Contemporary Engineering Sciences, Vol. 10, 2017, no. 31, 1509 - 1516
HIKARI Ltd, www.m-hikari.com
https://doi.org/10.12988/ces.2017.711180

Planes Detection SLAM for Minimal Data Storage for Indoors Applications

Diego Mauricio Rivera Pinzón

Universidad Pedagógica Nacional, Bogotá, Colombia

Nubia Nathaly Sánchez Galvis

Universidad Pedagógica Nacional, Bogotá, Colombia

Edwar Jacinto Gómez

Universidad Distrital Francisco José de Caldas, Bogotá, Colombia

Abstract

Slam systems are techniques that can geometrically reconstruct 3d maps of indoor scenes. In this work a slam algorithm is proposed that aims to generate a computational model of the interior of a building using as sensor a RGB-D device mounted on an unmanned aerial vehicle. The idea of the algorithm is to model the scenario using flat figures with irregular contours, for this the coordinates of the point cloud are not used but the parametric equations of the planes. The extraction of planes is done by calculating the gradient and later using unsupervised classification techniques. The information of the points of the contours and the information of the geometric parameters is detected and stored for each plane found in the scene. The information stored is sufficient to later reconstruct the entire scene. The advantage of this technique is that it needs very low memory to store a complete scenario, which can be useful for working with embedded systems or in large scenarios.

Keywords: SLAM, Planes detection, Unmanned Aerial Vehicle

1 Introduction

Slam is a technique whose objective is that if a robot moves within a scene and can build a map of the unknown environment and simultaneously locate its position on that same map. It has a wide range of applications including obstacle avoidance, surveillance and inspection, augmented reality, search, rescue and recognition [3]. Detecting flat surfaces of a point cloud is a common problem in computer vision and robotics. Because indoor scenes are usually composed of multiple planes [4] [13], which can be used to find a simplified model and as a strategy to manage the sensor's own noise [1]. The RANSAC algorithm has been widely used to adjust planes in 3d points, among them [1], [5], [8], [10], [12]. There are several alternatives to the use of RANSAC, for example: A point-based growing region (PBRG) [14] and connected component labeling [6]. Several of these techniques generate a model based on the point cloud, while others seek to produce a geometric model based on closed surfaces.

It is proposed the generation of a geometric model of the scene, the model is composed entirely of planes of different sizes and arbitrary contours. The planes are extracted from each frame using unsupervised classification algorithms using the gradient values obtained from the depth information. The fusion of frames is done by looking for features between several RGB images, which are converted to 3D coordinates, which allows us to use least squares to find the transformation matrix. Finally, the information of the complete planes is extracted and its αshape is extracted projecting the points of the planes to 2D as it describes [12].

The article continues as follows: Initially, a general description of each step of the procedure for the extraction of flat regions of a depth image is presented. Subsequently, the iterative process allows the fusion between planes finding correspondence between consecutive frames is presented. Finally, the details of the selection of parameters are exposed, discussing the variation of each of them and the criteria that determine their choice.

2 Methodology

2.1 Finding flat regions in a scene

Data Acquisition: For this implementation a Kinect sensor was used, the data was converted to a cloud of points X, Y and Z coordinates [7].

Planar surfaces classification: Each plane that composes the image contains gradient information which is different for each plane, this can be used as a criterion to perform the classification of the possible planes that are in the point cloud. The coordinate matrix Z contains the information of the neighborhood of the points, which is used to calculate the gradient, as result two data are obtained for each point of Z that correspond to the directional derivative. The idea is to work the extraction of planes as a problem of non-supervised pattern classification, for this it is possible to make a change of space, considering the values of directional derivative as characteristics of the planes, where each of the planes will generate a different cluster.

The number of planes may exist in the scene may change depending on the environment and the sensor position, i.e., the number of clusters per iteration is unknown. It is necessary to select a classification technique that never detects fewer classes than those that may exist in the scene. If it detects more classes of the truly existing ones, this algorithm, in a later step, will find the correspondence and join the classes. The concrete procedure to perform the calculation of the classification of the flat figures can be summarized in the following steps:

1. Take only the data matrix corresponding to the coordinates in Z from the point cloud (Fig. 1.a).
2. Directional derivatives are calculated for each point of Z (Fig. 1.b).
3. Directional derivatives are considered as the characteristics of a plane and are represented as clusters (Fig. 1.c).
4. A classification algorithm is used that overclassifies the samples.

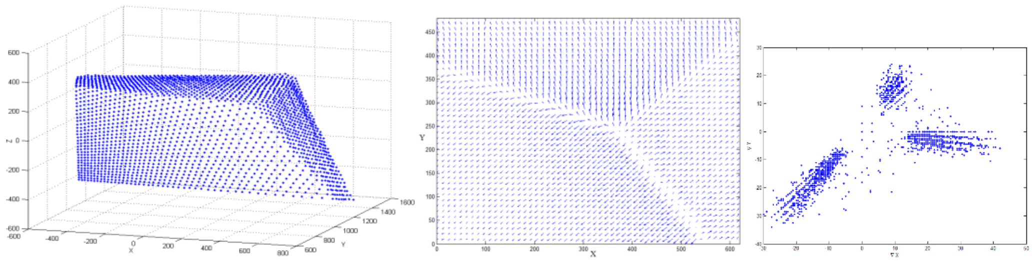

Figure 1. (left) Point cloud of one frame, (center) directional derivative map (right) directional derivative representation as clusters.

Coefficients of the parametric equation for each plane: in this point there are two vectors, one that contains the cloud of points and another that contains the classification of each point relating it to a plane. For each class detected, the associated points are used to find coefficients that allow a parametrization of the plane, it is necessary to use some regression technique. In Figure 2 the equation Z = AX + BY + C is used to adjust the data of one of the planes found in the scene. A vector containing the coefficients that allow determine the approximate equation that best fit plane points of each class is obtained at the end.

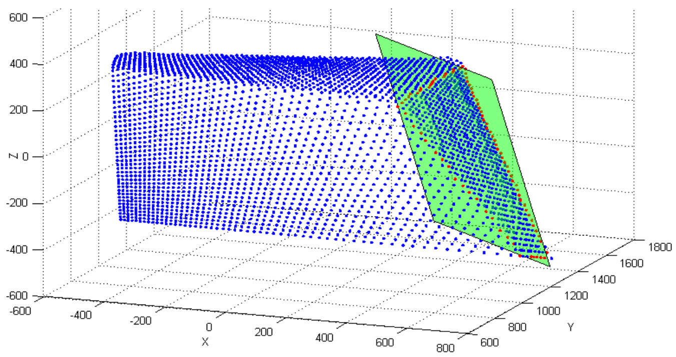

Figure 2. Point cloud with one plane found

Union of flat shapes that correspond to each other: Due to the over classification established it is possible that several points of the same plane have been separated as if they belonged to different planes. To make the union of the groupings of characteristics, the Euclidean distance normalized to the points between them is calculated, and later using a threshold value to join the ones that are closest to each other.

Calculation of the edge of each plane: Once the association of each point to a plane belonging to the scene is counted, the information of the points that are on the edge of each plane is extracted. The process of edge detection has two fundamental purposes: First, it is sought to store the least amount of information of each plane, the information of which points are on the edge is sufficient to perform a subsequent reconstruction of the plane. Secondly, in the joining process it is possible to consider as one only some planes that have the same parametric representation coefficients but that are not neighbors among them, for example, a wall that has a column in the middle or a door between two walls. In Figure 3 an example with the result of the algorithm is presented, there are three different views of a scene in which three different planes are found, a reconstruction of the scene is performed using the information of what are the parameters of the equation of each plane, and what are the points that are on the edges.

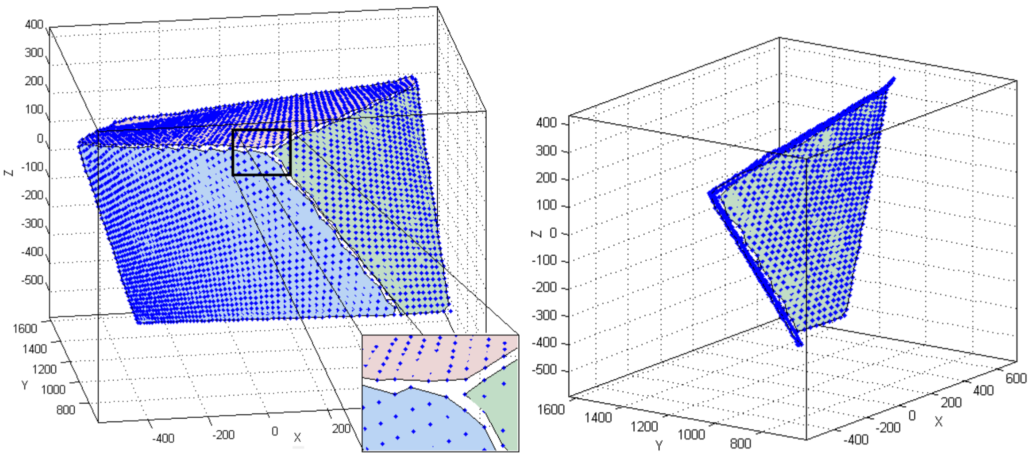

Figure 3. Two different views of the output

2.2 Correspondence between frames

It proceeds to work with multiple frames and their respective planes, it is necessary to find what has been the displacement made and the change in the orientation between the successive frames. For this, two images delivered by the Kinect sensor, an RGB color image and D-range image previously calibrated are processed. It is expected to find a rigid transformation matrix M composed of a rotation and a translation, which allows to determine which is the displacement and orientation between two consecutive images.

Search for points of correspondence between the RGB images: It is possible to find elements of correspondence between two images based on features, the features that are used more frequently are the edges and points of interest. The output that is expected in this step is a vector, in which the coordinates of the corresponding points of interest are stored in the two images. A variety of methods have been proposed in the literature to perform this step. In [11] a study and comparison of algorithms that can be used for this purpose is presented.
Search for RGB equivalences with the point cloud: The existing mathematical relationship between the RGB image and the depth image D is available, using this information it is possible to take the position of a pixel in RGB and find its equivalent reference as a point in the XYZ point cloud.
Calculation of the rigid transformation matrix: Using the information of the location of the descriptors common to the two images, the information of the corresponding points is obtained in each point cloud associated with a frame captured by the sensor. It is possible to find which is the transformation in position and in orientation between the frames, for this process it is necessary to use a technique in which, using known control points and their correspondence with other points, it allows to find which is the transformation matrix M that relates them. Because the correspondence between the points is known, the problem can be addressed from the solution through least squares, without the need to use techniques such as ICP (Iterative closest point).

3 Discussion

The explanation of the steps of the algorithm was carried out in a general way, in each of them there is a great variety of parameters that determine multiple aspects such as the speed of execution, the precision of the calculations and the general performance. Below is a description of the possible variants that each step may have in terms of coefficient values or possible algorithms.
Conditioning of the sensor data: working with a sub-sampled image, leads to improve the execution speeds of the algorithm, having as a disadvantage that accuracy is lost in the calculations made and in the final model obtained at the edges of the planes. Figure 4 shows two different results apply same technique to a stairs scene but changing de subsample.

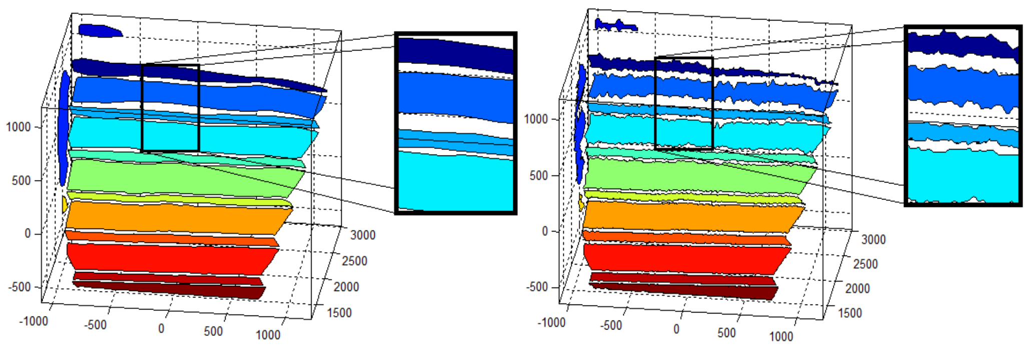

Figure 4. Stair scene reconstruction (left) full resolution. (right) half resolution.

Classification of the planes: The selected cluster technique was a variation of the k-means algorithm, which increases its computation speed [15], using as input values the feature data vector composed of the directional derivatives and the maximum number of clusters selected. Obviously, the result will result in the over classification of the flat figures. Empirically it was found that it is faster to use an over classification and later to join the planes found, compared with finding directly the number of classes in the classification with hierarchical techniques.
Coefficients of the parametric equation: For the implementation of this step, an algorithm based on least squares was used due to two reasons: a) the number of outliers in the data is very low and does not affect the performance of the approximation. b) For computational time reasons it is preferable to use a subset of data to perform the calculation with least squares compared to the use of techniques that consider outliers to find planes such as RANSAC [9] [16].
Calculation of the edge of each plane: The notion of αShape was used which is used to perform the reconstruction of a finite set of points without organization within a certain grouping, where the points that make up the αShape demarcate the border that contains the rest of the points which allows to determine the original form of the grouping. The calculation of αShape uses the triangulation of Delaunay fact that only one circumference can be found that passes through the vertices of each triangle, the value of a threshold of the radius of this circumference can be used to determine which points belong to the border and which do not.
Depending on the selected radius value, it is possible not only to obtain a contour but also to divide the shape into several contours, which in this case can be used to separate planes that are joined in the previous step. A case in which this phenomenon can occur is when you have two walls that are joined by a column, the two have the same values in their coefficients of the parametric equation, but belong to different planes. For the implementation of this algorithm, the input argument of the αShape are only the X and Y coordinates of the points in each plane, the value of the Z coordinate is not considered, which is equivalent to making a projection of the points with respect to the $z = 0$ plane, this substantially improves the speed of the algorithm by generating practically the same results as using all the coordinates. Finally, the information that is preserved for the next iteration are the αShape points and the coefficients of the parametric equation of each detected plane.
Correspondence between images: For the search of corresponding points between the RGB images the SIFT technique (Scale Invariant Feature Transform) described in [2] was used, whose idea is the transformation of an image to a representation composed of points of interest, these points They contain the characteristic information of certain parts of the image, which can then be used for the detection of correspondences. The same parameter values suggested by the author were used. For the Equivalence Search of the points in RGB with the XYZ points, it must be considered that the depth image D does not contain information for all the coordinates, usually information is lost in the edges of the objects, in the reflecting surfaces and in the distant objects. It is possible that some of the descriptors are located on coordinates in the color image that have no associated XYZ. Those descriptors that do not have an association in the point cloud are discarded.

Conclusions

In this article an algorithm to perform the reconstruction of a scene is presented, which can later be used as an input to the trajectory planning algorithm. Initially it was presented in a general way, showing each of its stages. Later it was shown that there are a great variety of parameters that determine multiple aspects such as speed of execution, accuracy of calculations and general performance.

The reliability of the conversion of the data taken by the distance sensor when passing them to spatial coordinates decreases as the distance to the focal point increases, additionally, points are lost at the edges of objects and in areas very far from the sensor, as a result there are regions in which depth information is not available or high measurement noise occurs. One way to minimize the noise of data at a great distance is by conducting a thresholding with the points being discarded which is the greatest distance from the sensor.

References

[1] H. E. Elghor, D. Roussel, F. Ababsa, and E. H. Bouyakhf, Planes Detection for Robust Localization and Mapping in RGB-D SLAM Systems, *2015 Int. Conf. 3D Vision*, (2015), 452–459. https://doi.org/10.1109/3dv.2015.73

[2] D. G. Lowe, Distinctive image features from scale invariant keypoints, *Int. J. Comput. Vis*., **60** (2004), no. 2, 91–110. https://doi.org/10.1023/b:visi.0000029664.99615.94

[3] Michael Kaess, Simultaneous Localization and Mapping with Infinite Planes, *2015 IEEE Int. Conf. Robot. Autom*., (2015), 4605–4611. https://doi.org/10.1109/icra.2015.7139837

[4] T. T. Pham, M. Eich, I. Reid and G. Wyeth, Geometrically consistent plane extraction for dense indoor 3D maps segmentation, *IEEE Int. Conf. Intell. Robot. Syst.*, (2016), 4199–4204. https://doi.org/10.1109/iros.2016.7759618

[5] Po-Chang Su, Ju Shen, Sen-ching S. Cheung, A Robust RGB-D Slam System for 3D Environment with Planar Surfaces, *2013 IEEE International Conference on Image Processing*, (2013), 275–279. https://doi.org/10.1109/icip.2013.6738057

[6] R. F. Salas-Moreno, B. Glocker, P. H. J. Kelly and A. J. Davison, Dense Planar SLAM, *Proc. Int. Symp. Mix. Augment. Reality*, (2014), 157–164. https://doi.org/10.1109/ismar.2014.6948422

[7] L. Shibo and Z. Qing, A new approach to calibrate range image and color

image from kinect, *Proc. 2012 4th Int. Conf. Intell. Human-Machine Syst. Cybern.* **2** (2012), 252–255. https://doi.org/10.1109/ihmsc.2012.156

[8] L. A. V. Souto and T. P. Nascimento, Object Subtraction Planar RGB-D SLAM, *2016 XIII Lat. Am. Robot. Symp. IV Brazilian Robot. Symp*., (2016), 19–24. https://doi.org/10.1109/lars-sbr.2016.11

[9] Y. Suttasupa, A. Sudsang and N. Niparnan, Plane detection for Kinect image sequences, *2011 IEEE Int. Conf. Robot. Biomimetics, ROBIO 2011*, (2011), 970–975. https://doi.org/10.1109/robio.2011.6181413

[10] A. J. B. Trevor, J. G. Rogers and H. I. Christensen, Planar surface SLAM with 3D and 2D sensors, *2012 IEEE International Conference on Robotics and Automation*, (2012), 3041–3048. https://doi.org/10.1109/icra.2012.6225287

[11] T. Tuytelaars and K. Mikolajczyk, Local Invariant Feature Detectors: A Survey, *Found. Trends Comput. Graph. Vis*., **3** (2007), no. 3, 177–280. https://doi.org/10.1561/0600000017

[12] C. Ulas and H. Temeltas, Plane-feature based 3D outdoor SLAM with Gaussian filters, *2012 IEEE Int. Conf. Veh. Electron. Safety, ICVES 2012*, (2012), 13–18. https://doi.org/10.1109/icves.2012.6294326

[13] J. Weingarten and R. Siegwart, 3D SLAM using planar segments, 2006 *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, (2006), 3062–3067. https://doi.org/10.1109/iros.2006.282245

[14] J. Xiao, B. Adler and H. Zhang, 3D point cloud registration based on planar surfaces, *2012 IEEE Int. Conf. Multisens. Fusion Integr. Intell. Syst.*, (2012), 40-45. https://doi.org/10.1109/mfi.2012.6343035

[15] J. Xie and S. Jiang, A Simple and Fast Algorithm for Global K-means Clustering, *2010 Second Int. Work. Educ. Technol. Comput. Sci.*, (2010) 36–40. https://doi.org/10.1109/etcs.2010.347

[16] L. G. Zhao and C. K. Wu, A Planes Detection Algorithm Based on Feature Distribution, *2011 Third Int. Conf. Intell. Netw. Collab. Syst.*, (2011), 172–177. https://doi.org/10.1109/incos.2011.33

Received: December 1, 2017; Published: December 17, 2017