Fast Pose Tracking Based on Ranked 3D Planar Patch Correspondences

Robert Cupec*, Emmanuel K. Nyarko*, Damir Filko*, Ivan Petrović**

**Faculty of Electrical Engineering, University of Osijek, Osijek, Croatia*
(e-mail: rcupec@etfos.hr; nyarko@etfos.hr; dfilko@etfos.hr)
***Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia*
(e-mail: ivan.petrovic@fer.hr)

Abstract: A fast robot pose tracking algorithm based on planar segments extracted from range images is presented. A range image obtained from a 3D sensor is transformed to a 2.5D triangle mesh from which planar segments are extracted. Using information provided by each planar segment based on its size and orientation, a directed search hypothesis generation algorithm using a tree structure is presented. The presented approach is experimentally evaluated using 3D data obtained by a Kinect sensor mounted on a mobile robot. Results indicate that the proposed method is much faster than similar previously proposed methods.

Keywords: ***robot vision, pose tracking, mobile robots, range image registration, range image segmentation.***

1. INTRODUCTION

The robot localization problem is considered to be one of the fundamental problems of autonomous mobile robotics (Thrun et al. 2000). This problem can be divided into two sub-tasks: local position tracking and global localization. Local position tracking, or pose tracking, compensates small, incremental errors in a robot's odometry given the initial robot's pose thereby maintaining the robot localized over time. Global localization, on the other hand, is the ability to determine the robot's pose in an a-priori or previously learned map, given no other information than that the robot is somewhere on the map.

The most common approach in vision-based robot localization systems is to use point features with assigned descriptors as landmarks (Harris & Stephens 1988, Se, Lowe & Little 2005). Alternatively, line features (Kosaka & Kak 1992, Vlassis, Motomura & Krose 2000) can be used. Methods implementing planar features are the least common (Weingarten & Siegwart 2006, Pathak et al. 2010).

In this paper, the application of 3D planar surfaces as features in pose tracking is considered. There are several advantages to using planar surfaces as features in localization compared to point or line features. Computational efficiency is improved and storage requirements are minimal since much less features are generated per scene. On the other hand, despite the fact that fewer features per scene are generated, a lot more information of the scene is available and it is also a lot easier to visualize the scene. Localization systems based on point or line features are impaired in both poorly and highly textured environments since: in the former, little or no features are generated, while in the latter, a lot of features are generated requiring a lot of processing time.

This work has been supported by the European Community's Seventh Framework Programme under grant No. 285939 (ACROSS) and by the University of Zagreb Development Fund under grant "Center of Excellence for Computer Vision".

We, however, consider the most important advantage to be the inherent saliency each planar feature holds. The larger the plane, the more significant the plane is as a feature. Larger planes are obtained from larger objects in the scene and the probability of such objects being moved or modified is rather low. Hence, large 3D planar surface features can be considered to be "stable" features which can be used in global localization.

A fast vision-based localization system based on 3D planar surfaces obtained from 2.5D images is proposed in this paper. First, the point cloud obtained by a 3D sensor is transformed to a 2.5D triangle mesh. Region growing is then applied to detect large 3D planar surfaces in the scene. These planar surfaces are used for local position tracking of a mobile robot. The experiments reported in this paper are performed using a commercially available low cost 3D sensor Kinect mounted on a mobile robot Pioneer 3DX. This paper is structured as follows. In Section 2 an overview of the related work is given. The proposed pose tracking algorithm is described in Section 3 and its experimental evaluation is given in Section 4.

2. RELATED WORK

Detecting planar surfaces from 2.5D images falls into the category of range image segmentation. Range image segmentation is the process of dividing, or segmenting, a range image such that all the points of the same surface belong to the same region. Basically, range images can be segmented into planar surfaces using either RANSAC (Fischler & Bolles, 1989), 3D Hough transform (Okada et al., 2001), Region growing or merging (Hoover et al., 1996) or Iterative Delaunay triangulation (Schmitt & Chen, 1991).

Ayache & Faugeras (1989) provide a framework for robot localisation using 3D lines and planes. They provide geometrical constraints to be used in the registration process. However, experimental validation is performed for 3D lines

only. Actual implementation of 3D planar surfaces used in image registration can be found in Weingarten & Siegwart, (2006) and Pathak et al. (2010).

The work by Pathak et al. (2010) is most related to ours. Surfaces are extracted from range images by implementing the region-growing algorithm mentioned in Poppinga et al. (2008). The plane parameter covariance matrix computed during plane extraction and the covariance of the registration solution play a central role in the plane matching. The hypotheses are generated using an algorithm which maximizes the overall geometric consistency within a search-space. A *consensus* approach similar to RANSAC is used but with two major differences: there is no random sampling involved and the solution is not based entirely on consensus maximization but also on the uncertainty volume of hypotheses. In the pre-processing step, the planes in both sets are initially sorted in descending order of evidence (the determinant of the pseudo-inverse of the covariance matrix of the plane) and a top fixed percentage is used only. This initial search-space is then pruned by finding all consistent two pairs of correspondences using six geometric constraints: size-similarity test, overlap test, cross-angle test, parallel consistency, and if available, rotation and translation agreement with odometry. In the main search step, each of these pairs is considered in turn and their largest rotation and translation consensus sets are built. For each of these consensus sets, the least-squares rotation and translation are determined, along with the volume of uncertainty given by the pseudo-determinant of the covariance matrix of the estimated pose. The pose corresponding to the consensus set with the minimum uncertainty volume having at least four pairs is selected as the chosen hypothesis. Finally, the authors show that their algorithm outperforms three other previously proposed algorithms: point-to-point ICP (Besl & McKay, 1992), point-to-plane ICP (Chen & Medioni, 1991) and 3D NDT (Magnusson, Lilienthal & Duckett, 2007).

3. POSE TRACKING ALGORITHM

In this section, details of the fast pose tracking algorithm will be provided. Initially, a formulation of the problem of pose tracking using planar surfaces is given followed by a short explanation of how the planes are generated and represented. A description of the geometrical constraints used for surface matching is then given. Finally, descriptions of the hypothesis generation and hypothesis evaluation stages of the algorithm are given.

3.1 Problem Formulation

Let us consider a set of planar surfaces observed by a 3D sensor from two different views. Let the pose of the 3D sensor corresponding to the first view be represented by reference frame S_A and the pose of the sensor corresponding to the second view by reference frame S_B. By processing the 3D sensor data acquired from each of these two views, two surface sets are obtained. The problem considered in this paper is to estimate the pose of S_A relative to S_B given two sets of surfaces obtained from the two views. Let this pose be represented by vector $\mathbf{w} = \left[\boldsymbol{\phi}^T \mid \mathbf{t}^T \right]^T$, where the vector $\boldsymbol{\phi} = [\alpha, \ \beta, \ \theta]^T$ represents the orientation of S_A relative to S_B defined by 3 angles α, β and θ and the vector $\mathbf{t} \in \mathbb{R}^3$ represents the position of S_A relative to S_B. Assuming that the surface parameters are obtained by a 3D sensor, the measurement uncertainty must be taken into account.

3.2 Detection and Representation of Planar Surfaces

2.5D images acquired by a 3D camera are segmented into sets of 3D points representing approximately planar surfaces using a similar split-and-merge algorithm as in Schmitt & Chen (1991), which consists of an iterative Delaunay triangulation method followed by region merging. This method provides a fast detection of planar surfaces as demonstrated in Section 4. At the end of the procedure, a set of planar surfaces is obtained, where each planar surface F is defined by: the unit normal $\mathbf{n}$; the offset of the plane ρ (distance of the plane from the origin in the direction of the normal); the centroid of the points supporting the surface $\mathbf{t}_F$; covariance matrix $\mathbf{C}_\mathbf{q}$ defining the uncertainty of the plane parameters $\mathbf{n}$ and ρ and covariance matrix $\mathbf{C}_F$ representing the distribution of the points belonging to the surface in the plane defined by $\mathbf{n}$ and ρ.

In order to define the uncertainty model of F, we introduce the surface reference frame S_F with the origin in $\mathbf{t}_F$, z-axis perpendicular to F and x- and y-axis defined by the eigenvectors of $\mathbf{C}_F$. Let us describe the uncertainty of the plane containing F by a disturbance vector $\mathbf{q} = \begin{bmatrix} s_1 & s_2 & r \end{bmatrix}^T$ representing the deviation of the true plane parameters $\mathbf{n}$ and ρ from the estimated parameters $\hat{\mathbf{n}}$ and $\hat{\rho}$. Vector $\mathbf{q}$ is defined by

$$\mathbf{n} = \frac{\hat{\mathbf{n}} + s_1 \cdot \mathbf{x}_F + s_2 \cdot \mathbf{y}_F}{\sqrt{1 + s_1^2 + s_2^2}}, \qquad \rho = r + \mathbf{n}^T \cdot \mathbf{t}_F, \tag{1}$$

where $\mathbf{x}_F$ and $\mathbf{y}_F$ represent x- and y-axis of S_F. The components of $\mathbf{q}$ are illustrated in Fig. 1.

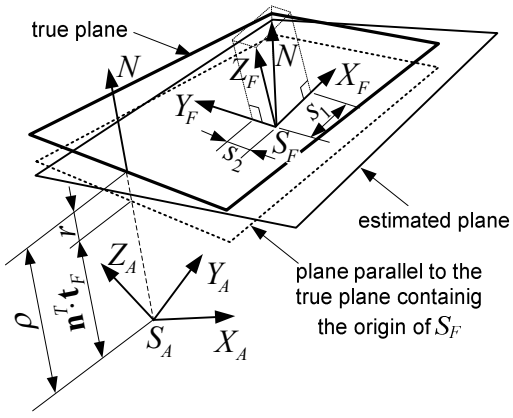

Fig. 1. Plane uncertainty model, where N represents the true plane normal.

Vector $\mathbf{q}$ is considered to be a random variable with zero mean and covariance matrix $\mathbf{C}_\mathbf{q} = \text{diag}(\sigma_{s,1}, \sigma_{s,2}, \sigma_r)$, where

$\sigma_{s,i} = \sigma_r / (\sigma_r + \lambda_i)$, i = 1, 2, σ_r is the estimated uncertainty of the surface centroid position in the direction of the surface normal and $\lambda_{1,2}$ are the eigenvalues of $\mathbf{C}_F$.

3.3 Geometrical Constraints for Surface Matching

In this sub-section, two geometric constraints used for surface matching in the proposed pose tracking algorithm are discussed. Let the pose of the reference frame S_A relative to the reference frame S_B be represented by the vector **w** described in Section 3.1. A surface pair (F, F') is considered to represent a correct match, if the parameters of the planes supporting these two surfaces are consistent with the pose **w** and if the overlapping of the point sets supporting these two surfaces is sufficient according to a predefined overlapping measure.

The first criterion is evaluated by transforming the plane supporting surface F into the reference frame $S_{F'}$ of surface F' using the pose **w**. In case of a perfect match, the projections of the normal of F onto the x- and y-axis of $S_{F'}$ as well as the offset of the transformed plane relative to $S_{F'}$ should be 0. Hence, the Mahalanobis distance between these 3 values and a zero-vector is taken as the measure of geometrical consistency of the match (F, F') with the pose **w**. Let us denote this distance by $d(F, F'; \mathbf{w})$. Thus, a surface pair (F, F') is considered to be consistent with the pose **w** if $d(F, F'; \mathbf{w}) \leq \tau_1$, where τ_1 is a predefined threshold which can be chosen according to the chi-square distribution for 3 DoF. The Mahalanobis distance is computed using the covariance matrices $\mathbf{C_q}$ of both surfaces as well as the covariance matrix describing the uncertainty of the pose **w**.

The second criterion is related to the overlap between the surface F transformed to S_B using the pose **w** and the surface F'. The measure of overlapping is the Mahalanobis distance between the centroids of surface F' and the transformed surface F computed using the covariance matrices $\mathbf{C}_F$ of both surfaces as well as the covariance matrix describing the uncertainty of the pose **w**.

3.4 Selection of Representative Surfaces

The computation time needed for registration of two surface sets can be significantly reduced by considering only a subset of all surfaces detected in a scene. This subset should be as small as possible but at the same time it should contain sufficient information for accurate estimation of the robot's motion. A straightforward criterion for selection of a representative surface set would be to sort the surfaces in descending order according to the number of supporting points and to take a certain number or a percentage of surfaces from the top of this list. In some cases, however, relatively small surfaces can contain information crucial for motion estimation, as explained in the following.

General rigid body motion has 6 degrees of freedom (DoF). In order to estimate all 6 DoF, at least three non-parallel plane pairs or correspondences are needed. With two non-parallel plane correspondences, 5 DoF are completely defined. A typical indoor scene contains at least 2 dominant non-parallel planar surfaces, e. g. the floor surface and a wall, as shown in Fig. 2a. In many cases, however, a scene is deficient in information needed to estimate the last DoF of the robot's motion. A typical example is the corridor shown in Fig. 2a, where the floor and the walls provide sufficient information for accurate estimation of 5 DoF of the robot's motion, while it lacks the surfaces perpendicular to the last DoF, i.e. the horizontal movement direction parallel to the walls. A rather small surface perpendicular to this direction (e.g. surface "A" marked in Fig. 2b) would have much greater importance then a much larger surface parallel to the floor or the sides of the corridor.

a) b)

Fig. 2. Sample images of different indoor scenes.

Hence, in this paper we propose using a surface ranking criterion based on the *information content factor* defined in the following. The distribution of the information available for motion estimation in a set of N surfaces can be represented by the matrix

$$\mathbf{Y} = \sum_{i=1}^{N} \mathbf{n}_i \cdot \mathbf{n}_i^T \cdot w_i \qquad (2)$$

where N is the number of considered surfaces, $\mathbf{n}_i$ is the normal of the i-th surface and w_i is the number of points supporting this surface. For a given surface, the value $\mathbf{n}^T \cdot \mathbf{Y} \cdot \mathbf{n}$ represents a measure of the total information provided in the direction of the surface normal **n** and the value

$$\omega = \frac{w}{\mathbf{n}^T \cdot \mathbf{Y} \cdot \mathbf{n}} \qquad (3)$$

represents a measure of the contribution provided by the surface to the total information in the direction of the surface normal. The strategy proposed in this paper is to rank the surfaces according to the value (3) and to consider only the first m surfaces in the registration process.

3.5 Hypotheses Generation

Due to the lack of descriptors, there is a high ambiguity in the feature correspondences in the initial correspondence set. Hence, standard RANSAC approach to hypothesis generation will not be effective. One approach would be to employ GCRANSAC approach as proposed in Cupec et al. (2009). GCRANSAC proved to be more reliable especially in the case of multiple possible correspondences and a high ambiguity in the initial correspondence set.

The general idea of the GCRANSAC approach is to update the pose and decrease the uncertainty at each step using more '*consistent*' pairs which are selected randomly. Implementing this basic idea of GCRANSAC, and in a desire to speed up the hypothesis generation step, we propose a new method which does not select pairs randomly, but rather sequentially

selects each pair in a given manner taking into account the inherent saliency of each plane i.e. plane-pair. Our method is similar in idea to the one proposed by Pathak et al. (2010), but has 2 main differences:

1. Instead of generating hypotheses from only 2 pairs of corresponding surfaces, our method builds a hypothesis by pairs one after another until the estimated orientation uncertainty becomes sufficiently low. Thereby, it allows for a case where none of 2 pairs of corresponding surfaces has sufficient information for accurate orientation estimation.
2. The hypothesis generation process is designed to generate more probable hypotheses before the less probable ones, allowing the algorithm to stop long before all possible hypotheses are considered. Thereby the necessary computation time is significantly reduced.

Let M represent the list of m planar surfaces obtained from the current scene and M' the list of m' planar surfaces obtained from the previous scene. The planar surfaces within each list are sorted in descending order of the information content factor, as explained in Section 3.4. Hence, the index of a surface in the list reflects its importance. The list of initial correspondences Q is created by finding surface pairs from M and M' which satisfy both geometrical constraints described in subsection 3.3. The matching candidates in the list Q are sorted in ascending order based on the sum of the indices of the surfaces in each pair. Thereby, the pairs with greater information content factor are positioned closer to the top of the list.

We propose a hypothesis generation algorithm, where each hypothesis is generated in two stages. The first stage involves using a tree structure to determine 5 DoF of the robot's motion: all 3 DoF defining the rotation and 2 DoF of the translation. In the second stage, the last DoF is estimated.

In the first stage, the hypothesis generation algorithm utilizes a tree structure. Each node in this tree structure is assigned pose information together with its uncertainty. The root node is assigned the initial pose. Apart from the root node, all other nodes on the tree correspond to a surface pair from the initial correspondence set, Q. The path from any node V to the root node represents a pose hypothesis and the pose assigned to node V is computed from the information contained in the matches corresponding to the nodes along this path.

The discussed tree structure is built by taking the first surface pair (F, F') from the list Q, and attaching a node V to every node V' in the tree for which the following conditions are satisfied.

1. Neither F nor F' is included in the pair corresponding to any node along the path from V' to the root node.
2. The pair satisfies the constraints given in Section 3.3 for the pose assigned to the node V'.

After including a new node in the tree, this node is assigned the pair (F, F') as well as the pose computed by updating the pose assigned to its parent node V' with the information provided by the pair (F, F') using an Extended Kalman Filter approach. If the estimated orientation uncertainty after the EKF update falls below a predefined threshold, the node is marked as an end node, i.e. it is not expanded and the second stage of the hypothesis generation algorithm is activated. A surface pair once taken from the list Q is removed from the list. The described procedure is repeated until either a predefined number of hypotheses n_{hyp} or a predefined number of surface matching operations n_{match} is reached. Since surface matching is the most time consuming operation in the hypothesis generation process, limiting the number of these operations has shown to be a good method for limiting the overall computation time of the algorithm.

The second stage of hypothesis generation begins with the pose estimated during the first stage. This pose is estimated with a rather low uncertainty in orientation and 2 translational DoFs. The eigenvector corresponding to the maximum eigenvalue of the covariance matrix describing the uncertainty of the estimated translation defines the direction of the last DoF which is to be determined in the second stage in order to complete a hypothesis. A set T of surface pairs which provide sufficiently reliable information for estimation of the last DoF is selected from the list Q. This selection is performed using the angle between the surface normals and the last DoF. Smaller angles indicate more reliable information. For each surface pair in the set T, a candidate value s of the last DoF is computed as the value which translates the surface F onto the position where it overlaps with the surface F'. Each pair is also assigned the variance σ describing the estimated uncertainty of s. For each pair (F, F') from T, the following *consensus measure* is then computed

$$\Im'(F,F';T)=\sum_{i=1}^{|T|}\frac{1}{\sqrt{2\sigma_i}}\exp\left(-\frac{(s-s_i)^2}{2\sigma_i}\right)\cdot\min\left\{|F|,|F'|\right\}, \quad (4)$$

where $|T|$ denotes the number of pairs in T, $|F|$ denotes the number of points in the range image supporting the surface F, s is the candidate value of the last DoF computed from the pair (F, F') and s_i and σ_i are the candidate value of the last DoF computed from the i-th pair in T and its estimated uncertainty respectively. Finally, the pose hypothesis is updated using EKF approach by the information provided by the surface pair from T with the highest consensus measure.

3.6 Hypotheses Evaluation

The result of the hypothesis generation procedure described in Section 3.5 is a set of hypotheses about the pose $\mathbf{w}$. For each hypothesis, the consensus set $\Omega(\mathbf{w}) \subseteq Q$ is identified containing all surface pairs satisfying the matching criteria described in Section 3.3. Each hypothesis is evaluated according to the following consensus measure

$$\Im(\mathbf{w})=\sum_{F\in M}\max_{(F,F')\in\Omega(\mathbf{w})}\left\{P\left(d\left(F,F';\mathbf{w}\right)\right)\cdot\min\left\{|F|,|F'|\right\}\right\}, \quad (5)$$

where d is the Mahalanobis distance explained in Section 3.3 and P is the probability that the Mahalanobis distance of two surfaces representing a correct match is $\geq d(F, F'; \mathbf{w})$ assuming that it is distributed according to the chi-square distribution with 3 DoF. The pose $\mathbf{w}$ with the highest

consensus measure $\Im(\mathbf{w})$ is taken as the final result of the pose tracking algorithm.

4. EXPERIMENTAL EVALUATION

In this section, the results of the experimental evaluation of the proposed pose tracking method are reported. The pose tracking algorithm described in Section 3 is implemented in C++ programming language and executed on a 3.40GHz Intel Pentium 4 Dual Core CPU with 2GB of RAM. The algorithm is experimentally evaluated using 3D data provided by a Kinect sensor mounted on a mobile robot Pioneer 3DX. The Kinect sensor is a 3D camera which produces 640 × 480 depth image with maximum 11-bit values which corresponds to 2048 levels of depth. In the experiments reported in this section, the depth images are subsampled to resolutions of 320 × 240 and 160 × 120.

The robot was teleoperated to move up and down in a straight line in two different hallways. Every 0.5m, the RGB image and depth image from the Kinect sensor and the robot odometry readings were automatically collected and stored. A total of 112 data readings were obtained. The algorithm proposed in this paper was tested offline using this database of images. The following parameters were used during the pose tracking experiment: $n_{hyp} = 20$; $n_{match} = 1000$ and $m = m' = 20$. The height and pitch of the Kinect sensor relative to the ground plane are considered constant since the sensor is mounted on a wheeled robot. An example of an image along with its point cloud and the extracted planar patches is given in Fig. 3.

In order to evaluate the pose tracking method, the translational pose obtained using vision was compared to that obtained using odometry. The pose data obtained using odometry cannot be assumed to represent exact ground truth values due to drift and slippage. However, the precision of the odometry is rather high at short distances. Hence, the variance of the error between the odometry pose and pose obtained by vision is used in this evaluation as a conservative estimate of the variance of the pose obtained by vision. The histograms of this pose error obtained using images of resolution 320 × 240 and 160 × 120 are shown in Fig. 4a. It can be noticed that there was only one sample with pose error greater than 100mm for images with a resolution of 160 × 120 while three such samples were obtained for images with resolution of 320 × 240. The range image registration time and the normalized cumulative histograms of the surface detection time are displayed in Fig. 4b. and Fig. 4c. – 4d. respectively.

As expected, the surface detection time depends on the image resolution, while the range image registration time is independent of the screen resolution since it depends on the predefined values of m, n_{hyp} and n_{match}. Statistical details of the pose error, surface detection time, range image registration time and total time are given in Table 1. The pose error was on average approximately 30 mm with a standard deviation of about 20 mm for both image resolutions. On average, for images with a resolution of 320 x 240, approximately 95ms was needed for surface detection while 25ms was needed for images with a resolution of 160 x 120. On the other hand, an average of 5ms was needed for image registration irrespective of the image resolution. Thus, on average, it took a total time of approximately 104ms for images with a resolution of 320 x 240 (the maximum was about 210ms) and approximately 34ms for images with a resolution of 160 x 120 (the maximum was about 90ms).

Apart from the fact that several parameters need to be set beforehand, we noticed that the proposed pose tracking algorithm had difficulties in situations where the scene had only 2 dominant non-parallel planar surfaces i.e. was deficient in information needed to estimate the last DoF of the robot's motion.

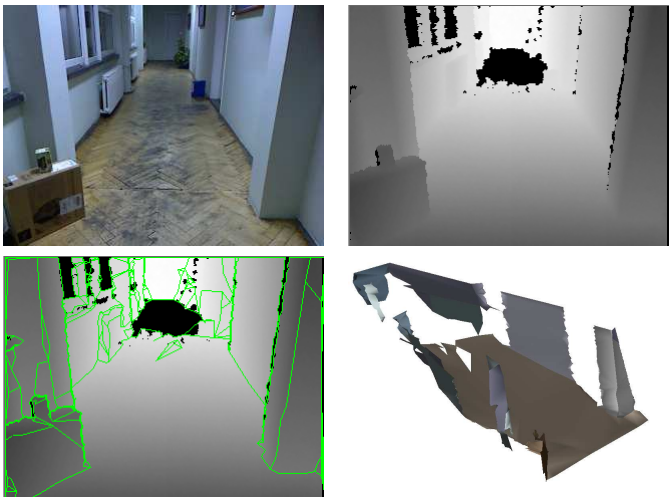

Fig. 3. Camera image (*top left*); depth image obtained by Kinect (*top right*); extracted planar patches (*bottom left*); extracted dominant planar patches shown in perspective (*bottom right*).

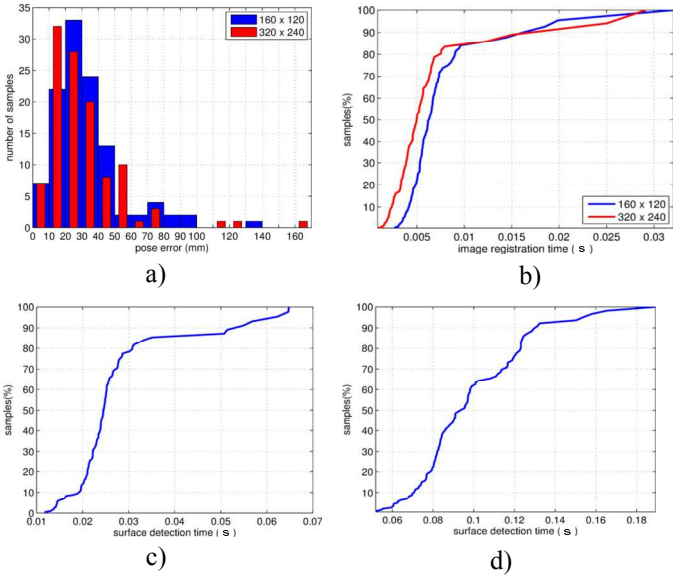

Fig. 4. Histograms of the pose error for both image resolutions (a); normalized cumulative histograms of the range image registration time for both image resolutions (b); normalized cumulative histogram of the planar surface detection time for images with resolution c) 160 × 120 d) 320 × 240.

Generally, depending on the dataset and hence the resolution of the images taken, the computation time for the surface detection varied from 0.43s on average for images with a resolution of 176 × 144 to 3s for images with a resolution of 541 × 361. The computation time for image registration varied from 0.04s to 4.7s depending on the percentage of the total number of planar surfaces obtained that were used in image registration. The experiments were performed on a 1.6GHz AMD Turion 2 × 64 laptop with 960MB of RAM. A comparison of our results to those obtained by Pathak et al. (2010) for images of similar resolution is summarized in Table 2. Considering these results, it can be concluded that the method proposed in this paper clearly outperforms their algorithm with respect to computational time.

Table 1. Statistical details of the pose error, surface detection time, image registration time and total time.

	Avg.	Std.	Min.	Max.
320 × 240				
Pose error (mm)	30.90	23.43	4.76	163.57
Surface detection time (ms)	93.13	25.12	51.82	189.30
Registration time (ms)	4.40	3.70	0.87	29.20
Total time (ms)	103.20	26.97	58.23	208.60
160 × 120				
Pose error (mm)	31.93	21.19	2.40	134.96
Surface detection time (ms)	24.88	9.80	11.72	64.78
Registration time (ms)	6.30	3.70	2.60	32.10
Total time (ms)	33.17	11.97	16.88	89.60

Table 2. Comparison of experimental results.

	Pathak et al. (2010)	Proposed method
Image resolution	176×144	160×120
No. of samples	5	108
Mean surface detect. time (ms)	430	24.88
Mean registration time (ms)	159.8*	6.30

*Mean image registration time is calculated for the situation where 50% of the total number of planar surfaces obtained was used in image registration.

Although the reported experiments were performed using a wheeled mobile robot, the odometry data was used only for evaluation of the pose tracking results and not for the motion estimation. Hence, the presented approach can also be used with a hand-held camera. Nevertheless, odometry or data obtained by an inertial sensor (Božek & Suriansky 2011), can be used to improve the speed and the robustness of the algorithm.

REFERENCES

Ayache, N. and Faugeras, O. D. (1989). Maintaining Representations of the Environment of a Mobile Robot, *IEEE Trans. on Robotics and Automation*, 5(6), 804–819.

Besl, P. J. and McKay, N. D. (1992). A method for registration of 3-d shapes, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(2), 239–256.

Božek, P. and Suriansky, J. (2011). Research into the Utilization of an Inertial Navigation System in Robotics, *Proc. of the 22nd Int. DAAAM Symposium*.

Chen, Y. and Medioni, G. (1991). Object modeling by registration of multiple range images, *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2724–2729.

Cupec, R., Nyarko, E.K., Kitanov, A. and Petrovic, I. (2009). Geometrically Constrained RANSAC for Stereo Image Registration in Presence of High Ambiguity in Feature Correspondence, *Proc. of the 4th European Conf. on Mobile Robots*, 179-185.

Fischler, M. A. and Bolles, R. C. (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, *Graphics and Image Processing*, 24(6), 381–395.

Harris, C. and Stephens, M. (1988). A combined corner and edge detector, *Proc. of the 4th Alvey Vision Conf.*, 147–151.

Hoover, A. et al. (1996) An Experimental Comparison of Range Image Segmentation Algorithms, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(7), 673-689.

Kosaka, A. and Kak, A. (1992). Fast Vision-Guided Mobile Robot Navigation using Model-Based Reasoning and Prediction of Uncertainties, *Computer Vision, Graphics, and Image Processing - Image Understanding*, 271–329.

Magnusson, M., Lilienthal, A. and Duckett, T. (2007). Scan registration for autonomous mining vehicles using 3D-NDT, *Journal of Field Robotics*, 24(10), 803–827.

Okada, K., Katami, S., Inaba, M. and Inoue, H. (2001). Plane Segment Finder: Algorithm, Implementation and Applications, *In Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), Seoul, Korea*, 2120–2125.

Pathak, K., Birk, A., Vaskevicius, N. and Poppinga, J. (2010). Fast Registration Based on Noisy Planes with Unknown Correspondences for 3D Mapping, *IEEE Trans. on Robotics*, 26 (3), 424 – 441.

Poppinga, J., Vaskevicius, N., Birk, A. and Pathak, K. (2008). Fast plane detection and polygonalization in noisy 3D range images, *IEEE Int. Conf. on Intelligent Robots and Systems (IROS 2008)*, 3378 – 3383.

Schmitt, F. and Chen, X., (1991). Fast segmentation of range images into planar regions, *Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR '91.)*, 710-711.

Se, S., Lowe, D.G. and Little J. (2005). Vision-Based Global Localization and Mapping for Mobile Robots, *IEEE Trans. on Robotics*, 21(3), 364–375.

Thrun, S., Fox, D., Burgard, W. and Dellaert, F. (2000). Robust Monte Carlo Localization for Mobile Robots. *Artificial Intelligence*, 128(1-2), 99–141.

Vlassis, N., Motomura, Y. and Krose, B. (2000). Supervised linear feature extraction for mobile robot localization, *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 3, 2979-2984.

Weingarten, J. and Siegwart, R. (2006). 3D SLAM using planar segments, *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 3062-3067.