3D Model Based Map Building

Diego Viejo and Miguel Cazorla

***Abstract*—Several works deal with 3D data in SLAM problem. Data come from a 3D laser sweeping unit or a stereo camera, both providing a huge amount of data. In this paper, we detail an efficient method to extract a 3D model from raw data. Each model includes planar patches that belong to planar surfaces on the real scene. Then, we use these models in an ICP-like method in order to compute the robot movement. Once we know the movement performed by the robot, we can build a 3D environment map. Using ICP with planes is not a trivial task. It needs some adaptation from the original ICP. Some promising results in 6DoF are shown for both indoor and outdoor environment and for different kinds of 3D sensors.**

***Index Terms*—Computer Vision, Mobile Robotics.**

I. Introduction

During the last years, new 3D sensor devices have been developed, and computing capabilities have been improved. These improvements can be used to obtain and process a better robot environment information in the field of mobile robotics [1], [2], [3]. By now, methods for achieving tasks such as localization [4], [5], [6], navigation [7], [8] or automatic map building [9], were restricted to the two dimensional world which could be captured by the robot sensors. Nevertheless, using the new 3D sensors such as stereo cameras or 3D laser range finders it is possible to improve the representation of observed objects in order to use them into applications such as augmented reality, architecture, manufacturing process, etc. Furthermore, this new dimension can be used to improve the methods and behaviors used by a robot in order to accomplish its objectives. In this way, the robots equipped with this new 3D sensors are able to move freely into a 3D space, without being confined to the ground, watching and avoiding 3D shape and volume objects.

In the mobile robot field, one of the most important aspects that has to be considered is the movement performed by the robot between two consecutive poses. This information can be obtained from different sources. The most common solution consists in using the robot internal odometers to estimate its movement. Nevertheless, using odometry cause two main problems. First, odometry information always includes measurement errors, which affects the results. Second, it is possible to work with a robot without odometer sensors, or whose odometry information is quite imprecise. This is a very common situation for robots that are able to perform six degrees of freedom (6DoF) movements into a real 3D environment. For this reason, it is necessary to obtain a movement estimation as accurate as possible. In order to compute this movement estimation, robot surroundings data grabbed by its sensors are used. This kind of solutions for robot movement estimation are known as egomotion or pose registration methods [10], [11], [12].

Another important issue that has to be addressed, if we want to perform real time 6DoF robot pose estimation, consists in reducing input data complexity. For solving this problem, we can obtain 3D models for input data in a pre-processing step. We can found several approaches for photo-realistic reconstruction [13], [14], [15], medicine [16], engineering [17], [18], Computer-Aided Design (CAD) [19], [20], [21]. Most of this approaches cannot be applied on mobile robotics problems directly [22], [23], [24] and a deeper study is necessary.

This paper is mainly divided into two parts. The first one, addressed in section III, is focused on the construction of a 3D data model built from the scene that has been captured by the robot. After the study of several modeling approaches, we propose two solutions. On one hand, we describe a method for extracting the main planar surfaces of a 3D scene. This model is complete and accurate, and can be used in some applications such as tele-presence, virtual reality, architectonic reconstruction, etc. On the other hand, we propose a method that estimates planar surface patches into the scene in a quicker way. This method is not so accurate as the previous one, but can be used in time restricted mobile robotic applications, for example, to estimate the robot movement at the same time as it is realized. This first part is completed with a study for adding new features to the resulting 3D model. In this way, we present how to partially use the process performed for obtaining planar patches in order to obtain creases into the scene. Furthermore, plane extraction procedure error has been studied.

The second part of this paper, that is held in section IV, is focused on robot movement estimation problem. In order to obtain this movement, we use its 3D environment information. Instead of using raw 3D data captured by robot sensors, we use the planar based 3D model computed with our quicker modeling method. Our proposal for robot movement estimation is a modification of the *Iterative Closest Points* (ICP) algorithm, but instead of points, we use planar patches for registration. One of the most important features of our proposal is its robustness in the presence of outliers. The correct work of our proposal is demonstrated by means of a huge number of experiments, performed both in indoors and outdoors, and using different kind of 3D sensors. The results obtained, that are shown in section V, bring us to the conclusion that our method can be used into dynamic semi-structured environments.

In section II we will see an overview on physical systems used for obtaining 3D data. Finally, we present all the conclusions obtained during this paper and the future work in section VI. The later is focused on obtaining a more complete

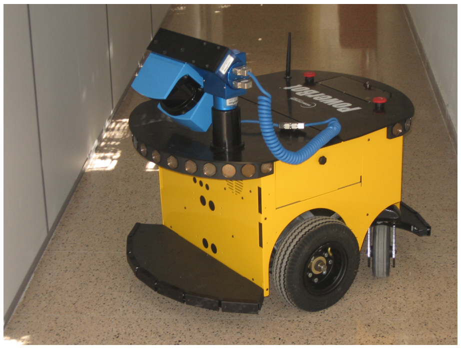
Diego Viejo is with University of Alicante.
E-mail: dviejo@dccia.ua.es

Miguel Cazorla is with University of Alicante.
E-mail: miguel@dccia.ua.es

3D model that could be used for estimating robot movements. We also plan to include robot movement estimation into a global error rectification algorithm.

II. Data Acquisition

The raw 3D data used in our experiments can be acquired with any kind of 3D sensor. Nevertheless, for the results showed in this paper, we have used a stereo camera, the Digiclops model from Point Grey, and a 3D laser range finder, composed by a standard 2D SICK laser with a laser sweeping unit. This sensors allows to capture 3D data, the scanned area being $180^o(h) \times 140^o(v)$ with both horizontal and vertical resolution of 0.5^o. Thus a 3D scene has 360×280 3D points and takes about 2 minutes to be scanned. Figure 2 shows an example of data obtained in both indoor and outdoor environment using our 3D laser system. Each data set is a 3D point cloud. In Figure 3 we can observe a typical 3D scene grabbed with our Digiclops stereo camera. The differences between our 3D sensors are relatives to the accuracy of grabbed data and the maximum distance at which objects can be observed.



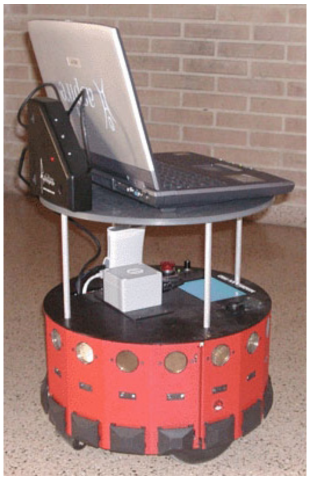


Fig. 1. Our mobile robots equipped with 3D sensors. Top image, Powerbot robot with laser Sick and a sweeping unit. Bottom, Magellan-Pro using a Digiclops stereo camera

These 3D sensors are usually mounted on a PowerBot, from Mobile Robotics, or a Magellan-pro, from iRobot (see Figure 1). 3D laser range finder is mounted on PowerBot for outdoors experiments, whereas Digiclops stereo camera is mounted on the other robot for just indoor experiments. While the robot is driven around its environment, it takes 3D observations from different poses. The robot movement is supposed to be 6 degrees of freedom (DoF), it can roll, pitch, yaw and translate along X, Y and Z axis.

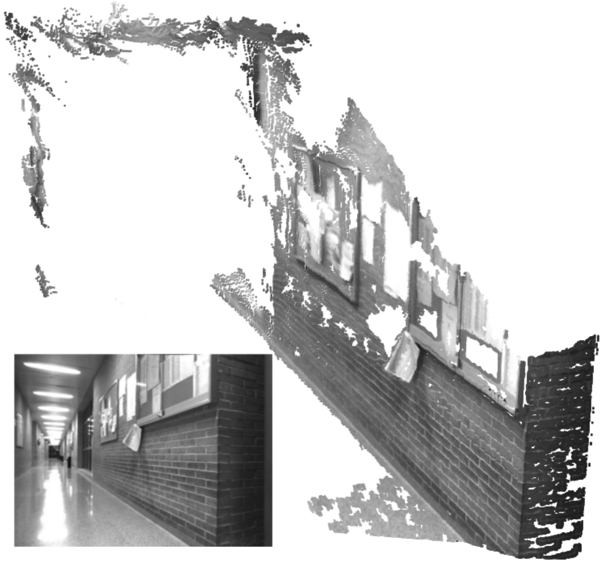


Fig. 3. Example of 3D data captured with our Digiclops stereo camera. Bottom-left square shows a picture of the real scene.

III. 3D modelling

In this section we present our approach on modeling 3D scenes captured by a mobile robot. The objective here consist in reducing the amount of input data that is usually recovered by 3D sensors. In spite of handling raw 3D points data, we'll perform a study of planar surfaces contained into the 3D scene. Planar surfaces can be used to register the movement done by a robot between two consecutive poses for both indoor and semi-structured outdoor environments like urban landscapes. Usually, building facades are formed by planar surfaces and we can exploit this feature to perform pose registration. The set of 3D points from an individual scene retrieved by our 3D sweeping unit has about 65,000 points for outdoor scenes up to 100,000 for indoor environments. In this section we propose an approach to reduce scene complexity to a few number of planar patches (less than 200 per scene).

We use the method proposed by Martin, Gomez and Zalama [25] to estimate local surface directions. Principal Component Analysis is performed over a point p_i and its neighborhood by mean of a Singular Value Decomposition. This process retrieves the underlying surface normal vector of a given set of points if these fit a planar surface. Furthermore, a threshold called *thickness angle* can be defined from singular values in order to determine in which situations a point, as well as its neighborhood, belong to a planar surface or not. This *thickness angle* can be used to measure the fitting of a 3D point set to a plane. The lower *thickness angle* we found, the better fitting between points and planar surface is. If this value is low enough, we can assume that all points into a window

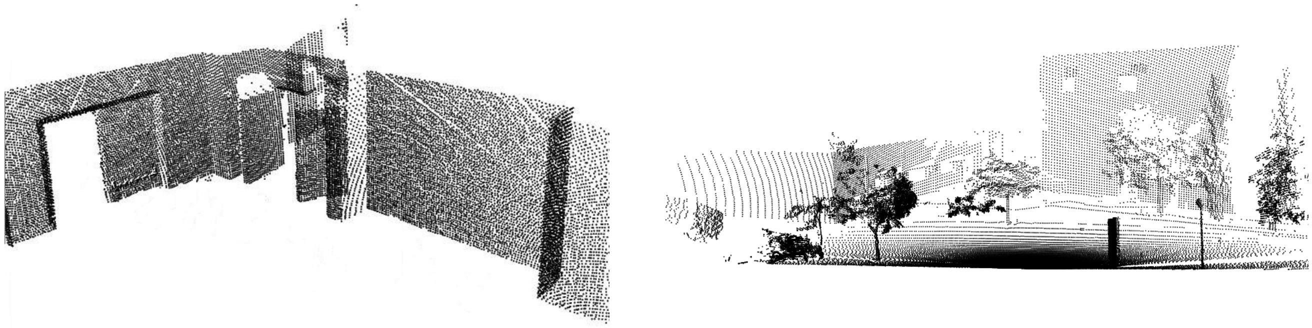


Fig. 2. 3D data from the sweeping unit. Left: indoor environment. Floor and roof points have been removed to improve visualization. Right: outdoor.

belong to a plane, and the estimated normal vector will be the normal vector at each point. Thus, we can easily compute the parameters of the planar patch into the window without computing the normal vector for all the points into the 3D scene as we see below. The size of the planar patches depends on the size of the window used to compute the Singular Value Decomposition.

As it is considered in [26], sample density of 3D laser range finder data presents large variations due to the divergence of consecutively sampled beams. If we take constant size neighborhood we can not obtain planar patches at a certain distance where points are too far from each other. So we propose a dynamic size window depending on the distance between a point in the 3D scene and the coordinate origin (viewpoint). Also, we have to ensure a minimum number of points inside the window in order to make the Singular Value Decomposition (SVD) result reliable. In [25] nine is the minimum number of points used for computing the SVD. We know that the angle between two consecutive sample beams is half degree, so we can compute easily the size of the window, depending on the distance of the point, that ensures at least nine points inside the window.

With this approach we can estimate normal vector direction of the planar patches, but we can not know the orientation of these vectors. Nevertheless, we are trying to find planar patches just in single 3D scene so we have viewpoint information. The correct orientation of a normal vector $\hat{n}_i$ can be set trivially. Given a viewpoint O_j we have to flip $\hat{n}_i$ if $(\hat{n}_i \frac{(v_i - p_i)}{|(v_i - p_i)|}) < 0$. We have no troubles with sharp edges because *thickness angle* avoids normal vector estimation near sharp edges.

Finally, we use an automatic seeded selection algorithm [27] [28] to perform an efficient planar patch extraction from the whole 3D scene. The main criterion to select a point and its estimated normal to be a planar patch will be its *thickness angle* since a point with a low thickness angle is appropriately representative of the points into its neighborhood. We select and remove points from the 3D scene randomly. A selected point (with its normal vector estimated) is stored in the final seeds vector if this point is not inside the neighborhood of any other selected seed and if its *thickness angle* is low enough to ensure reliability. At the end of this process, we obtain the planar patches of a 3D scene. We can see some results both from outdoor and indoor scenes in figure 4. Patch boundaries are computed from the size of the window used to estimate normal vector. We obtain from 900 patches for indoor to 500 patches for outdoor scenes in the resulting 3D models, therefore we achieve a substantial model complexity reduction.

IV. 3D POSE REGISTRATION

The following method is a modification of the classical Iterative Closest Point (ICP) algorithm [29]. ICP is widely used for geometric alignment of a pair of three-dimensional points sets. In mobile robotics, these points sets usually come from range sensors of a robot in two consecutive poses. It is assumed that the rigid transformation aligning the two point sets fits with the movement performed by the robot. Nevertheless, ICP not always reaches a global minimum in the presence of outliers. In fact, the movement of the robot itself produces outliers, that means some regions in the first pose can not be seen in the second and vice versa. The method we propose here resolves this problem and improves robot pose registration in presence of outliers as we explain below.

Furthermore, the high amount of data retrieved by 3D laser makes computations harder. In the literature there are a lot of variants of the ICP algorithm that improve its efficiency. Nevertheless, both original algorithm and its variants try to match two point sets. In fact, in [29] Besl and McKay specify that in order to register any two 3D shapes, a points set of each shape is needed. In this paper we propose a new approach based on ICP algorithm that registers directly two sets of planar patches instead of using two sets of points. The advantage of using planar patches is that we know not only the geometric position of each pacth but also its orientation given by its normal vector. Thus, performing registration with planes allows us to reduce the complexity of the 3D scenes as we have seen in the previous section.

Summarizing, let P^M and P^S be two points sets that represent the model and the scene that we want to register, the original ICP algorithm performs the matching between these

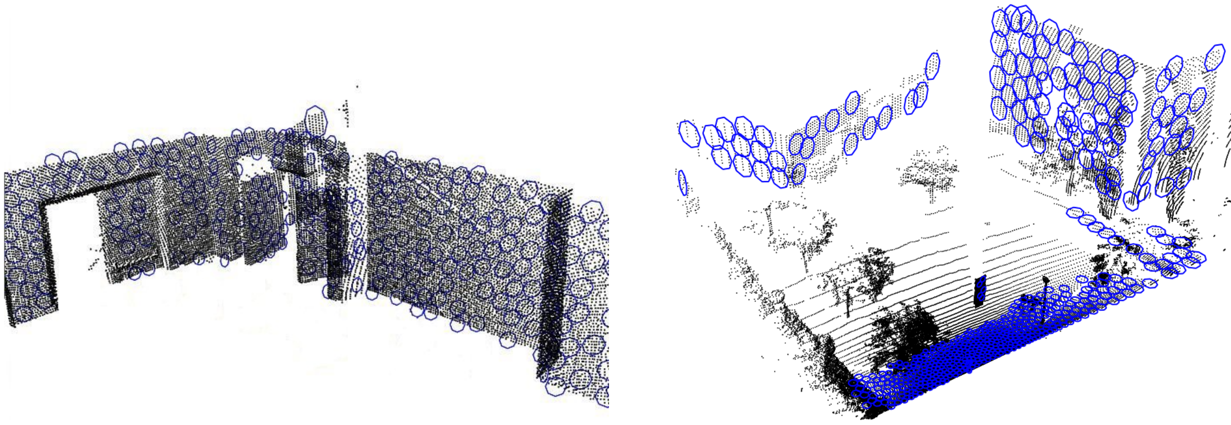


Fig. 4. Planar patches extracted from 3D laser range finder data. Patches are represented by blue circles. Radius of each circle depends on the size of the window used to compute the patch. Top: indoor environment. Bottom: outdoor environment

points sets in two steps. The algorithm begins with an initial transformation. This transformation is applied to the scene P^S and each point of this set is matched with the closest point in P^M. Closest points are selected using Euclidean distance. Next, registration is computed calculating the transformation that fits better the matches calculated in the previous step. This transformation is again applied to the initial point set. These two steps (find closest points and transformation calculation) are calculated iteratively until convergence is achieved. Some approaches use a modified version of ICP, calculating all the distances from one point to the entire second set and using the distance to weigh each match. This produces a faster and better registration.

In our case, we are going to exploit both the information given by the normal vector of the planar patches and its geometric position. Whereas original ICP computes both orientation and position at each iteration of the algorithm, we can take an advantage on the knowledge about planar patches orientation for decoupling the computation of rotation and translation. So we first register the orientation of planar patch sets and when the two planar patches sets are aligned we address the translation registration.

First of all, we define a new function for computing the distance between two planar patches. This planar-distance measurement is used for computing the closest patches from P^S to P^M. For us, P^M and P^S represent the planar patches set computed from the model and scene points set and p_i^m and p_j^s are patches in each set respectively. Let $d_e(p_i^m, p_j^s)$ be the Euclidean distance between the center of patches and $d_a(p_i^m, p_j^s)$ be the angle formed by normal vectors from each patch, therefore we define the distance between two patches

$$D(p_i^m, p_j^s) = \gamma d_e(p_i^m, p_j^s) + d_a(p_i^m, p_j^s) \tag{1}$$

where γ is used for transforming values from Euclidean distance scale to vector angle scale. For rotation computation, lower values of γ are better as we want not only to reduce the scale of Euclidean distance but also to increase the influence of normal vectors distance. Usually, values between 0.15 and 0.015 work correctly for fitting rotation. These values were found empirically. On the other hand, for translation computation of two set of aligned patches, the normal vectors distance is not so determinant and γ values close to 1.0 are more robust.

As we mentioned above, we use an ICP similar scheme in our algorithm. Since we don't use robot odometry information, the initial transformation is represented by the identity matrix. The first step consists in computing the corresponding closest patch from each $p_j^s \in P^S$ to P^M using distance from equation 1. After that, transformation is computed using the closest patches set. As we mentioned above, we compute separately rotation and translation so that two different methods are needed. Below, we describe both.

Another contribution to this paper is the avoidance of outliers into the results. During the closest finding step, we compute the mean μ and the standard deviation σ for the distances between corresponding patches into the closest patches set. Then, μ, σ and corresponding patches distance are used into the transformation computation step for representing the match reliability for each pair of patches. Let $d_{i,j}$ be the distance from the matched patches p_i^m to p_j^s, the reliability for this match is given by

$$w_{i,j} = e^{-\frac{(d_{i,j}-\mu)^2}{\sigma^2}} \tag{2}$$

With this method we assign more weight to those matches that are close to the mean, while those further have less reliability value and so its influence into the final transformation is reduced. We have test the accuracy of our method and the accuracy of the ICP implementation proposed in [30]. In figure 5 we compare the results on alignment error between original ICP and our proposal, depending on the amount of outliers. Results are divided into rotational and translational error. As we can observe, our method is quite robust in the presence of outliers.

The classic ICP mean squared minimization method for aligning the two input data set is modified in order to address planar patches input data. In this way, we compute rotation and translation separately. We use a complete ICP algorithm

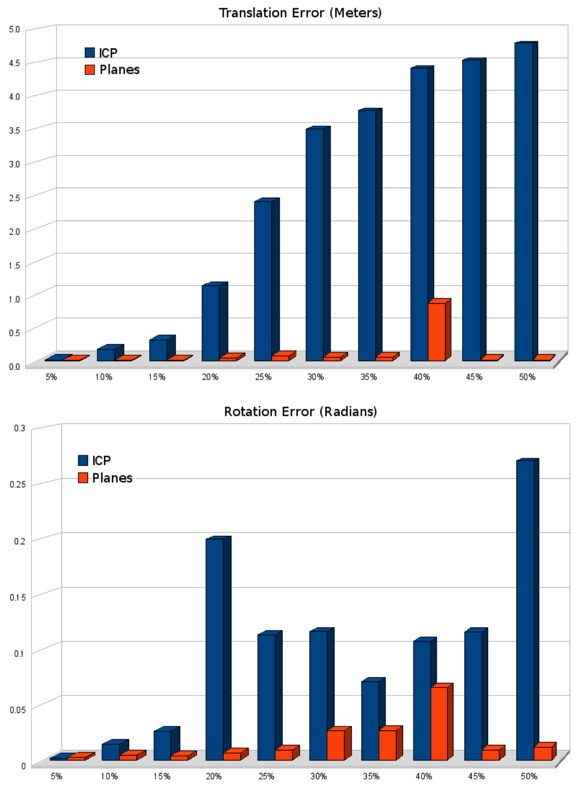


Fig. 5. Comparison chart on alignment error depending on the percentage of outliers. Top, rotation error in radians. Bottom, translation error expresed in meters.

scheme for computing both rotation and translation. Rotation is minimized using a Singular Value Decomposition (SVD) method. For translation, a spring force-based method is used. The resulting algorithm can be found in Figure 6.

```
function Plane_Based_Pose_Registration ($M$, $S$: 3DPlanar_Patch_set)
    $R = \mathbf{I}_3$
    Repeat
        $S^R$ = ApplyrRotation($S$, $R$)
        $P_t$ = findClosest($M$, $S^R$)
        $W_t$ = WeighMatches($P_t$)
        $R$ = MinimizeRotation($M$, $S$, $P_t$, $W_t$)
    Until convergence
    $S^R$ = ApplyrRotation($S$, $R$)
    $T$ = $[0, 0, 0]^t$
    Repeat
        $S^T$ = ApplyTranslation($S^R$, $T$)
        $P_t$ = findClosest($M$, $S^T$)
        $W_t$ = WeighMatches($P_t$)
        $T$ = MinimizeTranslation($M$, $S$, $P_t$, $W_t$)
    Until convergence
    Return [R | T]
```

Fig. 6. Planar patches based ICP algorithm for aligning two 3D scenes.

An example on our planar patch based ICP working process can be observed in Figure 7. The top-left corner image shows the initial point sets. The top-right image represents the planar patches extracted for each 3D point set. Next, images show different steps for the alignment of the two sets. Finally, image at bottom-right shows the result of this process.

V. Results

Several experiments have been performed in order to demonstrate the correct function of our hypothesis. All the data used for the experiments were taken from real scenarios using the 3D scanner described in Section II. We perform our experiments in both indoors and outdoors at the campus of the University of Alicante. Outdoor scenes are formed by a semi-structured scenario composed of isolated buildings surrounded by open green areas and trees. The robot used for experiments performs 6DoF movements along this scenario. 3D images were taken at irregular intervals from 0.5 up to 2.0 meters and 0 to $\pi/4$ radians. During the experiments, people were walking freely around the robot, which introduces noise into data. The approach described in Section III is used to extract planar patches for each set of points. After that, the resulting planar patches set (the model P^M)is used to register robot movement with the planar patches set (the scene P^S) from the next 3D scene taken by the robot. No odometry is used in all the process. Algorithms are written using Java language and run on a Dual Core at 3Ghz. The time needed to compute the planar patches of a 3D scene and to register it with the previous one is less than 15 seconds.

The first set of experiments shows complete environment reconstructions using our 6DOF egomotion described in Section IV for aligning all images into a common reference system. Ground truth is not provided for these experiments, so the results have to be studied visually. Figure 8 shows a zenithal view of the reconstruction obtained from our first outdoor experiment. Floor belonging points have been removed for a better visualization. Straight lines show walls of buildings, whereas circle-like objects are trees. Red line shows the robot path computed with our approach. In this experiment, the robot performs a path that is almost a cycle, i.e. it is not completely closed, of about 40 meters, and takes 30 3D scenes.

The results of another path reconstruction can be observed in Figure 9. This is the longest path the robot performed in these experiments. Complete reconstruction is formed by 116 3D scenes along a more than 100 meters cycled path. Not only man-made planar surfaces are found into the scenario but also a big semi-circular structure can be seen in the middle of the environment. The alignment of the final reconstruction appears to be quite good, about 30 cm. of total error, for such a long path registered just using local 6DoF egomotion and without global rectification.

Figure 10 is a 3D free view of the resulting map from the previous experiment. Here we can observe some details such as buildings facade, windows, trees, streetlights, etc.

VI. Conclusion

In this paper we have presented several methods for robot movement estimation. Firstly, a method to extract a 3D model from raw 3D data is shown. This process is efficient and allows us to reduce the size of the problem. Secondly, we propose a new method which uses these 3D models to obtain the movement performed by the robot with 6DoF. Then, these

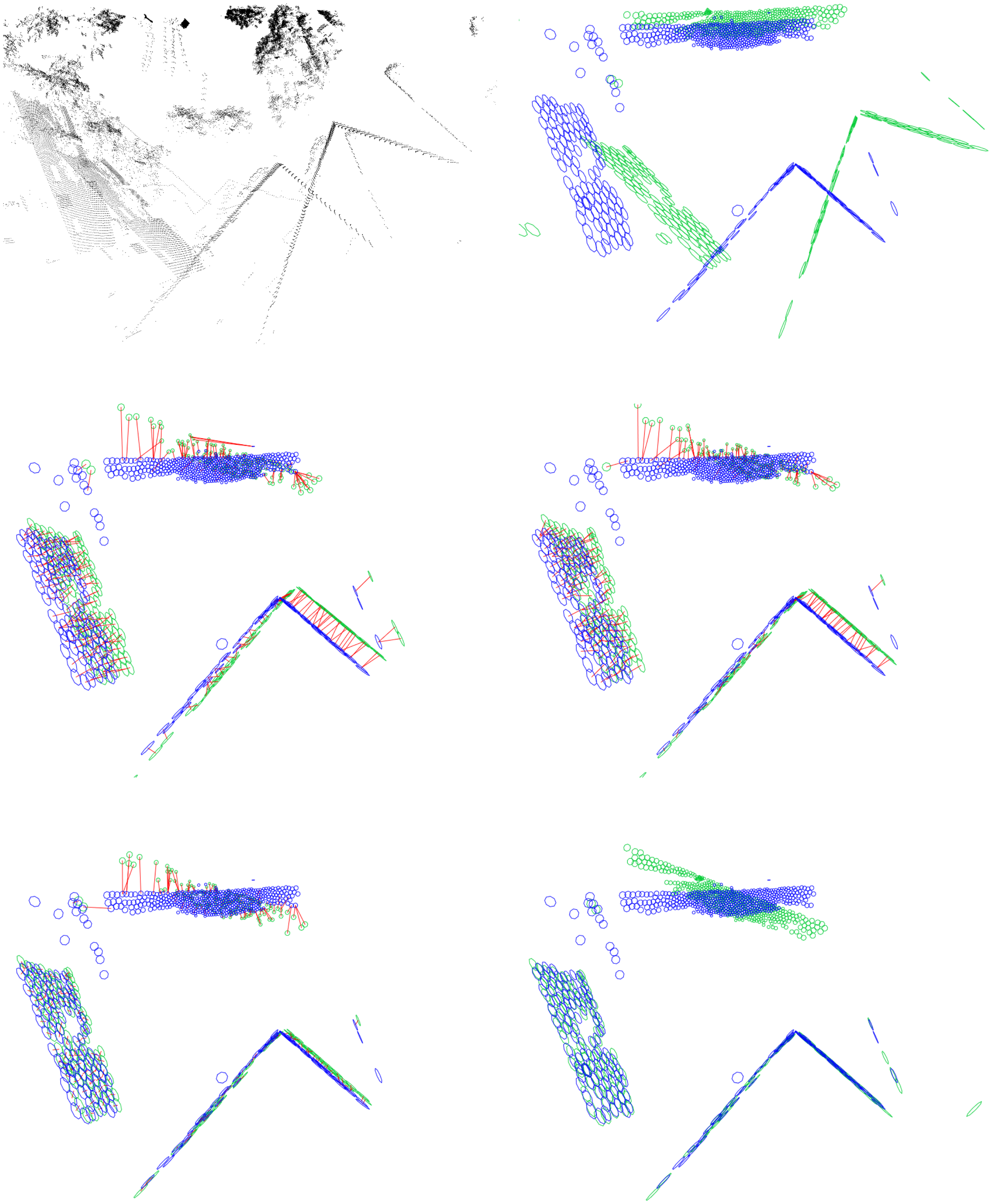


Fig. 7. Planar patches matching example. Top-left, initial raw 3D point sets situation. Top-right, planar patches sets extracted from raw points. For all the the images representing planar patches, those from the model are painted in blue whereas scene patches are represented in green. Next images shows alignment progress. Bottom-right, final alignment result.

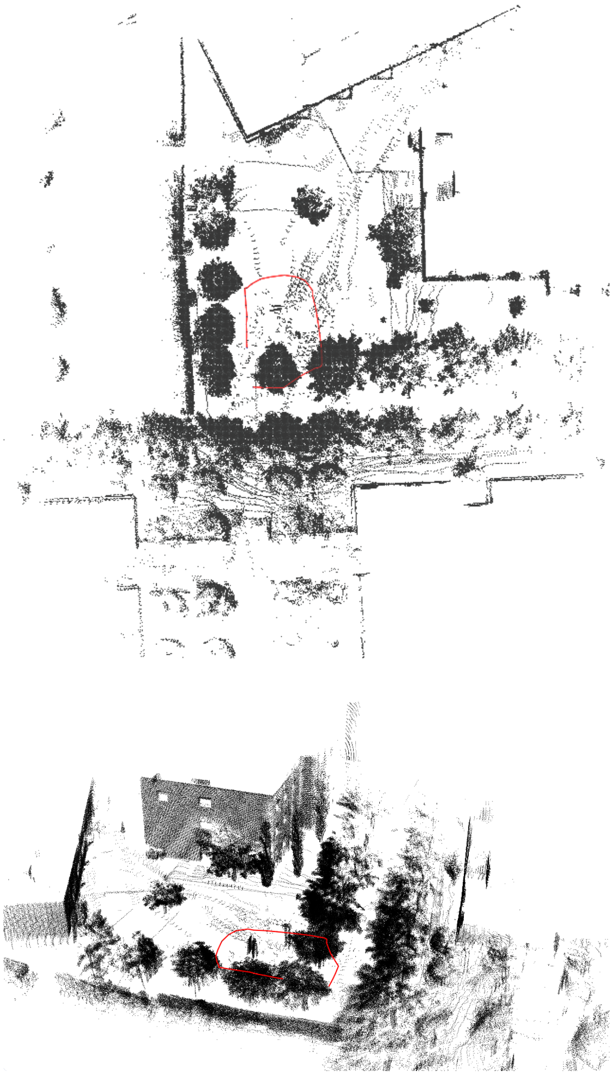


Fig. 8. First experiment results. Top image shows zenithal view of the complete map computed. Robot movement was obtained, from an outdoor semi-structured environment, using 6DoF planar patches based registration. Red line shows the registered trajectory obtained. Bottom, free 3D view of the first experiment computed map. We can observe some 3D objects like building walls, windows, trees, etc.

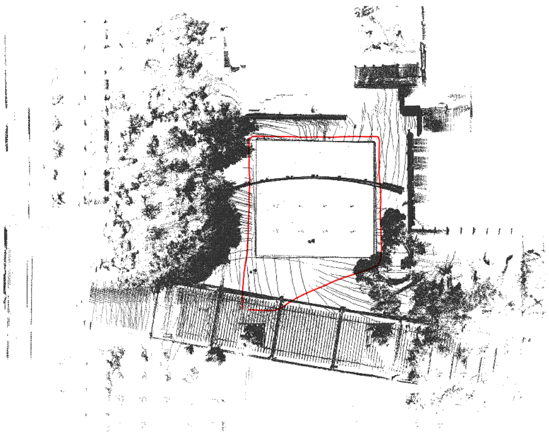


Fig. 9. Zenithal view of the first experiment. Robot movement registration in outdoor semi-structured environment using 6DoF planar patches based registration. Red line shows the registered trajectory obtained.

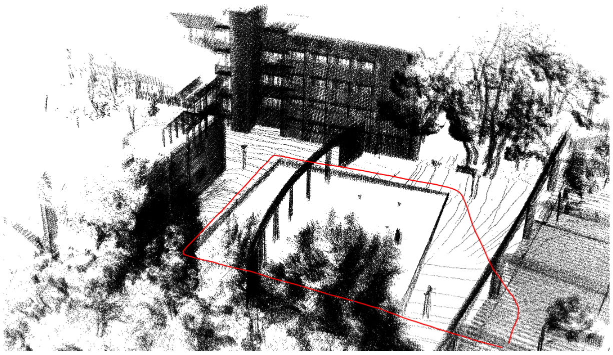


Fig. 10. A free view of the 3D map reconstruction performed for our longest experiment. 3D environment objects can be observed in detail.

patches are used in a modified ICP algorithm, which provides a method to pose registration. This ICP-based method works well in the presence of outliers, which is the main point the original ICP algorithm lacks. Several experiments show the validity of the method for both indoors and outdoors, compared with ICP.

Since people were freely moving during experiments, the proposed methods have been demonstrated to be valid for dynamic environments. This is because our approach uses planar scene surfaces for achieving pose registration. These surfaces usually remain invariant in time, apart from, for example, doors which can be opened or closed at different time and thus are considered outliers. Furthermore, a weight for each matching (in the correspondence process) is used in order to avoid outliers.

As future work, we are planning to include other geometric primitives, apart from planar patches, into the data models used for pose registration. We also plan to include our approach on robot 6DoF movement estimation into a 3D SLAM algorithm. This would be useful for those systems that can not trust on the odometry information.

Acknowledgment

This work has been supported by project DPI2005-01280 from Ministerio de Educación y Ciencia (Spain)...

References

[1] J. Weingarten, G. Gruener, and R. Siegwart, “A Fast and Robust 3D Feature Extraction Algorithm for Structured Environment Reconstruction,” in *None*, 2003.

[2] D. Schroter and M. Beetz, “Acquiring models of rectangular 3d objects for robot maps,” in *Proc. IEEE International Conference on Robotics and Automation ICRA ’04*, vol. 4, pp. 3759–3764, Apr 26–May 1, 2004.

[3] C.-T. Kuo and S.-C. Cheng, “3d model retrieval using principal plane analysis and dynamic programming,” *Pattern Recogn.*, vol. 40, no. 2, pp. 742–755, 2007.

[4] W. Burgard, D. Fox, and S. Thrun, “Active mobile robot localization,” Tech. Rep. IAI-TR-97-3, 25, 1997.

[5] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, “Monte carlo localization for mobile robots,” in *IEEE International Conference on Robotics and Automation (ICRA99)*, May 1999.

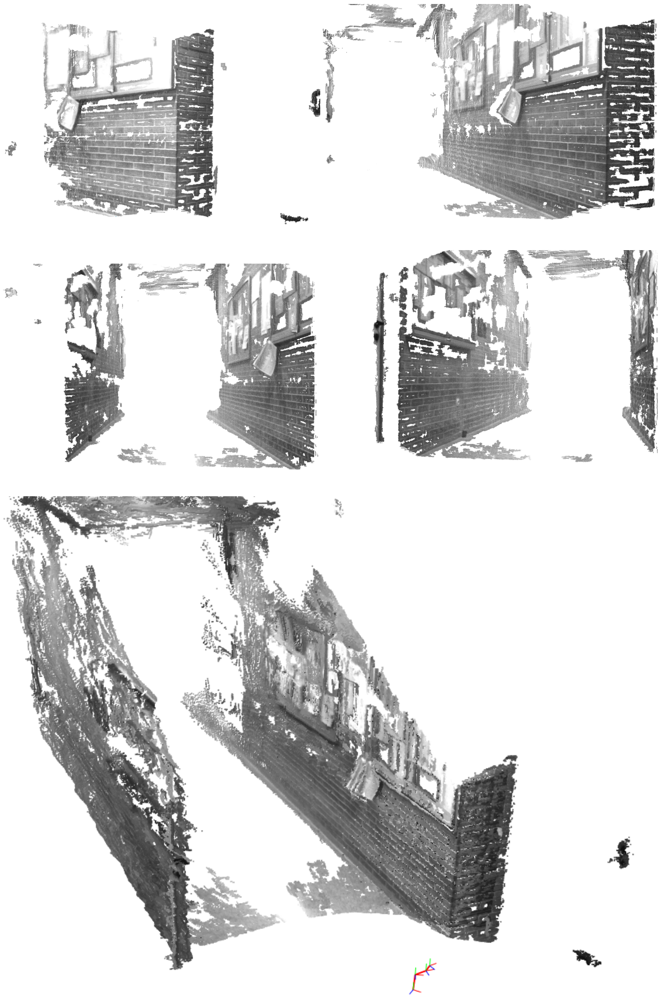


Fig. 11. Zenithal view of the first experiment. Robot movement registration in outdoor semi-structured environment using 6DoF planar patches based registration. Red line shows the registered trajectory obtained.

[6] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust monte carlo localization for mobile robots," *Artificial Intelligence*, vol. 128, no. 1-2, pp. 99–141, 2001.

[7] A. Tsalatsanis, K. Valavanis, and N. Tsourveloudis, "Mobile robot navigation using sonar and range measurements from uncalibrated cameras," in *Proc. 14th Mediterranean Conference on Control and Automation MED '06*, pp. 1–7, June 2006.

[8] G. L. Mariottini and D. Prattichizzo, "Image-based visual servoing with central catadioptric cameras," *Int. J. Rob. Res.*, vol. 27, no. 1, pp. 41–56, 2008.

[9] S. Thrun, D. Hahnel, D. Ferguson, M. Montemerlo, R. Triebel, W. Burgard, C. Baker, Z. Omohundro, S. Thayer, and W. Whittaker, "A system for volumetric robotic mapping of abandoned mines," in *Proc. IEEE International Conference on Robotics and Automation ICRA '03*, vol. 3, pp. 4270–4275, 14–19 Sept. 2003.

[10] M. Agrawal, "A lie algebraic approach for consistent pose registration for general euclidean motion," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1891–1897, Oct. 2006.

[11] O. Koch and S. Teller, "Wide-area egomotion estimation from known 3d structure," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition CVPR '07*, pp. 1–8, 17–22 June 2007.

[12] R. Goecke, A. Asthana, N. Pettersson, and L. Petersson, "Visual vehicle egomotion estimation using the fourier-mellin transform," in *Proc. IEEE Intelligent Vehicles Symposium*, pp. 450–455, 13–15 June 2007.

[13] I. Stamos and P. Allen, "Integration of range and image sensing for photo-realistic 3d modeling," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 2, pp. 1435–1440vol.2, 24-28 April 2000.

[14] V. Sequeira and J. Goncalves, "3d reality modelling: photo-realistic 3d models of real world scenes," in *3D Data Processing Visualization and Transmission, 2002. Proceedings. First International Symposium on*, pp. 776–783, 19-21 June 2002.

[15] P. K. Allen, A. Troccoli, B. Smith, S. Murray, I. Stamos, and M. Leordeanu, "New methods for digital modeling of historic sites," *IEEE Comput. Graph. Appl.*, vol. 23, no. 6, pp. 32–41, 2003.

[16] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, vol. 21, (New York, NY, USA), pp. 163–169, ACM Press, July 1987.

[17] D. Eggert, A. Fitzgibbon, and R. Fisher, "Simultaneous registration of multiple range views for use in reverse engineering," in *Proc. 13th International Conference on Pattern Recognition*, vol. 1, pp. 243–247, 25–29 Aug. 1996.

[18] D. L. Page, Y. Sun, A. F. Koschan, J. Paik, and M. A. Abidi, "Normal vector voting: crease detection and curvature estimation on large, noisy meshes," *Graph. Models*, vol. 64, no. 3/4, pp. 199–229, 2002.

[19] A. Johnson and M. Hebert, "Surface registration by matching oriented points," in *3-D Digital Imaging and Modeling, 1997. Proceedings., International Conference on Recent Advances in* (M. Hebert, ed.), pp. 121–128, 1997.

[20] R. B. Fisher, A. W. Fitzgibbon, and D. Eggert, "Extracting surface patches from complete range descriptions," in *NRC '97: Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, (Washington, DC, USA), p. 148, IEEE Computer Society, 1997.

[21] K. Pulli, "Multiview registration for large data sets," in *Proc. Second International Conference on 3-D Digital Imaging and Modeling*, pp. 160–168, 4–8 Oct. 1999.

[22] S. Thrun, D. Fox, and W. Burgard, "Probabilistic mapping of an environment by a mobile robot," in *Proc. IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1546–1551, 16–20 May 1998.

[23] M. Martin, J. Gomez, and E. Zalama, "Obtaining 3d models of indoor environments with a mobile robot by estimating local surface directions," *Robotics and Autonomous Systems*, vol. 48, pp. 131–143, Sept. 2004.

[24] R. Triebel, W. Burgard, and F. Dellaert, "Using hierarchical em to extract planes from 3d range scans," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2005.

[25] M. Martín, J. Gómez, and E. Zalama, "Obtaining 3d models of indoor environments with a mobile robot by estimating local surface directions," *Robotics and Autonomous Systems*, vol. 48, no. 2-3, pp. 131–143, 2004.

[26] A. R. H. Cole, David M. and P. M. Newman, "Using naturally salient regions for slam with 3d laser data," *Proc. of the IEEE International Conference on Robotics and Automation*, 2005.

[27] F. Y. Shih and S. Cheng, "Automatic seeded region growing for color image segmentation," *Image and Vision Computing*, vol. 23, pp. 877–886, 2005.

[28] J. Fan, M. Zeng, M. Body, and M. S. Hacid, "Seeded region growing: an extensive and comparative study," *Pattern Recognition Letters*, vol. 26, pp. 1139–1156, 2005.

[29] P. Besl and N. McKay, "A method for registration of 3-d shapes," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[30] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in *Proc. Third International Conference on 3-D Digital Imaging and Modeling*, pp. 145–152, 28 May–1 June 2001.