International Journal of Applied Engineering Research ISSN 0973-4562 Volume 11, Number 8 (2016) pp 5704-5713

Mapping of Indoor Environments using Point Cloud Library (PCL)

Leonardo A. Góngora Velandia
Professor, Department of Mechatronics Engineering,
Piloto University of Colombia Bogotá, Colombia
E-mail: ruben-hernandez1@upc. edu. co

Ruben D. Hernández
Professor, Department of Mechatronics Engineering,
Piloto University of Colombia Bogotá, Colombia
E-mail: ruben-hernandez1@upc. edu. co

Oscar F. Avilés
Titular Professor, Department of Mechatronics Engineering,
Militar Nueva Granada University Bogotá, Colombia
E-mail: oscar-aviles@upc. edu. co

João M. Rosário
Titular Professor, Mechanics Engineering Faculty,
Estatal of Campinas University Campinas-SP, Brazil
E-mail: rosario@fem. unicamp. br

Abstract
Simultaneous localization and mapping (SLAM) is one of the most important topics within the current robotics; two important processes for the development of SLAM; mapping and localization are defined in order to achieve the goal of building fully autonomous devices. This document focuses on the development of algorithms for processing point clouds to approach the SLAM concept in terms of mapping. Two algorithms for the registration process are proposed, which are based on two different operations; the first related to landmarks between each pair of samples to finally perform the calculation of the transformation matrix of point clouds, namely feature-based registration (FBR), the second based on the reduction of distances between points based on the ICP operation. Each algorithm achieves fairly good results regarding environment mapping, however the IPC based operation displays better results, as shown in the collected data.

Keywords: Feature Based Registration (FBR), Iterative Closest Point (ICP), Registration, Simultaneous Localization and Mapping (SLAM).

Introduction

Simultaneous localization and mapping is one of the most important issues within the probabilistic robotics. The main objective of SLAM is to provide a mobile robot in a totally unfamiliar environment with the skills needed in order to be able to build-through its sensors and implements-a consistent map of this environment while simultaneously determining in location within this map. [1]-[4].

The relevance of the SLAM concept within robotics is related to its solution; an important step to build fully autonomous devices. Therefore SLAM is recognized as the "Holy Grail" of robotics. In theoretical terms, the formulation of a solution for simultaneous localization and mapping (SLAM), has been one of the most notable achievements within the academic community, however, in many cases, handling the big amount of data collected in large environments and their processing have been problems to deal with, in order to provide optimal solutions regarding SLAM [5].

Simultaneous localization and mapping is a widely developed topic both for open environments (outdoor) [6]-[8], as for internal environments (indoor) [9]-[12]. Thus, it is necessary to consider that the implementation of this SLAM solution can be categorized according to the varying difficulty in terms of the type of environment to be applied, since one of the main differences between indoor and outdoor environments, is related to the types of sensors that can be used. This, because in outdoor locations it is possible to find global positioning technologies (GPS), while in indoor locations, the use of this resource is limited, making it necessary to use robust algorithms to reduce errors while calculating the location of the mobile robot [1].

Regarding the sensors used to gather depth information using SLAM, there are many options that can be used: time of flight cameras (TOF), stereoscopic cameras, structured light sensors [13], ultrasound sensors and even magnetic field sensors. However, with the emergence of new technologies related to depth data management as the Kinect of Microsoft [14] or the Xtion Pro Live of Asus [15], it has been possible to run low cost applications with accurate results aimed at SLAM in indoor environments, [16], which shows that the use of these tools is a viable and affordable alternative in terms of research development in these areas. However, despite being accurate and inexpensive tools, there is a drawback in terms of its use in outdoor environments due to the type of technology based on structured light [17], with shortcomings while reading depth data in the presence of natural light.

Despite these drawbacks, SLAM mapping applications can be implemented in indoor environments using this kind of technology. The output of depth sensors based on structured light condenses into point clouds, which are arrays of values with depth data and in some cases color (RGB-D [14]), with these data structures it is possible to perform the required procedure in order to associate, align and build an environment map through an algorithm receiving depth samples incrementally, as the robot moves in the work area [18].

This article aims to describe in detail the process of performing two registration algorithms based on the point cloud library (PCL), for processing data. The second section includes he theory to understand the main types of registration that can be implemented using the PCL library and the algorithms proposed for this task are shown. In the next section, a detailed explanation of the results obtained from these algorithms is made and the last section gathers the main conclusions and proposals for future works.

Methods

As mentioned above, the SLAM process comprises two important elements intrinsically clustered since they can't be developed independently to achieve the objective of simultaneous localization and mapping. This premise, as shown in [19], is founded on the fact that, for building a sufficiently accurate map of the environment in which the robot interacts, it is necessary to make an approximation of the robot positioning during its route, likewise, for precision and measurement accuracy, it is necessary to build an accurate virtual map with respect to the real place in order to get a correct estimate of these parameters. This paradox is deemed settled by experts in the field [5].

Mapping, in context with the SLAM theoretical approach, refers to the approximate building of a working environment map of a mobile robot. The methods used to perform this task depend on the available sensors for this work. These sensors can be ultrasound, magnetic, radar, time of flight cameras and structured light, among many others.

For the project described in this document, a structured light sensor known as Kinect from Microsoft is used. At the beginning, this sensor was used as a game controller interface of video games for Microsoft's Xbox platform. Furthermore, after its great success and its potential for the development of new applications, hardware development kits and (SDKs) software are available for researchers and product developers, becoming quite popular and being marketed as a development platform for Microsoft operating systems.

The first version of the Kinect sensor (Figure1) was used for the purpose of this project. Among the criteria for its use, it was chosen by the vast amount of information regarding its operation and existing development software for its applicability and solution thereof. In addition, its low economic cost and performance features in indoor environments make this sensor suitable for the purpose of the work presented.

Figure 1: First version of the Kinect sensor. (Taken from http://iphotopick. com/xbox-one-clip-art-gallery/)

Software Tools:

Within the software tools that can be used for developing applications with Kinect, SDK was recently released by the Microsoft team for version 1 and 2 of Kinect, using the Visual Studio software as a development platform. Also, open source tools designed specifically for depth data processing, such as point cloud library PCL have been made available to researchers, enabling the use of platforms for depth data processing, incorporating robust libraries widely documented for 3D image processing, besides including drivers and functions to manage different depth sensors as the Kinect, the Xtion, the Trimble MX8, the Fuji W3 3D, or the SICK LMS400 sensor, among many others.

The PCL library is available for all popular operating systems such as Microsoft Windows, some Linux systems and Apple OS X platforms. The point cloud library was chosen for developing an algorithm for mapping an indoor environment in this project.

The Registration

The registration process involves the association between different 3D images (captured through a depth sensor), performing the calculation of a global framework between images iteratively, until reaching an alignment according to preset parameters aligned with the reconstruction of an environment scene.

One of the main modules of the PCL library is specifically designed to perform tasks such as registration. This library of functions clusters the tools needed to process point clouds in the registration task, which are oriented according to two types of operations for information processing; feature-based registration (FBR) and Iterative Closest Point (ICP) registration.

Feature Based Registration (FBR):

The registration process using the PCL library consists of different essential steps for the correct determination of the alignment data among various point clouds, which are related from a global framework.

The feature based registration is one the processes that can be implemented with the point cloud library, its purpose is to calculate a transformation matrix linking two arrays of point clouds correlated through implicit and unchanging references, estimated from the geometry of the registered environment in relation to captures acquired with the depth sensor.

The following are the main steps for feature based registration:

1. Establish reference points to start processing.

2. From the landmarks, estimate the associated descriptors.
3. Perform correspondence calculation using the information of the descriptors associated to the landmarks used as input parameters.
4. Reject mis calculated correspondences. (Errors in calculating correspondences)
5. Calculate the transformation matrix between the captures to be registered.

The above list corresponds to the main steps for calculating the transformation matrix between captures to conclude the registration process.

Landmark extraction:

The landmarks are key elements for image processing, both in the 2D and the 3D domain. These points are quite relevant due to information processing and usually describe the geometry captured through point clouds.

A landmark must meet two important characteristics to be listed and used as such: points must be stable, which means that their detection should not depend on the relative position of the sensor after capturing information, allowing it to be detectable regardless of the angle, distance or position of the detected objects. Likewise, these landmarks must be unique with respect to the other detected in one capture, thus they should be recognized independently, even in dynamic environments.

Different kinds of land marks are clustered in PCL, including NARF, SIFT, AGAST, HARRIS and ISS.

The NARF method for extracting land marks was used for the development of this project. The NARF key points are typical features of point clouds, which are calculated from an image of estimated range usingthe depth variation associated to colors in the spectrum from blue to red as data processing. Figure 2 shows the estimating process of the NARF-type landmarks.

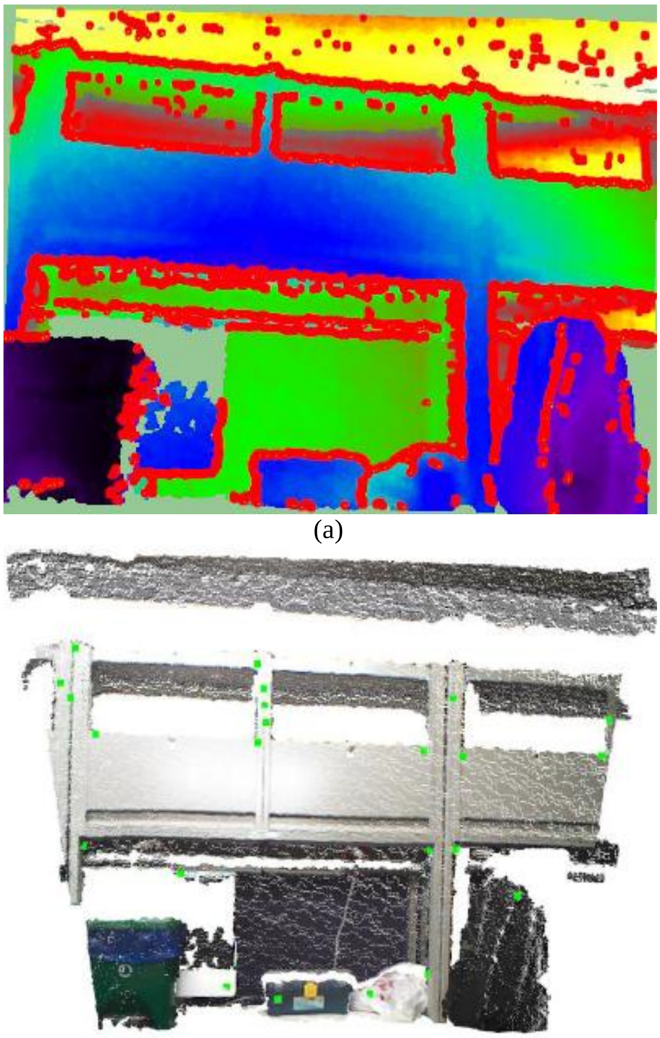
Descriptor extraction

Despite representing specific features related to the coordinates and distances of the point cloud acquired, landmarks do not contain information on the variation of the geometry of objects which have been calculated, thus, two landmarks from two different point clouds could have similar values, but they would not correspond to the same scene, and therefore post-processing these feature points could lead to errors.

This means that two landmarks (p_1, p_2) extracted at different time values (t_1, t_2) and despite having similar coordinates, do not guarantee that in the $t_2 - t_1$, time interval, the registered environment remains unchanged, making them unreliable to be classified as landmarks that can be used despite changes in position, angle, existing noise or other factors that may arise in the process of obtaining depth samples.

In these cases, complementary elements associated to the landmarks known as descriptors arise, which together with the landmarks form a point feature descriptor of the registration environment.

Local type descriptors such as PFH, FPFH and NARF were used for the development of this work as comparative elements for the registration process.



(a)

(b)

Figure 2*:* Landmark Calculation. a) Range image. b) NARF Points from Original image.

PFH Descriptor:

The PFH descriptor or Point Feature Histogram [20], is one of the descriptors that can be implemented using the point cloud library, based on the acquisition of landmarks as pre-processing stage. Figure 3 shows the p_q landmark in the middle of a work mesh, through which all points are interconnected.

The points framed within the dashed circle of Figure 3 refer to the neighboring points in relation to the p_q landmark, and interact together with other neighboring points, thus seeking to process certain characteristic values of the variability of this geometry in an area or surface represented by the point cloud, and primarily related to p_q.

Figure 4, shows in detail the calculation of the features between two landmarks (p_t, p_s). In this operation, p_t and p_s

are generic elements within the calculation operation of the PFH of p_q, whose illustration explains that the calculation of the surface and geometry features as objective of the PFH representation is performed around each of the neighboring points regarding p_q, and with each other.

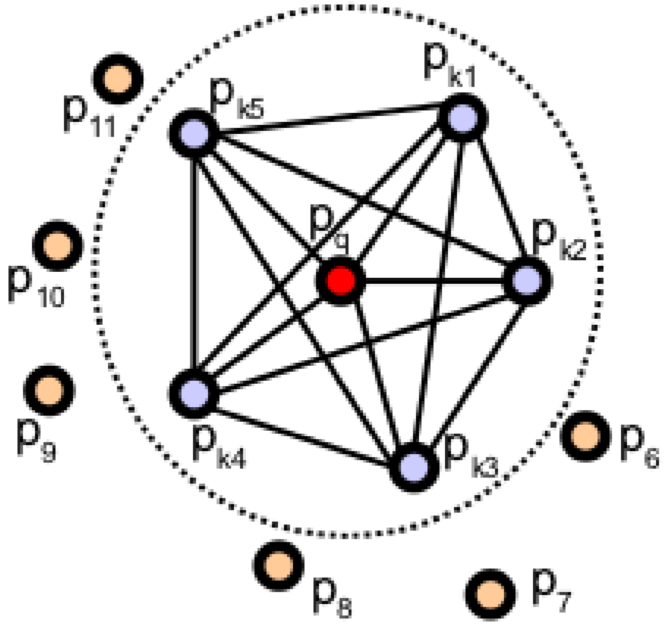


Figure 3: Characterization of a neighborhood area of a p_q landmark (in red). Taken from http://pointclouds. org.

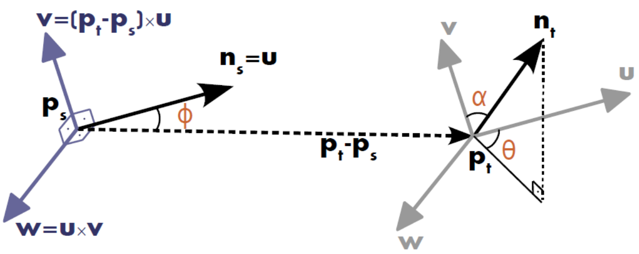


Figure 4: Feature Calculation of p_s and p_t . landmarks. Taken from http://pointclouds. org.

The data calculated from the PFH extraction represent the variation of the normal vectors to the orientation of thep_q neighboring points, then clustered together and taken to a histogram, which condenses the data for interpretation as descriptor elements of the surrounding geometric variation to the interest points.

FPFH descriptor:
The FPFH descriptor, or Fast Feature Point Histogram [20] is an improved version in terms of the performance of the algorithm for calculating the PFH descriptor, the aim is to reduce the theoretical operational complexity of the PFH $O(nk^2)$to an order of operations equal to$O(nk)$.
To accomplish this, the calculation of the features is performed only in cases of direct relationship between neighboring points and the p_q query point, as shown in.

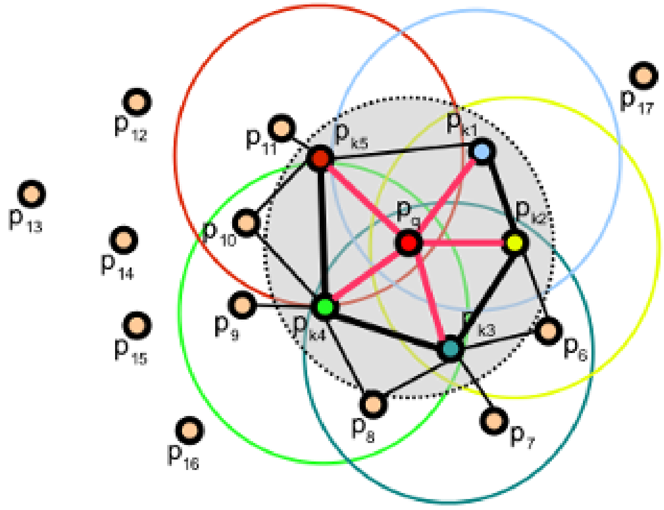


Figure 5: Relationship of neighborhood points in the calculation of FPFH descriptor. Taken from http://pointclouds. org.

Accordingly, the number of computational operations decreases, allowing management of these tools in real-time applications.

NARF descriptor:
NARF descriptor [21], encodes the information of the changes on the surface near the landmark. From the interest points calculated by the range image, a gradient range is created around the landmark, this can be classified as a small image of range, centered on the interest point which is aligned with the normal of the point.
By locating the focal point with the normal of the point, a star pattern with n beams is created, and the value for each point associated to the change of the surface that lies below the tip is calculated. The higher the change value reaching the center of the star, the larger the value. The descriptor associated with each landmark has n values resulting from the number of points of the star. Figure 6 shows this process illustratively.

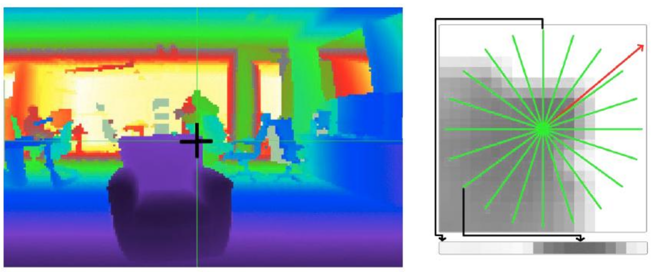


Figure 6: NARF Descriptor. Taken from [21].

Calculation, filtering and extraxction of the transformation matrix:
This step relates to the point clouds processing for the calculation of the estimated correspondences from the

information found through the descriptors in all captures acquired with the sensor.

Thus, the descriptors that are more similar to each point cloud are extracted in order to calculate the transformation matrix associated with the alignment within the overall framework of 3D images, and form the map of the environment. Figure 7 shows the correspondences found between two different scenes using the FPFH descriptor.

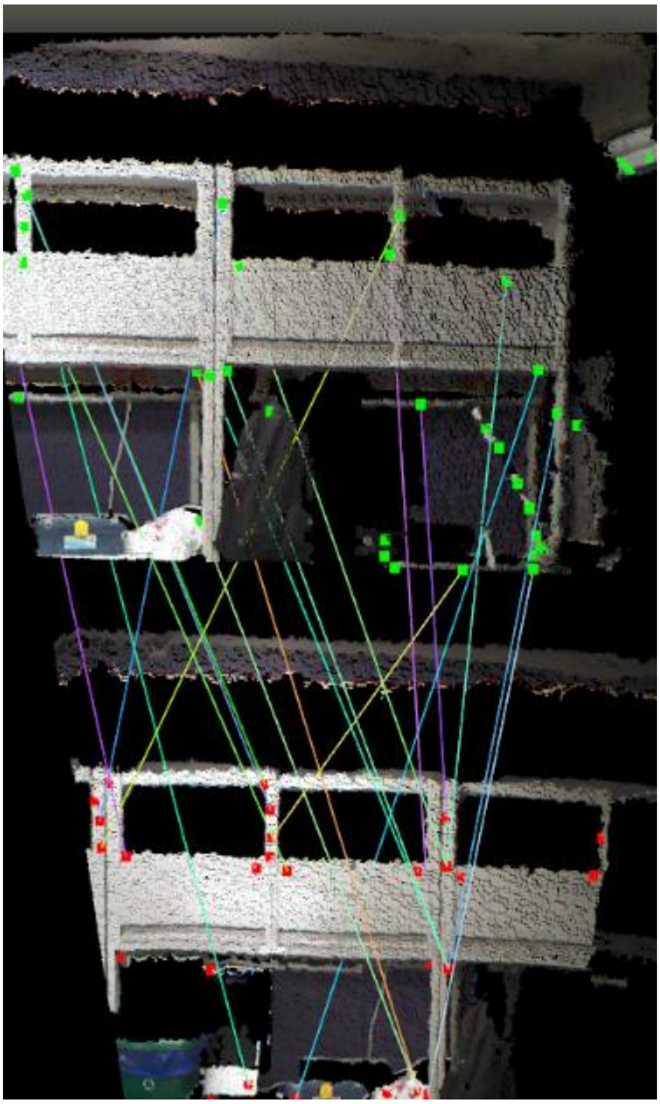
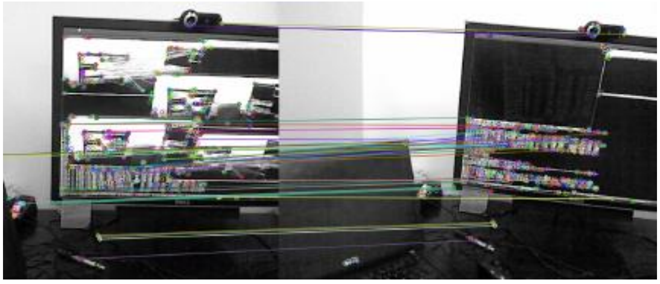


Figure 7: Two captures acquired with the Kinect sensor and their associated correlations.

Although there are correspondences deemed as correct at first glance, there may be estimation errors, which must be corrected. For these cases, it is necessary to implement a filtering correlation step in order to eliminate these errors for further processing of information. For this, the point cloud library has specific functions to perform this task. **Error! Reference source not found.** shows the implementation of the filtering step for two scenes captured.

(a)



(b)

Figure 8*:* Correspondence Extraction and Filtering. a) Unfiltered Correspondence. b) Filtered Correspondence. Taken fromhttp://pointclouds. org.

The calculation of the transformation matrix between the two captures is the last step in the feature-based registration, this process consists of finding the right angles and distances to correctly overlap the point cloud to join and shape one model to build an environment map. Figure 9 shows the two point clouds within a common global reference framework related through the features found.

Figure 9: Capturing point clouds after applying the transformation matrix.

Registration based on the ICP operation:

The development of an ICP-based operation or Iterative Closest Point algorithm has a list of basic steps for its building just as the feature-based registration

1. Search correspondences between captured point clouds.
2. Reject miscalculated correspondences.
3. Estimate the appropriate transformation using the filtered correspondences.
4. Iterate.

The difference between the two registration approaches lies in the developed processing operation, thus working along with the point matrix used as an input argument, while for the FBR, a point cloud pre-processing is performed in order to calculate some interest points to further processing.

The ICP operation estimates distances found between two point matrices, and from these data, minimize the error (decrease the distance among them) to join the matrices and overlap them, thus meeting the registration objective.

The steps listed above for the ICP operation are part of the unique features in the PCL library, making it impossible to use steps of the FBR registration in the ICP.

Registration algorythms

From the steps listed above, algorithms for the feature based registration and the ICP-based operation are developed. These algorithms are shown as a pseudo code for better understanding.

FBR algorithm:

1) Reading input point clouds and configuration parameters for processing
- For each configuration option (descriptor type) and point matrix
a) Read point matrix.
b) Run the down sampling operation to optimize the processing time (configured by the user).
c) Calculation of the NARF landmarks
i. Build a range image.
ii. Extracting borders from range image.
iii. Find landmarks.
iv. Save landmarks in vector.

2) Find landmark descriptors
- For the kind of descriptor selected by the user and applied to the landmarks extracted from each matrix.
a) Calculate the normal vectors for each of the landmarks; if the selected descriptor is the PFH or the FPFH type.
b) Enter the normal vectors as input parameter to the processing method (whether PFH or FPFH); if the descriptor is the NARF type, input the landmarks directly.
c) Calculate descriptors and save them in vectors.

3) Correspondence Extraction
- For each vector data with descriptors for each matrix.
a) Enter the vectors with the data of the descriptors as input parameter.
b) Establish the search of the correspondences through the kd-tree method.
c) Calculate correspondences.
d) Save the calculated data in the correspondence vector.

4) Correspondence Filtering
- For the correspondence vector.
a) Set the RANSAC method; SAC rejector.
b) Enter the landmarks and correspondence vector as input parameters.
c) Save the remaining correspondences after filtering.
d) From the remaining correspondences, calculate the transformation matrix.

5) Apply the calculated transformation
a) Save the transformation matrix.
b) ICP-based Registration algorithm:

1) Reading input point cloud input
- For each point cloud matrix:
d) Read point matrix.
e) Run the downsampling operation to optimize the processing time.
f) Determine point features; estimate the normal vectors for each point in the input matrix.
g) Save the data from normal calculated in the vectors.

2) Estimate the transformation among point clouds
a) Set method; use configuration variables and assign values for processing.
b) Enter the calculated vectors in step 1 as input to the processing algorithm.
c) Calculate processing recursively; iterate.
- For each processing cycle
i. Calculate the alignment between the previous saved state and the current state after calculating transformation
ii. Save the current state (vector after applying transformation) as a variable memory of the state.
iii. Run the transformation with the recently calculated alignment.
iv. Minimize the constant error by subtracting a unit from the current distance separation.

3) Apply calculated transformation
c) Save last transformation matrix.
d) Apply the transformation matrix to align the point clouds.
e) Save point clouds after calculating the transformation as a new point cloud file.

Results

As an example of the results obtained considering the proposed algorithms, the processing of point clouds for the acquisition of scenes of a library is displayed.

Feature based algorithm (FBR):

The purpose of performing an algorithm as the one proposed in the previous section for feature-based registration, is to provide the user with a tool for estimating correspondences across the various methods described herein.

Accordingly, this algorithm has four different input parameters for configuration, including the methods for calculating descriptors mentioned in the methodology section (NARF, PFH and FPFH), according to the estimation of the NARF type landmarks. The voxel grid which refers to the group of points in the input array is used as a parameter in order to reduce the rendering resolution, bringing advantages in the algorithm processing time, as will be shown below.

The point cloud samples chosen to test this algorithm are shown in **Error! Reference source not found.**.

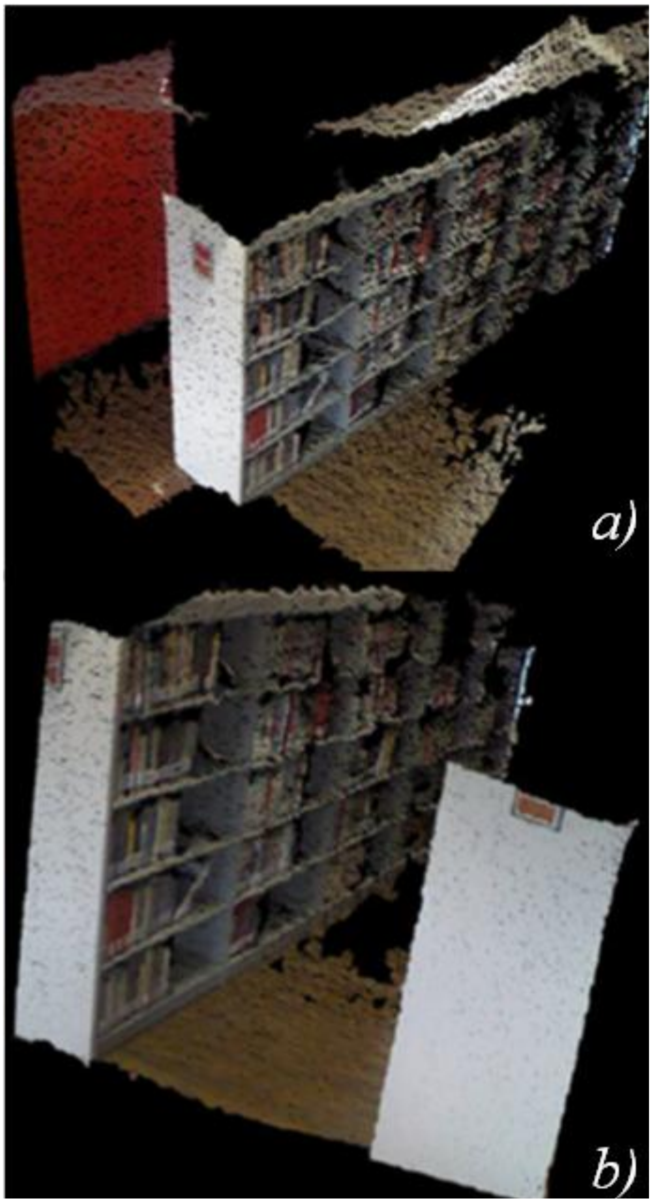


Figure 10: Scenes captured from a library.

The landmarks found for the two scenes of Figure 10were calculated using the NARF type method for estimating points. The landmarks of **Error! Reference source not found.**. a, and **Error! Reference source not found.**. b, can be seen in **Error! Reference source not found.**. a and **Error! Reference source not found.**. b respectively.

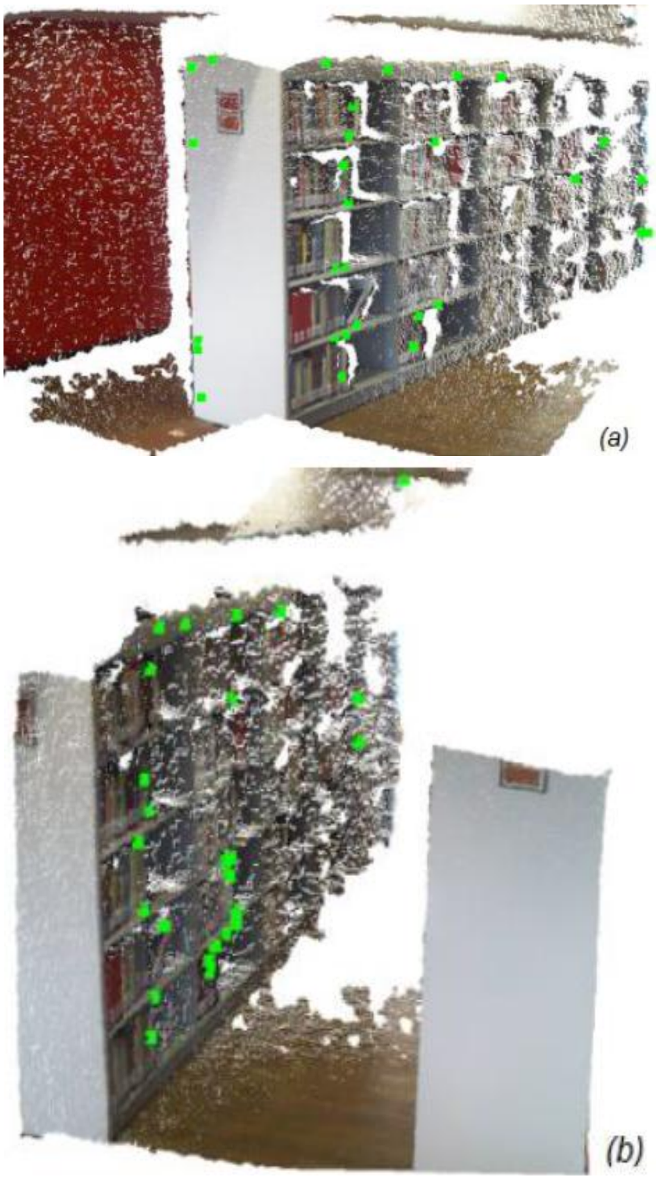


Figure 11: Landmarks of the library scenes.

After getting the landmarks of each of the scenes as shown in Figure 11, it is possible to continue with the calculation of the descriptors from the methods set out for PFH, FPFH and NARF, in order to calculate different correspondences as shown in Figure 7.

The results of the algorithm processing using different types of configuration are summarized in Tables 1-3, the configuration parameters were changed for each type of descriptor to be processed, according to the size of the voxel grid entered as a parameter, considering the values 0. 01, 0. 007 and 0. 005.

The processing values for the NARF descriptor are shown in Table 1, summarizing the different processing times taken by the correspondence processing from this descriptor, the larger the voxel grid, the lower the processing time, affecting significantly the estimation of correct correspondences. Therefore affecting the calculation of the transformation matrix, leading to errors while mapping the environment.

Table 1: Data Processing for NARF descriptor.

Voxel grid NARF	0. 005	0. 007	0. 01
Time (s)	1. 3	1. 03	0. 99
Correspondences	3	3	1

Table 2 shows the processing results using the PFH descriptor, and unlike the NARF descriptor, few errors occur in this process while making size variation of the voxel thus obtaining constant values of correspondences in all the tests.

Table 2: Data Processing for PFH descriptor.

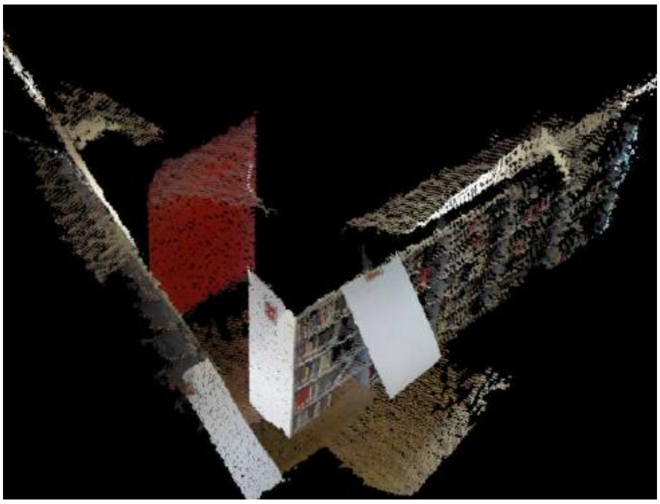
Voxel grid PFH	0. 005	0. 007	0. 01
Time (s)	4. 54	2. 92	1. 98
Correspondences	2	3	3

Table 3 shows the values of the tests performed with the FPFH descriptor, being the one with the highest processing time and the one that yielded the best results in the estimation of correspondences, obtaining the higher value of the three proposed methods.

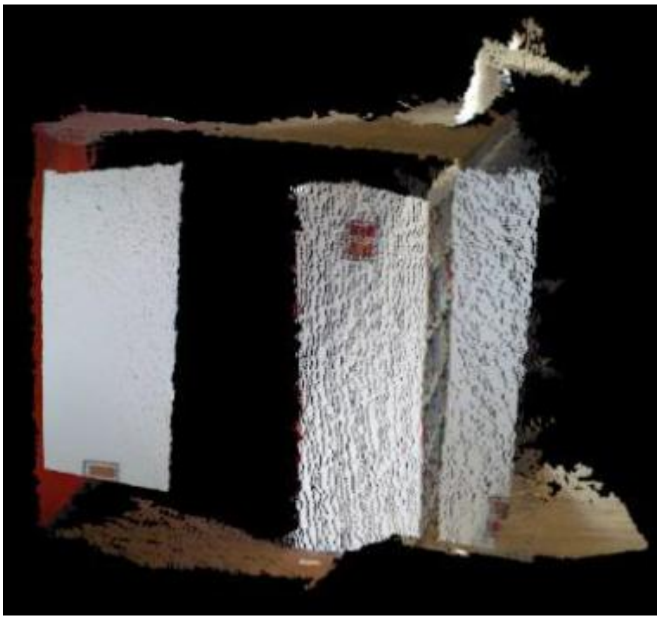
Table 3: Data Processing for FPFH descriptor.

Voxel grid FPFH	0. 005	0. 007	0. 01
Time (s)	5. 17	3. 19	2. 07
Correspondences	2	4	2

Figure 12 shows the result in the registration stage by applying the transformation matrix calculated from the different descriptors implemented, displaying clear and observable errors related to the calculation of the transformation matrix, errors caused by the correspondence estimation, suggesting a further development on the implementation of the landmarks, which are the basis of all the development.



(a)



(b)

(c)

Figure 12: Registration a) NARF descriptor b) PFH descriptor c) FPFH Descriptor

ICP based registration:
Unlike the registration algorithm based on features, this second implementation using the iterative closest point operation as shown below, delivers better results when using point clouds required for the registration. Figure 13 shows the scenes to be registered using this processing algorithm.

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 11, Number 8 (2016) pp 5704-5713

Figure 13: scenes to be registered using the ICP-based algorithm.

To perform the test processing and considering that the ICP algorithm is run in pairs of two point clouds, data capturing arrays points of Figure 14, thus proposing and ensuring that all point clouds are associated.

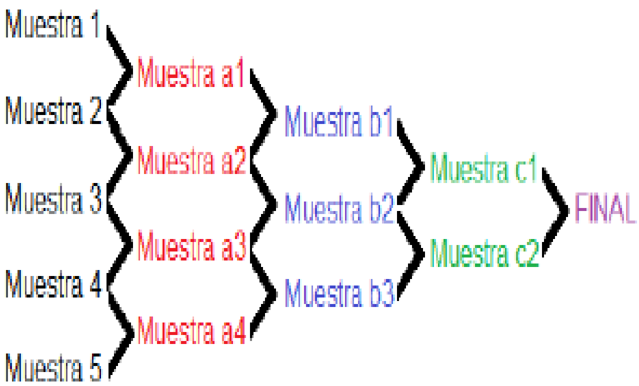


Figure 14: Step processing of the scenes to be registered.

The result of the conducted processing using the diagram of Figure 14is shown in **Error! Reference source not found.**. Figure 15 shows a clear association between the landmarks and the point clouds, although there is a processing error in one of the bookcases, probably due to the capturing method, due to sensor position when performing such capture.

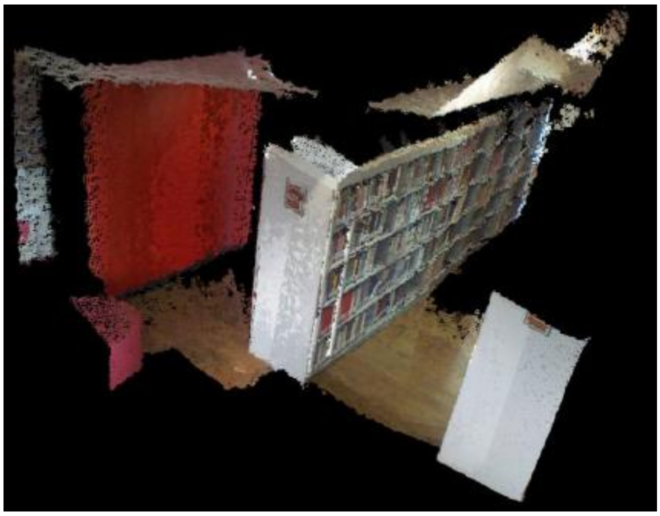


Figure 15: Final Point Cloud after applying the ICP based registration.

This registration method shows better results, however the computational burden is too high, compromising processing times and making impossible their use in real-time tasks, therefore many options for the improvement of this algorithm and its implementation are available, using other operating characteristics including the joint management of the two processing approaches, in other words, a fusion between the FBR and the PCI registration.

Conclusions and future work

Regarding the feature based registration and despite the comparison between the different descriptors, there are different aspects to be considered such as the voxel grid size, since it is possible to deduce that the smaller the size, the higher the resolution of the points, in other words more information to be used for processing leading to better results, however this does not apply to all methods of implemented descriptors, for instance, for the PFH and FPFH descriptor there were better results using a voxel grid intermediate value (0. 007), thus leading to different errors as for the landmark used, which is not robust in terms of the resolution operations performed in the point cloud. On the other hand it was possible to identify that although there were no equal number of correspondences using the PHF descriptor, this proves to be the most robust while changing point cloud resolution, becoming and implementable method for the registration in SLAM applications with the necessary changes to find better results while estimating the global framework of the point clouds to be registered

The registration based on the ICP operation showed better results, since it is a robust and implementable algorithm on post-processing tasks, due to computation times that are managed in order to get results like those shown in Figure 15.

The different configurations used in the algorithms proved to be necessary and constitute an important factor in the development of a registration stage that may be usable in real time. For this reason it is proposed to improve the algorithms to increase the processing and therefore the mapping results,

which implies further work while understanding the methods and used resources, likewise it is proposed the use of a GPU to reduce processing times for both the FBR and ICP algorithm.
Despite all the considerations raised, the algorithms proved to be important elements for the analysis of possible improvements for the development of a mapping stage as a first approach to SLAM, being the ICP based algorithm the best option at the time for the off-line processing of the information.

Bibliography

[1] S.-M. Lee, J. Jung, S. Kim, I.-J. Kim, and H. Myung, “DV-SLAM (Dual-Sensor-Based Vector-Field SLAM) and Observability Analysis, ” IEEE Trans. Ind. Electron., vol. 62, no. 2, pp. 1101-1112, Feb. 2015.
[2] G. Silveira, E. Malis, and P. Rives, “An Efficient Direct Approach to Visual SLAM, ” IEEE Trans. Robot., vol. 24, no. 5, pp. 969-979, 2008.
[3] L. M. Paz, J. D. Tardós, and J. Neira, “Divide and conquer: EKF SLAM in O(n), ” IEEE Trans. Robot., vol. 24, no. 5, pp. 1107-1120, 2008.
[4] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, “A tutorial on graph-based SLAM, ” IEEE Intell. Transp. Syst. Mag., vol. 2, pp. 31-43, 2010.
[5] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping (SLAM): part I The Essential Algorithms, ” Robot. Autom. Mag., vol. 2, pp. 99-110, 2006.
[6] D. Schleicher, L. M. Bergasa, M. Ocaña, R. Barea, and M. E. López, “Real-time hierarchical outdoor slam based on stereovision and GPS fusion, ” IEEE Trans. Intell. Transp. Syst., vol. 10, no. 3, pp. 440-452, 2009.
[7] C.-C. W. C.-C. Wang and C. Thorpe, “Simultaneous localization and mapping with detection and tracking of moving objects, ” Proc. 2002 IEEE Int. Conf. Robot. Autom. (Cat. No. 02CH37292), vol. 3, 2002.
[8] M. W. M. Gamini Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, “A solution to the simultaneous localization and map building (SLAM) problem, ” IEEE Trans. Robot. Autom., vol. 17, no. 3, pp. 229-241, 2001.
[9] K. Berns and E. von Puttkamer, “Simultaneous localization and mapping (SLAM), ” Auton. L. Veh., pp. 146-172, 2009.
[10] a. J. Davison and D. W. Murray, “Simultaneous localization and map-building using active vision, ” IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 7, pp. 865-880, 2002.
[11] D. Wolf and G. S. Sukhatme, “Online simultaneous localization and mapping in dynamic environments, ” IEEE Int. Conf. Robot. Autom. 2004. Proceedings. ICRA ’04. 2004, vol. 2, 2004.
[12] I. Vallivaara, J. Haverinen, A. Kemppainen, and J. Roning, “Simultaneous localization and mapping using ambient magnetic field, ” 2010 IEEE Conf. Multisens. Fusion Integr., pp. 14-19, 2010.
[13] S. Kean, J. Hall, and P. Perry, Meet the Kinect: An Introduction to Programming Natural User Interfaces. 2011.
[14] J. Smisek, M. Jancosek, and T. Pajdla, “3D with Kinect, ” Proc. IEEE Int. Conf. Comput. Vis., pp. 1154-1160, 2011.
[15] I. a. Syed and B. Sharma, “Hybrid 3D registration approach using RGB and depth images, ” 2013 IEEE 2nd Int. Conf. Image Inf. Process. IEEE ICIIP 2013, pp. 27-32, 2013.
[16] M. Li, R. Lin, H. Wang, and H. Xu, “An efficient SLAM system only using RGBD sensors, ” vol. 215006, no. December, pp. 1653-1658, 2013.
[17] C. C. Liebe, C. Padgett, J. Chapsky, D. Wilson, K. Brown, S. Jerebets, H. Goldberg, and J. Schroeder, “Spacecraft hazard avoidance utilizing structured light, ” 2006 IEEE Aerosp. Conf., 2006.
[18] L. A. Góngora Velandia, “localización y mapeo simultáneo basado en la librería de nube de puntos (PCL): etapa de registro., ” Universidad Piloto de Colombia, 2014.
[19] J. J. Leonard and H. F. Durrant-Whyte, “Simultaneous map building and localization for an autonomous mobile robot, ” in Proceedings IROS ’91:IEEE/RSJ International Workshop on Intelligent Robots and Systems '91, pp. 1442-1447.
[20] R. B. Rusu, “Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments, ” Technische Universität München, 2009.
[21] B. S. Radu, B. Rusu, K. Konolige, and W. Burgard, “NARF: 3D Range Image Features for Object Recognition, ” in Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics at the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2010.