Algorithms for the Multiple Variants of Registration in 3D Range Data

A dissertation submitted by **Xavier Mateo Prous** at Universitat Autonoma de Barcelona to fulfil the degree of **Doctor en Informática**.

Bellaterra, September 25, 2013

Director | **Dr. Xavier Binefa Valls**
Universitat Pompeu Fabra
Departament de Tecnologies de la Informació i les Comunicacions

Tutor | **Dr. Enric Mart´ Godia**
Universitat Autonoma de Barcelona
Departament de Ciencies de la Computació

This document was typeset by the author using LaTeX 2 .

The research described in this book was carried out at the Universitat Autonoma de Barcelona.

a Emi

Acknowledgements

Muchos son los que han tenido que sufrir mis historias sobre el doctorado durante estos años, animándome en esos momentos complicados y en los que parec´a que la cosa no tiraba para adelante; pero si estáis leyendo estas l´neas ya veis que parece que se ha conseguido.

Primer de tot voldria agrair en Xavier Binefa la oportunitat que em va donar per a fer aquest doctorat, una cosa que sempre havia volgut fer des de que vaig acabar la carrera pero que per unes histories o altres mai no es va presentar la oportunitat. Tot i portar uns quants anys apartat del món academic i tindre un perfil diferent a la resta dels companys del grup, va apostar per mi des del primer moment. Moltes gracies pel recolzament i pels consells oferts durant aquests anys.

Tot aixo tampoc hagués estat possible sense l Adria Pérez, el meu primer company de despatx i la persona que em va avisar de que hi havia una plaça disponible al seu grup d investigació. Després van venir nous companys de despatx com en Ferran, en Marc o en Polete, a més d altres que estaven als despatxos propers i a qui sempre podia visitar amb alguna excusa per desintoxicar-me una mica d equacions i de papers: Luis, Brais, Ciro, Oriol, Rafa, Luis Ruiz, Jota, Adria Ruiz, Yashin y Javier Vázquez, a més de les secretaries de la UPF. Dins d aquest grup també vull agrair la seva ajuda a en Xevi Orriols, que em va ajudar en la part més problematica del doctorat i que sense la seva ajuda potser no estaria escrivint aquestes l´nees.

També a professors i doctorands de la UAB, amb qui vaig poder compartir docencia a Sabadell durant aquests anys: Josep Lladós, Xavier Roca, Javier Orozco, David Fernández, Daniel Ponsa i David Gerónimo; aix´ com a Enric Mart´, tutor d aquesta tesis.

Unos agradecimientos no podr´an ser completos sin añadir en ellos a mi familia, en especial a mis padres que me han dado siempre todo su apoyo y siempre me preguntaban por cómo iba el doctorado. Como veis... ¡¡al final se ha logrado!!. También a mis t´os y primos, as´ como a mis hermanos Carlos y Nacho que desde pequeños, además de cariño y algunas peleas, me inculcaron la pasión por la tecnolog´a y la ingenier´a, junto con toda la familia que han ido creando durante estos años: Mónica, Teresa, Laura, Alba, Joan, Sof´a y Cristina, además de todos los peques que puedan ir llegando de aqu´ en adelante.

Entre esos peques estará, como no, mi futuro hijo, que si tanto sus plazos como los m´os se cumplen como están previstos llegará poco después de la presentación de esta tesis. Es-

peremos que no se adelante demasiado y quiera ver la presentación de su padre, aunque de momento seguro que se encuentra muy a gusto dentro de mi amada compañera Silvia, quien ya se ha ganado un trocito de cielo por aguantarme durante casi 10 años, más todos los que le esperan... Ella fue la persona gracias a quien, por cierto, descubr´ que exist´a un máster en visión por computador en la UAB y empezó toda esta historia del doctorado. Además de ella, también a mis nuevos familiares Marcelo, Loli y Alberto.

Antes de empezar este doctorado ya ten´a grandes amigos con los que por suerte he podido seguir con su amistad durante estos años. Me refiero a Paúl, Santi, Nachete, Joel, Jordi y Cristina Domingo, Óscar, Héctor, Jordi Aguilar, Jordi Llorens, V´ctor, Marc Aldea, Diego, Begoña, Jose, Lara, Natxo, Xavi Egozcue, Frank, Mofi, Gino, Charly, Sandra Ferrer, Ramon, Puchi, Carlitos, además de todas sus parejas. Seguramente olvidaré a alguien, pero si me conoce bien ya sabrá que soy un poco desastre para estas cosas y seguro que no me lo tiene en cuenta.

Por último, me gustar´a dedicar esta tesis a Emi, a quien conoc´ durante la carrera y con quien tan buenos veranos y sus famosas Fiestas Primavera he podido disfrutar. Se te echa mucho en falta por aqu´, aunque por lo menos podemos ver una parte de ti en tu pequeño Kilian, a quien vemos bastante a menudo junto con tu esposa Sandra. Apunta maneras, se nota que es hijo de su padre. Que sepas que ha sido un placer haberte conocido y disfrutar contigo durante todos estos años.

De hecho, ha sido un placer haberos conocido a todos y poder disfrutar durante muchos años más con vosotros.

Resum

Des del naixement de la fotografia hi ha hagut sempre un gran interes en la possibilitat de detectar una tercera dimensió en les imatges obtingudes per una camera. Aquesta tercera dimensió permetria la diferenciació i filtrat dels diferents objectes presents a una escena, i per tant facilitar molts del problemes actuals en la recerca de visió per computador. Per tal d aconseguir-la, diverses tecniques s han anat utilitzant a través del temps, des de la classica estereoscopia fins a altres metodes més actuals com el Structure from Motion.

Durant els darrers anys l evolució de la tecnologia ha fet possible l aparició de dispositius que permeten la captura directa d aquesta informació 3D sense la necessitat d una intervenció manual. L ús de diferents espectres de llum com el laser o la llum infraroja, aix´ com la seva integració en els dispositius, han possibilitat aquesta important millora, acompanyada al mateix temps d una reducció en el preu dels components que ha fet possible el seu ús per gran part de la comunitat de recerca.

Aquesta tesis esta enfocada en el principals problemes derivats de l ús de les anomenades cameres range que, a més de la imatge RGB usual, ofereixen una imatge addicional indicant la distancia als objectes de l escena. Gracies a l aplicació directa d aquestes imatges range, on cada p´xel correspon a una distancia, una recreació 3D de l escena capturada pot ser obtinguda facilment. Una de les seves principals avantatges és el fet de que, si la camera esta correctament calibrada, el 3D obtingut es troba expressat en unitat f´siques reals (per exemple, en metres) i no pas en p´xels.

Tot i aixo, l ús d aquestes cameres range no és tan ideal com es podria suposar. Normalment diverses captures d una escena o objecte són necessaries per tal d aconseguir una reconstrucció completa, i alguns materials poden produir problemes que interfereixen en el correcte posicionament dels objectes.

Les particularitats de la representació 3D obtinguda fan que aquesta sigui apropiada per fer-la servir com a suport per afegir-hi altres fonts d informació, com ara imatges RGB o imatges infraroges. L estructura 3D obtinguda pot ser texturitzada amb aquestes fonts d informació, donant un resultat integrat que pot ser molt útil per solucionar problemes que no serien possibles utilitzant les imatges de forma separada. Tot i aixo, diferencies en el procés d adquisició entre aquest tipus d imatges poden produir alguns problemes quan són fusionades.

A més, per tal d obtenir una reconstrucció 3D completa d una escena, normalment és necessari que la captura s hagi fet des de múltiples punts de vista diferents. L alineament de totes aquestes estructures 3D obtingudes és conegut com registració multivista, on és necessari identificar la posició i orientació de les cameres range en cadascuna de les presses per tal de poder alinear-les correctament. Aquest alineament s aconsegueix normalment fen servir dos passos diferenciats: la registració de imatges rang parell a parell, i la posterior minimització de l error considerant simultaniament totes aquestes parelles.

Abstract

Ever since the photography was born, there exists a high interest in the possibility of detecting a third dimension in the images obtained by a camera. This third dimension feature would allow the differentiation and easily filtering of the different objects present in the scene, and therefore to facilitate some of the main problematics in the computer vision research. In order to achieve this third dimension acquisition some techniques were historically applied, starting by the classical stereoscopy or other more current methods like Structure from Motion.

During last years, the evolution of the technology has made possible the appearance of devices which allows the direct retrieval of 3D information without the manual intervention of the user. The use of different light spectrums like laser or infrared light and their integration inside the camera case have allowed this important improvement, accompanied at the same time by a reduction of the components price which allows its use for the vast majority of the research community.

This thesis focuses on the main problems obtained in the use of the so-called range cameras, which, in addition to the usual RGB image, offers an additional image indicating the distance with respect to the objects in the scene. Thanks to the direct application of these range images, where each pixel corresponds to a distance, a 3D recreation of the observed scene can be directly obtained. One of the main advantages is that, if the camera is correctly calibrated, the 3D structure can be obtained with physical units of the real world (such as meters), and not with pixels.

Nevertheless, the use of these range cameras was not as ideal as supposed. Usually some captures of every object in the scene are needed in order to obtain a full reconstruction, and different materials could produce problems that interferes the correct position of the object. The presence of these inconveniences produce the necessity of using some algorithms to produce a correct final 3D structure.

The particularities of the 3D representation created from the range image become it appropriate to use as a support plate for placing other sources of information, like visible images or infrared images. The obtained 3D structure can be textured with these sources of information, giving an integrated result which could clarify some problems that can not be solved by using the images separately. However, differences in the acquisition process between these types of images produce difficulties when they are fused.

In addition, in order to obtain a full-side representation of a scene usually some 3D captures from different points of views are required. This addresses to the so-called multiview registration problem, where it is necessary to identify the position and orientation of the range camera for each viewpoint in order to correctly join the corresponding 3D structures. Current technology devices like GPSs or IMUs could give this information, but usually is not accurate enough, so common visual elements between different range images must be detected in order to align them. This alignment is usually achieved by using a two-steps procedure: the registration of pairs of range images between them, and the posterior minimization of the global error for the whole set of images.

Contents

List of Tables

List of Figures

Chapter 1

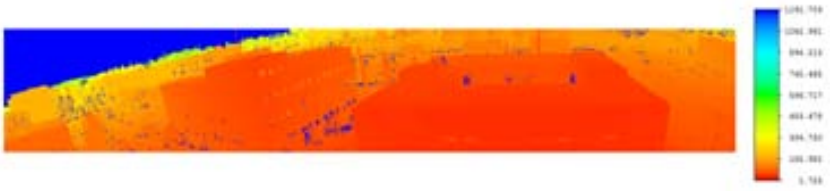
Introduction

During last years the range image analysis has focused a high attention in the research community and also in other sectors like videogaming, architecture, and geology. The possibility of acquiring an additional source of information regarding the distance to the objects allows new possibilities for the computer vision research, but also suffer from other drawbacks that should be taken into account. The research literature about this topic is continuously growing, and it is expected to be enlarged in the following years as the corresponding technology keeps improving the device possibilities and the technology increasingly arrives to the mass public. Despite this attention, some work must still be done in order to achieve a level of knowledge as high as the classical 2D image analysis.

The different possibilities for the registration of range images are the basis of this PhD thesis. Existing algorithms are analyzed in the different sections of the document, and novel methods are proposed in order to avoid the possible inconveniences in the treatment of this particular images. Some of these methods can be upgraded from the traditional 2D image methods, but it must be considered that the particularities of the range images require a special treatment and a different point of view for their processing.

1.1 From range image to 3D point cloud

The main particularity of the range images is their direct equivalence to a 3D point cloud, but with the advantage of being represented by a traditional 2D image and therefore with all the advantages of processing and transmission for a typical image. A range image is, as usually, composed by a number of pixels. What makes special a range image is that each pixel specifies, using a colormap, the distance between the sensor and a specific point in the scene. As can be seen in the range image shown in Figure 1.1(a), the distance to each point of the scene is encoded with a different color so, assuming that the scanner is correctly calibrated, the 3D point cloud corresponding to the range image can be easily obtained (using, for example, the concept of the pinhole camera [25]). The resulting 3D point cloud obtained can be seen in Figure 1.1(b).



(a)

(b)

Figure 1.1: (a) Example of range image. Each color of the range image specifies the distance of this point with respect to the 3D scanner. (b) Resulting 3D point cloud after the conversion.

Obtaining a 3D point cloud from a range image allows two important features which will be used along this PhD thesis. The first one is the particularity that the 3D point cloud delivers the information in the standard metric system, that is, we can extract the information between distances in meters instead of using image pixels and, therefore, we can avoid the usual problems in 2D imaging with respect to the distance of the objects and the scale factor.

In addition, the obtained 3D point cloud offers an untextured structure which can be textured with other sources of information (like visible image, infrared image or reflectance image, among others), making possible the visualization of multiple types of information simultaneously. These possibilities will be discussed in Chapter 2 of this thesis.

1.2 Range image acquisition techniques

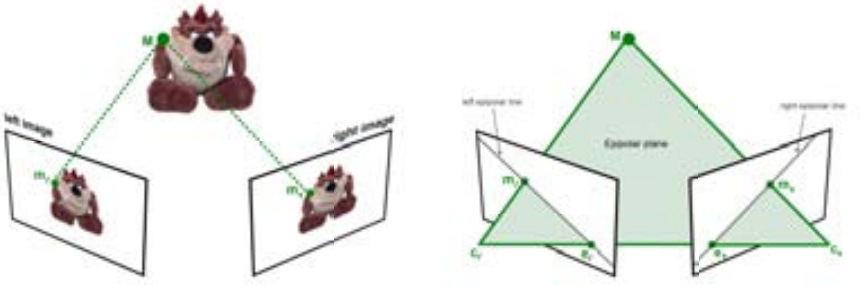
Different technologies have allowed the improvement of the range imagery analysis in the last years. From the traditional methods to the current systems, their basic purpose was to obtain the 3D structure of the scene with the minimum intervention of the user. In addition, some of these techniques are nowadays integrated in commercial devices and software, making even easier the process of the acquisition. In the following subsections the most usual techniques among the high number of possibilities are briefly introduced, explaining their basic operation and trying to group them into different categories. As it will be seen along this

thesis, different possibilities for the processing of a range image will be available depending on the particularities of its capture technique.

1.2.1 Multiple 2D images

The most usual technique in order to obtain the 3D structure of objects or scenes is the use of multiple 2D images. In fact, this can be considered as a family of techniques due to the high number of possibilities that exist in the current state of the art.

One of the most used techniques inside this category is the classical stereoscopy, where two 2D cameras separated by a predefined distance aim to a similar position, as shown in Figure 1.2. This method tries to emulate the vision of humans and some animals, where two eyes observe the scene and the subject is able to guess the distance of the objects. The geometric relationship established between these two cameras is known as epipolar geometry, and describes all the elements needed in order to obtain the 3D coordinate for each point in the scene as explained in [25].



(a) (b)

Figure 1.2: In (a) the same scene is captured simultaneously by 2 cameras, denoted as C_l and C_r. Taking as example the point M, it is possible to estimate its position by using the geometry shown in (b), known as epipolar geometry.

Using similar concepts to the stereoscopy but extending the number of views we can find powerful techniques like the classical bundle adjustment [59], which iteratively estimates the position of multiple 3D points in the scene as well as the parameters of the multiple cameras used to capture them. The main drawback of this algorithm is the required processing time, because of its condition as iterative algorithm and the high number of elements to be estimated. Despite this, bundle adjustment is frequently used in the computer vision topic of Structure from Motion, specially as the last refinement step.

Also inside the category of multiple 2D images we can consider the technique of defocusing [42], which uses a set of 2D images acquired under varying focus settings. As the previously explained techniques, depth from defocus also uses a set of 2D images from the scene but in this case the images are obtained from the same viewpoint. The simplified operation of this technique is depicted in Figure 1.3, where a camera model with two different focus settings are presented: a first case where an specific point P of the scene is unfocused

and a second case where the same point is correctly focused. Considering this second case, displayed in Figure 1.3(b), the following equation can be applied:

$$\frac{1}{d} + \frac{1}{d_i} = \frac{1}{f} \tag{1.1}$$

Equation (1.1) is known as Gaussian lens law and asserts that, assuming that the internal camera parameters of focal distance f and distance of the image plane d_i are known, it is possible to estimate the distance d of the exterior point P. Using this principle, and changing the focus parameter of the camera in order to focus different parts of the scene it is possible to estimate the depths of different parts of an object and therefore be able to estimate its 3D point cloud.

One of the main advantages of the depth from defocus technique is the non-necessity of detecting correspondences between the images, since they are perfectly aligned because they are obtained from the same viewpoint. On the other side, characteristics of the commercial lenses available nowadays, make the depth from defocus technique appropriate only for small distances and therefore only range images from small objects could be correctly obtained.

1.2.2 Time-of-Flight scanners

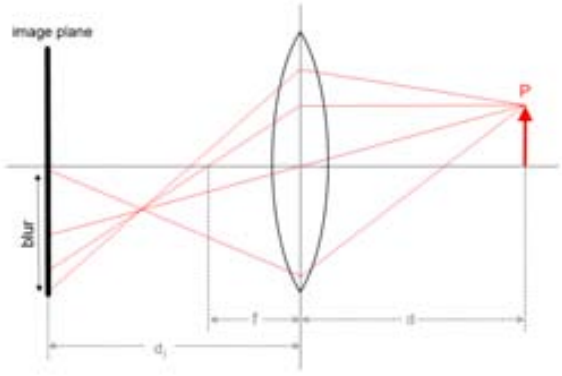
In opposition to the previously explained methods, current technologies for the 3D scene capture are based on active systems, i.e., they emit energy in some different forms in order to find out the distance of the objects with respect to the 3D scanner. Usual devices used nowadays are the scanners with time-of-flight sensors [31] [24]. The basic idea behind this technology is to estimate the distance of the objects thanks to the time delay of a signal emitted by the scanner which bounces in the selected object and returns to the sensor. The emitted signal can be a set of pulses, but nowadays a continuous wave signal is generally used, as the example displayed in Figure 1.4. In case of using a wave signal the distance to the object is estimated by using the phase difference Δ between the emitted and the received wave, using the following expression

$$Depth = \frac{c}{2}\frac{\Delta}{2\Pi f} \tag{1.2}$$

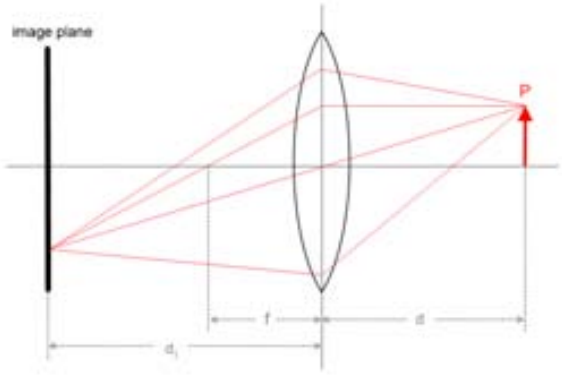
, where c corresponds to the light speed (300.000 km/s).

Time-of-flight scanners can use different types of sources in order to send the signal. The most used types are laser light or infrared light proceeding from a set of LEDs. The main particularity for this decision is, in addition to the price, the maximum distance which can be detected by the scanner. In order to avoid uncertainties in the distance estimation, the maximum distance which can be detected will be in case where $\Delta = 2\Pi$, so

$$Depth_{max} = \frac{c}{2 \cdot f} \tag{1.3}$$



(a)



(b)

Figure 1.3: The convex lens refract all the light rays coming from a point, concentrating them in an specific internal point. The refraction in the lens has two main particularities: horizontal light rays that are refracted by the lens are directed through an internal point which are placed at a distance equal to the focal distance f, and light rays that are passed though the center of the lens are not affected in their direction. In (a) the concentrating point is not placed near the image plane, so the image of the point P will be blurred. In (b), after moving the position of the lens (and therefore changing the focal distance), the concentrating point coincides with the image plane, so the image of the point will be correctly focused.

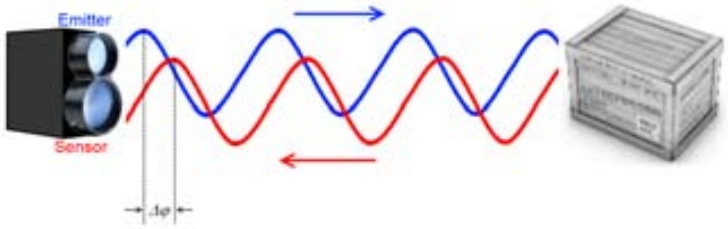


Figure 1.4: Principle of time-of-flight scanners. The emitted blue wave is reflected in the object and returns to the scanner. The sensor detects the reflected red wave and the distance to the object is estimated thanks to the phase difference.

As can be seen the maximum distance of the device depends on the frequency of the emitted signal, so, the higher the wavelength, the higher the distance which can be covered. Assuming a laser 3D scanner with a pulse frequency of 200 KHz, the maximum distance which could be covered by this time-of-flight scanner will be

$$Depth_{max} = \frac{300\ 000\ 000m\ \ s}{2 \cdot 200\ 000s^{-1}} = 750meters \tag{1.4}$$

The high accuracy achieved by the time-of-flight sensors based on laser light, in addition to the high distances allowed (in some commercial products these maximum distances can be higher than 5 kilometers), make them really appropriate for sectors like geology, archeology and architecture, as pointed in [63] and [6].

1.2.3 Structured light

With the structured light technique a pre-defined pattern is projected to the scene, illuminating all the objects in the field of view. At the same time, a sensor captures the different pattern points or lines illuminating the scene, and thanks to the modification of the pattern it is able to estimate the distance to all the objects. The pattern could have a high variability of shapes, from a static set of points, a static set of lines, or also a moving pattern covering all the scanned object. In addition, also the nature of the light can belong to different sources, having the possibility of using visible light, infrared light or other kind of sources.

The main advantage of this technology is the low price in comparison to the time-of-flight technology, due to the fact that it is not necessary to have an ordered matrix of laser or infrared sources, only with one source should be enough. The accuracy of the resulting 3D point cloud is usually not as accurate as with the time-of-flight sensors, and also the maximum distance for the objects is lower, but it is an appropriate technology for some cases without spending a high price. An example of scanning with structured light can be seen in Figure 1.5.

A well-known example of structured light scanner is the Microsoft Kinect device, which

Figure 1.5: Pattern of structured light composed by a static set of lines. The shape produced by the illumination in the objects make possible the estimation of their 3D point cloud.

floods the scene with a pattern of infrared points as can be seen in Figure 1.6. As previously stated, thanks to the modification of this point pattern, the Microsoft Kinect device is able to guess the distance to the objects in the scene.

Figure 1.6: Infrared pattern projected by the Microsoft Kinect device.

1.3 Outline of this thesis

This PhD thesis covers different variants for the registration of range images, accordingly separated with the different chapters of the document. The introduction explained so far does not include some general aspects related to the range images, as they are more related with some of the processes explained in the following chapters. For this reason, some state-of-the-art approaches and classical methods are accordingly explained in their respective chapters.

The outline of this thesis document can be separated into three main parts: the first

one includes the integration of range images with other types of information, producing a final 3D structure which includes all the information from multiple sources at the same time. The second part, including Chapter 3 and Chapter 4, deals with the registration of pairs of range images, producing the displacement of one of these images in order to perfectly fit with the other one. The multiple pairwise registrations achieved serve as basis for the so-called multiview registration, which minimize the global error of the final registration of all the range images at the same time. Finally, in the third part, an specific system for the 3D modeling using range scanners is presented.

Chapter 2

Multisensorial registration

In this chapter we tackle the problem of joining information proceeding from different data sources, like visible imagery, infrared imagery or thermal imagery. We start with a discussion of the nature of the camera intrinsic parameters and the need of a calibration in order to fully categorize all the sensors involved in the process. Once all the sensors are correctly calibrated, it is necessary to guess the relationship between them. The relative positions of the sensors establish their contribution in the final multisensorial registration, using the point cloud obtained from the 3D scanner as support plate for the other sources. The use of estimation algorithms for searching the relative position and orientation is analyzed, and the experimental results show the good adaptation of the estimation to changing configurations.

2.1 Introduction

In this second chapter we study the methods used to achieve the so-called multisensorial registration, i.e., the fusion of the data coming from multiple devices at the same time. With the multisensorial registration we can achieve a 3D representation of the scene textured with the information of another sources, achieving a more complete and realistic result. This step is quite straightforward and simple to implement, but its results are crucial in the posterior steps which will be explained in the following chapters. An example of final result of the multisensorial registration, joining 3D and visible information, can be seen in Figure 2.1, which corresponds to the same captured scene already shown in Figure 1.1.

In order to achieve the multisensorial registration two different steps are needed. The first one consists in the estimation of the intrinsic parameters for each one of the devices involved in the process. In the second step the so-called extrinsic parameters are also estimated. The extrinsic parameters represent the relationship between the devices involved in the process,

Figure 2.1: 3D point cloud textured with the information of a visible camera.

and include the relative positions and orientations between them.

The obtaining of both intrinsic and extrinsic parameters is an essential element in most of registration processes or georeferentiation processes. Their result depends in a high degree in the nature and the characteristics of the devices involved in the capture process. In Figure 2.2 two different cases composed of different devices can be seen, one with a platform including a 3D scanner, a visible camera and an IR camera, and a second one with a Microsoft Kinect device, which integrate in its fabrication design an RGB camera and an infrared camera which is used for the 3D capture.

2.2 Intrinsic parameters

Usually all the camera manufacturers give in their specifications the information about the camera. These specifications are in a higher part belonging to the camera optics: focal distance, CCD size, aperture of the iris, etc. However, small differences in the fabrication process or the incorrect maintenance of the device can produce that two cameras belonging to the same model have small differences in these values. The calibration consists in the obtaining of these value by an empirically process, using a set of algorithms which will allow us to adjust the camera specifications.

In order to obtain the multisensorial registration, all the cameras which are used in the fusion process should be previously calibrated to obtain their optical specifications. The main parameters obtained in the calibration process are stored in the calibration matrix K. This matrix defines the parameters which specify an ideal pinhole camera, and its format is the following:

$$K = \begin{bmatrix} \frac{f \cdot res_x}{CCD_x} & \frac{f \cdot res_x}{CCD_x} \cdot \cot(\Theta) & pp_x \\ 0 & \frac{f \cdot res_y}{CCD_y} & pp_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

(a) (b)

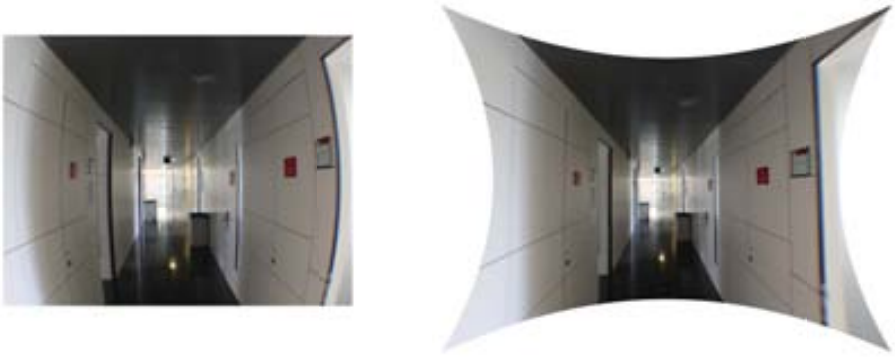
Figure 2.2: a)Platform of integration for a LADAR sensor, an RGB camera and an infrared camera. b) Microsoft Kinect device, including an RGB camera and an infrared camera, which is used to estimate the depth of the scene.

, where f is the focal distance, $(res_x\ res_y)$ are the horizontal and vertical resolution of the camera, $(CCD_x\ CCD_y)$ are the horizontal and vertical size of the CCD sensor, $(pp_x\ pp_y)$ are the coordinates of the principal point, and Θ is the angle between X and Y axes.

A parameter which does not appear in the matrix K, due to the fact that this is the parameter which produces that the camera does not represent an ideal pinhole camera, is the distortion. The distortion us usually produced by the imperfections of the camera optics, specially for the cases of very small focal distances. The distortion is the reason, among others, of some cases where straight lines appear as curved, specially in the periphery zones of the image. An example can be seen in Figure 2.3.

There exist two different types of distortion: the tangential distortion, which is produced in the center of the image; and the radial distortion, which is produced as we are separating from the center of the image. Usually the tangential distortion can be discarded, because modern camera offer a good behavior in the central zone of the image. With respect to the radial distortion, it is usually expressed as a vector of coefficients k_c, and is getting higher as long as we are getting further from the center of the image:

$$\begin{array}{l} distortion_x = (x_c - x)(k_1 r + k_2 r^2 + k_3 r^3 + \quad) \\ distortion_y = (y_c - y)(k_1 r + k_2 r^2 + k_3 r^3 + \quad) \end{array} \tag{2.2}$$



(a) (b)

Figure 2.3: Original image and image with distortion compensated. As can be seen in (b), in the periphery zones of the compensated image the walls remains straight.

, where (x, y) are the coordinates of the point where we want to evaluate the distortion, (x_c, y_c) is the center of the radial distortion, and $r = \frac{\sqrt{(x-x_c)^2+(y-y_c)^2}}{r_{max}}$, where r_{max} is equal to $\sqrt{x_c^2 + y_c^2}$.

The algorithm that we will use to estimate the intrinsic parameters of the cameras will be the one developed by Zhang in [68], where a pattern similar to a chessboard is captured from different viewpoints and positions. The transition points between black and white parts of the pattern are detected and, assuming that the plane defined from the pattern is assigned a value of $Z = 0$, the algorithm is able to estimate the intrinsic parameters of the camera thanks to the concept of homography between planes. Firstly only the parameters of the calibration matrix K are estimated (assuming that we are dealing with an ideal pinhole camera) and in a second step, once the parameters have been fixed, the distortion is estimated thanks to an iterative algorithm based in the maximum likelihood criteria.

2.3 Extrinsic parameters

As previously explained the so-called extrinsic parameters include the position and orientation of each device in the 3D space. As can be seen in Figure 2.4, the 3D scanner and the visible camera are separated at a defined distance and have different orientations. This implies that each device captures the scene using its own coordinate system, and therefore the images obtained from each device will be expressed in different coordinate systems. The difference of position between both devices is small, and it could be considered as zero if we are capturing objects at a high distance, but the difference of orientation it is more evident and can affect notably to the fusion process.

In terms of mathematics, the displacement and orientation differences between the 3D scanner and the camera can be modeled as a rotation matrix R and a translation vector t.

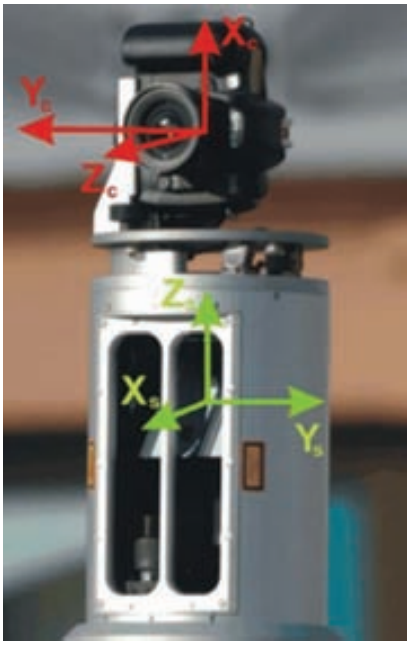


Figure 2.4: Coordinate system for the 3D scanner and for the visible camera.

Graphically speaking, and as can be seen in Figure 2.5, R would correspond to a 3×3 orthonormal matrix which specifies the rotation needed along the 3 axis in order to align the 3D scanner coordinate system and the camera coordinate system, and t would correspond to the tridimensional vector which indicates the 3 components of displacement between the origins of the coordinate systems.

A possible procedure in order to estimate the rotation and translation needed would be doing it manually, measuring the distances between the origin of the sensors and their orientations. Obviously, this method is not the most appropriate one and will obtain inaccurate results. In addition, it must be taken into account that probably it will be necessary to repeat

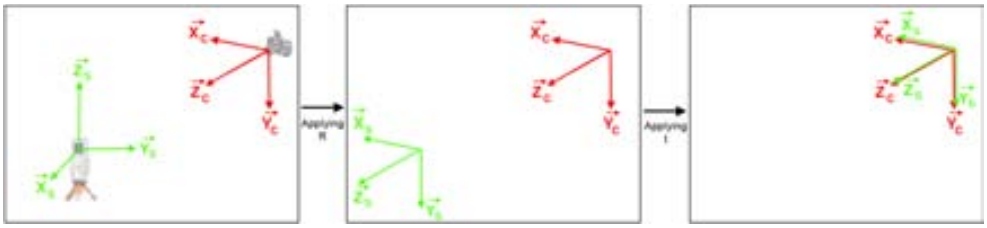


Figure 2.5: Transformation of the 3D scanner coordinate system to the camera coordinate system. A rotation and a translation are needed in order to match both coordinate systems.

this calibration regularly after specific time, because the physical disposition between the 3D scanner and the cameras can change occasionally due different factors like transportation or small vibrations. For these reasons it is necessary the availability of an automatic or semi-automatic method, which allows the obtaining of a reliable calibration and could be repeated regularly without big efforts. At this point appear the so-called pose estimation algorithms. These mathematic algorithms allow the estimation of the position and orientation of an object given a set of correspondences between pixels of an image and 3D coordinates of the scene. In other words, if we are able to known the 3D coordinates of some pixels in an image, these algorithms are able to find out the orientation and the position of the camera where the image has been taken.

Expressed in a mathematical form, and taking into account the particularities of the pin-hole camera, the pose estimation algorithms try to estimate R and t by solving an equation system like the following:

$$l_i \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = K \quad R \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} + t \tag{2.3}$$

, where $i = 1 \quad N$ represents the different correspondences between the image pixels $\begin{bmatrix} x_i \\ y_i \end{bmatrix}$ and the 3D coordinates $\begin{matrix} X_i \\ Y_i \\ Z_i \end{matrix}$. With respect to the rest of the parameters, K corresponds to the intrinsic parameters (as it has been defined in Equation (2.1)) and l_i represents the projection depth for each pixel. If we observe the equation system, it is possible to see that it contains N equations with a total of $N + 6$ unknown variables (N variables of l_i, 3 variables of t and 3 variables of R), so an specific method for its solving will be needed.

Traditionally, the problem of the pose estimation was solved by using photogrammetry techniques like the Direct Linear Transformation (DLT) [1]. The DLT method generate two lineally independent equations for each point correspondence, giving a over-dimensioned equation system which can be easily solved by using Singular Value Decomposition (SVD). In the 90 s decade DeMenthon developed his POSIT algorithm [12], which estimates a first approximation of the pose considering that the image was obtained by an orthographic projection instead of a perspective projection. In a second phase the algorithm repeats iteratively the same process, with the only difference that the image points are re-calculated considering that the image was taken from the pose obtained in the last iteration and assuming again an orthographic projection. A more recent approach was the one presented by Lu et al. [36], an iterative algorithm which successively improves the estimation of the rotation matrix, and thereafter estimates the translation vector. Each iteration tries to minimize the sum of the mean quadratic error between the original points of the image and the reconstructed points (points which will be observed in case that the image was really taken from the position obtained in the last iteration).

Another interesting algorithm is the one developed by Fiore [17]. Its main strategy was to develop some combinations of the equations obtained from (2.3), in order to cause that the rotation matrix and the translation vector could be discarded. The Fiore algorithm needs a minimum of 6 point correspondences between the 3D point cloud and the visible image, but

obviously better results can be obtained if we increase this number. Given N correspondences between pixels of the image and 3D coordinates, the Fiore algorithm can be summarized as [21]:

1. Express the Equation (2.3) in the following form:

$$K^{-1} l_i p_i = [R\ t]\, P_i \qquad i = 1 \qquad N \tag{2.4}$$

, or analogously

$$K^{-1}\,[l_1 p_1\ \ l_2 p_2 \qquad l_N p_n] = [R\ t]\,[P_1\ \ P_2 \qquad P_N] \tag{2.5}$$

, where

$$p_i = \begin{matrix} x_i \\ y_i \\ 1 \end{matrix} \quad \text{and} \quad P_i = \begin{matrix} X_i \\ Y_i \\ Z_i \\ 1 \end{matrix} \tag{2.6}$$

2. Let s denote $S = [P_1\ \ P_2 \qquad P_N]$ and $r = rank(S)$. We make the decomposition of S by using SVD, $S = UDV^T$ and denote V_2 to the matrix formed by the last $N - r$ columns of V, i.e., the ones that form the null-space of S. Therefore $SV_2 = 0_{3\times(N-r)}$, so:

$$K^{-1}[l_1 p_1\ \ l_2 p_2 \qquad l_N p_N] V_2 = 0_{3\times(N-r)} \tag{2.7}$$

3. This equation can be re-formulated as

$$\begin{matrix} K^{-1}p_1 & 0 & & 0 & l_1 \\ 0 & K^{-1}p_2 & & 0 & l_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & & K^{-1}p_N & l_N \end{matrix}^{(3)} V_2 = 0_{3\times(N-r)} \tag{2.8}$$

, or expressing the matrices in a more compact form

$$(DL)^{(3)} V_2 = 0_{3\times(N-r)} \tag{2.9}$$

, where the symbol "(3)" indicates vector transposition [40].

4. Applying vector transposition to both parts of the equation we obtain:

$$\left((DL)^{(3)} V_2\right)^{(3)} = 0^{(3)}_{3\times(N-r)} \qquad \left((V_2^T \otimes I_{3\times 3}) D\right) L = 0 \tag{2.10}$$

, where symbol "$\otimes$" indicates Kronecker product [40].

5. From Equation (2.10) we can estimate the depth matrix L (which contains all the depth values l_i), depending on a scale factor, by solving a null-space problem.
6. Once that all the variables placed at the left side of the Equation (2.4) are known, we obtain a classical problem known as Absolute Orientation, which solution can be found by different methods, like the ones presented by Horn [26] or by Arun et al. [3].

2.4 Experimental results

In order to obtain an evaluation of the multisensorial registration some tests were performed. The estimation of the intrinsic parameters is well documented in the literature and therefore some experiments have been already realized, as can be found in [67] and [68]. For this reason, the main motivation is to analyze the behaviour of Fiore s pose estimation algorithm, and two different experiments are realized: the study of the error in the estimation of the extrinsic parameters of the camera with respect to the 3D scanner, and the study of the reprojection error of the pixels.

2.4.1 Accuracy of the extrinsic parameters estimation

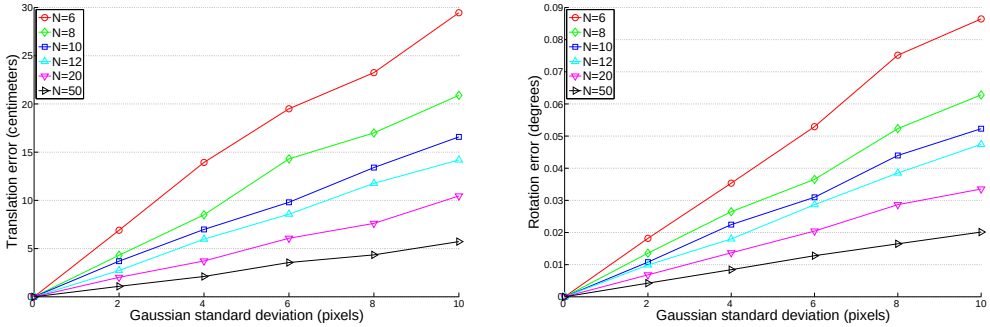
The aim of this experiment is to evaluate the estimation of the rotation matrix and the translation vector of the camera with respect to the 3D scanner, evaluating different errors in the selection of the pixels in the image. For this reason an ideal scenario with perfect correspondences between image pixels and 3D coordinates is established, and afterwards a gaussian noise is applied to the image pixels in order to simulate the possible inaccuracies in their manual selection.

Two different contexts will be analyzed, depending if we are acquiring 3D points placed at a short distance or a high distance from the 3D scanner. In general terms, we could consider that these two contexts correspond to the two situations displayed in Figure 2.2, i.e., a LADAR sensor which can deliver 3D information until about 1 kilometer of distance and a Microsoft Kinect camera which has a maximum range below 10 meters.

(a) Accuracy in the short range case

For the evaluation of the accuracy a Matlab script was created. In each execution the following steps are performed:

- Assuming that the origin of the 3D scanner is placed at point [0,0,0], a 3D point is randomly placed at a distance lower than one meter from this origin, acting as optical center of the camera. In addition a random orientation of the camera axes is considered, but giving some restrictions: the Z direction of the camera is considered similar to the Z direction of the 3D scanner, so only a random value between -10 degrees and 10 degrees is considered for both the yaw and the pitch angles. On the other side, the roll angle value is completely unpredictable and a random value between 0 and 360 degrees is considered.



(a) (b)

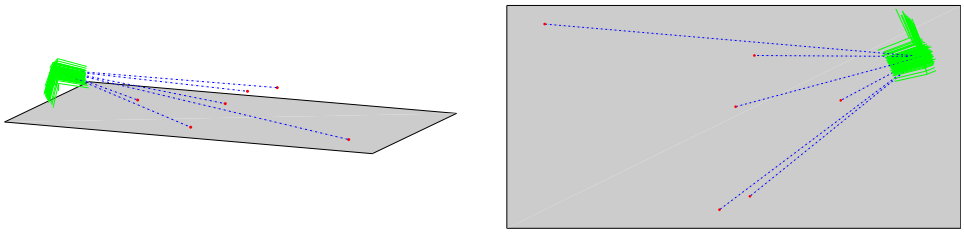
Figure 2.6: Accuracy of the translation and the rotation estimation in the short range case, for different values in the number of correspondences.

- A total number of N random 2D points are created, acting as pixels of the image. These random values are obtained between [0,0] and [2240,1488], considering this last pair as the resolution of our camera. Also N random values between 20 and 200 are computed, acting as depth values for each one of the pixel images. Assuming a focal distance of 50 millimeters for our camera, this gives a set of 3D points placed between 1 and 10 meters away from the 3D scanner.
- For the simulation performance a variable gaussian noise is added, in order to simulate the possible inaccuracy in the selection of the set of pixels. The Fiore s algorithm is applied, and their results are compared with the original values. Two different measures are computed: the norm between the obtained translation vector and the original one, and the mean of the 3 angular errors corresponding to the roll, yaw and pitch angles.

One hundred simulations are performed for each gaussian noise level, and the mean of the obtained results are displayed in Figure 2.6. In addition, some possible values for the number of correspondences N have been considered, starting from the minimal number of 6 to a final value of 50 correspondences.

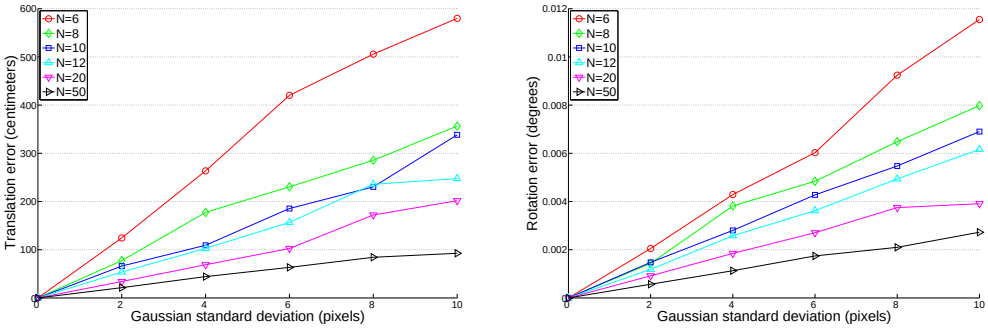
As can be seen in the figures, Fiore s method works perfectly if the point correspondences are really accurate and noise level is near to zero. In case of noise, the errors increase almost in a linear way. In addition, as expected, if the number of correspondences is increased the result is more accurate both in the translation and in the rotation estimation. This improvement shows a behaviour similar to an exponential decay. As can be seen, there exists a high reduction of error when the number of correspondences N is increased from 6 to 8, but this improvement is not proportional when the number of correspondences is set to higher values like 10 or 12. It is necessary a high increase in the value of N ($N = 20$ or $N = 50$) in order to obtain a significant reduction of the error.

In addition to the global results displayed in Figure 2.6, also the results of the estimated camera pose for 100 executions can be seen in Figure 2.7. This particular example is



(a) (b)

Figure 2.7: Perspective view and zenith view of 100 pose estimations in a single simulation, using 6 point correspondences and noise with standard deviation 6. Plane $z = 0$ is displayed for a better scene understanding.



(a) (b)

Figure 2.8: Accuracy of the translation and the rotation estimation in the large range case, for different values in the number of correspondences.

obtained by using a total number of 6 correspondences and a gaussian noise level of 6 pixels, and obtains a set of camera poses affected by their specific noise in the points correspondences. As can be seen, the estimated camera poses are similar between themselves, with little variations both in position and orientation of their axes.

(b) Accuracy in the large range case

In order to simulate the acquisition of 3D points in the large range case the same procedure was applied, with the only difference in the depth values of the pixel images. In this case, the values are comprised between 200 and 20000, which produces 3D points placed between 10 and 1000 meters away from the 3D scanner. Applying also 100 simulations for each gaussian noise level, the obtained results are displayed in Figure 2.8.

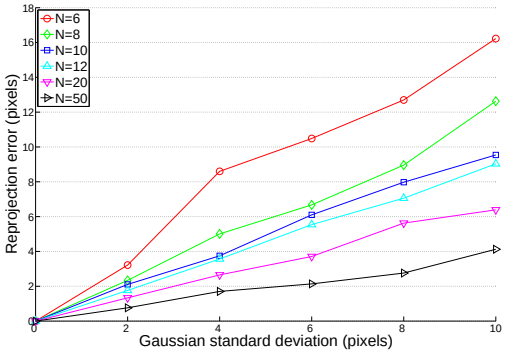


Figure 2.9: Reprojection error for different values in the number of correspondences.

In comparison to the results in the short range case, we can see that, as expected, there exists a high improvement in the rotation error but on the other side the translation error is getting worse. In addition, the estimation results in the large range case seems to be less predictable than the short range case, as it can be observed in cases where the accuracy of the pose estimation does not vary substantially when the gaussian noise is increased or the number of correspondences is changed. This behaviour can be explained by the higher diversity in the position of the 3D points (from 10 to 1000 meters) in comparison with the short range case (from 1 to 10 meters).

2.4.2 Reprojection error

We can consider that in the case of multisensorial registration we are not specially interested on the accuracy of the rotation matrix and the translation vector, but on the error in the projection of the image pixels in the 3D point cloud. For this reason we study the so-called reprojection error, i.e., the difference, in the image plane, between the original pixel and the reprojected pixel according to the estimated pose of the camera.

Again, a Matlab script is executed several times and with a variable value of N. In addition to the steps already explained in Section 2.4.1, for each execution a new image plane is created according to the estimated rotation and translation, and the 3D points are backprojected to this image plane. The pixel position of these points are compared to the original ones, and the resulting values for 100 simulations are shown in Figure 2.9. In this experiment there exist no separation between the sort range case and the large range case, since the reprojection error between these two cases does not vary substantially.

As a representative example, also the representation of the image plane for different values of correspondences can be seen in Figure 2.10 and Figure 2.11, showing at the same time the original pixel values, the pixel values with the added gaussian noise, and the reprojected pixels. As expected, for a higher number of correspondences N the set of obtained results in the reprojection error is usually more compacted.

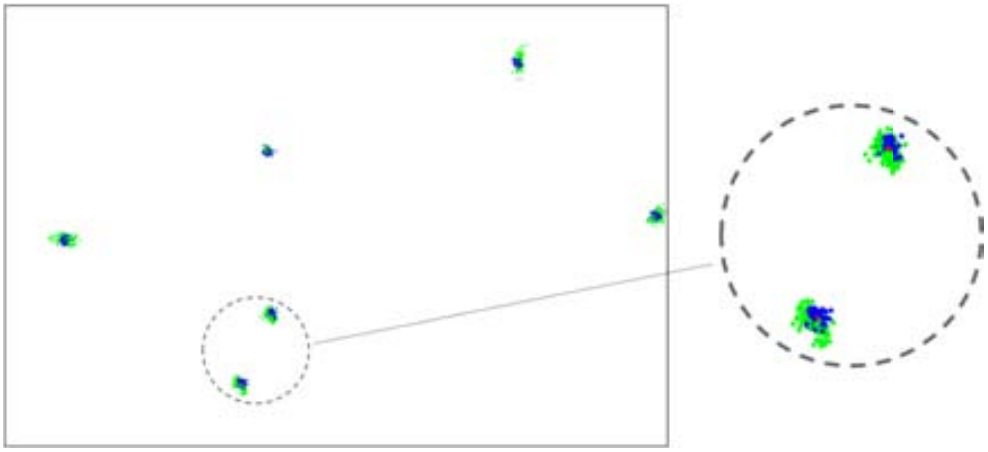


Figure 2.10: Image plane representation of a single simulation using 6 point correspondences and noise with standard deviation 10. Red points indicate the original position of the image points, blue points indicate the original image points added by the gaussian noise (100 blue points) and green points indicate the reconstructed points using the current estimated camera pose (100 green points).

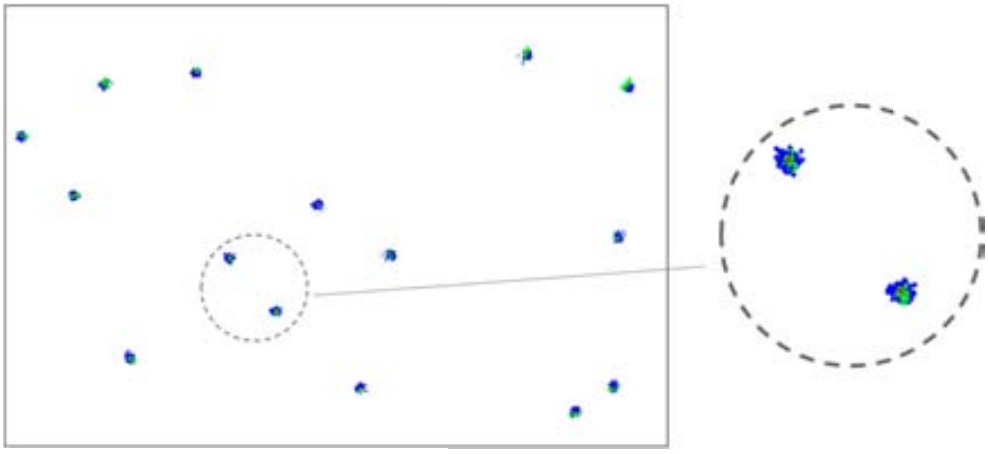


Figure 2.11: Image plane representation of a single simulation using 15 point correspondences and noise with standard deviation 10. Red points indicate the original position of the image points, blue points indicate the original image points added by the gaussian noise (100 blue points) and green points indicate the reconstructed points using the current estimated camera pose (100 green points).

2.5 Conclusions

Multisensorial registration consists in the joining of multiple sources of information at the same time. In the case of the study of this PhD thesis, it is a required and crucial process, since its results will be used afterwards in the following chapters.

Two main steps are needed in the execution of the multisensorial registration. The first one is the estimation of the intrinsic parameters of all the devices involved in the process, which is a typical situation in computer vision literature. The second step consists in the estimation of the extrinsic parameters between the cameras, which include the relative position and orientation between them.

For the estimation of the extrinsic parameters we decided for the algorithm presented by Fiore, which combines some equations regarding correspondences between pixels in the image and 3D points in the point cloud. The experimental results obtained with the Fiore algorithm achieve a good estimation of the extrinsic parameters of the camera with respect to the 3D scanner, but it must be considered the case if we are using 3D points far away or near to the scanner. In case we are obtaining point clouds in the near range (up to 10 meters) we can obtain a better estimation of the position of the camera. On the other side, using 3D points at a high distance, the estimation of the camera orientation will be more accurate. For this reason it is advisable to use both cases in the calibration process, achieving an accurate result in position as well in orientation.

Chapter 3

Pairwise Registration

Registration of range data becoming from different scanner positions is a current topic in the computer vision research. In this chapter we introduce its estimation using different algorithms in the state-of-the-art literature. Matching between the image descriptors allows the estimation of an initial rigid transformation between the views, which later can be refined with the Iterative Closest Point algorithm in order to achieve a more accurate registration. A new descriptor based on the covariance between features is presented, and also a method taking advantage of the typical structures of an urban scene is proposed, detecting large planes representing walls of buildings and filtering them in order to achieve a better accuracy.

3.1 Introduction

As seen in the last chapters, range scanning has become a quite popular system for the capture of 3D environments. The possibility of combining the 3D representation and a camera in order to apply a texture achieves a precise representation of scenes with a minimum effort.

However, one of the main problems of the range imaging is the necessity of joining 3D structures belonging to different captures in order to obtain the full representation of an object or a scene. Registration of 3D structures is a current topic in modern literature. The typically used variant is the so-called pairwise registration, where two range images taken from unknown positions are registered to each other. In this Chapter 3 different possibilities for this pairwise registration will be studied and evaluated. In the following Chapter 4 the registration of multiple 3D structures at the same time will be analyzed, but in order to achieve this multiple registration the pairwise registrations between pairs of 3D views must be previously finished.

The structure of this chapter is as follows: in Section 3.2 we introduce the typical proce-

dure to achieve the pairwise registration, composed by a first coarse registration which is used as approximation to the fine registration. In Section 3.3 an overview of the state-of-the-art algorithms of keypoint descriptors is presented. A novel descriptor based on covariance is presented in Section 3.4, and a possible pre-processing of the range data in order to improve the registration is explained in Section 3.5. Finally, the conclusions for this chapter can be found in Section 3.6.

3.2 Coarse registration and fine registration

The usual method for the pairwise registration of 3D point clouds is the so-called Iterative Closest Point (ICP) [4] algorithm, which performs an iterative process in order to minimize the mean square distance between two 3D point clouds, one of them including absolutely the other one. For each 3D point in the first point cloud, a point of the second point cloud is chosen according to the lowest Euclidean distance. The algorithm find the rotation and translation which minimizes the mean distance between all the point pairs, and this process is iteratively repeated until a minimum value is achieved.

Since the publication of ICP some modifications of the algorithm have appeared in the literature, achieving better results and advantages [8, 22, 48]. Some of these modifications include the use of weights for the pairs of points, changes in the error metric, or the use of additional attributes of the point clouds (for example, the use of color).

Also, at the same time where ICP was published, Chen and Medioni developed also its fine registration algorithm [7], based on the distance minimization between points and planes instead of the minimization between points. In this case, for each point belonging to the first set its normal vector with respect to the surface is computed, and the intersection of this normal vector with the second surface is estimated. This intersection on the second set of points defines a tangent plane on the second surface. The distance which must be minimized is the distance between the initial point and this tangent plane. Each iteration of this algorithm is generally slower than the point-to-point version used in ICP, but experimental results in the literature demonstrates that the convergence of the point-to-plane algorithm achieves a better convergence [48].

Nonetheless, the main problem of the ICP and their derived algorithms is the need of having a good initialization, otherwise the registration could converge to a local minimum and not to a global minimum. For this reason usually the main topic on the 3D registration literature is focused on the obtaining of this initial approximation, known as coarse registration. The usual method in order to obtain this initial approximation is by estimating some correspondences between points of the two 3D point clouds, allowing the rigid transformation of a limited set of 3D points in order to register with the other set of 3D points, and therefore making also possible the registration of the two whole 3D point sets. A simple example can be seen in Figure 3.1, where the visible image associated to each 3D view is shown.

Once these correspondences have been established, both sets of points can be registered by solving the classical problem of Absolute Orientation, defined by the following equation:

Figure 3.1: Point correspondences between two images representing a similar scene from different viewpoints. In addition to these visual images, it is expected that the corresponding range images were also available. The combination of these point correspondences determine a rigid transformation between the 3D structures.

$$A_i = RB_i + t \qquad i = 1, ..., N \tag{3.1}$$

, where A_i and B_i specify the two sets of N points and $[R, t]$ is the rigid transformation needed to align all the point belonging to B_i with the points belonging to A_i.

As explained in [3], the problem of Absolute Orientation can be easily solved by using the Singular Value Decomposition. Letting

$$\begin{aligned} \widehat{A_i} &= A_i - \overline{A} \qquad i = 1, ..., N \\ \widehat{B_i} &= B_i - \overline{B} \qquad i = 1, ..., N \end{aligned} \tag{3.2}$$

, where $\overline{A} = \frac{1}{N}\sum_{i=1}^{N} A_i$ and $\overline{B} = \frac{1}{N}\sum_{i=1}^{N} B_i$, we can compute the 3 x 3 matrix M:

$$M = \sum_{i=1}^{N} \widehat{A_i}\widehat{B_i}^T \tag{3.3}$$

Applying SVD we obtain $M = UDV^T$, and we can estimate the rotation matrix by

$$R = V \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(VU^T) \end{bmatrix} U^T \tag{3.4}$$

Once we obtained the rotation matrix, the translation vector can be easily calculated by

$$t = \overline{A} - R\overline{B} \tag{3.5}$$

The establishment of these correspondences between the 3D point clouds could be done by multiple forms. The easiest way is by selecting them manually after the scanning process, or it is also possible to use physical markers which are attached at the scanned object. Finally,

the other possibility is by using computer vision algorithms in order to detect similar features in both 3D point clouds. In the following sections the different possibilities using computer vision algorithms are explained, and also a novel descriptor using covariance concepts is evaluated.

3.3 Correspondences establishment

The detection, description and matching of points from complex scenes is a challenging task for many computer vision applications such as tracking, object modeling and recognition or scene reconstruction. Existing approaches make use of all the available cues in the usual two channels of information: visual photometry such as color or textures, and shape and depth information from 3D sensors. In this section some different approaches in order to establish these matchings are presented, separated according to their basic functioning.

3.3.1 Using visual information

Thanks to its vast study in the computer vision field, the matching of features in 2D images is a main topic in the literature. Among the different possibilities, the SIFT method [35] is probably the most usual and well-known algorithm. SIFT delivers features for specific keypoints in an image, being invariant to changes in translation, rotation and scale; and partially invariant to affine projections and changes in illumination. One of the main steps of this method is the creation of its descriptor, which encode the gradient magnitudes and orientations for the neighborhood of some pixels in the image. In order to achieve the invariance with respect to the orientation, these magnitude coordinates and the orientations are rotated with respect to the orientation of the keypoint, which was found in the keypoint detection process.

In our case of study, since we have registered the range images with their corresponding visible images as explained in Chapter 2, we can directly search for correspondences in the visible images. Once the correspondences have been established, we can easily convert these 2D pixel matching in 3D points matching.

3.3.2 Using 3D structure information

In opposition to the 2D visual descriptors, there also exist descriptors which use exclusively the 3D information from the scene. Inside this category exist some popular approaches like the point signatures [9], the 3D shape contexts [20] or, more recently, the Fast Point Feature Histograms [49]. However, as in the case of 2D descriptors, there exist a predominant method in the literature, which corresponds to the spin images by Johnson and Hebert [30].

The basic idea of the spin images is to represent the proximity structure for every 3D point in a surface or object. First step for its computation is the estimation of the surface normal n for the point p where we want to create the spin image. Combination of the 3D point p with its normal vector n is called oriented point. The oriented point defines a plane and also a cylindrical coordinate system. Considering a single point in the proximity called

p_i, two coordinates can then be defined: a radial coordinate α and an elevation coordinate β. Coordinate α defines the distance of point p_i with respect to the line defined by the oriented point, and coordinate β defines the distance of point p_i to the tangent plane defined by the oriented point. The function which maps a 3D point into a 2D representation is known as spin map, and its graphical and mathematical representations are shown in Figure 3.2 and in Equation (3.6).

$$\begin{aligned} \beta &= (p_i - p) \cdot n \\ \alpha &= \sqrt{(p_i - p)^2 - \beta^2} \end{aligned} \tag{3.6}$$

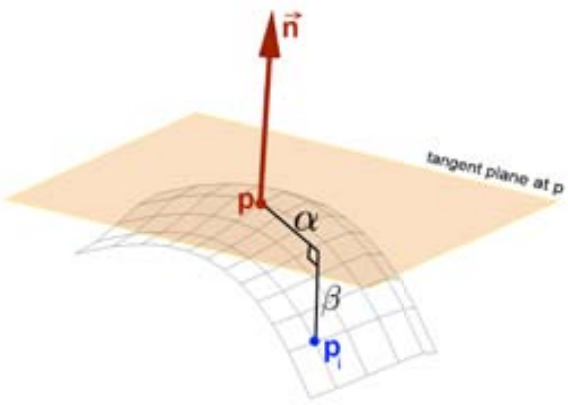


Figure 3.2: Computation of the spin image at point p.

Once all the points in the proximity have been expressed according to the coordinates α and β, we obtain a 2D image with a cluster of dots. At this point the second step of the spin image generation starts: the 2D image can be seen as an accumulator, resulting in darker areas where the accumulation of points is higher and lighter areas where the accumulation is lower. For this accumulation result we must previously define a bin size, defined as the geometric width of the bins in the spin image. The final result of the spin image should be a gray-level image normalized between 0 (white color) and 1 (black color). A correct establishment of this bin size can be crucial for the success of the pairwise registration, as seen in Figure 3.3. The use of this accumulation using the bin size makes possible the resolution invariance of the spin image descriptor.

Spin images from two different scans representing the same object will be similar but not exactly, so in order to compute the possible matching between two spin images we can use a correlation coefficient. The higher the correlation coefficient, the more probable that both points represent the same vertex in the object or surface.

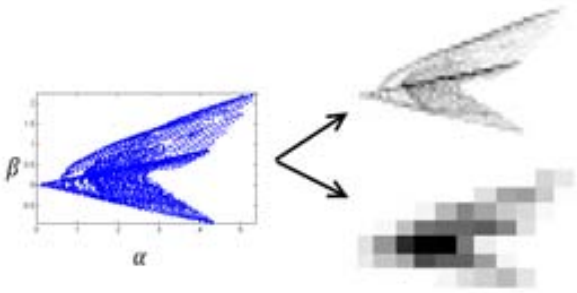


Figure 3.3: Dependence of bin size in the creation of the spin image

3.3.3 Using simultaneously 3D and visual information

While previously explained methods have given successful outcomes in both areas, there exist also the possibility of fusing information from both two worlds and provide a descriptive unit which is able to encode shape and visual information together.

A first approximation to the problem of fusing both cues of information can be found in works like [55] and [64]. Taking into consideration the problem of the SIFT descriptor in order to deal with high differences in the viewpoints of an scene, these works make use of the 3D information by estimating the surface normal of the 3D coordinate and performing an homography of the visible image as it would be seen from the front side of the keypoint. An example of this process can be seen in Figure 3.4.

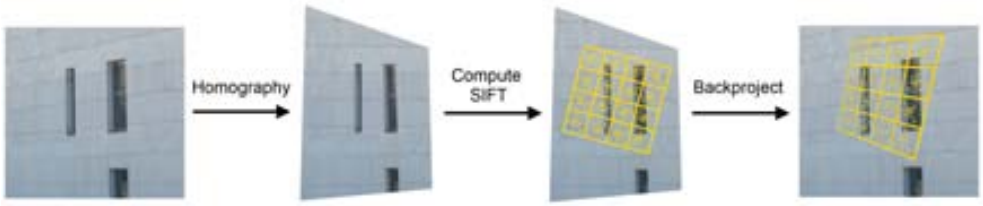


Figure 3.4: Thanks to the 3D information of the range image, we can estimate the homography. The SIFT descriptor is computed in this second image and afterwards backprojected to the original image.

With this homography we can, in part, avoid some of the possible drawbacks produced by a high difference in the viewpoint. However, it is not expected to solve completely these problems. For this reason, during last years some descriptors which encode intrinsically at the same time information from the 3D shape and the texture have been published in the literature. A good example is the MeshHOG descriptor [66], which performs a histogram of gradient of the neighborhood of a 3D point by using separately the texture information and the 3D curvature. In order to include both cues in the final descriptor, both representations can be directly concatenated. This same methodology is also used from the authors of the CSHOT

descriptor [57], which concatenates their SHOT descriptor [58] and the color information.

In addition to the previously presented descriptors, we have developed a covariance-based descriptor which is able to gather shape and visual information together within a radial 3D area. Thanks to its fundamentals, this descriptor is robust to noise and point-view changes, and because of its low computational cost it can be extended to a multi-scale context for better discrimination performance. In addition, the own descriptor offers a procedure for keypoint extraction, so salient points in the scene can be detected at the same stage where descriptors are being obtained. The detailed functioning and formulation of this novel descriptor is presented in the following section.

3.4 Covariance descriptor for fusion of 3D shape and texture information

Covariance matrices in the computer vision context arose as a way for relating several image feature statistics inside a region of interest. From a statistical point of view, the concept of covariance is considered as a measure for the strength of correlation between two or more sets of random variables, or, more informally, "how several variables change together". This usage of covariance magnitudes as descriptive units was first introduced by Tuzel in [61] for the detection of objects and faces. The approach was extended to more concrete cases for pedestrian detection [62]; or used by other authors also for detection of objects, not only from visual cues [65], but also for 3D shape description as explained in [16].

Following the aforementioned works, a novel descriptor which is capable of encoding shape information as well as visual cues in 3D scenes is presented.

3.4.1 3D covariance descriptor construction

A brief reminder of statistical notions says that the covariance measure between two random variables X and Y can be computed as:

$$cov(X\ Y) = \sum_{n=1}^{N} \frac{(x_n - \bar{x})\,(y_n - \bar{y})}{N} \tag{3.7}$$

where N is the number of samples of each variable and $\bar{x}$ and $\bar{y}$ are their sample mean. Covariance value is zero if the variables are absolutely uncorrelated, positive if both of them tend to increase together, and negative if one of the variables increases while the other decreases.

For a set of F different variables, this formulation can be extended to a matrix notation:

$$C_{i,j} = cov(x_i\ x_j) = \ (x_i - \overline{x_i})\,(x_j - \overline{x_j}) \qquad i\ j \qquad 1\ \ F \tag{3.8}$$

Thus, the result will be a symmetric matrix C of size $F \times F$ where the diagonal entries

will represent the variance of each one of the variables, and the non-diagonal entries will represent their correlations.

Bringing back these notions to the descriptor definition, the set of random variables must correspond to a set of observable features which can be extracted from points in the scene, e.g. pixel color values, depth magnitudes, 3D coordinates, etc. Therefore, the first step for the computation of the descriptor is the definition of a feature selection function for a given point p and its neighborhood of radius r in the scene, $\Phi(p, r)$. For the case of color and depth fusion, we define the feature selection function as follows:

$$\Phi(p, r) = \{ R_{p_i} \, G_{p_i} \, B_{p_i} \, \alpha_{p_i} \, \beta_{p_i} \, \gamma_{p_i} \} \, \forall p_i \;\; \text{s.t.} \;\; |p - p_i| \leq r \tag{3.9}$$

, where visual information is taken into account in terms of R, G and B color values; while α, β and γ values are angular measures which encode the shape information of the points within the neighborhood of the descriptor center. Their graphical representation can be seen in Figure 3.5, and are obtained in the following way: assuming p as the center point of the descriptor and p_i as each one of the points within its neighborhood, α is the angle between the normal vector in p and the segment from p to p_i; β is the angle between the same segment and the normal vector in p_i; and γ is the angle between both normal vectors in p and p_i. As these selected features are relative measures in terms of shape description, their usage in the covariance descriptor formulation guarantees a rotation and view invariance, which is a desired behaviour in terms of descriptor performance.

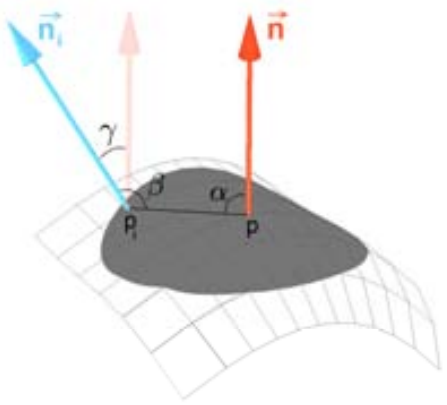


Figure 3.5: Scheme of the used features for shape information encoding. For each p_i in the neighborhood of p, α, β and γ are the rotational invariant angular measures.

Then, for a given point p of the scene, and being $\{\phi_i\}_{i=1 \;\; N}$ the set of 6-dimensional points obtained by the feature selection function within its neighborhood (according to the terms defined in Equation (3.9)) the covariance descriptor can be obtained as:

$$C_d \left(\Phi(p, r) \right) = \frac{1}{N - 1} \sum_{i=1}^{N} \left(\phi_i - \mu \right) \left(\phi_i - \mu \right)^T \tag{3.10}$$

where μ is the mean of the points $\beta_i \ _{i=1..N}$.

In addition, in order to improve the possibilities of the descriptor, it is easy to extend it to a multi-scale framework by just adding several radius magnitudes for the neighborhoods around the descriptor center point. Therefore, each point in the scene will have a set of descriptors:

$$C_M(p) = C_d\left(\Phi(p \ r_s)\right) \ \ r_s \tag{3.11}$$

The idea behind using several neighborhood radios is that discrimination performance can be improved if a point is supported by more than one descriptor, regarding a narrow to coarse set of surrounding areas. Then, we are intentionally seeking matches of points which are locally similar, but also related in a more global area. This can help to avoid repeatability problems and improve detection of points in edges or borders of the objects.

An example of covariance descriptor is displayed in Figure 3.6, where the different features representing shape and color are shown and also the different multi-scale radius can be seen in its left image.

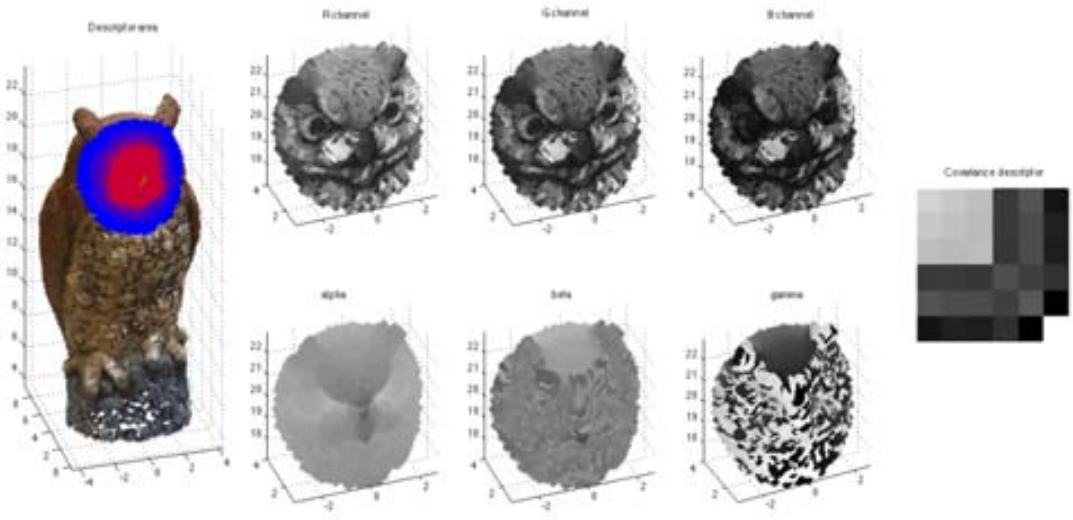


Figure 3.6: Example of a scene view where a multi-scale covariance descriptor is extracted. The left image shows the original 3D scene where the overlap gradient of colors from red to blue depicts 5 different scales used for obtaining a multi-scale descriptor. The 6 central subfigures show the different used features, in terms of color (upper row) and shape description (bottom row). Finally, on the right, a single scale 6x6 covariance descriptor is graphically represented.

As a summary, a covariance descriptor can be seen as a very compact representation which looses all the spatial notion of the region, and just encodes the joint distribution of the

features which are observed within this region. This statistical representation of the surface is more flexible than a histogram-based approach, which is based on a counting methodology. The same can be considered regarding noise tolerance: in front of outliers, a histogram-based descriptor will suffer from the uncertainty of assigning those spurious samples to the correct bin. Covariance matrices a) capture this noise distribution as a natural part of the samples in the represented area, b) by the mean subtraction in its formulation help to lower the noise affection, and c) do not need to be parameterized so the chance of unintentionally affecting its performance is avoided.

3.4.2 Matching between covariance descriptors

Covariance matrices, being symmetric positive definite matrices, do not lay on a Euclidean space, but on a Riemannian manifold. As a quick, simple example: a covariance matrix can be multiplied by a negative real value, and it will no longer be a symmetric positive definite matrix, so the Euclidean assumption is broken. Indeed, covariance descriptors form the $d \times d$ dimensional space of symmetric positive definite matrices, where d is the number of used features. This has several implications, but the most important one is the computation of distances on top of their manifold.

As Euclidean distance is no longer valid, a new metric for the computation of geodesic distances on top of the manifold must be derived. In order to measure the similarity of descriptors, the metric for computing distances between two covariance matrices C_d^1 and C_d^2, was proposed by Förstner in [19] as

$$\delta(C_d^1\ C_d^2) = \sqrt{\sum_{i=1}^{n} ln^2 \lambda_i(C_d^1\ C_d^2)} \tag{3.12}$$

where $\lambda\left(C_d^1\ C_d^2\right)$ are the generalized eigenvalues of C_d^1 and C_d^2.

Again, this can be seen as a performance boost regarding other state-of-the-art descriptors: the way of computing descriptor likelihoods is a geometrically-aware metric, rather than a distance approximation as would be the case on histogram-based approaches. This supports the consideration of covariance matrices as a powerful and robust representation.

Similarly to the introduction of the multi-scale framework in the descriptor construction, we can extend this idea to the metric computation:

$$\delta_M(C_M^1\ C_M^2) = \min_j \sum_{i=1}^{5} \delta(C_M^1 i\ C_M^2 i) \quad - \delta(C_M^1 j\ C_M^2 j) \tag{3.13}$$

where $C_M^1 i$ and $C_M^2 i$ are the covariance descriptors belonging to each one of the $i = 1 \quad 5$ scales, at each one of both scenes respectively. The formulation behind Equation (3.13) takes into account the similarities of all scales except the less matching one, which is ignored because it might contain a major dissimilarity at a given scale. This equation is then used

as the function for comparison between the matching of two multi-scale descriptors, and it satisfies all the required properties of a metric:

$$\delta_M(C_M^1\ C_M^2) \geq 0 \text{ and } \delta_M(C_M^1\ C_M^2) = 0 \text{ iif } C_M^1 = C_M^2$$
$$\delta_M(C_M^1\ C_M^2) = \delta_M(C_M^2\ C_M^1)$$
$$\delta_M(C_M^1\ C_M^2) + \delta_M(C_M^1\ C_M^3) \geq \delta_M(C_M^2\ C_M^3)$$

3.4.3 Covariance Descriptor as a keypoint detector

Covariance matrices as descriptors have still other desirable outcomes thanks to their mathematical underlying fundamentals. One of them is that they can be also used as keypoint detectors in a direct way. As defined in Equation (3.8), a covariance matrix C contains the variance of the observed features on its diagonal, and the covariance on the other entries. Computing the determinant of a covariance matrix is equivalent to obtaining the so-called *generalized variance*, which can be interpreted as a measure of the degree of homogeneity of each point in the scene.

Once the covariance descriptors have been computed, one can observe their determinants and consider that the ones with higher values will belong to real interest points, with inner significant variation. It must be taken into account that these interest points combine both visual and shape saliency. Therefore, even in the case of an homogeneously coloured object like in Figure 3.7, keypoints are still obtained on significant parts such as eye holes or borders. On the same theoretical basis, points with zero or near-to-zero generalized variances will belong to constant areas and could be discarded.

It is also possible to perform a point suppression stage thanks to the analysis of the rank of the covariance matrix descriptors: in the case where points must contain some sort of correlation between observations, that is, when there is no significant variance between features, the rank of the descriptor matrices will be lower than the number of used feature dimensions.

3.4.4 Experimental results

The proposed covariance descriptor is validated on a model dataset combining 3D shape with visual information. The dataset contains 12 scenes which have been obtained by the Autodesk 123D Catch service [1], combining own acquired objects and others available under a Creative Commons license. These models are stored as 3D point clouds with photometric texture, where each vertex has a unique identifier. See Figure 3.8 for a visual representation of the 12 models used, including different characteristics like high and low diversity of colors, repeated areas, homogeneous surfaces and textures, or symmetries.

[1] http://www.123dapp.com/catch

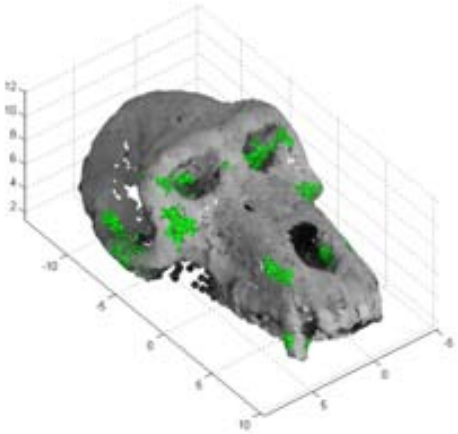


Figure 3.7: Visual example of keypoint detection by generalized variance. This figure shows the 1500 most significant coordinates of the scene, marked by sorting the covariance descriptor determinants in descendant order. Even if the color information of the object is rather homogeneous, interest points have been detected on salient areas of the scene. The computational cost of such task is only related to the determinant calculation: no derivatives or gradient information are needed.

Experiment 1: Descriptors comparison

In order to test the descriptor performance, we compare our approach against the state-of-the-art methods spin images [30], MeshHOG [66] and CSHOT [57]. For doing so, we have selected 100 arbitrary points from three models in the database, and have computed the descriptor likelihoods regarding the same 100 points on a different instance of each model, under arbitrary rotations and translations. This has been done under different levels of additive noise for color and surface coordinates: 0, 2, 6, 8 and 10 per cent of the standard deviation of data.

The evaluation method consists on observing the amount of false and true positives, and false and true negatives, in terms of matching scene points by their descriptor likelihood measures. According to a *ratio* parameter, we present two methods for evaluation of matches:

- In the first method, called *exclusive ratio*, we consider a match as a positive if and only if the best descriptor likelihood for a given point is *ratio* times better than the second best match candidate likelihood. This method has the particularity of being more restrictive on finding true positive matches, but also adds the advantage of reducing the apparition of false positives. By moving the *ratio* coefficient amongst a range of 1 to 5, we can obtain a set of meaningful ROC curves depicting the behaviour of the different tested descriptors, as seen in Figure 3.9. For a numerical comparison between these curves, Tables 3.1, 3.2 and

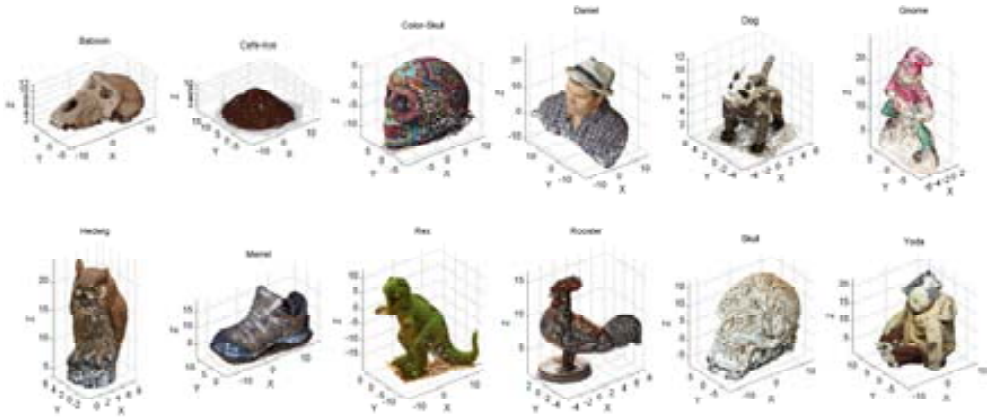


Figure 3.8: 3D plot of the 12 models included on our database. Full scenes are shown without added noise.

3.3 include their *area under the curve (AUC)* evaluation.

- In the second method, called *inclusive ratio*, we consider a set of point matches as positives if the descriptor likelihoods of those matches are within the boundaries of *ratio* times the best likelihood of this set of candidates. In this method the rate of true positive candidates is increased, but this has the expense of adding false positives and needing a posterior non-desired match suppression stage. Again, by moving the *ratio* coefficient amongst a range of 1 to 5, we can obtain a set of ROC curves for comparing the behaviour of the different tested descriptors, as shown in Figure 3.10. For a numerical comparison between these curves, Tables 3.4, 3.5 and 3.6 include their *area under the curve (AUC)* evaluation.

The results of this experiment validate the formulation of the covariance descriptor proposal specially in the cases of robustness to noise. The covariance formulation implicitly uses the mean of the used features as random variables, which contributes to smoothing the possible data deviations in a natural way, still capturing the distribution of data. Also, as commented in Section 3, the metric definition in Equation (3.12) is a coherent distance defined on top of a topological space, and not as a histogram distance approximation as would be the case in other descriptors where changes in data might imply a different representation in bin counting.

Experiment 2: Global matching evaluation

For testing the overall performance of the descriptor and its associated keypoint detector, we have designed an exhaustive scene registration test where each one of the 12 models has been split in halves of different common overlap (from 10% to 70% of the surface in common). A random rotation and translation are applied to one of the halves. In addition, each model is tested under different levels of noise, from 0% to

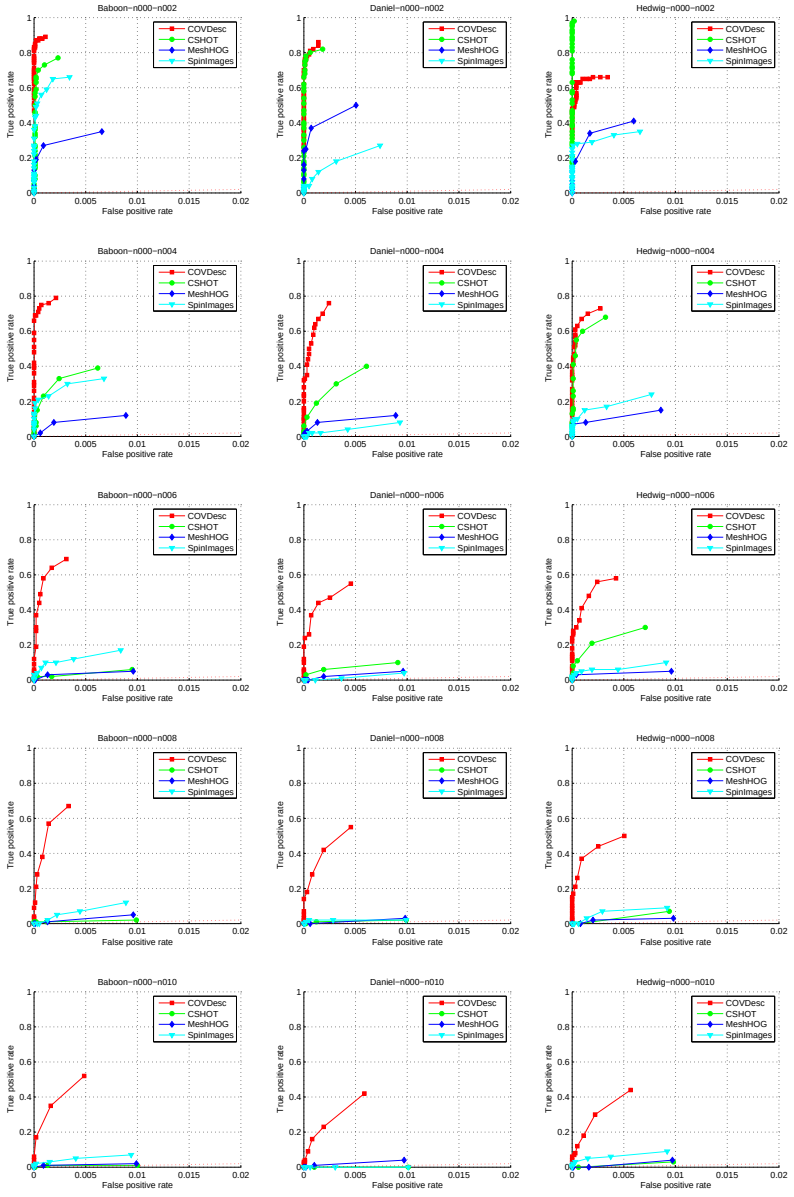


Figure 3.9: ROC curves for comparison of several 3D and visual information descriptors. Each column depicts a test on a different scene of our database. Each row shows the behaviour of the descriptor under different levels of additive noise over data (0, 2, 6, 8 and 10 per cent of the standard deviation of color and surface coordinates). In the first row, under no noise, we can see how our descriptor is similar in performance to other state-of-the-art approaches. Despite of that, when data is modified with higher noise values, our descriptor outperforms any other current method. This is due to the flexibility of a covariance-based formulation, which is capable to deal with noise on data in a more robust way than any histogram based approach.

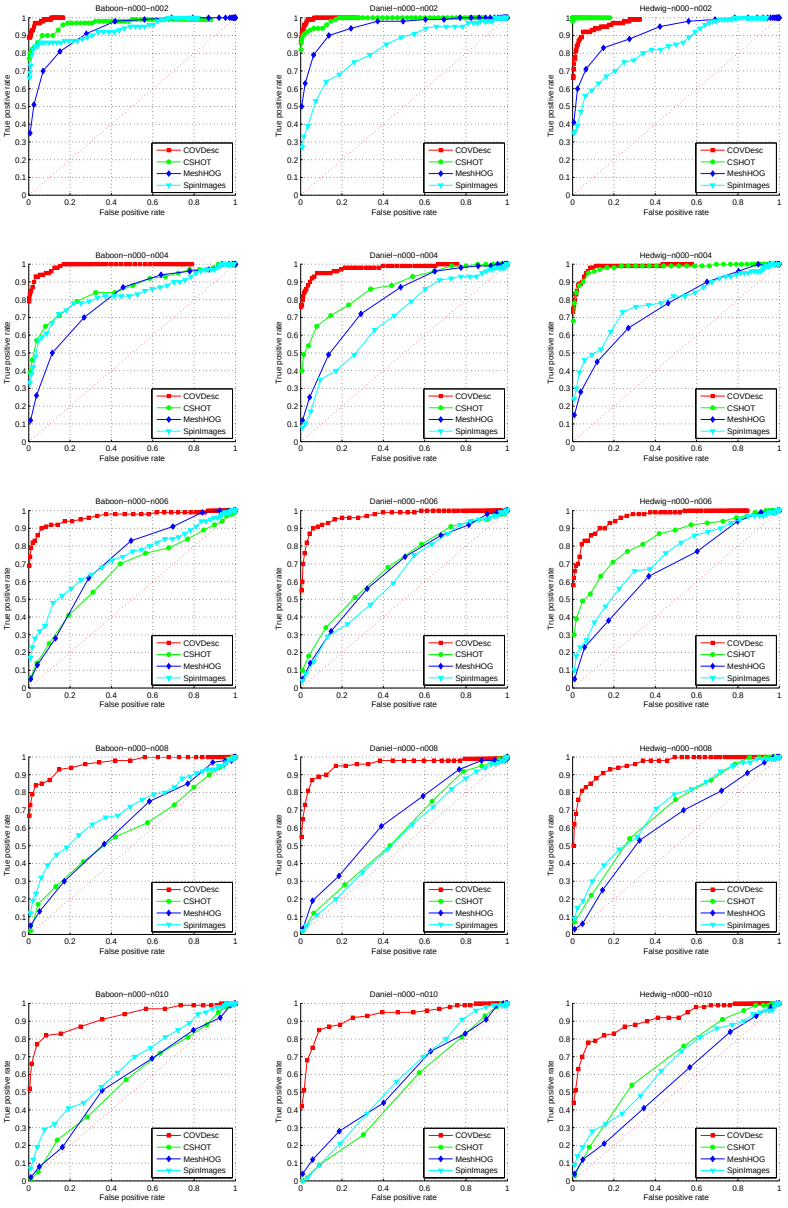


Figure 3.10: ROC curves for comparison of several 3D and visual information descriptors. Each column depicts a test on a different scene of our database. Each row shows the behaviour of the descriptor under different levels of additive noise over data (0, 2, 6, 8 and 10 per cent of the standard deviation of color and surface coordinates). In the first row, under no noise, we can see how our descriptor is similar in performance to other state-of-the-art approaches. Despite of that, when data is modified with higher noise values, our descriptor outperforms any other current method. This is due to the flexibility of a covariance-based formulation, which is capable to deal with noise on data in a more robust way than any histogram based approach.

	n002	n004	n006	n008	n010
COVDesc	**0.944**	**0.894**	**0.844**	**0.833**	**0.758**
CSHOT	0.884	0.692	0.525	0.505	0.500
MeshHOG	0.672	0.555	0.520	0.520	0.505
SpinImages	0.829	0.662	0.581	0.555	0.530

Table 3.1: Area Under the Curve (AUC) measures for the scene *Baboon*, using the *exclusive ratio* evaluation.

	n002	n004	n006	n008	n010
COVDesc	**0.919**	**0.879**	**0.773**	**0.773**	**0.707**
CSHOT	0.909	0.697	0.545	0.505	0.494
MeshHOG	0.748	0.555	0.520	0.510	0.515
SpinImages	0.631	0.535	0.515	0.505	0.494

Table 3.2: Area Under the Curve (AUC) measures for the scene *Daniel*, using the *exclusive ratio* evaluation.

	n002	n004	n006	n008	n010
COVDesc	0.829	**0.864**	**0.788**	**0.748**	**0.717**
CSHOT	**0.989**	0.839	0.646	0.530	0.510
MeshHOG	0.702	0.570	0.520	0.510	0.515
SpinImages	0.672	0.616	0.545	0.540	0.540

Table 3.3: Area Under the Curve (AUC) measures for the scene *Hedwig*, using the *exclusive ratio* evaluation.

	n002	n004	n006	n008	n010
COVDesc	**0.995**	**0.989**	**0.963**	**0.962**	**0.919**
CSHOT	0.967	0.851	0.645	0.581	0.562
MeshHOG	0.918	0.790	0.713	0.611	0.569
SpinImages	0.934	0.822	0.722	0.694	0.649

Table 3.4: Area Under the Curve (AUC) measures for the scene *Baboon*, using the *inclusive ratio* evaluation.

	n002	n004	n006	n008	n010
COVDesc	**0.997**	**0.977**	**0.968**	**0.954**	**0.925**
CSHOT	0.989	0.867	0.683	0.576	0.507
MeshHOG	0.946	0.784	0.662	0.650	0.558
SpinImages	0.826	0.691	0.627	0.554	0.566

Table 3.5: Area Under the Curve (AUC) measures for the scene *Daniel*, using the *inclusive ratio* evaluation.

	n002	n004	n006	n008	n010
COVDesc	0.975	**0.986**	**0.960**	**0.957**	**0.907**
CSHOT	**0.999**	0.979	0.830	0.685	0.668
MeshHOG	0.914	0.750	0.668	0.612	0.560
SpinImages	0.833	0.781	0.727	0.701	0.638

Table 3.6: Area Under the Curve (AUC) measures for the scene *Hedwig*, using the *inclusive ratio* evaluation.

10% of the standard deviation of color and surface coordinate values. For each scene, the experiment is conducted 5 different times, leaving to a total of 12x7x11x5 = 4620 executions.

In order to be able to evaluate the performance of the whole system, we need an additional algorithm for the detection of the possible outliers in the correspondences establishment. This algorithm should select the correspondences which are geometrically consistent between themselves, producing a final rigid transformation which register the two views. For this outlier detection we have selected a solution based on a game theory approach, similar to the one presented in [2]. The use of this algorithm take into account not only the geometric positions of the correspondences, but also the value of the likelihood of the covariance descriptors. In addition, it does not require any input parameters and always estimate the global minimum of the system, independently of its initialization.

The registration error measure is evaluated by looking at the average Euclidean distance of points in the common overlap surface. In the case of executions with noise, the system is solved on noisy data but the performance is evaluated on the equivalent un-noised scenes in order to be coherent on performance comparison. Objects are normalized so they fit within the boundaries of a prism of unitary volume.

An error acceptance threshold of 0.02 is chosen, which means that objects of one cubic meter of volume should have an average error lower than 2 centimeters. By establishing this threshold we can represent the execution of all registrations by a histogram of how many of them are considered as correct, for each condition of noise and overlap. Such histogram is displayed in Figure 3.11, where can be seen that the method works extremely well for cases with overlap of 20% and higher, while for an overlap of 10% between the two views the method shows a reduction in its performance when the noise level is increased.

In Figure 3.12 we can see the distribution of the error magnitudes only for the aforementioned correct scene registrations. Again, and as expected, the most challenging conditions are those where the system is tested with a smaller overlap and a higher noise. Nevertheless, by watching the value distributions on these two figures, we can conclude that the approach is more sensitive to the minimum overlap than to the noise tolerance.

By looking individually at the experiments for each one of the models, we observe that the best results are obtained for the models *Yoda*, *Rooster* and *Merrel*. On the other side, the worst results are obtained by *Cafe-rice* and *Daniel*, specially for the executions with low overlap. These last two cases can be easily argued: *Cafe-rice* represents an scene with a low variability in color and a clear axial symmetry, while a high part of the model *Daniel* is composed by a repetitive pattern (the shirt). Examples of incorrect registrations for both cases can be seen in Figure 3.13.

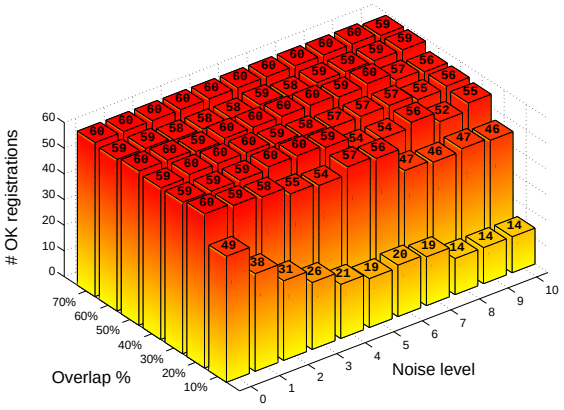


Figure 3.11: Histogram of correct registrations (for an error threshold of 0.02). As we can see, the performance of our approach is rather homogeneous on most of the experimental conditions, even with low overlap between scenes and high levels of noise applied to data.

3.5 Pre-processing for urban scenarios: plane filtering

The first step in order to achieve the pairwise registration is the detection of some keypoints in the images, which will be used afterwards to compute the corresponding descriptors. This detection is usually carried out by using the whole scene, without considering the particularities of the specific environment. Depending on the nature of the scene, and specially in the cases of urban environment, traditional techniques of keypoint detection will find a huge number of keypoints in places with low interest for the registration. As an example, see the image shown in Figure 3.14, where the keypoints detected with a DoG detector [35] are shown. As can be seen, elements with low importance for the registration as could be the autos have most of the detected keypoints. These elements can disappear in posterior scans in the future and therefore it is not desirable to use them as basis for the registration process. On the other side, the wall of the building, which should have the higher importance, does not have any keypoints by itself and only thanks to the presence of windows some keypoints are detected.

In this section we introduce the possibility of achieving the registration assuming that the images are captured in an urban environment, and therefore looking for specific structures typical in urban scenarios. For these reasons, we first make a filtering in order to find planes with a minimum area, which will correspond to the walls of buildings. After this preprocessing, existing methods can be applied achieving an improved result. As we have access to the 3D structure of the scene thanks to the range image, we could use this information to have a pre-processing of the data and filter only the parts of the scene that are plane or near to plane. In this way, once the possible planes are detected, we can in a second step perform the keypoint detection and obtain only the significant information.

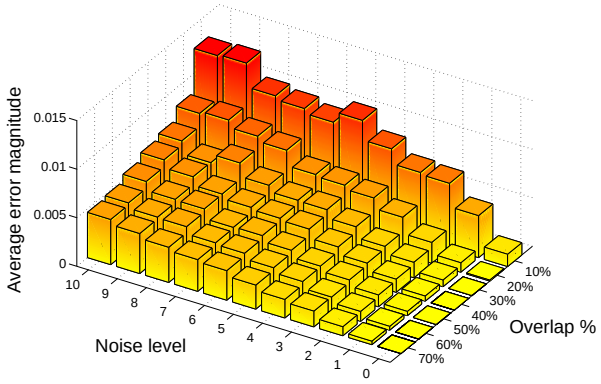


Figure 3.12: Average error distribution of those registrations considered as correct. As one could expect, major errors occur on the cases of higher noise levels and less overlap.

3.5.1 Proposed Method

As mentioned, basis of our method is the processing of the 3D scanner data in order to find planes with large area. Some methods exist for this plane detection, as the one explained by Cantzler in [5]. In our case we estimate the surface normal for every 3D point and afterwards a grouping process of normal vectors is computed. An easy way for the normal vector estimation in every 3D point is by computing the SVD for the covariance in the neighborhood of the point and select as normal vector $\overline{N}$ the eigenvector corresponding to the smallest eigenvalue.

The collection of all the normals for every range image can be expressed in polar coordinates and grouped together creating a 2D graph, where every normal is represented as a coordinate indicating angles θ and β with respect to the axes. Result can be seen in Figure 3.15. Applying a clustering method (e.g. gaussian mixtures [37]) to this result it can be detected that exist different groups of normals with similar orientation. Assuming that the 3D scanner can capture the scene in a pivoted position, all the points in the 2D graph will be usually formed by two groups of points sets, separated Π 2 in the β angle between them. The first group, the one with higher β value and usually with a higher number of normal vectors, correspond to the 3D points belonging to the floor of the scene. The second group, which can have different subgroups along the θ value, corresponds to the different walls of the scene.

Once the different groups have been filtered, we should determine where are the different walls by checking that every 3D point present after the filtering should have at least a pre-defined number of neighbors in the proximity with a maximum distance to the plane defined by its normal, and also with a similar normal orientation. The proximity size and the maximum distance to the plane depend on the resolution of the scanning process. The idea is to filter the 3D points that, even having a surface normal with an orientation similar to the selected, are isolated points or belonging to little walls. In order to find the distance of the neighbor points it can be used a concept similar to the one used in spin images.

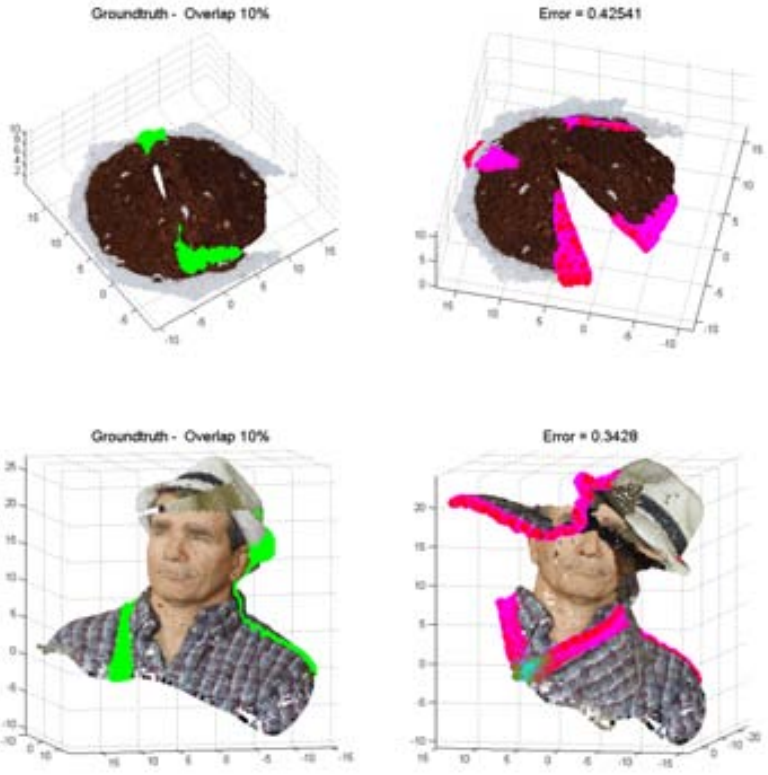


Figure 3.13: Examples of incorrect registration results. Left column shows the groundtruth of the two scenes, where two halves have been overlapped. Green points show the points of common surface. Right column show the evaluated registration, with points ranging from green to red color according to their distance respect to groundtruth labeled points. In the first row, depicting the *Cafe-rice* scene, the low overlap and the axial symmetry do not allow a global awareness as big enough for our system to discard mismatches. The second scene, *Daniel* is also selected with a low overlap, including a high repeatability. We can see how the reconstruction, again, has failed due to taking into consideration only those regions with repeated areas.

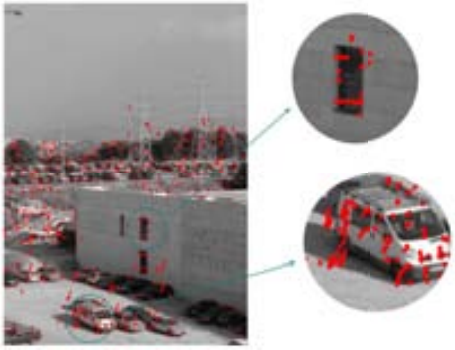


Figure 3.14: Keypoints detected in an image using DoG detector. The presence of autos produce a high number of keypoints in the image, while the wall of the building receive a lower number.

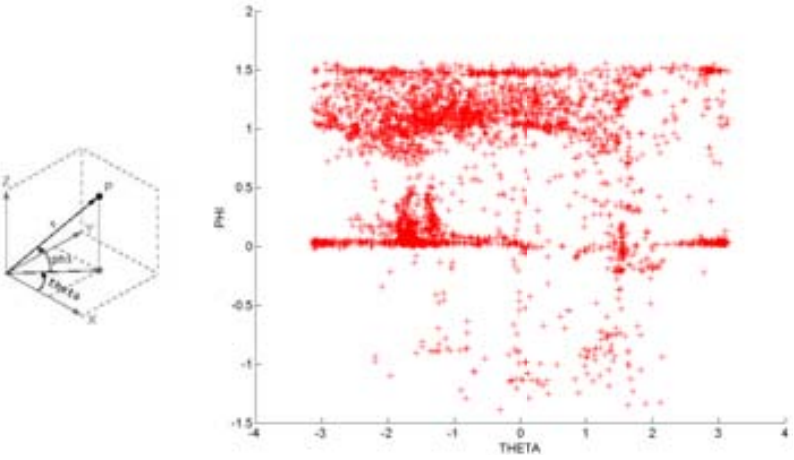


Figure 3.15: Accumulation of normal vectors expressed in polar coordinates

Once all the points in the proximity have been projected we obtain a 2D image with a set of dots, as seen in Figure 3.16. At least a pre-defined number of points should have a maximum distance to the plane defined by the current 3D point and its normal, that is, this number of points should have a small value of β in the spin image representation.

After this filtering of 3D points we must project them to the associated visible image in order to process them with 2D image techniques. As we have previously estimated the multisensorial registration between the camera and the 3D scanner, it is easy to find this projection to the 2D image. The resulting image should contain a set of nearly-equispatiated points at the zones where a wall is detected. A postprocess based on morphology methods is required in order to join these separated points forming areas. An example can be seen in Figure 3.17.

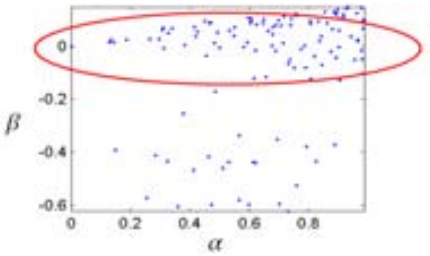


Figure 3.16: Spin map and checking of neighboring points with low distance to the plane

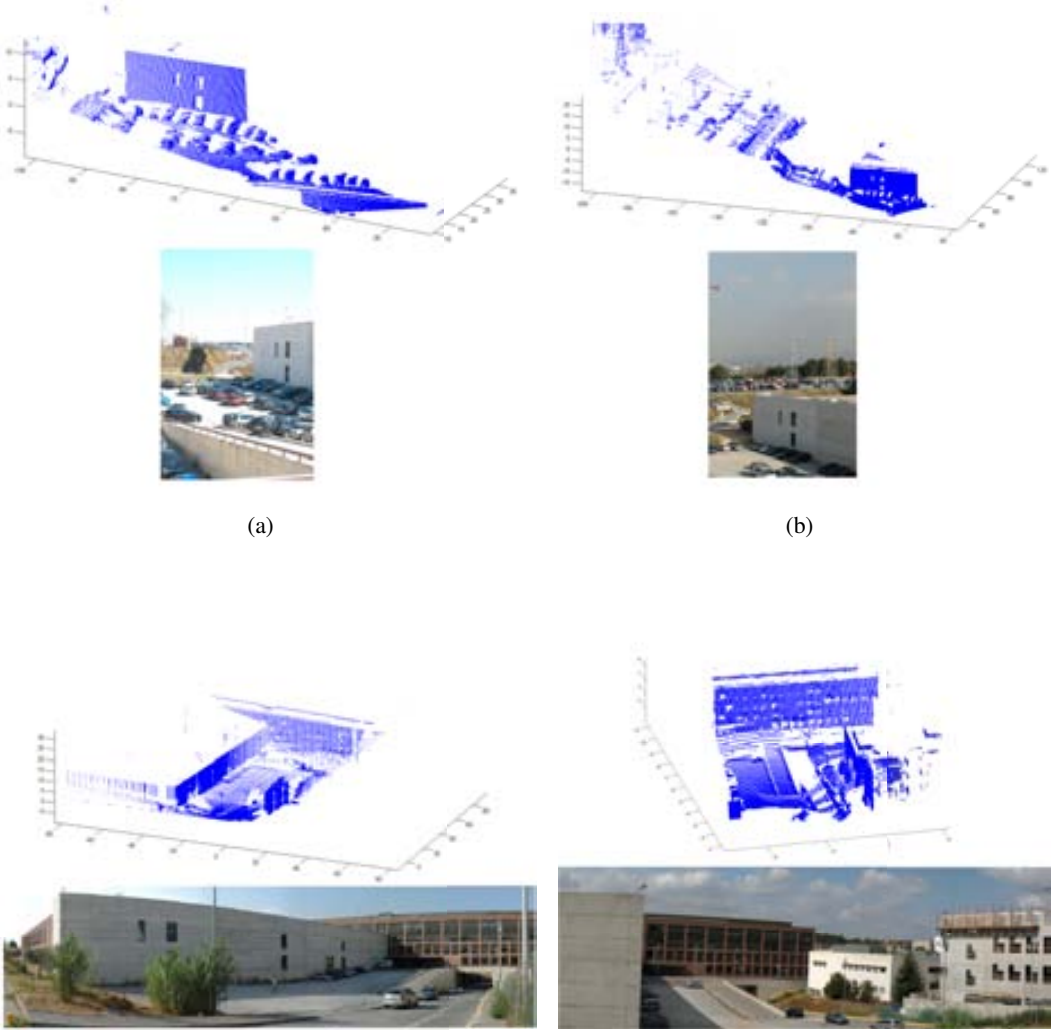
Figure 3.17: Visible image with projected points and generation of the filter image

With all the planes of the scan image detected, it is now possible to apply the methods described in the state of the art (SIFT descriptor, spin images, etc...), but now with the advantage of the filtering of possible disturbance elements.

3.5.2 Experimental results

The experiments have been carried out with four scans captured with a laser scanner Riegl LMS-Z420i and an attached camera Nikon D100, which was previously calibrated with the laser scanner. The scans capture a similar portion of a scene, containing walls, vegetation and vehicles. Dates of capture were different, so there is no correlation between the vehicles and persons. The four scans can be seen in Figure 3.18. Also, for a better scene understanding, their associated visible images are shown.

The proposed filtering method has been applied to the range images and the detection of the walls for *Scan1* and *Scan2* is shown in Figure 3.19. The registration is achieved by searching keypoints using the DoG detector only to places where a plane is detected, and applying afterwards the SIFT descriptor. Finally, among all the possible matches detected a RANSAC method [18] is applied in order to find the rigid transformation between the scans. The registrations achieved are shown in Figure 3.20.



(a) (b)

(c) (d)

Figure 3.18: Scans used for the experiments, called (a) *Scan1*, (b) *Scan2*, (c) *Scan3* and (d) *Scan4*. The scans have been captured with a laser scanner Riegl LMS-Z420i, and their associated visible images associated are also shown for a better scene understanding.

Figure 3.19: Filter images for *Scan1* and *Scan2* after applying the plane detection.

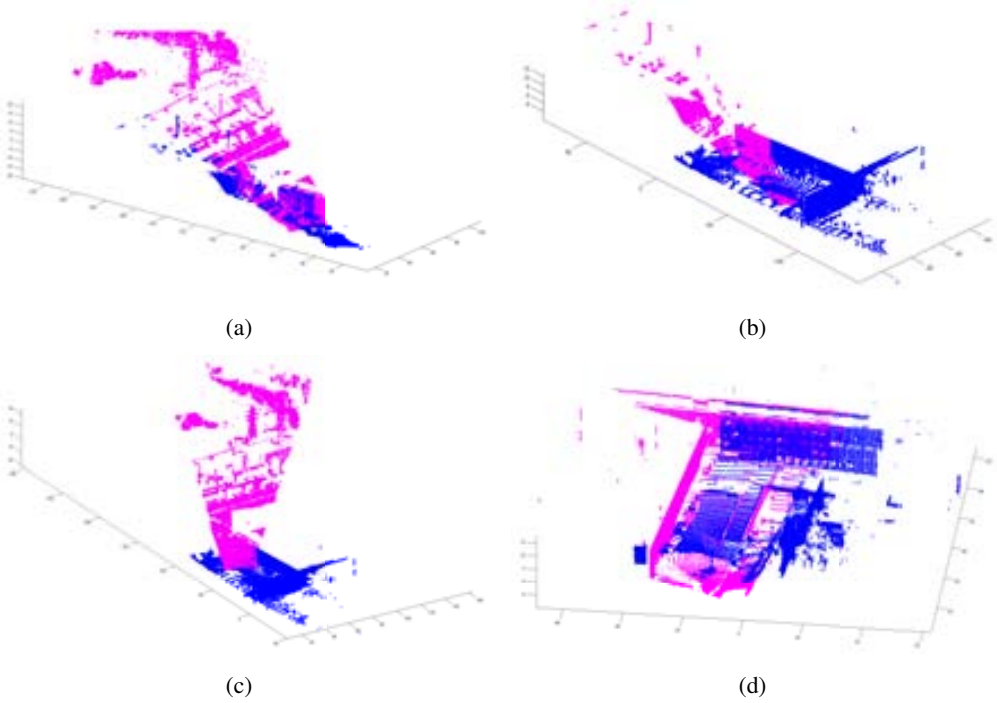
3.6 Conclusions

This chapter contains the developments of the so-called pairwise registration, which estimate the rigid transformation in order two align two 3D point clouds obtained from different unknown positions. The registration is based on the computation of specific keypoint descriptors and the posterior matching between them.

We have introduced a novel descriptor for fusion of 3D shape and visual information which works under changes of viewpoint and noise. The rather simple formulation of this descriptor has several benefits: it can be extended with additional features in the future, it can be used as keypoint detector thanks to its underlying statistical notions, and the computational cost is low.

Our results have been presented in conjunction with a database of twelve scenes which include variant objects in order to represent handicaps of repeated textures, homogeneous regions and symmetric areas. We have demonstrated how the proposed descriptor has a representative and discriminative capability which outperforms other state-of-the-art methods, specially in the case of noise over data.

Also an algorithm for the detection of planes is presented. This algorithm can be specially useful for registrations containing buildings and walls. The detection of these typical forms will allow a filtering of non-static elements (e.g. cars and persons) and thus a better registration between the 3D points sets. For other kind of scenarios also different typical forms could be studied in the future, like cylinders (for trees, streetlight or traffic lights) or any other forms that could be representative for different objects present in typical scenarios.



(a) (b)

(c) (d)

Figure 3.20: Results of the registrations for (a) *Scan1* against *Scan2*, (b) *Scan1* against *Scan3*, (c) *Scan2* against *Scan3* and (d) *Scan3* against *Scan4*.

Chapter 4

Multiview registration

The registration of multiple 3D structures in order to obtain a full-side representation of a scene is a long-time studied subject. Even if the multiple pairwise registrations are almost correct, usually the concatenation of them along a cycle produces a non-satisfactory result at the end of the process due to the accumulation of the small errors. This situation can still be worse if, in addition, we have incorrect pairwise correspondences between the views. In this chapter we embed the problem of global multiple views registration into a Bayesian framework, by means of an Expectation-Maximization (EM) algorithm, where pairwise correspondences are treated as missing data and, therefore, inferred through a maximum a posteriori (MAP) process. The presented formulation simultaneously considers uncertainty on pairwise correspondences and noise, allowing a final result which minimizes their negative impact. Experimental results show a reliability analysis of the presented algorithm with respect to the percentage of a priori incorrect correspondences and their consequent effect on the global registration estimation.

4.1 Introduction

As seen in the last chapter, acquisition techniques usually have problems with occluded surfaces or the limited field of view, so it is usually necessary to combine different views of the same object or scenario in order to obtain a full representation. Using this process another problem then arises: the registration of these individual 3D views which will enable, at the end of the process, to a whole 3D reconstruction of the desired object or scene.

First step for this objective is the pairwise registration, already studied in Chapter 3. Depending if the two structures have overlap, pairwise registration methods will give as a result a transformation which registers the first 3D view to the second one.

Multiview registration is the second step of this process, and is usually a more complex situation. Assuming that pairwise registrations are correct, their concatenation along a cycle will probably result in a non-satisfactory multiview registration because of the accumulation of the different pairwise errors. An example of this effect can be seen in Figure 4.1.

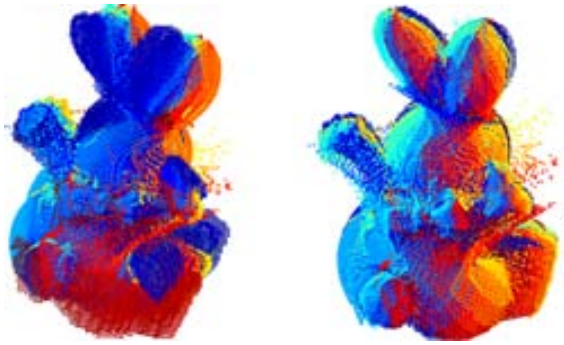


Figure 4.1: Left side: result after applying only the pairwise information for all the views of an object (each color represents a different view). Right side: desired result, where only the noise produced by the sensor can be appreciated.

In addition, there could exist another situation which produces more problems. Even if two structures register perfectly in the pairwise registration process, their transformation could be incorrect in a global environment. This could happen specially if we are working with objects with symmetries, planes or repetitive patterns. In these cases, most of the current multiview registration algorithms will fail because they are not ready to deal with this kind of errors.

The main contribution of the method presented in this chapter, in opposition to state-of-the-art papers, is the possibility of detecting the incorrect registrations between different views and therefore minimize their impact in the global registration process. This feature is achieved thanks to the use of different weights which encode the reliability we have in the correspondences between the views.

The structure of this chapter is as follows: a review of different registration algorithms is presented in Section 4.2, followed by an introduction to the main problematic of multiview registration in Section 4.3. An existing method which is the basis for our approach is studied in Section 4.4, while our proposed algorithm is explained in detail in Section 4.5. The obtained experimental results are shown in Section 4.6. Finally, conclusions and possible future improvements are explained at the end of the chapter in Section 4.7.

4.2 State of the art

Basically there exist two families of multiview registration algorithms: sequential registration methods and simultaneous registration methods.

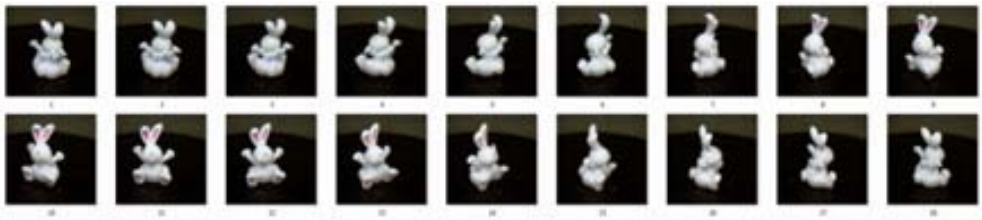
The first ones consist in the sequential pairwise registration of additional views to an

aggregated view which is continuously growing. This aggregated view is also denominated metaview in some literature, as stated in Matabosch et al. [38]. This method was initially proposed by Chen and Medioni in [7] and afterwards improved in the works presented by Pulli [47] and Nuchter et al. [43]. This kind of multiview registration methods have the advantage that they do not need to previously obtain all the views implicated in the registration process, so they do not suffer problems of memory in their execution and can also be useful for applications where the 3D structures are acquired at the same time as the registration is produced, achieving a performance which could be considered near to real-time. However, their main problem is the non-possibility of modifying the already registered views, producing a final result which is usually not as ideal as desired.

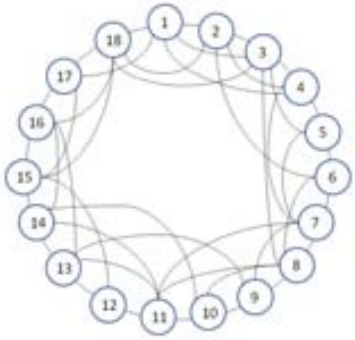
On the other side, simultaneous registration methods make use of all the information of pairwise registrations at the same time. A good example of this kind of method is the work presented by Eggert et al. [14], where the authors present an iterative algorithm which simultaneously updates the transformations of all the views by using the information of position and normal of the correspondence points. In Silva et al. [54] the multiview registration is achieved thanks to the use of genetic algorithms and a metric defined by the authors called Surface Interpenetration Measure, which indicates the level of confidence on a registration according to the multiple crossings of the surfaces between themselves (i.e. the interpenetration between themselves).

Inside the category of simultaneous registration methods a high number of papers base their algorithm on the use of a registration graph which encodes the different pairwise relations between the 3D views. Each node on a registration graph represents an specific view of the object, while each edge encodes the rigid transformation (rotation and translation) between two views. As can be seen in the registration graph of Figure 4.2, each node is connected, at least, with the previous and the following node of the sequence of views and, in addition, with a variable number of additional nodes.

In order to obtain the multiview registration some properties of the registration graph are usually applied, like the property that rotations and translations along a loop of the registration graph should be null. This is the basis for the work presented by Sharp et al. [52], where, for the different basic cycles of the registration graph, the error in rotation and translation is distributed along all the edges. Also the concept of graphs minimization is used in Shih et al. [53] where, using concepts of Lie algebra and circuit theory, the authors develop an algorithm which achieves good results in standard databases, also with the advantage of having a low computational cost. Other papers do not mention explicitly the concept of graph but use it intrinsically, using the same properties and ideas. A good example are the two papers presented by Krishnan et al. [32] [33] which are based on the notion that, concatenating translations and rotations along a path, the same 3D view should be obtained independently on the direction we use to arrive to it. One particularity of this method is that it directly works with the correspondences itself and not with the rigid transformations obtained from the pairwise registration methods, so the algorithm uses all the data without losing any information. This fact causes an apparent complexity of the formulation, but using some concepts of Lie algebra and manifolds the authors achieve a compact representation of the problem. These two papers of Krishnan et al. are used as the basis of our algorithm due to the interesting conceptualization of the problem and the good results obtained after their implementation.



(a)



(b)

Figure 4.2: a) Sequence of views associated to object *bunny*. Only the visible image of each view is shown for a better understanding, every visible image is associated to a range image. b) Registration graph of views associated to the object *bunny*. Every node in the registration graph represents a view and every edge indicates that a pairwise registration between these two views has been estimated. Encoded in the edge the rigid transformation composed by a rotation matrix and a translation vector can be found.

Independently of the election of using a sequential registration method or a simultaneous registration method, one of the main lacks of the aforesaid papers and the majority of other state-of-the-art publications is that they do not take into account the possibility of having absolutely bad registrations or bad correspondences. One of the few papers that deal with this possibility is the work presented by Hubert and Hebert in 2003 [27], where using the joint distribution probabilities the incorrect correspondences are eliminated and, in consequence, the graph can be separated into different splitted sub-graphs which can afterwards be studied individually. As developed in the following sections of the present chapter, our algorithm will also serve for the detection of incorrect correspondences between the views, allowing to minimize the impact of this incorrect information into the global registration process.

4.3 Problematic issues in the multiview registration process

The most usual problem related with the multiview registration process is the minimization of the global error which is produced due to the small errors in every pairwise registration. These errors could be produced by different factors, like a noisy acquisition process, differences in the 3D structure of two overlapping views due to the different date of acquisition, or, in most of the cases, by an inaccurate selection of correspondences. Having a look at the example already shown in the left side of Figure 4.1 it can be easily seen that the result using only the information from the pairwise registration is not enough in order to obtain a good result.

There is another possible problem in the multiview registration process which is in fact more problematic and difficult to solve: the possibility of having not only small errors on the registrations but also absolutely incorrect registrations. Consider the example shown in Figure 4.3, where different views of a horn are shown. Pairwise registration between different views can be correctly established in most of the cases, but there are some ones which could cause problems. Imagine the pairwise registration of view number 3 or 4 against view number 15. Although the object has been rotated around its longitudinal axis (the color of the balls inside the horn demonstrate this rotation) a pairwise registration algorithm could estimate a registration which, after applying it with the other pairwise registrations, produce a result similar to the one shown in Figure 4.4. Similar problems can also be produced if there exist objects with symmetries, planes or repetitive patterns.

In order to solve these two types of problems, and especially the second one, we need to combine both local and global information, or in other words, the information from the pairwise registrations and from the whole multiview result. In order to achieve this objective, a novel algorithm is proposed in Section 4.5. However, this algorithm is based in some concepts to the works presented by Krishnan, so first a brief explanation of its procedure is presented in the following Section 4.4

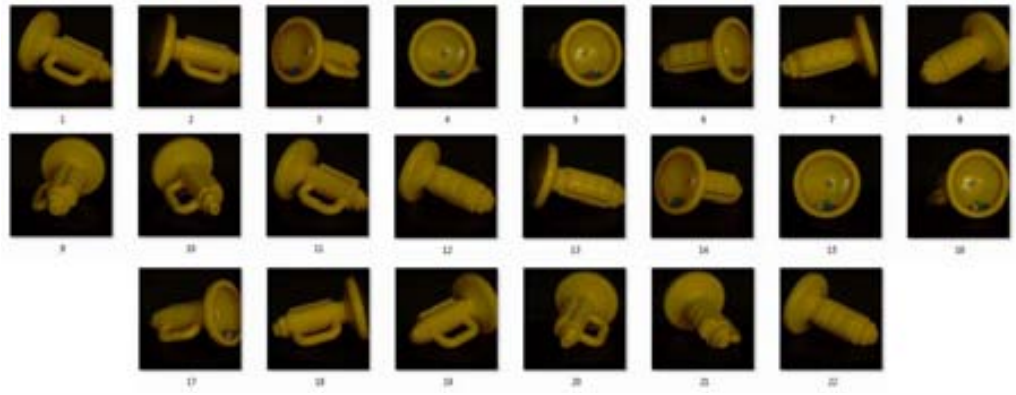


Figure 4.3: Sequence of views associated to the object horn. Images from 1 to 11 are obtained normally by using a swivel platform, from image 11 to 12 the object horn is rotated along his own longitudinal axis, and finally from image 12 to 22 again the swivel platform is used.

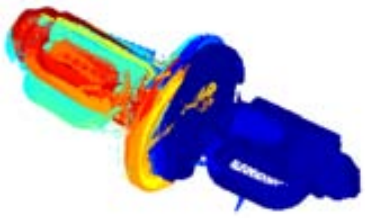


Figure 4.4: Possible result after applying only the pairwise information along a cycle. The axial symmetry of the 3D object causes an incorrect representation even though that the pairwise registrations could seem correct in a local environment.

4.4 Krishnan method: Registration using Optimization-on-a-Manifold

The main disadvantage of most of the multiview registration methods is the use of the results obtained directly from the pairwise registration, that is, the rigid transformations needed in order to register a view with another one. Using only this information we loose all the intrinsic information belonging to the correspondences between the views but, on the other side, we avoid to work with a huge amount of correspondences simultaneously and complicate the algorithm in a high degree.

The Krishnan method explained in this section does indeed achieve this challenge: it estimates the multiview registration working with all the correspondences at the same time. For this reason it is used as basis for our algorithm due to its good results and its simplification of the initial complex formulation, achieving a compact final equation used to estimate the final solution.

4.4.1 Algorithm notation

Let s consider a set of n points $W = \begin{array}{cc} w^1 & w^n \end{array}$ in a world reference frame. These points can be seen from N different views $V_1 \quad V_N$, where $V_i = \begin{array}{cc} v_i^1 & v_i^{n_i} \end{array}$. Each view V_i can only see a limited number of the total n points, so it is supposed that each n_i must be lower or equal to n. In addition, the notation of V_{ij} is used to describe the set of points from V_i which can be seen also from V_j, so in consequence n_{ij} defines the number of points from V_i which can be seen from V_j and therefore $n_{ij} = n_{ji}$.

Each view V_i connects with the world reference frame by means of relative rotation and translation matrices $(R_i \ t_i)$, such that:

$$\begin{array}{l} w^k = R_i v_{ij}^k + t_i \\ w^k = R_j v_{ji}^k + t_j \end{array} \tag{4.1}$$

In a noise free context there would be an equivalence $R_i v_{ij}^k + t_i = R_j v_{ji}^k + t_j$ between both sets of points, but in a more realistic case and considering the presence of noise, the minimization error function used to estimate the parameters $(R_i \ t_i)$ and $(R_j \ t_j)$ takes the following form:

$$error = \sum_{k=1}^{n_{ij}} (R_i v_{ij}^k + t_i) - (R_j v_{ji}^k + t_j)^{\,2} \tag{4.2}$$

If we apply this expression to all the correspondences between the views we will obtain the global error function, called $g(\mathcal{R} \ \mathcal{T})$:

$$g(\mathcal{R} \ \mathcal{T}) = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \sum_{k=1}^{n_{ij}} (R_i v_{ij}^k + t_i) - (R_j v_{ji}^k + t_j)^{\,2} \tag{4.3}$$

, where $\mathcal{R}$ and $\mathcal{T}$ correspond to the two parameters which will be the main variables for the whole process, and are defined as

$$\mathcal{R} \equiv [R_1\ R_2 \quad R_N] \quad \mathbb{R}^{3\times 3N}$$

$$\mathcal{T} \equiv [T_1\ T_2 \quad T_N] \quad \mathbb{R}^{3\times N}$$

Let e_i be the ith column of a $N \times N$ identity matrix, and $e_{ij} = e_i - e_j$. Also let s define:

$$c_{ij}^k \equiv (e_i \otimes I_3)v_{ij}^k - (e_j \otimes I_3)v_{ji}^k$$

where $\otimes$ indicates the *Kronecker product* [15].

The cost function can now be re-written as follows:

$$g(\mathcal{R}\ \mathcal{T}) = \sum_{i=1}^{N}\sum_{j=i+1}^{N}\sum_{k=1}^{m_{ij}} {c_{ij}^k}^T \mathcal{R}^T\mathcal{R}c_{ij}^k + 2e_{ij}^T\mathcal{T}^T\mathcal{R}c_{ij}^k + e_{ij}^T\mathcal{T}^T\mathcal{T}e_{ij} \tag{4.4}$$

Applying the property that $u \cdot v = tr(uv^T)$ we obtain

$$g(\mathcal{R}\ \mathcal{T}) = tr\Big(\mathcal{R} \sum_{i=1}^{N}\sum_{j=i+1}^{N}\sum_{k=1}^{m_{ij}} c_{ij}^k {c_{ij}^k}^T\ \mathcal{R}^T + 2\mathcal{T}e_{ij} \sum_{i=1}^{N}\sum_{j=i+1}^{N}\sum_{k=1}^{m_{ij}} {c_{ij}^k}^T\ \mathcal{R}^T + \\ + \mathcal{T} \sum_{i=1}^{N}\sum_{j=i+1}^{N}\sum_{k=1}^{m_{ij}} {c_{ij}^k}^T\ e_{ij}e_{ij}^T\mathcal{T}^T\Big) \tag{4.5}$$

If we define:

$$\mathcal{A} \equiv \sum_{i=1}^{N}\sum_{j=i+1}^{N}\sum_{k=1}^{m_{ij}} c_{ij}^k {c_{ij}^k}^T$$

$$\mathcal{B} \equiv e_{ij} \sum_{i=1}^{N}\sum_{j=i+1}^{N}\sum_{k=1}^{m_{ij}} {c_{ij}^k}^T$$

$$\mathcal{C} \equiv \sum_{i=1}^{N}\sum_{j=i+1}^{N}\sum_{k=1}^{m_{ij}} e_{ij}e_{ij}^T$$

then, the error function can be represented as

$$g(\mathcal{R}\ \mathcal{T}) = tr\quad [\mathcal{R}\ \mathcal{T}] \begin{pmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{B}^T & \mathcal{C} \end{pmatrix} \begin{bmatrix} \mathcal{R}^T \\ \mathcal{T}^T \end{bmatrix} \tag{4.6}$$

This total registration error function can be written only in terms of rotations $\mathcal{R}$, since the optimal set of translations can be computed in terms of $\mathcal{R}$ as:

$$\mathcal{T}(\mathcal{R}) = -\mathcal{R}\mathcal{B}\mathcal{C} \tag{4.7}$$

where $\mathcal{C}$ is the pseudo-inverse of $\mathcal{C}$. In this case the Equation (4.6) transforms into:

$$g(\mathcal{R}) = tr(\mathcal{R}\mathcal{M}\mathcal{R}^T) \tag{4.8}$$

where $\mathcal{M} \equiv \mathcal{A} - \mathcal{B}\mathcal{C}\ \mathcal{B}^T$ is the Schur complement of the matrix $\begin{pmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{B}^T & \mathcal{C} \end{pmatrix}$.

4.4.2 Initialization

In order to obtain an initial approximation of $\mathcal{R}$ and $\mathcal{T}$, the Krishnan algorithm propose two different situations: the noise free case and the noisy case. In normal situations the noisy case will be always used but the noise free case could serve us as an introduction.

Assuming the noise free case, the Equation (4.8) should be zero, so

$$g(\mathcal{R}) = tr(\mathcal{R}\mathcal{M}\mathcal{R}^T) = vec^T(\mathcal{R}^T)(I_3 \otimes \mathcal{M})vec(\mathcal{R}^T) = 0 \tag{4.9}$$

This implies that

$$vec^T(\mathcal{R}^T)vec(\mathcal{M}\mathcal{R}^T) = 0 \qquad vec(\mathcal{M}\mathcal{R}^T) = 0 \qquad \mathcal{M}\mathcal{R}^T = 0 \tag{4.10}$$

Matrix $\mathcal{M}$ is symmetric, so we can apply SVD:

$$\mathcal{M} = U\Sigma U^T = [U_a U_b] \begin{bmatrix} \Sigma_a & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_a^T \\ U_b^T \end{bmatrix} = \quad \mathcal{M}U_b = 0 \tag{4.11}$$

We could simply determine that $\mathcal{R} = U_b^T$, but we also want to force that $R_1 = I_3$, so

$$\mathcal{R} = \left[\begin{bmatrix} I_3 & 0 \end{bmatrix} U_b\right]^{-T} U_b^T \tag{4.12}$$

In case that we consider the presence of noise, obviously the Equation (4.8) should not be equal to zero. In this case U_b should be the right singular vector associated with the three least singular vectors of $\mathcal{M}$.

$$G_i = \left[\left[\begin{array}{cc} I_3 & 0 \end{array}\right] U_b\right]^{-T} U_b^T (e_i \otimes I_3) \tag{4.13}$$

These singular vectors could not form a real rotation matrix, so for each G_i a projection to the manifold of 3D rotations SO_3 could be necessary:

$$R_i^{opt} = \arg \min_{R_i \ SO_3} \ R_i - G_i \ = \arg \max_{R_i \ SO_3} tr(R_i^T G_i) \tag{4.14}$$

Assuming that we apply an SVD decomposition on G_i, giving $G_i = W \Lambda Z^T$, then the optimal R_i should be

$$R_i^{opt} = W \left[\begin{array}{cc} I_2 & 0 \\ 0 & \det(WZ^T) \end{array}\right] Z^T \tag{4.15}$$

4.4.3 Iteration process

The objective of the iteration process is to find the minimal of the function $g(\mathcal{R})$. Each iteration is composed of two steps, called π_1 and π_2 by the author:

Step π_1: Optimization in local parameter space

The optimization is achieved by using the Newton s method, so we need the first and second derivative of the function. Instead of differentiating directly $g(\mathcal{R})$ we will use a parametrization , which ensures that we are working in the tangent space of the manifold.

$$g(\mathcal{R}) = tr(\mathcal{R}\mathcal{M}\mathcal{R}^T) = \ g \circ \ (\ ) = tr(\mathcal{R} e^{\widetilde{\Omega}(\omega)} \mathcal{M} e^{\widetilde{\Omega}(\omega)^T} \mathcal{R}^T) \tag{4.16}$$

, where $e^{\widetilde{\Omega}(\omega)}$ corresponds to the exponential map of the Lie Algebra of SO_3 [23].

A point $\mathcal{R}$ on the product manifold SO_3^N is mapped to the affine tangent space that minimizes $(g \circ \ )(0)$. We need to estimate an optimal direction $_{opt}$ and a step length λ_{opt} which minimize the cost function. The first derivative of $g \circ \ (0)$ is denoted by $(g \circ \ )(0)$, while the second derivative is denoted by $H_{(f \circ \varphi)(0)}$, so, according to the Newton s method the optimal direction should be:

$$_{opt} = H^{-1}_{(f \circ \varphi)(0)} \ (g \circ \ )(0) \tag{4.17}$$

Once the optimal direction is estimated, it is necessary to find the step length which ensures reduction in the cost function. For this purpose it is used the Backtracking

Line Search [46]. Starting with a value of λ equal to 1, the algorithm increases this value while the following condition is maintained

$$g \circ \quad _{\mathcal{R}}(\lambda \quad _{opt}) > g \circ \quad _{\mathcal{R}}(0) + \alpha\lambda[\quad _{g\circ\varphi_{\mathcal{R}}(0)}]^T \quad _{opt} \tag{4.18}$$

, where $\alpha \quad (0 \;\; 0\; 5)$.

Step π_2: Projection to the manifold

Once the optimal direction and step length have been estimated, we need to project the resulting values to the manifold SO_3. Although we have worked in the tangent space of the manifold, probably the resulting values do not lie in the manifold itself, so it must be ensured that the resulting matrices of $\mathcal{R} = [R_1 \;\; R_2 \quad R_N]$ are real rotation matrices. In order to project them to the manifold, we apply the exponential map from the tangent space to the manifold.

$$\mathcal{R} = \mathcal{R}(e^{\Omega(\lambda_{opt}\omega_1^{opt})} \oplus \quad \oplus e^{\Omega(\lambda_{opt}\omega_N^{opt})}) \tag{4.19}$$

4.5 Bayesian-Based Multiview Registration method

In this Section our developed method for the 3D registration of multiple views is explained, which will be called Bayesian-Based Multiview Registration (BBMR) method in the following. As an introduction to the problem, we first describe the differences in the notation used with respect to the Krishnan method and in a second subsection the proposed Bayesian framework and its resolution using the Expectation Maximization algorithm is presented.

4.5.1 Introducing the correspondence uncertainty matrix

In order to evaluate the possible incorrect correspondences, let s assume that there is no prior knowledge between points from two different views, and therefore, the model used from Krishnan in Equation (4.3) should be generalized in the following way:

$$= \sum_{i=1}^{N} \sum_{j=i+1}^{N} \sum_{k=1}^{n_{ij}} \sum_{k'=1}^{n_{ji}} {}^{kk'}_{ij} \left(R_i v_{ij}^k + t_i\right) - \left(R_j v_{ji}^{k'} + t_j\right)^2 \tag{4.20}$$

where ${}^{kk'}_{ij}$ is an element of the weight matrix that indicates the degree of confidence for a correspondence of points v_{ij}^k and $v_{ji}^{k'}$.

As a simplification of the expression, we can define a new index k as a re-ordered version of the double indexing k and k . Applying this indexing change, Equation (4.20) would be expressed as:

$$= \sum_{i=1}^{N} \sum_{j=i+1}^{N} \sum_{\widehat{k}=1}^{m_{ij}} {}_{ij}^{\widehat{k}} \left(R_i v_{ij}^{\widehat{k}} + t_i\right) - \left(R_j v_{ji}^{\widehat{k}} + t_j\right)^2 \tag{4.21}$$

where $m_{ij} \equiv n_{ij} \times n_{ji} = n_{ij}^2$

4.5.2 Bayesian framework

Let s represent a point correspondence between two specific views V_i and V_j as $x_{ij}^{\widehat{k}} = (v_{ij}^k \;\; v_{ji}^{k'})$, and let $\theta_{ij} = (R_i \;\; t_i \;\; R_j \;\; t_j)$ represent the model parameters of these two views which should be estimated. The objective is to maximize the joint likelihood distribution $P(X_{ij}\; \theta_{ij})$ of the observed data $X_{ij} = [x_{ij}^1 \quad x_{ij}^{m_{ij}}]$.

The optimization of $P(x_{ij}^1 \quad x_{ij}^{m_{ij}}\; \theta_{ij})$ can be a difficult task if pairwise relations are unknown. In order to solve this problem, we introduce a set of binary latent variables $Z_{ij} = [z_{ij}^1 \quad z_{ij}^{m_{ij}}]$, which can be considered as indicator variables of the correspondence between a pair of points $x_{ij}^{\widehat{k}} = (v_{ij}^k \;\; v_{ji}^{k'})$. In an ideal case, where the correspondences were "a priori" known, $z_{ij}^{\widehat{k}}$ should be 1 only for those valid $x_{ij}^{\widehat{k}}$ correspondences and 0 for the rest of pairs of points between the sets V_i and V_j. However, the variables Z_{ij} are hidden or, in other words, cannot be directly observed. If their value was known, then $X_{ij} \;\; Z_{ij}$ would be considered as the complete data set, and therefore, this estimation problem could be easily solved.

Assuming conditional independence on the observations $X_{ij} = [x_{ij}^1 \quad x_{ij}^{m_{ij}}]$ and the latent variables $Z_{ij} = [z_{ij}^1 \quad z_{ij}^{m_{ij}}]$, the joint distribution of the complete data set factorizes as follows:

$$P(X_{ij} \;\; Z_{ij}\; \theta_{ij}) = \prod_{\widehat{k}=1}^{m_{ij}} P(x_{ij}^{\widehat{k}} \;\; z_{ij}^{\widehat{k}}\; \theta_{ij}) \tag{4.22}$$

which according to the binary nature of the latent variables Z takes the following form:

$$P(X_{ij} \;\; Z_{ij}\; \theta_{ij}) = \prod_{\widehat{k}=1}^{m_{ij}} P(x_{ij}^{\widehat{k}}\; \theta_{ij})^{z_{ij}^{\widehat{k}}} \tag{4.23}$$

Applying the logarithm we obtain the joint log-likelihood of the data set as follows:

$$\log[P(X_{ij} \;\; Z_{ij}\; \theta_{ij})] = \sum_{\widehat{k}=1}^{m_{ij}} z_{ij}^{\widehat{k}} \log[P(x_{ij}^{\widehat{k}}\; \theta_{ij})] \tag{4.24}$$

where, if Z_{ij} is known a priori, only the terms indicating correspondence between pairs of points, i.e. $z_{ij}^{\widehat{k}} = 1$ would contribute to the summation of this log-likelihood function. In addition, if we consider Equation (4.21) as the result of a negative logarithm of a Gaussian

distribution, we can see a convergence of the probabilistic formulation in Equation (4.24) and the previous least squares error problem:

$$\log[P(x_{ij}^{\widehat{k}} = [v_{ij}^{k} \;\; v_{ji}^{k'}]\; \theta_{ij})] \quad - (R_i v_{ij}^{k} + t_i) - (R_j v_{ji}^{k'} + t_j)^{\,2} \tag{4.25}$$

and therefore:

$$\sum_{\widehat{k}=1}^{m_{ij}} z_{ij}^{\widehat{k}} \log\left[P(x_{ij}^{\widehat{k}} = [v_{ij}^{k}, v_{ji}^{k'}]|\theta_{ij})\right] \propto -\sum_{i=1}^{N}\sum_{j=i+1}^{N}\sum_{\widehat{k}=1}^{m_{ij}} \omega_{ij}^{\widehat{k}} |(R_i v_{ij}^{k} + t_i) - (R_j v_{ji}^{k'} + t_j)|^2 \tag{4.26}$$

The context of observations X_{ij} and the simultaneous inference of latent variables Z_{ij} and estimation of model parameters θ_{ij} can be tackled by means of the Expectation-Maximization (EM) algorithm [13] [39]. The EM algorithm is a fixed-point fashion procedure that operates in two steps, which are repeated alternatively until convergence:

1. Expectation

Given an intermediate iteration step l with an instance of the model parameters θ^l, compute the posterior probability for latent variables z_i.

2. Maximization

Given the posterior probability $P(Z \; X \; \theta^l)$, compute the expected value of the joint log-likelihood of the complete data set $X \;\; Z$ and find the parameters θ^{l+1} that maximize it. As pointed out in Appendix A:

$$\theta^{l+1} = \arg\max_{\theta} \sum_{\widehat{k}=1}^{m_{ij}} P(z^{\widehat{k}} \; x^{\widehat{k}} \; \theta^l) \log[P(x^{\widehat{k}} \; \theta)] \tag{4.27}$$

Following the equivalence in Equation (4.25) the maximization process could be expressed as follows:

$$\theta^{l+1} = \arg\max_{\theta} \Big\{ -\sum_{\widehat{k}=1}^{m_{ij}} P(z^{\widehat{k}}|v_{ij}^{k}, v_{ji}^{k'}, \theta^l)|(R_i v_{ij}^{k} + t_i) - (R_j v_{ji}^{k'} + t_j)|^2 \Big\} \tag{4.28}$$

where $\theta^l = [R_i^l \;\; R_j^l \;\; t_i^l \;\; t_j^l]$.

In this sense, we can see that the estimation of the posterior probabilities and the optimization of the model parameters according to the weights provided by these posterior probabilities has the property of converging to a local maximum. Moreover, Equation (4.28) shows that this formulation is equivalent to robust statistics weighted techniques, where the posteriors $P(z^{\widehat{k}}\ v_{ij}^{k}\ v_{ji}^{k'}\ \theta^{l})$ take care of the contribution of each observation $(v_{ij}^{k}\ v_{ji}^{k'})$ with respect to the estimated model θ^{l}. In the following, these two iterated steps are developed for the specific case of the multiview registration explained in this chapter.

Expectation step

In the Expectation step the objective is to compute the sufficient statistics for the latent variables posterior distributions $P(Z_{ij}\ X_{ij}\ \theta_{ij})$, in order to infer Z_{ij} from the observations X_{ij} and the parameters of the model θ_{ij}. Inference occurs when computing the values for $z_{ij}^{\widehat{k}}$ that maximize the a posteriori probability of a given data point $x_{ij}^{\widehat{k}}$ and a specific instance for the model s parameters θ_{ij}.

In particular, when inspecting the joint distribution in Equation (4.23) and taking into account the Bayes theorem, we can see that the form of the posterior probability $P(z_{ij}^{\widehat{k}} = 1\ x_{ij}^{\widehat{k}}\ \theta_{ij})$ is proportional to $P(x_{ij}^{\widehat{k}}\ \theta_{ij})^{z_{ij}^{\widehat{k}}=1}$. If we consider the following approximation:

$$P(z_{ij}^{\widehat{k}} = 1\ x_{ij}^{\widehat{k}}\ \theta_{ij}) \quad P(x_{ij}^{\widehat{k}}\ \theta_{ij}) \tag{4.29}$$

then Equation (4.28) would penalize all those pairs of points $(v_{ij}^{k}\ v_{ji}^{k'})$ with biggest error, while it would consider with higher priority the pairs with lower error in the next estimation iteration of θ. Given this behavior and the empirical observations showing an exponential decay of the posterior probability (these observations will be shown in the experimental results in Section 4.6), we can model it as a negative exponential distribution:

$$P(z_{ij}^{\widehat{k}} = 1\ v_{ij}^{k}\ v_{ji}^{k'}\ \theta_{ij}) \quad \exp\ -\alpha\ (v_{ij}^{k}\ v_{ji}^{k'}\ \theta_{ij}) \tag{4.30}$$

where α corresponds to the **suf cient statistics** of the exponential distribution. In an ideal case without uncertainties in the correspondences and therefore without weight values this value would be estimated as:

$$\alpha = \frac{{}_{i}\ {}_{j}1}{{}_{i}\ \ {}_{j}\ (v_{ij}^{k}\ v_{ji}^{k'}\ \theta^{l})} \tag{4.31}$$

In our case we want to give more importance to the correspondences with higher weight values ${}_{ij}^{\widehat{k}}$, so this equation would be adapted as:

$$\alpha = \frac{{}_{i}\ {}_{j}\mathbb{E}\ \ {}_{ij}^{\widehat{k}}\big|P(\ {}_{ij}^{\widehat{k}}\ v_{ij}^{k}\ v_{ji}^{k'}\ \theta^{l})}{{}_{i}\ \ {}_{j}\mathbb{E}\ \ {}_{ij}^{\widehat{k}}\big|P(\ {}_{ij}^{\widehat{k}}\ v_{ij}^{k}\ v_{ji}^{k'}\ \theta^{l})\ \ (v_{ij}^{k}\ v_{ji}^{k'}\ \theta^{l})} \tag{4.32}$$

where $\mathbb{E}\ \ {}_{ij}^{\widehat{k}}\big|P(\ {}_{ij}^{\widehat{k}}\ v_{ij}^{k}\ v_{ji}^{k'}\ \theta^{l})$ is the posterior expectation of the unobserved ${}_{ij}^{\widehat{k}}$ indicator variables, which have been inferred from the model θ^{l} and the observed data $(v_{ij}^{k}\ v_{ji}^{k'})$:

$$\mathbb{E}\Big[\omega_{ij}^{\widehat{k}}\big|P(\omega_{ij}^{\widehat{k}}|v_{ij}^{k},v_{ji}^{k'},\theta^{l})\Big] = \sum_{\omega_{ij}^{\widehat{k}}=0}^{1}\omega_{ij}^{\widehat{k}}P(\omega_{ij}^{\widehat{k}}|v_{ij}^{k},v_{ji}^{k'},\theta^{l}) = P(\omega_{ij}^{\widehat{k}}=1|v_{ij}^{k},v_{ji}^{k'},\theta^{l}) \tag{4.33}$$

Maximization step

According to the Maximization step, let s consider the cost function to be optimized:

$$E(\theta) = -\sum_{i=1}^{N}\sum_{j=i+1}^{N}\sum_{\widehat{k}=1}^{m_{ij}}\mathbb{E}\ \ (v_{ij}^{k}\ v_{ji}^{k'}\ \theta_{ij})\big|P(\ {}_{ij}^{\widehat{k}}=1\ v_{ij}^{k}\ v_{ji}^{k'}\ \theta^{l}) \tag{4.34}$$

which corresponds to the posterior expectation of:

$$\begin{aligned}\varepsilon &= \sum_{i=1}^{N}\sum_{j=i+1}^{N}\sum_{\widehat{k}=1}^{m_{ij}}\omega_{ij}^{\widehat{k}}|(R_i v_{ij}^{k}+t_i)-(R_j v_{ji}^{k'}+t_j)|^2 = \\ &= \sum_{i=1}^{N}\sum_{j=i+1}^{N}\sum_{\widehat{k}=1}^{m_{ij}}\omega_{ij}^{\widehat{k}}((R_i v_{ij}^{k}-R_j v_{ji}^{k'})^2+2(t_i-t_j)^T(R_i v_{ij}^{k}-R_j v_{ji}^{k'})+|t_i-t_j|^2)\end{aligned} \tag{4.35}$$

At this point, a similar sequence of equations than Krishnan method is used. Although they can be reviewed in the previous sheets of this chapter, they are shown again in order to ensure the readability of the whole process. Now we can define:

$$\mathcal{R} \equiv [R_1\ R_2 \quad R_N] \quad \mathbb{R}^{3\times 3N}$$

$$\mathcal{T} \equiv [T_1\ T_2 \quad T_N] \quad \mathbb{R}^{3\times N}$$

and let e_i be the ith column of a $N \times N$ identity matrix, and $e_{ij} = e_i - e_j$.

Also let s define:

$$c_{ij}^{\widehat{k}} \equiv (e_i \otimes I_3)v_{ij}^k - (e_j \otimes I_3)v_{ji}^{k'}$$

where $\otimes$ indicates the *Kronecker product* [15].

The cost function after these changes of notation can be seen as:

$$= \sum_{i=1}^{N}\sum_{j=i+1}^{N}\sum_{\widehat{k}=1}^{m_{ij}} {}^{\widehat{k}}_{ij}\ c_{ij}^{\widehat{k}\,T}\mathcal{R}^T\mathcal{R}c_{ij}^{\widehat{k}} + 2e_{ij}^T\mathcal{T}^T\mathcal{R}c_{ij}^{\widehat{k}} + e_{ij}^T\mathcal{T}^T\mathcal{T}e_{ij} \tag{4.36}$$

So, applying the property that $u \cdot v = tr(uv^T)$:

$$= tr\Big(\mathcal{R}\sum_{i=1}^{N}\sum_{j=i+1}^{N}\sum_{\widehat{k}=1}^{m_{ij}} {}^{\widehat{k}}_{ij}c_{ij}^{\widehat{k}}c_{ij}^{\widehat{k}\,T}\ \mathcal{R}^T + 2\mathcal{T}e_{ij}\sum_{i=1}^{N}\sum_{j=i+1}^{N}\sum_{\widehat{k}=1}^{m_{ij}} {}^{\widehat{k}}_{ij}c_{ij}^{\widehat{k}\,T}\ \mathcal{R}^T + \\ + \mathcal{T}\sum_{i=1}^{N}\sum_{j=i+1}^{N}\sum_{\widehat{k}=1}^{m_{ij}} {}^{\widehat{k}}_{ij}c_{ij}^{\widehat{k}\,T}\ e_{ij}e_{ij}^T\mathcal{T}^T\Big) \tag{4.37}$$

Now, we can define the following elements:

$$\mathcal{A} \equiv \sum_{i=1}^{N}\sum_{j=i+1}^{N}\sum_{\widehat{k}=1}^{m_{ij}} {}^{\widehat{k}}_{ij}c_{ij}^{\widehat{k}}c_{ij}^{\widehat{k}\,T}$$

$$\mathcal{B} \equiv e_{ij}\sum_{i=1}^{N}\sum_{j=i+1}^{N}\sum_{\widehat{k}=1}^{m_{ij}} {}^{\widehat{k}}_{ij}c_{ij}^{\widehat{k}\,T}$$

$$\mathcal{C} \equiv \sum_{i=1}^{N} \sum_{j=i+1}^{N} \sum_{\widehat{k}=1}^{m_{ij}} \overset{\widehat{k}}{ij} \; e_{ij} e_{ij}^T$$

and represent the error function as

$$= tr \quad [\mathcal{R} \; \mathcal{T}] \begin{pmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{B}^T & \mathcal{C} \end{pmatrix} \begin{bmatrix} \mathcal{R}^T \\ \mathcal{T}^T \end{bmatrix} \tag{4.38}$$

As can be seen, the obtained Equation (4.38) corresponds to the same error function presented by Krishnan in Equation (4.6), but in this case the components $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ integrate inside them the uncertainty of the weight coefficients $\overset{\widehat{k}}{ij}$.

Finally, like in the Krishnan method the set of translations can be expressed in terms of $\mathcal{R}$ as:

$$\mathcal{T}(\mathcal{R}) = -\mathcal{R}\mathcal{B}\mathcal{C}^{\dagger} \tag{4.39}$$

where $\mathcal{C}^{\dagger}$ is the pseudo-inverse of $\mathcal{C}$, and

$$= tr(\mathcal{R}\mathcal{M}\mathcal{R}^T) \tag{4.40}$$

where $\mathcal{M} \equiv \mathcal{A} - \mathcal{B}\mathcal{C}^{\dagger}\mathcal{B}^T$ is the Schur complement of the matrix $\begin{pmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{B}^T & \mathcal{C} \end{pmatrix}$.

According to the EM algorithm formulation, the Maximization step will be the posterior expectation of Equation (4.40):

$$E(\theta) = -\sum_{i=1}^{N} \sum_{j=i+1}^{N} \sum_{\widehat{k}=1}^{m_{ij}} \mathbb{E}\big[\; (v_{ij}^k \;\; v_{ji}^{k'} \; \theta) \big| P(\;\; \overset{\widehat{k}}{ij} = 1 \; v_{ij}^k \;\; v_{ji}^k \;\; \theta^l) \big] =$$

$$= tr\Big(\mathcal{R}\mathbb{E}\big[\mathcal{M}\big|P(\;\; \overset{\widehat{k}}{ij} = 1 \; v_{ij}^k \;\; v_{ji}^{k'} \;\; \theta^l)\big]\mathcal{R}^T\Big) \tag{4.41}$$

Note that, since the connection weights $\widehat{k}_{ij}$ are not directly observable, the elements of $\mathbb{E}[\mathcal{M}\ P(\ \widehat{k}_{ij} = 1\ v_{ij}^{k}\ v_{ji}^{k'}\ \theta^{l})]$ matrix must be estimated in each iteration process of the optimization algorithm, i.e.:

$$\mathbb{E}[\mathcal{A}] = \sum_{i=1}^{N}\sum_{j=i+1}^{N}\sum_{\widehat{k}=1}^{m_{ij}} P(\ \widehat{k}_{ij}\ v_{ij}^{k}\ v_{ji}^{k'}\ \theta^{l}) c_{ij}^{\widehat{k}} c_{ij}^{\widehat{k}\,T} \tag{4.42}$$

$$\mathbb{E}[\mathcal{B}] = e_{ab} \sum_{i=1}^{N}\sum_{j=i+1}^{N}\sum_{\widehat{k}=1}^{m_{ij}} P(\ \widehat{k}_{ij}\ v_{ij}^{k}\ v_{ji}^{k'}\ \theta^{l}) c_{ij}^{\widehat{k}\,T} \tag{4.43}$$

$$\mathbb{E}[\mathcal{C}] = \sum_{i=1}^{N}\sum_{j=i+1}^{N}\sum_{\widehat{k}=1}^{m_{ij}} P(\ \widehat{k}_{ij}\ v_{ij}^{k}\ v_{ji}^{k'}\ \theta^{l})\ e_{ij} e_{ij}^{T} \tag{4.44}$$

where $P(\ \widehat{k}_{ij} = 1\ v_{ij}^{k}\ v_{ji}^{k'}\ \theta^{l}) \quad \exp\ -\alpha\ (v_{ij}^{k}\ v_{ji}^{k'}\ \theta^{l})$ and α is obtained from the **Expectation step** in Equation (4.32).

4.5.3 Algorithm summary

As a summary of the whole process, in order to implement the proposed method the schema shown in Algorithm 1 should be followed

Initialization

- Obtain an initial estimate for $\theta^0 = [R_1^0, ..., R_N^0, t_1^0, ..., t_N^0]$ using the original Krishnan method, which corresponds to an ideal case without correspondences uncertainty
- Initialize the correspondences uncertainty matrix $\omega_{ij}^{\widehat{k}}$ to 1 for all the correspondences

while $\|\varepsilon(v_{ij}^k, v_{ji}^{k'}|\theta^{l+1}) - \varepsilon(v_{ij}^k, v_{ji}^{k'}|\theta^l)\| > \xi$ **do**

Expectation

- Find the sufficient statistics of the exponential distribution shown in Equation (4.30):

$$\alpha = \frac{\sum_{i=1}^{N}\sum_{j=i+1}^{N}\mathbb{E}\big[\omega_{ij}^{\widehat{k}}\big|P(\omega_{ij}^{\widehat{k}}|v_{ij}^k, v_{ji}^{k'}, \theta^l)\big]}{\sum_{i=1}^{N}\sum_{j=i+1}^{N}\mathbb{E}\big[\omega_{ij}^{\widehat{k}}\big|P(\omega_{ij}^{\widehat{k}}|v_{ij}^k, v_{ji}^{k'}, \theta^l)\big]\varepsilon(v_{ij}^k, v_{ji}^{k'}|\theta^l)}$$

- For each pairwise correspondence used in the registration process, infer the new uncertainty factor

$$\omega_{new} = \omega_{old}\frac{\alpha\exp\{-\alpha\varepsilon(v_{ij}^k, v_{ji}^{k'}|\theta_{ij})\}}{\sum_{i=1}^{N}\sum_{j=i+1}^{N}\alpha\varepsilon(v_{ij}^k, v_{ji}^{k'}|\theta_{ij})}$$

, where, for simplicity, ω_{new} indicates $[\omega_{ij}^{\widehat{k}}]^{l+1}$ and ω_{old} indicates $[\omega_{ij}^{\widehat{k}}]^l$

Maximization

- Calculate the matrices $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$:

$$\mathcal{A} \equiv \Big[\sum_{i=1}^{N}\sum_{j=i+1}^{N}\sum_{\widehat{k}=1}^{m_{ij}}\omega_{ij}^{\widehat{k}}c_{ij}^{\widehat{k}}{c_{ij}^{\widehat{k}}}^T\Big]$$

$$\mathcal{B} \equiv e_{ij}\Big[\sum_{i=1}^{N}\sum_{j=i+1}^{N}\sum_{\widehat{k}=1}^{m_{ij}}\omega_{ij}^{\widehat{k}}{c_{ij}^{\widehat{k}}}^T\Big]$$

$$\mathcal{C} \equiv \Big[\sum_{i=1}^{N}\sum_{j=i+1}^{N}\sum_{\widehat{k}=1}^{m_{ij}}\omega_{ij}^{\widehat{k}}\Big]e_{ij}e_{ij}^T$$

- Estimate $\theta^{l+1} = [R_1^{l+1}, ..., R_N^{l+1}, t_1^{l+1}, ..., t_N^{l+1}]$ by minimizing Equation (4.40) with the posterior expectation

end

Algorithm 1: Summary of the proposed method

4.6 Experimental results

In this chapter we focus on two main issues: the uncertainty on pairwise correspondences and the combination of global and local information in the registration process of multiple views.

Our technique needs, as a starting point, setting up correspondences between pairs of points belonging to two different overlapping views. These pairwise corre-

spondences may contain some errors due to the fact that they are established by using only the information from the corresponding pair of views, which in practice means using only the local information.

These pairwise point correspondences are taken as input data for our global registration method. The proposed algorithm aims to overcome the uncertainty introduced by the local set up of correspondences between pairs of views, evaluating the contribution of each correspondence into the global registration result. The goal is to find an optimal configuration, in global terms, of Euclidean transformations based on the contribution of each pairwise correspondence. To this end, we introduce a weight factor to each pairwise correspondence so the algorithm can determine which of them have a higher contribution in the parameters estimation.

The experiments have been designed in order to study a) the weighted contribution of each pairwise correspondence into the global estimation process and b) their evolution along the consecutive iterations of our probabilistic method. Both analysis assume that there has been introduced a level of uncertainty in terms of mistaken pairwise correspondences (due to errors in the annotation process). We aim to evaluate the performance of our Bayesian formulation when dealing with this type of uncertainties.

To this end we propose three approaches:

1. First, we study in deep the "horn case" presented in Figures 4.3 and 4.4, where certain pairwise correspondence points have been wrongly set up due to poor local information and symmetries between pairs of views. The goal is to show how the introduction of weighting factors minimize the contribution of those correspondences which are not coherent with the rest according to the global registration result. In this first experiment, we show how an initial result that assumes equal weights for all pairwise correspondences (obtained by using Krishnan technique) leads to a wrong configuration, but it can be afterwards corrected by iterating our Expectation-Maximization process.

2. A second experiment is presented with the aim of evaluating the robustness of the presented Bayesian formulation in terms of manually introducing wrong correspondences in pairs of overlapping views. If we assume that some of the correspondences can be incorrect (for instance, due to human factors), our goal is to study how they can affect to the final registration. In particular, we aim to study the relation between the percentage of incorrect correspondences and the algorithm performance, using manually corrupted data as input parameters.

3. We create a synthetic 3D object with all the necessary a-priori known information, in order to be employed as ground truth. Assuming that all correspon-

dences are correct and that the only source of error is due to noise, we can compare in terms of accuracy Krishnan method and our method by studying how close both estimations are to the ground truth.

The first two experiments have been tested with 3D objects from the Ohio State University (MSU/WSU) Range Image Database [11], used also in the state-of-the-art papers [52] and [53]. The pairwise correspondences between the views have been carried out manually, assuming little errors between the correspondences.

4.6.1 Correction of degraded correspondences - The horn case

In cases where we want to register 3D objects with symmetries, planes or repetitive patterns it is quite probable to obtain incorrect pairwise correspondences. Due to the limited field of view, the pairwise registration between two views should seem correct in a local environment but, in a global environment and considering the rest of the views, this pairwise information can be seen as clearly incorrect. We can see an example of this situation in the case explained in Figure 4.4 (the horn case), where a simple incorrect pairwise registration can ruin the whole 3D object.

Our method is applied to this specific horn case. In Figure 4.5 we can see the evolution of the registration along the different iterations of the algorithm presented in this chapter. The first iteration result shown in the figure consists in the execution of the algorithm before applying the use of weights for the correspondences, so in fact it corresponds to the result of the Krishnan algorithm itself. The result using Krishnan method is not satisfactory at all, but is quite near to the desired result because of the fact that there exist only two or three incorrect correspondences among the total number of 196 correspondences. Using our BBMR method the different parts of the model can be registered correctly and approximately at iteration 5 we can achieve a good registration, and finally at iteration 6 the algorithm converges according to the condition specified in Algorithm 1.

Also in Figure 4.5 the evolution of the error is presented, where $(v_{ij}^k \; v_{ji}^k \; \theta^l)$ is evaluated for the 196 different correspondences of the model and displayed in a histogram. As can be seen in the figure, at iteration 1 the majority of correspondences have their two components near each other, but there exist also some correspondences with a higher distance. Along the different iterations of our BBMR method the weight of these correspondences with higher distance are reduced, preventing their bad influence in the registration process. Looking at the form of the histograms displayed in these iterations (especially in the first and the second one), we can see that the election of a negative exponential distribution in order to model Equation (4.30) was a good approximation.

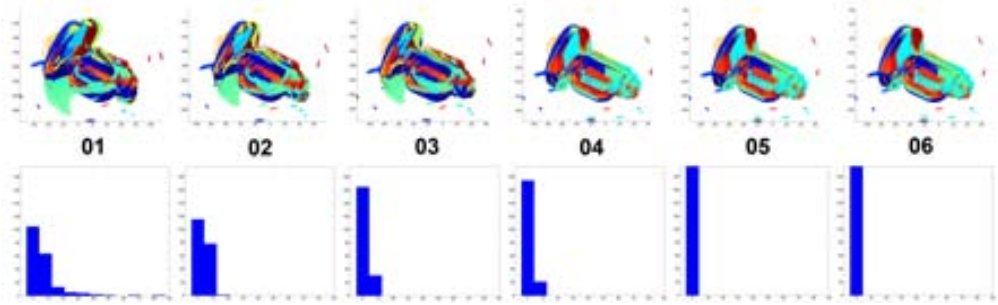


Figure 4.5: Evolution of iterations for the registration of the horn case. The upper row shows the registration for the 6 iterations needed before the algorithm converges. The lower row shows the evolution of the error for the same iterations, confirming the good election of the negative exponential distribution in order to model the behavior of the error.

4.6.2 Correction of degraded correspondences - Percentage evaluation

The experimental result shown in Section 4.6.1 was a specific case which helped us in the understanding of the algorithm evolution. The objective in this second experiment is to evaluate the robustness of the presented method in a higher range of situations, by introducing in the system an increasing number of incorrect correspondences. For this purpose we create a Matlab script which assigns random correspondences according to a specific percentage. Taking into account the total amount of correspondences of an object, increasing percentage from 5 percent to 50 percent of them are corrupted, preserving the first component of the correspondence but changing the second component to a random point of the 3D surface. This new second component should be distanced to the original second component by, at least, a distance equal to a fifth of the distance between the two most distanced points of the 3D surface, otherwise another point is randomly sought. Our BBMR method is then applied 100 times per corruption percentage, and the final result is evaluated in order to check the correct consistency of the obtained 3D surface.

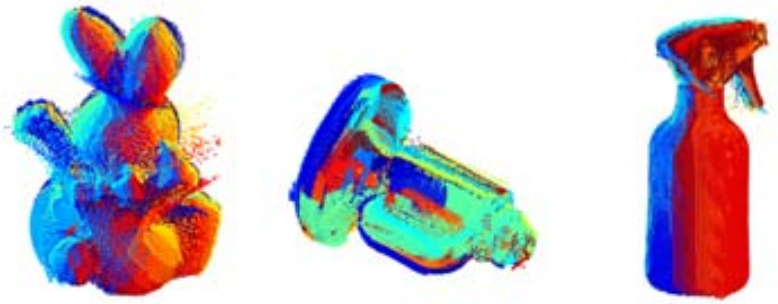
The evaluation of the correctness is carried out by means of calculating the distance of all the 3D points to an already registered 3D object, which has been obtained from a previous registration process using the Krishnan method and without any degraded correspondence (this previously registered 3D object will be taken as ground truth in this experiment). If only one point has a distance value higher to a pre-defined threshold, the whole registration process is discarded. In our experiment this threshold is assigned to a twentieth of the distance between the two most distanced points of the 3D surface.

Three different objects have been used for the experiments, called *bunny*, *horn*

	bunny	*horn*	*bottle*
Total number of points	171085	247947	109453
Distance between the most distanced points	67.06	146.48	198.75
Number of views	18	22	11
Number of correspondences	169	196	122

Table 4.1: Main characteristics of the three objects used in the experiment. The values shown in "Total number of points" correspond to the sum of all the points for all the views, independently if a same 3D point can be seen from different views.

and *bottle*. A preview of them and their main characteristics can be seen in Figure 4.6 and in Table 4.1.



(a) (b) (c)

Figure 4.6: Preview of the three objects used for the experiment: a) *bunny*, b) *horn* and c) *bottle*.

Results obtained after the simulations can be seen in Figure 4.7, where the vertical axis indicates the number of experiments which have been correctly registered among the 100 simulations for each degradation percentage. Both the results of our method and the results of Krishnan method are displayed. Krishnan method is used as a base for our BBMR method but, as expected, is not designed to deal with incorrect correspondences and therefore its results are not satisfactory.

The first observation after looking at Figure 4.7 is that algorithm performance can vary depending on the complexity of the 3D object. Having a look at the characteristics of the 3D objects in Table 4.1 it can be easily seen that the three objects have a similar relation between the total number of points and the number of views but, on the other side, the object *bottle* has a higher number of correspondences in relation to the number of views or the total number of points. Even if a higher number of these correspondences are degraded, the algorithm can achieve a good registration thanks

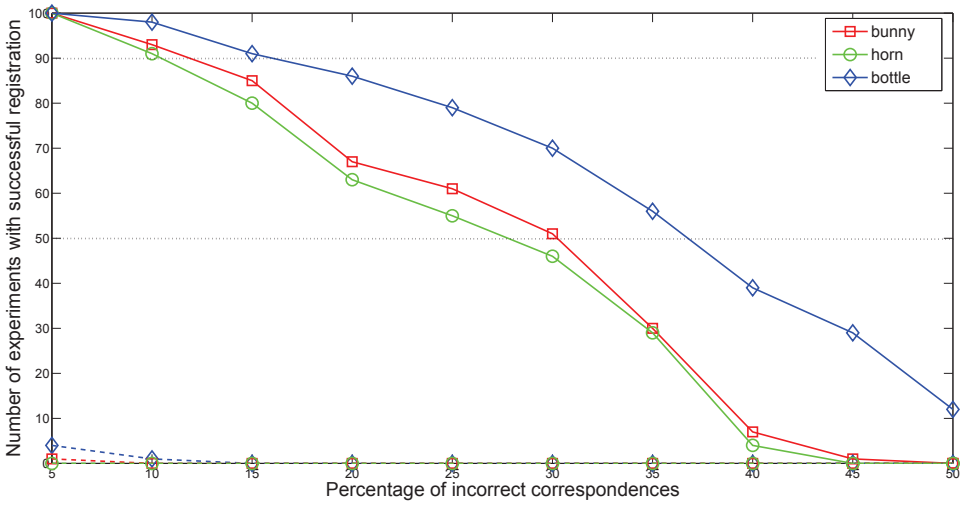


Figure 4.7: Results for the different 3D objects of the database. Horizontal axis indicates the percentage of degraded correspondences applied to the object, and vertical axis indicates the percentage of successful registrations. The solid red, green and blue lines show the performance for our BBMR method, while the dashed red, green and blue lines at the bottom show the result for the Krishnan method. In addition, the horizontal dotted black lines serve only as a reference to indicate the performance levels of 90% and 50%.

to the additional redundancy of the object *bottle* in comparison with the other two objects (speaking in the relative term according to the complexity of the object).

Independently of the complexity of the object, looking at the reference lines displayed on Figure 4.7 it can be observed that our algorithm achieves a 90% of success registrations for cases where there exist approximately 10% or 15% of incorrect correspondences between the 3D views. In addition, around half of the experiments achieve a good registration for degradation percentages between 30% and 35%, i.e. in cases where approximately one out of three correspondences were incorrect. Finally, the algorithm performance is almost null for cases with 50% of degradation percentage and beyond.

In addition to the results displayed in Figure 4.7, it must also be noted that, specially for the experiments with degradation errors below 40%, the incorrect registrations results obtained are usually composed of a high amount of views correctly registered with just one or two views incorrectly aligned, giving a final result which is obviously incorrect but in any case highly better than the result using the original Krishnan method. An example of this event can be seen on Figure 4.8.

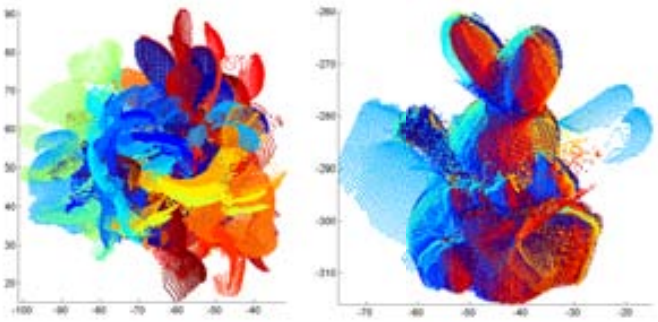


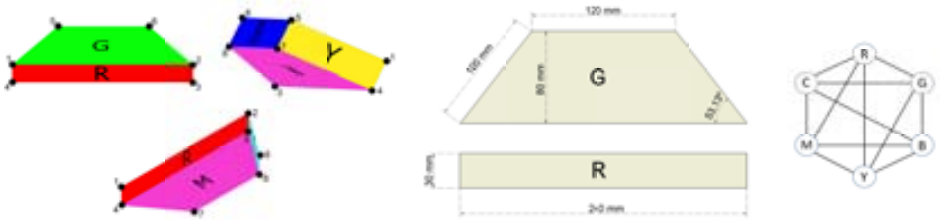
Figure 4.8: Result of the presented algorithm after iteration 1 and iteration 15 for the object *bunny* and a corruption percentage of 35%. The right result, even though is incorrect, is clearly better than the left result (which is in fact the original Krishnan method).

4.6.3 Improvement on the accuracy

The previous experiments were related to the robustness of the presented method in order to deal with incorrect correspondences. This is the main property of our method, but there exists also the possibility of improving the registration in cases where the correspondences are just affected by small deviations produced by the noise, inaccurate manual selection of points or other factors. The iterated processes and the use of weight factors help to improve the final result of the multiview registra-

tion with respect to the Krishnan method, giving more relevance to the most correct correspondences and less relevance to the least ones.

In order to evaluate with precision the accuracy improvement a synthetic 3D model has been created. This synthetic model is taken as ground truth for this experiment. Different views of the model, its measures and the associated registration graph of views can be seen in Figure 4.9. The model is composed by 6 faces (identified by colors R-G-B-C-M-Y) and 8 vertices. The different faces are considered the views of the model, and every face is pairwise registered against its 4 face neighbors by manual correspondence selection, assuming little errors.



(a) (b) (c)

Figure 4.9: Design of the object *syntheticModel* created for the experiment, including a) different perspectives of the model, b) horizontal layout and vertical profile and c) registration graph of views associated to the object.

Two different experiments are carried out with the synthetic model described: a) vertex precision and b) normal vector precision.

(a) Vertex precision

The objective of the experiment is to evaluate the accuracy improvement with respect to the Krishnan method in the different intersections of the synthetic model. According to the structure of the model (see Figure 4.9), each vertex has the intersection of three different faces (for example, vertex number 3 has, as can be seen in Figure 4.10, the intersection of the faces R, M and C), so in total the object *syntheticModel* has 24 intersections.

Our registration method is applied to the synthetic model, and the results obtained are compared with the ones obtained by using the Krishnan method. Specifically, the distances between the vertices of the 24 correspondences are observed, and the results are shown in Figure 4.11.

Results show that our method improves the registration in 20 intersections, but the remaining 4 intersections are getting worse. Averaging the distances we can

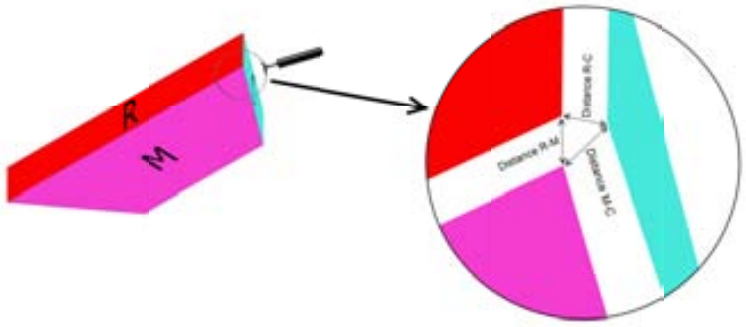


Figure 4.10: Magnification of the vertex number 3 of the object *syntheticModel*. For each vertex of the object there exist three distances that are evaluated, in the case of vertex number 3 we evaluate the distance between faces R and M, between R and C, and between M and C.

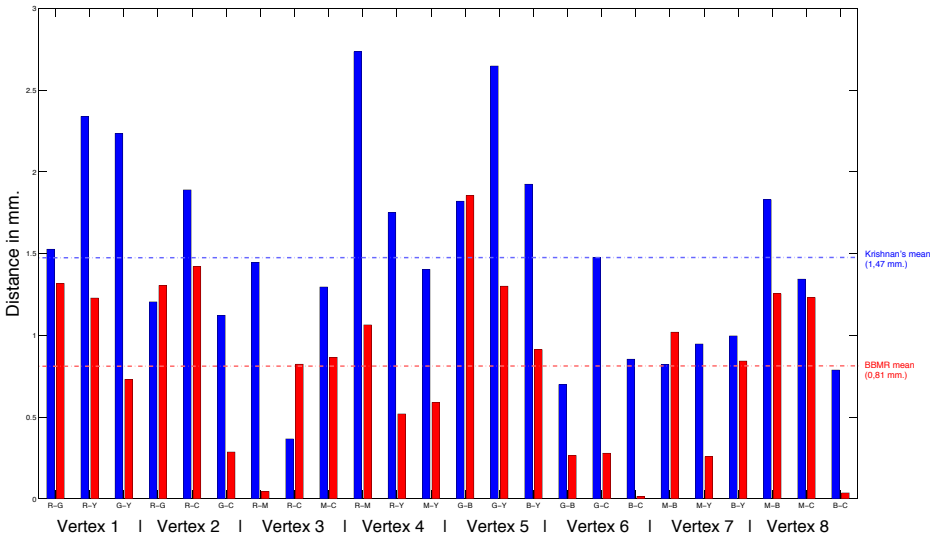


Figure 4.11: Distance difference between Krishnan method and our method. On the horizontal axis the 8 vertices of the object *syntheticModel* are displayed, and each vertex is composed by 3 distances between the faces. Vertical axis indicates these distances in millimeters. Krishnan method obtains a mean distance of 1,47 mm., while our method obtains a mean distance of 0,81 mm.

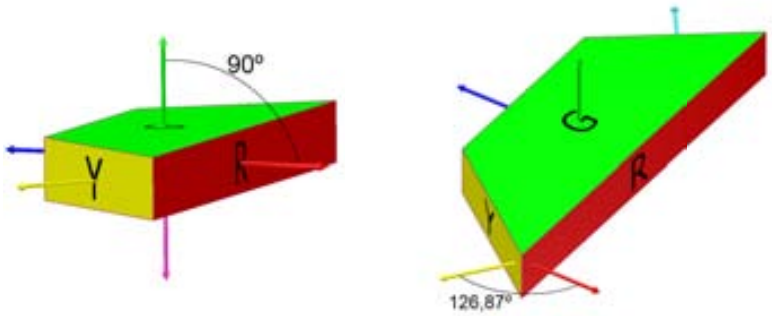


Figure 4.12: Representation of the ideal angle values between the normal vectors of the faces. According to the design of the object *syntheticModel*, the normal vector of face R with the normal vector of face G should form an angle of 90 degrees, and the normal vector of face R with the normal vector of face Y should form an angle of 126,87 degrees.

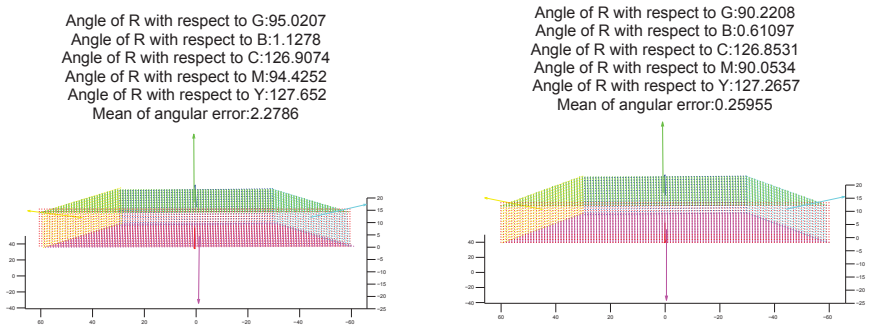
see that Krishnan method obtains a mean value of 1,47 mm. and our method obtains a mean value of 0,81 mm., so a reduction of 44.89% from the initial distance is achieved using our BBMR method.

(b) Normal vector precision

In addition to the vertex precision explained in the previous experiments other characteristics of the synthetic model can be studied after the registration. According to the design of the synthetic model in Figure 4.9, and as can be seen with more detail in Figure 4.12, it must be accomplished that:

- The normal vector of R with respect to the normal vector of G form an angle of 90 degrees.
- The normal vector of R with respect to the normal vector of B form an angle of 180 degrees.
- The normal vector of R with respect to the normal vector of M form an angle of 90 degrees.
- The normal vector of R with respect to the normal vector of C form an angle of 126,87 degrees.
- The normal vector of R with respect to the normal vector of Y form an angle of 126,87 degrees.

The results after the simulations are displayed in Figure 4.13, where the iteration 1 and 18 are shown. Despite that the visual evaluation does not reflect a big



(a) (b)

Figure 4.13: Result of the registration after a) iteration 1 and b) iteration 18. In both cases the upper part of the result display in text the angle of the normal vector of face R with respect to the other normal vectors. The last line "Mean of angular error" indicates the mean of the values obtained by subtracting the experiment values with the ideal values.

difference, the exact measurement shown in the text reflects a better approximation to the ideal values at iteration 18. At iteration 1, which corresponds to the Krishnan method, the mean of angular error value is 2,2786 degrees. At iteration 18, last iteration before the algorithm converges, the mean of angular error value is 0,25955. The evolution of this mean of angular error is shown in Figure 4.14, where it can be seen that the error is decreasing continuously like a negative exponential function.

4.7 Conclusions

This chapter presents our Bayesian-Based Multiview Registration (BBMR) method for the registration of multiple 3D scans. The main property of our BBMR method consists in the property of being tolerant to a certain number of incorrect correspondences which could be caused by different factors like an incorrect manual selection, symmetries on the scanned 3D object or repetitive patterns. This tolerance is achieved thanks to the use of an additional layer placed over an existing multiview registration method, by using weight values which are applied to the point correspondences depending on their reliability. The value of these weights is estimated iteratively by means of a Bayesian framework, and the global registration problem is solved thanks to the Expectation Maximization method.

Results obtained show that the presented algorithm is able to register correctly

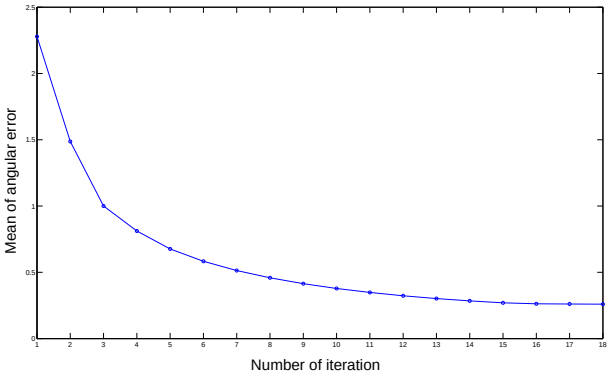


Figure 4.14: Evolution of the mean of angular error for the different iterations of our method, showing an exponential decay until the algorithm converges.

approximately half of the 3D scenes with an incorrect pairwise correspondence tax between 30% and 35%. This result outperforms other existing registration methods, allowing a higher flexibility in the establishment of pairwise correspondences either by manual or automatic selection. In addition, in cases with a low degradation percentage, our BBMR method achieves also a better registration compared to the Krishnan algorithm. As a drawback the algorithm takes a relative high time, as it is basically composed of multiple iterations of this Krishnan algorithm.

Possible studies to develop in the near future include the possibility of defining a distance metric in order to evaluate the compactivity or consistency of the registration result after each iteration of the BBMR method. In the experimental results section we partly solved this situation by comparing the resulting final registration against a pre-registered object, but in a more general situation it is not expected to have access to this reference object. Of course, this distance metric should be defined "from an external point of view" and independent of the beforehand obtained pairwise correspondences, because it could be the case that these pairwise correspondence could be incorrect. Methods like the ones explained by Huber and Hebert in [28] regarding the space violation with respect to the visibility or the Surface Interpenetration Measure presented by Silva et al. in [54] could be used and combined for this purpose.

Chapter 5

Single View System for the Human 3D Modeling

In this chapter we explore the automatic 3D modeling of a person using images acquired from a range camera. Using only one range camera and two mirrors, the objective is to obtain the complete 3D model. The combination of the camera and the two mirrors give us three non-overlapping meshes, making impossible to use common zippering algorithms based on overlapping meshes. Therefore, Dynamic Time Warping algorithm is used to find the best matching between boundaries of the meshes. Experimental results and error evaluations are given to show the robustness and efficiency of our method.

5.1 Introduction

In the previous chapters of this thesis we have studied different possibilities in order to achieve the complete 3D representation of a model or an scene. Usually this final representation is obtained by estimating the pairwise registration between some multiple views and afterwards minimize their global error by using a multiview registration algorithm. In this chapter a different and novel mechanism is presented, achieving the complete 3D representation of the model by using only one single view.

The work presented in this chapter describes the development of an specific real-time 3D modeling system. This system has been mainly designed for human body reconstruction, due to its specific particularities. However, also objects of similar or lower size could be applied.

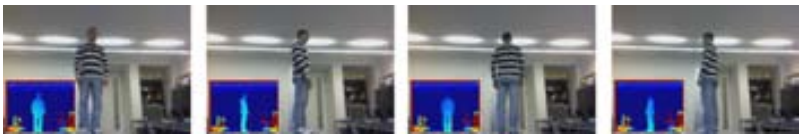


Figure 5.1: Sequence of the scanning process using a turning table example. Only 4 scans are shown, but the sequence can be composed by a large number of scans. For each scan, the RGB image and the depth image are shown.

The presented system consists of two main phases: data acquisition and mesh generation. In the data acquisition phase the 3D information of the whole model is acquired trying to minimize the needed space. In the mesh generation phase we analyze the method for creating a closed mesh based on the particularities of the data acquisition.

5.2 Problems with current used methods

Current existing methods propose different ways to solve the problem of 3D modeling. In this work we study the two more relevant ones: the turntable approach and the multiple cameras approach. Although they are conceptually equivalent, each one of them has their own mains and drawbacks.

(a) Turntable approach

The most common method used in 3D modeling consists on placing the object on a turning table, allowing the capture of the object from several viewpoints. The 3D sensor can be fixed in an appropriate place and successive 3D captures of the object are obtained during its rotation. The result of this scanning process is a set of partial scans of the object, including both the depth and the RGB information. An example using a model person is shown in Figure 5.1.

Once the different partial scans have been obtained, the multiple views are registered together in order to obtain a full-side representation. For this purpose usually a the classical 2-step method including pairwise registration and multiview registration is used.

However, this kind of acquisition method is not suitable for human modeling. Minimal movements of the subject during his rotation can produce errors in the final registration. Also, many people are reluctant to be rotated and this can be a problem for a possible commercial product.

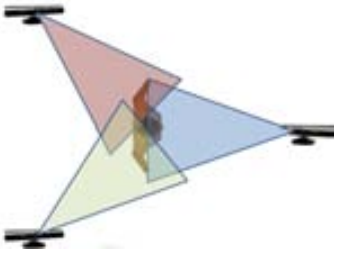


Figure 5.2: Multiple viewpoints surrounding the person in order to cover the 360 degrees. Although only three cameras are shown in the image, this number can be increased.

(b) Multiple cameras approach

In opposition to the rotation of the subject, a similar possibility would be the rotation of the camera around him or, equivalently, the disposition of multiple range cameras surrounding the person as shown in Figure 5.2. The main advantage of this approach is that the person should not be rotated, avoiding then their possible movements.

Although this alternative has a simple implementation it requires a lot of space on the scene. In order to capture the whole height of a medium height person every range camera must be at around 2-3 meters away from the person and this distance must be free of any occluding object.

In addition it must be considered that, depending on the nature of the 3D scanner used, the presence of multiple devices can produce interferences between them. A possible solution would be to take snapshots of the scene for every range camera but at different times, so some kind of synchronization between the devices should be needed.

5.3 Proposed approach

The proposed system is composed of two main phases: the model acquisition phase where the 3D points of the person are captured, and the mesh triangulation and zippering phase, where the mesh is created from the set of 3D points. Both phases are described in the following subsections.

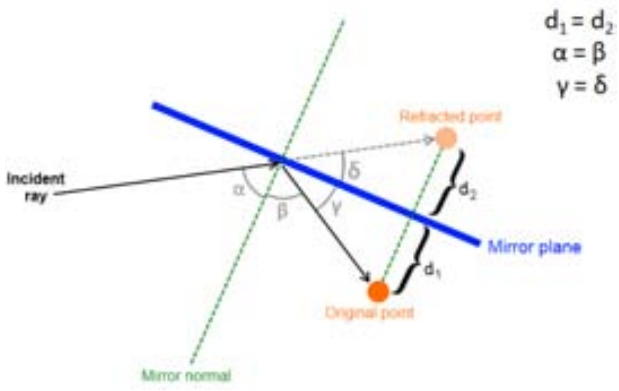


Figure 5.3: Reflection of a single point on the mirror. The original point is placed in front of the mirror, and the incident ray indicating the view of the camera aims to the mirror and can see the original point thanks to the reflection. However, the depth camera only detects a distance to the point, and this distance is placed in straight line according to the direction of the incident ray. According to the ideal reflection rules, the angle α between the incident ray and the normal plane is equal to the angle β produced between the reflected ray and the normal plane. In the same way, the angle γ is equivalent to the angle δ, and therefore d_1 and d_2 have the same distance.

5.3.1 First phase: model acquisition

The acquisition of the data is carried out by the well-known Microsoft Kinect camera. As seen in Chapter 1, this device uses the triangulation between the captured infrared image and a known pattern emitted by the infrared projector in order to estimate the distance to all the points in the scene. This working procedure is the basis for the special technique used in the acquisition phase.

A standard mirror reflects the visible light, but it also reflects the IR light. If we place a Microsoft Kinect pointing to a mirror, the IR pattern emitted by the camera reflects in this mirror and therefore the sensor is able to capture the 3D structure of the objects present in the reflection. This method has however a little disadvantage: the range camera does not recognize that this is a reflected pattern, so it will place the reflected 3D structure in straight line, i.e., at the other side of the mirror.

A simple explanation for a single point is shown in Figure 5.3. As it can be seen in the figure, the Microsoft Kinect will place the distance to the point in straight line and therefore the refracted point will be placed at the other side of the mirror. According to the ideal reflection rules, this new refracted point will be placed in perfect symmetry regarding the original point with respect to the mirror plane.

Applying this theoretical idea to our study case, we can see the frontal view and

(a) (b) (c)

Figure 5.4: In (a), the RGB image and the depth image obtained from Kinect are shown (objects with a depth higher than a threshold have been filtered out in the depth image for a better scene understanding). Fusing the information of both images we can represent the 3D model of the scene, shown in (b) and (c). Although it can not be observed in (b), in (c) is clearly seen that the reflected parts of the person are placed at the other side of the mirrors.

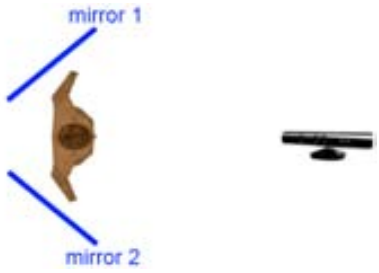


Figure 5.5: Example of the acquisition process setup. Using the reflection of the mirrors, the rest of the body can be inferred.

the two reflected views of a person in Figure 5.4. As expected, the reflected 3D structures are placed accordingly at the other side of the mirror.

The main advantage of this technique is that it reduces the total need of space. As the field of view of the Kinect camera is enough for capturing the whole mirror, the IR pattern is already extended in the reflection plane and therefore we do not need the additional space behind the person. An schema of the final disposition of the elements is shown in Figure 5.5, where the total space needed is significantly lower than in Figure 5.2 and more appropriate for small spaces. For example, it can be appropriated for a dressing room in a clothing store where, in addition, the presence of mirrors can also be useful for their "traditional" purpose.

On the opposite side, also some drawbacks appear by using this technique. The main one is the need of re-positioning the two posterior views of the person, which are

placed on the other side of their respective mirrors. As seen in Figure 5.3, a symmetry with respect to the mirror plane will be enough to align the views. Assuming that $F = \quad f_i \quad$ represent some detected points of the mirror frame, we can compute the matrix M_{ms} by concatenation of the three eigenvectors of the covariance matrix of F. Note that the first too eigenvectors specify the plane containing the two directions with maximum variance of $F = \quad f_i \quad$ and the third eigenvector corresponds to the normal vector of this plane, i.e. the mirror normal.

Matrix M_{ms} converts points from the mirror reference frame to the 3D scanner reference frame so, assuming that we have a point p_s expressed in scanner coordinates we can convert it into mirror coordinates by applying

$$p_m = M_{ms}^{-1}(p_s - \overline{f}) \tag{5.1}$$

, where $\overline{f}$ corresponds to the centroid of all the frame points $F = \quad f_i \quad$. Once we obtain the point p_m expressed in the mirror coordinate frame it is needed, in order to indicate the symmetry with respect to the mirror, to change the sign of its third coordinate value. Finally, the point must be expressed again in the 3D scanner coordinate frame by using the matrix M_{ms}.

In addition to the flip of the points with respect to the mirror plane, two other problems arise with this technique. The first one is produced by the extra distance that the IR pattern must travel after reflecting in the mirror, which produces that the 3D resolution of this pattern will be slightly lower when it illuminates the posterior part of the person. In consequence the posterior views of the 3D modeled person will have a lower resolution in comparison to the frontal view, giving us a model which is not uniform in all its surface.

Finally, another disadvantage that must be taken into account is that using this mirrors technique, we will always obtain three point sets without overlapping regions between them. If a point of the scanned surface is illuminated by the direct IR pattern and a reflected IR pattern, both patterns will interfere themselves and therefore the range camera will not be able to decide which is the correct range. In fact, this is the same effect as if an object is illuminated by two range cameras at the same time or when a camera aims directly at another range camera. This problem is partly solved in the following subsection by using a zippering algorithm.

5.3.2 Second phase: mesh reconstruction

Although existing methods for 3D triangulation produce good results, they usually require a set of 3D points with low noise and, if possible, with a uniform resolution along the object. Images obtained with the Microsoft Kinect sensor are noisy and in

Figure 5.6: Frontal mesh and back-right mesh. Views are intentionally separated in Z axis for better comprehension.

addition the use of mirrors produces an irregular resolution due to the extra distance caused by the reflection, so the triangulation of the set of 3D points obtained from the acquisition phase usually gives a non-satisfactory result.

However, we must consider that the image obtained from the Kinect sensor can be fast triangulated thanks to the ordered 3D points obtained by the IR pattern. In each single snapshot of the sensor we obtain three different point clouds at a time (the frontal point cloud and the two posterior point clouds obtained from the mirrors), so the triangulation can be done individually for each one of them. In Figure 5.6 we can see two of the resulting meshes, which have been previously fast triangulated.

Once these individual triangulations are done, it is necessary a process for connecting the 3 generated meshes: frontal, back-left and back-right. In the literature we can find some works related to stitching meshes [60] [56] [51], but all of them are focused on overlapping meshes. However, as previously explained, our system contains three meshes which cannot overlap and, in addition, they have different resolutions due to the higher distance traveled by the IR pattern in the reflection of the mirrors.

In order to solve these problems an approach for mesh zippering based on Dynamic Time Warping is proposed. Dynamic Time Warping (called DTW in the following) [50] [41] is an algorithm to find the optimal alignment between two sequences. It was designed to compare different speech patterns in automatic speech recognition, but is also usual in fields like handwriting or signature recognition. The objective of DTW is to compare two ordered sequences $X = (x_1\ x_2 \quad x_N)$ and $Y = (y_1\ y_2 \quad y_M)$ of length N and M respectively. To compare two different features $x \quad X$ and $y \quad Y$ a local cost measure $c(x\ y)$ needs to be defined. Evaluating the local cost measure for each pair of elements of the sequences X and Y, the cost matrix $C \quad R^{N \times M}$ is obtained. Having this cost matrix, the optimal alignment

between X and Y can be found by looking for the path along C with minimal cost.

Since mirrors are oriented vertically, the way that the meshes must be joined is through the coronal plane of the person, that is, the vertical plane which divides the human body into front and back. During the process of stitching we have to decide which side of every mesh matches the side of the other mesh. Thus, we need to find the points where coronal and sagittal plane intersect for every mesh.

For every possible match, DTW retrieves a warping matrix and an accumulated distance that brings us the value of similarity between boundaries. This warping matrix stores the correspondences between each point of the sequence. In Figure 5.7 a distance matrix is shown, where the horizontal axis corresponds to the first sequence and the vertical axis corresponds to the second one.

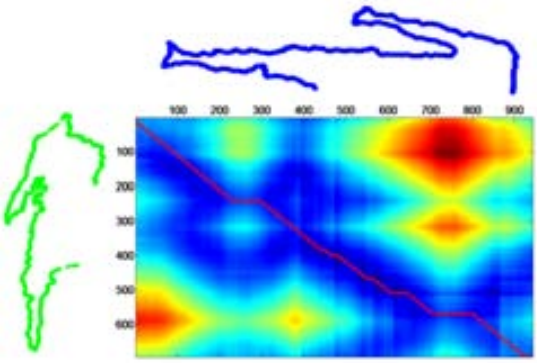


Figure 5.7: Cost matrix between the contour of the frontal view (blue color) and the contour of one posterior view (green color). As can be seen, the contour of the posterior view has a lower number of points because of the extra distance traveled by the IR pattern. In the cost matrix representation, the red line indicates the optimal path which produces a minimum overall cost.

Finally, we can use the information of this warping matrix in order to triangulate the two meshes. In Figure 5.8 we can see the model correctly zippered by our implementation.

5.4 Experimental results

In order to evaluate the accuracy of the system 3 different experiments are proposed, focusing on the two major contributions of this system: the presence of mirrors for a single view 3D modeling and the use of Dynamic Time Warping for zippering meshes without overlap.

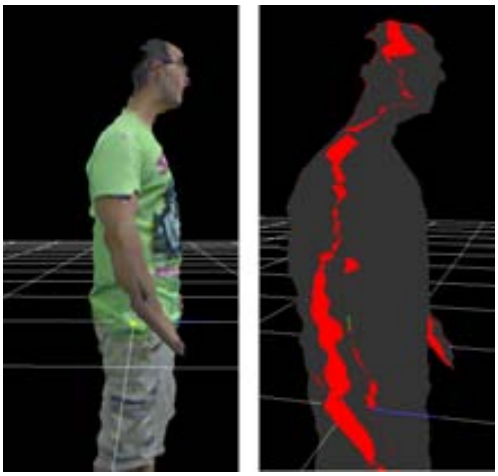


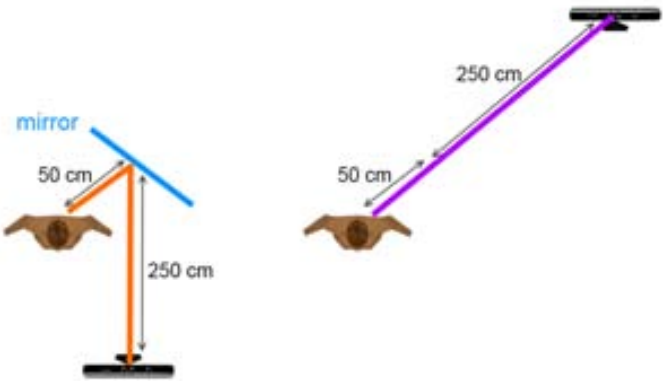
Figure 5.8: Model stitching using DTW. In the right image is shown, in red color, the zippered faces between the meshes.

5.4.1 Loss of information produced by the mirrors re ection

Using a mirror to reflect the object helps us to reduce the global space needed in the scene. However, a loss of information is produced by this reflection, and this loss of information affects to the final reconstruction. This loss can be produced either by the quality of the reflection caused by the mirror or by the extra distance done by the IR projection which produces a loss of quality in the generated mesh. In order to evaluate only the loss produced by the reflection the following experiment is proposed. First, the person is placed in front of a mirror, and the 3D data produced by the reflection is stored, annotating also the distance of the camera with respect to the mirror an the distance between the mirror and the posterior part of the person. In a second part of the experiment, without moving the object and discarding the mirror, the camera is placed behind the object (in the direction of the reflection) at the same distance than the sum of the two distances stored before. A schema of this process can be seen in Figure 5.9, where the only difference between two captures is the reflection of the mirror, because the total distance will be equivalent. Having these two 3D images available, we can now compare both in order to see if it exists a loss in range accuracy or in the resolution.

In order to avoid the possible movements of a person between the captures in the experiment a mannequin will be used. In addition, to avoid the noisy 3D images produced by the Microsoft Kinect a total amount of 10 range frames is captured in both setups and the mean value for each 3D point is assigned.

In Figure 5.10 the results of this experiment are shown. In order to compare both



(a) (b)

Figure 5.9: Diagram of the experiment. In (a) the posterior part of the person is captured by the reflection of the IR pattern. The resulting 3D is obtained at a distance which is equivalent to the distance of the camera to the mirror plus the distance of the mirror to the object. In a second phase, in (b), the mirror is discarded and the camera is placed at the same distance but in straight line, so the total distance will be equivalent.

meshes the Hausdorff distance [10] between the meshes is used, sampling one of the meshes and computing for each sample the Hausdorff distance to the closest point on the other mesh. Visually comparing the 3D meshes obtained from the experiment (subfigures 5.10(b) and 5.10(d)), it can be seen a change of the texture color in the shirt (produced by the light reflections in the mirror) and a loss of 3D points in the edges of the reflected mesh. A clear example can be seen in the hand, which is less defined in the reflected 3D view in subfigure 5.10(b). This loss of resolution in the edges is confirmed after computing the Hausdorff distance, which is close to zero in the inner part of the meshes and tends to be higher in the edges. The maximum distance between both meshes is 1,2897 cm. and the mean distance for all the samples is 0,2083 cm.

In addition to the Hausdorff distance between meshes, the loss of resolution due to the mirror reflection is analyzed. The 3D mesh obtained with reflection has a total number of 18097 vertices and 34985 faces. On the other side, the mesh obtained with direct capture has 22755 vertices and 44303 faces, so the percentage of loss using a mirror is about 20%, both for vertices and faces.

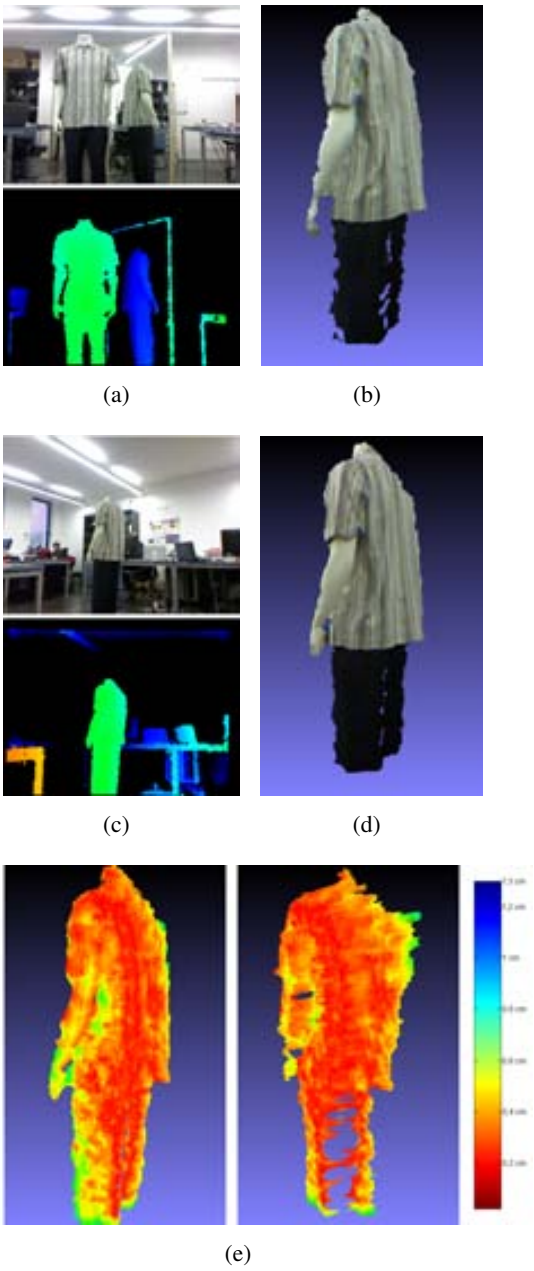
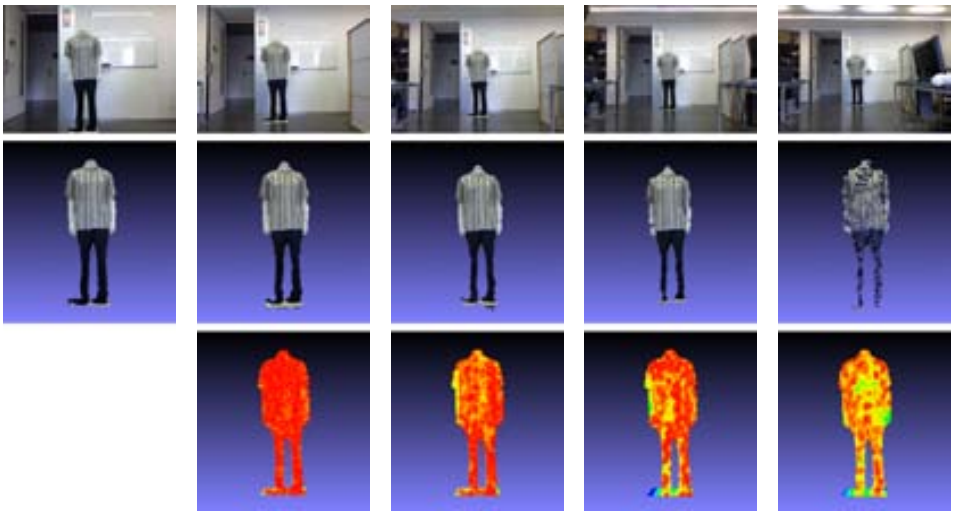


Figure 5.10: (a) Visible image and depth image using the mirror. With this information, and after computing the flip of the mirror, the obtained 3D representation is shown in (b). Discarding the mirror and placing the Kinect at the back side of the mannequin with the same distance, the resulting images and the 3D representation are shown in (c) and (d). In (e) we can see the result after comparing both 3D meshes using the Hausdorff distance, using the same point of view used previously and another view looking at the back.

5.4.2 Loss of information produced by the extra distance in the mirrors

The following experiment was based on taking snapshots of the mannequin at different distances in order to evaluate their possible implication in the quality of the generated mesh. Different captures at 300, 350, 400, 450 and 500 cm. are obtained, and the results can be seen in Figure 5.11. Results show that the distance with respect to the mannequin affects to the quality of the generated mesh, where for higher distances the quality of the mesh is greatly reduced.



(a) (b) (c) (d) (e)

Figure 5.11: At the top, captured visible image of the mannequin at 300, 350, 400, 450 and 500 cm. respectively. In the middle row the resultant 3D meshes are shown, having a degradation of the mesh for the higher distances. At the bottom, Hausdorff distance of the 3D meshes against the first mesh, which is considered as reference. We can see that due to the range camera resolution, the farther is the object, the bigger the difference.

In Figure 5.12 the mean value of the Hausdorff distances for each separation of the mannequin are shown, starting from 300 cm. (which has a distance of 0 cm. because is compared to itself) to the 500 cm. We can observe an exponential behavior, where for each additional 50 cm. the Hausdorff distance is near to be doubled.

In addition to the inaccuracy produced by the distance, also the loss of vertices and faces is evaluated. In Figure 5.13, a plot indicating the number of vertices and faces for each capture is shown. We can see that the results fits with an exponential decay model.

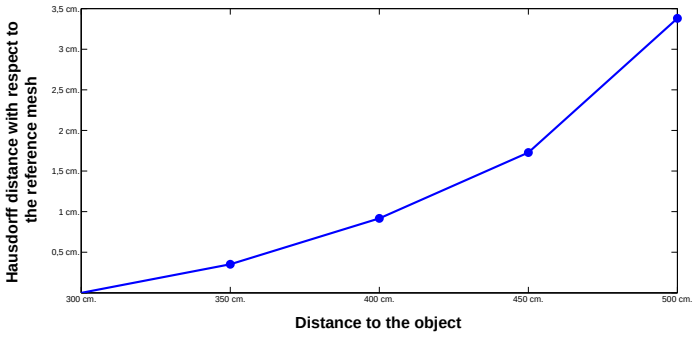


Figure 5.12: Mean value of the Hausdorff distance for a separation of 300 cm.(Hausdorff 0 cm.), 350 cm. (Hausdorff 0.3517 cm.), 400 cm. (Hausdorff 0.9170 cm.), 450 cm. (Hausdorff 1.7277 cm.) and 500 cm. (Hausdorff 3.3804 cm.).

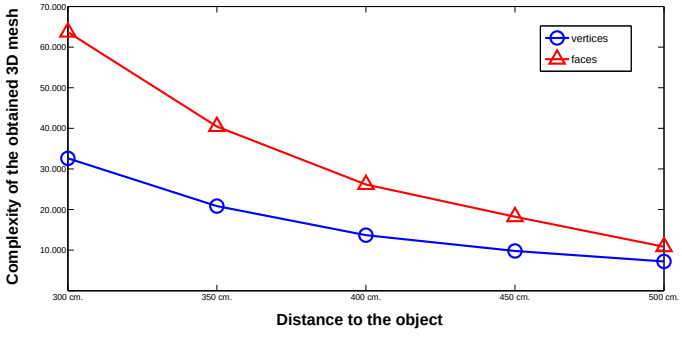


Figure 5.13: Relation between complexity of the mesh and distance to the object.

5.4.3 Evaluation of the zippering process

In this subsection we discuss results obtained with the stitching process using Dynamic Time Warping. The objective is to evaluate the loss of information which is produced in the 3D model due to the zippering between the 3 meshes.

Since the noise produced by the range camera used causes random mesh generation, a reliable experiment with a captured 3D model or person cannot be done. For this reason a synthetic human model is used, splitting it in 3 parts and later zippering using our method. The synthetic model is splitted in two parts by the coronal plane, and afterwards the back part is splitted again by its sagittal plane, giving us the 3 parts obtained as we would use the mirrors.

The split of the parts is done by subtracting points of the synthetic mesh. Since the triangulation of the mesh depends on these 3D points, the faces composed by the subtracted points will disappear, giving us an irregular split which is similar to the split produced by the mirrors.

Using the mirrors approach proposed in this document, in addition to the split of the model, a loss of resolution on the back of the model is produced. To emulate this loss on the synthetic model a simplification on the two back meshes between 0% and 50% is done. To evaluate the zippering result, Hausdorff distance between the result of the zippering and the original synthetic mesh is computed.

In Figure 5.14 the result of zippering the splitted model with a loss of 40% for the posterior meshes can be seen. As expected, Hausdorff distance increases in the zones where there are more difference in the resolution.

In order to evaluate the accuracy of the zippering process with respect to the resolution degradation on the back meshes, the mean value of the Hausdorff distance is analyzed for degradation of 10%, 20%, 30%, 40% and 50%. The results are displayed in Figure 5.15, where the evolution of the accuracy has a linear behavior.

5.5 Conclusions

In this chapter we have presented a novel system for the efficient modeling of a human body using only one range camera and two mirrors. Our method presents good characteristics in terms of efficiency, compactness and low memory usage.

The experiments show that, with a low-cost range camera like the Microsoft Kinect and two mirrors, a fast 3D reconstruction can be done. The use of mirrors allows a reduction of the space needed for the modeling, but on the other side produce a degradation on the created 3D model. This degradation is produced by two

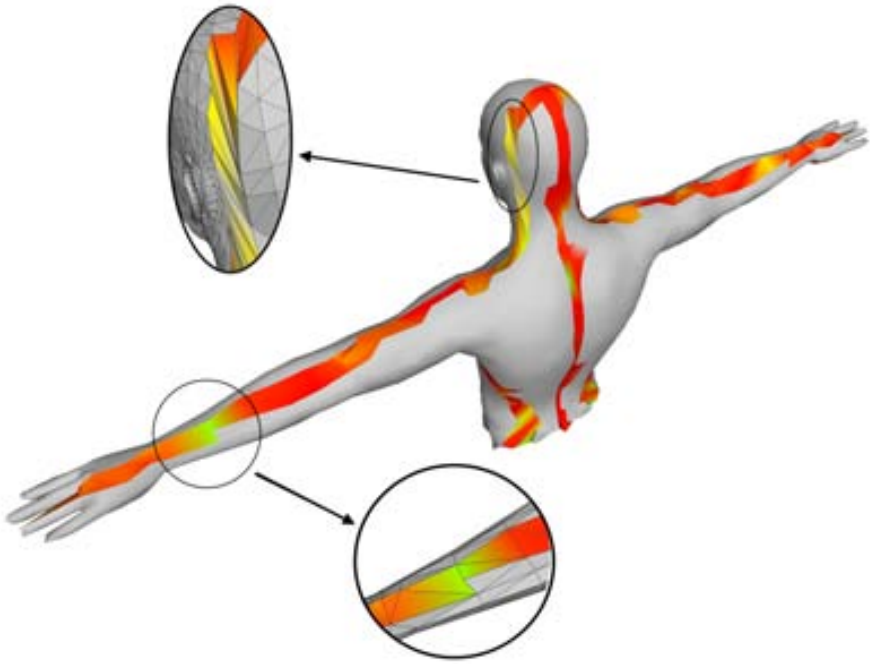


Figure 5.14: Result of the zippering for a reduction of 40% for the two back meshes. Red color indicates a low Hausdorff distance, while blue color indicates a high Hausdorff distance. In the image magnification of the head it can be seen that the high Hausdorff distance is produced by the high difference between the resolutions of the frontal and the back mesh. In the image magnification of the arm, a discontinuity of the mesh produce a high Hausdorff distance because the original mesh had two triangles in this position, while our zippering process only triangulates with one triangle.

factors: the reflection itself, which produces a loss of about 20% in the number of vertices and faces, and the additional distance of the IR pattern after bouncing at the mirror.

Due to the use of the mirrors there was no overlap between the meshes and therefore the traditional techniques for stitching could not be implemented. Dynamic Time Warping has demonstrated that is a powerful algorithm not only suitable for speech recognition, but also for many other fields.

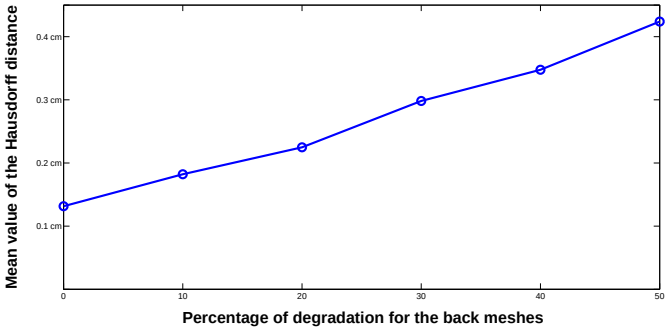


Figure 5.15: Evolution of the mean Hausdorff distance for different degradation percentage of the back meshes.

Chapter 6

Concluding Remarks and Future Work

In this thesis different variants for the registration of range images are presented. From the first step of obtaining the 3D point cloud to the registration of multiple range images proceeding from different scan positions, different possibilities have been explained along these pages. Some conclusions can be extracted from the work of these years, and also new lines of research can be opened after the obtained experience.

6.1 Conclusions

Basically, the thesis can be divided in 3 different parts:

- In the first part we study the multisensorial registration between range cameras and other sources of information, obtaining as result a textured 3D representation which combines all the information at the same time. Although some possibilities exist, we decided for a semi-automatic method which needs the manual selection of some correspondences between the range image and the other source of information.

    1. The experiments have been divided in two different scenarios: the short range case and the large range case. The obtained results shows a better estimation of the camera displacement in the short range case, but on the other side the orientation of the camera is better estimated in the large range case.
    2. Experimental results show how the multisensorial registration error behaves in comparison to the selection of the correspondences. The results

reflect that the error is almost zero in case of an accurate selection of the correspondences, and is growing linearly as long as this selection is more inaccurate. In addition, the increase in the number of correspondences between the images improves the accuracy of the estimations, although it shows an exponential decay behaviour.

3. The reprojection error (the error comparing the original image and the image composed of the backprojection of the 3D points in the estimated camera pose) shows also a lineal behaviour when the gaussian noise is added. In addition, the experimental results show that, from an specific number of correspondences on, the error in the manual selection of the points can be minimized after the reprojection.

- The second part includes Chapter 3 and Chapter 4, and integrates the core part of this thesis. In Chapter 3 the problem of pairwise registration is presented, explaining some existing possibilities and presenting a new descriptor based on the covariance for both 3D shape and texture aspects. Individual range images must be registered pair-to-pair with their neighboring range images giving, if possible, as much registrations as possible in order to deliver a valuable redundance information. This redundance is the basis for the second step, the minimization of the global error for the whole set. A novel algorithm by means of a Bayesian framework is proposed, achieving good results in cases with incorrect correspondences. In particular, we aim to have the possibility of including a probability weight to each correspondence between the 3D scans, making possible to detect which of these correspondences were incorrectly established.

  1. Traditional methods for pairwise registration are usually based on descriptors which only take into account the 3D shape of the scene. During last years and currently, some approaches which also rely on the information of the texture are also appearing in the literature.

  2. The proposed fusion covariance descriptor achieves a good performance in 3D point clouds with a limited noise, and it maintains this behaviour when the noise is highly incremented, outperforming other state-of-the-art proposals.

  3. A possible implementation for the filtering of typical structures in the 3D scene is presented. Specifically, a detection of large planes representing walls of buildings is explained, allowing a better identification of the elements which better represent the scenario.

4. The multiview registration methods goes a step further in the registration process, achieving a global minimization of all the set of range images. A new algorithm based on Bayesian framework is proposed, allowing the detection and minimization of possible incorrect correspondences produced by an incorrect selection or by challenging situations like symmetries or repetitive patterns.

5. Results demonstrates that the multiview registration obtains better results if the object has a higher number of correspondences in relation to the number of views or the total number of points.

6. Even in cases where the algorithm fails, the obtained registration is usually incorrect only because of a single registration which is not able to align with the set of the other ones, giving a global registration which is obviously incorrect but near to the desired result.

7. The proposed algorithm can serve not only for situations where there exist degraded correspondences, but also for registration cases where we want to obtain a better accuracy. This increase of the accuracy must be pondered with the increase of time execution, due to the use of an iterative algorithm.

- In the third part of this thesis an specific system using one single view is presented. The system is based on the use of a Microsoft Kinect device and two mirrors.

1. The system allows the 3D modeling of a person (or an object of similar dimensions) by using one single range camera. The main core of this system is the use of two mirrors placed behind the person, allowing the range camera to detect his posterior part. The disposition of the system allows a high reduction in the needed space.

2. The use of the mirrors produces, however, two main problems which must be considered: the loss of quality produced by the reflection in the mirror and the loss of quality produced by the extra distance that the IR pattern must travel when colliding with the mirrors. Experimental results show that this second source of error is considerably higher than the loss due to the reflection itself.

3. A novel method for the stitching of non-closed surfaces is presented, by using the Dynamic Time Warping algorithm. The experiments revealed that this method works correctly, producing a low error if we compare to the original surface. However, this error is getting higher in cases where we have a discontinuity in the surface, or when we have a high difference in resolution between the frontal and the back mesh

6.2 Future Work

The aim of this thesis was to study the different possibilities of range image and their combination in order to obtain a global 3D structure which combines all the information together. There are, however, some aspects which can be deeply studied or also new possibilities which can be explored.

- One of the most interesting lines of continuation is the extension to perform the registration of the 3D scans by using information from the other sources of images, i.e., infrared image, gamma image, or even the reflectance image produced by the own 3D scanner (image produced by some 3D scanners which gives an idea about the material of the scanned objects). Some experiments were done during this thesis, but we finally decided to use the visible image because of its richness in definition and the high availability of methods for its processing in the computer vision literature.

- For the current implementation of the covariance descriptor we have used the RGB space of color and the resulting angles α, β and γ between the center point of the descriptor and its neighborhood. However, other different features can be easily integrated in the covariance descriptor, allowing a more complete representation. Use of other spaces of color like CIELab, or additional geometric concepts which preserve their structure in case of rotations and translations can be researched in the future.

- Some experimental work should be still done with the covariance descriptor. Challenging scenarios considering the reduction of the resolution for the 3D point clouds or the addition of clutter to the scene should be considered.

- A possible improvement of the multiview registration algorithm presented could be the possible removal of views, or the separation of the final result in two dif-

ferent registrations. Extending this possibility it could be useful in cases like the one shown in Figure 4.8 and similar ones.

- Finally, it is expected that range cameras with higher possibilities and lower cost will be available during the next years. Microsoft Kinect is just the beginning of a new era of consumer range cameras, which will be probably integrated in laptops and smartphones in the future. Their possibilities and drawbacks will depend on this integration, so we must be aware of their possibilities.

Appendix A

Variational EM Algorithm

Consider the joint distribution $P(X\ Z\ \theta)$ over the complete data set in our pursue of maximizing $P(X\ \theta)$, since:

$$P(X\ \theta) = \sum_{Z} P(X\ Z\ \theta) \tag{A.1}$$

which is equivalent to maximize the log-likelihood:

$$\mathcal{L}(X\ \theta) = log[P(X\ \theta)] = \log \sum_{Z} P(X\ Z\ \theta) \tag{A.2}$$

where a set of arbitrary distributions $Q(Z)$ can be introduced without losing generality:

$$\mathcal{L}(X\ \theta) = \log \sum_{Z} Q(Z) \frac{P(X\ Z\ \theta)}{Q(Z)} \tag{A.3}$$

By means of Jensen s inequality [29] [45], a lower bound to this log-likelihood can be computed

$$\mathcal{L}(X\ \theta) = \log \sum_{Z} Q(Z) \frac{P(X\ Z\ \theta)}{Q(Z)} \geq \sum_{Z} Q(Z) \log \frac{P(X\ Z\ \theta)}{Q(Z)} \equiv F(Q\ \theta) \tag{A.4}$$

This lower bound is known as *free energy term* [44] and it corresponds to the sum of the Kullback-Leibler divergence [34] of the approximating Q-functions and

the true posterior and the marginal log-likelihood:

$$F(Q\ \theta) \equiv \sum_Z Q(Z) \log \frac{P(X\ Z\ \theta)}{Q(Z)} = \sum_Z Q(Z) \log \frac{P(Z\ X\ \theta)P(X\ \theta)}{Q(Z)} =$$
$$= \sum_Z Q(Z) \log P(X\ \theta) + \sum_Z Q(Z) \log \frac{P(Z\ X\ \theta)}{Q(Z)} = \mathcal{L}(X\ \theta) - KL(Q\ P(Z\ X\ \theta)) \tag{A.5}$$

The non-decreasing updating of a lower bound such as $F(Q\ \theta)$ of a log-likelihood function (satisfying the condition above) implies getting closer to the maximum value at each step. Since such a maximum value is finite, there will be a time in the procedure when the free energy $F(Q\ \theta)$ reaches that value. Maximizing $F(Q\ \theta)$ has to be performed in two steps: i) first, with respect to the approximating Q-functions, and, ii) with respect to the model s parameters θ.

In this case, the Expectation-Maximization consist of two optimization steps: one implies finding the "nearest" (in terms of the Kullback-Leibler divergence) Q-distribution to the *a posteriori* probabilities for the latent variables, and another that involves maximizing with respect to the model s parameters:

- **Expectation**: Assume an intermediate stage with θ^l (l-th iteration). Now, the statement "*Compute the suf cient statistics for the latent variables posterior distributions* $P(Z\ X\ \theta)$" can be translated into:

$$Q(Z) = \arg\max_{Q'(Z)} [F(Q'(Z)\ \theta^l)] \tag{A.6}$$

  that can be found by taking functional derivatives on $F(Q\ \theta)$, and whose solution is:

$$Q(Z)^{l+1} = P(Z\ X\ \theta^l) \tag{A.7}$$

  which is determined by its *suf cient statistics*, i.e., the expected moments.

- **Maximization**: Given the new value Q^{l+1}, the function to be maximized with respect to the model s parameters θ is $F(Q^{l+1}\ \theta)$:

$$\theta^{l+1} = \arg\max_{\theta} [F(Q^{l+1}\ \theta)] \tag{A.8}$$

Thus, after substituting the computed values for Q^{l+1} into $F(Q^{l+1}\ \theta)$, the quantity to be maximized is:

$$\theta^{l+1} = \arg\max_{\theta} \sum_{Z} Q(Z)^{l+1} \log \frac{P(X\ Z\ \theta)}{Q(Z)^{l+1}} \tag{A.9}$$

which, in practice, means:

$$\theta^{l+1} = \arg\max_{\theta} \sum_{Z} Q(Z)^{l+1} \log[P(X\ Z\ \theta)] \tag{A.10}$$

and, thus, according to Equation (A.7):

$$\theta^{l+1} = \arg\max_{\theta} \sum_{Z} P(Z\ X\ \theta^{l}) \log[P(X\ Z\ \theta)] \tag{A.11}$$

List of Publications

This dissertation has led to the following communications:

Journal Papers

- Xavier Mateo, Xavier Orriols, and Xavier Binefa. Bayesian Perspective for the Registration of Multiple 3D Views. Accepted in ***Computer Vision and Image Understanding***. Publication pending, non-final version available online at http://dx.doi.org/10.1016/j.cviu.2013.09.003

Conference Contributions

- Luis Ruiz, Xavier Mateo, Ciro Gracia, and Xavier Binefa. Single Snapshot System for the Fast 3D Modeling using Dynamic Time Warping. In ***Proceedings of the International Conference on Computer Vision Theory and Applications***, pp. 317–326, vol. 2, 2012.

- Xavier Mateo, and Xavier Binefa. Plane Filtering for the Registration of Urban Range Laser Imagery. In ***Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis***, pp. 136–143, 2009.

- Xavier Mateo, and Xavier Binefa. Laser Range Data Registration using Spin Images. In ***Proceedings of the International Conference on Computer Vision Theory and Applications***, pp. 541–545, vol. 2, 2009.

- Xavier Mateo, and Xavier Binefa. Georeferencing Image Points using Visual Pose Estimation and DEM. In ***Proceedings of the International Conference of the Catalan Association of Artificial Intelligence***, pp. 233–242, 2007.

In Preparation

- Pol Cirujeda, Xavier Mateo, Yashin Dicente, and Xavier Binefa. A visual and shape fusion covariance descriptor for 3D scene registration via a game theoretic solution method.

Bibliography

[1] Y. I. Abdel-Aziz and H. M. Karara. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. *Symposium on Close-Range Photogrammetry*, 1971.

[2] A. Albarelli, E. Rodola, and A. Torsello. A game-theoretic approach to fine surface registration without initial motion estimation. In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, pages 430–437, 2010.

[3] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-D point sets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 9(5):698–700, 1987.

[4] P.J. Besl and N.D. McKay. A method for registration of 3-D shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14:239–256, 1992.

[5] H. Cantzler. *Improving architectural 3D reconstruction by constrained modelling*. PhD thesis, University of Edinburgh, 2003.

[6] A.F. Chase, D.Z. Chase, J.F. Weishampel, J.B. Drake, R.L. Shrestha, K.C. Slatton, J.J. Awe, and W.E. Carter. Airborne LiDAR, archaeology, and the ancient maya landscape at Caracol, Belize. *Journal of Archaeological Science*, 38(2):387–398, 2011.

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

[8] D. Chetverikov, D. Stepanov, and P. Krsek. Robust euclidean alignment of 3D point sets: the trimmed iterative closest point algorithm. *Image and Vision Computing*, 23(3):299–309, 2005.

[9] C. Chua and R. Jarvis. Point signatures: A new representation for 3D object recognition. *International Journal of Computer Vision*, 25(1):63–85, 1997.

[10] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: Measuring error on simplified surfaces. *Computer Graphics Forum*, 17(2):167–174, 1998.

[11] The Ohio State University OSU(MSU/WSU) Range Image Database. 2004.

[12] D.F. Dementhon and L.S. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15(1-2):123–141, 1995.

[13] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[14] D.W. Eggert, A.W. Fitzgibbon, and R.B. Fisher. Simultaneous registration of multiple range views for use in reverse engineering of CAD models. *Computer Vision and Image Understanding*, 69(3):253–272, 1998.

[15] H.W. Eves. *Elementary Matrix Theory*. Phoenix Edition Series. Dover, 1980.

[16] D. Fehr, A. Cherian, R. Sivalingam, S. Nickolay, V. Morellas, and N. Papanikolopoulos. Compact covariance descriptors in 3D point clouds for object recognition. In *Robotics and Automation, IEEE International Conference on*, pages 1793–1798. IEEE, 2012.

[17] P.D. Fiore. Efficient linear solution of exterior orientation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(2):140–148, 2001.

[18] M.A. Fischler and R.C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[19] W. Förstner and B. Moonen. A metric for covariance matrices. *Quo vadis geodesia*, pages 113–128, 1999.

[20] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik. Recognizing objects in range data using regional point descriptors. In *IEEE European Conference on Computer Vision*, pages 224–237. 2004.

[21] A. Fusiello. Elements of geometric computer vision, 2006.

[22] G. Godin, D. Laurendeau, and R. Bergevin. A method for the registration of attributed range images. In *3-D Digital Imaging and Modeling, International Conference on*, pages 179–186, 2001.

[23] B. Hall. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*. Graduate Texts in Mathematics. Springer, 2003.

[24] M. Hansard, S. Lee, O. Choi, and R.P. Horaud. *Time of Flight Cameras: Principles, Methods, and Applications*. Springer Briefs in Computer Science. Springer, 2012.

[25] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[26] B.K.P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987.

[27] D.F. Huber and M. Hebert. 3D modeling using a statistical sensor model and stochastic search. In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, pages 858–865, 2003.

[28] D.F. Huber and M. Hebert. Fully automatic registration of multiple 3D data sets. *Image and Vision Computing*, 21(7):637–650, 2003.

[29] J.L.W.V. Jensen. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, 30(1):175–193, 1906.

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

[31] A. Kolb, E. Barth, R. Koch, and R. Larsen. Time-of-flight sensors in computer graphics. In *Proceedings of Eurographics - State of the Art Reports*, pages 119–134, 2009.

[32] S. Krishnan, P.Y. Lee, J.B. Moore, and S. Venkatasubramanian. Global registration of multiple 3D point sets via optimization-on-a-manifold. In *Eurographics Symposium on Geometry processing*, 2005.

[33] S. Krishnan, P.Y. Lee, J.B. Moore, and S. Venkatasubramanian. Optimisation-on-a-manifold for global registration of multiple 3D point sets. *International Journal of Intelligent Systems Technologies and Applications*, 3(3/4):319–340, 2007.

[34] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):pp. 79–86, 1951.

[35] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[36] C.P. Lu, G.D. Hager, and E. Mjolsness. Fast and globally convergent pose estimation from video images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(6):610–622, 2000.

[37] G.J. MacLachlan and K.E. Basford. *Mixture Models: Inference and Applications to Clustering*. Statistics Series. Marcel Dekker Incorporated, 1988.

[38] C. Matabosch, D. Fofi, J. Salvi, and E. Batlle. Registration of surfaces minimizing error propagation for a one-shot multi-slit hand-held scanner. *Pattern Recognition*, 41(6):2055–2067, 2008.

[39] G.J. McLachlan and T. Krishnan. *The EM algorithm and extensions*. Wiley series in probability and statistics. Wiley, 2nd edition, 2008.

[40] T.P. Minka. Old and new matrix algebra useful for statistics, MIT Media Lab note. Technical report, 1997.

[41] M. Müller. *Information Retrieval for Music and Motion*. Springer-Verlag, 2007.

[42] S.K. Nayar, M. Watanabe, and M. Noguchi. Real-time focus range sensor. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(12):1186–1198, 1996.

[43] A. Nüchter, H. Surmann, K. Lingemann, J. Hertzberg, and S. Thrun. 6D SLAM with an application in autonomous mine mapping. In *In Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1998–2003, 2004.

[44] R.M. Neal and G.E. Hinton. A new view of the EM algorithm that justifies incremental and other variants. In *Learning in Graphical Models*, pages 355–368. Kluwer Academic Publishers, 1993.

[45] T. Needham. A visual explanation of Jensen s inequality. *The American Mathematical Monthly*, 100(8):768–771, 1993.

[46] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.

[47] K. Pulli. Multiview registration for large data sets. In *3-D Digital Imaging and Modeling, International Conference on*, pages 160–168, 1999.

[48] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *3-D Digital Imaging and Modeling, International Conference on*, 2001.

[49] R.B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (FPFH) for 3D registration. In *Robotics and Automation, IEEE International Conference on*, pages 3212–3217, 2009.

[50] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 26(1):43–49, 1978.

[51] A.D. Sappa and M.A. Garcia. Incremental multiview integration of range images. In *Pattern Recognition, International Conference on*, volume 1, pages 546–549, 2000.

[52] G.C. Sharp, S.W. Lee, and D.K. Wehe. Multiview registration of 3D scenes by minimizing error between coordinate frames. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(8):1037–1050, 2004.

[53] S.W. Shih, Y.T. Chuang, and T.Y. Yu. An efficient and accurate method for the relaxation of multiview registration error. *Image Processing, IEEE Transactions on*, 17(6):968–981, 2008.

[54] L. Silva, O.R.P. Bellon, and K.L. Boyer. Multiview range image registration using the surface interpenetration measure. *Image and Vision Computing*, 25(1):114–125, 2007.

[55] E.R. Smith, B.J. King, C.V. Stewart, and R.J. Radke. Registration of combined range-intensity scans: Initialization through verification. *Computer Vision and Image Understanding*, 110(2):226–244, 2008.

[56] M. Soucy and D. Laurendeau. A general surface approach to the integration of a set of range views. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(4):344–358, 1995.

[57] F. Tombari, S. Salti, and L. Di Stefano. A combined texture-shape descriptor for enhanced 3D feature matching. In *Image Processing, IEEE International Conference on*, pages 809–812, 2011.

[58] F. Tombari, S. Salti, and L. Stefano. Unique signatures of histograms for local surface description. In *IEEE European Conference on Computer Vision*, pages 356–369. 2010.

[59] B. Triggs, P.F. McLauchlan, R.I. Hartley, and A.W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Vision Algorithms: Theory and Practice*, Lecture Notes in Computer Science, pages 298–372. Springer Berlin Heidelberg, 2000.

[60] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *Conference on Computer graphics and interactive techniques*, pages 311–318, 1994.

[61] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. *IEEE European Conference on Computer Vision*, pages 589–600, 2006.

[62] O. Tuzel, F. Porikli, and P. Meer. Pedestrian detection via classification on riemannian manifolds. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(10):1713–1727, 2008.

[63] V. Verma, Rakesh Kumar, and S. Hsu. 3D building detection and modeling from aerial lidar data. In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, volume 2, pages 2213–2220, 2006.

[64] C. Wu, B. Clipp, X. Li, J.M. Frahm, and M. Pollefeys. 3D model matching with viewpoint-invariant patches (VIP). In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, pages 1–8, 2008.

[65] J. Yao and J.M. Odobez. Fast human detection from videos using covariance features. In *International Workshop on Visual Surveillance*, 2008.

[66] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud. Surface feature detection and description with applications to mesh matching. In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, pages 373–380, 2009.

[67] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *Computer Vision, IEEE International Conference on*, volume 1, pages 666–673, 1999.

[68] Z. Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, 2000.