Autonomous Scanning of Structural Elements in Buildings

B. Quintana(✉), S.A. Prieto, A. Adán, and A.S. Vázquez

3D Visual Computing and Robotics Lab, Universidad de Castilla-La Mancha,
Ciudad Real, Spain
{Blanca.Quintana, Samuel.Prieto, Antonio.Adan,
Andress.Vazquez}@uclm.es

Abstract. This paper presents a method for automatic scanning of structural elements in indoors. Whereas most of the next-best-scan (NBS) based methods do not separate clutter and useful data, we present a scanning strategy in which potential structural components (SE) of the building are recognized as a new scan is carried out. This makes our method more efficient and less time consuming compared with the rest. Besides, our approach gives a response to essential issues in the scanning world, such as the data discrimination, the hypotheses about the workspace and the complexity of the scanned scene. The method has been tested in indoors under occlusion and clutter yielding promising results. Additionally, a comparison with three techniques close to ours is included in the paper.

Keywords: 3D sensing · 3D data processing · Next best view · 3D modeling

1 Introduction: Key Points and Contributions

Nowadays automatic scanning of buildings is a challenging and very active research in which there are still underlying questions (scan's objective, hypothesis and the scene complexity) that are rarely debated and that determine the validity of the method.

In the majority of the approaches the scanning strategy does not depend on the final objective [1] (preliminary version of our approach). Thus, the objective is frequently to scan everything which lies inside [2] or outside [3] the building. Data redundancy and cluttering are therefore ignored. Those methods are inefficient because a great part of the gathered 3D data can be irrelevant for the final goal. In contrast with these methods, our proposal aims to capture the data belonging to structural elements of the scene (i.e. ground, walls and ceiling) to further generate a realistic 3D CAD model of the building. Consequently, we can highly reduce the volume of data and simplify the algorithmia of future processes.

Hypotheses in the scanning process determine the soundness and versatility of a proposal. Some NBS algorithms a priori assume bounding boxes or convex-hulls that contain the scene [2]. Others take manually a set of preliminary sparse scans of the environment and later tackle the NBS algorithm [3].

© Springer International Publishing AG 2017

C. Beltrán-Castañón et al. (Eds.): CIARP 2016, LNCS 10125, pp. 60–67, 2017.
DOI: 10.1007/978-3-319-52277-7_8

An important issue related with this idea, ignored in most of the papers, is the updating of the workspace with a new scan. This makes the earlier methods less reliable and credible in real scenarios composed of several irregular adjacent rooms. In this matter, we define a dynamic workspace that contains the accumulated point cloud and that it is updated with new scans. Therefore, the boundaries of our workspace are not hypothesized and fixed but they are updated as a new scan is added.

The complexity of the scene is determined essentially by the shape of the room, the occlusion properties and the number of rooms to be scanned. Depending on the complexity of the geometry of the sensed area, we find simple [4] or complex 3D data processing [3, 5]. As regard interiors, most of the works deal with scenes composed of a corridor and several rectangular rooms connected to it [5–7]. However, indoors composed of concave rooms are rarely addressed in the field of 3D reconstruction. An exception can be found in the work by Jun et al. [8]. Besides, not all the approaches are able to overcome occlusion and clutter problems. [4, 9] and only a few of them consider obstacles or clutter in the scene [6, 7].

Our proposal is able to deal with concave rooms connected by doors in a building story. Due to this type of scenes our approach addresses the occlusion issue from the beginning. Figure 1 shows a prototype of the scene in which we are testing our automatic scanning method.

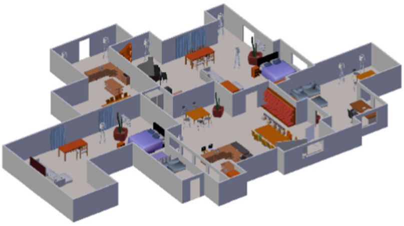


Fig. 1. Prototype of the scenario in our work: story with several non-rectangular inhabited rooms connected by doors.

2 A Brief Overview of the Method

Although the NBS procedure is the main objective of the paper, this is just a part of a more complex system composed of a mobile robot that carries a 3D laser scanner.

We assume that the scenario is composed of several rooms and that the system carries out the complete scanning of the current room before passing to an adjacent one. Thus, when the room scanning process ends, the robot places under the doorframe that separates adjacent rooms and launches the first scan of a new room. To detect openings (in this case, doors) we follow the algorithm described in [10].

The automatic scanning of the current room is viewed as a cyclic process which begins with the data acquisition from a 3D laser scanner with a field of view of

360º × 100º, and ends with the output of the NBS algorithm, that is, the coordinates of the next scan position in the world coordinate system. The main stages of a room scanning process are shown in Fig. 2. The stages are: (1) raw point cloud preprocessing and alignment, (2) RoI (Region of Interest) definition, (3) wall identification, (4) space labeling and NBS decision. The next sections are devoted to briefly explaining these stages.

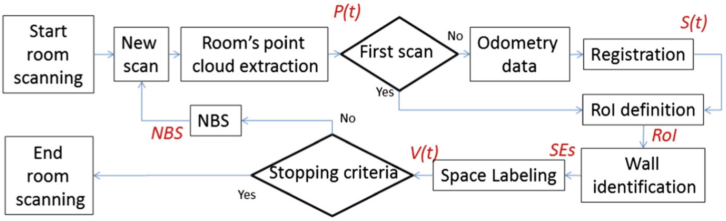


Fig. 2. Outline of the scanning process.

3 Structural Elements Recognition

3.1 Finding the Region of Interest

Our automatic scanning approach is a cyclic process in which each single point cloud coming from a new position of the scanner is registered into the accumulated point cloud of the scene. First, a coarse registration is carried out through the robot localization sensors. Then, the earlier registration is refined by applying a 6D (x, y, z, roll, pitch, yaw) ICP (Iterative Closest Point) technique [11]. We will denote *S(t)* as the accumulated point cloud at time *t*.

In this framework, the region of interest (RoI) is defined as the region which establishes the boundaries of the current space at time *t*. Thus, the RoI can be implemented as the prism that contains *S(t)*.

In practice, the RoI is obtained through the top projection of *S(t)*. The projection of the points is quantized over a horizontal grid that finally we convert into a binary image. See Fig. 3(a) for a better understanding. A polygonal contour is calculated in this image with the help of Hough and Harris' algorithms. On one hand, the segments that compose this contour lead us to obtain, first, the planes that fit to the data points and, second, the vertical parallelograms nearest the points. Note that, these contours themselves determine the polygons at the top and bottom of the RoI, which represents the ceiling and floor of the scene (right column of Fig. 3(a)). Figure 3(b) illustrates the RoI updating for three consecutive scans. The polygons that represent the ceiling and the floor are not shown for a better vision.

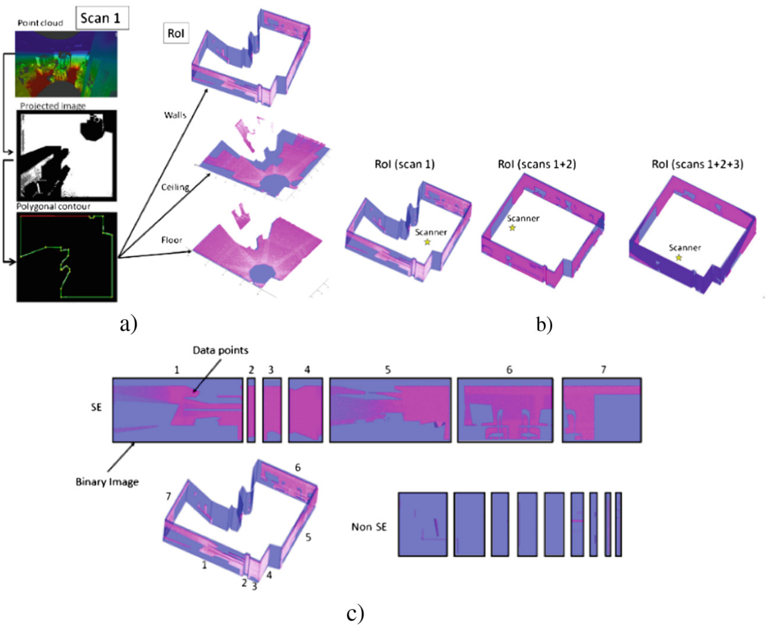


Fig. 3. (a) Steps to obtain the first RoI. The figure shows the data points (in magenta) superimposed to the faces of the RoI (in blue). (b) RoI evolution for the first three scans. (c) Structural elements classified in the first RoI. (Color figure online)

3.2 Structural Elements Classification

After having obtained the RoI, we identify which of its faces are structural elements (SE) and which are not. Note that, apart from all RoI's sides, we know the data points which are near to them. Each face and its associated data points generate a binary image in which we can infer whether the face is considered as SE or not. Of course, top and bottom polygons (which correspond to ceiling and floor) are a priori assumed structural elements, so that the SE classification is uniquely accomplished for walls.

The decision function that classifies the faces of the RoI as SE or non-SE has been implemented by means of a binary Support Vector Machine (SVM) classifier.

Let us assume *I* be a binary image generated from a polygon of the RoI in which a white (magenta in Fig. 3(c)) pixel means a data point. Let also assume d_1d_2 be the size of *I* (d_1 and d_2 are whatever image dimensions in pixels) and *n* be the number of white pixels contained in *I*. We consider a feature vector $F(\alpha, \delta, \varepsilon)$ that contains the following information:

- Occupancy percentage (α). This means the occupancy of the hypothetic wall. In Eq. (1), d_1 and d_2 are the image dimensions.

$$\alpha = \frac{n}{d_1 d_2} \tag{1}$$

- Clusters' compactness (β). We cluster the data points in *I* and calculate the density per cluster. The cluster's density is defined as the as the ratio of the number of data points (n_i) to the area of its corresponding bounding box ($d_{1i}d_{2i}$). Equation (2) provides the mean clusters density or the clusters' compactness (where *k* is the number of clusters).

$$\delta = \frac{1}{k}\sum\nolimits_{i}^{k} \frac{n_i}{d_{1i}d_{2i}} \tag{2}$$

- Data dispersion (ε). This is calculated through the area occupied by the bounding boxes of the clusters.

$$\varepsilon = \frac{\sum_i d_{1i}d_{2i}}{d_1 d_2} \tag{3}$$

Figure 3(c) shows a set of images *I* corresponding to the faces of the RoI for the first scan and the classification result. A deeper discussion and explanation of this method can be done in a more extended publication.

4 NBS

The NBS is computed in a discretized 3D space, that is, in a voxel-space V, where a voxel is a small cube. We define five labels in the space V, divided into two classes (occupied and non-occupied):

- Occupied voxels are: *Clutter* (the voxel contains points that do not belong to SEs) and *Structure* (the voxel contains points belonging to SEs).
- Non-occupied voxels are*:* *Empty* (the voxel has been seen from earlier scanner positions but does not contain data), *Occluded-clutter* (the voxel has not been sensed because it lies between an occupied voxel and a SE) and *Occluded-structure* (the voxel has not been sensed because it is behind an occupied voxel and lies in SEs).

To calculate the NBS we estimate the amount of *Occluded-structure* voxels that would turn into *Structure* voxels from a set of next valid positions (NVP) of the scanner. Since a mobile robot carries the scanner, a NVP is defined under robot path-planning requirements, that is, there must be at least one secure path from the current to the next position of the robot. A secure path entails that the robot moves through *Empty* voxels and the distance from such voxels to any occupied voxel along the path is higher than certain security distance (in our case 20 cm). From each NVP and by means of a ray-tracing algorithm, we calculate the number of conversions from *Occluded-structure* to *Structure* voxels. A ranked list of NVP is then established taking into account the amount of new *Structure* voxels. The NBS corresponds to the first NVP of the list. Figure 4 shows the labeled space V for two consecutive scans. *Structure* voxels are in blue, *Occluded-structure* in green and *Clutter* (*Occluded-clutter* is not shown for a better visualization). Note the increment of *Structure* voxels from the next best position of the scanner.

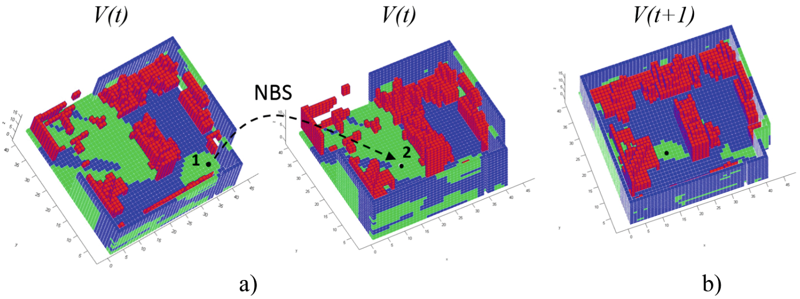


Fig. 4. (a) Current voxel-space V(t) with the current position (1) and the NBS position (2) in black (b) Next voxel-space V(t + 1). (Color figure online)

5 Test and Experimental Comparison

In this section we present an experimental comparison of our scanning approach with the ones of Stachniss and Burgard [12], Blaer and Allen [3] and Potthast Sukhatme [2], which can be considered related proposals. The comparison has been done in a scene composed of 5 adjacent concave-shape rooms, with clutter and occlusion (See Fig. 1(a)). This complex scenario has been created in Blender and its add-on Blensor [13]. This tool simulates real scanning with commercial 3D laser scanners similar to ours (Riegl VZ-400).

It is worth mentioning that, in order to make possible the experimental comparison, we needed to make some adaptations on those methods. Since the original version of those methods do not detect doors and besides impose a fixed-size occupancy grid for the scenario, we added our door detection algorithm to their codes and also updated the size of their voxel spaces with the beginning of a new scan. A brief report of the results follows.

Despite the adaptations, methods [2, 12] were not able to scan completely rooms #4 and #5. This is mainly owing to the fact that these approaches do not deal with concave-shape regions. Our method completed the scanning process taking one or two scan less than the rest per room. A total of 22 scans were taken for the whole scenario. Apart from ours, approach [3] was also able to complete the scanning process after taking 25 samples.

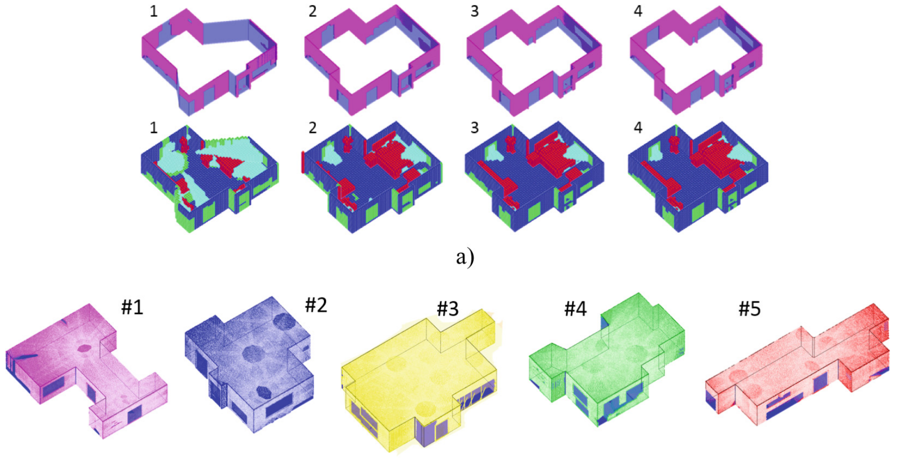
The computational cost was measured in terms of processing time. On this point, our algorithm spent less time than the others. Our total time for scanning and processing was 14868 s. We reduced 78,7%, 83,7% and 33,6% the time spent by [2, 3, 12] respectively. These reductions would have been much impressive without code adaptations.

Our percentage of the total structural surface sensed is also higher than the rest. We reached 88,57% compared with 87,26% for [4]. Note that Blaer et al.'s approach does not recognize structural elements. For rooms #1, #2 and #3, [2, 12] achieved 87,0% and 84,3% respectively. In summary, our method takes less scans and achieves higher percentages.

With regard to the size of the processed data points, we obtain smaller point clouds in most of the rooms. Note that we uniquely process SE points, whereas the rest of the approaches deal with the whole point cloud. We processed a total of 23.576.419 points. The mean reduction percentages in the number of processed points with respect to the other approaches were 7,2% [12], 23% [3] and 14% [2].

Concerning the path length, our approach yielded better results in rooms #1, #2, and #4. For rooms #3 and #5, the results were better than in [2, 12], and similar to the ones of [3]. We cover 30% less distance than [12] (43, 34 m versus 62, 28 m in rooms #1, #2 and #3), 25% less than [2] (32.71 m versus 43, 82 m in rooms #1 and #3) and 2,8% more than [3] (88, 26 versus 85, 82 m).

Figure 5 illustrates some details of the scanning process in room #2 and the whole point cloud of structural elements superimposed to the CAD model generated.



a)

b)

Fig. 5. (a) Evolution of RoI and Voxel-space of room #2 for four scans. (b) Data points assigned to structural elements in five different rooms and the corresponding CAD models superimposed to them.

6 Conclusions

The primary goal in our work is to accumulate data belonging to structural elements of a building to obtain a precise 3D model. The main contributions of our proposal are:

- A new NBS algorithm addressed to capture structural elements that highly reduces the volume of data and alleviates the algorithmic complexity in further processes.
- A dynamic RoI which allows effectively dealing with more complex scenarios composed of several non-rectangular spaces with occlusion and clutter.

Many aspects need to be improved in future developments. Our method works very well for flat structural elements but neither for curve shapes nor for several ceilings/floors within the same room. Therefore, new approaches for more complex scenes, including several stories, must be addressed in the future.

Acknowledgements. This work was supported by the Spanish Economy and Competitiveness Ministry [DPI2013-43344-R project] and by Castilla-La Mancha Government [PEII-2014-017-P project].

References

1. Adán, A., Quintana, B., Vázquez, A., Olivares, A., Parra, E., Prieto, S.: Towards the automatic scanning of indoors with robots. Sensors **15**(5), 11551–11574 (2015)
2. Potthast, C., Sukhatme, G.S.: A probabilistic framework for next best view estimation in a cluttered environment. J. Vis. Commun. Image Represent. **25**(1), 148–164 (2014)
3. Blaer, P.S., Allen, P.K.: Data acquisition and view planning for 3-D modeling tasks. In: International Conference on Intelligent Robots and Systems, pp. 417–422 (2007)
4. Charrow, B., Kahn, G., Patil, S., Liu, S.: Information-theoretic planning with trajectory optimization for dense 3D mapping. In: Proceedings of Robotics: Science and Systems (2015)
5. Borrmann, D., Nüchter, A., Đakulović, M., Maurovic, I., Petrović, I., Osmankovic, D., Velagić, J.: A mobile robot based system for fully automated thermal 3D mapping. In: Advanced Engineering Informatics (2014)
6. Biswas, J., Veloso, M.: Depth camera based indoor mobile robot localization and navigation. In: IEEE International Conference on Robotics and Automation (2012)
7. Strand, M., Dillrnann, R.: Using an attributed 2D-grid for next-best-view planning on 3D environment data for an autonomous robot. In: Proceedings of the IEEE International Conference on Information and Automation (2008)
8. Jun, C., Youn, J., Choi, J., Doh, N.L.: Convex Cut: a realtime pseudo-structure extraction algorithm for 3D point cloud data. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (2015)
9. Thrun, S., Hähnel, D., Montemerlo, M., Ferguson, D., Triebel, R., Burgard, W.: A system for volumetric robotic mapping of underground mines. In: IEEE International Conference on Robotics and Automation (2003)
10. Quintana, B., Prieto, S.A., Adan, A., Bosché, F.: Doors detection with 3D laser. Scanners for autonomous navigation in multi-room scenes. In: IPIN: Indoor Positioning and Indoor Navigation Conference (in press)
11. Besl, P., McKay, N.: A method for registration of 3-D shapes. IEEE Trans. Pattern Anal. Mach. Intell. **14**(2), 239–256 (1992)
12. Stachniss, C., Burgard, W.: Exploring unknown environments with mobile robots using coverage maps. In: Proceedings of the International Conference on Artificial Intelligence, pp. 1127–1132 (2003)
13. Gschwandtner, M., Kwitt, R., Uhl, A., Pree, W.: BlenSor: blender sensor simulation toolbox. In: Bebis, G., et al. (eds.) ISVC 2011. LNCS, vol. 6939, pp. 199–208. Springer, Heidelberg (2011). doi:10.1007/978-3-642-24031-7_20