

A robust and fast method for 6DoF motion estimation from generalized 3D data

Diego Viejo · Miguel Cazorla

Received: 3 April 2012 / Accepted: 4 July 2013 / Published online: 23 July 2013
© Springer Science+Business Media New York 2013

Abstract Nowadays, there is an increasing number of robotic applications that need to act in real three-dimensional (3D) scenarios. In this paper we present a new mobile robotics orientated 3D registration method that improves previous Iterative Closest Points based solutions both in speed and accuracy. As an initial step, we perform a low cost computational method to obtain descriptions for 3D scenes planar surfaces. Then, from these descriptions we apply a force system in order to compute accurately and efficiently a six degrees of freedom egomotion. We describe the basis of our approach and demonstrate its validity with several experiments using different kinds of 3D sensors and different 3D real environments.

Keywords 6DoF pose registration · 3D mapping · Mobile robots · Scene modeling

1 Introduction

This paper is focused on studying the movement performed by a mobile robot just using the environment information collected by this robot with a 3D sensor device such as a 3D laser, a stereo or a time-of-flight camera. The trajectory followed by the robot can be reconstructed from the observations at each pose and thus a 3D map of the robot environment can be built. The problem of automatic map building has been challenging mobile robotics researchers during the last years. The

main contribution of this paper is a new method for computing accurately the movement performed by a mobile robot in 6DoF just using the 3D data collected from two consecutive poses, independently of the 3D sensor used.

The actual movement performed by a robot between two consecutive poses for a given action command is one of the main issues that has to be known in order to perform any mobile robotic task. This information can be obtained by several means, depending on the accuracy needed for a given system. The least accurate approach consists in assuming that the robot has performed exactly the movement corresponding to the received action command. In practice, however, this assumption is far away from the real movement.

Measuring the movement of the robot with its internal sensors is a more accurate solution. This information called odometry is the most widely used solution to estimate robot's movement. However, odometry is not error free. Wheel slipping or bouncing, rough terrains or even mechanical wear may produce errors in the odometry data.

Moreover, for many mobile robotic applications odometry is not enough as more accurate movement information is required. In these cases the approach known as egomotion or pose registration can be used (Agrawal 2006; Koch and Teller 2007; Goecke et al. 2007). These methods compute the robot's movement from the differences between two consecutive observations. In this paper we describe a new method that uses three dimensional (3D) information in order to compute the movement performed by a robot. Since our method uses 3D data, it can register a six degrees of freedom (6DoF) movement. We show how this method can be used to build a 3D map. In addition, we designed our approach to work regardless of the hardware used and therefore our method can be applied on different robots and 3D sensors. The input for our method consists in two sets of 3D points from which the 6DoF transformation that best aligns them is obtained.

D. Viejo · M. Cazorla (✉)
Ciencia de la Computación e Inteligencia Artificial, University of Alicante, P.O. Box 99, 03080 Alicante, Spain
e-mail: miguel@dccia.ua.es

D. Viejo
e-mail: dviejo@dccia.ua.es

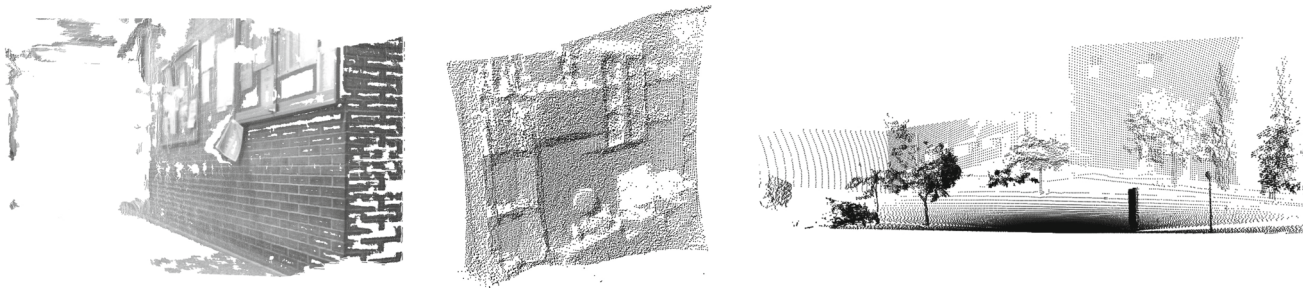


Fig. 1 Input 3D point scenes. The scene on the *left* was captured indoors by a stereo camera. A SR4000 infrared camera was used to capture the image shown in the *middle*. Finally, the *right one* comes from an outdoors environment captured by a 3D range laser

Figure 1 shows a couple of 3D scenes captured by different 3D sensors. The one on the left is captured by a stereo camera, an infrared time-of-flight camera was used to obtain the scene in the middle, while the right one corresponds to a 3D range laser scanner.

The process of computing the robot movement from two consecutive 3D observations can be seen as a particular case of the most general data range registration problem. This problem is faced in many areas such as industrial parts modeling, topography or photogrammetry. The Iterative Closest Point (ICP) methodology is widely used for registering two sets of 3D range data. Usually, these sets are called the *scene* that we are observing and the *model* represented by the previous observation respectively. ICP was first proposed in [Chen and Medioni \(1991\)](#) and [Besl and McKay \(1992\)](#), and has become the most successful method for aligning 3D range data. Summarizing, ICP is an iterative method that is divided in two main steps. In the first one, the relationship between the two sets of input data is obtained by searching for the closest points. This search gives a list of matched points. The second step consists in computing the transformation that best aligns the two sets from the matches found in the previous step. This transformation is used to transform the input sets before the first step of the next iteration. The algorithm iterates until some convergence parameter is reached. The method presented in this paper represents an ICP modification. However, its main contribution is that it improves the time and the accuracy of ICP and makes it suitable for use in a wide variety of mobile robotic applications.

ICP method needs an initial approximate transformation between the two sets before computing its alignment. In mobile robotics, this initial transformation can be obtained from the odometry or by applying some method in order to obtain a coarse alignment ([Brunnstrom and Stoddart 1996](#); [Chung et al. 1998](#); [Chen et al. 1999](#); [Johnson and Hebert 1997](#); [Kim et al. 2003](#)) before applying ICP. Some of these approaches, like [Brunnstrom and Stoddart \(1996\)](#), take too much time to obtain the result in order to be used in a mobile robotic task. On the other hand, other methods, like those based on PCA ([Chung et al. 1998](#); [Kim et al. 2003](#)), can

obtain a result in a short time but with less accuracy. The approach described in this paper does not require an initial coarse transformation in order to compute the 3D registration. This is possible because we use the information of the planar surfaces present in the scenes. The extraction of the descriptions for these surfaces in a 3D scene gives us not only the information about the position of objects in the scene but also information about its orientation. This extra information can be used both to find the correct alignment for two 3D scenes without an initial approximate transformation and to reduce the size of the input data and therefore the time needed to obtain the result.

In the literature there are several proposals for improving the original ICP method both in speed and accuracy. In [Turk and Levoy \(1994\)](#), [Masuda et al. \(1996\)](#), [Weik \(1997\)](#), and [Rusinkiewicz and Levoy \(2001\)](#) different approaches are shown to reduce the size of the input data by selecting just a subset of points for the matching step. However, the amount of information discarded by these techniques is quite important and may affect the accuracy of the results. The addition of some constraints, such as color similarity ([Godin et al. 1994](#); [Pulli 1997](#)) or surface normal vectors ([Pulli 1999](#)), in the matching step can improve the consistency of the matched pairs and thus the final alignment computed by the ICP. The main problem in these methods is that these constraints depend on the kind of sensor used. Once the matches have been established, outliers should be detected in order to improve the results. We use a similar approach like the ones used in [Godin et al. \(1994\)](#), [Pulli \(1999\)](#), and [Rusinkiewicz and Levoy \(2001\)](#) in which each match is weighed by the inverse of the distance between the points involved in the match, but in the present paper, both object position and orientation are used to compute this distance. Other approaches used to reject invalid matches based on some compatibility test ([Turk and Levoy 1994](#); [Masuda et al. 1996](#); [Dorai et al. 1998](#)) also depend on the 3D sensor used. In [Salvi et al. \(2007\)](#) a deeper study comparing different solutions for ICP in terms of speed and accuracy is performed. Nowadays, more ICP variations are still appearing ([Du et al. 2007a,b](#); [Nuchter et al. 2007](#); [Censi 2008](#); [Armesto et al. 2010](#)).

Here we propose a new mobile robotics oriented 3D registration method that improves previous ICP solutions both in speed and accuracy. The successful reconstruction is based in the use of three-dimensional features extracted from 3D scenes. Thus we can reduce the number of iterations required as well as improve the response to outliers and enhance the independence of the initial situation of the data sets. Our work began with [Viejo and Cazorla \(2007\)](#) where we proposed a planar patches based method for pose registration limited to 3DoF. Here we present a generalization of this method for a robot movement in 6DoF. Also, we include a comprehensive testing, both quantitative and qualitative, demonstrating the functionality of the method. Although starting from isolated research, the proposals here are similar to those in [Segal et al. \(2009\)](#).

The rest of the paper is organized as follows. Preliminary computations to obtain both scene modelization and rotation matrix for aligning two sets of planar patches, together with a description of the mapping problem, are covered in Sect. 2. Then, our 3D planar patches alignment method is described in Sect. 3. After that, in Sect. 4 the results for several experiments are shown. Finally in Sect. 5 we will discuss our conclusions.

2 Preliminaries

Our goal is to construct a 3D map from a set of observations $D = \{d_t\}, t = 1 \dots T$ taken from any 3D measurement device (from now, 3D device). The 3D device, placed on the top of a robot or other platform, is moving around the environment (in Fig. 2 the different robots and 3D devices used in this paper are shown).

Our method can process 3D data coming from different devices, each of them having different errors and features. First, we use a stereo camera Digiclops from Point Grey. It has a range of 8 m and is ideal for indoor environments. It can provide up to 24 images per second with gray level information for each point. However, it suffers from the lack of texture: areas in the environment without texture, can not provide 3D data. Moreover, it has a measurement error of 10 %. For outdoor environments we use a 3D sweeping laser unit, a LMS-200 Sick laser mounted on a sweeping unit. It does not suffer from the lack of texture and its range is 80 m with a measurement accuracy of ± 15 mm. The main disadvantage of this unit is the data capturing time: it takes it more than one minute to take a shot. Also, it does not provide color information. In addition, we test a SR4000 camera, which is a time-of-flight camera, based on infrared light. It also does not suffer from the lack of texture, but its range is limited up to 10 m, providing gray level color from the infrared spectrum. Finally, we use a data set ([Borrmann and Nüchter 2012](#)) from a Riegl VZ400 3D range laser. It



Fig. 2 Two robots used in this paper together with different 3D devices: from left to right, a stereo camera, a 3D sweeping laser and a SR4000 camera

has a very fast capturing time (up to 300 measurements per second), a high measurement range (up to 600 m) and a high accuracy (15 mm).

At each time t , an observation, i.e. 3D data, is collected, $d_t = \{d^1, d^2, \dots, d^n\}$, where $d^i = (X, Y, Z)$. However this data can contain more information (gray or color information, etc.). The mapping (or registration) problem can be summarized as:

$$m^* = \arg \max_m P(m|D, U) \quad (1)$$

finding the best map which fits the observations D and the movements of the robot $U = \{u_1, u_2, \dots, u_t\}$. In our case, instead of using the robot movements, we propose the use of observations for both mapping and pose estimation from two consecutive frames.

Thus, we estimate each robot movement u_i using the observations from the previous and posterior poses of that movement. Once the movements have been estimated, a classical method to build an incremental map, using the calculated movements, is applied.

2.1 3D scene modelization

Our 3D registration method exploits geometrical and spatial relationships between scenes surfaces. This extra information can be obtained by performing a model of the input raw data. Furthermore, the amount of information gathered in each raw 3D scene is usually so huge that the time needed to perform pairwise registration can increase dramatically. This model is also useful for reducing the input data complexity without reducing the amount of information in the scene. In a previous work ([Viejo and Cazorla 2008](#)), we described a method with which we can obtain a model for the planar surfaces in a 3D scene using planar patches. Furthermore, this process can be obtained in a logarithmic time.

Summarizing, our method for extracting planar patches from a 3D point set is done in two steps. In the first one, we select a subset of points by means of classical computer vision seed selection methods ([Fan et al. 2005](#); [Shih and Cheng](#)

2005). Seeds are spread along the surfaces of the scene. In the second step, for each 3D point selected, we check whether its neighbourhood corresponds to a planar surface or not. The basic idea consists in analyzing each point in the local neighbourhood by means of a robust estimator. A singular value decomposition (SVD) based estimator is used to obtain surface normal vectors. Using this method, when the underlying surface is a plane, the minimum singular value σ_3 is quite smaller than the other two singular values (σ_1 and σ_2), and the singular vector related to the minimum singular value is the normal vector of the surface at this point. Following singular values we can compute a *thickness* angle (Martin Nevado et al. 2004) that measures the dispersion angle for the points in the planar patch neighbourhood.

$$\gamma = \arctan \left(\frac{2}{\sqrt{3}} \cdot \frac{\sigma_3}{\sqrt{\sigma_1 \sigma_2}} \right) \quad (2)$$

Thickness angle gives us an idea about how 3D points in a neighbourhood fit to a planar surface. We use it to discard patches with a thickness angle greater than 0.05° . In this way we achieve the highest accuracy for our planar patches extraction method. Figure 3 shows the result of computing 3D planar patches model for two different 3D scenes. Planar patches are represented with blue circles. The radius of these circles depends on the size of the neighbourhood used to test *planarity*. This method is proved to work with 3D scenes captured from different kinds of 3D sensors and it can be applied to other applications, like Munyoz-Salinas et al. (2008) or Katz et al. (2010).

3 3D scene planar patch based alignment

The main problem for all ICP solutions is that they need an initial approximate transformation in order to obtain an accurate solution. The method described in this section resolves this problem computing the best alignment for the input data without this initial approximation. This becomes possible

by using the efficient scene model described in the previous section. In this way, the need for an initial transformation is replaced by the 3D scene geometry information extracted during the modeling step. Furthermore, the use of a 3D model makes possible the improvement of the accuracy in the same way as the use of surface tangents proposed in Zhang (1994). For our method this means that, in contrast to the 3D points based ICP solutions, it is not necessary to find exactly the same planar patches in the two scenes in order to match them. It is enough to find planar patches that belong to the same surface. Scene geometry resolves the problem of identifying the different surfaces since it is invariant to changes of the robot pose in 6DoF. In the same way, we use scene geometry in order to detect and remove outliers.

We want to find the 6DoF movement done by the robot between two consecutive poses, from the data taken in those poses, d_i and d_{i+1} . In the general case, we need to find a 6DoF transformation which minimizes:

$$(\mathbf{R}^*, \mathbf{T}^*) = \arg \min_{(\mathbf{R}, \mathbf{T})} = \frac{1}{N_s} \sum_{i=1}^{N_s} \|\mathbf{n}_i^m - (\mathbf{R}\mathbf{n}_i^s + \mathbf{T})\|^2 \quad (3)$$

where \mathbf{n}_i^m and \mathbf{n}_i^s are the matched points from the two consecutive poses.

Instead of calculating rotation and translation iteratively, we propose to decouple them. As planar patches provide useful orientation information, we first calculate the rotation and then, with this rotation applied to the model set, we calculate the translation. In the next subsections we describe how to calculate both.

3.1 Distance between planar patches

Here, we define a new distance function between two planar patches in order to compute the alignment of two sets of 3D planar patches. This new function allows us to search the closest patches between the two input sets. From now we define a planar patch by a point belonging to the patch

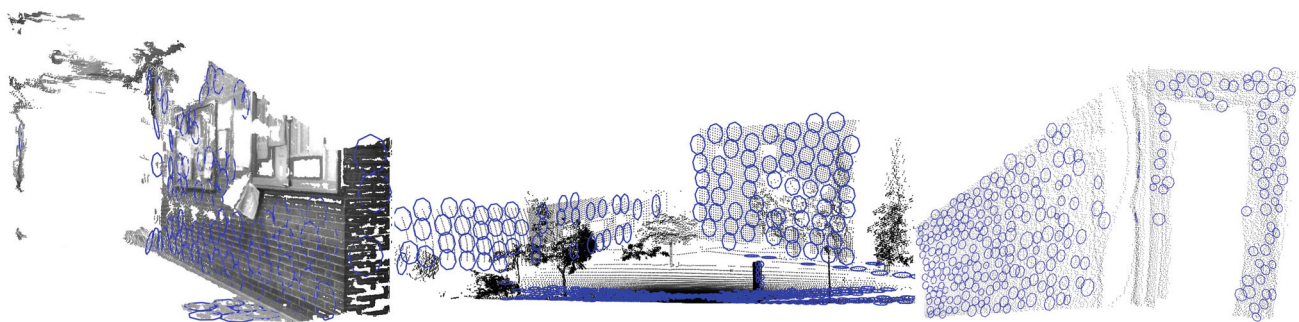


Fig. 3 3D scene model. *Blue circles* represent the 3D planar patches extracted from the input raw scenes. The scene shown on the *left* was captured indoors by a stereo camera. For the *middle one*, a 3D range

laser was used outdoors. Finally, a SR4000 infrared camera was used to capture the one on the *right* (Color figure online)

(usually its center) and its normal vector $\pi = \{\mathbf{p}, \mathbf{n}\}$. Let $d_a(\mathbf{n}_1, \mathbf{n}_2)$ be the angle between two vectors \mathbf{n}_1 and \mathbf{n}_2 . Then the distance from a plane π_i to another π_j can be expressed as

$$d(\pi_i, \pi_j) = d_a(\mathbf{n}_i, \mathbf{n}_j) + \xi d(\mathbf{p}_i, \mathbf{p}_j) \tag{4}$$

where ξ is a factor used to convert euclidean distances between the centers of the patches into the range of normal vector angles.

Once we have a way to compare two planar patches, we can obtain a list of the patches from the scene set closest to the model. Then we have to deal with the outliers. In order to avoid the influence of outliers, we test each match to be consistent with the others. The concept of consistency is based on the kind of transformation we are looking for. Since this is an affine rigid transformation, we can assume that the distance between the patches in a non-outlier pair is similar. In contrast, we set a pair as an outlier when its distance is far away from the mean of distances of all the pair in the matches list. In this way, we compute the weight for the i th pair with the following expression

$$w_i = e^{-(d(\pi_i, \pi_j) - \mu_k)^2 / \sigma_k^2} \tag{5}$$

where μ_k and σ_k are the mean and the standard deviation for the distribution of the distances between matched planar patches in the k th iteration. In this way, we give a weight close to 1 to those matches whose distances are close to the mean and thus, the further the pair distance from the mean is, the lower its weight is set.

3.2 Rotation alignment

From the description of planar patches extracted from two 3D scenes, we can compute the best rotation for the alignment of the two scenes. This can be done by applying a well known method based on the singular value decomposition (SVD). Let us briefly review how to compute this SVD based rotation alignment. Let S be the scene set of N_s planar patches and M be the model set of N_m planar patches. We first find the correspondences between both sets, using the distance in Eq. 4. So each planar patch $\pi_i^s \in S$ is matched with one $\pi_j^m \in M$. Let \mathbf{R} be a 3×3 rotation matrix, then we have to find a matrix \mathbf{R}^* which minimizes the following function:

$$\mathbf{R}^* = \arg \min_{\mathbf{R}} f_a(\mathbf{R}) = \frac{1}{N_s} \sum_{i=1}^{N_s} (d_a(\mathbf{n}_i^m, \mathbf{R}\mathbf{n}_i^s))^2. \tag{6}$$

f_a is the mean square error for the angular distance between matched planar patches. Note that we only use d_a , the angular distance between two planar patches. In order to minimize f_a , we compute the cross-covariance 3×3 matrix

Σ_{sm} for planar patches normal vector with the expression

$$\Sigma_{sm} = \sum_{i=1}^{N_s} w_i \mathbf{n}_i^s (\mathbf{n}_i^m)^t, \tag{7}$$

where w_i is the weight of the i th match (see Eq. 5). This weight is used for outliers rejection. The matrix Σ_{sm} can be decomposed as $\Sigma_{sm} = V \Delta U^T$, where Δ is a diagonal matrix whose elements are the singular values of Σ_{sm} . The rotation matrix that best aligns the two set of planar patches is then computed as

$$\mathbf{R} = V \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \delta \end{pmatrix} U^T \tag{8}$$

where $\delta = 1$ when $\det(VU^T) \geq 0$ and $\delta = -1$ if $\det(VU^T) < 0$. The result of this procedure is a rotation matrix that best aligns the two input planar patches sets. Further information about this method can be found in Kanatani (1993) and Trucco and Verri (1998).

3.3 Translation alignment

Once the rotation has been obtained, we have to compute the transformation that minimizes the quadratic mean error for the alignment of the input planar patches. Original ICP uses a closed-form solution based on the use of quaternions to resolve the least square problem (Eq. 3) that gives the rotation matrix for the solution. From this rotation transformation, translation is simply computed as the vector that joins the center of mass of the two input sets of points. Nevertheless, the final transformation obtained is inaccurate due to outliers. As we mentioned before, a lot of methods to avoid the error produced by outliers have been proposed. These methods rely on a good initialization in order to obtain accurate results. Our contribution here consists in a new method that is less sensitive to the initial approximation of the input data sets by exploiting the extra information given by planar patches in order to obtain more accurate results.

Let us suppose that the rotation needed to align the input sets has already been computed and applied following the method described in the previous section. Then the problem is how to obtain the best estimation for the translation. The solution that we propose consists in considering that the matches list represents a force system that can be used to compute the correct solution. For the i th match, we set the force vector \mathbf{f}_i in the direction of the normal from the model’s patch in the match and with the magnitude of the distance between the planes defined by the patches. The resultant translation for a force system created from N matches is a vector computed by the following expression:

$$\mathbf{T} = \frac{\sum_i^N \mathbf{f}_i}{N} \tag{9}$$

Furthermore, local minima can be avoided more successfully if we consider the main directions of the scene. These can be calculated from the Singular Value Decomposition of the cross-covariance matrix of all the force vectors. Let us call $\mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3$ the three main direction vectors. The Eq. 4 can be rewritten to represent how much each pair of planar patches contributes to a specific main direction

$$d_k(\pi_i, \pi_j) = (d_a(\mathbf{n}_i, \mathbf{n}_j) + \xi d(\mathbf{p}_i, \mathbf{p}_j)) \cdot \cos(d_a(\mathbf{n}_i, \mathbf{D}_k)) \quad (10)$$

At this point, $d_a(\mathbf{n}_i, \mathbf{n}_j)$ should be close to zero as rotation alignment has been solved. Using 10 we can obtain the mean and standard deviation for the distances along each main direction. In this way, Eq. 5 is redefined as follows.

$$w_i^d = e^{-(d(\pi_i, \pi_j) - \mu_{k,d})^2 / \sigma_{k,d}^2} \quad (11)$$

Accordingly to Eqs. 10 and 11, Eq. 9 is then transformed to take into account the main directions.

$$\begin{aligned} \mathbf{t}_1 &= \tau \sum_{i=1}^N (\mathbf{f}_i \cdot \mathbf{D}_1) w_i^1 \\ \mathbf{t}_2 &= \tau \sum_{i=1}^N (\mathbf{f}_i \cdot \mathbf{D}_2) w_i^2 \\ \mathbf{t}_3 &= \tau \sum_{i=1}^N (\mathbf{f}_i \cdot \mathbf{D}_3) w_i^3 \end{aligned} \quad (12)$$

where \mathbf{f}_i is the force vector from the i th match, N is the size of the matches list and τ is a normalizing factor. Then, the transformation that reduces the forces system's energy is

$$\mathbf{T}_j = \mathbf{t}_1 + \mathbf{t}_2 + \mathbf{t}_3 \quad (13)$$

This computation is included in an ICP-like loop, in which matches are recomputed after each iteration, in order to reach a minimum energy state for the system. Using this approach we can reduce the number of iterations and we can achieve registration with greater independence from the initialization. This method achieves the alignment of the two input segment sets with a significantly smaller error in comparison to other methods, like ICP.

3.4 Complete algorithm

The method described in this section can be used for computing the transformation that best aligns two sets of 3D planar patches. The complete method for computing the planar patches based 3D egomotion is divided into two main ICP-like loops. The first one is used to compute the best rotation alignment for the normal vector of the 3D planar patches. Once the normal vectors of the planar patches are aligned, the second loop is used to compute the minimum energy force system that represents the best translation alignment.

The resulting algorithm is shown in Algorithm 1. A complete example on how this method works is shown in Fig. 4. Input data was taken by a robot in a real scenario. In this case, the movement performed was a turn. The sequence goes from left to right and from top to bottom. The scenes are shown from the top, looking towards the floor. The picture in the upper-left corner shows the initial state in which two sets of 3D points have been recovered by a 3D range laser. The next picture shows the planar patches extracted from the two input sets. They are represented in green and blue. After that, different steps of our alignment algorithm are shown in the next pictures in which matches are represented by red segments. The resulting 3D registration for the two set of 3D planar patches is shown in the bottom-right corner picture. Since the input data was taken from a real scenario, there are also outliers. It can be observed in the last picture where a lot of planar patches from one set do not have a corresponding patch in the other set and vice versa.

Algorithm 1 Plane_Based_Pose_Registration (M, S : 3DPlanar_Patch_set)

```

R = I3
repeat
  SR = ApplyRotation(S, R)
  Pt = FindClosests(M, SR)
  Wt = ComputeWeights(Pt)
  R = UpdateRotation(M, S, Pt, Wt)
until convergence
SR = ApplyRotation(S, R)
T = [0, 0, 0]t
repeat
  ST = ApplyTranslation(SR, T)
  Pt = FindClosests(M, ST)
  {D1, D2, D3} = FindMainDirections(Pt)
  Wt = ComputeWeights(Pt, {D1, D2, D3})
  T = UpdateTranslation(M, S, Pt, Wt, {D1, D2, D3})
until convergence
return [R | T]

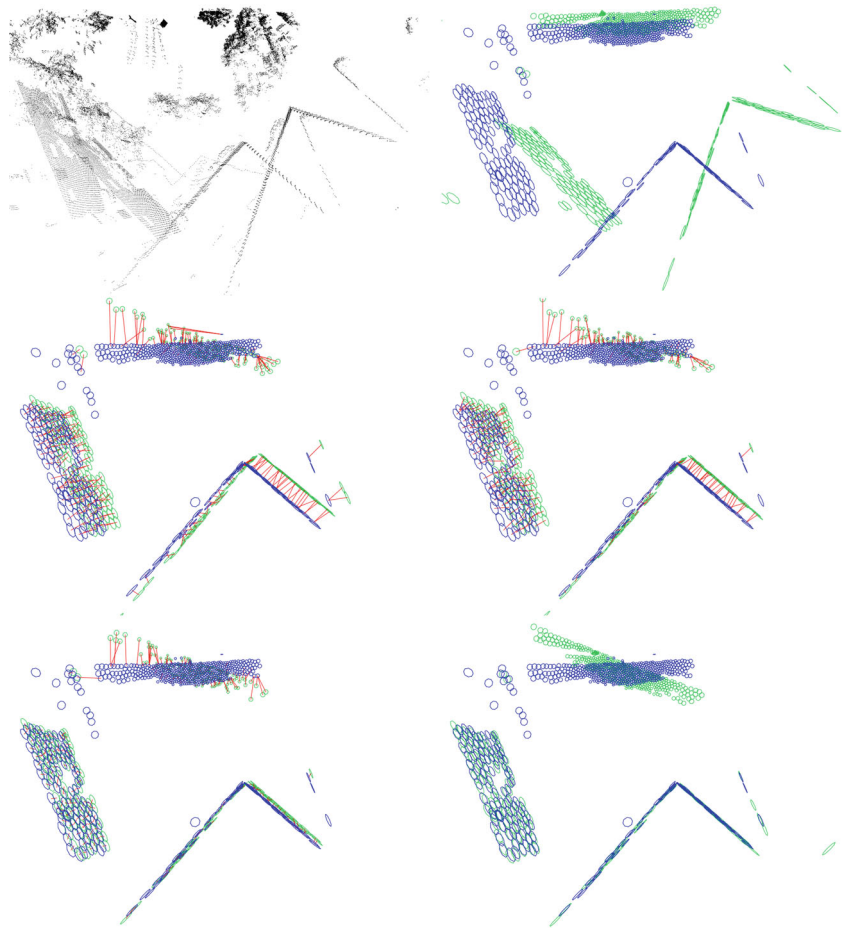
```

4 Results

In this section we show our results from different experiments using the method presented in this paper for aligning 3D planar patch sets. First, we study the advantages we get by using our method in comparison to the ICP algorithm. This ICP algorithm incorporates the outlier rejection approach proposed in Dalley and Flynn (2002). KD-tree and point subsampling techniques are also included for reducing computational time as described in Rusinkiewicz and Levoy (2001). We analyze the performance and the accuracy of the results. After that we show different results on automatic map building using this approach.

The first experiment consists in testing the accuracy of the computed registration. We test both the incidence of the

Fig. 4 Overview of the working method. From the two input 3D point sets (*upper-left corner*) planar patches are extracted (*upper-right corner*). Using the planar patches, the best alignment transformation is computed with our modified ICP-like approach. After a few steps, the resulting transformation is obtained (*bottom-right corner*). The 3D planar patches that do not have a corresponding one in the other set are outliers



outliers present in the scene and the influence of different initial scene alignments. For the first test, we have chosen a 3D scene captured by our robot in a real environment to be the model set for the alignment. The scene set is obtained by applying a transformation to the model. Since this transformation is known, it represents a ground truth for computing the error for the alignment. Outliers are simulated by removing areas from the scene set. This test is performed on a large set of three-dimensional scenes to obtain a robust estimation of the errors. A comparison between the improved ICP and our approach for different outliers rate can be observed in Fig. 5. However, our proposal always obtains better results than ICP. This is because we do not require an initial approximate transformation and because of the amount of outliers. It has been described that ICP does not reach a global minimum for a given outlier rate. In contrast, we can obtain the correct alignment even for an outlier rate of 40–50 %.

For the second test we have used a rotating unit from PowerCube that allows us to rotate a 3D sensor along Y axis at different known orientations. The experiment consists in performing a complete turn of the sensor around itself getting 3D data sets at regular intervals. The experiment is performed several times using different angular rotations at regular

intervals of 4° . The objective of this experiment consists in testing the response of the method for different initializations. Again, we compare our method with the above mentioned ICP version. The results can be observed in Fig. 6. As expected, ICP depends on the initialization and for changes of more than 10° the algorithm always ends in a local minimum. On the other hand, our approach significantly improves this behavior and allows us to obtain accurate results for displacements of up to 40° . Below 45° the symmetries of the environment can not be resolved and make our method end in a local minimum.

We also compare the time needed to compute the transformation with ICP and using our method. Both methods have been implemented using Java and have been executed on the 1.6 version of the Java Virtual Machine running on a Intel T5600 1.83 GHz and 2 Gb of RAM memory. While ICP needs more than 40 s, our algorithm can find the solution in less than 5 s. Here we have to add the time needed to extract the planar patches for each 3D scene. Nevertheless, this is a computational low cost algorithm that usually takes less than one second to modelling a 3D scene.

For the next experiments, our method has been used to perform incremental 3D map building. As the robot moves

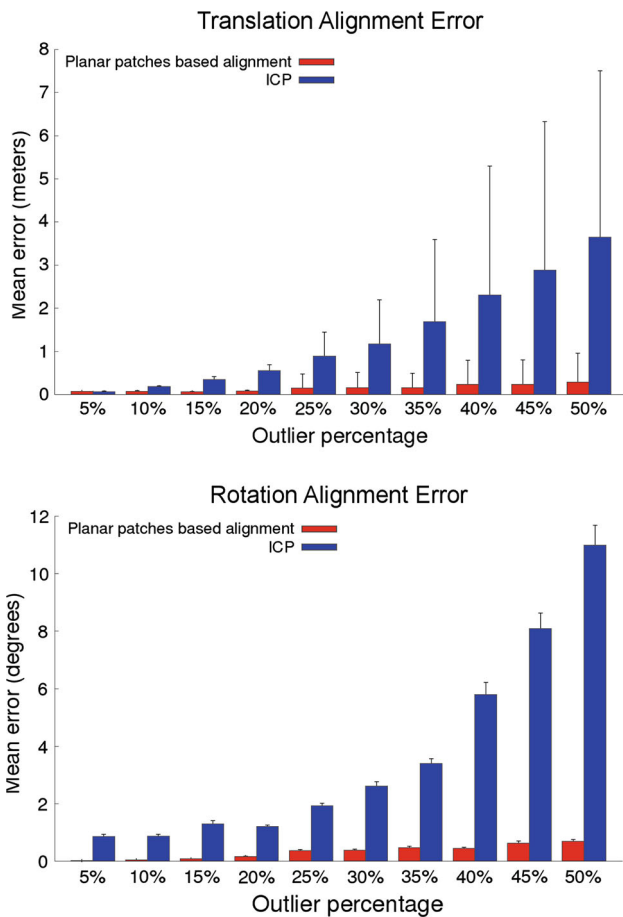


Fig. 5 Mean error and standard deviation for different rates of outliers using our approach and an improved ICP algorithm. Results are separated for translation, in the *top*, and rotation, in the *bottom*

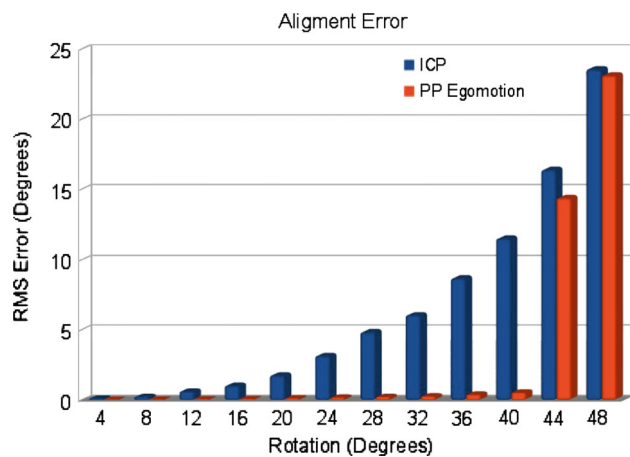


Fig. 6 Alignment error for different initializations. Planar patched based egomotion is less affected by initial displacement of input data sets

collecting 3D data with its sensor, the transformation between each two consecutive poses is obtained. This transformation is used to hold all the captured 3D points in a reference frame.

A global rectification is not performed, so the errors accumulate as the map is being built. The proposed method is so accurate that these errors do not affect the results too much. In order to fuse the 3D points that come from the same object, an occupancy grid has been used. At the time these experiments were performed, we did not have the equipment necessary to obtain a ground truth in order to compute a trajectory error estimation. For this reason, in most of the experiments the trajectory was chosen to be a loop in order to compare the position in the map for re-observed objects.

4.1 Indoors stereo data set

For the first experiment on map building we have used a Digi-clops stereo camera as 3D sensor device. The experiment was performed indoors. It is well known that stereo systems do not work correctly when scene’s surfaces have a flat texture. Moreover, the maximum range for our stereo camera is too short (up to 5–6 m). Our method needs planar surfaces in three orthogonal directions. If this requirement is not fulfilled, our method will fail to compute the robot movement correctly. The lower information is obtained by the sensor, the fewer planar descriptions are found and thus the method accuracy is decreased. The resulting 3D reconstruction for a four consecutive 3D scenes alignment can be observed in Fig. 7. This experiment shows how our method can deal with an inaccurate 3D sensor like a stereo camera.

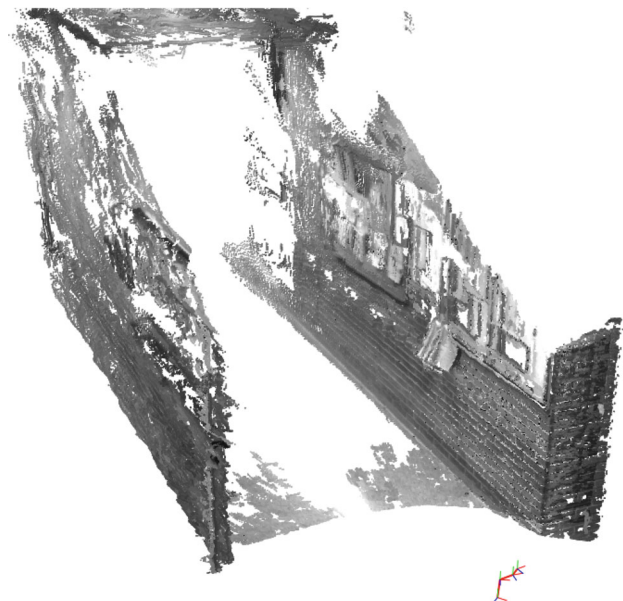


Fig. 7 Indoor map building experiment using a stereo camera: resulting 3D reconstruction



Fig. 8 Indoor map building experiment using a SR4000 camera: zenithal view of the resulting 3D reconstruction. The red line shows the path follow by the robot (Color figure online)

4.2 Indoors SR4000 data set

In this experiment we used a SR4000 infrared camera. It provides an interesting advantage against a stereo camera: it obtains information independently of the scene's surface texture. The maximum range of this camera is 10 m, which is quite enough for an indoor environment. Figure 8 shows a 360 frames sequence in an indoor environment.

4.3 Polytechnic college data set

Our next experiment was performed outdoors in a square between the buildings of the Polytechnic College at the University of Alicante. This time the 3D sensor used was a sweeping range laser. 30 3D scenes were captured during the 45 m long almost-closed loop trajectory. In Fig. 9 the reconstructed map can be observed from a zenithal view. It is superimposed to a real aerial picture from the environment. The points that come from the floor have been removed for a better visualization. Furthermore, it can be noticed that there are many points that were captured in the open space. These points come from people that were moving freely around the robot during the experiment. Our registration method is not



Fig. 9 Zenithal view of the resulting map of the Polytechnic College data set experiment using a 3D laser. Map points are superimposed to a real aerial picture. The red line represents the computed robot trajectory (Color figure online)



Fig. 10 3D map view of the experiment performed in the Polytechnic College. Some objects in the environment such as trees, street-lamps and buildings can be observed

affected by dynamic objects since it uses planar surfaces that remain mostly still. Those planar surfaces that may change its position, like doors, are detected by the outliers rejection step. The computed trajectory for this experiment is represented by a red line. The resulting map appears to be quite accurate as the estimated final pose error is around 15 cm.

Figure 10 shows a free view of the Polytechnic College data set experiment reconstructed 3D map. The computed 3D trajectory is also represented. Building facades, street lamps, trees and other objects can be recognized. Double representation for some objects is produced by accumulated errors.

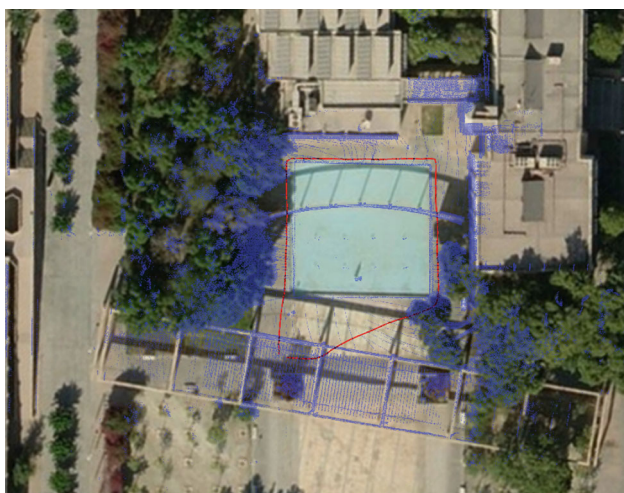


Fig. 11 Reconstructed map from the experiment performed in the Faculty of Science. Data come again from a 3D range laser. The computed trajectory is represented by the *red line* (Color figure online)

4.4 Faculty of science data set

Our next experiment represents the longest trajectory performed by the robot. It corresponds to the surroundings of the Faculty of Science at the University of Alicante. This is a low structured environment with lots of trees, hedges, street-lamps, etc. The reconstructed 3D map can be observed from a zenithal view in Fig. 11. This time, the map is built from 117 3D scenes along a 130 m trajectory. This time the error, it is said, the difference between objects observed at the beginning and at the end of the trajectory is about 30 cm. 3D map from a free view can be observed in Fig. 12 where environment details can be appreciated.

4.5 University of Alicante's Museum data set

Since our previous experiments were performed in environments in which the floor was almost plane, for our next experiment we looked for a place in which the movement performed by the robot was clearly a 6DoF one. For this reason, we chose the entrance of the University of Alicante's Museum. This place is a covered passageway with different slopes. In this place, the movements performed by the robot are more complex since we chose the robot to turn in the middle of the slopes. Figure 13 shows the resulting 3D map for this experiment and gives a clear idea of the trajectory performed by the robot. In the upper picture the zenithal view can be observed while the bottom picture shows a lateral view in which slopes can be observed. The map was built from 71 3D scenes captured by a 3D range laser in a 35 m long trajectory. The difference between the first and the last pose is about 20 cm.

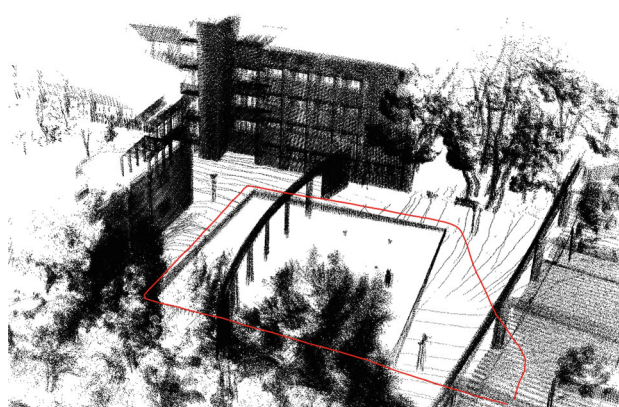


Fig. 12 Free view of the 3D map reconstructed for the Faculty of Science data set experiment. Objects such as buildings, different kind of trees, street-lamps, hedges, etc. can be distinguished

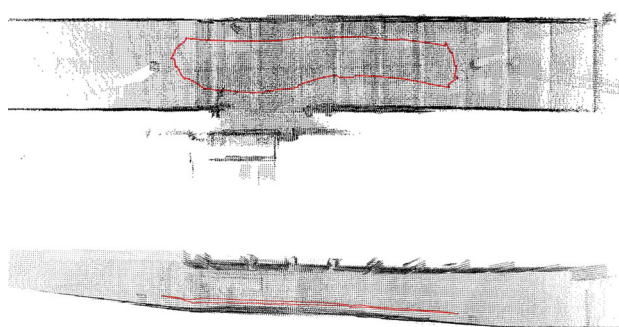


Fig. 13 3D map built from the data obtained in the University of Alicante's Museum. The *upper image* shows a zenithal view in which the trajectory represented by a *red line* can be appreciated. The *bottom image* shows a lateral view for the reconstructed 3D map. The slopes present in the environment can be observed here (Color figure online)

4.6 Bremen data set

For our last experiment we use a data set recorded by Dorit Borrmann and Andreas Nüchter from Jacobs University Bremen gGmbH, Germany, as part of the ThermalMapper project, and it is available to download for free (Borrmann and Nüchter 2012). This data set was recorded using a Riegl VZ-400 and a Optris PI IR camera. It contains several 3D scans taken in the city center of Bremen, Germany. The data set consists of data collected at 11 different poses. The points from the laser scans are attributed with the thermal information from the camera. Approximately, each two consecutive poses are separated by 40 m. For this experiment, a ground truth is available as it includes pose files calculated using 6D SLAM (Andreas Nüchter and Surmann 2007). Comparing the trajectory obtained using the method presented in this paper with the ground truth, we get a root mean square (RMS) error of 0.89 m in translation and 1.271° in rotation. Even though huge movement was performed between poses, no odometry information was needed for most of the poses.

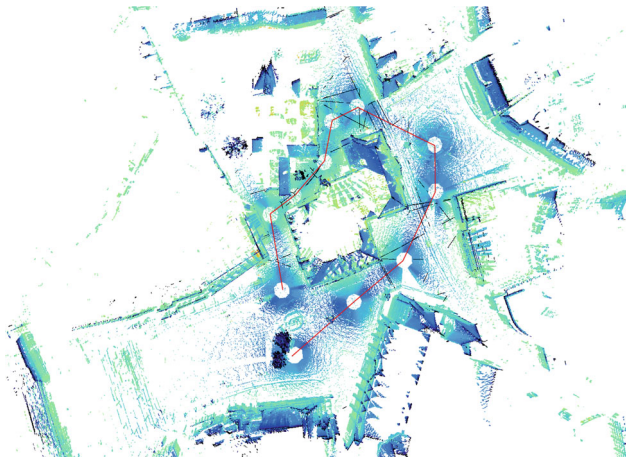


Fig. 14 Reconstructed map from Bremen data set. In this case, data come from a Riegl VZ-400. Thermal information is used to color the points. The computed trajectory is represented by the red line (Color figure online)

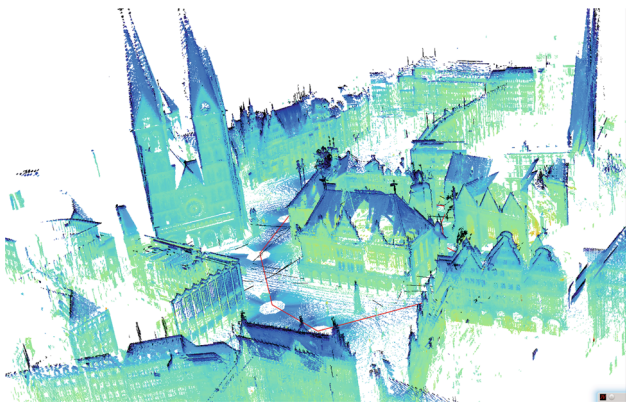


Fig. 15 Free view of the 3D map reconstructed for Bremen data set experiment. The reconstruction of the movement in such a big environment can be achieved using our approach

Just in three of them, whose rotations are bigger than 45° , odometry was used to obtain the correct transformation (Figs. 14, 15).

Finally, in Table 1 we present several statistics recovered during these experiments. The first column shows the number of 3D scenes registered. The mean number of 3D points per scene and the mean number of 3D planar patches are shown in the second and third column. The last two columns show the mean time needed to extract the planar patches and to apply the 3D registration. All our data sets are public and available to download (Viejo and Cazorla 2013).

5 Conclusion

This paper covers the problem of finding the movement performed by a robot using the information collected by its 3D

Table 1 Statistical data recovered during 3D map building experiments

Set	Scenes	Points (mean)	Patches (mean)	Patch (mean)	Registration (mean)
Stereo	4	134,662	54	1.042 s	0.065 s
SR4000	360	19,371	106	0.098 s	0.567 s
P. College	30	67,611	640	0.984 s	2.022 s
F. of Science	117	56,674	657	0.600 s	2.413 s
Museum	71	75,859	941	0.674 s	5.917 s
Bremen	11	7037,852	2,208	1.617 s	54.813 s

For each data set number of scenes, mean number of points per scene, mean number of planar patches extracted per scene, mean time for extracting planar patches and mean time for computing the transformation between two consecutive scenes are shown

sensor. This process is called 3D pose registration and can be used for automatic map building. Our method is designed to obtain accurate results even when the scenes present a big amount of outliers. This makes unnecessary the use of an initial rough transformation between each pair of scenes. Furthermore, since our method is designed to work with 3D data it can obtain the registration of a 6° of freedom movement. This makes it highly suitable for those robotics applications in which odometry information is not available or is not accurate enough such as aerial or underwater robots.

Another important feature of our method is that it works with raw 3D data, it is said, unorganized 3D points sets. In contrast to other previous works on 6D mapping like Borrmann et al. (2008), Kümmerle et al. (2008), Censi (2008), and Armesto et al. (2010), this feature makes our method independent of the 3D sensor used to obtain the data. In this way, applying our method for different robots equipped with different sensors does not require a big programming effort. The first step for our algorithm is the modeling of the input data. This step gives us three advantages. The first one is the reduction of the data size without a significant reduction of the amount of information present in the original scene. This highly decreases the time needed by the algorithm to obtain the results. The second advantage consists in the extraction of important structural relations for the data that are exploited in our algorithm in order to improve the results. Finally, the third advantage we get using planar surfaces description is that it makes easier to obviate dynamic objects, such as people, while computing the registration.

The ICP algorithm is an unstable method that not always reaches an optimal solution. Usually, small changes in the input data make the ICP finish in a local minimum. Although there are many approaches that represent an improvement in the reduction of outliers effects, ICP can still not obtain good results when the amount of outliers is quite representative. For mobile robotics, we consider outliers all the data that can be observed from a pose but not from the previous one

and vice versa. As a robot moves, the maximum range of the robot sensors and the movement itself introduce outliers in the data. The amount of outliers depends on the distance travelled by the robot between two consecutive poses. The better the aligning method for detecting and avoiding outliers is, the bigger distance the robot can move and be registered. In contrast to most of the methods that can be found in the literature for improving the consistency of matched points and for reducing the influence of the outliers, our method does not use the internal data representation from the sensor and in this way our method is independent of the sensor used to collect the data.

The proper functioning of the proposed method has been demonstrated with several experiments. The first experiment was designed to study the influence of outliers in the resulting transformation. As it was demonstrated, our method can handle up to 40 % of outliers in the data, that is quite better than the amount of outliers tolerated by ICP (less than 15 %). This is an important result for mobile robotics since outliers are introduced into the data mainly by the robot movement. In order to appreciate the accuracy of the proposed method, several experiments for 3D automatic map building were carried out. Several scenarios were chosen, both outdoors and indoors. We also test the use of different 3D sensors such as a stereo camera, a time-of-flight camera and two different 3D range lasers. In all the experiments the map was built from the alignment of every two consecutive 3D images without any kind of global rectification. This means that the errors produced during the alignment of two images are propagated to the next poses of the robot's trajectory. Although this error propagation may highly affect the results, the accuracy of the proposed method makes possible the reconstruction of the 3D maps. In our first outdoor experiments, carried out with a common SICK 3D laser, we obtain very good mapping results. The RMS error for these experiments is about 2 cm for a typical displacement of 1 m and 1.25° for a turn of 25° . A similar error is obtained in the experiment carried out with thermal mapper data set. Each scene in this data set is obtained in a semi-structured environment from a 360° scan, and sensor used features are high accuracy and long range. Under these conditions we achieve a RMS of 0.89 m and 1.27° for displacement and turn, but in this case, the movement between each pose is bigger than 40 m. In the case of using sensors with low accuracy, as in our indoor experiments, the RMS error is about 1 cm for a 10 cm displacement and 1° for a turn of 3° . The main problem for the indoor experiments is the short range of the 3D sensor used. This problem may lead to some situations in which the robot can obtain almost no information about its environment. Under this kind of circumstances, our method fails to find planar descriptions for the scene's surfaces and so, the robot movement estimation is incorrect. Our data sets are public and available to download (Viejo and Cazorla 2013). As future

work, we are working on the extraction of other 3D surface features such as corners. We intend to improve the robustness of our method by including these new 3D features.

Another similar proposal was made by Weingarten et al. (2004), Weingarten and Siegwart (2005). He maintains an extended Kalman filter (EKF) on every plane extracted from each 3D scene. EKF are then used to compute a global rectification for the trajectory performed by the robot. The main difference is that Weingarten uses planes as landmarks and robot odometry in order to update the EKF information. Then, when a loop is detected, a robot pose error reduction is back-propagated. Our proposal is mainly focused on replacing the odometry information. In fact, both methods can be used together, ours for obtaining the robot movement estimation and Weingarten's for computing a global rectification.

Acknowledgments This work has been supported by project DPI2009-07144 from Ministerio de Educación y Ciencia (Spain) and GRE10-35 from Universidad de Alicante (Spain).

References

- Agrawal, M. (2006). A lie algebraic approach for consistent pose registration for general euclidean motion. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems* (pp. 1891–1897).
- Armesto, L., Minguez, J., & Montesano, L. (2010). A generalization of the metric-based iterative closest point technique for 3D scan matching. In *IEEE international conference on robotics and automation, ICRA 2010*, Alaska.
- Besl, P., & McKay, N. (1992). A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *14*(2), 239–256.
- Borrmann, D., & Nüchter, A. (2012). Thermo bremen data set. <http://kos.informatik.uni-osnabrueck.de/3Dscans/>
- Borrmann, D., Elseberg, J., Lingemann, K., Nüchter, A., & Hertzberg, J. (2008). Globally consistent 3D mapping with scan matching. *Robotics and Autonomous Systems*, *56*(2), 130–142.
- Brunnstrom, K., & Stoddart, A. (1996). Genetic algorithms for free-form surface matching. In: A. Stoddart (Ed.), *Proceedings of the 13th international conference on pattern recognition*, Vol. 4 (pp. 689–693).
- Censi, A. (2008). An ICP variant using a point-to-line metric. In *Proceedings of the IEEE international conference on robotics and automation (ICRA 2008)*, Pasadena, CA.
- Chen, C. S., Hung, Y. P., & Cheng, J. B. (1999). RANSAC-based DARCES: A new approach to fast automatic registration of partially overlapping range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *21*(11), 1229–1234.
- Chen, Y., & Medioni, G. (1991). Object modeling by registration of multiple range images. In: G. Medioni (Ed.), *Proceedings of the IEEE international conference on robotics and automation 1991*, Vol. 3 (pp. 2724–2729).
- Chung, D., Yun, I., & Lee, S. (1998). Registration of multiple range views using the reverse calibration technique. *Pattern Recognition*, *31*, 457–464.
- Dalley, G., & Flynn, P. (2002). Pair-wise range image registration: A study in outlier classification. *Computer Vision and Image Understanding*, *87*(1–3), 104–115.
- Dorai, C., Wang, G., Jain, A., & Mercer, C. (1998). Registration and integration of multiple object views for 3d model construction. *IEEE*

- Transactions on Pattern Analysis and Machine Intelligence*, 20(1), 83–89.
- Du, S., Zheng, N., Ying, S., & Wei, J. (2007a). ICP with bounded scale for registration of m-d point sets. In *Proceedings of the IEEE international conference on multimedia and expo, 2007* (pp. 1291–1294)
- Du, S., Zheng, N., Ying, S., You, Q., & Wu, Y. (2007b). An extension of the icp algorithm considering scale factor. In *Proceedings of the IEEE international conference on image processing, ICIP 2007*, Vol. 5 (pp. V-193–V-196).
- Fan, J., Zeng, M., Body, M., & Hacid, M. S. (2005). Seeded region growing: An extensive and comparative study. *Pattern Recognition Letters*, 26, 1139–1156.
- Godin, G., Rioux, M., & Baribeau, R. (1994). Three-dimensional registration using range and intensity information. In S. F. El-Hakim (Ed.), *Proceedings of SPIE, Videometrics III*, Vol. 2350 (pp. 279–290). Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference
- Goecke, R., Asthana, A., Pettersson, N., & Petersson, L. (2007). Visual vehicle egomotion estimation using the fourier-mellin transform. In *Proceedings of the IEEE intelligent vehicles symposium* (pp. 450–455).
- Johnson, A., & Hebert, M. (1997). Surface registration by matching oriented points. In M. Hebert (Ed.), *Proceedings of international conference on recent advances in 3-D digital imaging and modeling, 1997* (pp. 121–128).
- Kanatani, K. (1993). *Geometric computation for machine vision*. Oxford: Oxford University Press, Inc.
- Katz, R., Nieto, J., & Nebot, E. (2010). Unsupervised classification of dynamic obstacles in urban environments. *Journal of Field Robotics*, 27, 450–472.
- Kim, S., Jho, C., & Hong, H. (2003). Automatic registration of 3D data sets from unknown viewpoints. In *Workshop on frontiers of computer vision*
- Koch, O., & Teller, S. (2007). Wide-area egomotion estimation from known 3D structure. In *Proceedings of the IEEE conference on computer vision and pattern recognition CVPR '07* (pp. 1–8)
- Kümmerle, R., Triebel, R., Pfaff, P., & Burgard, W. (2008). Monte carlo localization in outdoor terrains using multilevel surface maps. *Journal of Field Robotics*, 25(6–7), 346–359.
- Martin Nevado, M., Gomez Garcia-Bermejo, J., & Zalama Casanova, E. (2004). Obtaining 3D models of indoor environments with a mobile robot by estimating local surface directions. *Robotics and Autonomous Systems*, 48(2–3), 131–143.
- Masuda, T., Sakaue, K., & Yokoya, N. (1996). Registration and integration of multiple range images for 3-D model construction. In *Proceedings of the 1996 international conference on pattern recognition (ICPR '96)*. IEEE Computer Society, Vol. 1–7270 (p. 879).
- Munoz-Salinas, R., Aguirre, E., García-Silvente, M., & Gonzalez, A. (2008). A multiple object tracking approach that combines colour and depth information using a confidence measure. *Pattern Recognition Letters*, 29(10), 1504–1514.
- Nüchter, A., Hertzberg, J., Lingemann, K., & Surmann, H. (2007). 6D SLAM—3D mapping outdoor environments. *Journal of Field Robotics*, 24(8–9), 699–722.
- Nuchter, A., Lingemann, K., & Hertzberg, J. (2007). Cached k-tree search for icp algorithms. In *Proceedings of the sixth international conference on 3-D digital imaging and modeling 3DIM '07* (pp. 419–426)
- Pulli, K. (1999). Multiview registration for large data sets. In *Proceedings of the second international conference on 3-D digital imaging and modeling* (pp. 160–168)
- Pulli, K. A. (1997). Surface reconstruction and display from range and color data. PhD thesis, University of Washington.
- Rusinkiewicz, S., & Levoy, M. (2001). Efficient variants of the ICP algorithm. In *Proceedings of the third international conference on 3-D digital imaging and modeling* (pp. 145–152)
- Salvi, J., Matabosch, C., Fofi, D., & Forest, J. (2007). A review of recent range image registration methods with accuracy evaluation. *Image and Vision Computing*, 25(5), 578–596.
- Segal, A., Haehnel, D., & Thrun, S. (2009). Generalized-icp. In *Proceedings of robotics: Science and systems*, Seattle.
- Shih, F. Y., & Cheng, S. (2005). Automatic seeded region growing for color image segmentation. *Image and Vision Computing*, 23, 877–886.
- Trucco, S., & Verri, A. (1998). *Introductory techniques in computer vision*. Upper Saddle River: Prentice-Hall.
- Turk, G., & Levoy, M. (1994). Zippered polygon meshes from range images. In *Proceedings of the 21st annual conference on computer graphics and interactive techniques, ACM* (pp. 311–318)
- Viejo, D., & Cazorla, M. (2007). 3D plane-based egomotion for slam on semi-structured environment. In *IEEE/RSJ international conference on intelligent robots and systems, IROS 2007* (pp. 2761–2766). doi:10.1109/IROS.2007.4399138.
- Viejo, D., Cazorla, M. (2008). 3D model based map building. In *International symposium on robotics, ISR 2008*, Seul.
- Viejo, D., & Cazorla, M. (2013). Rovit data set. <http://www.rovit.ua.es/dataset/>
- Weik, S. (1997). Registration of 3-D partial surface models using luminance and depth information. In *Proceedings of the international conference on recent advances in 3-D digital imaging and modeling* (pp. 93–100)
- Weingarten, J., & Siegwart, R. (2005). EKF-based 3D SLAM for structured environment reconstruction. In R. Siegwart (Ed.), *IEEE/RSJ international conference on intelligent robots and systems, 2005 (IROS 2005)* (pp. 3834–3839)
- Weingarten, J., Gruener, G., Siegwart, R. (2004). Probabilistic plane fitting in 3D and an application to robotic mapping. In *Proceedings of the IEEE international conference on robotics and automation ICRA '04*, Vol. 1 (pp. 927–932)
- Zhang, Z. (1994). Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2), 119–152.



Diego Viejo obtained his Bachelor Degree in Computer Science in 2002 and his PhD in 2008 both from the University of Alicante. Since 2004, he is a lecturer and a researcher for the Department of Computer Science and Artificial Intelligence (DCCIA) at the University of Alicante. His research interests are focused on computer vision applied on mobile robotics.



Miguel Cazorla received a BS degree in Computer Science from the University of Alicante (Spain) in 1995 and a PhD in Computer Science from the same University in 2000. He is currently Associate Professor with the Department of Computer Science and Artificial Intelligence of the University of Alicante. He has been post-doctoral fellow with Dr. Alan L. Yuille at the Smith-Kettlewell Eye Research Institute of San Francisco and with Dr. Sebastian Thrun at

Robot Learning Lab at Carnegie-Mellon University. His research interest areas are computer vision and mobile robotics (mainly using vision to implement robotics tasks).