Place Recognition Based on Matching of Planar Surfaces and Line Segments

Robert Cupec, Emmanuel Karlo Nyarko, Damir Filko, Andrej Kitanov, Ivan Petrović

Abstract

This paper considers the potential of using 3D planar surfaces and line segments detected in depth images for place recognition. A place recognition method is presented which is based on matching sets of surface and line features extracted from depth images provided by a 3D camera to features of the same type contained in a previously created environment model. The considered environment model consists of a set of local models representing particular locations in the modeled environment. Each local model consists of planar surface segments and line segments representing the edges of objects in the environment. The presented method is designed for indoor and urban environments. A computationally efficient pose hypothesis generation approach is proposed which ranks the features according to their potential contribution to the pose information, thereby reducing the time needed for obtaining accurate pose estimation. Furthermore, a robust probabilistic method for selecting the best pose hypothesis is proposed which allows matching of partially overlapping point clouds with gross outliers. The proposed approach is experimentally tested on a benchmark dataset containing depth images acquired in indoor environment with changes in lighting conditions and presence of moving objects. A comparison of the proposed method to FAB-MAP and DLoopDetector is reported.

Keywords

place recognition, computer vision, 3D camera, point clouds, planar surfaces, line segments

1. Introduction

Place recognition is one of fundamental problems in mobile robotics which has recently drawn much attention. From a practical point of view, efficient and reliable place recognition solutions covering a wide range of environment types would open a door for numerous applications of intelligent mobile machines in industry, traffic, public services, household etc. From a purely scientific point of view, our human curiosity urges us to find out how close to us an artificial agent can get in the ability to recognize places, which is a task that we perform with ease in everyday life.

The basic robot localization problem can be defined as determining the robot pose relative to a reference coordinate system. According to this definition, following six problems are considered in literature:

1. *initial global localization* – determining the robot pose relative to a global coordinate system assigned to its environment without any prior information;
2. *local pose tracking* – determining the robot pose relative to a global coordinate system knowing its previous pose;
3. *kidnapped robot problem* – detecting that the current robot pose estimation is incorrect and determining its true pose relative to a global coordinate system;
4. *motion estimation, odometry* – determining the robot pose relative to its previous pose;
5. *loop closing* – detecting situations where the robot arrives in the vicinity of a previously visited location;
6. *simultaneous localization and mapping (SLAM).*

The first three problems are related to robot localization in an existing environment map, while the last three problems are related to environment map building. Motion estimation and loop closing are necessary tools for SLAM, though they can be analyzed as separate problems. The first, third and fifth problems are actually variants of the place recognition problem. The main difference between them is related to the prior information and possibilities of the changes in the environment.

Initial global localization assumes no prior information about the robot's pose, while a loop closing algorithm can use the estimate of the robot's pose obtained by a SLAM system which includes the loop closing algorithm. A solution to the kidnapped robot problem can be regarded as an initial global localization algorithm in combination with a mechanism for estimating the reliability of the current pose estimate, which triggers the global localization if the pose information is assessed as unreliable.

Considering the problem of changing environment, there are two main types of changes: (i) changes in lighting conditions and (ii) appearance or disappearance of objects or presence of moving objects. The second type of environment change is referred to in this paper as presence of *dynamic objects*.

The application of loop closing is in the map building process which usually takes several minutes or few hours. Although, it is possible that during the map building process some changes in the lighting conditions or appearance/disappearance of some objects in the environment occur, such events can be considered accidental. Hence, the robustness of a loop closing algorithm to such changes, although desirable, is not crucial. Moreover, if necessary, it is possible to keep the environmental conditions under control for some limited time period, while the map building is in progress. On the other hand, if we consider the place recognition in context of global robot localization or the kidnapped robot problem, it is required that a robot localization system operates reliably during long time periods, in which significant changes in the environment can occur.

There are two main classes of vision-based robot localization approaches, appearance-based approaches and feature-based approaches.

In appearance-based approaches, each location in a robot's operating environment is represented by a camera image. Robot localization is performed by matching descriptors assigned to each of these images to the descriptor computed from the current camera image. The location corresponding to the image which is most similar to the currently acquired image according to a particular descriptor similarity measure is returned by the localization algorithm as the solution. The appearance-based techniques have recently been very intensively explored, especially the approaches based on bag-of-words (BoW) descriptors, for which impressive results have been reported (Cummins and Newman, 2009). BoW descriptors are created from local descriptors of point features detected in camera images by methods like SIFT (Lowe, 2004) and SURF (Bay et al., 2008).

In feature-based approaches, the environment is modeled by a set of geometric features such as point clouds (Thrun et al., 2005), points with assigned local descriptors (Se et al., 2005), line segments (Kosaka and Kak, 1992; Faugeras, 1993) or planar surface segments (Cobzas and Zhang, 2001; Pathak et al., 2010; Fallon et al., 2012), where all features have their pose relative to a local or a global coordinate system defined. The localization is performed by matching a set of features detected by the applied sensor to the features in the environment model. A search for a set of model features with a similar geometric arrangement to that of the detected features is performed and the robot pose which maps these two sets is selected as the solution. If more than one match is found, the one which minimizes some cost function is selected or multiple hypotheses with different probabilities are considered.

In this paper, a feature-based place recognition approach is considered, which uses planar surface segments and line segments as features. The features are extracted from depth images obtained by a 3D camera. The environment model which is used for localization is a topological map consisting of local metric models. Each local model consists of planar surface segments and line segments represented in the local model reference frame.

The feature based environment models consisting of line segments were intensively explored in the 90s (Kosaka and Kak, 1992; Faugeras, 1993). In this paper, we revisit the usage of line features, combine them with planar surface segments and compare this feature-based approach to the recently popular BoW-approach.

An advantage of a feature-based approach such as the one considered in this paper over appearance-based techniques is that it provides accurately estimated robot pose relative to its environment which can be directly used for visual odometry or by a SLAM system. An additional advantage which is expected to be gained by using depth information obtained by an active 3D camera instead of a 'standard' RGB or grayscale image is its less sensitivity to changes in lighting conditions. Furthermore, since the considered approach uses completely different type of features than the methods based on point feature detectors, it can be expected that it will perform better in situations where geometric features are predominant or more stable in a sense that they do not change with time. For example, large surfaces are more suitable for being used as landmarks since they usually represent parts of buildings, such as walls, floor, ceiling or large furniture whose position in the environment is fixed.

The performance of a feature-based localization system strongly depends on the efficiency of the applied hypothesis generation approach and reliability of the measure used to select the best hypothesis. Therefore, these two components are given extra focus in this paper. The hypothesis selection method we use is based on transforming the model to the camera reference frame using a pose hypothesis and measuring the similarity between the transformed model and the currently acquired depth image. This principle is commonly used for hypothesis evaluation based on laser scans (Thrun et al., 2005) or 3D point clouds (Fallon et al., 2012), where the independent beam model is assumed. This approach is, however, suitable for complete metric models of the environment, i. e. models which include all relevant surfaces completely reconstructed. Using a camera to build a complete map of an environment containing all relevant surfaces of the modeled environment can be an exhaustive process taking many images which must cover the entire mapped environment. The mapping is much easier if a 3D laser scanner is used. However, in the case of a ‘standard’ camera or a 3D camera with a relatively narrow field of view, the mapping can take a lot of time and effort.

Our approach allows localization using incomplete maps, i.e. maps with some parts of the environment missing. Such a map can be obtained by driving a robot with a camera mounted on it along a path the robot would follow while executing its regular tasks. Furthermore, the proposed approach is designed for maps consisting of a series of independent point clouds acquired by a 3D camera without their fusion, analogously to the appearance based approaches which use RGB images.

The original contributions of the research presented in this paper are:

- To the best of our knowledge this is the only research where a combination of line and surface features is systematically evaluated for application in place recognition under significant changes in lighting conditions and presence of dynamic objects. The proposed approach is applied in a form of image retrieval common to appearance-based approaches and compared to representative BoW-based methods FAB-MAP (Cummins and Newman, 2009) and DLoopDetector (Galvez-Lopez and Tardos, 2012).
- In order to make such an approach highly efficient, we propose a novel hypothesis generation method which generates a relatively small number of incorrect hypotheses even in cluttered environments.

- A novel probabilistic hypothesis evaluation method is proposed which is suitable for matching partially overlapping feature sets with gross outliers. This enables simple map building from a sequence of depth images which can be acquired during a single tour along a typical path the robot is expected to follow during its regular operation.

The scope of the research reported in this paper is restricted to the following:

- Although the proposed method is well suited for application within a SLAM system which exploits prior probability by using sequence information like those presented in (Kawewong et al., 2011; Milford, 2013; Milford and Wyeth, 2012), this paper is focused on place recognition from a single image, i.e. usage of any information about the relative camera poses from which the images are acquired is not considered.
- The application of the proposed method is constrained to indoor environments, since it relies on the presence of dominant planar surfaces and objects with straight edges. Furthermore, the 3D camera used in the reported research has diminished capabilities in daily light. Nevertheless, the proposed approach could easily be adapted for application with 3D laser scanner, which extends its applicability to urban environments.

The proposed place recognition approach is experimentally evaluated using two sets of depth images acquired by Microsoft Kinect sensor in indoor environments. The first set represents places which the evaluated system should recognize (it is simply a database of reference images) and the second set contains test images of the same places but acquired under different lighting conditions and with presence or absence of dynamic objects.

The rest of the paper is structured as follows. In Section 2, a short survey of the related research is given. Section 3 provides a definition of the place recognition problem considered in this paper and an overview of our approach. Detection and representation of surface and line features is described in Section 4. Sections 5 and 6 explain the methodology we apply for hypothesis generation and selection respectively. In Section 7, the results of the experimental evaluation of the proposed approach are reported. Finally, the paper is concluded with Section 8, where obtained experimental results are discussed and some directions for future research are suggested.

2. Related research

Recently, sophisticated 3D sensors at very affordable price appeared on the market, motivating a number of research teams to develop algorithms for processing of 3D point clouds obtained by such sensors. Highly efficient feature-based algorithms for visual odometry (Huang et al., 2011), SLAM (Endres et al. 2012; Stückler and Behnke 2013) and local pose tracking (Fallon et al., 2012) have been reported. While approaches based on registration of sets of 3D geometric features like points and surface segments are mainly used for visual odometry and pose tracking, place recognition research is dominated by appearance-based approaches which rely on intensity images only.

Impressive results have been achieved in the field of appearance-based place recognition (Cummins and Newman, 2009; Ciarfuglia et al., 2012; Liu and Zhang 2012; Milford, 2013; Galvez-Lopez and Tardos, 2012) by using 'standard' camera image.

A reasonable question which is still open is: "Does the geometry of 3D structures in an observed scene provide sufficient information to allow reliable distinction between different locations in the environment?"

Besides the information obtained from a 'standard' camera image, the approach presented in (Badino et al., 2012) uses range data obtained by two lidars. An approach which uses only 3D point clouds for loop closing is proposed in (Granström et al., 2011) Point clouds are described by rotationally invariant geometric and statistical features which are used as input to a non-linear classifier based on AdaBoost algorithm.

A variant of FAB-MAP which uses 3D information obtained by a range sensor is presented in (Paul and Newman, 2010). This approach incorporates the observation of spatial ranges corresponding to pairs of visual words. The image is then described by a random graph which models a distribution over word occurrences as well as their pairwise distances.

Our approach belongs to a class of methods which perform registration of data obtained by a 3D sensor based on planar surface segments and then select the best camera pose according to a surface matching score.

In (Cobzas and Zhang, 2001) a trinocular stereo system is used for building a topological map where each node consists of 360˚ grayscale and disparity image of the surrounding space. Edges in grayscale images are detected by the Canny algorithm and then Delaunay triangulation is performed followed by a region growing segmentation based on average triangle intensity to form planar segments. Localization is performed by matching currently detected planar segments with those from the previously built map. Since the initial correspondence matching relies on the average segment intensity it can be expected that this system is very sensitive to scene illumination changes. Our approach, on the other hand, does not use intensity information which makes it more robust to changes in lighting conditions. Furthermore, although our approach could use images acquired by turning in place for 360˚, it can also work with a reduced set of images which cover the robot's environment only partially.

The work (Pathak et al., 2010) is methodologically most related to ours. Surfaces are extracted from range images and matched to the surfaces in the environment model. 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: similarly to our approach, 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. Our method is similar in idea to the one proposed by Pathak et al. (2010), but has three main differences: (i) Instead of generating hypotheses from only 2 pairs of corresponding surfaces, our method builds a hypothesis by considering 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. (ii) 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. (iii) In addition to planar surfaces, our approach uses line features, which are very useful in the situations where the surfaces in the observed scene do not provide sufficient information for estimating all degrees of freedom of the camera pose.

The proposed method builds upon the approach presented in (Cupec et al., 2012; Cupec et al., 2013). In this paper, an improved approach is presented, which includes application of line segment features as well as a novel probabilistic hypothesis evaluation method based on surface sampling. Furthermore, a contribution of this paper to the place recognition research field is the evaluation of a geometric feature-based approach using a benchmark dataset consisting of depth images of scenes with moving objects and illumination changes similar to the one presented in (Pronobis et al., 2010) which consists of RGB images.

3. Problem description and overview of the approach

The place recognition problem considered in this paper can be formulated as follows. Given an environment map consisting of local models representing particular locations in the considered environment together with spatial relations between them and a camera image acquired somewhere in this environment, the goal is to identify the camera pose at which the image is acquired. The term 'image' here denotes not only a standard RGB-image but also a depth image or a point cloud acquired by a 3D camera such as the Microsoft Kinect sensor. In general, a local model is any information describing a particular location which can be used to identify this location. This can be a single RGB-image, a point cloud, a laser scan or a set of features extracted from one or a series of such perception sensor outputs. The proposed method is based on local models consisting of planar surface segments and edge line segments extracted from a single depth image. This type of local model is metric in the sense that it consists of geometric features with defined poses relative to the local model reference frame. In general, the spatial relations between local models in a map can be topological or metric. In a metric map consisting of local models, which can be built by a SLAM algorithm, each local model is assigned the absolute pose of its reference frame relative to the global coordinate system of the map. Our approach provides the estimate of the camera pose relative to the local model reference frame, which can easily be transformed into a global pose in the case of a metric map. Nevertheless, the investigation reported in this paper does not consider spatial relations between local models nor the global structure of the map. We just focus on identifying the local model representing the particular location at which the camera image is taken as well as the camera pose relative to this local model's reference frame. Formally, given a set of local models $\{M_1, M_2, ..., M_N\}$ each representing a cloud of 3D points described by the point coordinates relative to the model reference frame $S_{M,i}$, $i = 1, ..., N$, the localization method described in Sections 4, 5 and 6 returns the index i of the local model M_i representing the current camera location together with the pose of the camera reference frame S_C relative to $S_{M,i}$. The camera pose can be represented by vector $\boldsymbol{w} = \left[\boldsymbol{\phi}^T \;\vdots\; \boldsymbol{t}^T\right]^T$, where $\boldsymbol{\phi}$ is a 3-component vector describing the orientation and $\boldsymbol{t}$ is a 3-component vector describing the position of S_C relative to $S_{M,i}$. Throughout the paper, symbol $\boldsymbol{R}(\boldsymbol{\phi})$ is used to denote the rotation matrix corresponding to the orientation vector $\boldsymbol{\phi}$.

The basic structure of the proposed place recognition approach is the standard feature-based localization scheme consisting of the following steps:

1. feature detection,
2. feature matching,
3. hypothesis generation,
4. selection of the best hypothesis.

The considered approach uses planar surface segments obtained by segmentation of a depth image and line segments obtained by segmentation of depth discontinuity contours. These features are common in indoor scenes, thus making our approach particularly suited for this type of environments. Extraction of surface and line features from depth images and their representation is described in Section 4.

Place recognition is accomplished by registration of the feature set extracted from the currently acquired image of a scene with the feature sets of the local models in the map. The first feature set is referred to in the following as *scene features* and the second one as *local model features.* For registration of these two feature sets an appropriate optimization tool could be used. A common optimization approach used for registration of point features is bundle adjustment. Usually a bundle adjustment method proceeds after a correspondence pruning step. When point features are used, the initial correspondences are established using appropriate local descriptors, which usually provide over 80% correct correspondences (Lowe, 2004) and then the obtained initial correspondence set is pruned using an appropriate method such as RANSAC. For planar surfaces, however, there are no commonly accepted descriptors which are shown to provide such a high matching rate. Since we do not use local descriptors for the initial surface matching, but only weak geometric constraints, the hypothesis generation step must handle many false correspondences. Therefore, we use an approach based on Extended Kalman Filter (EKF) in the hypothesis generation step which allows efficient selection of correct correspondences between many false correspondences, as explained in Section 5. Our approach is based on the following idea. At least three scene surface segments must be matched to local model surface segments in order to determine all six degrees of freedom (DoF) of the camera pose relative to the local model's reference frame. An efficient way of selecting a set of correct feature correspondences is to do it sequentially, where each selected correspondence poses a geometric constraint for the selection of the next correspondence. Since each surface is assigned measurement uncertainty information, this information is used for formulating the aforementioned geometric constraints. We use EKF as a commonly used tool for sequential estimation using probabilistic models. It is used in our approach to sequentially estimate the camera pose relative to a local model by a series of measurement updates, where the measurements are relative poses of corresponding scene and local model features. After each measurement update step of EKF, the uncertainty of the estimated pose is reduced resulting in a stronger geometric constraint for selection of the next feature pair. EKF can also be regarded as an optimizer, which minimizes the variance of the estimation error. The procedure stops when a desired accuracy is achieved.

Assuming that some a priori information about the camera pose relative to the environment is available, even if its uncertainty is rather high, it can help reject many false matches. For example, in the case of a wheeled robot, the camera pose relative to the gravity axis can be determined quite accurately. This information can then be used to distinguish between the horizontal and vertical surfaces, thereby reducing the number of initial feature correspondences significantly. Nevertheless, even with geometric constraints, a high percentage of false pairs of corresponding features can be expected. Since a sequence of at least three such pairs is needed for camera pose estimation, the data association in the case of surface and line features assumes examining many combinations of possible feature correspondences. A pose computed from a particular sequence of feature correspondences is referred to herein as a *pose hypothesis*. A pose hypothesis can be represented by a pair (i, $\boldsymbol{w}$), where i is the index of a local model and $\boldsymbol{w}$ is the estimated camera pose relative to that model.

Examining all possible sequences of feature pairs would in general require an enormous computational effort. A method proposed in (Cupec et al., 2012) ranks the feature pairs according to their potential usefulness for camera pose estimation and generates camera pose hypotheses using EKF approach starting with the most promising pairs. A detailed explanation of this method is given in Section 5. This method is used within the discussed approach to generate one or more camera pose hypotheses for each local model in the map.

Finally, the most probable of all generated hypotheses must be selected. This hypothesis represents the final solution of the place recognition problem. The hypothesis selection method used within our approach is described in Section 6.

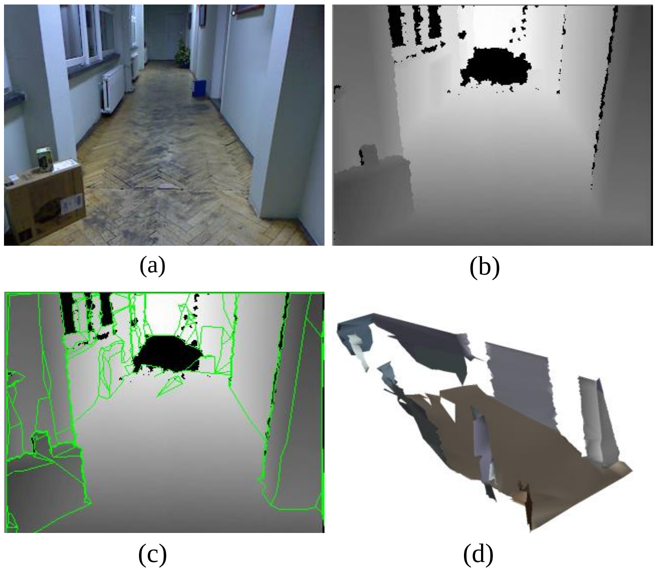
4. Planar surface and line segment Model

The feature detection stage of our approach results in a 3D model of a scene captured by a 3D camera. This 3D model consists of planar surface segments and edge line segments. The extraction and representation of surface and line features are described in Sections 4.1 and 4.2 respectively.

4.1 Detection and representation of surface segments

Depth images acquired by a 3D camera are segmented into sets of 3D points representing approximately planar surface segments using a similar split-and-merge algorithm as in (Schmitt and Chen, 1991), which consists of an iterative Delaunay triangulation method followed by region merging. Instead of a region growing approach used in the merging stage of the algorithm proposed in (Schmitt and Chen, 1991), we applied a hierarchical approach proposed in (Garland et al., 2001) which produces less fragmented surfaces while keeping relevant details. By combining these two approaches a fast detection of dominant planar surfaces is achieved. The result is segmentation of a depth image into connected sets of approximately coplanar 3D points each representing a segment of a surface in the scene captured by the camera. An example of image segmentation to planar surface segments is shown in Fig. 1.

The parameters of the plane supporting a surface segment are determined by least-square fitting of a plane to the supporting points of the segment. Each surface segment is assigned a reference frame with the origin in the centroid of the supporting point set and z-axis parallel to the supporting plane normal. The orientation of x and y-axis in the supporting plane are defined by the eigenvectors of the covariance matrix $\boldsymbol{\Sigma_p}$ representing the distribution of the supporting points within this plane. The purpose of assigning reference frames to surface segments is to provide a framework for surface segment matching and EKF-based pose estimation explained in Section 5.


(a) (b)

(c) (d)

Fig. 1. An example of image segmentation to planar surface segments: (a) RGB image; (b) depth image obtained by Kinect, where darker pixels represent points closer to the camera, while black points represent points of undefined depth; (c) extracted planar surface segments and (d) 3D model consisting of dominant planar surface segments.

The uncertainty of the supporting plane parameters is described by introducing three random variables forming the disturbance vector $\boldsymbol{q} = [s_x, s_y, r]^T$. These three variables describe the deviation of the true plane parameters from the measured plane parameters. The true plane is defined by the equation

$$ {}^F\boldsymbol{n}^T \cdot {}^F\boldsymbol{p} = {}^F\rho, \tag{1}$$

where ${}^F\boldsymbol{n}$ is the unit normal of the plane represented in the surface segment reference frame S_F, ${}^F\rho$ is the distance of the plane from the origin of S_F and ${}^F\boldsymbol{p}$ is an arbitrary point represented in S_F. In an ideal case, where the measured plane is identical to the true plane, the true plane normal is identical to the z-axis of S_F, which means that ${}^F\boldsymbol{n} = [0, 0, 1]^T$, while ${}^F\rho = 0$. In a general case, however, the true plane normal deviates from the z-axis of S_F and this deviation is described by the random variables s_x and s_y, representing the deviation in directions of the x and y-axis of S_F respectively, as illustrated in Fig. 2 for x direction.

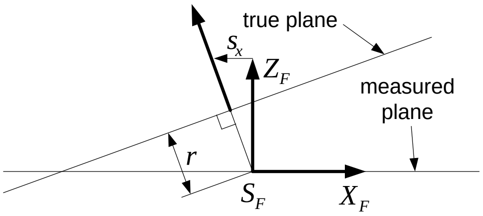


Fig. 2. Plane uncertainty model.

The unit normal vector of the true plane can then be written as

$$ {}^F\boldsymbol{n} = \frac{1}{\sqrt{s_x^2 + s_y^2 + 1}} \begin{bmatrix} s_x & s_y & 1 \end{bmatrix}^T \tag{2}$$

The random variable r represents the distance of the true plane from the origin of S_F, i. e.

$$ {}^F\rho = r\,. \tag{3}$$

We use a Gaussian uncertainty model, where the disturbance vector $\boldsymbol{q}$ is assumed to be normally distributed with $\mathbf{0}$ mean and covariance matrix $\boldsymbol{\Sigma_q}$. Covariance matrix $\boldsymbol{\Sigma_q}$ is diagonal matrix with variances σ_{sx}^2, σ_{sy}^2 and σ_r^2 on its diagonal. These variances are computed from the uncertainties of the supporting point positions, which are determined using a triangulation uncertainty model analogous to the one proposed in (Matthies and Shafer, 1987). Computation of covariance matrices $\boldsymbol{\Sigma_q}$ is explained in (Cupec et al., 2013).

Finally, a scene surface segment is denoted in the following by the symbol F associated with the quadruplet

$$ F = \left({}^C\boldsymbol{R}_F,\ {}^C\boldsymbol{t}_F, \boldsymbol{\Sigma_q}, \boldsymbol{\Sigma_p} \right), \tag{4}$$

where ${}^C\boldsymbol{R}_F$ and ${}^C\boldsymbol{t}_F$ are respectively the rotation matrix and translation vector defining the pose of S_F relative to the camera coordinate system S_C. Analogously, a local model surface segment is represented by

$$ F' = \left({}^M\boldsymbol{R}_{F'},\ {}^M\boldsymbol{t}_{F'}, \boldsymbol{\Sigma_{q'}}, \boldsymbol{\Sigma_{p'}} \right). \tag{5}$$

4.2 Detection and representation of line segments

Discontinuities in depth images often appear as contours representing the edges of objects in a scene. The edges of objects with a fixed position in the robot's environment can be used for feature-based robot localization and pose tracking.

In our approach, depth discontinuity contours are segmented into line segments by Douglas-Peucker algorithm (Douglas and Peucker, 1973). Only the line segments longer than a predefined threshold are considered.

A scene line segment is denoted in the following by symbol F associated with quadruplet

$$ F = \left({}^C\boldsymbol{p}_1,\ {}^C\boldsymbol{p}_2, \boldsymbol{\Sigma}_{\boldsymbol{p},1}, \boldsymbol{\Sigma}_{\boldsymbol{p},2} \right), \tag{6}$$

where ${}^C\boldsymbol{p}_1$ and ${}^C\boldsymbol{p}_2$ represent the coordinate vectors of the endpoints of the line segment and $\boldsymbol{\Sigma}_{\boldsymbol{p},1}$ and $\boldsymbol{\Sigma}_{\boldsymbol{p},2}$ covariance matrices describing their uncertainties. Covariance matrices $\boldsymbol{\Sigma}_{\boldsymbol{p},1}$ and $\boldsymbol{\Sigma}_{\boldsymbol{p},2}$ are computed using the aforementioned triangulation uncertainty model. Local model line segments are represented analogously. The same symbol F is used for both surface and line features in order to simplify explanations related to both types of features.

5. Hypothesis generation

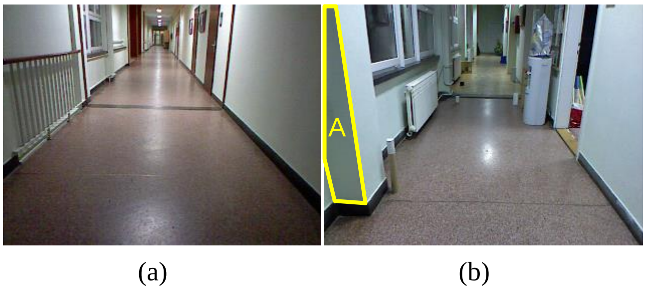
After feature detection, the discussed place recognition algorithm proceeds by data association, i.e. by matching the features detected in a scene to the features of all local models in the map. The result of this matching is an initial set of feature pairs (F, F'), where each pair associates a local model feature F' to a scene feature F. This pair set is then used to generate pose hypotheses. Each pose hypothesis is generated from a sequence of feature pairs using EKF. A commonly used hypothesis generation approach is RANSAC (Fischler and Bolles, 1981). This approach assumes generating pose hypotheses from the sequences of randomly selected feature pairs. In the case considered herein, such an approach is inefficient because a large number of hypotheses would have to be generated in order to obtain a correct hypothesis. We give two reasons for that. First, there is a high probability that a sequence consisting of a fixed number of randomly selected pairs contains only small surfaces whose parameters have a high uncertainty, resulting in inaccurate pose estimation. Increasing the number of pairs per sequence would also increase the computational effort. Second, there is a high probability that a randomly selected sequence consists of approximately parallel surfaces or two sets of approximately parallel surfaces, from which a 6 DoF camera pose cannot be computed accurately.

The hypothesis generation approach described in this section is designed to reduce these problems by using a data driven selection method to form sequences of feature pairs instead of random sampling. The proposed approach has the following properties (i) feature pairs are ranked according to a measure of their usefulness in the pose estimation process and hypotheses are generated from a relatively small number of the most 'useful' pairs and (ii) hypotheses are generated by introducing feature pairs one by one, where the selection of the next pair is subject to the geometrical constraints provided by the already selected pairs.

5.1 *Surface segment ranking*

If very small surface segments are used for pose estimation, three surface segments might not be sufficient for computing the camera pose with a desired accuracy. There are two main reasons why it is reasonable to generate hypotheses using large surface segments prior to smaller ones. First, surfaces supported by a high number of image points have small parameter uncertainty. Thus, a few large surface matches are usually sufficient to estimate the camera pose with a desired accuracy. Second, larger surfaces are more probable to have a fixed position in the environment. Hence, a straightforward approach would be to generate hypotheses by considering only a representative subset of the largest surface segments. In some cases, however, relatively small surfaces can contain information crucial for motion estimation, as explained in the following.

With two non-parallel plane correspondences, 5 of 6 DoF are completely defined. A typical indoor scene contains at least two dominant non-parallel planar surfaces, e. g. the floor surface and a wall, as shown in Fig. 3. In many cases, however, a scene is deficient in information needed to estimate the last, sixth DoF of the robot's motion. A typical example is the corridor shown in Fig. 3(a), 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 sixth DoF, i.e. the horizontal movement direction parallel to the walls. A rather small surface perpendicular to this direction (e.g. surface denoted by “A” in Fig. 3(b)) would have much greater importance then a much larger surface parallel to the floor or the sides of the corridor.



(a) (b)

Fig. 3. Sample images of typical indoor scenes.

In (Cupec et al., 2012) surface ranking based on the *information content factor* is proposed. The idea of this approach is explained in the following. A planar surface provides information for estimating three of the total 6DoF of the camera motion. A set Z of planar surface segments contains sufficient information for estimating all 6DoF of the camera motion only if for every $\boldsymbol{v} \in \mathbf{R}^3$ there is a segment $F_i \in Z$ with normal $\boldsymbol{n}_i$ such that $\boldsymbol{v}^T\boldsymbol{n}_i \neq 0$. Let us assume that each image point lying on a particular surface segment $F_i \in Z$ is assigned a normal of that surface. Then the distribution of normal directions over the entire set Z can be represented by a covariance matrix

$$\boldsymbol{Y} = \sum_{F_i \in Z} \boldsymbol{n}_i \cdot \boldsymbol{n}_i^T \cdot \mu_i \tag{7}$$

where $\boldsymbol{n}_i$ is the normal of F_i and μ_i is the number of points supporting this surface segment. For a given unit vector $\boldsymbol{v}$, the value $\boldsymbol{v}^T \cdot \boldsymbol{Y} \cdot \boldsymbol{v}$ can be regarded as a measure of the total information for pose estimation in the direction $\boldsymbol{v}$ contained in Z. All points of a surface segment F_i contribute to the total information in the direction of the surface normal. Hence, the contribution of that surface segment in the direction of its normal $\boldsymbol{n}_i$ is equal to its number of supporting points μ_i. Since the total information in direction $\boldsymbol{n}_i$ contained in Z is $\boldsymbol{n}_i \cdot \boldsymbol{Y} \cdot \boldsymbol{n}_i^T$, the value

$$\omega_i = \frac{\mu_i}{\boldsymbol{n}_i^T \cdot \boldsymbol{Y} \cdot \boldsymbol{n}_i} \tag{8}$$

represents a measure of the contribution of F_i to the total information in the direction of its surface normal. This value is referred to in the following as the *information content factor*. The strategy proposed in (Cupec et al., 2012) is to rank the surfaces according to the value (8) and to consider only the first n_{surf} surface segments in the hypothesis generation process. The result of this ranking is a list of surface segments sorted by the information content factor. The index of a surface segment in this list is referred to in the following as *information content index*. Assuming that the list is sorted in descending order, the smaller the information content index the more useful the segment is for the purpose of pose estimation.

5.2 Surface segment matching

The hypothesis generation process starts by forming the set of initial surface segment matches. This set is formed by considering all possible pairs of surface segments (F, F') and accepting only those which satisfy two criteria, *coplanarity criterion* and *overlapping criterion*, with respect to the initial estimate of the camera pose relative to the local model.

A scene surface segment and a local model surface segment satisfy the coplanarity criterion if the difference between their parameters is within an uncertainty region determined by the uncertainty of the parameters of both surface segments as well as the uncertainty of the initial pose estimate. This approach is commonly used in EKF-based registration approaches. In our implementation of this approach, the parameters of the supporting plane of the surface segment F are transformed from the camera reference frame S_C into the local model reference frame S_M using the initial camera pose estimate and then from S_M to the reference frame of segment F'. The difference between the transformed plane parameters of F and the plane parameters of F' is formulated as innovation function $\boldsymbol{e}(F, F', \boldsymbol{w})$ which maps the parameters of F and F' and the initial camera pose $\boldsymbol{w}$ to a 3-component innovation vector. This vector is then evaluated using Mahalanobis distance

$$d_{\boldsymbol{q}}(F, F', \boldsymbol{w}) = \boldsymbol{e}^T(F, F', \boldsymbol{w})\boldsymbol{Q}_{\boldsymbol{q}}^{-1}\boldsymbol{e}(F, F', \boldsymbol{w}), \tag{9}$$

where $\boldsymbol{Q_q}$ is a covariance matrix computed from the covariance matrices $\boldsymbol{\Sigma_q}$ and $\boldsymbol{\Sigma_{q'}}$ describing the uncertainty of the plane parameters of the compared surface segments and the covariance matrix $\boldsymbol{\Sigma_w}$ describing the uncertainty of the initial camera pose estimate. The coplanarity constraint can be formulated as

$$d_{\boldsymbol{q}}(F, F', \boldsymbol{w}) \le \varepsilon_{\boldsymbol{q}}, \tag{10}$$

where the threshold $\varepsilon_{\boldsymbol{q}}$ can be computed according to a desired matching probability assuming χ^2 distribution of $d_{\boldsymbol{q}}$ distance.

In addition to the coplanarity criterion, a second matching criterion related to the overlap between the transformed scene surface segment and a local model surface segment is used. A computationally efficient overlap measure is formulated by describing the spatial distributions of the supporting point sets of the matched surface segments by the first and second order statistics. Matched surface segments F and F' are represented each by a single point positioned at their centroids ${}^S\boldsymbol{t}_F$ and ${}^M\boldsymbol{t}_{F'}$ respectively, where the uncertainty of the position of these representative points is described by the covariance matrices $\boldsymbol{\Sigma_p}$ and $\boldsymbol{\Sigma_{p'}}$. The representative point of F is then transformed from the camera reference frame into the local model reference frame and the Mahalanobis distance between the obtained point and the representative point of F' is used as a measure of overlap between the matched surface segments.

The details regarding the two matching criteria are given in Appendix A.

5.3 Hypothesis tree generation guided by surface information content

The hypothesis generation approach proposed in this section is based on building a tree structure for each local model, where each node in this tree is related to a feature pair and each path from a leaf node to the root node represents a sequence of feature pairs from which a pose hypothesis is generated. Although it is possible to use both the surface and line features in equal manner, the current implementation uses only surface segments for estimating all three rotational and two translational DoF, while both the surface and line segments are used for determining the last translational DoF.

The initial matching results in a set of surface segment pairs. Each pair is assigned a weight representing the sum of the information contents indices of its surface segments. The pairs are then sorted in ascending order forming a *match queue*. The next stage in hypothesis generation is building the *hypothesis tree*. The root of the tree represents the initial pose estimate with its uncertainty. The hypothesis tree building proceeds by taking a pair from the top of the match queue, testing this pair for compatibility with each of the nodes in the tree and appending a new node representing the considered pair to all nodes with which it is compatible. A pair (F, F') is compatible with a node V if (i) neither F nor F' is included in the pair corresponding to any node along the path from V to the root node and (ii) the pair satisfies the coplanarity and overlap constraint explained in Section 5.2 with respect to the pose $\boldsymbol{w}$ assigned to the node V.

After a new node is appended to the tree, the pose estimate assigned to the parent node is updated by EKF using the measurement provided by the considered pair. The new node is assigned the updated pose together with its uncertainty. This procedure continues until a predefined number N_H of branches are generated, where each branch represents a pose hypothesis. Such a hypothesis tree is constructed in (Kosaka and Kak, 1992). The novelty of our approach is in using information contents factor to determine the order in which the surface segment pairs are appended to the tree.

We also introduced a mechanism for reducing the number of similar hypotheses. Any sequence of correct correspondences results in a correct hypothesis. Since there can be a number of correct correspondence sequences, many similar hypotheses are generated. After a new node is appended to the hypothesis tree, the uncertainty of the pose corresponding to this node is evaluated. If all three rotational DoF are estimated with a satisfactory accuracy, the pose is compared to the other poses estimated during the construction of the hypothesis tree. The accuracy of the estimated three rotational DoF is considered to be sufficient if the maximum eigenvalue of the rotational part of the covariance matrix $\boldsymbol{\Sigma}_w$ is less than some predefined value τ_ϕ.

If the camera orientation is estimated with a low uncertainty, two translational DoF are also sufficiently accurate. For example, two pairs of non-parallel surface segments are sufficient to determine all three rotational DoF. If the involved surface segments are sufficiently large, which implicates that their parameters are estimated with low uncertainty, then the estimated orientation would also be accurate. Furthermore, correspondence between two non-parallel planes defines also two translational DoF. The remaining undetermined translational DoF is in the direction of the line defined by the intersection of the two planes.

The pose obtained after appending a new node to the hypothesis tree with three rotational DoF estimated with a specified accuracy is referred to in the following as a *5 DoF hypothesis*. The new 5 DoF hypothesis is compared to all previously generated 5 DoF hypotheses. Only if it is not similar to any other hypotheses, the procedure for determining the remaining DoF is started in order to complete the pose hypothesis. Hence, the sixth DoF is estimated only once for all similar 5 DoF hypotheses, which reduces the computational effort. The method applied for estimation of the sixth DoF is described in Section 5.4.

The proposed hypothesis generation algorithm is given in the following.

Algorithm 1: Hypothesis generation

```
Input: match queue, $i$, $\boldsymbol{w}_{init}$, $\boldsymbol{\Sigma}_{w,init}$, $N_H$, $N_V$, $\tau_\phi$, $\chi$
Output: $\chi$
Create a hypothesis tree consisting of a single root node $V$ with assigned pose $\boldsymbol{w}_{init}$ and its covariance matrix $\boldsymbol{\Sigma}_{w,init}$.
Repeat until the match queue is empty or $|\chi| = N_H$ or the total number of nodes in the tree is $N_V$
    Take pair ($F$, $F'$) from the top of the match queue and remove it from the queue.
    For every node $V$ in the hypothesis tree
        If ($F$, $F'$) is compatible with $V$, then
            Perform measurement update of the pose $\boldsymbol{w}$ assigned to $V$ using EKF, where the parameters of $F$ and $F'$ are used to formulate innovation. The result is a new pose $\boldsymbol{w}'$ and covariance matrix $\boldsymbol{\Sigma}_{w'}$.
            Create a new node with assigned pair ($F$, $F'$), pose $\boldsymbol{w}'$ and matrix $\boldsymbol{\Sigma}_{w'}$. Append this node to the hypothesis tree.
            If the maximum eigenvalue of the rotational part of $\boldsymbol{\Sigma}_{w'}$ is $\leq \tau_\phi$, then
                Compare $\boldsymbol{w}'$ to all already generated 5 DoF hypotheses. If $\boldsymbol{w}'$ is not similar to any of them, then
                    Determine the sixth DoF by the procedure described in Section 5.4.
                    If the sixth DoF is successfully determined, then add hypothesis $H = (i, \boldsymbol{w})$ to the set $\chi$.
                end if
            end if
        end if
    end for
end repeat
```

The algorithm is performed for every local model M_i in the map. The inputs to the algorithm are the match queue, the local model index i, the initial pose estimate $\boldsymbol{w}_{init}$ and its covariance matrix $\boldsymbol{\Sigma}_{w,init}$ the maximum number of hypotheses per local model N_H, the maximum allowed number of nodes in the hypothesis tree N_V, the maximum orientation uncertainty τ_ϕ and set χ of hypotheses generated for the previously considered local models. For each local model, the algorithm adds a set of hypotheses to χ which is initially empty.

When the proposed approach is used for local pose tracking the initial pose estimate $\boldsymbol{w}_{init}$ can be obtained from the previous robot pose using odometry or inertial sensors. Since this paper considers global localization without any prior pose information, in all presented experiments $\boldsymbol{w}_{init}$ is set to $\mathbf{0}$ with a very high uncertainty, as described in Section 7.

5.4 Determining the sixth DoF

After a 5 DoF pose is generated, the hypothesis is completed by determining the last, sixth DoF. The sixth DoF of the estimated pose represents translation in the direction in which the translational uncertainty of the estimated pose $\boldsymbol{w}$ is highest. This direction is computed as the eigenvector corresponding to the greatest eigenvalue of the translational part of the covariance matrix $\boldsymbol{\Sigma}_w$. Let us denote this vector by $\tilde{\boldsymbol{t}}$ and the translation along this direction which needs to be found by l. The method applied to determine the sixth DoF is based on a voting scheme which includes both surface and line features. The process starts by forming pairs consisting of a local model feature and a scene feature transformed into the local model reference frame by $\boldsymbol{R}(\boldsymbol{\phi}')$ and $\boldsymbol{t}'$, where $\boldsymbol{\phi}'$ and $\boldsymbol{t}'$ represent orientation and position of the camera relative to the local model reference frame estimated in the process of generating the 5 DoF hypothesis, i.e. prior to determining the sixth DoF. A feature pair qualifies for voting if its elements satisfy the following conditions: (i) both features are oriented at a sufficiently large angle to $\tilde{\boldsymbol{t}}$ (In the experiments reported in Section 7, this angle is 45°), (ii) there is a value l such that the scene feature translated by $l \cdot \tilde{\boldsymbol{t}}$ overlaps sufficiently with the corresponding local model feature.

The votes are entered into an accumulator array of bins, where each bin corresponds to an interval of values l. Each feature pair makes a sequence of votes in the accumulator array centered in the bin corresponding to the value l which provides the best overlapping between features. A pair increases the value of the bin corresponding to the value l and the neighboring bins. The number of bins updated by a feature pair depends on the uncertainty of l. Since the feature parameters are measured with uncertainty, this uncertainty is used to compute the uncertainty of l. Hence, a value l provided by a feature pair is regarded as a random variable with variance σ_l. The accumulator array is updated by this feature pair by increasing the value of each bin by the amount

$$\frac{\gamma}{\sqrt{2\pi\sigma}}\exp\left(-\frac{\left(l-l_{bin}\right)^2}{2\sigma}\right), \qquad (11)$$

where l_{bin} is the central value of the bin and γ is a feature overlapping measure. Computation of l, σ and γ for surface and line features is explained in Appendix B and Appendix C respectively. An example is shown in Fig. 4. After all feature pairs have been processed, the bin with the maximum value is determined and the feature pair which contributes the most to this bin is selected for estimation of the sixth DoF. This selection is performed according to the formula

$$\left(F_{i*},F'_{j*}\right)=\underset{(F_i,F'_j)\in T_{last}}{\arg\max}\frac{\gamma_{ij}}{\sqrt{2\pi\sigma_{ij}}}\exp\left(-\frac{\left(l_{ij}-l_{max}\right)^2}{2\sigma_{ij}}\right), \qquad (12)$$

where T_{last} is the set of all feature pairs relevant for estimation of the sixth DoF, $\left(F_{i*},F'_{j*}\right)$ is the selected feature pair and l_{max} is the center of the bin with the highest accumulated vote value.

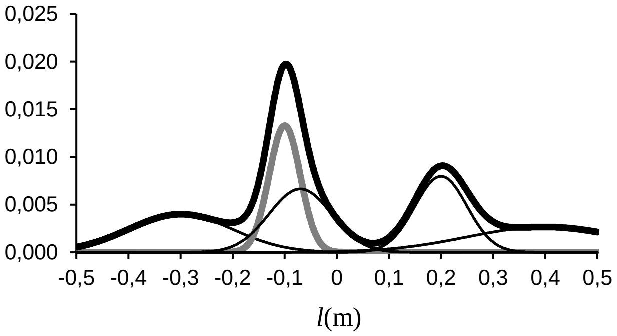


Fig. 4. Vote accumulation for the sixth DoF. The contributions of four matches are depicted by thin black lines, the contribution of the match selected for estimation of the sixth DoF is depicted by a thick gray line and the total of all accumulated votes is depicted by a thick black line.

Finally, the sixth DoF is estimated by EKF update of the pose $\mathbf{w}'=\left[\boldsymbol{\phi}' \,\vdots\, \mathbf{t}'\right]$ using the measurement provided by the selected feature pair.

6. Selecting the best hypothesis

The hypothesis generation stage described in Section 5 results in a set of hypotheses χ. The hypothesis evaluation consists in comparing the currently acquired 3D point cloud with the 3D point cloud corresponding to the hypothesis model and selecting the best hypothesis according to a particular measure of similarity between these two point clouds. The approach proposed in this section is to represent each surface segment by a set of samples and to match the scene samples to local model samples. The result of this matching is classification of each sample to one of the following four classes: *matched*, *occluded*, *transparent* and *invisible* as described in Section 6.2. A local model sample is classified as *transparent* if its projection onto the scene depth image occludes samples in that image. It is assumed that transparent surfaces cannot be detected and consequently the surface segment which has transparent samples is not present in the currently observed scene. This surface segment is then classified as *dynamic* since it is removed from the location represented by the local model after map building. A scene sample is classified as *transparent* if it occludes local model samples projected onto the scene depth image. Analogously, a scene surface segment whose sample is classified as *transparent* is classified as *dynamic* since it appeared after map building. A surface segment is classified as *matched* if it is not *dynamic* and at least N_s of its samples are classified as *matched*. The matched surface segments participate in a probabilistic decision process which assigns a probability to every hypothesis. Each pair of matched surface segments increases the probability of the evaluated hypothesis. The hypothesis with the highest probability is selected as the final solution.

Performing the described hypothesis evaluation procedure for each generated hypothesis would be time consuming. Therefore, before applying this procedure, pruning of the generated hypotheses is performed. The maximum consensus set of all generated hypotheses is determined, and only the hypotheses whose consensus set is close to the maximum consensus set are considered for further hypothesis evaluation. The hypothesis consensus set is determined as follows. Each scene surface segment is transformed to the hypothesis model coordinate system using the hypothesis pose and matched to all the local model surface segments according to the coplanarity criterion and overlapping criterion described in Section 5.2. The matched scene surface segments represent the hypothesis consensus set. A hypothesis consensus set is considered to be sufficiently close to the maximum consensus set if its size is less than the size of the maximum consensus set for at most 2 elements.

The following subsections provide details of the proposed hypothesis selection approach.

6.1 Data driven surface sampling

A common approach to reducing the amount of data provided by a complex sensor such as a camera is to apply sampling on a uniform grid. An image is segmented into rectangular regions and each region is represented by a single sample, as shown in Fig. 5(c). A sample can be simply a point, e.g. the center of a region, but it can also be a data structure describing the properties of the entire region. The image region represented by a sample is referred to in the following as a *sample window*. An image obtained by a 3D camera consists of pixels representing the surfaces detected in the observed scene and pixels without assigned depth. Each planar surface is represented in the image by a connected set of pixels. An example of image segmentation into regions corresponding to planar surface segment is shown in Fig. 5(b). The surface sampling approach proposed herein is to sample these regions and assign a set of samples to each of the relevant planar surfaces.

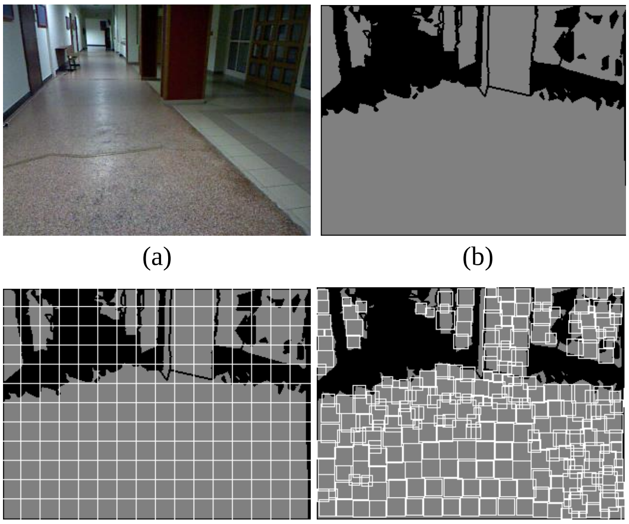


Fig. 5. (a) RGB-image; (b) image regions corresponding to planar surface segments represented by gray color; (c) uniform sampling; (d) data driven sampling.

The desirable properties of the sampling method are given in the following:

1. Each sample window is completely contained inside an image region representing a planar surface in the scene. Thereby, the data associated with the sample is restricted to a single surface.
2. All pixels representing relevant planar surfaces detected in the scene are covered by sample windows.
3. Overlapping between the adjacent sample windows should be low, thereby reducing the redundancy in the information encoded by the samples.
4. The size of the sample windows should be inside specified limits. Very small sample windows should be avoided since they cover a small part of the scene, while taking approximately the same amount of processing time as the other samples. Very large sample windows, however, when transformed between two model views for the comparison purpose, can be significantly distorted by the perspective transformation. These large deviations from their original rectangular shape complicate the evaluation of the overlapping of the sample windows between the compared models.
5. The sampling process should be as fast as possible.

Regular sampling on a uniform grid has all the properties stated above except the first one. A sampling approach which meets all the requirements to a great extent, yet not perfectly, is proposed in the following. It is based on the Voronoi diagram as a means of selecting pixels in the middle of a particular image region. The Voronoi diagram of an image region with Chebyshev distance as metric can be computed using Algorithm 2.

Algorithm 2: *h*-map update

```
Input: $Q$, $h$
Output: $h$
Repeat until $Q$ is empty
    Take $\boldsymbol{m}$ from the bottom of $Q$
    For every $\boldsymbol{m}'$ from the 8-neighborhood of $\boldsymbol{m}$
        If $h(\boldsymbol{m}') > h(\boldsymbol{m}) + 1$ then
            $h(\boldsymbol{m}') \leftarrow h(\boldsymbol{m}) + 1$
            Put $\boldsymbol{m}'$ at the top of $Q$
        end if
    end for
end repeat
```

This is a region growing algorithm referred to in the following as *h-map update* which constructs a mapping $h(\boldsymbol{m})$ assigning a value h to each image point $\boldsymbol{m}$. Given an image region, Algorithm 2 can be used to compute the Chebyshev distance of all points belonging to this region from the region boundary. Let Q be a FIFO queue containing all image points $\boldsymbol{m}$ which do not belong to the considered region and let $h(\boldsymbol{m})$ be set to 0 for all these points and to ∞ for all pixels belonging to the considered region. After executing Algorithm 2, values $h(\boldsymbol{m})$ represent Chebyshev distance of all pixels belonging to the considered region to the region boundary. This process can be applied in parallel to all image regions. A 1D example is shown in Fig. 6, which can be regarded as e.g. an image row segmented into three regions corresponding to three planar surfaces in a scene.

The Chebyshev distances for all points inside the regions shown in Fig. 6(a) are represented by the h-map depicted in Fig. 6(b). The local maxima of $h(\boldsymbol{m})$ represent the Voronoi diagram of the region. Assuming that image regions represent planar surfaces in the scene, the pixel $\boldsymbol{m}^*$ corresponding to the highest value $h(\boldsymbol{m}^*)$ is a reasonable choice for a sample, since it is the center of the largest square window completely contained inside a region. The next sample can be obtained by setting $h(\boldsymbol{m}^*)$ to 0, which has the effect of removing point $\boldsymbol{m}^*$ from the sampled region, and executing Algorithm 2 in order to recompute the Chebyshev distance map. The h-map after removing the point denoted by a circle in Fig. 6(b) is shown in Fig. 6(c). By repeating this procedure until $\max_{\boldsymbol{m}} h(\boldsymbol{m}) < h_{max}$ a set of samples is obtained, as shown in Fig. 6(d), where sample points are denoted by squares positioned in the local minima of $h(\boldsymbol{m})$. The sample window for each sample can be determined by the local maxima of $h(\boldsymbol{m})$ in the neighborhood of the sample point. The described method is represented by Algorithm 3, where h_{min} = 0 while $b(\boldsymbol{m})$ represents a mapping which assigns each pixel $\boldsymbol{m}$ value 1 if it belongs to a region and 0 otherwise. Fig. 5(a) represents a visualization of the mapping $b(\boldsymbol{m})$.

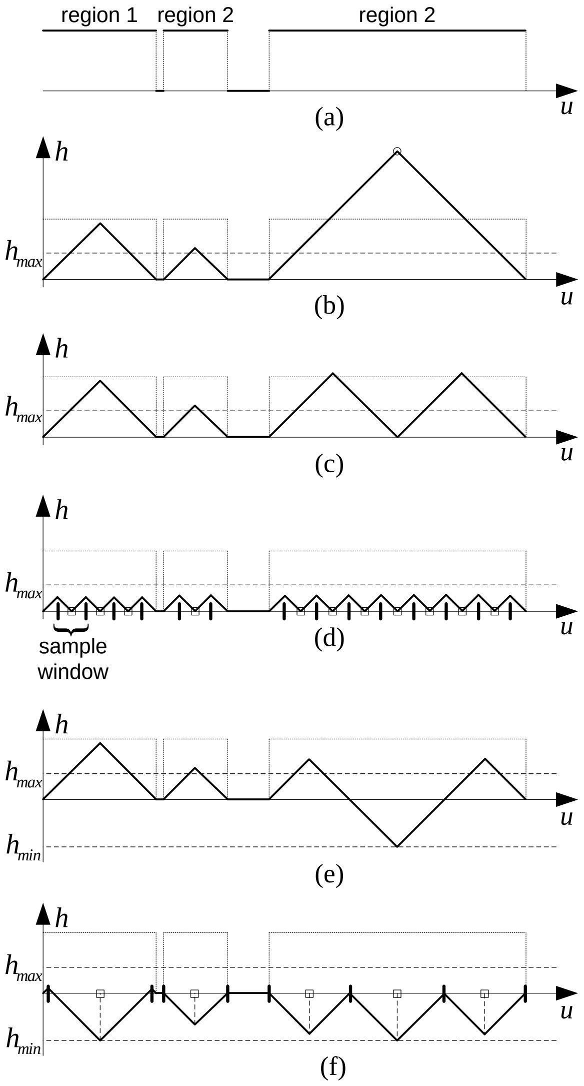


Fig. 6. (a) An image row segmented into three regions. (b) Chebyshev distance assigned to all points. (c) Chebyshev distance after removing the first sample from the region. (d) Final sampling result for $h_{min} = 0$. Sample centers are represented by squares and sample windows by thick vertical lines. (e) h-map after selection of the first sample for $h_{min} < 0$. (f) Final sampling result for $h_{min} < 0$.

Algorithm 3: Data driven region sampling

```
Input: $b$, $h_{min}$, $h_{max}$
Output: $\Omega$
$\Omega \leftarrow \varnothing$
Set $h(\boldsymbol{m})$ for all pixels $\boldsymbol{m}$ to $\infty$.
Form empty queue $Q$.
Put all pixels $\boldsymbol{m}$ for which $b(\boldsymbol{m}) = 0$ as well as those
    that lie on a region boundary into $Q$ and set their
    value $h(\boldsymbol{m})$ to 0.
Run h-map update algorithm.
Repeat
    $\boldsymbol{m} \leftarrow \arg\max_{\boldsymbol{m}'} h(\boldsymbol{m}')$
    If $h(\boldsymbol{m}) < h_{max}$, then stop the procedure.
    Insert $\boldsymbol{m}$ into $\Omega$.
    $h(\boldsymbol{m}) \leftarrow \max\{-h(\boldsymbol{m}), h_{min}\}$
    Empty $Q$ and put $\boldsymbol{m}$ into it.
    Run h-map update algorithm.
end repeat
Return $\Omega$.
```

Although this method produces the sample windows positioned inside the sampled region, its drawback is that it results in a large number of samples if h_{max} is set to a small value, while by increasing h_{max} some relatively small surfaces are not represented by samples. A better result is obtained if h_{min} is set to some negative value, as illustrated in Fig. 6. The h-map shown in Fig. 6(e) is obtained after removing the sample denoted by a circle in Fig. 6(b). The final result is depicted in Fig. 6(f). The minimum width of an image region which is sampled is determined by the parameter h_{max} and the maximum sample window size is $2(h_{max} - h_{min})$. Furthermore, a property of the proposed method is that the inside of a large region is represented by large sample windows, while narrow regions are represented by small sample windows. By this adaptive sampling strategy, the total number of samples is reduced without a significant loss of information. An example of the proposed data driven region sampling is shown in Fig. 5(d).

Each sample represents a fragment of a surface defined by the sample window. A sample consists of the following data: $\boldsymbol{m} = [u, v]^T$ – image point representing the sample centre, d – depth of the sample center, $\boldsymbol{p} = [x, y, z]^T$ – position of the sample center point relative to the camera or local model reference frame, $\sigma_{\boldsymbol{p}}$ – maximum eigenvalue of the covariance matrix which describes the uncertainty of $\boldsymbol{p}$, $\boldsymbol{v}$ –eigenvector corresponding to that eigenvalue and w – sample window size. The covariance matrix describing the uncertainty of the sample center point is determined using a triangulation uncertainty model analogous to the one proposed in (Matthies and Shafer, 1987). This matrix is reduced to its maximum principal axis for computational efficiency.

Note that although the sample windows are square, the surface patches they represent are not since the projection of a square onto an arbitrary surface is not square in general, as illustrated in Fig. 7.

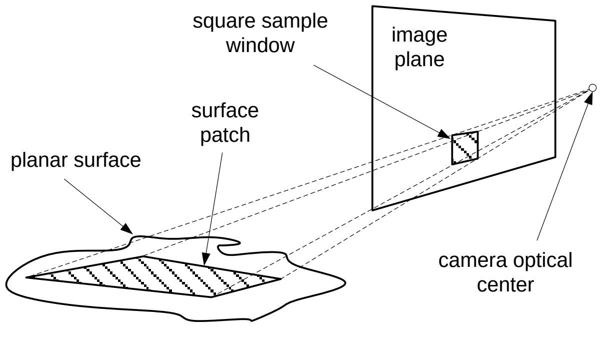


Fig. 7. Surface patch corresponding to a sample window.

6.2 Sample matching

Comparison of a scene model to a hypothesis model is performed by matching surface samples. The surface samples of a hypothesis model are transformed into the coordinate system of the scene model, projected onto the camera image and matched to the scene surface samples with respect to their position and orientation.

A hypothesis model sample is matched to a scene sample if there is sufficient overlapping between their windows, if they have similar depths and if their corresponding surface patches have similar orientations. The overlapping between a hypothesis model sample and a scene sample is considered sufficient if the center of each sample is inside the window of the other sample, as shown in Fig. 8. The details are given in Appendix D.

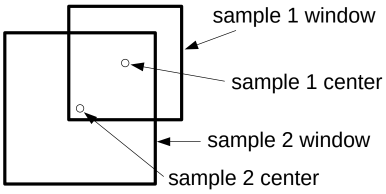


Fig. 8. Overlapping samples.

A scene sample and a local model sample which overlap with each other can be in one of the following 2 relations: (i) if their depths and orientations are sufficiently similar they match or (ii) if the depth of one of them is significantly less than the depth of the other, the first one occludes the other. According to these relations, each sample is classified in one of three classes: (i) if a local model sample matches at least one of the scene samples, then it is classified as *matched*, (ii) if a local model sample does not match any of the scene samples, but it is occluded by at least one scene sample, then it is classified as *occluded* and (iii) if a local model sample is neither matched nor it is occluded, but it occludes at least one scene sample, then it is classified as *transparent*. Scene samples are classified analogously. If a sample does not overlap with any sample from another sample set, then it is classified as *invisible*.

Since transparent samples of a model surface segment indicate that this segment is not present in the scene, thereby causing a rejection of the segment from further processing, as explained at the beginning of Section 6, the hypothesis evaluation algorithm must have mechanisms which avoid misclassification of segments. Let us consider one situation related to this problem. A depth image obtained by a 3D camera often contains regions with undefined depth due to limited range of the sensor, specular reflections or materials which do not reflect the structured light or laser beams emitted by the sensor. For these reasons, a 3D camera can fail to detect a part of a surface in a scene from a particular view, while the same surface is completely reconstructed from another view. Thus, it can happen that e.g. a local model sample is projected onto an undefined region in the scene image, although the pose hypothesis has correctly registered the acquired point cloud with the local model. In this case, the considered sample is not matched to a scene sample representing the same surface. However, because of a small error in the estimated pose, a small fragment of the considered sample can overlap with a scene sample representing another surface, resulting in misclassification of the considered sample. If the hypothesis evaluation algorithm was designed to reject every surface segment which has at least one transparent sample, a significant number of dominant segments would be falsely rejected. In order to reduce the probability of misclassification and false rejection of segments, it is required that the percentage of visible surface samples which are classified as transparent must exceed a predefined threshold in order for the segment to be classified as *dynamic*. In the experiments presented in Section 7, this threshold is set to 20%.

6.3 Estimating the hypothesis probability

In this section, a method for estimating the probability of the hypotheses, generated as described in Section 5, is described. A straightforward approach to estimating a hypothesis probability would be to use the independent beam model (IBM) (Thrun et al., 2005) by assigning the probability to each sample match and then to estimate the hypothesis probability by multiplying the probabilities of individual sample matches. This approach gives advantage to hypotheses which match large number of samples, which is reasonable. Nevertheless, in the case of matching local models which cover the scene image only partially, the number of matched samples does not always reflect the probability of a hypothesis. Let us consider the example shown in Fig. 5. Since the floor surface is represented by the majority of samples, the IBM approach would give advantage to a false hypothesis which matches a single local model surface to the floor surface, over a correct hypothesis which matches model surfaces to many other scene surfaces but not the floor surface. Some other drawbacks of IBM are discussed in (Krainin et al., 2012).

In order to make our approach suitable for matching partially overlapping point clouds, we use an approach which matches entire surface segments instead of matching individual samples. The idea is to give advantage to hypotheses which match complex structures, which are not probable to appear accidentally. The proposed approach is described in the following.

Let $Z = \{F_1, F_2, ... \}$ be a set of all surface segments detected in a camera image. The conditional probability of a hypothesis H_k given Z can be computed by Bayes' rule

$$P(H_k \mid Z) = \frac{p(Z \mid H_k) P(H_k)}{p(Z)}, \tag{13}$$

where $p(Z \mid H_k)$ is the conditional probability density function (PDF) of detecting a particular set of surface segments if hypothesis H_k is correct, $P(H_k)$ is the prior probability of H_k and $p(Z)$ is prior PDF of obtaining a particular set of surface segments. Assuming that one of the hypotheses from a hypothesis set $\tilde{\chi} = \{H_1, H_2, \ldots\}$ is correct and that only one of them can be correct,

$$P(H_k \mid Z) = \frac{p(Z \mid H_k) P(H_k)}{\sum\limits_{H_{k'} \in \tilde{\chi}} p(Z \mid H_{k'}) P(H_{k'})}. \tag{14}$$

Assuming that prior probability of all hypotheses is equal,

$$P\left(H_k \mid Z\right) = \frac{p\left(Z \mid H_k\right)}{\sum\limits_{H_{k'} \in \tilde{\chi}} p\left(Z \mid H_{k'}\right)}. \qquad (15)$$

In the context of place recognition for loop closure in a model building process, the assumption that one and only one hypothesis is correct is not fulfilled, since it implies that the robot never visits a place not already recorded in the map. This restricts the problem to robot localization given a previously built environment model. In our future work we will adapt the proposed method to the cases where a currently observed scene is not covered by the model, e.g. by adding an additional term to the denominator on the right side of (14), which models the probability that a particular set of surface segments is detected and none of the generated hypotheses is correct.

Let us assign to each hypothesis H_k a correspondence vector $\boldsymbol{c}_k$ defined as in (Thrun et al., 2005). The *i*th element of this vector is $c_{ki} = j$ if the scene surface segment F_i corresponds to the local model surface segment F'_j or $c_{ki} = 0$ if F_i does not correspond to any surface segment. In order to estimate the conditional PDF $p(Z \mid H_k)$, we apply *independent surface model*, i.e. we assume that the errors in the parameter measurements of the surface segments are mutually independent. Then, the conditional PDF of obtaining particular parameters of all detected surface segments if the hypothesis H_k is correct can be written as

$$p\left(Z \mid H_k\right) = \prod_{c_{ki} \neq 0} p\left(F_i, F'_j \mid c_{ki} = j, \boldsymbol{w}_k\right) \prod_{c_{ki} = 0} p\left(F_i \mid c_{ki} = 0\right)$$

(16)

The conditional PDF for obtaining particular parameters of the segments F_i and F'_j will be obtained by feature detection, assuming that the pose $\boldsymbol{w}_k$ of the camera relative to the local model and a correspondence vector $\boldsymbol{c}_k$ are correct, can be approximated by the probability that F_i transformed into the local model reference frame using $\boldsymbol{w}_k$ shares the same supporting plane with F'_j. Let us assume that both F_i and F'_j lie in a plane with normal $\boldsymbol{n}$ and offset ρ. In that case, the PDF of obtaining particular measurements of the supporting plane parameters of these two surface segments is given by

$$\begin{gathered} p\left(F_i, F'_j \mid c_{ki} = j, \boldsymbol{w}_k\right) = \\ \int\limits_{\boldsymbol{n}} \int\limits_{\rho} p\left(F_i \mid \boldsymbol{n}, \rho, \boldsymbol{w}_k\right) \cdot p\left(F'_j \mid \boldsymbol{n}, \rho\right) \cdot d\boldsymbol{n} \cdot d\rho . \end{gathered} \qquad (17)$$

where $p\left(F_i \mid \boldsymbol{n}, \rho, \boldsymbol{w}_k\right)$ denotes the PDF of obtaining particular measurement of the supporting plane parameters of F_i under the assumption that F_i lies in a plane defined by $\boldsymbol{n}$ and ρ. PDF $p\left(F'_j \mid \boldsymbol{n}, \rho\right)$ is defined analogously.

Let us now consider the case of a scene surface segment F_i which is not matched to any local model surface segment, i.e. for $c_{ki} = 0$. Assuming uniform distribution of the measured supporting plane normal of F_i over the space of all unit vectors, it follows that the prior PDF of obtaining a particular normal vector is

$$p\left(\boldsymbol{n}_i \mid c_{ki} = 0\right) = \frac{1}{4\pi}, \qquad (18)$$

since 4π is the area of the unit sphere. Without loss of generality, it can be assumed that all surfaces detected by a 3D camera have a normal with an angle less than 90° relative to the camera optical rays, i.e. for any detected surface the surface with the oppositely directed normal cannot be detected. Hence, it can be assumed that the normals of the detected surfaces are uniformly distributed over approximately half of the unit vector space, i.e.

$$p\left(\boldsymbol{n}_i \mid c_{ki} = 0\right) = \frac{1}{2\pi}. \qquad (19)$$

The prior PDF of measurement of the plane parameter ρ is rather difficult to estimate. Therefore, we decided to consider only plane normals in our probabilistic model. Hence, the following approximations are used

$$p\left(F_i, F'_j \mid c_{ki} = j, \boldsymbol{w}_k\right) \approx \int\limits_{\boldsymbol{n}} p\left(F_i \mid \boldsymbol{n}, \boldsymbol{w}_k\right) \cdot p\left(F'_j \mid \boldsymbol{n}\right) \cdot d\boldsymbol{n} \qquad (20)$$

$$p\left(F_i \mid c_{ki} = 0\right) = \frac{1}{2\pi}. \qquad (21)$$

In order to compute the PDFs on the right side of (20), both surface segments F_i and F'_j are transformed into the coordinate system $S_{F'}$ assigned to F'_j as described in Section 4.1. Let $\boldsymbol{s}$ be the vector formed by the first two components of a plane normal $\boldsymbol{n}$ represented in $S_{F'}$. This vector can be used to represent the deviation of the plane normal $\boldsymbol{n}$ from the measured plane normal of the surface segment F'_j. Therefore,

$$p\left(F_i \mid \boldsymbol{n}, \boldsymbol{w}_k\right) = p\left(F_i \mid \boldsymbol{s}, \boldsymbol{\phi}_k\right). \qquad (22)$$

Using this approach, the deviation of the normal of the supporting plane of F_i transformed into $S_{F'}$ can be described by vector ${}^{F'}\boldsymbol{s}_i$ representing the first two components of this normal. This vector can be computed by

$${}^{F'}\boldsymbol{s}_i = \frac{1}{\sqrt{\boldsymbol{s}_i^T \boldsymbol{s}_i + 1}} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} {}^{M}\boldsymbol{R}_{F'}^T \boldsymbol{R}\left(\boldsymbol{\phi}_k\right) {}^{C}\boldsymbol{R}_F \begin{bmatrix} \boldsymbol{s}_i \\ \hdashline 1 \end{bmatrix}, \qquad (23)$$

where $\boldsymbol{s}_i = [s_{x,i},\ s_{y,i}]^T$. The equation (23) follows from (2) and from definitions of the rotation matrices ${}^{C}\boldsymbol{R}_F$ and ${}^{M}\boldsymbol{R}_{F'}$ given in Section 4.1. The uncertainty of the normal of F_i can be described by a normal distribution with mean

$${}^{F'}\hat{\boldsymbol{s}}_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} {}^{M}\boldsymbol{R}_{F'}^T \boldsymbol{R}\left(\boldsymbol{\phi}_k\right) {}^{C}\boldsymbol{R}_F \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad (24)$$

and covariance matrix ${}^{F'}\boldsymbol{\Sigma}_{\boldsymbol{s},i}$. The covariance matrix ${}^{F'}\boldsymbol{\Sigma}_{\boldsymbol{s},i}$ can be obtained by transforming covariance matrix $\boldsymbol{\Sigma}_{s,i}$ = diag([$\sigma_{sx,i}$, $\sigma_{sy,i}$]) into $S_{F'}$, where $\sigma_{sx,i}$, $\sigma_{sy,i}$ are variances introduced in Section 4.1. This transformation can be performed by

$$
{}^{F'}\boldsymbol{\Sigma}_{\boldsymbol{s},i} = \boldsymbol{J}\left(F_i, F'_j, \boldsymbol{\phi}_k\right)\boldsymbol{\Sigma}_{\boldsymbol{s},i}\boldsymbol{J}^T\left(F_i, F'_j, \boldsymbol{\phi}_k\right), \tag{25}
$$

where

$$
\boldsymbol{J}\left(F_i, F'_j, \boldsymbol{\phi}_k\right) = \frac{\partial\, {}^{F'}\boldsymbol{s}_i}{\partial \boldsymbol{s}_i}. \tag{26}
$$

Now, having the mean and covariance matrix of ${}^{F'}\boldsymbol{s}_i$, the conditional PDF of measuring particular value ${}^{F'}\hat{\boldsymbol{s}}_i$ given the actual normal of its supporting plane $\boldsymbol{n}$ and pose $\boldsymbol{w}_k$ can be computed by

$$
\begin{aligned}
p\left(F_i \mid \boldsymbol{s}, \boldsymbol{\phi}_k\right) = {} & \frac{1}{2\pi\sqrt{\det\left({}^{F'}\boldsymbol{\Sigma}_{\boldsymbol{s},i}\right)}} \cdot \\
& \cdot \exp\left(-\frac{1}{2}\left({}^{F'}\hat{\boldsymbol{s}}_i - \boldsymbol{s}\right)^T {}^{F'}\boldsymbol{\Sigma}_{\boldsymbol{s},i}^{-1}\left({}^{F'}\hat{\boldsymbol{s}}_i - \boldsymbol{s}\right)\right)
\end{aligned} \tag{27}
$$

Analogously, the uncertainty of the normal of F'_j represented in $S_{F'}$ can be described by a random vector $\boldsymbol{s}'_j$ with 0 mean and a covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{s}',j}$. Consequently, the conditional PDF of measuring value $\boldsymbol{s}'_j = \boldsymbol{0}$ given the actual normal of its supporting plane $\boldsymbol{n}$ and pose $\boldsymbol{w}_k$ can be computed by

$$
p\left(F'_j \mid \boldsymbol{s}\right) = \frac{1}{2\pi\sqrt{\det\left(\boldsymbol{\Sigma}_{\boldsymbol{s}',j}\right)}} \exp\left(-\frac{1}{2}\boldsymbol{s}^T \boldsymbol{\Sigma}_{\boldsymbol{s}',j}^{-1}\boldsymbol{s}\right). \tag{28}
$$

By substituting the PDFs on the right side of (20) with (27) and (28) and integrating the obtained product, the following is obtained

$$
\begin{aligned}
& p\left(F_i, F'_j \mid c_{ki} = j, \boldsymbol{w}\right) \approx \\
& \frac{1}{2\pi\sqrt{\det\left({}^{F'}\boldsymbol{\Sigma}_{\boldsymbol{s},i} + \boldsymbol{\Sigma}_{\boldsymbol{s}',j}\right)}} \exp\left(-\frac{L_{ij}}{2}\right),
\end{aligned} \tag{29}
$$

where

$$
L_{ij} = \left({}^{F'}\hat{\boldsymbol{s}}_i - \hat{\boldsymbol{s}}_{ij}\right)^T {}^{F'}\boldsymbol{\Sigma}_{\boldsymbol{s},i}^{-1}\left({}^{F'}\hat{\boldsymbol{s}}_i - \hat{\boldsymbol{s}}_{ij}\right) + \hat{\boldsymbol{s}}_{ij}^T \boldsymbol{\Sigma}'^{-1}_{\boldsymbol{s}',j}\hat{\boldsymbol{s}}_{ij}, \tag{30}
$$

$$
\hat{\boldsymbol{s}}_{ij} = \left({}^{F'}\boldsymbol{\Sigma}_{\boldsymbol{s},i}^{-1} + \boldsymbol{\Sigma}'^{-1}_{\boldsymbol{s}',j}\right)^{-1} {}^{F'}\boldsymbol{\Sigma}_{\boldsymbol{s},i}^{-1}\, {}^{F'}\hat{\boldsymbol{s}}_i. \tag{31}
$$

The derivation of (29) is given in Appendix E.

Finally, the probabilities of all hypotheses H_k given a set of surface segments detected in the scene can be estimated as follows. For each pair (F_i, F'_j), vector ${}^{F'}\hat{\boldsymbol{s}}_i$ is computed by (24) and it's corresponding covariance matrix ${}^{F'}\boldsymbol{\Sigma}_{\boldsymbol{s},i}$ by (25) and (26) using the parameters of F_i and F'_j together with the estimated orientation $\boldsymbol{\phi}_k$. Then, $\hat{\boldsymbol{s}}_{ij}$ is computed by (31) and substituted into (30) in order to obtain L_{ij}. The obtained value L_{ij} is then used to compute $p\left(F_i, F'_j \mid c_{ki} = j, \boldsymbol{w}\right)$ by (29). The PDFs $p\left(F_i, F'_j \mid c_{ki} = j, \boldsymbol{w}\right)$ computed for each pair (F_i, F'_j) are substituted into (16) together with PDFs (21) of scene segments which are not matched. As a result, PDF $p\left(Z \mid H_k\right)$ is obtained for each hypothesis H_k. The final probability estimate for all hypotheses is obtained by normalization (15).

In order to meet the assumption that two hypotheses cannot be correct, only one hypothesis for a local model should be generated and two local models should not represent the same location in the map. Since several hypotheses are commonly generated for each local model, the first constraint can be met by forming the set $\tilde{\chi}$ only of hypotheses with the highest value $p\left(Z \mid H_k\right)$ for each local model. The second constraint is in practice difficult to assure. Nevertheless, selection of the best hypothesis is influenced only by the PDF values $p\left(Z \mid H_k\right)$, while the purpose of normalization (15) is just to obtain probabilities from the interval [0, 1].

The proposed hypothesis probability estimation method also assumes unique correspondences between surface segments. However, a scene surface segment can, in general, be matched to more than one local model surface segment. In order to meet the aforementioned assumption, for every hypothesis a list of all pairs (F_i, F'_j) is formed sorted by the number of shared sample matches in descending order. Then the pairs are taken one by one starting from the top of the list and used in the hypothesis probability estimation process. After a pair (F_i, F'_j) is removed from the list, all pairs containing either F_i or F'_j are removed from the list.

7. Experimental evaluation

In this section, experimental evaluation of the proposed approach is considered. We implemented our system in C++ programming language using OpenCV library (Bradski and Kaehler 2008) and execute it on a standard PC. Two experiments were performed. The first experiment tested the robustness of our approach to changes in the environment, while the objective of the second experiment was to provide an insight into the localization accuracy which can be achieved by the proposed approach.

In both experiments, the initial pose estimate $\boldsymbol{w}_{init}$ in the hypothesis generation step is set to **0**. The uncertainty of this pose information, i. e. the covariance matrix $\boldsymbol{\Sigma}_{w,init}$, is computed using an uncertainty model of the camera mounted on a mobile robot Pioneer 3DX rolling on a bumpy horizontal surface. Assuming that the global coordinate system S_0 assigned to the robot's environment is defined in such a way that its z-axis is parallel to the gravity axis, the bumps on the floor surface are modelled by white noise of standard deviation σ_f = 0.005 m in z-direction. The uncertainty of the robot's position and orientation within the xy-plane of S_0 is modelled by normal distributions with standard deviations of 1 m and

20° respectively. The uncertainty of the camera orientation with respect to the gravity axis due to the bumps on the floor surface is also modelled by a normal distribution with a standard deviation of approximatelly 2.7° for the inclination angle and 1.6° for the rotation around the optical axis of the camera.

7.1 Robustness to changing environment

Robustness of the proposed approach to changes in the environment is tested using a benchmark dataset consisting of two sets of depth images acquired by a Microsoft Kinect sensor. The first is a *reference set* which is used to create the environment model, while the second is a *test set* containing depth images of the scenes covered by the reference set. The reference set consists of 45 depth images representing different locations in a public building and a private household. The test dataset contains 2165 depth images covering 22 of 45 reference scenes acquired under varying lighting conditions and with objects and people appearing and disappearing from the observed scenes. The test set is subdivided into 4 subsets covering 4 separately analyzed cases presented in Table 1. Several sample images from the considered dataset are shown in Fig. 9. Although only depth images are used by our system, RGB images are displayed for the visualization purpose.

Table 1 Test datasets

subset	different lighting conditions than in the corresponding reference image	dynamic objects
1	-	-
2	-	+
3	+	-
4	+	+

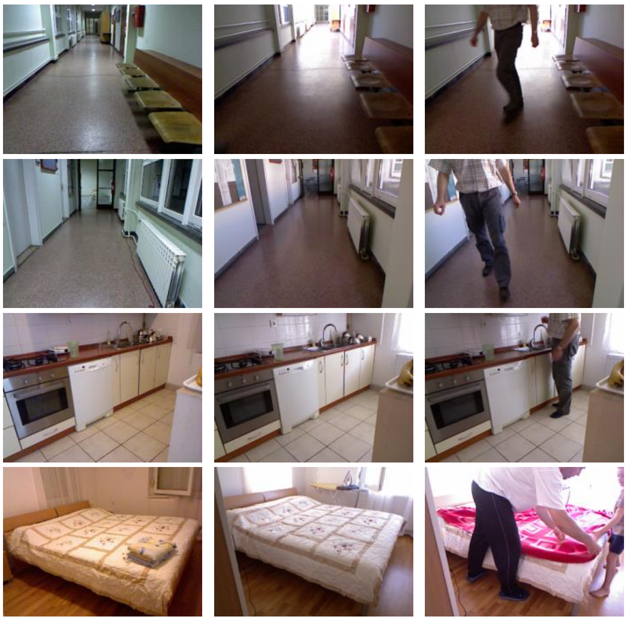


Fig. 9. Sample test images. Reference images are represented in the left column, images taken after the lighting conditions are changed are shown in the middle, while the images with a moving person are shown in the right column.

The place recognition results are visualized in Fig. 10, where surface samples are depicted by squares of different colors according to the classification described in Section 6. Notice how the samples representing the moving person are classified as transparent, thereby indicating the presence of a dynamic object. Many samples classified as invisible in the top right image in Fig. 10 indicate that only a relatively small part of the scene overlaps with the corresponding local model.

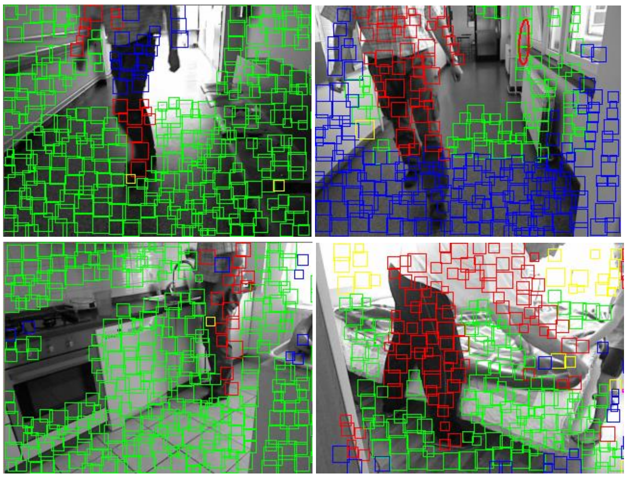


Fig. 10. Place recognition results. Matched surface samples are depicted by green squares, occluded by yellow squares, transparent by red squares and invisible by blue squares.

The approach presented in this paper is based on the approach proposed in (Cupec et al., 2012; Cupec et al., 2013). There are three main improvements to the original method.

1. In addition to the planar surface segments, line segment features are used in the hypothesis generation process.
2. Instead of approximating the surface segments by ellipsoids, each surface segment is represented by a set of samples, which should provide a better description of the surface shape and a novel probabilistic hypothesis evaluation approach is used to select the best hypothesis.
3. A mechanism for detection and rejection of dynamic surfaces is introduced.

In order to analyze the influence of all aforementioned steps, the experiments are performed with four variants of our algorithm:

1. the original algorithm presented in (Cupec et al., 2012);
2. the original algorithm with line segment features used in hypothesis generation step;
3. the novel algorithm with surface sampling, but without rejection of dynamic surfaces and
4. the complete novel algorithm.

We also performed experiments with an alternative hypothesis generation step, where we used PROSAC (Chum and Matas, 2005), an efficient variant of RANSAC, instead of the hypothesis generation approach based on the hypothesis tree, which is described in Section

5.3. Analogously to the hypothesis tree approach presented in Section 5.3, PROSAC exploits ranking of feature pairs according to an appropriate criterion. The criterion used in the experiments with PROSAC presented in the following is the information content factor, which is also used in the hypothesis tree approach. PROSAC is used to generate hypotheses using the initial surface segment pairs and determine the best hypothesis for each local model. The obtained hypotheses are then forwarded to the hypothesis evaluation step described in Section 6. All parameters of the PROSAC are set to the values used in (Chum and Matas, 2005). The variant of our place recognition approach with PROSAC-based hypothesis generation is referred to in the following as variant 5.

The performance of the proposed place recognition approach for the considered test dataset is presented in Table 2, where the results obtained by all five variants are given. Introduction of line features (variant 2) notably improves the performance of the original algorithm (variant 1) by allowing generation of hypothesis in cases where surfaces do not provide sufficient information for estimation of all 6DoF of the camera pose. The surface sampling step (variant 3) improves significantly the results for the subsets 3 and 4, while for sets 1 and 2 no improvement is achieved. Introduction of the dynamic surface rejection step (variant 4) resulted in slightly higher recognition rate for subsets 2 – 4. The contribution of this step is not so obvious because it helps in resolving situations where a surface in an observed scene appears exactly at the place of a surface of a falsely matched model. In such situations, information about the surface shape can help in distinguishing between the correct and false hypothesis. Such cases were not very frequent in the experiments presented in this paper. However, the probability of such cases is expected to rise with the number of local models. The performance data measured over the entire dataset (the last row of Table 2) show that the introduction of each step improves the algorithm performance.

Table 2 Performance of five variants of the proposed approach. Symbol # denotes the number of correctly matched images and % their percentage.

		correct recognitions							
		variant 1		variant 2		variant 3		variant 4	
subset	total	#	%	#	%	#	%	#	%
1	68	67	99	67	99	63	93	63	93
2	676	639	95	650	96	597	88	606	90
3	246	138	56	150	61	199	81	204	83
4	1175	616	52	711	61	936	80	984	84
Σ	2165	1460	67	1578	73	1795	83	1857	86

		variant 5	
subset	total	#	%
1	68	62	91
2	676	582	86
3	246	100	41
4	1175	527	45
Σ	2165	1271	59

The results obtained by the PROSAC-based hypothesis generation (variant 5) are similar to those obtained by the hypothesis tree approach for the subsets 1 and 2, while for the subsets 3 and 4, the percentage of correct recognitions is much lower. A drawback of the applied implementation of PROSAC is that it does not use line segments. Another drawback which is the inherent property of this method is that it returns only a single hypothesis for each local model.

For comparison purposes, OpenFABMAP (Glover et al., 2012) and DLoopDetector (Galvez-Lopez and Tardos, 2012) were applied in the same way as our method to the same dataset. Both methods are based on BoW principle, however DLoopDetector also performs a geometric consistency check (GCC) based on epipolar geometry in order to improve precision. The main purpose of making a comparison with both FAB-MAP and DLoopDetector is to analyze the influence of GCC to the algorithm performance.

Both FAB-MAP and DLoopDetector were configured to perform localization without map building. All other parameters of FAB-MAP were set to their default values that came with the source code. DLoopDetector was configured as follows. Since place recognition from a single image is considered in this paper, the temporal consistency of DLoopDetector was turned off by setting the number of temporally consistent matches to 0. FLANN method was used for determining feature correspondences. The efficiency of DLoopDetector can be improved by appropriately selecting the rejection threshold, whose purpose is to reduce the number of matches for GCC to be performed on the most reliable ones. Nevertheless, in order to provide a fair comparison of the considered algorithms according to their precision and recall properties, the rejection threshold was set to 0, thereby allowing the GCC for all matches. Furthermore, although the authors in (Galvez-Lopez and Tardos, 2012) favor using BRIEF descriptor with FAST detector as an optimal choice between speed and precision, SURF64 descriptor was used in all experiments presented in this paper, since our experiments showed that BRIEF has significantly lower precision than SURF descriptor especially in the cases where illumination conditions in the test images are different from the map images.

For both FAB-MAP and DLoopDetector we performed experiments with the vocabularies provided with the program code as well as with a vocabularies we created from the considered reference images using their programs. For both algorithms, the vocabulary that we created showed better results than the one provided with the program code. Therefore, in this section, the results obtained with the vocabularies created from our reference images are presented. The results of the experiments performed with FAB-MAP and DLoopDetector are presented in Table 3 together with the results obtained by the proposed approach.

Assuming that the considered place recognition algorithms return multiple solutions whose estimated probability exceeds a specified threshold, they can be compared using their precision-recall curves. The precision-recall curves of the proposed approach, FAB-MAP and DLoopDetector are depicted in Fig. 11.

Table 3 Performance of the proposed approach, FAB-MAP and DLoopDetector. Symbol # denotes the number of correctly matched images and % their percentage.

		correct recognitions					
		proposed approach		FAB-MAP		DLoopDetector	
subset	total	#	%	#	%	#	%
1	68	63	92.6	66	97.1	56	82.4
2	676	606	89.6	574	84.9	577	85.4
3	246	204	82.9	114	46.3	6	2.4
4	1175	984	83.7	453	38.6	49	4.2
Σ	2165	1857	85.8	1207	55.8	688	31.8

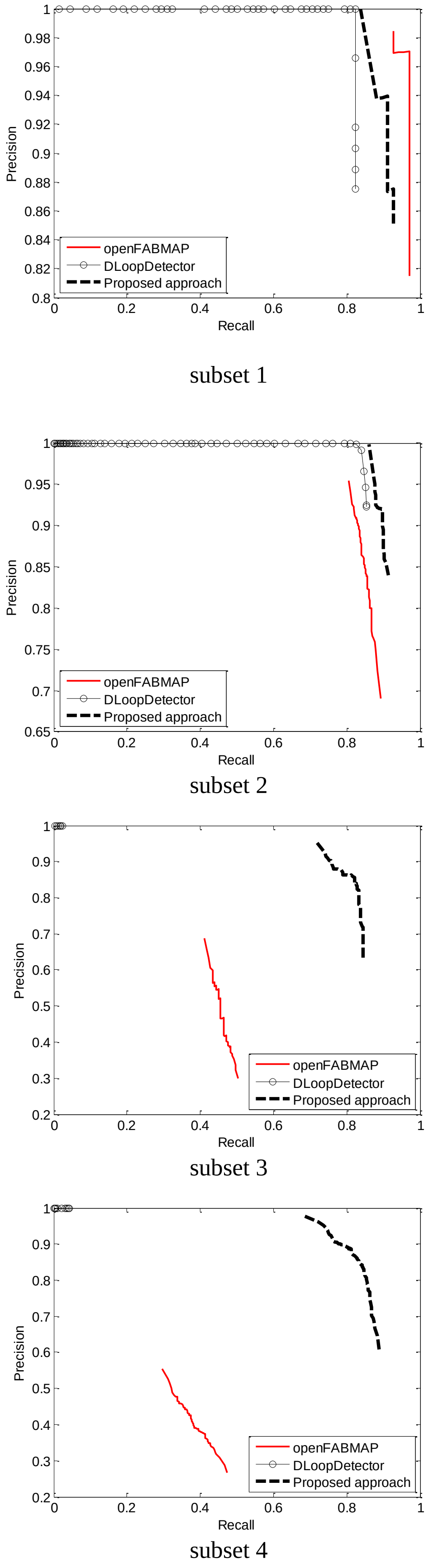
The presented results show that the proposed approach outperforms FAB-MAP in the case of environment changes. In the case where the lighting conditions resemble those of the reference images and there are no dynamic objects, FAB-MAP performs slightly better.

DLoopDetector is designed primarily for loop closing and it has a very high precision. All best scored hypotheses of DLoopDetector which passed GCC, were correct in the case of all four image subsets. However, for the images taken in different lighting conditions than the reference images, a very low number of hypotheses passed GCC.

Possible complementarity between the proposed approach and FAB-MAP is analyzed in Table 4. This result shows a significant number of cases where one of the compared methods fails and the other succeeds, which indicates the potential of combining these two methods.

Table 4 Complementarity analysis. The column denoted by "proposed approach" contains the number of samples which are correctly recognized by our approach and falsely by FAB-MAP, while the column denoted by "FAB-MAP" contains the number of samples which are correctly recognized by FAB-MAP and falsely by our approach.

subset	total	proposed approach	FAB-MAP
1	68	2	6
2	676	65	33
3	246	115	25
4	1175	591	60



subset 1

subset 2

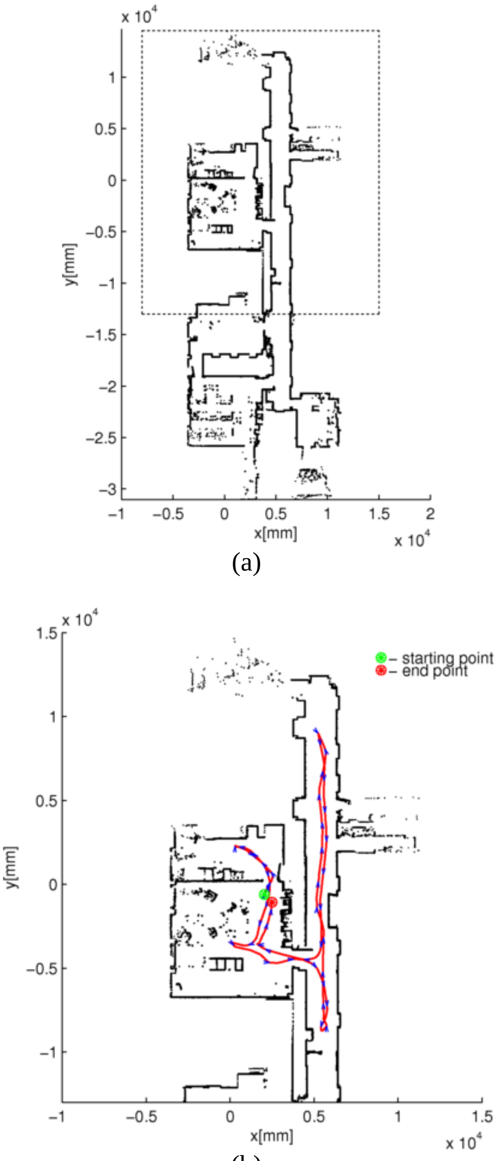
subset 3

subset 4

Fig. 11. Precision-recall curves of the proposed approach, FAB-MAP and DloopDetector for the four test subsets.

7.2 Global localization accuracy

The accuracy of the proposed approach is determined in an *initial global localization* experiment. The algorithm is experimentally evaluated using 3D data provided by a Microsoft Kinect sensor mounted on a wheeled mobile robot Pioneer 3DX also equipped with a laser range finder SICK LMS-200. For the purpose of this experiment, two datasets were generated by manually driving the mobile robot on two different occasions through a section of a previously mapped indoor environment of the Department of Control and Computer Engineering, Faculty of Electrical Engineering and Computing, University of Zagreb. Fig. 12(a) shows the previously mapped indoor environment generated with the aid of SLAM using data from the laser range finder, while Fig. 12(b) shows the trajectory of the robot when generating the first dataset.



(a)

(b)

Fig. 12.(a) Part of the map of the Department of Control and Computer Engineering, (FER, Zagreb) obtained using SLAM and data from a laser ranger finder (b) Trajectory of the wheeled mobile robot while generating images used in creating the topological map.

The first dataset consists of a sequence of RGB-D images recorded along with the odometry data. The corresponding *ground truth* data as to the *exact* pose of the robot in the global coordinate frame of the map was determined using laser data and Monte Carlo localization. A subset of RGB-D images from this dataset was used to create the environment model – a database of local metric models with topological links. This environment model or topological map consisted of a sequence of images, or local models, generated such that the local model of the first image was automatically added to the map and every consecutive image or local model added to the map satisfied at least one of the following conditions: (i) the translational distance between the candidate image and the latest added local model in the map was at least 0.5m or (ii) the difference in orientation between the candidate image and the latest added local model in the map was at least 15°.

This translational distance and difference in orientation between images (local models) in the map were determined using their corresponding ground truth data. The generated topological map had 142 local models for the mapped area shown in Fig. 12(b). Each local model consisted of planar surface segments and line segments represented in the local model reference frame generated from each depth image.

The second dataset, obtained by manually driving the robot on the second occasion, was used to generate the test sequence. The trajectory of the robot during the generation of this sequence was not the same as the first sequence but covered the same area. With the aid of odometry information from the robot, the test sequence was generated by recording RGB-D images every 0.5m or 5° difference in orientation between consecutive images. The corresponding ground truth data was determined using laser data and Monte Carlo localization and recorded as well. The second dataset consisted of a total of 267 images. Examples of database images and test images are given in Fig. 13.

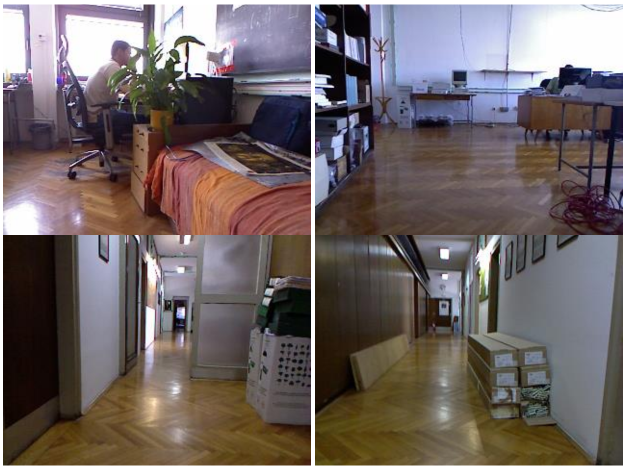


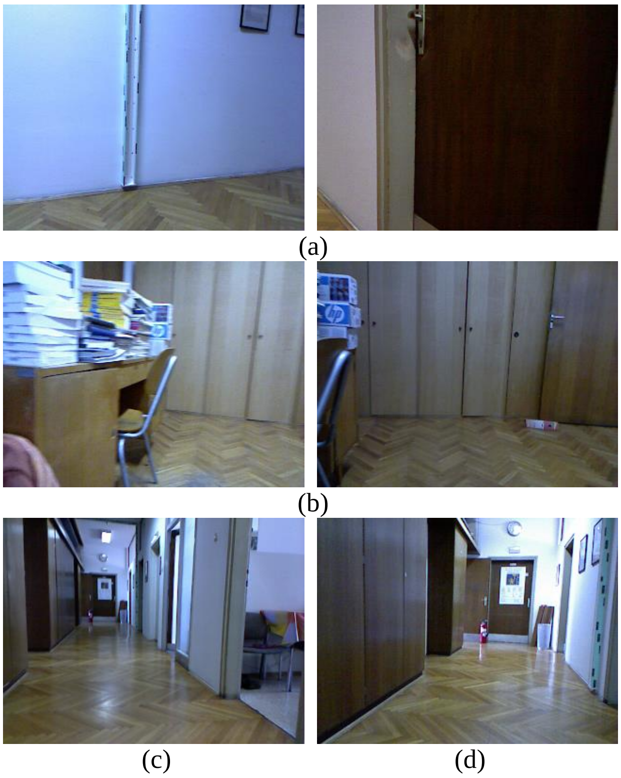
Fig. 13. Examples of images used in the initial global localization experiment.

The proposed place recognition procedure was performed for each image in the test sequence, with the topological map serving as the environment model. For each test image, the best evaluated hypothesis, i.e., the hypothesis with the highest probability is selected as the solution. Since the hypothesis provides the index of the local model from the topological map as well as the relative pose of the test image with respect to the local model, the accuracy of the proposed place recognition approach can be determined. The *calculated* pose of a given test image is determined using the relative pose provided by the best hypothesis as well as the ground truth data of the corresponding local model. By comparing the calculated pose of the test image to the corresponding ground truth pose of the test image, the accuracy of the proposed approach in initial global localization can be determined. An overview of the results of the initial global localization experiment is given in Table 5.

Table 5. Global localization results.

	Number of images	Percentage (%)
Total number of images in the test sequence	267	100
Number of images not localized	5	1.87
Number of images wrongly localized	22	8.24
Number of images correctly localized	240	89.89

Of the 267 test images, the proposed approach was not able to generate any hypothesis in 5 cases. In all 5 cases, the scenes were deficient in information needed to estimate the last DoF of the robot's motion. Examples of such images are shown in Fig. 14(a). Such situations normally arise when the robot comes too close to a wall or when the robot is turning around in a corridor.


(a)

(b)

(c) (d)

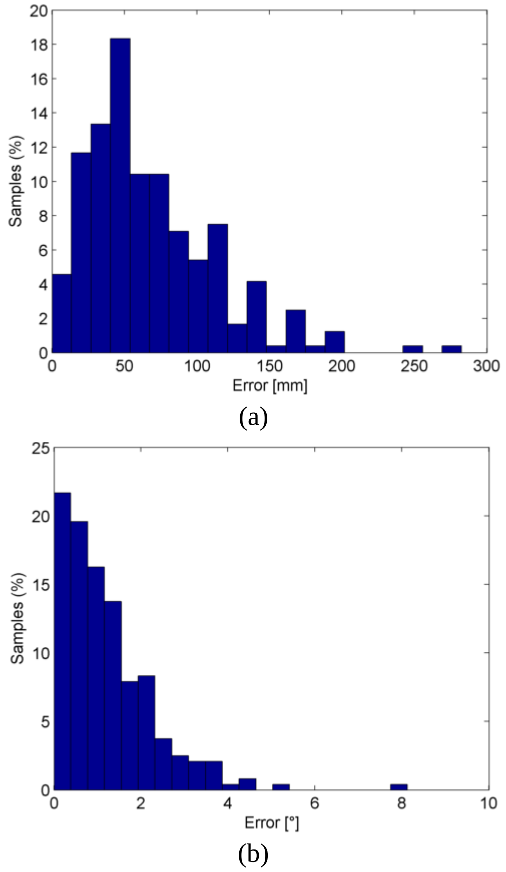
Fig. 14. Examples of images either not localized or wrongly localized. (a) Test images deficient in information needed to estimate the last DoF of the robot's motion; (b) test images not covered by local models in the topological map; (c) image in the topological map containing a repetitive structure (similar doorways); (d) corresponding test image wrongly localized having a similar doorway.

In 22 cases, the best hypothesis generated by the proposed approach wrongly localized the test images. There were two main reasons for such errors: (i) the topological map did not contain a local model covering the scene of the test image. Examples of such images are shown in Fig. 14(b); (ii) the existence of repetitive structures in the indoor environment. An example of this can be seen in in Fig. 14(c), which represents an image in the topological map, where one can notice the similar *repeating* doorways on the left. Fig. 14(d), representing a test image, was localized such that the visible doorway on the left was matched to the first doorway on the left in Fig. 14(c).

The accuracy of the proposed approach is determined using the 240 correctly localized images. The results are shown statistically, in Table 6 as well as in Fig. 15 and Fig. 16, in terms of the absolute error in position and orientation between the calculated pose and corresponding ground truth pose of the test sequence images.

Table 6. Statistical details of the global localization pose error.

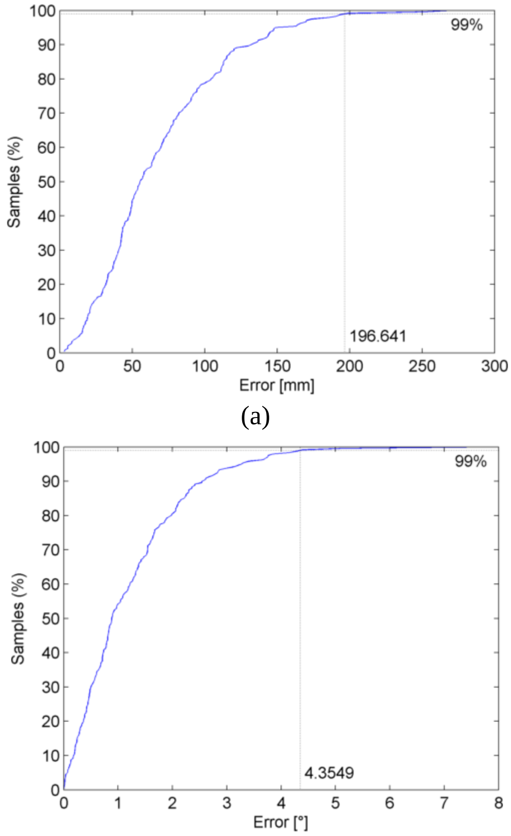
	Error [mm]	Error [°]
Avg.	68.209	1.216
Std.	45.757	1.069
Min.	3.098	0.004
Max.	268.909	7.735



(a)

(b)

Fig. 15. Histogram of the error in (a) position (b) orientation.



(a)

(b)

Fig. 16 Normalized cumulative histogram of the error in (a) position (b) orientation.

The error in position was on average approximately 68 mm with a standard deviation of about 46 mm, while the difference in orientation was on average approximately 1.2° with a standard deviation of about 1°. Looking at Fig. 16, it can be concluded that for 99% of the samples, the pose error was at most 197mm and 4.4°.

7.3 Computational Complexity

The proposed method represents a sequence of the following processing steps:

1. detection of planar surface segments,
2. detection of line segments,
3. sampling of surface segments,
4. hypothesis generation,
5. hypothesis evaluation and
6. rejection of dynamic surfaces.

The computational complexity of the first three steps does not depend on the map size, while the computational complexity the last three steps depends on the total number of the local models N_M in the map.

The computational complexity of the hypothesis generation step is linear in the number of generated nodes of the hypothesis tree. This number, however, varies significantly depending on the characteristics of the observed scene. The total number of generated nodes per local model is limited by the user specified parameter N_V. However, in cases where the scene contains sufficient information to determine all 6DoF of the camera pose, the maximum number of N_H hypotheses are generated before reaching N_V nodes. In the opposite case, the hypothesis generation step stops after generating N_V before N_H hypotheses are generated.

The computational complexity of the hypothesis evaluation step varies significantly depending on the geometry of the observed scene and the entire modelled environment. If many local models have similar geometry, then a high percentage of the generated hypotheses will remain after the hypothesis pruning and the computationally consuming hypothesis evaluation step will be applied to all of them.

If we assume a uniform distribution of the processing time of the last three steps over all local models, the computational complexity of the proposed method increases linearly with the number of local models in the model database. This is, in general, a drawback in comparison to FAB-MAP whose complexity is linear in the size of the used dictionary, which means that the localization time of FAB-MAP does not rise with the number of model nodes.

The computation time needed to obtain the final pose hypothesis from a depth image is presented in Table 7. In this table, the computation times for particular steps of the proposed approach are also shown. The data in this table represent the average values computed over 1175 images on an Intel Core 2 Duo CPU at 2GHz and 4 GB RAM.

Table 7 Average computation time in seconds for each algorithm step.

step	per image	per local model
surface segment detection	0.152	-
line segment detection	0.011	-
surface segment sampling	0.027	-
hypothesis generation	0.242	0.00537
hypothesis evaluation	0.059	0.00132
dynamic surface rejection	0.001	0.00002
feature detection (first 3 steps)	0.190	-
localization (last 3 steps)	0.302	0.00672
total	0.492	

The localization time of our approach is 0.302 s for the map consisting of 45 local models, which is about 3.5 times higher than that of OpenFABMAP, whose average localization time is 0.085 s. The feature detection time of our approach is 0.190 s, while OpenFABMAP can be used with different feature detectors and its feature detection time depends on the feature detector used. Nevertheless, despite a higher computation time in comparison to FAB-MAP, the approach proposed in this paper is more suitable for applications where robustness to changes in lighting conditions is critical.

8. Conclusion

In this paper, the potential of using 3D planar surfaces and line segments detected in depth images for place recognition is investigated. A place recognition approach based on the aforementioned geometric features is proposed and experimentally evaluated. The proposed approach includes two novel solutions, an efficient hypothesis generation method which ranks the features according to their potential contribution to the pose information, thereby reducing the time needed for obtaining accurate pose estimation and a robust probabilistic method for selecting the best pose hypothesis. The presented place recognition system is designed for indoor and urban environments where planar surfaces and straight object edges are dominant structures. Since the proposed approach allows matching of partially overlapping point clouds, it enables fast and simple generation of environment maps by taking a sequence of depth images while driving along a path which the robot is expected to follow during its regular operation, in contrast to the approaches which need complete metric environment models.

The developed place recognition system is experimentally evaluated using a benchmark dataset consisting of a reference dataset from which an environment model is created and a test dataset which includes images of the reference scenes with changes in the lighting conditions and presence of dynamic objects. The proposed approach has shown better results than FAB-MAP and DLoopDetector in the case of significant changes in the environment and comparable performance in the case where the conditions correspond to those in the reference images.

Nevertheless, only indoor scenes are included in the benchmark dataset used in the reported research, which gives an advantage to our method. Therefore, from the obtained results it cannot be concluded that our approach is generally better than FAB-MAP or some other appearance-based method.

Besides its good properties, the proposed approach has also significant limitations. It actually determines the pose of the camera relative to a particular geometric structure. The more complex the structure the more convinced is the place recognition algorithm in its decision. Consequently, the method can be expected to fail in the cases where a frequently moving object of a complex structure is present within a local model. Such objects are e.g. tables, chairs and other movable furniture. If a robot tries to localize itself by analyzing a camera image taken after this object changes its pose relative to the local model reference frame, a false pose hypothesis will be generated. Actually, in that case the robot localizes itself correctly relative to the considered object, but incorrectly with respect to the local model. In order for the proposed method to make a correct place recognition, the structure of a stable (not-moving) part of the observed scene must have a higher complexity then the dynamic objects present in that scene.

Like many other computer vision methods, our approach has a number of user defined parameters which we determine experimentally. Determining the optimal values for all these parameters or an analysis of their influence to the performance of the proposed approach could be a topic of further research. Nevertheless, in this paper we demonstrated that the proposed approach is feasible and provides compelling results in comparison to other state-of-the-art techniques even with a non-optimal parameter set.

The results of the research presented in this paper indicate that the considered approach based on geometric features is applicable for robust place recognition in indoor environments. Since geometric features are substantially different from features like those obtained by SIFT, SURF or similar methods, a potential exists of combining these two types of features in a system which would rely on geometric features where the intensity image features are not reliable and opposite.

Funding

This work has been supported by European Community's Seventh Framework Programme under Grant 285939 (ACROSS).

References

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

Badino H, Huber D and Kanade T (2012) Real-Time Topometric Localization, In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 1635-1642.

Bay H, Ess A, Tuytelaars T and Van Gool L (2008) SURF: Speeded Up Robust Features, *Computer Vision and Image Understanding (CVIU)*, Vol. 110, No. 3, pp. 346 – 359.

Bradski G and Kaehler A (2008) *Learning OpenCV*, O'Reilly.

Chum O and Matas J (2005) Matching with PROSAC-progressive sample consensus, In: *Computer Vision and Pattern Recognition (CVPR)*, vol. 1

Ciarfuglia TA, Constante G, Valiagi P and Ricci E (2012) A Discriminative Approach for Appearance Based Loop Closing, In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3837-3843.

Cobzas D and Zhang H (2001) Mobile Robot Localization using Planar Patches and a Stereo Panoramic Model, *Vision Interface*, pp. 94-99.

Cummins M and Newman P (2009) Highly scalable appearance-only SLAM - FAB-MAP 2.0. In *Robotics: Science and Systems*, Seattle, USA.

Cupec R, Nyarko EK, Filko D and Petrović I (2012) Fast Pose Tracking Based on Ranked 3D Planar Patch Correspondences. In *IFAC Symposium on Robot Control*, Dubrovnik, Croatia.

Cupec R, Nyarko EK, Filko D, Kitanov A and Petrović I. (2013) Global Localization Based on 3D Planar Surface Segments. In: *Croatian Computer Vision Workshop (CCVW)*, Zagreb, Croatia, pp. 31-36

Douglas D and Peucker T (1973) Algorithms for the reduction of the number of points required for represent a digitized line or its caricature, *Canadian Cartographer* 10(1973): 112–122.

Endres F, Hess J, Engelhard N, Sturm J, Cremers D and Burgard W (2012) An Evaluation of the RGB-D SLAM System, In: *IEEE Int. Conf. on Robotics and Automation (ICRA)*.

Fallon MF, Johannsson H and Leonard JJ (2012) Efficient scene simulation for robust monte carlo localization using an RGB-D camera, *2012 IEEE International Conference on Robotics and Automation*, pp. 1663–1670.

Faugeras O (1993) Three-Dimensional Computer Vision: A Geometric Viewpoint. *Cambridge, Massachusetts: The MIT Press*.

Fischler MA and Bolles RC (1981) Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, *Graphics and Image Processing*, vol. 24, no. 6, pp. 381–395.

Galvez-Lopez D and Tardos JD (2012) Bags of Binary Words for Fast Place Recognition in Image Sequences, *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197.

Garland M, Willmott A and Heckbert PS (2001) Hierarchical Face Clustering on Polygonal Surfaces, In *ACM Symposium on Interactive 3D Graphics*.

Glover A, Maddern W, Warren M, Reid S, Milford M and Wyeth G (2012) OpenFABMAP: an open source toolbox for appearance-based loop closure detection. In *International Conference on Robotics and Automation*, pp. 14-18.

Granström K, Schön TB, Nieto JI, Ramos FT (2011) Learning to close loops from range data, *The International Journal of Robotics Research*, vol. 30, no. 14, pp. 1728-1754.

Huang AS, Bachrach A, Henry P, Krainin M, Maturana D, Fox D and Roy N (2011) Visual Odometry and Mapping for Autonomous Flight Using an RGB-D Camera, *Int. Symposium on Robotics Research (ISRR)*, Flagstaff, Arizona, USA.

Kawewong A, Tongprasit N, Tangruamsub S and Hasegawa O (2011) Online and Incremental Appearance-based SLAM in Highly Dynamic Environments, *The International Journal of Robotics Research*, vol. 30, no. 1, pp. 33–55.

Kosaka A and Kak A (1992) Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties, *CVGIP: Image Understanding*, vol. 56, pp. 271–329.

Krainin M, Konolige K and Fox D (2012) Exploiting Segmentation for Robust 3D Object Matching, In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 4399 – 4405.

Liu Y and Zhang H (2012) Indexing Visual Features: Real-Time Loop Closure Detection Using a Tree Structure, In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 3613 – 3618.

Lowe DG (2004) Distinctive Image Features from Scale-Invariant Keypoints, *The International Journal of Computer Vision*, vol. 60, no. 2, pp. 91 – 110.

Matthies L and Shafer SA (1987) Error Modeling in Stereo Navigation, *IEEE Journal of Robotics and Automation*, vol. 3, pp. 239 – 248.

Milford M (2013) Vision-based place recognition: how low can you go, *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 766–789.

Milford MJ and Wyeth GF (2012) SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights, In *IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 1643 – 1649*

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.

Paul R and Newman P (2010) FAB-MAP 3D: Topological Mapping with Spatial and Visual Appearance, In: *IEEE Int. Conf. on Robotics and Automation (ICRA)*.

Pronobis A, Caputo B, Jensfelt P and Christensen HI (2010) A realistic benchmark for visual indoor place recognition, *Robotics and Autonomous Systems*, vol. 58, no. 1, pp. 81-96.

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

Se S, Lowe DG and Little JJ (2005) Vision-based global localization and mapping for mobile robots, *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 364-375, 2005.

Stückler J and Behnke S (2013) Multi-Resolution Surfel Maps for Efficient Dense 3D Modeling and Tracking, *Journal for Visual Communication and Image Representation*.

Thrun S, Burgard W and Fox D (2005) Probabilistic Robotics, The MIT Press.

Appendix A: Surface Segment Matching Criteria

Given an estimated pose $\boldsymbol{w}$ of a scene relative to a local model, matching of a scene surface segment F to a local model surface segment F' is performed by transforming the parameters of F into the reference frame of F' using the pose $\boldsymbol{w}$ and comparing the transformed parameters to the parameters of F'. The parameters of the plane supporting F can be transformed from the reference frame S_F to the reference frame $S_{F'}$ by transforming the plane equation (1). Given a vector ${}^{F'}\boldsymbol{p}$ representing the position of a point relative to $S_{F'}$, the same point is represented in S_F by

$${}^{F}\boldsymbol{p} = {}^{A}\boldsymbol{R}_F^T \left(\boldsymbol{R}^T(\phi)\left({}^{B}\boldsymbol{R}_{F'}\ {}^{F'}\boldsymbol{p} + {}^{B}\boldsymbol{t}_{F'} - \boldsymbol{t} \right) - {}^{A}\boldsymbol{t}_F \right) \qquad (32)$$

By substituting (32) into (1) we obtain

$${}^{F'}\boldsymbol{n}^T \cdot {}^{F'}\boldsymbol{p} = {}^{F'}\rho \qquad (33)$$

where

$${}^{F'}\boldsymbol{n} = {}^{B}\boldsymbol{R}_{F'}^T \boldsymbol{R}(\phi)\, {}^{A}\boldsymbol{R}_F\, {}^{F}\boldsymbol{n}, \qquad (34)$$

$${}^{F'}\rho = {}^{F}\rho + {}^{F}\boldsymbol{n}^T \cdot {}^{A}\boldsymbol{R}_F^T \left({}^{A}\boldsymbol{t}_F + \boldsymbol{R}^T(\phi)\left(\boldsymbol{t} - {}^{B}\boldsymbol{t}_{F'} \right) \right). \qquad (35)$$

Vector ${}^{F'}\boldsymbol{n}$ and value ${}^{F'}\rho$ are the normal of F represented in $S_{F'}$ and the distance of the plane supporting F from the origin of $S_{F'}$. The deviation of the plane supporting the scene surface segment from the plane containing the local model surface segment can be described by the difference between the plane normals and their distances from the origin of $S_{F'}$. Assuming that F and F' represent the segments of the same planar surface, the following equations hold

$${}^{F'}\boldsymbol{n} = {}^{F'}\boldsymbol{n}', \qquad (36)$$

$${}^{F'}\rho = {}^{F'}\rho', \qquad (37)$$

where ${}^{F'}\boldsymbol{n}'$ and ${}^{F'}\rho$ are the parameters of the plane supporting F' represented in reference frame $S_{F'}$. Since ${}^{F'}\boldsymbol{n}$ and ${}^{F'}\boldsymbol{n}'$ are unit vectors with two degrees of freedom, it is appropriate to compare only their two components. We choose the first two components to formulate the coplanarity constraint

$$\begin{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \left({}^{F'}\boldsymbol{n} - {}^{F'}\boldsymbol{n}' \right) \\ \hdashline {}^{F'}\rho - {}^{F'}\rho' \end{bmatrix} = \boldsymbol{0} \qquad (38)$$

Note that the vector on the left side of equation (38) is actually a function of the disturbance vectors $\boldsymbol{q}$ and $\boldsymbol{q}'$ representing the uncertainty of the parameters of the planes supporting F and F' respectively, the pose $\boldsymbol{w}$ and the estimated poses of F and F' relative to S_C and S_M respectively. Hence, the left side of (38) can be represented by the function $\boldsymbol{h}(\boldsymbol{q}, \boldsymbol{q}', \boldsymbol{w}; F, F')$ which maps particular values $\boldsymbol{q}$, $\boldsymbol{q}'$ and $\boldsymbol{w}$ to a 3-component vector representing the deviation from coplanarity. Assuming that $\boldsymbol{w}$ is a normally distributed random variable with mean $\hat{\boldsymbol{w}}$ and covariance $\boldsymbol{\Sigma}_w$, the coplanarity between F and F' can be measured by the Mahalanobis distance (9) where

$$\boldsymbol{e}\left(F,F',\boldsymbol{w}\right)=\boldsymbol{h}\left(\mathbf{0},\mathbf{0},\boldsymbol{w};F,F'\right), \tag{39}$$

$$\boldsymbol{Q_q}=\boldsymbol{E\Sigma_q E}^T+\boldsymbol{E'\Sigma_{q'}E'}^T+\boldsymbol{C_q\Sigma_w C_q}^T, \tag{40}$$

$\boldsymbol{\Sigma_q}$ and $\boldsymbol{\Sigma_{q'}}$ are the covariance matrices describing the uncertainty of the parameters of the matched surface segments and ***E***, ***E'*** and ***C*** are Jacobians

$$\boldsymbol{E}=\frac{\partial\boldsymbol{h}\left(\boldsymbol{q},\boldsymbol{q}',\boldsymbol{w};F,F'\right)}{\partial\boldsymbol{q}},\quad \boldsymbol{E}'=\frac{\partial\boldsymbol{h}\left(\boldsymbol{q},\boldsymbol{q}',\boldsymbol{w};F,F'\right)}{\partial\boldsymbol{q}'}, \tag{41}$$

$$\boldsymbol{C_q}=\frac{\partial\boldsymbol{h}\left(\boldsymbol{q},\boldsymbol{q}',\boldsymbol{w};F,F'\right)}{\partial\boldsymbol{w}}. \tag{42}$$

Overlapping between two surface segments could be measured according to the area of their overlapping parts. However, since computing the exact value of this area is computationally expensive, an approximate measure is applied. Each surface segment is represented by an elliptical planar patch approximating the distribution of points supporting this segment, as described in Section 4, and overlapping between two surface segments is measured by the Mahalanobis distance between two points representing the centroids of these surface segments, where the position uncertainty of these points is described by the covariance matrices $\boldsymbol{\Sigma_p}$ and $\boldsymbol{\Sigma_{p'}}$. This Mahalanobis distance is computed by

$$d_{\boldsymbol{p}}\left(F,F',\boldsymbol{w}\right)=\left({}^M\boldsymbol{t}_F-{}^M\boldsymbol{t}_{F'}\right)^T\boldsymbol{Q}_{\boldsymbol{p}}^{-1}\left({}^M\boldsymbol{t}_F-{}^M\boldsymbol{t}_{F'}\right), \tag{43}$$

$${}^M\boldsymbol{t}_F=\boldsymbol{R}(\boldsymbol{\phi})\cdot{}^C\boldsymbol{t}_F+\boldsymbol{t}, \tag{44}$$

$$\boldsymbol{Q_p}=\boldsymbol{R}(\boldsymbol{\phi})\cdot\boldsymbol{\Sigma_p}\cdot\boldsymbol{R}^T(\boldsymbol{\phi})+\boldsymbol{\Sigma_{p'}}+\boldsymbol{C_p}\cdot\boldsymbol{\Sigma_w}\cdot\boldsymbol{C_p}^T, \tag{45}$$

$$\boldsymbol{C_p}=\frac{\partial\,{}^M\boldsymbol{t}_F}{\partial\boldsymbol{w}}.$$

The overlapping constraint is formulated as

$$d_{\boldsymbol{p}}\left(F,F',\boldsymbol{w}\right)\le\varepsilon_{\boldsymbol{p}}, \tag{46}$$

Threshold $\varepsilon_{\boldsymbol{p}}$ can be computed according to a desired matching probability assuming χ^2 distribution of $d_{\boldsymbol{p}}$ distance.

Appendix B: Measurement of the Sixth DoF Provided by a Pair of Surface Segments

This appendix explains the estimation of the sixth DoF for the considered voting scheme from a surface segment pair $\left(F,F'\right)$, where *F'* is a hypothesis model surface segment and *F* is a scene surface segment transformed into the reference frame of *F'*. Assuming that *F* and *F'* are approximately parallel, the translation value *l* can be estimated from the distance of the supporting plane of *F* and the origin of the reference frame of *F'*. The translation vector between the camera reference frame and the hypothesis model reference frame can be written as

$$\boldsymbol{t}=\boldsymbol{t}'+l\cdot\tilde{\boldsymbol{t}} \tag{47}$$

By substituting (47) into (35) and the obtained equation into (37) we obtain

$${}^{F'}\!\rho'={}^F\rho+{}^F\boldsymbol{n}^T\cdot{}^A\boldsymbol{R}_F^T\left({}^A\boldsymbol{t}_F+\boldsymbol{R}^T\left(\phi\right)\left(\boldsymbol{t}'+l\cdot\tilde{\boldsymbol{t}}-{}^B\boldsymbol{t}_{F'}\right)\right) \tag{48}$$

From (48) follows an explicit expression for *l* as a function of ***q*** and ***q'***

$$l\left(\boldsymbol{q},\boldsymbol{q}'\right)= \\ =\frac{{}^{F'}\!\rho-{}^F\rho-{}^F\boldsymbol{n}^T\cdot{}^A\boldsymbol{R}_F^T\left({}^A\boldsymbol{t}_F+\boldsymbol{R}^T\left(\phi\right)\left(\boldsymbol{t}'-{}^B\boldsymbol{t}_{F'}\right)\right)}{{}^F\boldsymbol{n}^T\cdot{}^A\boldsymbol{R}_F^T\boldsymbol{R}^T\left(\phi\right)\cdot\tilde{\boldsymbol{t}}} \tag{49}$$

The value *l* can be regarded as a normally distributed random variable with mean $\hat{l}=l\left(\mathbf{0},\mathbf{0}\right)$ and variance

$$\sigma_l=\sigma_r\left(\frac{\partial l\left(\boldsymbol{q},\boldsymbol{q}'\right)}{\partial r}\right)^2+\sigma_r'\left(\frac{\partial l\left(\boldsymbol{q},\boldsymbol{q}'\right)}{\partial r'}\right)^2,$$

where σ_r and σ_r' are variances of the third component of the disturbance vectors ***q*** and ***q'*** respectively, introduced in Section 4.

For surface segments, γ represents the smaller number of the supporting points of *F* and *F'*.

Appendix C: Measurement of the Sixth DoF Provided by a Pair of Line Segments

This appendix explains estimation of the sixth DoF for the considered voting scheme from a line segment pair $\left(F,F'\right)$, where *F'* is a local model line segment and *F* is a scene line segment transformed from the camera reference frame into the reference frame of the local model S_M. This transformation is performed by transforming the both endpoints ${}^C\boldsymbol{p}_1$ and ${}^C\boldsymbol{p}_2$ as well as their covariance matrices $\boldsymbol{\Sigma_{p,1}}$ and $\boldsymbol{\Sigma_{p,2}}$.

An auxiliary coordinate system S_L is defined with origin identical to the origin of the local model reference frame S_M, x-axis identical to $\tilde{\boldsymbol{t}}$ and y-axis defined by

$${}^M\boldsymbol{y}_L=\frac{\left({}^M\boldsymbol{u}+{}^M\boldsymbol{u}'\right)\times\tilde{\boldsymbol{t}}}{\left\|\left({}^M\boldsymbol{u}+{}^M\boldsymbol{u}'\right)\times\tilde{\boldsymbol{t}}\right\|},$$

where

$${}^M\boldsymbol{u}=\frac{\boldsymbol{p}_2-\boldsymbol{p}_1}{\left\|\boldsymbol{p}_2-\boldsymbol{p}_1\right\|}$$

and ${}^M\boldsymbol{u}'$ is defined analogously. Both line segments are transformed into the coordinate system S_L and correspondence between their points is established according to their z-coordinate in this coordinate system. A point on the line segment *F* corresponds to a point on the line segment *F'* if their z-coordinates with respect to

S_L are equal, as illustrated in Fig. 17. The subsets of *F* and *F'* which have their corresponding points represent overlapping parts of these two line segments.

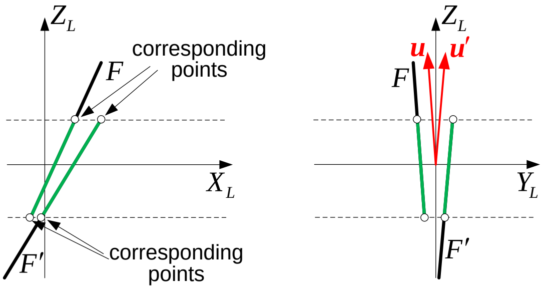


Fig. 17. Two line segments *F* and *F'* represented in coordinate system S_L. Examples of corresponding points are denoted by circles. The overlapping parts of the considered line segments are located between the dashed lines. Unit vectors $\boldsymbol{u}$ and $\boldsymbol{u}'$ are parallel to *F* and *F'* respectively and have equal angle relative to xz-plane of the coordinate system S_L.

Any pair of corresponding points represents a measurement of the sixth DoF. Let $\overline{\boldsymbol{p}}$ and $\overline{\boldsymbol{p}}'$ be two-component vectors representing the x and y coordinate of two corresponding points in the coordinate system S_L and let $\boldsymbol{\Sigma}_{\overline{\boldsymbol{p}}}$ and $\boldsymbol{\Sigma}_{\overline{\boldsymbol{p}}'}$ be the covariance matrices representing the measurement uncertainty of these two points. The correction of the translation vector $\boldsymbol{t}'$ according to the points $\overline{\boldsymbol{p}}$ and $\overline{\boldsymbol{p}}'$ is given by

$$\overline{\boldsymbol{t}} = \overline{\boldsymbol{p}} - \overline{\boldsymbol{p}}'.$$

The uncertainty of this measurement can be computed by propagating the uncertainties of the two considered points according to the equation

$$\boldsymbol{\Sigma}_{\overline{\boldsymbol{t}}} = \frac{\partial\overline{\boldsymbol{t}}}{\partial\overline{\boldsymbol{p}}}\boldsymbol{\Sigma}_{\overline{\boldsymbol{p}}}\left(\frac{\partial\overline{\boldsymbol{t}}}{\partial\overline{\boldsymbol{p}}}\right)^T + \frac{\partial\overline{\boldsymbol{t}}}{\partial\overline{\boldsymbol{p}}'}\boldsymbol{\Sigma}_{\overline{\boldsymbol{p}}'}\left(\frac{\partial\overline{\boldsymbol{t}}}{\partial\overline{\boldsymbol{p}}'}\right)^T. \tag{50}$$

Since

$$\frac{\partial\overline{\boldsymbol{t}}}{\partial\overline{\boldsymbol{p}}} = -\frac{\partial\overline{\boldsymbol{t}}}{\partial\overline{\boldsymbol{p}}'} = \boldsymbol{I}^{2\times2}. \tag{51}$$

it follows that

$$\boldsymbol{\Sigma}_{\overline{\boldsymbol{t}}} = \boldsymbol{\Sigma}_{\overline{\boldsymbol{p}}} + \boldsymbol{\Sigma}_{\overline{\boldsymbol{p}}'}. \tag{52}$$

In our implementation the value *l* is estimated using both endpoints of the overlapping parts of the line segments *F* and *F'* denoted in Fig. 17 by circles. Fusion of these two measurements is given by

$$\overline{\boldsymbol{t}}_{12} = \overline{\boldsymbol{\Sigma}}_{\overline{\boldsymbol{t}},12}\left(\overline{\boldsymbol{\Sigma}}_{\overline{\boldsymbol{t}},1}^{-1}\overline{\boldsymbol{t}}_1 + \overline{\boldsymbol{\Sigma}}_{\overline{\boldsymbol{t}},2}^{-1}\overline{\boldsymbol{t}}_2\right), \tag{53}$$

where

$$\overline{\boldsymbol{\Sigma}}_{\overline{\boldsymbol{t}},12} = \left(\overline{\boldsymbol{\Sigma}}_{\overline{\boldsymbol{t}},1}^{-1} + \overline{\boldsymbol{\Sigma}}_{\overline{\boldsymbol{t}},2}^{-1}\right)^{-1}. \tag{54}$$

Vector $\overline{\boldsymbol{t}}_{12}$ represents correction of the translation vector $\boldsymbol{t}'$ estimated by the two considered measurements and $\overline{\boldsymbol{\Sigma}}_{\overline{\boldsymbol{t}},12}$ is the covariance matrix describing its uncertainty. Value *l* is the x-component of $\overline{\boldsymbol{t}}_{12}$, i.e.

$$l = \begin{bmatrix}1 & 0\end{bmatrix}\cdot\overline{\boldsymbol{t}}_{12}$$

and its estimated variance is

$$\sigma_l = \begin{bmatrix}1 & 0\end{bmatrix}\overline{\boldsymbol{\Sigma}}_{\overline{\boldsymbol{t}},12}\begin{bmatrix}1\\0\end{bmatrix}.$$

For line segments, γ corresponds to the size of the overlapping parts of the matched line segments.

Appendix D: Surface Sample Matching

Let ${}^M\boldsymbol{p}' = \left[\,{}^Mx',\ {}^My',\ {}^Mz'\right]^T$ be the vector defining the position of a sample center point relative to the hypothesis model. Then, the coordinates of this point relative to the camera reference frame are given by vector

$${}^C\boldsymbol{p}' = \boldsymbol{R}^T(\boldsymbol{\phi})\left({}^M\boldsymbol{p}' - \boldsymbol{t}\right) \tag{55}$$

and its projection onto the scene image is given by

$${}^C\boldsymbol{m}' = \boldsymbol{f}_{pr}\left(\boldsymbol{P}\cdot{}^C\boldsymbol{p}'\right), \tag{56}$$

where $\boldsymbol{f}_{pr}$ is function defined by

$$\boldsymbol{f}_{pr}\left(\left[x,\,z,\,y\right]^T\right) = \left[x/z,\ y/z\right]^T$$

and $\boldsymbol{P}$ is the camera projection matrix defined by

$$\boldsymbol{P} = \begin{bmatrix} f_u & 0 & u_c \\ 0 & f_v & v_c \\ 0 & 0 & 1 \end{bmatrix},$$

with f_u, f_v, u_c and v_c being intrinsic camera parameters, according to a commonly used pinhole camera model (Bradski G, 2008).

Although in general the surface sample patches do not project to squares in the image, under assumption of a small displacement between the current scene view and the hypothesis model view, their image projections can be approximated by squares of size

$${}^Cw' = w'\frac{{}^Mz'}{{}^Cz'}, \tag{57}$$

where w' is the original sample window size and ${}^Cw'$ is the size of the approximated projection of the sample patch onto the camera image.

The overlapping condition, illustrated in Fig. 8 can be formulated by the following equation

$$\left\|{}^C\boldsymbol{m}' - \boldsymbol{m}\right\|_{cheb} - \min\left\{{}^Cw', w\right\} \le \delta_{so}, \tag{58}$$

where $\boldsymbol{m}$ and is the center of a scene sample, w is the window size of the scene sample, $\left\|\cdot\right\|_{cheb}$ denotes the Chebyshev distance and δ_{so} is tolerance due to the uncertainty of the hypothesis pose. Tolerance δ_{so} is computed by

$$\delta_{so} = \max\left\{ f_u, f_v \right\} \frac{\delta_{so\phi} \left\| {}^{C}\boldsymbol{p}' \right\| + \delta_{sot}}{{}^{C}z'} , \qquad (59)$$

where $\delta_{so\phi}$ and δ_{sot} are constants representing the expected orientation and translation uncertainty respectively. The term in the nominator of (59) represents the estimated uncertainty of the position of the hypothesis model sample center transformed into the scene coordinate system. Tolerance δ_{so} is the image projection of this uncertainty. Alternatively, the uncertainty of the transformed sample position could be computed using the estimated pose uncertainty obtained by the EKF. However, this would require a more complex computation. Since the condition (58) is evaluated for many sample pairs, it is important that this evaluation does not take much computation time and therefore, the approximate formula (59) is used, where constants $\delta_{so\phi}$ and δ_{sot} are determined experimentally.

The similarity between depths and orientations of the surface patches corresponding to two samples is evaluated by taking into account the uncertainties of their parameters. We use a simple uncertainty model illustrated in Fig. 18, which is based on the uncertainty parameters σ_p and $\boldsymbol{v}$ assigned to each sample, as explained in Section 6.1.

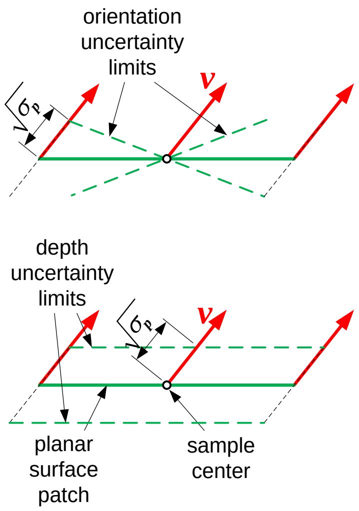


Fig. 18. Uncertainty model of the surface patch corresponding to a surface sample.

Appendix E: Derivation of (29)

By multiplying the left and the right sides of the equations (27) and (28) the following is obtained

$$p\left(F_i \mid \boldsymbol{s}, \phi_k\right) p\left(F_j' \mid \boldsymbol{s}\right) = \frac{1}{(2\pi)^2 \sqrt{\det\left({}^{F'}\boldsymbol{\Sigma}_{s,i}\right) \det\left(\boldsymbol{\Sigma}_{s',j}\right)}} \cdot \exp\left(-\frac{g(\boldsymbol{s})}{2}\right) \qquad (60)$$

where

$$g(\boldsymbol{s}) = \left({}^{F'}\hat{\boldsymbol{s}}_i - \boldsymbol{s}\right)^T {}^{F'}\boldsymbol{\Sigma}_{s,i}^{-1} \left({}^{F'}\hat{\boldsymbol{s}}_i - \boldsymbol{s}\right) + \boldsymbol{s}^T \boldsymbol{\Sigma}_{s',j}^{-1} \boldsymbol{s} \cdot$$

Term $g(\boldsymbol{s})$ can be expanded as follows

$$\begin{aligned} g(\boldsymbol{s}) = &\left({}^{F'}\hat{\boldsymbol{s}}_i - \hat{\boldsymbol{s}}_{ij} + \hat{\boldsymbol{s}}_{ij} - \boldsymbol{s}\right)^T {}^{F'}\boldsymbol{\Sigma}_{s,i}^{-1} \left({}^{F'}\hat{\boldsymbol{s}}_i - \hat{\boldsymbol{s}}_{ij} + \hat{\boldsymbol{s}}_{ij} - \boldsymbol{s}\right) + \\ &\left(\boldsymbol{s} - \hat{\boldsymbol{s}}_{ij} + \hat{\boldsymbol{s}}_{ij}\right)^T \boldsymbol{\Sigma}_{s',j}^{-1} \left(\boldsymbol{s} - \hat{\boldsymbol{s}}_{ij} + \hat{\boldsymbol{s}}_{ij}\right) \end{aligned} \qquad (61)$$

By rearranging the right side of (61) the following is obtained

$$\begin{aligned} g(\boldsymbol{s}) = &\left({}^{F'}\hat{\boldsymbol{s}}_i - \hat{\boldsymbol{s}}_{ij}\right)^T {}^{F'}\boldsymbol{\Sigma}_{s,i}^{-1} \left({}^{F'}\hat{\boldsymbol{s}}_i - \hat{\boldsymbol{s}}_{ij}\right) + \hat{\boldsymbol{s}}_{ij}^T \boldsymbol{\Sigma}_{s',j}^{-1} \hat{\boldsymbol{s}}_{ij} \\ &+ \left(\boldsymbol{s} - \hat{\boldsymbol{s}}_{ij}\right)^T \left({}^{F'}\boldsymbol{\Sigma}_{s,i}^{-1} + \boldsymbol{\Sigma}_{s',j}^{-1}\right)\left(\boldsymbol{s} - \hat{\boldsymbol{s}}_{ij}\right) \end{aligned} \qquad (62)$$

By substituting $g(\boldsymbol{s})$ in (60) with (62) we obtain

$$\begin{aligned} &\int_{\boldsymbol{s}} p\left(F_i \mid \boldsymbol{s}, \phi_k\right) p\left(F_j' \mid \boldsymbol{s}\right) d\boldsymbol{s} = \\ &\frac{1}{(2\pi)^2 \sqrt{\det\left({}^{F'}\boldsymbol{\Sigma}_{s,i}\right) \det\left(\boldsymbol{\Sigma}_{s',j}\right)}} \cdot \exp\left(-\frac{L_{ij}}{2}\right) \cdot \\ &\int_{\boldsymbol{s}} \exp\left(-\frac{\left(\boldsymbol{s} - \hat{\boldsymbol{s}}_{ij}\right)^T \left({}^{F'}\boldsymbol{\Sigma}_{s,i}^{-1} + \boldsymbol{\Sigma}_{s',j}^{-1}\right)\left(\boldsymbol{s} - \hat{\boldsymbol{s}}_{ij}\right)}{2}\right) d\boldsymbol{s} \end{aligned} \qquad (63)$$

Assuming that the components of $\boldsymbol{s}$ can obtain arbitrary large values, the last term on the right side of (63) can be integrated over interval $\left\langle -\infty, +\infty \right\rangle$. Hence

$$\int_{\boldsymbol{s}} \exp\left(-\frac{\left(\boldsymbol{s} - \hat{\boldsymbol{s}}_{ij}\right)^T \left({}^{F'}\boldsymbol{\Sigma}_{s,i}^{-1} + \boldsymbol{\Sigma}_{s',j}^{-1}\right)\left(\boldsymbol{s} - \hat{\boldsymbol{s}}_{ij}\right)}{2}\right) d\boldsymbol{s} = 2\pi \sqrt{\det\left(\left({}^{F'}\boldsymbol{\Sigma}_{s,i}^{-1} + \boldsymbol{\Sigma}_{s',j}^{-1}\right)^{-1}\right)} \qquad (64)$$

By substituting (64) into (63) we obtain

$$\begin{aligned} &\int_{\boldsymbol{s}} p\left(F_i \mid \boldsymbol{s}, \phi_k\right) p\left(F_j' \mid \boldsymbol{s}\right) d\boldsymbol{s} = \\ &\frac{\sqrt{\det\left(\left({}^{F'}\boldsymbol{\Sigma}_{s,i}^{-1} + \boldsymbol{\Sigma}_{s',j}^{-1}\right)^{-1}\right)}}{2\pi \sqrt{\det\left({}^{F'}\boldsymbol{\Sigma}_{s,i}\right) \det\left(\boldsymbol{\Sigma}_{s',j}\right)}} \cdot \exp\left(-\frac{L_{ij}}{2}\right) \end{aligned} \qquad (65)$$

Obtaining (29) from (65) is straightforward.