment using range images from different devices: one from a TOF camera (left) and one from a triangulation laser scanner (right). **(a)** Range images. **(b)** Corresponding points (normal vector in magenta) that obtained the maximum similarity value. **(c)** CIRCON images associated with those points. **(d)** Alignment: reduced cloud (left) and 3D rendering using the original point clouds (right).

10.5 3D Alignment in cluttered scenes

Figure 10.11 shows how the coarse alignment algorithm can be used to determine the pose of an object in a cluttered scene. The only difference compared with the previous results is that the radius of the point-of-interest environment has been limited to the maximum radius that can be obtained for the object (in this example, this limit is fixed in 120 mm).



(a)

(b)

(c)

(d)

Figure 10.11: 3D alignment in cluttered scenes (point-of-interest environment radius equal to 120 mm). **(a)** Range images: scene (left) and object model (right). **(b)** Corresponding points (normal vector in magenta) that obtained the maximum similarity value. **(c)** CIRCON images associated with those points. **(d)** Alignment: reduced cloud (left) and 3D rendering using the original point clouds (right).

Chapter 11

Conclusions and Future Work

11.1 Conclusions

Within the problem of robotic object manipulation, we aimed at obtaining the position and orientation of the objects using the information provided by 3D acquisition devices such as laser scanners or time-of-flight cameras. If an object model is available, this problem can be reduced to finding a rigid transformation that aligns two point clouds (scene and model) whose relative positions and orientations are unknown. In addition, the point cloud alignment can be used to build 3D models even in complex object recognition tasks such as the '*Bin Picking Problem*'.

After reviewing the state of the art, the following problems were found in the most significant alignment algorithms: lack of generality (they work well with objects of a given topology), excessive computation time, problems with symmetries, misbehaviour when point clouds have low density and when they overlap each other in a small region, need for a method to discard false correspondences and for a valid correspondence group to obtain the Euclidean transformation. All the methods reviewed show, to a greater or lesser extent, at least two of these drawbacks.

The three characteristics to which we have given more 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 parts) and good behaviour when the overlap and the density of the point clouds are low. However, we have also taken into account the rest of the problems that may occur.

Considering the characteristic of generality, the algorithm based on spin images is perhaps the method that has the best performance. However, it does not work well with symmetrical objects or when some regions of the objects are repeated (due to an excessive locality). Moreover, this method (as well as others studied) needs to obtain a group of correspondences to compute a Euclidean transformation that roughly aligns the point clouds. Furthermore, since the process of finding correspondences does not suitably discard those that are false, it is necessary to carry out an evaluation process to establish correspondence groups that have geometric consistency.

To avoid the problems mentioned, we propose a descriptor which represents the local geometry around the points of interest selected from the clouds.

Unlike the spin images or 3D shape contexts (for some examples), the proposed descriptor does not use histograms, but the pixel values represent the height values of cells into which the neighbourhood of a point of interest is divided. This favours its use with point clouds of low density and low overlap, since the value of the pixels in the descriptor does not depend on the number of points per cell, but it is enough to have a single point in a cell so that its corresponding pixel can be assigned a value.

Moreover, unlike the local log-polar range images that use a logarithmic radial division, our descriptor uses a linear one, so the description of the point-of-interest environment will be more detailed and therefore, the similarity between descriptors can be better evaluated. Furthermore, we took advantage of its matrix structure and its geometric properties to design a similarity measure based on distances between its pixels, since they are closely related to the distance between the points of the cloud. In addition to allowing the classification of correspondences based on distance and on the degree of overlap of the descriptors, the evaluation of similarity measure will provide extra information. Instead of trying to make the descriptor invariant to rotation, we take advantage of its property of being cyclical to determine the row shift that produces the highest similarity and then, to calculate a rotation matrix around the normal of the point of interest. This enables the calculation of a Euclidean transformation matrix using only one correspondence.

Furthermore, having designed a similarity measure that exploits the specific characteristics of the descriptor and takes into account the percentage of overlap (and not the absolute number of the overlapping pixels, as in the algorithm based on spin images), the radius of the point-of-interest environment (and thus the discriminant power) can be increased and thus the problems with symmetries and repeated regions can be avoided.

Another important advantage is that our algorithm can finish at any time, it does not need to necessarily evaluate all combinations between the points of interest of both clouds. When a correspondence has a similarity value that exceeds the maximum value found until that moment, we evaluate the stopping criterion. This criterion is based on two measures of distance (one for rotation and one for translation) between the Euclidean transformations associated with the current correspondence and with a fictitious correspondence created from the current one and two others that are obtained from two corresponding cells of the distribution around the point found by the algorithm. In other words, the proposed algorithm does not need to carry out an evaluation process of the correspondences in order to create geometrically consistent groups.

On the other hand, the results show that the similarity measure supports low densities when the overlap is high and that, when this decreases, the point cloud density must be increased to maintain the rotation and translation errors within the predetermined limits. It has also been observed that the algorithm performs reasonably well with objects of different topology, in conditions of low overlap and low density of the point cloud, besides being able to align point clouds from different devices and different densities. However, when the overlap is very low (less than 20%) the alignment algorithm tends to select false correspondences; the same occurs when the density of the point clouds is very low (in this case, the probability of obtaining a good alignment is more dependent on the object topology).

In order to accelerate the search for correspondences we also proposed a version that uses a multilevel wavelet decomposition of the CIRCON image at the nearest point to the centroid of each cloud. This algorithm uses the simplified 3D shape of point cloud 1 (which can be obtained from the wavelet approximations of that *basic descriptor*) to build new descriptors and evaluate their correspondence with the *basic descriptor* of point cloud 2. Although this significantly reduces the execution time of the algorithm (since it uses point clouds of very low density), it is also true that their success depends largely on the object topology. As all evaluated descriptors are built up from the *basic descriptor* information, the normal vector at a large number of points in the cloud can not be very different from that of the point of interest where that *basic descriptor* was generated. For that reason, this simplified version of the original algorithm should only be used when objects meet these requirements because, otherwise, the possibility of obtaining a good alignment is significantly reduced.

11.2 Contributions

We consider that this thesis makes the following contributions to 3D point cloud alignment:

- We have proposed a new 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 divided 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.

- We designed a similarity measure that takes into account both the distances between the pixels of the descriptors as their degree of overlap. It also 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. Since the descriptors can be compared without having to restrict the neighbourhood of the point of interest, the discriminant power could be increased in order to avoid problems of misalignment when the objects have symmetries or repeated regions.

- With the descriptor and the similarity measure proposed a Euclidean transformation matrix can be calculated by determining one single correspondence.

- We have designed a coarse alignment algorithm that combines the descriptor and similarity measure proposed in order to classify the correspondences according to their similarity value and thus reduce the occurrence of false matches. One of the main advantages of this algorithm is that it has a chance to finish when it finds a correspondence that exceeds the maximum similarity value found until that moment.

- 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 the cells in which the points are distributed to create a fictitious correspondence with which another transformation matrix is computed. Then this matrix is compared with that obtained by the algorithm. The execution is stopped if a distance measure for the rotation and another for the translation do not exceed their respective thresholds.

- We have designed an algorithm based on wavelet decomposition of the descriptors and the use of the simplified 3D shape to build new descriptors with which to evaluate the correspondences. This simplified algorithm is recommended only for the alignment of objects that have a predominant direction of the normal vectors.

- We have proposed a vision system based on CIRCON descriptor that uses active laser triangulation to reconstruct radial contours of objects. We also propose a method to adapt this vision system for mounting on a robotic manipulator.

- We propose an adaptation of the ICP algorithm that establishes new correspondences between the point clouds by using the cell distributions around the points of interest that obtained the highest similarity value in the alignment algorithm. These correspondences allow an iterative refinement of the Euclidean transformation obtained by the coarse alignment algorithm.

11.3 Future work

Although the main objectives we set ourselves in this thesis have been covered, there are still several aspects that could be improved. Thus, we consider that the future work line should seek to address the following issues:

- Although a method of selecting the similarity measure parameters is proposed, we believe that the algorithm's effectiveness could be increased if we select these parameters using a genetic algorithm that finds the optimal pair to correctly classify the correspondences between point clouds from different object topology and with varying degrees of overlap.

- 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 at the time 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.

- Other possible improvements could address to the reduction of point clouds. The simple reduction algorithm that we use shows, according to the results, that our algorithm is robust enough to find a suitable coarse alignment. However, it might be interesting to use a reduction algorithm that preserves more information from the original point cloud.

- We can also pay attention to the implementation. The algorithm has been implemented in *Matlab*®, so its efficiency could be improved if the algorithm were programmed in another high-level language.

- We will use our experience in pose determination with ultrasound vision [23, 24, 33] to extend the algorithm to the alignment of ultrasound data and range images from laser scanners and TOF cameras.

- As already mentioned, besides being used to determine the position and orientation of objects given their corresponding model, the alignment algorithm could also be used to recognize objects in complex situations in which they are stacked and mixed with other different objects ('*Bin picking problem*'). This will be the preferential future work line due to its great difficulty and challenge.

Bibliography

[1] K. Brunnström. A. J. Stoddart. Free-form surface matching using mean field theory. In *In British Machine Vision Conference*, pages 33–42, 1996.

[2] A. P. Ashbrook, R. B. Fisher, C. Robertson, and N. Werghi. Aligning arbitrary surfaces using pairwise geometric histograms. In *Proc. NMBIA98*, pages 103–108, 1998.

[3] A. P. Ashbrook, R. B. Fisher, C. Robertson, and N. Werghi. Finding surface correspondence for object recognition and registration using pairwise geometric histograms. In *in Computer Vision-ECCV'98*, pages 674–686, 1998.

[4] Gill Barequet and Micha Sharir. Partial surface matching by using directed footprints. In *In Proc. 12th Annual Symp. on Computational Geometry*, pages 409–410, 1996.

[5] P. J. Besl and H. D. McKay. A method for registration of 3-d shapes. *IEEE Pattern Analysis and Machine Intelligence*, 14(2):239–256, Feb. 1992.

[6] D. L. Hill C. Studholme and Y. Bizais D. J. Hawkes. Multiresolution voxel similarity measures for mr-pet registration. In *Proc. Information Processing in Medical Imaging*, pages 287–298, 1995.

[7] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image Vision Comput.*, 10(3):145–155, 1992.

[8] J.C. Cheng and H.S. Don. A graph matching approach to 3-d point correspondences. *Pattern Recognition and Artificial Intelligence*, 5:399–412, 1991.

[9] Chin S. Chua and Ray Jarvis. 3d free-form surface registration and object recognition. *International Journal of Computer Vision*, 17(1):77–99, January 1996.

[10] Chin S. Chua and Ray Jarvis. Point signatures: A new representation for 3d object recognition. *International Journal of Computer Vision*, 25(1):63–85, October 1997.

[11] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, and G. Marchal. Automated multi-modality image registration based on information theory. *Information Processing in Medical Imaging*, pages 263–274, 1995.

[12] Jacques Feldmar and Nicholas Ayache. Rigid, affine and locally affine registration of free-form surfaces. *Int. J. Comput. Vision*, 18(2):99–119, 1996.

[13] Natasha Gelfand, Niloy J. Mitra, Leonidas J. Guibas, and Helmut Pottmann. Robust global registration. In *Symposium on Geometry Processing*, 2005.

[14] K. Higuchi, Martial Hebert, and Katsushi Ikeuchi. Building 3d models from unregistered range images. In *Proceedings of IEEE Conference on Robotics and Automation (ICRA '94)*, volume 3, pages 2248 – 2253, May 1994.

[15] Daniel F. Huber and Martial Hebert. Fully automatic registration of multiple 3d data sets. *Image and Vision Computing*, 21:637–650, 2001.

[16] A. E. Johnson and M. Hebert. Surface registration by matching oriented points. In *Proc. International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, pages 121–128, 12–15 May 1997.

[17] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Pattern Analysis and Machine Intelligence*, 21(5):433–449, May 1999.

[18] A. Stoddart. K. Brunnstrom. Genetic algorithms for free-form surface matching. In *Proceedings of the International Conference on Pattern Recognition (ICPR '96) Volume IV-Volume 7472, p.689, August 25-29, 1996*, pages 689–693, 1996.

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

[20] Shankar Krishnan, Pei Yean Lee, John B. Moore, and Suresh Venkatasubramanian. Global registration of multiple 3d point sets via optimization-on-a-manifold. In *SGP '05: Proceedings of the third Eurographics symposium on Geometry processing*, pages 187–196, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association.

[21] Pavel Krsek, Tomás Pajdla, Václav Hlavác, and Ralph Martin. Range image registration driven by a hierarchy of surfaces. In *22nd Workshop of the Austrian Association for Pattern Recognition*, pages 175–183, 1998.

[22] Marcel Körtgen, Marcin Novotni, and Reinhard Klein. 3d shape matching with 3d shape contexts. In *In The 7th Central European Seminar on Computer Graphics*, 2003.

[23] J. R. Llata, E. G. Sarabia, and J. P. Oria. Three-dimensional robotic vision using ultrasonic sensors. *J. Intell. Robotics Syst.*, 33(3):267–284, 2002.

[24] Jose R. Llata, Carlos Torre-Ferrero, Esther G. Sarabia, and Sandra Robla. Optical-ultrasound data processing for hybrid 3d vision on air. In *ASMCSS'09: Proceedings of the 3rd International Conference on Applied Mathematics, Simulation, Modelling, Circuits, Systems and Signals*, pages 181–186, Stevens Point, Wisconsin, USA, 2009. World Scientific and Engineering Academy and Society (WSEAS).

[25] Frederik Maes, André Collignon, Dirk V, Guy Marchal, and Paul Suetens. Multimodality image registration by maximization of mutual information. *IEEE transactions on Medical Imaging*, 16:187–198, 1997.

[26] S. G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.

[27] 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*, pages 216–223, 14–16 June 2006.

[28] C.R. Maurer, G.B. Aboutanos, B.M. Dawant, R.J. Maciunas, and J.M. Fitzpatrick. Registration of 3-d images using weighted geometrical features. 15(6):836–849, December 1996.

[29] B. M. Planitz, A. J. Maeder, and J. A. Williams. The correspondence framework for 3d surface matching algorithms. *Comput. Vis. Image Underst.*, 97(3):347–383, 2005.

[30] Kari Pulli. Multiview registration for large data sets. pages 160–168, Los Alamitos, CA, USA, 1999. IEEE Computer Society.

[31] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *Proceedings of the Third Intl. Conf. on 3D Digital Imaging and Modeling*, pages 145–152, 2001.

[32] Joaquim Salvi, Carles Matabosch, David Fofi, and Josep Forest. A review of recent range image registration methods with accuracy evaluation. *Image Vision Comput.*, 25(5):578–596, 2007.

[33] E. G. Sarabia, J. R. Llata, C. Torre, S. Robla, and J. P. Oria. Pose recognition by using neural networks and ultrasounds for a robotic picking up application. In *Proc. CIMSA Computational Intelligence for Measurement Systems and Applications 2005 IEEE International Conference on*, pages 128–133, 20–22 July 2005.

[34] G.C. Sharp, S.W. Lee, and D.K. Wehe. Multiview registration of 3d scenes by minimizing error between coordinate frames. *PAMI*, 26(8):1037–1050, August 2004.

[35] Chu song Chen, Yi ping Hung, and Jen bo Cheng. Ransac-based darces: A new approach to fast automatic registration of partially overlapping range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:1229–1234, 1999.

[36] Fridtjof Stein and Gérard Medioni. Structural indexing: Efficient 2d object recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(12):1198–1204, 1992.

[37] C. Studholme, D.L.G. Hill, and D.J. Hawkes. An overlap invariant entropy measure of 3d medical image alignment. *Pattern Recognition*, 32(1):71–86, January 1999.

[38] C. Torre Ferrero, J.R. Llata, S. Robla, and E.G. Sarabia. A similarity measure for 3d rigid registration of point clouds using image-based descriptors with low overlap. In *S3DV09, IEEE 12th International Conference on Computer Vision, ICCV Workshops 2009*, pages 71–78, 2009.

[39] C. Torre-Ferrero, S. Robla, E. G. Sarabia, and J. R. Llata. Circon: An alternative 3d shape representation for surface matching. In *Proc. IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics*, pages 3320–3323, 6–10 Nov. 2006.

[40] C. Torre-Ferrero, S. Robla, E. G. Sarabia, and J. R. Llata. 3d registration by using an alternative 3d shape representation. In *ISCGAV'07: Proceedings of the 7th WSEAS International Conference on Signal Processing, Computational Geometry & Artificial Vision*, pages 101–106, Stevens Point, Wisconsin, USA, 2007. World Scientific and Engineering Academy and Society (WSEAS).

[41] C. Torre-Ferrero, S. Robla, E. G. Sarabia, and J. R. Llata. A coarse-to-fine algorithm for 3d registration based on wavelet decomposition. *WTOS*, 7(7):655–664, 2008.

[42] William M. Wells-III, Paul Viola, Hideki Atsumi, Shin Nakajima, and Ron Kikinis. Multi-modal volume registration by maximization of mutual information. *Medical Image Analysis*, 1(1):35 – 51, 1996.

[43] Joris Vanden Wyngaerd, Reinhard Koch, Marc Proesmans, and Luc Van Gool. Invariant-based registration of surface patches. *IEEE International Conference on Computer Vision,*, 1:301, 1999.

[44] S. M. Yamany and A. A. Farag. Surface signatures: an orientation independent free-form surface representation scheme for the purpose of objects registration and matching. *IEEE Pattern Analysis and Machine Intelligence*, 24(8):1105–1120, Aug. 2002.

[45] Dongmei Zhang. Harmonic shape images: A 3-d free-form surface representation and its application in surface matching. In *Ph.D. dissertation, Carnegie Mellon Univ*, 1999.

[46] Timo Zinsser, Jochen Schmidt, and Heinrich Niemann. A refined icp algorithm for robust 3-d correspondence estimation. In *Proceedings of the International Conference on Image Processing*, pages 695–698, 2003.