POZNAN UNIVERSITY OF TECHNOLOGY

A new probabilistic approach to global localization in robotics

by

Jan Wietrzykowski

in the

Institute of Robotics and Machine Intelligence
Faculty of Control, Robotics, and Electrical Engineering

Supervisor: Prof. Piotr Skrzypczyński, PhD, DSc

June 22, 2022

Abstract

This thesis presents research on indoor global localization methods using planar segments and various types of cameras as sensors. It debates various aspects of global localization, from planar segment representation to inference algorithms, and presents a complete solution to the problem. It is based on a series of articles published in renowned journals and presented at top-tier conferences. The importance of the topic of this work stems from the fact that global localization is essential in virtually every mobile autonomous system that operates for an extended period of time. It facilitates solving problems such as kidnapped robot or loop closure detection, and making it reliable and robust is of utmost importance. Therefore, this dissertation introduces a novel global localization method that builds a probability density function (PDF) representing the belief about the pose of an agent. The PDF is constructed from local, partial, and uncertain cues from planar segment features. The maximum of this PDF defines the agent's pose that is expressed with respect to a global map of planar segments and has 6 degrees of freedom. New map building and map management algorithms are proposed that enable construction of the global map. The same algorithms are also used to build a local map that represents the current scene and is matched against the global map. Two versions of the system are presented and evaluated, one using an RGB-D camera, and one using a passive stereo camera. As a results of using an RGB-D camera, the first version, PlaneLoc, is capable of accurately reconstructing the geometry of the scene. It utilizes a pose retrieval algorithm that computes the pose using equations of infinite planes supporting the segments. Initial match candidate retrieval is done using color-histogram-based descriptors. However, the limited effective range of RGB-D sensors restricts the observations to nearby objects, which greatly reduces the number of geometric constraints that can be used to retrieve the camera pose. The second version, PlaneLoc2, avoids this problem by using a passive stereo camera that has a larger effective range. To exploit the full potential of passive stereo cameras, a novel planar segment detection method was introduced. The method exploits a new deep neural network (DNN) architecture that is inspired by the Plane R-CNN and uses cost volume to extract geometry information from stereo data. Additionally, it uses a novel camera-agnostic representation of normal vectors to improve geometry reconstruction performance and robustness to camera parameter changes. The DNN also contains an appearance description branch that produces planar segment descriptors used to retrieve match candidates from the global map. The match candidates are used by an improved pose retrieval algorithm that accommodates the uncertainty of depth estimation, making it suitable for stereo cameras. Moreover, the algorithm exploits more constraints by considering the boundaries of planar segments. The methods are evaluated in real-world scenarios and prove their usefulness in global localization. The detection network outperforms the existing state-of-the-art methods in terms of detection and geometry reconstruction accuracy. The proposed learned descriptors allow to fetch fewer match candidates than the ones based on color histograms. As a result of many improvements and novel solutions, the PlaneLoc2 achieves better results than other global localization systems, yielding a high pose recognition rate without incorrect recognitions (false positives).

Streszczenie

Niniejsza rozprawa przedstawia badania nad metodami globalnej lokalizacji wewnątrz pomieszczeń z wykorzystaniem segmentów płaszczyzn.Omówiono w niej poszczególne aspekty globalnej lokalizacji, od reprezentacji segmentów płaszczyzn po algorytmy wnioskowania, oraz przedstawiono kompletne rozwiązanie problemu. Opiera się ona na serii artykułów opublikowanych w renomowanych czasopismach i zaprezentowanych na uznanych konferencjach międzynarodowych. Znaczenie tematu pracy wynika z faktu, że globalna lokalizacja jest niezbędna w praktycznie każdym mobilnym systemie autonomicznym, który jest zaprojektowany do działania przez dłuższy czas. Ułatwia ona rozwiązywanie problemów takich jak *kidnapped robot* czy wykrywanie zamknięcia pętli, a zapewnienie jej niezawodności jest niezwykle ważne. Z tego powodu w niniejszej pracy przedstawiono nowatorską metodę globalnej lokalizacji, która buduje funkcję gęstości prawdopodobieństwa (ang. *probability density function*, PDF) reprezentującą przekonanie o położeniu agenta. PDF jest konstruowana na podstawie lokalnych, częściowych i niepewnych przesłanek pochodzących z wykrytych segmentów płaszczyzn. Maksimum tego PDF wyznacza położenie agenta o 6 stopniach swobody, określone względem globalnej mapy segmentów płaszczyzn. W rozprawie zaproponowano nowe algorytmy budowy i zarządzania mapą, które umożliwiają budowę mapy globalnej. Te same algorytmy są również wykorzystywane do budowania mapy lokalnej, która reprezentuje bieżącą scenę i jest dopasowywana do mapy globalnej. W toku prac przedstawiono i zweryfikowano eksperymentalnie dwie wersje systemu, jedną wykorzystującą kamerę RGB-D oraz drugą wykorzystującą pasywną kamerę stereo. Pierwsza wersja, PlaneLoc, w wyniku zastosowania kamery RGB-D jest w stanie dokładnie zrekonstruować geometrię sceny. Wykorzystuje ona algorytm obliczania pozy, który oblicza pozycję przy użyciu równań nieskończonych płaszczyzn wspierających segmenty. Wstępne wyszukiwanie kandydatów do dopasowania odbywa się z wykorzystaniem deskryptorów opartych na histogramach kolorów. Jednakże niewielki efektywny zasięg czujników RGB-D ogranicza obserwacje do pobliskich obiektów, co znacznie zmniejsza liczbę ograniczeń geometrycznych, które można wykorzystać do obliczenia pozy kamery. Druga wersja, PlaneLoc2, rozwiązuje ten problem przez zastosowanie pasywnej kamery stereo, która posiada większy efektywny zasięg. Aby w pełni wykorzystać potencjał pasywnych kamer stereo, wprowadzono nowatorską metodę wykrywania segmentów płaszczyzn. Metoda ta wykorzystuje nową architekturę głębokiej sieci neuronowej (ang. *deep neural network*, DNN), inspirowaną siecią Plane R-CNN, która wykorzystuje *cost volume* do rekonstrukcji informacji o geometrii z danych stereo. Dodatkowo, wykorzystuje ona nową, niezależną od parametrów kamery reprezentację wektorów normalnych w celu poprawy dokładności rekonstrukcji geometrii. Sieć ta zawiera także gałąź opisu wyglądu, która tworzy deskryptory segmentów płaszczyzn wykorzystywane do pobierania kandydatów do dopasowania z globalnej mapy. Informacja o potencjalnych kandydatach do dopasowania jest następnie wykorzystywana przez ulepszony algorytm obliczania pozy, który uwzględnia niepewność estymacji głębi, dzięki czemu lepiej wykorzystuje potencjał kamer stereo. Co więcej, algorytm ten wykorzystuje więcej ograniczeń poprzez uwzględnienie granic segmentów płaszczyzn. Metody te są ewaluowane w rzeczywistych scenariuszach i potwierdzają swoją przydatność w globalnej lokalizacji. Sieć wykrywająca segmenty przewyższa istniejące metody state-of-the-art pod względem skuteczności wykrywania i dokładności rekonstrukcji geometrii sceny. Zaproponowane deskryptory pozwalają na pobranie mniejszej liczby kandydatów do dopasowania niż te oparte na histogramach kolorów. Dzięki licznym usprawnieniom i oryginalnym rozwiązaniom PlaneLoc2 osiąga lepsze wyniki niż inne systemy globalnej lokalizacji, uzyskując wysoki współczynnik rozpoznawania pozycji bez błędnych rozpoznań (rozpoznań fałszywie dodatnich).

Acknowledgements

I would like to thank my supervisor Prof. Piotr Skrzypczyński and my auxiliary supervisor Prof. Dominik Belter for their help and many insightful suggestions. Thanks should also go to the Polish National Science Centre (NCN) and the European Commission for funding my PRELUDIUM and TERRINet projects. I am also thankful to my other work colleagues for hours of productive discussions. Lastly, I would like to recognize my family and friends that supported me throughout the process and listened to my complaints.

Abbreviations

AHRS	**A**ttitude and **H**eading **R**eference **S**ystem
BRIEF	**B**inary **R**obust **I**ndependent **E**lementary **F**eatures
CAD	**C**omputer **A**ided **D**esign
DNN	**D**eep **N**eural **N**etwork
FAST	**F**eatures from **A**ccelerated **S**egment **T**est
GPS	**G**lobal **P**ositioning **S**ystem
LiDAR	**Li**ght **D**etection **A**nd **R**anging
ORB	**O**riented FAST and **R**otated **B**RIEF
PDF	**P**robability **D**ensity **F**unction
RANSAC	**RAN**dom **SA**mple **C**onsensus
RMSE	**R**oot **M**ean **S**quared **E**rror
SIFT	**S**cale-**I**nvariant **F**eature **T**ransform
SLAM	**S**imultaneous **L**ocalization **A**nd **M**apping
VO	**V**isual **O**dometry

Notation

x_t	agent state at time t
z_t	measurement at time t
u_t	control command at time t
$\mathbf{K}$	camera matrix with intrinsic parameters
$\mathbf{R}_{l,g}$	rotation matrix from the local to the global frame of reference
$\mathbf{p}_g$	3-D point expressed in the global frame of reference
$\mathbf{t}_{l,g}$	translation vector from the local to the global frame of reference
$\mathbf{q}_l$	2-D point expressed in the local camera frame of reference
$\mathbf{0}$	vector with zeros
$\boldsymbol{\pi} = (n_x, n_y, n_z, -d)^T$	plane equation with the normal vector equal to $(n_x, n_y, n_z)^T$ and the distance to the origin equal to d

Please note that each publication included in this thesis has its own notation.

Contents

Chapter 1

Introduction

1.1 Motivation

The enormous development in the area of assisting technologies created an ever-rising demand for precise and robust localization systems. For example, a personal navigation system that can guide a person to a specific room in a large public building or shopping mall has to know the position of the user to compute navigational instructions. Moreover, virtually every autonomous agent must be equipped with the ability to localize itself, whether it is a mobile robot for inspecting oil rigs or an augmented reality system used for 3-D design [Cadena et al. 2016]. It is predicted that the global share of assisting technologies will be rising [World Health Organization 2019] and that new areas of applications will be available as the technology becomes more reliable. With aging population, autonomous medical assistants will be at hand, as well as autonomous workers in small- and medium-sized companies for performing tedious and repetitive tasks. However, all these applications require a robust and reliable localization technology. It has to work despite varying conditions, such as lighting, in all types of environments, but what is more important, it has to be able to determine whether it has lost tracking and be able to recover from such a situation. Unfortunately, contrary to the outdoor case, where satellite navigation systems greatly alleviated the problem, localizing indoors still poses many challenges, especially with determining the position with respect to a single global frame of reference.

Usually, the localization problem is solved in a recursive manner, where consecutive poses are computed using the assumption that the previous ones are known with sufficient certainty. However, there are many cases where that assumption is not true. An example is the initialization, when the starting pose is unknown, or an occluded sensor that prevents tracking from being continued. This problem is known as the kidnapped robot problem, because it resembles the situation of a kidnapped and blindfolded person who has the blindfold removed at a different place and has to determine their location [Thrun et al. 2005]. Moreover, it is not necessarily a special state of the agent or some kind of malfunction that prevents it from knowing precisely where it is. An example can be a mall with a large loop around its center part as in Fig. 1.1. When moving around the center part, the agent estimates its location by integrating displacement over time. However, with every displacement it adds a little error to the estimate, which accumulates as the trajectory grows. When it returns to the initial location, the accumulated error can be so large that the agent will not recognize this fact. In all these situations, it is crucial to embed the ability to determine the agent pose with respect to the global map, which is known as global localization [Bresson et al. 2017]. Global localization introduces a significant difficulty because of a much larger search space than in the case of recursive localization. Moreover, the fact that it was not well researched yet makes it even more challenging due to the scarce knowledge of the underlying problem [Cadena et al. 2016].

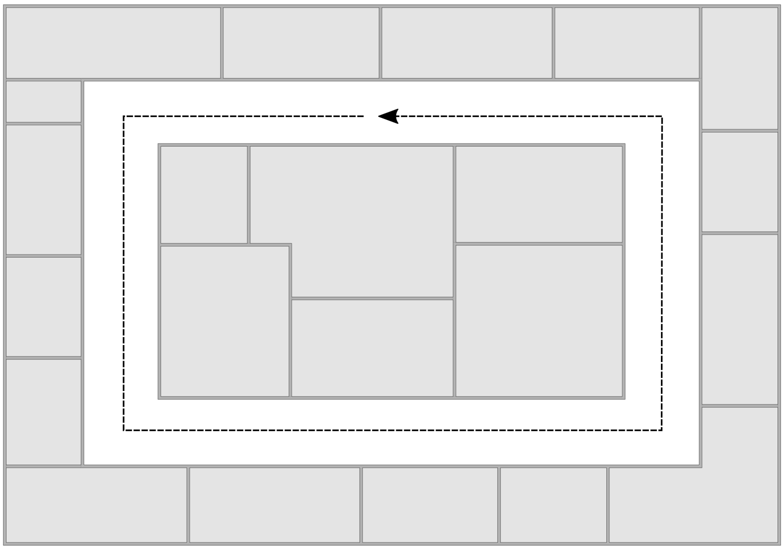

FIGURE 1.1: An example mall environment where traversing a trajectory denoted by a dashed line would cause error accumulation. To correct the error, loop closure using global localization is necessary.

Following [Thrun et al. 2005], localization can be formulated in terms of probability. The goal is to compute the probability distribution of the robot's state x_t at time t. In the case of recursive localization, the state is assumed to be complete, that is, knowing x_{t-1}, no prior variables provide additional information that would be helpful in estimating the next states. This property is also called the Markov's assumption. Additionally, all measurements $\mathbf{z}_{1:t}$ and all control commands $\mathbf{u}_{1:t}$ up to time t are known. Therefore, the distribution can be calculated as follows:

$$p(x_t|\mathbf{z}_{1:t}, \mathbf{u}_{1:t}) = \eta p(z_t|x_t) p(x_t|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) = \eta p(z_t|x_t) p(x_t|x_{t-1}, u_t) p(x_{t-1}|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1}), \quad (1.1)$$

where η is a normalizing factor. Contrary to this, global localization assumes no knowledge of the previous states, measurements, and control commands. In terms of probability it is equivalent to finding the following distribution:

$$p(x_t|z_t). \quad (1.2)$$

Without the prior knowledge of the history of the robot's activity, the problem is much more difficult, because the whole space of possible solutions has to be considered. There are no initial constraints as to where to begin the search (see Fig.1.2).

To be able to operate and respond to events in the environment, every autonomous agent has to have means of perceiving it and therefore has to be equipped with sensors. A natural choice for

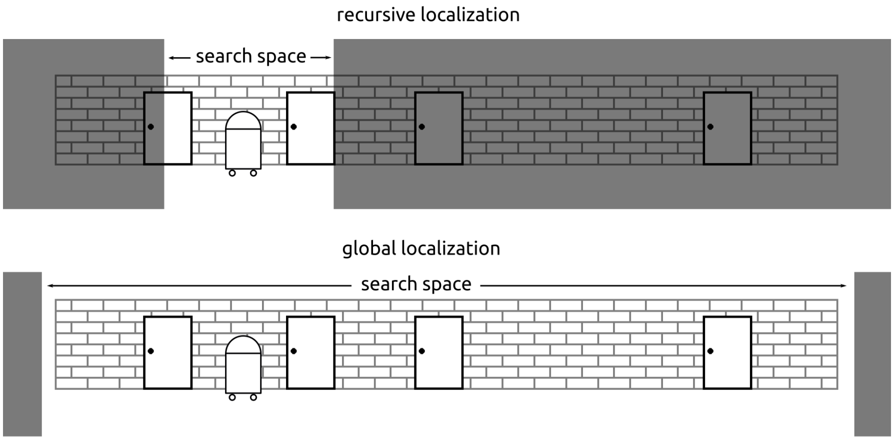

FIGURE 1.2: A schematic illustration of the difference between recursive localization, where solutions are sought only in the vicinity of the agent (upper part) and global localization, where solutions are sought in the whole considered world (lower part).

many autonomous agents is a camera. Cameras provide rich information about the environment, because they are able to capture many details, such as texture and shading. They also resemble human perception, and in many cases it is possible to recover geometric information about the scene in a way that humans do. However, recovering geometry is a challenging task, requiring strong prior knowledge about visible objects and structures. Humans gain this knowledge by years of observations and trials, and it is not easy to implement it in an artificial system. Recovering geometry is greatly simplified in the case of RGB-D cameras, where, usually, an infrared projector is used. The projector illuminates the scene with a structured light and by observing deformations of this pattern, the sensor estimates the distances of particular points from the sensor [Halmetschlager-Funek et al. 2019]. Although the geometry reconstructed with this technique is usually more accurate than the one reconstructed using a monocular camera, it deteriorates with increasing the distance up to the point where it is not reliable any more. This limits the effective range of RGB-D sensors to 4 - 6 m indoors, depending on conditions [Halmetschlager-Funek et al. 2019], and hinders their usage outdoors, where sunlight makes the projected pattern less visible. The range limitation is especially troublesome in the localization task as only the geometry of a nearby part of the scene can be precisely reconstructed, limiting the available information and narrowing the observable context of the scene. Another option to facilitate the recovery of scene geometry is to use a passive stereo camera. By observing an object in two images and knowing the stereo setup geometry, one is able to determine the distance to that object. However, it is not a trivial task, because of problems with matching the pixels in one image to the corresponding pixels in the second image, i.e. the pixels being the observations of the same physical point. Nonetheless, passive stereo cameras enable unambiguous geometry reconstruction [Smolyanskiy et al. 2018], and are relatively affordable, compared to e.g. 3-D LiDARs, which are at least an order of magnitude more expensive.

1.2 Challenges

The methods of computer vision were, until recently, mainly applied to appearance-based global localization problems, where output information is strictly topological, i.e. adjacency of considered views. Relatively few papers are tackling the problem of metric global localization, where the pose is described by coordinates in a six-dimensional space, related to the six degrees of freedom. The reason is the size of the space of possible solutions that needs to be searched. Because it is infeasible to exhaustively search this space, it is necessary to perform matching of observations. By matching two observations of the same physical object, one from the current view and one from the map, it is possible to impose constraints on the pose of the agent and reduce the search space. Usually, it is required to narrow down the search space to a single point or rather its vicinity, because of the measurement noise that generates inaccuracies. Most of the metric global localization systems use salient points, i.e. points corresponding to distinct features in an image (i.e. corners, intersections), as observations. It is well motivated by the ease of use of points in equations constraining the pose in the $SE(3)$ space:

$$\mathbf{K}(\mathbf{R}_{l,g}\mathbf{p}_g + \mathbf{t}_{l,g}) \times \mathbf{q}_l = \mathbf{0}, \tag{1.3}$$

where $\mathbf{K}$ is a calibrated camera matrix, $(\mathbf{R}_{l,g}, \mathbf{t}_{l,g})$ is the rotation and translation that transforms the 3-D point $\mathbf{p}_g$ expressed in the global frame of reference to the local frame of reference, and $\mathbf{q}_l$ is a 2-D point in the current image. These equations are linear and computing the pose by minimizing the quadratic error is straight-forward. Unfortunately, this approach has many disadvantages. A typical scene contains hundreds of salient points, which creates a very large number of possible matching combinations that cannot be exhaustively examined in a reasonable time. Moreover, because they occupy a small patch of the image, it is difficult to describe them to limit the number of potential match candidates. There are numerous papers on salient points description methods, including classic ones, such as SIFT [Lowe 1999] or ORB [Rublee et al. 2011], and learned ones, such as SuperPoint [DeTone et al. 2018] and DualRC-Net [Li et al. 2020]. The learned ones usually improve discrimination by using a hierarchical approach, where features are matched at different resolutions to include as much surrounding as possible. Nonetheless, unambiguously describing a small patch in the image is a difficult problem, because the appearance of a large surrounding changes with a view-point change. Moreover, salient points are sparsely scattered in the space, so a map built using these features is not particularly useful in other tasks, such as path planning, navigation, and visualization. A map composed from salient points has many holes (see Fig. 1.3a), especially in the areas where there is no texture, e.g. empty walls, so it is impossible to perform reliable collision detection required in path planning and navigation [Cadena et al. 2016].

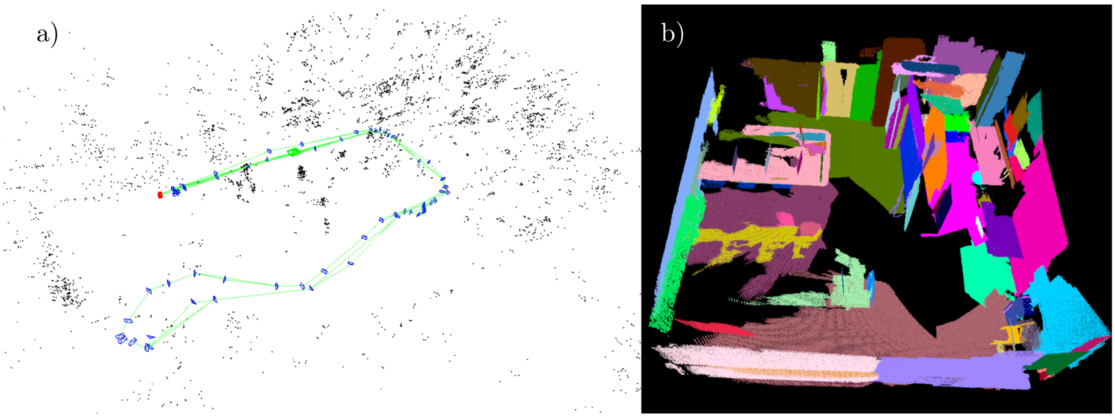

Figure 1.3: a) An example sparse salient point map from ORB-SLAM3 [Campos et al. 2021], and b) an example planar segment map from PlaneLoc [Wietrzykowski and Skrzypczyński 2019].

1.3 Contribution

The aim of this thesis is to remove the above-mentioned shortcomings, by using more complex reference objects for global localization. The author chose planar segments, because they are abundant in man-made environments, and the pose constraints imposed by them are relatively easy to express using equations. Since this work tackles the problem of indoor global localization, it can be assumed that the environment is man-made and contains enough planes to perform global localization. As proven in [Wietrzykowski and Skrzypczyński 2019; Wietrzykowski 2022], planes supporting planar segments can be detected using RGB-D and passive stereo cameras and used to constrain the $SE(3)$ pose. What is more important, the solution to this problem can be found quickly, unambiguously, and using a minimal number of equations (it is enough to use three matched planes of which no two are parallel). Planar segments have spatial dimensions greater than zero, occupying a certain area, so they are easier to describe than points. One can describe not only the surrounding of the segments, as in the case of points, but also the appearance of the segment itself. Another advantage of planar segments is the fact that they are less numerous in a typical scene than salient points. Having fewer objects to match, it is easier to find correct associations, because of fewer potential combinations to examine. Other, more complex objects, such as whole items or pieces of furniture, would be even better to describe, but geometrical constrains imposed by them are more difficult to exploit and additional information, such as CAD models, is necessary [Salas-Moreno et al. 2013]. Therefore, the author considers planar segments as a good compromise between descriptiveness and ease of exploitation of the geometric properties.

All the above-mentioned reasons induce that the problem of metric global localization in indoor environments is important and planar segments detected using RGB-D or passive stereo cameras are a promising choice for reference objects. Thus, the aim of this work is to research a novel approach to the problem exploiting these ideas, and the main thesis is as follows: **Local, partial, and uncertain cues from planar segment features allow to build a probability density function describing the global metric pose of an agent in a man-made environment.** Auxiliary theses can be formulated as follows:

- By considering many small sets of local geometric features in kernel density estimation it is possible to build a function with the maximum corresponding to the global pose of an agent.
- Observations of planar segments in man-made environments enable to determine the pose of an agent with six degrees of freedom with respect to a predefined map of planar segments.
- Deep neural network facilitates detection and description of planar segments using a passive stereo camera.

This work improves upon the existing methods by proposing novel solutions to the problems arising in the global localization pipeline that are crucial for the overall performance and reliability. Proper tailoring of the algorithms used in consecutive steps helps to achieve good results, therefore the contribution of this work can be summarized as follows:

- A framework for global localization and a probabilistic inference algorithm for reasoning about the pose of an agent in $SE(3)$, expressed with 6 degrees of freedom [Wietrzykowski and Skrzypczyński 2017]. The algorithm uses many weak and incomplete cues to build a probability density function describing the whereabouts of the agent. The cues have the form of minimal sets of matched observations of planar segments that enable pose retrieval. As a result, it is possible to quickly generate many pose hypotheses and use them in a Gaussian kernel representation of the PDF.
- A method for building a map of the environment that is composed of planar segments [Wietrzykowski 2022]. The method properly handles new observations by inserting new objects, merging existing ones, and deleting incorrect ones. It is based on views and instead of explicitly merging point clouds representing different observations it stores separate information about the observations from different viewpoints. This reduces computational burden of the merging procedure and enables proper uncertainty propagation.
- A system for detecting planar segments using a stereo camera that precisely recovers geometric information required in the task of global localization [Wietrzykowski and Belter 2022]. The system utilizes the concept of DNN that has proven to be effective in this type of tasks, where rich data is available but the structure of the underlying problem remains unknown [Sünderhauf et al. 2018]. Due to the use of a camera-agnostic representation of normal vectors and a segmentation method suitable for the employed DNN architecture, it outperforms the state-of-the-art methods and achieves a sufficient accuracy for global localization.
- A method for describing planar segments using a DNN [Wietrzykowski and Skrzypczyński 2021; Wietrzykowski 2022]. The descriptors reduce the computational complexity of matching by reducing the number of potential candidates that need to be considered to find a correct match. The method mostly exploits existing DNN layers, adding only a few to process the existing latent representation and a few necessary during training.
- Datasets used to train and evaluate detection, geometry reconstruction, and localization methods. The datasets are made publicly available to benefit the community and to enable verification of the presented results. The first dataset is PUT RGB-D/Workshop

collected in a workshop at Poznan University of Technology that contains reference pose information and RGB-D images [Wietrzykowski and Skrzypczyński 2017]. The second one is the synthetic SceneNet Stereo dataset that includes calibrated stereo images, along with reference poses, reference depth maps, and reference surface normal vectors [Wietrzykowski and Belter 2022]. The last one is the real-world TERRINet dataset that also contains calibrated stereo images, reference poses, and reference depth maps [Wietrzykowski and Belter 2022].

1.4 Publications guide

This thesis is based on a series of publications that present incremental development of a global localization method. The relations between articles and how they constitute a complete research project are described in this section. The dissertation includes articles presented at top-tier conferences, such as the International Conference on Intelligent Robots and Systems (IROS) and the European Conference on Mobile Robots (ECMR), and published in renowned journals, such as Robotics and Autonomous Systems (RAS) and Robotics and Automation Letters (RA-L).

1. Jan Wietrzykowski, “On the Representation of Planes for Efficient Graph-based SLAM with High-level Features”, Journal of Automation Mobile Robotics and Intelligent Systems,10 (3), 2016, pp. 3-11. [Wietrzykowski 2016]

As mentioned in Sec 1.1, to recover a pose in the global localization task, one has to know the positions of objects or features of reference. In the case of 3-D points it is straightforward by providing X, Y, and Z coordinates. However, in the case of planes it is more difficult, because the representation using a normal vector and a distance to the origin $\boldsymbol{\pi} = (n_x, n_y, n_z, -d)^T$ is not minimal. This representation is the easiest to use in transformations and to impose geometric constraints, therefore it not being minimal is a major drawback, because a minimal representation is crucial when optimization is performed, as in localization and SLAM systems. The work in [Wietrzykowski 2016] tests suitability of two plane representations to provide geometric constraints to localize an agent. The first one is based on $SE(3)$ pose with a covariance matrix describing constrained and not constrained directions, and the second one is based on a quaternion that encodes all values of $\boldsymbol{\pi}$, proposed by Kaess [Kaess 2015]. The work focuses on a pose and feature optimization backend using the g^{2}o framework [Kümmerle et al. 2011]. To test the representations in the context of optimization, a number of scenarios are simulated with varying strength of constraints in certain directions and the accuracy of localization is measured. The results suggest that minimal representation performs better when constraints are weak, which is a common situation in real-world applications. This representation is used in [Wietrzykowski and Skrzypczyński 2017] to assess whether two planes are similar and could be observations of the same planar segment. However, as explained in [Wietrzykowski and Skrzypczyński 2019], the major drawback of this representation is the different nature of rotational (normal vector) and translational parameters, which makes the comparison unstable in the presence of noise. For this reason, the comparison of two planar segments in [Wietrzykowski and Skrzypczyński 2019; Wietrzykowski 2022] is done using point-to-plane metrics.

2. Jan Wietrzykowski, Piotr Skrzypczyński, "A Probabilistic Framework for Global Localization with Segmented Planes", European Conference on Mobile Robotics, pp. 1-6, 2017. [Wietrzykowski and Skrzypczyński 2017]

The quaternion-based minimal representation evaluated in [Wietrzykowski 2016] is used in [Wietrzykowski and Skrzypczyński 2017], where a probabilistic framework for global localization was introduced. This article proposes a novel global pose inference method that uses many small sets of geometric features to build a PDF in the 6-D space of the agent pose. This idea is at the core of the global localization method presented in this thesis and is a basis for the localization algorithms presented in [Wietrzykowski and Skrzypczyński 2019; Wietrzykowski 2022].

The observation that planar segments are less numerous than keypoint features, recognized also in [Fernandez-Moral et al. 2013; Taguchi et al. 2013], enables a different approach to reference object matching. If there are fewer potential matches, it is possible to consider more combinations. In this article, all plausible triplets of potential matches are examined. As a result, the space of possible solutions is exhaustively searched and the pose supported by the majority of weighted hypotheses is found. To evaluate the performance of pose recognition, the ElasticFusion [Whelan et al. 2015] is used to build point clouds representing the local scene view (local map) and the global map. The point clouds are then segmented using supervoxel clustering and the resultant planar patches are merged into larger segments. The performance is measured using the rate of correct and incorrect pose recognitions. It is worth noting that including even one incorrect pose recognition can significantly deteriorate the pose estimate in SLAM and navigation tasks. Therefore, it is of utmost importance to eliminate such recognitions. The data used during evaluation was collected in a workshop and made publicly available as the PUT RGB-D/Workshop dataset. It includes reference information about poses captured using OptiTrack motion capture system. With proper parametrization, the proposed method achieves a high recognition rate without incorrect recognitions.

The contribution of the author is the main idea, implementation, and preparation of Sec. I, III, IV, V, VI of the article.

3. Jan Wietrzykowski, Piotr Skrzypczyński, "PlaneLoc: Probabilistic global localization in 3-D using local planar features", Robotics and Autonomous Systems, vol. 113, pp. 160-173, 2019. [Wietrzykowski and Skrzypczyński 2019]

The method from [Wietrzykowski and Skrzypczyński 2017] is extended in [Wietrzykowski and Skrzypczyński 2019] by adding a completely new component responsible for building and managing the map of planar segments, making it a standalone system that does not rely on the ElasticFusion [Whelan et al. 2015]. The local and the global map can therefore be built using the same pipeline, enabling extension into a SLAM system. The article also introduces a new measure of distance between planar segments that avoids the problems with the different nature of rotational and translational plane parameters. The solution is compared with the state-of-the-art salient point relocalization mechanism from the ORB-SLAM2 [Mur-Artal and Tardós 2017], based on the DBoW2 algorithm [Gálvez-López and Tardós 2012]. While the ORB-SLAM2 yields slightly higher recognition rates, they are concentrated in short fragments of the trajectory, where enough salient features are visible. PlaneLoc, on the other hand, is able to localize even when only textureless walls are visible. However, during this research, three main problems were identified that had to be addressed before further development:

- Planar segments further away than 4 m are not used due to the limited range of the RGB-D sensor. This leads to discarding a large amount of valuable information about the scene. The problem is addressed in [Wietrzykowski and Belter 2022] by introducing a DNN-based segment detector that exploits a passive stereo camera.
- The appearance descriptors of planar segments used to retrieve candidate matches are based on color histograms and do not discriminate well between segments. This forces considering more potential matches in order to include the correct ones. A new description method is proposed in [Wietrzykowski and Skrzypczyński 2021] to mitigate this issue.
- The pose retrieval algorithm assumes that planar segments are infinite planes and produces implausible solutions that have to be verified. A pose retrieval procedure that avoids this problem is described in [Wietrzykowski 2022].

The contribution of the author is the main idea, implementation, and preparation of Sec. 2, 3, 4, and 5 of the manuscript.

4. Jan Wietrzykowski, Dominik Belter, "Stereo Plane R-CNN: Accurate scene geometry reconstruction using planar segments and camera-agnostic representation", IEEE Robotics and Automation Letters, vol. 7(2), pp. 4345-4352, 2022. [Wietrzykowski and Belter 2022]

To extend the range of sensing and include distant planar segments, a different sensor had to be used and a new detection method had to be developed. RGB-D sensors have a limited effective range of 4-6 m [Halmetschlager-Funek et al. 2019] and it is not possible to accurately recover the geometry of planar segments that are outside of this range. Therefore, the author resorted to a passive stereo camera that has a larger effective range and is not as expensive as a LiDAR. Unfortunately, the stereo-estimated depth has many holes and fluctuations that make geometric scene segmentation impossible using the methods of [Wietrzykowski and Skrzypczyński 2017, 2019]. The holes and fluctuations also make classic plane fitting using RANSAC unreliable. The above-mentioned problems are addressed in [Wietrzykowski and Belter 2022] by introducing a DNN that detects planar segments in a single image and recovers their geometry using a cost volume created from a pair of stereo images. The DNN was trained on the synthetic SceneNet Stereo dataset introduced in this article, and tested on the real-world TERRINet dataset collected during the TERRINet project[1].

The detection module of the DNN is based on the Plane R-CNN [Liu et al. 2019] system that in turn uses the Mask R-CNN architecture [He et al. 2017]. By improving the plane segmentation procedure to suit the Mask R-CNN architecture, the proposed method achieves better detection results. The detection performance is evaluated using a geometry-based measure to mitigate the ambiguity of scene segmentation into planar segments. The overall score for the proposed method is better than for the baseline, with superior performance observed for almost all segment

[1] This dataset was collected during the author's visit to LAAS-CNRS in Touluse, within the TERRINet project funded by EU H2020 under GA No.730994

sizes. An improved geometry reconstruction performance is achieved by proposing a novel DNN architecture based on a cost volume and by using a camera-agnostic representation of normal vectors. The use of a cost volume for normal estimation was inspired by the work of [Kusupati et al. 2020]. To assess the quality of geometry reconstruction, two measures are used. The first one is the root mean square (RMS) error of the depth reconstruction accuracy, and the second one is the RMS error of the normal vector reconstruction accuracy. In the case of both measures, Stereo Plane R-CNN performs better, with results of normal vectors estimation being better by a large margin than all other solutions. The improved results for both detection and geometry reconstruction enable using a passive stereo camera for the task of global localization in [Wietrzykowski 2022].

The contribution of the author is the main idea, DNN architecture, implementation, and preparation of Sec. I, II, III, IV, V of the article.

5. Jan Wietrzykowski, Piotr Skrzypczyński, "On the descriptive power of LiDAR intensity images for segment-based loop closing in 3-D SLAM" IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 79-85, 2021. [Wietrzykowski and Skrzypczyński 2021]

The problem of appearance descriptors is addressed in [Wietrzykowski and Skrzypczyński 2021], where a method of training a DNN to produce descriptors of segments is proposed. This work extends the SegMap [Dubé et al. 2020] system by adding an appearance descriptor that is used during global localization. The SegMap is a LiDAR SLAM system that uses segments of point clouds as reference objects, therefore adding an appearance descriptor is not trivial. The author utilizes intensity readouts from the LiDAR and treats them as an image in order to add appearance information. Despite a different source of data than with camera-based indoor global localization, the goal is the same, i.e. to describe a segment that occupies a fragment of the image and then to find the corresponding segment in the global map. The source of data and the architecture of the DNN can vary, depending on the specific problem, but the vital part is the supervision of learning. When descriptors are being learned, they have no target values. The only information is whether two segment observations should be close to each other in the descriptor space (they are observations of the same segment) or distant from each other (they are observations of different segments). This problem can be solved in many ways, i.e. using triplet loss [Schroff et al. 2015], but most of them require a large database of samples to enable large batches with multiple samples of the same segment and/or computationally expensive data mining. A different approach to training the descriptors of geometry, used also in [Dubé et al. 2020], is to build a DNN that will classify the segment observations, such that each segment will be assigned to a separate class. Then, every observation of the same segment should be classified as the same class. To obtain a descriptor from such an architecture, a latent representation can be extracted from the DNN, as in Fig. 1.4. A similar architecture and the same supervision method is also used in [Wietrzykowski 2022]. To evaluate the performance of the descriptors, a *rank* is computed for each observation. The *rank* is the number of neighbors from the database of all descriptors that have to be fetched to get a correct match. The proposed solution achieves lower *ranks* for all test sequences and all sizes of segments than the baseline solution of SegMap. For bigger segments, in more than 50% of cases only one nearest neighbor is needed to get a correct match. Moreover, the influence on global localization is evaluated using the same metric as in [Wietrzykowski and Skrzypczyński 2017, 2019; Wietrzykowski 2022], i.e. the correct recognition rate. Using an improved matching procedure, new descriptors achieve better results for sequences gathered using the Ouster OS1-64 sensor. The results for sequences where the Velodyne HDL-64 sensor is used are inconclusive, which can be attributed to a poor quality of the intensity images.

The contribution of the author is the main idea, DNN architecture, implementation, and preparation of Sec. I, III, IV, V of the article.

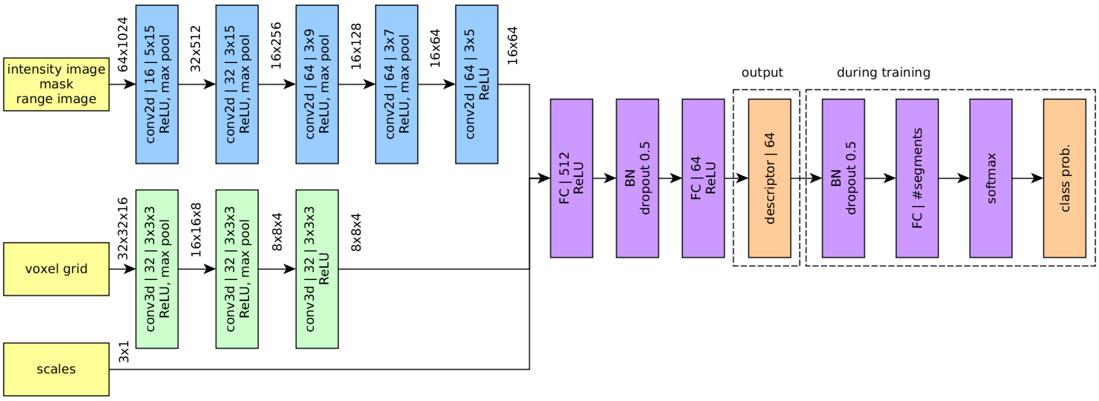

Figure 1.4: Architecture of the DNN describing segments in [Wietrzykowski and Skrzypczyński 2021]. The part in the right dashed rectangle is used to supervise learning and is later removed.

6. Jan Wietrzykowski, "PlaneLoc2: Indoor global localization using planar segments and passive stereo camera", IEEE Access, 2022. [Wietrzykowski 2022]

The work presented in [Wietrzykowski 2022] addresses the issues identified during development of the PlaneLoc [Wietrzykowski and Skrzypczyński 2019] and introduces the PlaneLoc2. It draws inspiration from [Wietrzykowski and Skrzypczyński 2021] to propose an improved appearance descriptor and a method to train the DNN that produces this descriptor. The appearance descriptor module is added to a properly adapted DNN from [Wietrzykowski and Skrzypczyński 2021], and together they constitute a detection module. Moreover, a new view-based map and a new view-based pose retrieval procedure are introduced that better suit the characteristics of a passive stereo camera. The solution is evaluated by comparing the correct pose recognition rate with the state-of-the-art ORB-SLAM3 [Campos et al. 2021] and HLoc [Sarlin et al. 2019] systems. Among the cases without incorrect recognitions, the PlaneLoc2 achieves the best results and also does not produce any incorrect recognitions for all cases. Additionally, the appearance descriptors are evaluated by comparing them to ones based on color histograms, used in the PlaneLoc. The same as in [Wietrzykowski and Skrzypczyński 2021], *ranks* are computed for segments of various sizes. For all sizes, the learned descriptor outperforms the hand-crafted one, proving its suitability for global localization.

The PlaneLoc2 concludes the research presented in this dissertation and proves the main thesis, along with the auxiliary theses. The article presents a complete global localization system that can be used for other tasks, such as SLAM and navigation. The system uses planar segments as reference objects that are detected using a DNN. During pose inference, it considers many small sets of matched segments to build a PDF that describes the 6-D pose of an agent.

Chapter 2

Publications

On the Representation of Planes for Efficient Graph-based SLAM with High-level Features

Submitted: 18th July 2016; accepted: 16th August 2016

Jan Wietrzykowski

DOI: 10.14313/JAMRIS_3-2016/18

Abstract:

Despite the fact, that dense SLAM systems are extensively developed and are getting popular, feature-based ones still have many advantages over them. One of the most important matters in sparse systems are features. The performance and robustness of a system depends strictly on the quality of constraints imposed by feature observations and reliable matching between measurements and features. To improve those two aspects, higher-level features can be used, and planes are a natural choice. We tackle the problem of plugging planes into the g^{2}o optimization framework with two distinct plane representations: one based on a properly stated SE(3) parametrization and one based on a minimal parametrization analogous to quaternions. Proposed solutions were implemented as extensions to the g^{2}o framework and experiments that verify them were conducted using simulation. We provide a comparison of performance under various conditions that emphasized differences.

Keywords: *SLAM, features, plane parametrization, graph-based optimization*

1. Introduction

The simultaneous localization and mapping (SLAM) problem has to be solved whenever a mobile robot explores unknown environment. It can be a scenario of exploring a disaster site or a previously unvisited building. A variety of potential applications fosters development of new SLAM solutions and improvement of the existing ones. Particularly interesting is the domain of 3D SLAM systems based on affordable depth sensors, such as Kinect, because of their high availability, low price and ability to provide rich information about the environment [14]. Despite the recent growth of the number of dense SLAM systems, feature-based solutions still outperform them with respect to the precision of camera motion estimation and real-time performance [12]. The main component in feature-based systems are sparse features. Features have to provide enough information to determine the sensor/robot position relatively to an existing map. They have to be distinctive enough to prevent wrong associations between the new observations and the map. As the state estimation techniques, either filtration-based, like EKF [19], or optimization-based [11] are not suited for handling incorrect feature associations, the features in 3D SLAM have to be chosen carefully to fulfill those requirements.

Until recent, most systems were based on photometric point features, such as SURF [1] or ORB [15] or their geometric counterparts extracted from point clouds [14]. They are easy to compute and manage, but constrains produced by them are often inaccurate and they can be easily mismatched, which is a major issue. It is caused by the fact that point features are computed from a small local patch of the photometric or depth image, where the pixel values depend on many factors, such as lighting, camera exposition parameters or the depth range. A solution to this problem is to use higher level geometric features, whose positions relative to the sensor can be precisely determined from more global data. It is expected that features that describe spatially extended structures of the scene will be more distinctive and repeatable when re-observed by the sensor. A natural extension to point features are edge or plane features. The planes are particularly interesting, as they commonly exist in man-made environments, such as building interiors, and can be easily detected and isolated using a Kinect-like depth sensor. Walls, ceiling and floor are examples of large planar segments that can be used in localization and mapping. Due to the relatively small number of detected planes in a typical environment, they can be also easily matched between consecutive frames in the data stream.

Beside the issues related to the front-end part of a modern, optimization-based SLAM system [2], that deals with processing of the measurements and determination of measurement-to-object associations, attention has to be paid to the back-end. The back-end handles an optimization process that finds the positions of robot and features that minimizes certain criterion, given measurements and measurement-to-object associations. Among many such systems, particularly interesting are the factor-graph-based libraries, because of their flexibility and intuitive problem formulation. Thus, in section 5 we propose an extension to the popular factor graph g^{2}o back-end system [11] in the form of a new constraint edge and a corresponding feature vertex. The extension enables a fast and accurate optimization of pose-to-plane constraints by means of a minimal parametrization of the planes. We also compare the new approach to a simplified solution based on an overparametrized representation using the standard g^{2}o edges and vertices. This simplified solution is presented in section 4. We show at first that using the standard vertices and constraints available in g^{2}o is inefficient for planar features, and then we compare the two solutions proposed in the paper by applying the Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) metrics [17], widely used in the SLAM research community. As this is a preliminary work on SLAM based on high level features, we focus on the optimization backend and conduct simulations using a simplified system that lacks a real front-end for the extraction and matching of planar features. Similarly to the approach introduced in [3] we replace the front-end by a simulation that allows us to control the uncertainty of measurements by adding Gaussian noise and to control the number and location of features (i.e. planes in our case) in the environment.

2. Related Work

Since the introduction of the Kinect, that started an era of cheap depth sensors, many 3D SLAM solutions using depth and visual information emerged.

Pixel intensities and depth measurements are directly used in dense SLAM systems, like the one by Kerl *et al.* [10]. A motion between two consecutive frames is estimated by minimizing a difference between a predicted and an actual measurement in both, photometric and depth, domains. A large scale system, using a stereo camera instead of a depth sensor, is presented by Engel *et al*. [5]. A transformation between camera pose at two different frames can be also computed using iterative closest point, as in [13], where Kinect sensor is used to map and track dense surfaces. Another approach is to extract point features and use a sparse representation of measurements, as in the work by Belter *et al.* [2].

One of the earliest attempts to use planes as features was by Weingarten and Siegwart [19]. They adopted SPmodel [4] to represent planes and harnessed Extended Kalman Filter (EKF) to update the SPmap containing robot pose and feature locations. The SPmodel (symmetries and perturbation model) uses a probabilistic representation of the imprecision in the location of features, and the theory of symmetries to represent the partiality of the uncertainty due to the parametrization of the feature. Unfortunately, the plane representation is overparametrized and EKF-SLAM cannot exploit the sparsity of feature observations, in contrast to our solution.

Salas-Moreno *et al.* [16] proposed a method to densely map an environment with usage of bounded planes and surfels. Planar regions are refined and extended during camera's movement and can serve as a display for an augmented reality content.

A solution based on both, point and plane features was presented by Taguchi *et al.* [18]. They use a general form equation to parametrize planes, which is a non-minimal representation, and a sparse linear solver in the Gauss-Newton iterative optimization algorithm. Error calculation between the estimated and the measured plane is accomplished by means of random sampling of the measurement points. A sparse solver is also employed in the work by Kaess [8], where a minimal representation of planes based on quaternions was introduced. Optimization is done by the iSAM algorithm [9].

A popular tool for graph-based optimization is the g^{2}o framework, that outperforms many other systems, including iSAM [11]. It is widely used in point-feature-based SLAM systems, such as those by Mur-Artal *et al.* [12] or Belter *et al.* [3]. Thus, the g^{2}o library was chosen as the framework we want to test for handling optimization of pose-to-plane constraints, and then extend by a new minimal representation of the planar features.

3. Problem Formulation Using Graph

The part of the SLAM problem related to the back-end operation is to find the camera and feature positions that best fit the collected observations. To solve this problem efficiently, a proper representation of the constraints is needed. One of the possibilities is to model the system as a factorized probabilistic equation:

$$p(\mathbf{x}|\mathbf{z}) = \frac{1}{Z}\prod_{a\in F}\Psi_a(\mathbf{x}_a,\mathbf{z}_a), \tag{1}$$

where $\mathbf{x}$ are random variables, $\mathbf{z}$ are measurement variables, that are observed, Z is a normalization constant, F is a set of factors, $\Psi_a(\mathbf{x}_a,\mathbf{z}_a)$ is a value of a factor a, $\mathbf{x}_a$ is a subset of random variables that the factor a depends on, and $\mathbf{z}_a$ is a subset of measurement variables that the factor a depends on. Factor $\Psi_a(\mathbf{x}_a,\mathbf{z}_a)$ is a function, usually based on the Gaussian distribution, that measures how likely the state of variables $\mathbf{x}_a$ explain measurements $\mathbf{z}_a$. Throughout this paper, we use the following form of factors:

$$\Psi_a(\mathbf{x}_a,\mathbf{z}_a) = \exp\left\{-\frac{1}{2}\mathbf{e}_a(\mathbf{x}_a,\mathbf{z}_a)^T\Omega_a\mathbf{e}_a(\mathbf{x}_a,\mathbf{z}_a)\right\}, \tag{2}$$

where $\mathbf{e}_a(\mathbf{x}_a,\mathbf{z}_a)$ is an error function and an information matrix is denoted by Ω_a.

The error function returns a vector of differences between measurement prediction $\mathbf{h}_a(\mathbf{x}_a)$ based on the state of variables $\mathbf{x}_a$ and an actual measurement $\mathbf{z}_a$:

$$\mathbf{e}_a(\mathbf{x}_a,\mathbf{z}_a) = \mathbf{h}_a(\mathbf{x}_a)\ominus\mathbf{z}_a, \tag{3}$$

where $\ominus$ is an operator that is a generalization of the subtraction operation, for example taking into account rotation ambiguities, defined depending on the representation. Dimensionality of the error vector depends on the type of measurement and its representation. In the case of minimal representation of planes it is 3, and in the case of SE(3) it is 6.

The problem can be presented using a probabilistic graphical model, as shown in Fig. 1. The robot and feature positions are encoded by subsets of variables organized in a proper representation, e.g. a translation vector and 3 imaginary components of an unit quaternion for SE(3). If the position i has SE(3) representation, then the set $\mathbf{x}_{vi}$ contains 6 variables. There is no difference between variables representing feature and robot positions, besides from their meaning. Factor is dependent on all variables that represent the robot and feature positions connected to it.

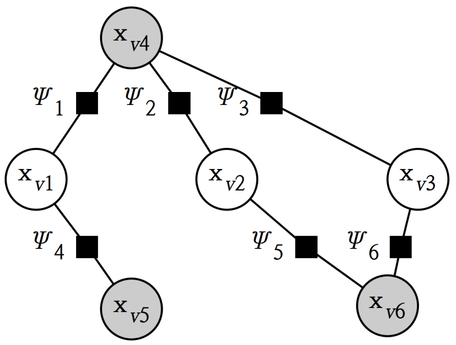

Fig. 1. Probabilistic graphical model representing the SLAM optimization problem. Variables encoding robot positions are marked as white circles, variables encoding feature positions are marked as gray circles, factors are marked as black squares and subset of variables that represent position i is denoted by $\mathbf{x}_{vi}$

This form emphasizes the sparsity of dependencies between the robot positions and feature positions. Properly exploiting the sparsity enables efficient optimization and is in the core of back-end systems development.

The optimization can be formulated as a process of finding values of variables $\mathbf{x}$ that maximizes the probability (1), denoted by $\mathbf{x}^*$. By taking the logarithm of probability it can be written as:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} \frac{1}{Z} \prod_{a\in F} \Psi_a(\mathbf{x}_a, \mathbf{z}_a) \tag{4}$$

$$= \arg\min_{\mathbf{x}} \sum_{a\in F} \mathbf{e}_a(\mathbf{x}_a, \mathbf{z}_a)^T \Omega_a \mathbf{e}_a(\mathbf{x}_a, \mathbf{z}_a) \tag{5}$$

$$= \arg\min_{\mathbf{x}} \sum_{a\in F} l_a(\mathbf{x}_a, \mathbf{z}_a). \tag{6}$$

Although, in general, the problem is not convex, an iterative method is used to find the optimal values of $\mathbf{x}$. The assumption is made, that initial guess is good enough not to cause a divergence of the algorithm. At every iteration step, the functions $l_a(\mathbf{x}_a, \mathbf{z}_a)$ are linearized in the currently estimated state of variables $\mathbf{x}$ and an optimal step is calculated, denoted by $\Delta\mathbf{x}^*$. The linearization is expressed by (we omit dependence on $\mathbf{z}_a$ to simplify notation as values of $\mathbf{z}_a$ are constant):

$$l_a(\mathbf{x}_a + \Delta\mathbf{x}_a) \tag{7}$$

$$= \mathbf{e}_a(\mathbf{x}_a + \Delta\mathbf{x}_a)^T \Omega_a \mathbf{e}_a(\mathbf{x}_a + \Delta\mathbf{x}_a) \tag{8}$$

$$\simeq \left[\mathbf{e}_a(\mathbf{x}_a) + \mathbf{J}_a(\mathbf{x}_a)\Delta\mathbf{x}_a\right]^T \Omega_a \left[\mathbf{e}_a(\mathbf{x}_a) + \mathbf{J}_a(\mathbf{x}_a)\Delta\mathbf{x}_a\right] \tag{9}$$

$$\begin{aligned} &= \mathbf{e}_a(\mathbf{x}_a)^T \Omega_a \mathbf{e}_a(\mathbf{x}_a) \\ &\quad + 2\mathbf{e}_a(\mathbf{x}_a)^T \Omega_a \mathbf{J}_a(\mathbf{x}_a)\Delta\mathbf{x}_a \\ &\quad + \Delta\mathbf{x}_a^T \mathbf{J}_a(\mathbf{x}_a)^T \Omega_a \mathbf{J}_a(\mathbf{x}_a)\Delta\mathbf{x}_a \end{aligned} \tag{10}$$

$$= c_a(\mathbf{x}_a) + \mathbf{b}_a(\mathbf{x}_a)\Delta\mathbf{x}_a + \Delta\mathbf{x}_a^T \mathbf{H}_a(\mathbf{x}_a)\Delta\mathbf{x}_a, \tag{11}$$

where $\mathbf{J}_a$ is a Jacobian matrix of the error function with respect to variables $\mathbf{x}_a$ in the current point. If we expand all vectors and matrices in equation (11) to include all $\mathbf{x}$ variables, the iteration step can be written as:

$$\Delta\mathbf{x}^* = \arg\min_{\Delta\mathbf{x}} \sum_{a\in F} l_a(\mathbf{x}_a, \mathbf{z}_a) \tag{12}$$

$$\simeq \arg\min_{\Delta\mathbf{x}} \sum_{a\in F} c_a + \mathbf{b}_a\Delta\mathbf{x}_a + \Delta\mathbf{x}_a^T \mathbf{H}_a \Delta\mathbf{x}_a \tag{13}$$

$$= \arg\min_{\Delta\mathbf{x}} c + \mathbf{b}\Delta\mathbf{x} + \Delta\mathbf{x}^T \mathbf{H}\Delta\mathbf{x}, \tag{14}$$

where $c = \sum_{a\in F} c_a$, $\mathbf{b} = \sum_{a\in F} \mathbf{b}_a$, and $\mathbf{H} = \sum_{a\in F} \mathbf{H}_a$. The $\Delta\mathbf{x}^*$ value is calculated using equation:

$$\mathbf{H}\Delta\mathbf{x}^* = -\mathbf{b}. \tag{15}$$

After finding $\Delta\mathbf{x}^*$, current estimate is updated according to formula:

$$\mathbf{x}^{i+1} = \mathbf{x}^i \oplus \Delta\mathbf{x}^{i*}, \tag{16}$$

where $\oplus$ is a generalization of addition operator, defined depending on the representation. Note that increments are computed by considering derivatives of the error function with respect to variables, therefore units, in which those variables are expressed, are irrelevant.

Iterations are performed until an optimal solution is found. Usually, algorithms like Gauss-Newton or Lavenberg-Marquardt are used in combination with sparse linear optimizers to solve equation (15). The sparsity is encoded in the $\mathbf{H}$ matrix, since only some values are non-zero (value at position (i, j) can be non-zero only if variables x_i and x_j are related by a factor).

The g^{2}o framework organizes probabilistic graphical models in a form of vertices and edges. Vertices are representing subsets of variables denoting robot or feature poses and have to determine a proper representation of those variables. Therefore, vertices also implement $\oplus$ operator suitable for chosen representation. Edges are analogues of factors and, as such, they bind vertices with measurements. Generally, edges can connect multiple vertices, but in our system they always connect two. The operation that has to be implemented in an edge is error calculation, therefore they implement $\ominus$ operation. Optionally, edges can also implement analytical calculation of the Jacobian matrices, which are by default computed numerically [11].

4. SE(3) Plane Representation

This section presents a simplified solution based on a SE(3) representation of planes. It was necessary to introduce some assumptions and simplifications to plug planes into overparametrized representation.

Usually, using depth sensors, a plane measurement is represented as a normal vector $\mathbf{n}$ in the sensor frame of reference and distance d to the sensor (hereinafter called camera for convenience). Some assumptions have to be made to convert this representation to the SE(3) one, since the number of such overparametrized representations is infinite. Hence, we assumed that a plane coordinate system has the following properties:

- The origin is located in the plane point nearest to a camera.

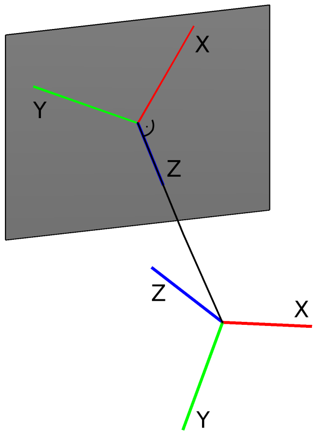

Fig. 2. A schematic view of assumptions about coordinate system of a plane

- The z axis is perpendicular to a plane.
- The x axis direction is determined by a cross product of the normal vector and the $[1,0,0]^T$ vector (if they are parallel, with $[0,1,0]$ vector). This assumption is only important to assure that the x axis will be parallel to the z axis.
- The y axis direction is determined by a cross product of the unit vectors in the z axis and the x axis directions.

Obviously, there is an ambiguity in the representation, and the global coordinates of a plane depend on the camera position (they won't be the same for different camera poses). It is caused by the fact that a plane is an object with 3 degrees of freedom (DOF), whereas the SE(3) representation has 6 DOF. Therefore, the frame of reference of an infinite plane can move freely along the x and y axes of this plane, and can rotate around it's z axis, which gives extra 3 DOF. The different placement of the origin cannot be avoided in a real-world scenario, but, as it will be shown, the difference has no effect on results thanks to a proper information matrix formulation. A schematic illustration of a plane coordinate system is presented in Fig. 2.

The standard g^{2}o SE(3) vertex represents the position in the form of a translation-quaternion (TQ) vector:

$$\mathbf{v} = \begin{bmatrix} t_x \\ t_y \\ t_z \\ q_x \\ q_y \\ q_z \end{bmatrix}, \qquad (17)$$

where t_x, t_y, t_z are Euclidean coordinates and q_x, q_y, q_z are imaginary components of an unit quaternion with the real component $q_w \geqslant 0$. In the SE(3) edge values $\mathbf{z}_a$ represent measurement of a transformation from the camera frame of reference to the plane frame and is also represented by a TQ vector. As $\mathbf{x}$ and $\mathbf{z}$ are just sets of numbers, we use $\mathbf{v}(\cdot)$ operator to indicate that they should be treated as a TQ vector. The symbols $\mathbf{x}_{ac}$ and $\mathbf{x}_{ap}$ denote subsets of $\mathbf{x}_a$ variables representing global camera and global robot position, respectively. Transformations between frames of reference are depicted in Fig 3.

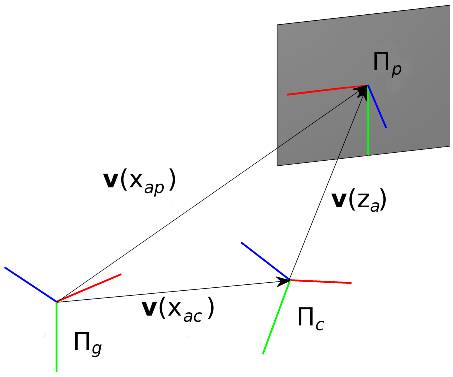
Π_p
$\mathbf{v}(\mathrm{x}_{ap})$
$\mathbf{v}(\mathrm{z}_a)$
$\mathbf{v}(\mathrm{x}_{ac})$
Π_g
Π_c

Fig. 3. Transformations between frames of reference. Π_g is a global frame of reference, Π_c is a camera frame of reference and Π_p is plane frame of reference

An error, introduced in the equation (3), is defined as follows:

$$\mathbf{e}_a(\mathbf{x}_a, \mathbf{z}_a) = \mathbf{v}(\mathbf{z}_a)^{-1}\left[\mathbf{v}(\mathbf{x}_{ac})^{-1}\mathbf{v}(\mathbf{x}_{ap})\right], \qquad (18)$$

where multiplication is a concatenation of transformations (not a matrix multiplication), $\mathbf{v}^{-1}$ is an inversion of the transformation $\mathbf{v}$, and $\mathbf{v}(\mathbf{x}_{ac})^{-1}\mathbf{v}(\mathbf{x}_{ap})$ can be interpreted as measurement prediction.

The error is defined in the plane's frame of reference, therefore an information matrix $\mathbf{\Omega}_a$ has to be defined in the same frame. In an overparametrized representation, such as the one considered here, the information matrix is particularly important. It should define large (theoretically infinite) uncertainty in the dimensions, in which the representation is ambiguous. If the i-th dimension is a surplus, then the element of $\mathbf{\Omega}_a$ at location (i,i) should be equal to zero. Unfortunately, the matrix constructed in such way would be rank deficient and impossible to invert. The inability to invert would discard a large number of optimization algorithms. Therefore, we decided to circumvent this limitation by inserting a very small number instead of 0. Another problem was how to specify an information matrix for rotation represented by a quaternion. To overcome the problem, we constructed a covariance matrix for the extended representation (including q_w value) in the form:

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_t & \mathbf{0}_{3\text{x}4} \\ \mathbf{0}_{4\text{x}3} & \mathbf{C}_{rq} \end{bmatrix}, \qquad (19)$$

where $\mathbf{C}_t$ was defined as follows:

$$\mathbf{C}_t = \mathrm{diag}(c, c, 1) \qquad (20)$$

and $\mathbf{C}_{rq}$ as follows:

$$\mathbf{C}_{rq} = \mathbf{J}_{qr}\mathbf{C}_{rr}\mathbf{J}_{qr}^T. \qquad (21)$$

Here $\mathbf{C}_{rr}$ is the covariance matrix for a rotation expressed by a 3×3 rotation matrix, defined as (for a row-major order of matrix elements):

$$\mathbf{C}_{rr} = \mathrm{diag}(c, c, 1, c, c, 1, 1, 1, 1) \qquad (22)$$

and $\mathbf{J}_{qr}$ is Jacobian matrix of the conversion from a rotation matrix to a quaternion at the identity point (derived from the equation converting rotation matrix representation to a quaternion representation):

$$\mathbf{J}_{qr} = \begin{bmatrix} 0 & 0 & 0 & 0.125 \\ 0 & 0 & 0.250 & 0 \\ 0 & -0.250 & 0 & 0 \\ 0 & 0 & -0.250 & 0 \\ 0 & 0 & 0 & 0.125 \\ 0.250 & 0 & 0 & 0 \\ 0 & 0.250 & 0 & 0 \\ -0.250 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.125 \end{bmatrix}^T. \quad (23)$$

In the above equations c is some large value that indicates that in this dimension a variance is infinite. We used $c = 1000$ which was a good compromise between accuracy and numerical stability. The final information matrix is the 6×6 upper-left corner part of the inverse of the $\mathbf{C}$ matrix:

$$\Omega_a = [\mathbf{C}^{-1}]_{6\times 6}. \quad (24)$$

The $\oplus$ operator is realized by multiplying a transformation represented by the current state of variables by a transformation expressed by the computed increment $\Delta\mathbf{x}^{i*}$:

$$\mathbf{v}(\hat{\mathbf{x}}_{av}^{i+1}) = \mathbf{v}(\hat{\mathbf{x}}_{av}^{i})\mathbf{v}(\Delta\mathbf{x}_{av}^{i*}). \quad (25)$$

The implementation was intended to be simple and demand small amount of work, so it was realized using standard g^{2}o edges and vertices. The additional advantage of this approach is that the g^{2}o framework implements analytical calculation of Jacobian matrices for standard classes. It was necessary to compute information matrix and implement a measurement simulation. The simulation was accomplished by adding Gaussian noise to the normal vector $\mathbf{n}$ components and a distance value d, and then calculating $\mathbf{v}(\mathbf{z}_a)$ using the previously defined assumptions about the coordinate system of a plane.

5. Minimal Plane Representation

To represent a plane only 3 values are required, since it is an object with 3 DOF. In this section we present a solution based on a representation that uses only 3 values and therefore is minimal. Nevertheless, using a normal vector and a distance requires 4 values: 3 components of $\mathbf{n} = [n_x, n_y, n_z]^T$ and d. The same problem occurs with a general plane equation:

$$p_1 x + p_2 y + p_3 z + p_4 = 0. \quad (26)$$

The solution is to normalize the general plane equation, so $\|\mathbf{p}\| = \|[p_1, p_2, p_3, p_4]^T\| = 1$ and restrict $p_4 \geqslant 0$. After doing so, only the first 3 components of the vector $\mathbf{p}$ are relevant, since the last one can be retrieved using formula:

$$p_4 = \sqrt{p_1^2 + p_2^2 + p_3^2}. \quad (27)$$

It is an analogue to an unit quaternion and all operations on quaternions can be transfered to this representation [8]. Therefore, it is a minimal representation without singularities, what makes it suitable for optimization purpose.

With the minimal representation, we used exponential and logarithm map to calculate the error and update current positions. An exponential map is a map from the Lie algebra to a Lie group and a logarithm map is a map in the reverse direction. The error calculation in an edge connecting a camera position and a plane position is performed using the logarithm map of quaternions:

$$\mathbf{e}_a(\mathbf{x}_a, \mathbf{z}_a) = \mathbf{q}\left[\mathbf{T}(\mathbf{x}_{ac})^T \mathbf{p}(\mathbf{x}_{ap})\right] \ominus \mathbf{q}(\mathbf{z}_a) \quad (28)$$

$$= \log\left\{\mathbf{q}\left[\mathbf{T}(\mathbf{x}_{ac})^T \mathbf{p}(\mathbf{x}_{ap})\right]^{-1} \mathbf{q}(\mathbf{z}_a)\right\}, \quad (29)$$

where $\mathbf{q}(\mathbf{x}_v)$ denotes a quaternion formed from variables $\mathbf{x}_v$, $\mathbf{T}(\mathbf{x}_v)$ is a homogeneous transformation matrix constructed from variables $\mathbf{x}_v$, and $\mathbf{p}(\mathbf{x}_v)$ is a plane equation based on variables $\mathbf{x}_v$. Note that transformation of a general plane equation from Π_g frame of reference to Π_c frame is expressed differently than the same transformation for a point. It is done by the equation ($\mathbf{T}_{i,j}$ denotes a homogeneous transformation matrix for a transformation from Π_i to Π_j):

$$\mathbf{p}_c = \mathbf{T}_{c,g}^{-T}\mathbf{p}_g \quad (30)$$

$$= \mathbf{T}_{g,c}^{T}\mathbf{p}_g. \quad (31)$$

The logarithm map is a 3 dimensional vector given by the equation:

$$\log(\mathbf{q}) = 2\frac{\cos^{-1}(q_w)}{\|\mathbf{q}_v\|}\mathbf{q}_v, \quad (32)$$

where $\mathbf{q}_v$ is an imaginary part of the quaternion $\mathbf{q}$.

Updates of variables are done using exponential maps for both, camera positions and plane positions. The update for planes is done using the following formula:

$$\mathbf{q}(\mathbf{x}_{ap}^{i+1}) = \exp\left[\omega(\Delta\mathbf{x}_{ap}^{i*})\right]\mathbf{q}(\mathbf{x}_{ap}^{i}), \quad (33)$$

where exponential map for a plane is defined as:

$$\exp(\omega) = \begin{bmatrix} \frac{1}{2}\sin(\frac{1}{2}\|\omega\|)\omega \\ \cos(\frac{1}{2}\|\omega\|) \end{bmatrix}. \quad (34)$$

The update for camera positions is done in a similar way, but instead of quaternions, operations are performed on TQ vectors:

$$\mathbf{v}(\mathbf{x}_{ac}^{i+1}) = \exp\left[\mathbf{d}(\Delta\mathbf{x}_{ac}^{i*})\right]\mathbf{v}(\mathbf{x}_{ac}^{i}). \quad (35)$$

The increment used in the above equation comprises a translational and a rotational part, same as the TQ vector:

$$\mathbf{d} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (36)$$

Denoting the result of the exponential mapping by:

$$\exp(\mathbf{d}) = \begin{bmatrix} \mathbf{t}_d \\ \mathbf{q}_d \end{bmatrix}, \tag{37}$$

the calculation can be done using equations:

$$\mathbf{t}_d = \mathbf{V}v \tag{38}$$

and

$$\mathbf{q}_d = \mathbf{q}(\mathbf{R}). \tag{39}$$

Note that, in the notation above, a quaternion is constructed from a rotation matrix, not from a 4-dimensional vector. The matrices $\mathbf{V}$ and $\mathbf{R}$ are given by equations:

$$\mathbf{V} = \mathbf{I} + \frac{1-\cos(\|\omega\|)}{\|\omega\|^2}[\omega]_\times + \frac{\|\omega\| - \sin(\|\omega\|)}{\|\omega\|^3}[\omega]_\times^2 \tag{40}$$

and

$$\mathbf{R} = \mathbf{I} + \frac{\sin(\|\omega\|)}{\|\omega\|}[\omega]_\times + \frac{1-\cos(\|\omega\|)}{\|\omega\|^2}[\omega]_\times^2, \tag{41}$$

where $[\omega]_\times$ is a skew-symmetric matrix:

$$[\omega]_\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \tag{42}$$

We implemented an extension to the standard set of g^{2}o edges and vertices by adding a vertex representing a plane position and an edge connecting camera position and plane position. As the camera position vertex we used a SE(3) vertex that uses exponential map, included in the framework. During experiments, measurements were simulated by adding Gaussian noise to the components of the normal vector $\mathbf{n}$ and to the distance d value. The quaternion representation was obtained by constructing a general plane equation, and then properly normalizing a vector of parameters. The vector had the following form:

$$p = \begin{bmatrix} n_x \\ n_y \\ n_z \\ -d \end{bmatrix}. \tag{43}$$

In the current version, the Jacobian matrix was computed numerically.

6. Experiments and Results

To experimentally evaluate the proposed models of planar features and the constraints related to them, we simulated motion of a camera in an empty room. As demonstrated in [3], such a simple experiment clearly reveals how the behavior of the optimization back-end depends on the parametrization of the uncertainty model of the features. The simulated front-end does not introduce any errors due to wrong feature associations or multiplicated features, thus the results are isolated from the qualitative errors that are unavoidable in a real front-end. In order to make the simulation maximally realistic as to the dynamics of the sensor motion we used an example trajectory from the *ICL-NUIM Office Room Dataset* [7] and inserted a virtual floor and two walls that were always observed by the sensor. An overview of the simulated environment is shown in Fig. 4. Normal vectors $\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3$ and distance values d_1, d_2, d_3 were the common source of information for both parametrizations. All representations were obtained from those values with added Gaussian noise of the standard deviation equal to 0.01.

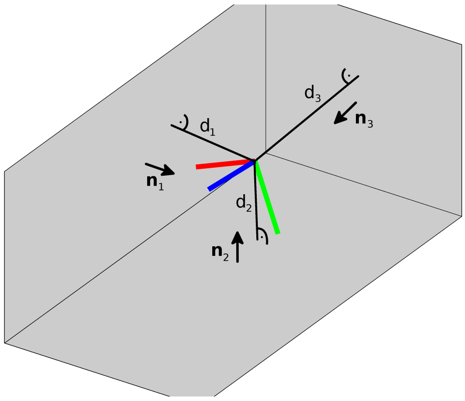

Fig. 4. An overview of the simulated environment. Normal vectors $\mathbf{n}_1$, $\mathbf{n}_2$, $\mathbf{n}_3$ and distance values d_1, d_2, d_3 were noised measurement values

Optimization was done in batch mode. First, all vertices, along with their initial position estimations, and edges were added to the graph and than the optimization process was triggered. In both cases we used Gauss-Newton algorithm with the Preconditioned Conjugate Gradient (PCG) linear solver. The number of iterations was limited to 100, although in all tests the algorithm converged earlier. We tested when a change of the error value l between iterations will drop below $10^{-6}l$. For the SE(3) representation it was after the 26-th iteration and took 0.343 s. In the case of the minimal representation, it happened after the 9-th iteration and took 0.257 s. Both results enable real-time operation, but the optimization with the minimal representation converges faster and needs fewer iterations.

An initial guess to camera positions was obtained by simulating dead reckoning (e.g. visual odometry, as used in [2]). We calculated differences between consecutive poses in the ground truth trajectory, added noise to every difference and then build an odometry trajectory by stacking noised difference transformations. If $\tilde{T}_{i,i+1}$ is a noised transformation from ground truth trajectory pose i to pose $i+1$, then the odometry pose $i+1$ is expressed by:

$$O_{i+1} = O_i \tilde{T}_{i,i+1}. \tag{44}$$

First, we investigated a behavior of the system when the information matrix $\mathbf{\Omega}_a$ for SE(3) representation was set to identity to highlight that using this representation for planes is not obvious. Setting the information matrix to identity is a common practice, but should be done carefully, in particular when the measurements or the state variables span over non-Euclidean manifold spaces [6]. Effects of neglecting

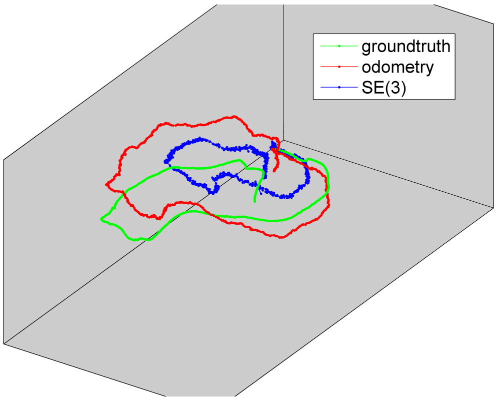

Fig. 5. Visualization of the SE(3) optimization results with the information matrices set to identity

Tab. 1. Results of the SE(3) and minimal representation optimizations with perpendicular planes

measure		SE(3)	minimal
RPE translational [m]	rmse	0.034	0.031
	mean	0.031	0.028
	median	0.029	0.026
	std	0.014	0.013
	max	0.111	0.107
RPE rotational [°]	rmse	1.001	1.139
	mean	0.923	1.049
	median	0.016	0.018
	std	0.389	0.446
	max	2.859	2.960
ATE [m]	rmse	0.021	0.017
	mean	0.019	0.016
	median	0.019	0.015
	std	0.008	0.007
	max	0.062	0.042

the partiality of uncertainty in planar features can be seen in Fig. 5. As expected, when the identity matrices are used, the optimized trajectory is a degenerated version of the ground truth one, because the least-squares minimization could not be constrained to the proper manifold.

The next experiment compared the SE(3) and minimal representations with properly set information matrices for the situation when the plane features were perpendicular each to the other. This "natural" configuration of walls in a room provides also the best constraints to the simple system under study, as there are similar constraints along each axis of the global coordinate system. Quantitative results are gathered in Tab. 1, while the estimated trajectories are visualized in Fig. 6. We apply the ATE and RPE metrics. ATE compares the distance between the estimated and ground truth trajectories, whereas RPE corresponds to the drift of the trajectory [17]. From the trajectories it is clearly visible that both solutions reconstructed the camera motion with small errors. The numeric results are slightly better for the minimal representation, but the differences are rather irrelevant.

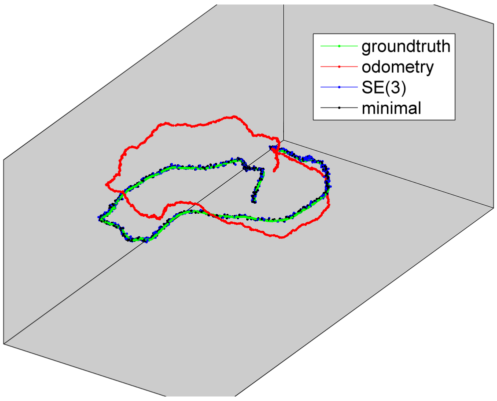

Fig. 6. Results of the SE(3) and minimal representation optimizations with perpendicular planes

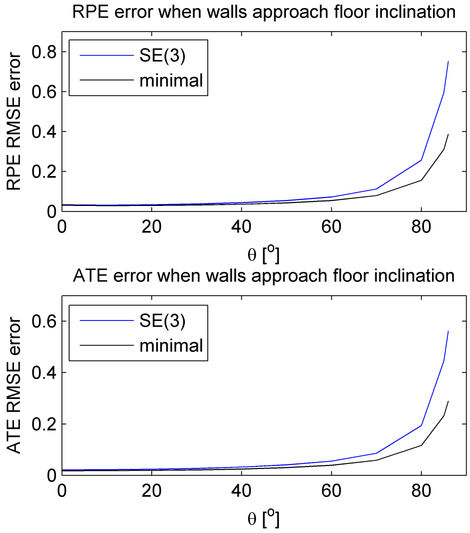

Fig. 7. Results of the SE(3) and minimal representation optimizations when walls approach floor inclination. The θ is an angle by which walls were tilted

Differences emerged with more challenging setups, in which walls were not perpendicular to the floor and each to the other. A dependency between the angle by which the walls were tilted and the errors in trajectory estimation is visible in Fig. 7. When the angle is small and the measurements of the positions of walls impose strong constraints, the error values are similar to the ones obtained in the previous experiment, but when the walls approach the floor inclination and are close to being parallel to the ground plane, the error for estimation with the SE(3) representation grows faster. Note that the ATE metrics plot behaves exactly as the relative positional error plot, which is caused by the fact, that there are no significant loop closures in the small simulation environment, hence the trajectory does not change much after the final optimization. The results with walls tilted by 80°are visualized in Fig. 8.

7. Conclusions

We proposed two solutions to the problem of representing plane-based features in the g^{2}o framework. First solution was based on a standard set of ver-

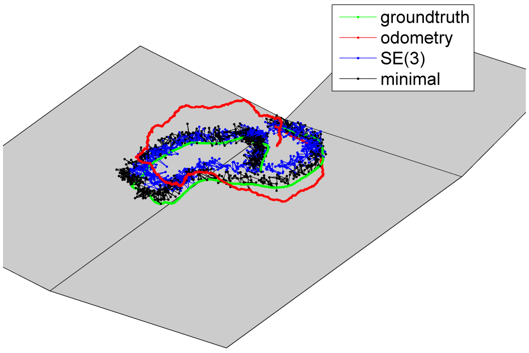

Fig. 8. Visualization of the SE(3) and minimal representation optimizations results with walls tilted by 80°

tices and edges from the framework and represented planes using SE(3) parametrization with carefully prepared information matrix. Second solution used minimal representation of planes and required an implementation of the vertex and the edge that extended the g^{2}o functionality. The implementation is an important contribution as it can be easily used in further development, as well as in other applications. Experiments verified that both approaches give reasonable results and can operate in real-time. When overparametrized representation is used, it is important to carefully construct information matrix. The matrix instructs the optimization algorithm which dimensions are relevant and what are relations between coordinate uncertainties. Despite the fact that presented approaches are theoretically equivalent, when conditions are harsh, the more specific solution performs better as comes to accuracy and convergence time. The work gives an insight how surplus dimensions affect the optimization process. The difference could be more significant if Jacobian matrices were computed analytically in both cases. Although experiments in a synthetic environment, without a real front-end, give no possibility to compare the performance of our approach with other systems, the presented solution provides a good start point for development of a complete SLAM system based on higher-level features.

Future work will focus on adding a front-end functionality to the system. We want to develop an algorithm for detecting and isolating planes, matching planes between consecutive frames and recognizing visited places. Considering other types of features in a single framework to build a robust and versatile system is also planned.

AUTHOR

Jan Wietrzykowski* – Poznań University of Technology, Institute of Control and Information Engineering, ul. Piotrowo 3A, 60-965 Poznań, Poland, e-mail: jan.wietrzykowski@cie.put.poznan.pl.

*Corresponding author

REFERENCES

[1] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-Up Robust Features (SURF)”, *Comput. Vis. Image Underst.*, vol. 110, no. 3, 2008, 346–359.

[2] D. Belter, M. Nowicki, and P. Skrzypczyński. “Accurate Map-Based RGB-D SLAM for Mobile Robots”. In: L. P. Reis, A. P. Moreira, P. U. Lima, L. Montano, and V. Muñoz Martinez, eds., *Robot 2015: Second Iberian Robotics Conference*, volume 418 of *Advances in Intelligent and Soft Computing (AISC)*, 533–545. Springer International Publishing, 2016.

[3] D. Belter, M. Nowicki, and P. Skrzypczyński, “Improving accuracy of feature-based RGB-D SLAM by modeling spatial uncertainty of point features”. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, 1279–1284.

[4] J. A. Castellanos, J. M. M. Montiel, J. Neira, and J. D. Tardos, “The SPmap: a probabilistic framework for simultaneous localization and map building”, *IEEE Transactions on Robotics and Automation*, vol. 15, no. 5, 1999, 948–952.

[5] J. Engel, J. Stückler, and D. Cremers, “Large-scale direct SLAM with stereo cameras”. In: *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, 2015, 1935–1942.

[6] G. Grisetti, R. Kümmerle, and K. Ni, “Robust optimization of factor graphs by using condensed measurements”. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, 581–588.

[7] A. Handa, T. Whelan, J. McDonald, and A. J. Davison, “A benchmark for rgb-d visual odometry, 3d reconstruction and slam”. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, 1524–1531.

[8] M. Kaess, “Simultaneous Localization and Mapping with Infinite Planes”. In: *IEEE Intl. Conf. on Robotics and Automation, ICRA*, Seattle, WA, 2015, 4605 – 4611.

[9] M. Kaess, A. Ranganathan, and F. Dellaert, “iSAM: Incremental Smoothing and Mapping”, *IEEE Transactions on Robotics*, vol. 24, no. 6, 2008, 1365–1378.

[10] C. Kerl, J. Sturm, and D. Cremers, “Dense visual SLAM for RGB-D cameras”. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, 2100–2106.

[11] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, “G^{2}o: A general framework for graph optimization”. In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, 3607–3613.

[12] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, “ORB-SLAM: A Versatile and Accurate Monocu-

lar SLAM System", *IEEE Transactions on Robotics*, vol. 31, no. 5, 2015, 1147–1163.

[13] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking". In: *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, 2011, 127–136.

[14] M. Nowicki and P. Skrzypczyński, "Experimental Verification of a Walking Robot Self-Localization System with the Kinect Sensor", *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 7, no. 4, 2013, 42–52.

[15] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF". In: *2011 International Conference on Computer Vision*, 2011, 2564–2571.

[16] R. F. Salas-Moreno, B. Glocken, P. H. J. Kelly, and A. J. Davison, "Dense planar SLAM". In: *Mixed and Augmented Reality (ISMAR), 2014 IEEE International Symposium on*, 2014, 157–164.

[17] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A Benchmark for the Evaluation of RGB-D SLAM Systems". In: *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, 2012.

[18] Y. Taguchi, Y. D. Jian, S. Ramalingam, and C. Feng, "Point-plane SLAM for hand-held 3D sensors". In: *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2013, 5182–5189.

[19] J. Weingarten and R. Siegwart, "3D SLAM using planar segments". In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, 3062–3067.

A Probabilistic Framework for Global Localization with Segmented Planes

Jan Wietrzykowski and Piotr Skrzypczyński

Abstract— **This paper proposes a novel approach to global localization using high-level features. The new probabilistic framework enables to incorporate uncertain localization cues into a probability distribution that describes the likelihood of the current robot pose. We use multiple triplets of planes segmented from RGB-D data to generate this probability distribution and to find the robot pose with respect to a global map of planar segments. The algorithm can be used for global localization with a known map or for closing loops with RGB-D data. The approach is validated in experiments using the publicly available NYUv2 RGB-D dataset and our new dataset prepared for testing localization on plane-rich scenes.**

I. INTRODUCTION

Solutions to the Simultaneous Localization and Mapping (SLAM) problem in 3-D [1], [2], [3] usually assume incremental localization, relying on the robot/sensor pose prior for matching the current perception to the map. However, if a reliable prior is unavailable, the robot has to find the pose with respect to the already learned or *a priori* known map by means of a global localization method.

Consider exemplary indoor scene views depicted in Fig. 1a. Are those the same scenes observed from different viewpoints or are they just similar? A human can tell that easily (Fig. 1b) using the semantic context, but a robot has to rely on numerical computations on the basis of some scene representation. The abstraction level of scene representation is perhaps the most important problem in developing robust global localization methods. Point features are most common in 3-D SLAM [1], [3], but if we exploit higher-level features [4], the local environment geometry can resolve ambiguities stemming from an abundance of repetitive or similar visual patterns [5]. As recovering the geometry from passive vision data requires intensive computations we focus on active RGB-D sensors, which are cheap, compact, and provide a rich description of the scene.

Therefore, we propose a novel approach to the problem of global localization using higher-level features. Unlike many 3-D SLAM solutions that focus on accurate mapping of small areas, our localization system[1] works on larger indoor scenes, with loopy sensor trajectories of extended duration. We contribute: (i) a new probabilistic localization framework utilizing Gaussian kernel approximation; (ii) a localization solution using this framework and segmented planes as features; (iii) RGB-D dataset suitable for evaluation of the proposed method.

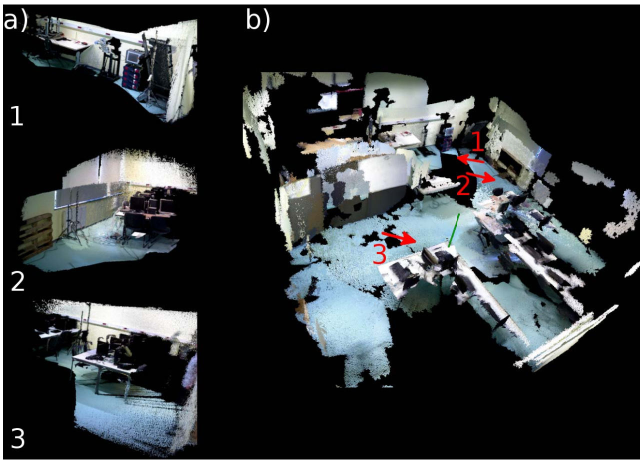

Fig. 1. Example of place recognition in a 3-D environment: perceived local views (a), and the global map view (b)

II. RELATED WORK

In keyframe-based 3-D SLAM systems, large loop closures are detected using appearance-based place recognition techniques [6]. One of the most widely used algorithms from this group is FAB-MAP [7]. In ORB-SLAM/ORB-SLAM2 [3] such a technique, based on the fast to compute ORB features is used also for relocalization [8]. Although Williams *et al.* [9] demonstrated that appearance-based methods scale better than map-based SLAM algorithms for large environments, these methods do not provide accurate estimates of the robot pose with respect to the map. SLAM systems that do explicit map reconstruction usually need to have a reasonable guess of the sensor pose before they attempt to match the local perception to the map [1]. The recent ElasticFusion system [10] that maintains a dense environment model applies an appearance-based approach for location recognition, but the database of locations contains predicted views of the dense map. If the global map is feature-based, synthesizing frame views becomes infeasible. Among the feature-based approaches, Heredia *et al.* [11] proposed a two-stage point-feature matching algorithm that facilitates global localization. Higher-level geometric features, that provide more localization constraints than points have been employed in a number of systems. An early attempt to use plane features was the 3-D EKF-SLAM by Weingarten and Siegwart [12]. More recently, an optimization-based approach to SLAM exploiting both, point and plane features was presented [13]. Optimization-based approaches to SLAM with infinite planes

*This work was funded by the PUT DSMK/0172 and DSPB/0162 grants

Jan Wietrzykowski and Piotr Skrzypczyński are with the Institute of Control and Information Engineering, Poznań University of Technology, 60-965 Poznań, Poland {jan.wietrzykowski; piotr.skrzypczynski}@put.poznan.pl

[1] Available at https://github.com/LRMPUT/PlaneLoc

as features are described in [14] and [4], whereas [2] explores plane segments in dense visual SLAM. Those approaches tackle, however, the incremental SLAM problem. Pathak *et al.* [15] proposed a fast method for registration of noisy planes. Although the Minimally Uncertain Maximal Consensus algorithm was demonstrated in [15] assuming limited translations and rotations between consecutive views, this method can solve unknown correspondences between planes, hence it has a potential for application in global localization. Similarly, Cupec *et al.* demonstrated in [16] that their earlier planar surface segments registration algorithm can be used for global localization employing a multi-hypothesis EKF to handle correspondence outliers. The solution of [17] is similar in spirit to our approach, but it addresses the place recognition problem by matching subgraphs representing the local topology of neighboring planar patches in a plane-based global map. Hence, although it uses a geometric rather than appearance-based approach, it is unable to produce an accurate estimate of the global robot pose.

III. PROBLEM STATEMENT AND SOLUTION

Consider two scenes visible in Fig. 2a, where the same place in a global map was shown from two different views using planes. These planes can be matched in a number of ways (Fig. 2b), but, if the views present the same place, only one association is valid. However, as examining all possible combinations is intractable, we build a probability distribution of the robot pose using cues from small subsets of planes.

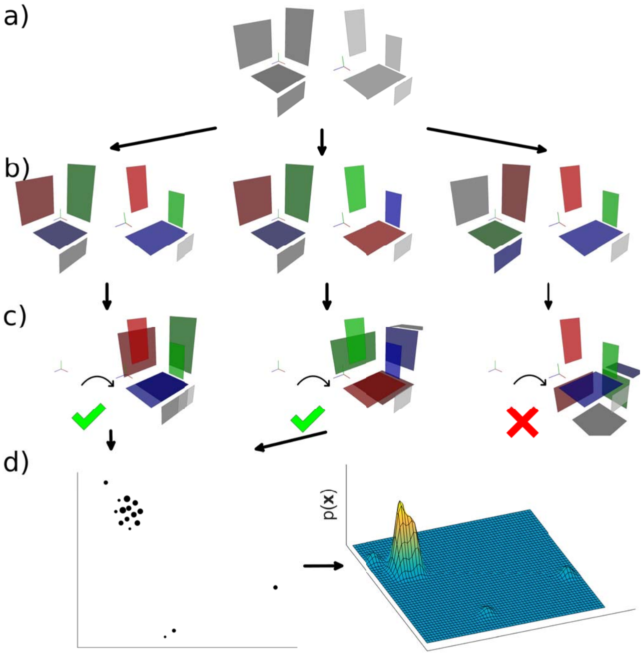

Fig. 2. Schematic illustration of generating the global robot pose PDF by matching sets of planes

At least three non-parallel pairs of matching planes are required to obtain an SE(3) transformation. Unfortunately, we don't know the associations between plane segments. We can try to discover them employing appearance (e.g. color), size and global location. However, using such criteria is insufficient, and there is a need to examine constraints imposed by the geometric transformations between the potentially matching features. Unfortunately, even using multiple above-mentioned criteria for matching we sometimes obtain wrong associations that act as strong outliers in typical estimation procedures, e.g. involving Kalman filtering [16]. A common solution to this problem is to embed the estimation into a RANSAC procedure, which however spawns other issues, such as extensive hypothesis evaluation and setting proper thresholds for outlier rejection. Therefore, our novel idea is to use multiple triplets of potentially corresponding planes to generate a kernel-based global probability density function (PDF) that describes the likelihood of the robot/sensor pose.

The triplets consist of three pairs of associated plane features (cf. Fig. 2b). Each triplet is evaluated if it induces a plausible transformation by projecting planes from the coordinate frame of the current sensor view into the coordinate frame of the global map (Fig. 2c). But even the plausible triplets should not contribute equally to the final pose hypothesis, as some triplets contain better matches than others. Therefore the contribution has to be weighted, which is illustrated in a simplified form in Fig. 2d. Inspired by [18], where a probability distribution was constructed from samples to find a feasible grasping sequence, we construct a PDF to find the transformation that best explains the observed triplets of segmented planes. A triplet supports the transformation by introducing a weighted Gaussian kernel that adds to the PDF. The kernels are placed in the location space, i.e. each point in that space corresponds to a possible transformation between the sensor view and the map. Hence, if many kernels are placed in some area, the probability density in that area is high. During localization, we seek the maximum of the PDF and finally test it for being the correct transformation.

IV. TRIPLETS OF PLANES

This section describes data processing steps used in generation and evaluation of triplets. The process begins with plane segmentation that isolates planar surfaces from a point cloud. The extracted planes are then matched to a set of planes in a global map and outcome pairs are used to form triplets of pairs representing possible transformations. Each transformation is evaluated to test if it is plausible and then passed to the probabilistic framework.

A. Extracting planes

We extract planar surfaces from a point cloud representing the observed scene using a simple method based on segmentation and segment merging by flood fill. The method is designed for the presented system because none of the off-the-shelf algorithms (e.g. [19]) satisfied our requirements. The point cloud is segmented by means of supervoxel clustering (Fig. 3a) and each segment with a low curvature is considered as a seed. The algorithm, using supervoxel adjacency list, recursively merges all segments connected to a seed that are sufficiently flat, their normals are approximately parallel to the normals of the seed, and there are no steps between them. Merged segments are tested to have at least minimal size to avoid adding many small segments. The last operation is to compute plane equations using all points belonging to the generated segments (Fig. 3b).

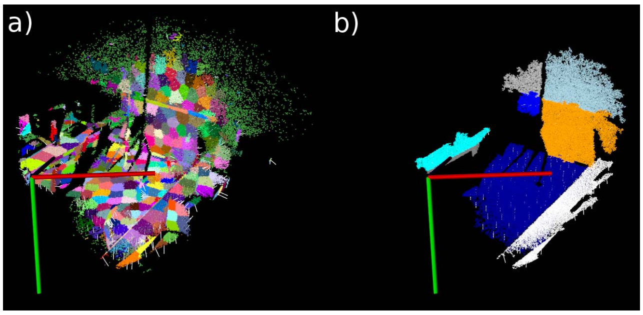

Fig. 3. Segmentation of planar surfaces with visualized normals (white lines): supervoxels (a), and merged segments (b)

B. Selecting triplets

Having the current view and the global map represented as planes, we pick pairs of planes, one plane from the current view, and another one from the map. Those pairs are potential matches, and each of them may be either correct if the two planes indeed represent the same planar surface, or incorrect if they don't. To limit the number of pairs only planes that are visually similar are considered. The appearance of each plane is represented as a histogram of the Hue and Saturation components of the HSV color model and is embedded into a vector $\mathbf{h}_i^s$ for the i-th plane from the current view, and into a vector $\mathbf{h}_j^m$ for the j-th plane from the map. Planes i and j are considered similar if the difference between their histograms doesn't exceed the predefined threshold:

$$h_{(i,j)} = |\mathbf{h}_i^s - \mathbf{h}_j^m| < \tau_h. \tag{1}$$

As three pairs allow to compute an SE(3) transformation, we form triplets of pairs that represent a valid transformation if all three matches are correct. Again, to limit the size of the search space, each triplet has to fulfill the following conditions:

- Each plane i and j has to appear in at most one of three pairs, as the same plane segment cannot be matched more than once.
- The map planes must not be further than τ_d each from the other. The map can be large in comparison to the current view, therefore if two planes are far from each other, they won't be visible in the same view.

C. Computing transformations

To evaluate correctness of the established triplets it is necessary to calculate an alleged SE(3) transformations between the frames of reference of the local view and the global map induced by those triplets. We use a general method that takes as input $n \geq 3$ pairs of planes and outputs a transformation given by the translation vector $\mathbf{t} = [t_x \quad t_y \quad t_z]^T$ and the rotation quaternion $\mathbf{r} = [r_x \quad r_y \quad r_z \quad r_w]^T$. The method consists of two steps. At first it calculates the rotation using normal vectors of the planes $\mathbf{n}_i^s$ and $\mathbf{n}_j^m$, then the translation is obtained using distances from origins d_i^s and d_j^m also. The normal vector and distance from the origin are parameters of the plane, and can be used to form an equation satisfied by every point $\mathbf{q}$ belonging to that plane: $\mathbf{n} \cdot \mathbf{q} - d = 0$. A derivation of the rotation calculation algorithm is based upon the method of Walker *et al.* [20]. The algorithm tries to minimize the differences between the views's plane normal vectors and the transformed map's plane normal vectors:

$$\begin{aligned} \mathbf{e}_{(i,j)} &= |\mathbf{W}(\mathbf{r})^T \mathbf{Q}(\mathbf{r}) \mathbf{n}_j^m - \mathbf{n}_i^s|^2 \\ &= 2\left[1 - \mathbf{r}^T \mathbf{Q}(\mathbf{n}_i^s)^T \mathbf{W}(\mathbf{n}_j^m) \mathbf{r}\right]. \end{aligned} \tag{2}$$

The total energy to minimize is given by equation:

$$E = \sum_{(i,j)} 2\left[1 - \mathbf{r}^T \mathbf{Q}(\mathbf{n}_i^s)^T \mathbf{W}(\mathbf{n}_j^m) \mathbf{r}\right] = \mathbf{r}^T \mathbf{C} \mathbf{r} + \text{const.}, \tag{3}$$

where: $\mathbf{C} = -2 \sum_{(i,j)} \mathbf{Q}(\mathbf{n}_i^s)^T \mathbf{W}(\mathbf{n}_j^m)$. Taking a derivative of (3) with respect to $\mathbf{r}$ and using a Lagrangian multiplier we obtain equation:

$$\underbrace{-\frac{1}{2}(\mathbf{C} + \mathbf{C}^T)}_{\mathbf{D}} \mathbf{r} = \lambda \mathbf{r} \tag{4}$$

where the solution $\mathbf{r}^*$ is given by the eigenvector of matrix $\mathbf{D}$ that corresponds to the largest eigenvalue. The computed rotation is an unambiguous solution to (4) if the largest eigenvalue is unique as well. The planes are considered to be one-sided, and the normal vectors point in direction opposite to the direction which a plane was observed from. The one-side assumption is justified by the fact that we want to use planar surfaces of objects that have some volume.

Computing the translation between frames of references is done by minimizing square error between the sensor view's plane distance, and the transformed global map's plane distance [21]:

$$S = \sum_{(i,j)} (d_i^s - d_j^m - (\mathbf{n}_j^m)^T \mathbf{t})^2 = |\mathbf{A} - \mathbf{B}\mathbf{t}|^2, \tag{5}$$

where:

$$\mathbf{A} = \left[d_{i,1}^s - d_{j,1}^m, \ldots, d_{i,n}^s - d_{j,n}^m\right]^T, \tag{6}$$

$$\mathbf{B} = \left[\mathbf{n}_{j,1}^m, \ldots, \mathbf{n}_{j,n}^m\right]^T. \tag{7}$$

The solution is given by the pseudo-inverse method:

$$\mathbf{t}^* = (\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T \mathbf{A}. \tag{8}$$

D. Evaluating transformations

Unfortunately, as the transformation computation is only minimizing the errors, the solution may actually be implausible in case of planes mismatch. To verify this, we use plane parametrization represented as a unit quaternion [14]:

$$\mathbf{p} = \frac{1}{\sqrt{1 + d^2}} [n_x, n_y, n_z, -d]^T, \tag{9}$$

and the transformation from map's frame of reference to the local view's frame of reference as a homogeneous matrix

$\mathbf{T}_{m,s}$. If the transformation is valid, for each pair of planes a representation of map's plane transformed to the view's frame should be approximately equal to the representation of the view's plane. The difference between those representations is computed using the logarithm map of quaternions:

$$\mathbf{f}_{(i,j)} = \log\left\{\left[\mathbf{T}_{m,s}^{T}\mathbf{p}_j^m\right]^{-1}\mathbf{p}_i^s\right\}. \tag{10}$$

Note that using a homogeneous matrix to transform quaternions is different than when transforming points:

$$\mathbf{p}^s = \mathbf{T}_{s,m}^{-T}\mathbf{p}^m = \mathbf{T}_{m,s}^{T}\mathbf{p}^m, \tag{11}$$

and the result has to be normalized afterwards. Finally, the criterion has a form $|\mathbf{f}_{(i,j)}| < \tau_f$, i.e. a norm of the logarithm map difference has to be below a certain threshold τ_f.

Planes are infinite and the computed transformation can be implausible, because the solution can point to a distant location, far away from the investigated environment. Therefore, it is also necessary to check if the transformation is justified by the available observations. In this work we assume the solution to be plausible when the convex hulls of the points belonging to the map planes, denoted by $\mathrm{chull}(\mathbf{P}_j^m)$, after transformation to the current view frame of reference, overlap with the convex hulls of the points belonging to the current view planes, denoted by $\mathrm{chull}(\mathbf{P}_i^s)$:

$$\mathbf{G}_{(i,j)} = \left[\mathbf{T}_{m,s}^{-1}\mathrm{chull}(\mathbf{P}_j^m)\right] \cap \mathrm{chull}(\mathbf{P}_i^s). \tag{12}$$

In other words, we check if for each pair of planes the same segment of the plane is observed and represented in the global map and the current scene (Fig. 4). The area of the co-observed part has to have a certain size: $\mathrm{area}(\mathbf{G}_{(i,j)}) > \tau_g$.

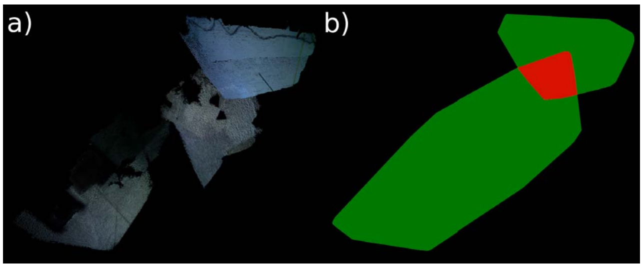

Fig. 4. Transformed point cloud of the map and point cloud of the current view (a), and intersection of convex hulls (b)

V. PROBABILISTIC FRAMEWORK

The set of triplets $\mathcal{T}$ generated in previous processing steps is next included in the probabilistic framework. Each triplet is scored and the computed scores are treated as weighting factors used to build a PDF. The final outcome of the method is the SE(3) transformation, which is most probable according to the supporting evidence.

A. Assigning weights

The triplet weight takes into consideration the appearance difference $h_{(i,j)}$ (1) and the area of the convex hulls intersection $\mathrm{area}(\mathbf{G}_{(i,j)})$ (12). Moreover, the weight depends on how frequently the respective plane segment was employed, as the same plane can be a part of multiple triplets, and therefore can introduce a bias to the PDF. We handle it by calculating an occurrence factor for each triplet a and each pair $(i,j) \in \mathcal{S}_a$ within the triplet. The more triplets including the same plane in a vicinity of induced transformation, the lesser the weight:

$$w_{a,(i,j)} = \left[\sum_{b\in\mathcal{T}}\sum_{(k,l)\in\mathcal{S}_b}\mathbb{I}_{i=k}\exp(-y_{(a,b)})\right]^{-1}, \tag{13}$$

where $\mathbb{I}_{i=k}$ is an indicator function equal to 1 whenever $i = k$ and 0 otherwise, and $y_{(a,b)}$ is a norm of a SE(3) logarithm map of a difference between transformations $\mathbf{v}$ for the triplets a and b:

$$y_{(a,b)} = \left|\log(\mathbf{v}_a^{-1}\mathbf{v}_b)\right|. \tag{14}$$

Note that, in opposition to Section IV, transformations are parametrized as 6-D vectors $\mathbf{v}$ that contain 3 translation variables t_x, t_y, t_z and 3 rotation variables r_x, r_y, r_z (the r_w can be always restored assuming that is non-negative and the quaternion is a unit quaternion). The overall weight for the triplet a is expressed by the equation:

$$w_a = \sum_{(i,j)\in\mathcal{S}_a}\mathrm{area}(\mathbf{G}_{(i,j)})w_{a,(i,j)}\exp(-h_{(i,j)}). \tag{15}$$

B. Constructing localization distribution

The SE(3) transformations induced by triplets are represented as points in a 6-D space with 3 variables for the position, and 3 for the rotation. The strength of the contribution is controlled by a weight given by (15), and we convert this contribution to the probabilistic language by placing a weighted Gaussian kernel in each transformation point. The final PDF is therefore a sum of all kernels:

$$p(\mathbf{x}) = \frac{1}{Z}\tilde{p}(\mathbf{x}) = \frac{1}{Z}\sum_{a\in\mathcal{T}}K_a(\mathbf{x}), \tag{16}$$

where Z is a normalizing constant, and the kernel is:

$$K_a(\mathbf{x}) = w_a\exp\left\{-\log(\mathbf{v}_x^{-1}\mathbf{v}_a)^T\mathbf{I}_a\log(\mathbf{v}_x^{-1}\mathbf{v}_a)\right\}. \tag{17}$$

The distance between kernel's center $\mathbf{v}_a$ and the transformation $\mathbf{v}_x$ represented by the point $\mathbf{x}$ is computed using logarithm map, and is embedded in the square form of multidimensional Gaussian distribution with the information matrix $\mathbf{I}_a$.

Having a probability distribution, we seek for the point with the highest probability. Inference in general distributions can be complicated and to avoid this, we exploit the fact that we already have a list of possible solutions. For each triplet, we evaluate the transformation induced by this candidate, and choose the one with the highest probability, denoted further as $\mathbf{x}_1$. Additionally, we search for the second-best solution $\mathbf{x}_2$ that is properly distant from the best one. To decide that the transformation $\mathbf{x}_1$ is correct, its probability has to exceed a certain threshold:

$$\tilde{p}(\mathbf{x}_1) > \tau_p \tag{18}$$

and has to be significantly greater than for the second-best maximum:

$$\tilde{p}(\mathbf{x}_1) - \tilde{p}(\mathbf{x}_2) > \tau_{pd}. \tag{19}$$

The last test is the fitness score test. It refers back to point clouds and assesses if the current view's point cloud transformed to the map's frame of reference is aligned with the map's point cloud. This test involves computation of the sum of squared distances from each point of the transformed views's point cloud $\mathbf{P}^s$ to the nearest neighbor in the map's point cloud $\mathbf{P}^m$, and has to be below the threshold:

$$\sum_{\mathbf{q}_l^s \in \mathbf{P}^s} (\mathbf{q}_l^s - \hat{\mathbf{q}}_l^m)^2 < \tau_{fs}, \tag{20}$$

where $\hat{\mathbf{q}}_l^m$ is the nearest neighbor in map's point cloud. It's worth noting that the fitness score test examines only points belonging to plane segments, and is much faster than ICP, as (20) is equivalent to a single iteration of ICP on a reduced size point cloud. If the transformation fulfills all three conditions, it is assumed to be the correct one.

VI. Experimental Evaluation

We evaluated the proposed algorithm in the global localization task with a known map, as the software is not yet integrated within a SLAM system. For global localization, the algorithm requires a point cloud representing the current view (local scene), and a global map composed of plane segments. As we are not aware of any experimental RGB-D dataset for which a map of plane features is available, we built a "global" point cloud using the ElasticFusion software [10], with the dataset's ground truth trajectory used for registration to avoid drift. The fused point cloud was then segmented into plane features, as described in Section IV. Our approach needs also the local point cloud to extract planes in the current view. Using a single RGB-D frame for that purpose is not enough because a sufficient number of planes has to be detected. Hence, to widen the local view context, we used again ElasticFusion to fuse together point clouds from the last 100 RGB-D frames. Considering the Kinect frame rate this short sequence takes few seconds, and in most cases does not accumulate significant drift. For the local perception, the ElasticFusion's trajectory estimates are used in the presented experiments.

The main dataset used is the PUT RGB-D/Workshop (PUT RGB-D/W)[2], which consists of 10 sequences (*seq1* to *seq10*) acquired in a 8×8 metres robotic workshop. The ground truth data was captured using the OptiTrack motion capture system. Three non-overlapping sequences *seq5*, *seq6* and *seq7* were used to build the global map that contains 56 segments. Moreover, we used a sequence from the publicly available NYUv2 dataset [22] acquired in a typical household environment. Unfortunately, in NYUv2 no ground truth data for the sensor trajectory is provided. Thus, we had to use the poses computed by ElasticFusion as ground truth.

The localization performance was measured by counting the locations that were correctly recognized, and those that were recognized incorrectly (if any). For the performance tests, we applied the algorithm to every 10-th pose of the recorded sequence, attempting to localize the sensor with respect to the global map. We treated as correct the sensor poses that were distant at most 0.11 from the respective ground truth pose, using the metrics given by (14). The 0.11 value was chosen, because re-localization within this range usually enables to recover tracking in our RGB-D SLAM [1], and should be suitable for other similar SLAM systems. Additionally, mean Euclidean $\bar{d}$ and angular $\bar{\alpha}$ distances between the computed poses and the ground truth trajectory were computed in all tests. Results are gathered in Tab. I, where four sets of parameters were evaluated: optimal for the PUT RGB-D/W dataset, optimal for the NYUv2 dataset, with probability difference test switched off, and with point cloud alignment test switched off. Visualizations of the recognized places are presented in Fig. 5 and 6, for the PUT RGB-D/W and NYUv2 datasets, respectively. We used the following parameter values: $\tau_d = 5.0$, $\tau_h = 2.5$ (1.3 for NYUv2), $\tau_f = 0.05$ and $\tau_g = 0.1$.

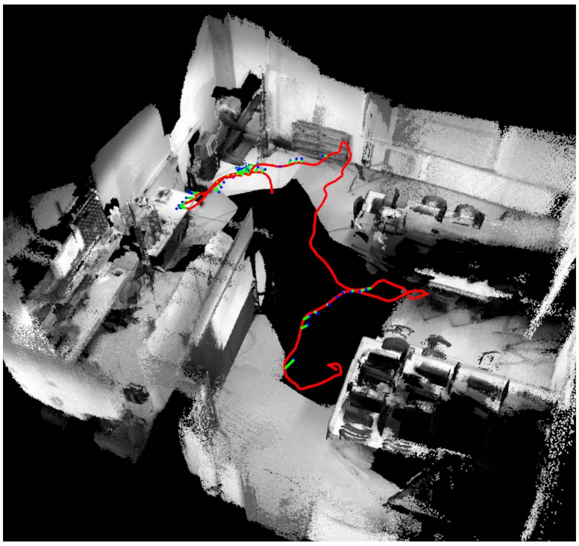

Fig. 5. Global localization results for the PUT RGB-D/W dataset *seq2* for $\tau_p = 1.1$, $\tau_{pd} = 0.2$, and $\tau_s = 0.07$. Test trajectories are marked with red lines, recognized places with blue dots, whereas green lines connect recognized locations to their respective ground truth poses

TABLE I

Global localization results for different parameter sets

parameters			PUT RGB-D/W					NYUv2				
τ_p	τ_{pd}	τ_{fs}	corr	incorr	unk	$\bar{d}$ [m]	$\bar{\alpha}$ [°]	corr	incorr	unk	$\bar{d}$ [m]	$\bar{\alpha}$ [°]
1.1	0.2	0.07	49	0	112	0.123	0.24	134	6	78	0.113	2.23
1.2	0.2	0.03	33	0	128	0.113	0.28	113	0	105	0.085	0.15
1.2	0.0	0.03	33	3	125	0.492	12.81	113	0	105	0.085	0.15
1.2	0.2	∞	49	13	99	0.559	35.82	147	7	64	0.122	2.07

The results obtained using two different datasets indicate that the proposed method is reliable, finding a large number of locations along the test trajectories. If the algorithm is correctly parametrized, it produces no false positives, which is of pivotal importance in localization task. The main cause for the number of places (local views) that remained unrecognized was the insufficient quality of depth data used to create the global maps. Maps produced from a single sequence (NYUv2) or few sequences of very limited overlapping had many areas that were empty or contained

[2] Available at `http://lrm.put.poznan.pl/rgbdw/`

point clouds of insufficient density to extract correct planes.

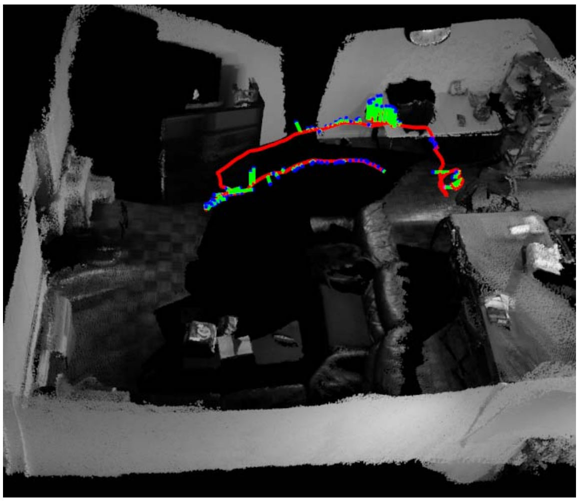

Fig. 6. Global localization results for the NYUv2 dataset for $\tau_p = 1.2$, $\tau_{pd} = 0.2$, and $\tau_s = 0.03$. Test trajectories are marked with red lines, recognized places with blue dots, whereas green lines connect recognized locations to their respective ground truth poses

The mean computing time for a single global localization act in the experiments was 21 s on a Core i5 2.6 GHz laptop. However, the most time-consuming step (14 s in average) was the computation of the fitness score (20). This step can be made much faster using approximate nearest neighbor search, taking less than 1 s, as shown by a preliminary implementation employing octree. Further optimization of the computation time is a matter of our current research.

VII. CONCLUSIONS

We tackled the problem of global localization applying a novel approach that creates a PDF describing the likelihood of sensor's pose using plane features. The experimental results suggest that the proposed method performs well in large-room-sized environments, yielding correct and accurate pose estimates whenever it is possible to provide a good quality of the *a priori* map and there are features available in the environment.

An important advantage of the new probabilistic framework is that not only paired planes can contribute to the PDF. The framework can handle localization cues coming from other feature types, or from other sensing modalities, e.g. an orientation sensor like AHRS. Applications of the presented algorithm are not limited to global localization and loop-closing. It can be easily adapted for matching plane features in graph-based SLAM utilizing planes [4].

ACKNOWLEDGMENT

The OptiTrack was funded under grant 6419/IA/SP/2015 and used courtesy of the Chair of Control and Systems Engineering. The authors also would like to thank M. Kiełczewski for the technical assistance with OptiTrack.

REFERENCES

[1] D. Belter, M. Nowicki, and P. Skrzypczyński, "Improving accuracy of feature-based RGB-D SLAM by modeling spatial uncertainty of point features," in *IEEE Int. Conf. on Robotics and Automation*, Stockholm, 2016, pp. 1279–1284.

[2] L. Ma, C. Kerl, J. Stückler, and D. Cremers, "CPA-SLAM: Consistent plane-model alignment for direct RGB-D SLAM," in *IEEE Int. Conf. on Robotics and Automation*, Stockholm, 2016, pp. 1285–1291.

[3] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo and RGB-D cameras," in *arXiv preprint, arXiv:1610.06475v1*, 2016.

[4] J. Wietrzykowski, "On the representation of planes for efficient graph-based SLAM with high-level features," *Journal of Automation, Mobile Robotics & Intelligent Systems*, vol. 10, no. 3, pp. 3–11, 2016.

[5] K. Ho and P. Newman, "Combining visual and spatial appearance for loop closure detection in SLAM," in *Proc. of European Conference on Mobile Robots (ECMR)*, Ancona, 2005, pp. 62–67.

[6] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual place recognition: A survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, 2016.

[7] M. Cummins and P. Newman, "FAB-MAP: Probabilistic localization and mapping in the space of appearance," *Int. Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[8] R. Mur-Artal and J. D. Tardós, "Fast relocalisation and loop closing in keyframe-based SLAM," in *IEEE Int. Conf. on Robotics and Automation*, Hong Kong, 2014, pp. 846–853.

[9] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. Tardós, "A comparison of loop closing techniques in monocular SLAM," *Robotics and Autonomous Systems*, vol. 57, no. 12, pp. 1188–1197, 2009.

[10] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison, "ElasticFusion: Dense SLAM without a pose graph," in *Robotics: Science and Systems (RSS)*, Rome, 2015.

[11] M. Heredia, F. Endres, W. Burgard, and R. Sanz, "Fast and robust feature matching for RGB-D based localization," in *arXiv, CoRR abs/1502.00500*, 2015.

[12] J. Weingarten and R. Siegwart, "3D SLAM using planar segments," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Beijing, 2006, pp. 3062–3067.

[13] Y. Taguchi, Y. D. Jian, S. Ramalingam, and C. Feng, "Point-plane SLAM for hand-held 3D sensors," in *IEEE Int. Conf. on Robotics and Automation*, Karlsruhe, 2013, pp. 5182–5189.

[14] M. Kaess, "Simultaneous localization and mapping with infinite planes," in *IEEE Int. Conf. on Robotics and Automation*, Seattle, 2015, pp. 4605–4611.

[15] K. Pathak, A. Birk, N. Vaskevicius, and J. Poppinga, "Fast registration based on noisy planes with unknown correspondences for 3D mapping," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 424–441, 2010.

[16] R. Cupec, E. K. Nyarko, D. Filko, A. Kitanov, and I. Petrović, "Global localization based on 3D planar surface segments detected by a 3D camera," in *Proc. of the Croatian Computer Vision Workshop*, Zagreb, 2013, pp. 31–36.

[17] E. Fernández-Moral, W. Mayol-Cuevas, V. Arévalo, and J. González-Jiménez, "Fast place recognition with plane-based maps," in *IEEE Int. Conf. on Robotics and Automation*, Karlsruhe, 2013, pp. 2719–2724.

[18] M. Kopicki, R. Detry, M. Adjigble, R. Stolkin, A. Leonardis, and J. L. Wyatt, "One-shot learning and generation of dexterous grasps for novel objects," *Int. Journal of Robotics Research*, vol. 35, no. 8, pp. 959–976, 2016.

[19] T. T. Pham, M. Eich, I. Reid, and G. Wyeth, "Geometrically consistent plane extraction for dense indoor 3D maps segmentation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Daejeon, 2016, pp. 4199–4204.

[20] M. W. Walker, L. Shao, and R. A. Volz, "Estimating 3-D location parameters using dual number quaternions," *CVGIP: Image Understanding*, vol. 54, no. 3, pp. 358–367, 1991.

[21] O. D. Faugeras and M. Hebert, "A 3-D recognition and positioning algorithm using geometrical matching between primitive surfaces," in *Proc. of the Eighth International Joint Conference on Artificial Intelligence - Volume 2*, 1983, pp. 996–1002.

[22] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proceedings of the 12th European Conference on Computer Vision*. Berlin: Springer, 2012, pp. 746–760.

Robotics and Autonomous Systems 113 (2019) 160–173

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

PlaneLoc: Probabilistic global localization in 3-D using local planar features

Jan Wietrzykowski *, Piotr Skrzypczyński

Institute of Control, Robotics, and Information Engineering, Poznań University of Technology, ul. Piotrowo 3A, PL 60-965, Poznań, Poland

ARTICLE INFO

Article history:
Available online 18 January 2019

Keywords:
Global localization
SLAM
Planar segments
RGB-D data

ABSTRACT

The global localization problem concerns situations when a map of the environment is known but there is no initial guess of the agent position. Whereas the ability to perform global localization is required in many practical situations, it is still an open problem, particularly if the agent requires to find an accurate estimate of its 3-D pose. In this article, we describe PlaneLoc, a novel probabilistic approach to 3-D global localization, which integrates multiple local cues to construct a probability distribution that describes the likelihood of the agent pose. This framework enables to incorporate various types of localization cues but we demonstrate its feasibility using segmented planes abstracted from RGB-D data. We use multiple triplets of planar segments to generate candidate probability distribution and employ it to find the most probable pose with respect to a global map of planar segments. The PlaneLoc implementation uses the ORB-SLAM2 system that serves as visual odometry and makes it possible to generate observation in a form of sets of local segments online. The proposed approach can be used for global localization with a known map or for loop closing and re-localization in Simultaneous Localization and Mapping. The implemented system is validated in experiments using publicly available RGB-D data sets, including our own data set acquired specifically for testing localization methods based on planar features.

1. Introduction

1.1. Motivation and problem formulation

Continuous development in the field of mobile robotics and other autonomous systems (e.g. personal localization using smartphones) establishes an ever-rising demand for accurate and reliable localization. As far as GPS-denied indoor environments are considered, the localization problem is typically solved in an incremental manner, employing one of the Simultaneous Localization and Mapping (SLAM) algorithms [1]. In SLAM, consecutive agent poses are tracked assuming that the pose has the Markov property, and the previous pose is known with an acceptable uncertainty.

Unfortunately, such prior pose is not accurate enough or is not available in many practical scenarios, due to a wrong matching of the local perception to the map or ''kidnap situation'', when the Markov assumption is violated. Also if the agent comes back to an already mapped area after making a large loop, its current pose is often too uncertain to be used for matching with the map. To recover from these situations, the agent has to localize itself again with respect to the known part of the environment map. In the SLAM terminology, this process is known alternatively as re-localization for the tracking failure, and loop closing for re-visiting already known environment. Both cases can be regarded as variants of the global localization problem, which we define as finding the pose of an agent in a known map without any knowledge of the past agent poses. In the experiments presented in this paper, we do not use any pose prior, and we treat all the localization acts as "kidnap situations".

As far as 3-D localization is considered, an important matter is the source of data. Whereas visual place recognition is arguably the most popular way of recognizing locations in robotics [2], such algorithms localize the agent only topologically, by finding the image in the global map that is most similar to the current perception. Then, local point features are required to compute the camera metric pose, e.g. using the PnP algorithm [3]. On the other hand, the knowledge of the local view and the map geometry enables the agent to directly compute an SE(3) transformation between the local and the global frame. However, recovering this geometry from vision data requires intensive computations, while 3-D lidars are still expensive and often suitable for outdoor applications only. Thus, for the indoor use, we focus on RGB-D sensors, because they are cheap, compact, and provide rich enough description of the scene, including both photometric and geometric features [4].

But even considering both, the visual and geometric, features it is not easy to figure out whether the sensor is observing a known location or not (Fig. 1). The main reason for this difficulty is the locality of perception, which makes the observed features

[*] Corresponding author.
E-mail addresses: jan.wietrzykowski@put.poznan.pl (J. Wietrzykowski), piotr.skrzypczynski@put.poznan.pl (P. Skrzypczyński).

https://doi.org/10.1016/j.robot.2019.01.008
0921-8890/© 2019 Elsevier B.V. All rights reserved.

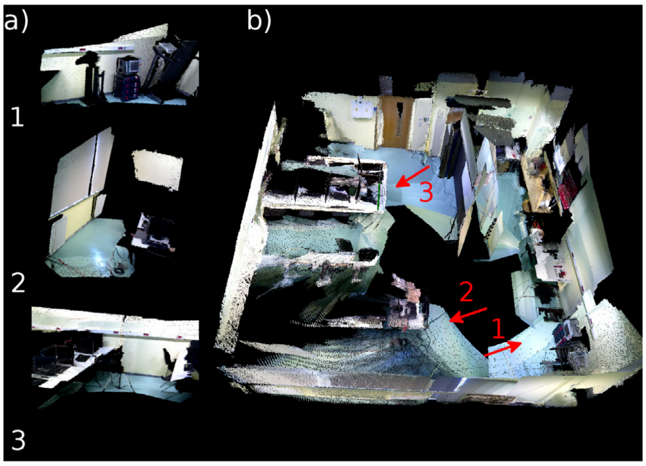

Fig. 1. Indoor place recognition from local 3-D views: local views rendered from RGB-D point clouds (a), and a global map of the same environment (b).

susceptible to changes due to the changing viewpoint, as seen in the snapshots in Fig. 1a. A human can tell with a high dose of confidence that the local views correspond to the numbered places in the global map (Fig. 1b). But artificial autonomous agents cannot synthesize all the cues that are evident for a human being and have to rely on numerical computations on the basis of some scene representation.

The abstraction level of scene representation is perhaps the most important problem in developing robust global localization methods. Point features with descriptors [5,6] or image patches are the most common in contemporary 3-D SLAM systems. However, in global localization, it is difficult to unambiguously match a small region of an image to hundreds or thousands of other regions from the map, due to the huge number of combinations that quickly makes examining all of them intractable as the map size increases.

Therefore, geometric features at a higher level of abstraction should be considered for global localization if we would like to estimate the agent metric pose in a one-shot manner. Among the simple geometric features that can be extracted from RGB-D point clouds planar segments seem to be the most useful and most universal, and they are abundant in all man-made indoor environments.

However, to efficiently use the planar features to constrain the agent pose in 3-D we need a method that computes possible SE(3) transformations between the current perception (expressed as a set of segments), and the global map made of compatible features. Inspired by [7], where a probability distribution was constructed from samples to find a feasible grasping sequence, we construct a probability density function (PDF) to find the transformation that best explains the constraints between the local set of segments and the map. This formulates a probabilistic framework that enables to incorporate even partial and uncertain localization cues into a distribution that describes the likelihood of the current pose of the agent. In this paper, we demonstrate PlaneLoc, a practical localization solution within this framework, which uses planar segments extracted from RGB-D data to infer about the agent whereabouts.

1.2. Related work

The majority of existing global localization systems employ appearance-based methods that rely on visual data provided by cameras [2]. Although the place recognition algorithms scale better than map-based SLAM algorithms for large environments [8], they provide only topological localization, not metric information about the agent pose.

One of the most widely used algorithms from this group is FAB-MAP [9], that applies the Bag of Visual Words method [10] to decide whether the currently observed scene is similar to a previously visited one. This algorithm can also be applied as stand-alone topological SLAM [11]. It performs well outdoors, and in environments rich in point features, but its feasibility for practical indoor localization with a hand-held camera is limited [12]. The Bag of Visual Words technique is used in a number of other place recognition methods, such as DBoW2 [13] that applies vocabulary tree and binary features to reduce the computation time. Appearance-based methods also use different frame encoding schemes: randomized ferns [14] or features containing geometric properties [15]. In the last few years, learning-based approaches employing convolutional neural networks (CNN) rose to prominence in computer vision. These methods are successfully applied in appearance-based localization [16]. However, adopting the deep learning paradigm for metric global localization was only shown in few very recent papers, e.g. for outdoor scenarios [17]. The adoption of CNN promises better performance and robustness due to using features learned from data instead of hand-crafted ones, but deep learning methods require very large labeled data sets for training. So far, such data sets for indoor metric localization at the global scale are hardly available.

A place recognition approach can be also used for re-localization in incremental SLAM, assuming that the adopted algorithm ensures at least approximately real-time performance. For example, a further development of DBoW2 employing ORB features [18] is used in ORB-SLAM [3] and ORB-SLAM2 [6] for both re-localization and loop closing. However, after finding the most similar image frame in the learned map a lot of effort is necessary for map management to ensure correct matching between the already mapped features and the observed ones [3].

If explicit geometric map reconstruction is performed, the SLAM system usually needs to have a reasonable guess of the sensor pose before it attempts to match the local perception to the map [5]. Therefore, accurate metric global localization is highly desirable for loop closing and re-localization. Older SLAM systems, operating with 2-D maps built from laser scans solved this problem by matching of a number of local features (2-D line segments, corners, etc.) to the map and tracking multiple hypotheses with the Extended Kalman Filter (EKF) to handle matching ambiguity [19]. In contrast, the Markov Localization algorithm [20] established a probability distribution over the discretized space of possible agent positions and then localized the agent using the Markov assumption. One prominent approach for localization with a known environment map is particle filtering [21]. Whereas localization algorithms employing particle filtering can re-localize a "kidnapped" robot [22], they suffer from problems with representing the uniform probability distribution in global localization. As particles have to be sampled over the entire state space, which increases with the map size, the filter converges slowly and may require to move the sensor until ambiguities are eliminated. This problem can be also circumvented providing a coarse initial position of the sensor, as shown by Ito et al. [23], who combined RGB-D perception with WiFi signal strength used to compute the initial guess.

Considering 3-D localization with images or RGB-D data we find only a few examples of metric global localization. When maintaining a dense model of the environment, it is possible to synthesize frame views, as in ElasticFusion [24]. If the current RGB-D frame matches the one synthesized from the global, surfel-based map, the map gets globally aligned. The efficiency is ensured using randomized ferns to encode frames at the matching stage. For sparse map representation, Heredia et al. [25] proposed a two-stage point-feature matching algorithm that facilitates global localization. However, relying solely on point features has a number of drawbacks. To compute the SE(3) transformation it is necessary to match a number of feature points, while any incorrect match can result in a wrong transformation. It is also difficult to incorporate information about the scene geometry and the underlying semantics in the matching of point features.

Higher-level features, as line segments or planar segments, are more distinguishable in the environment than point features. A number of approaches to solving the incremental SLAM problem (without global localization) employ such higher-level geometric features that provide more strict localization constraints than points. An early 3-D EKF-based SLAM employing planes as features was described in [26]. The system presented in [27] used infinite planes as features in an optimization-based approach to SLAM, while employing both, point and plane features, was proposed in [28]. Extraction of planes was also included in a direct SLAM approach [29] to reduce drift.

Efficient detection of planes and planar segments has been widely studied in the context of computer graphics, computer vision, and robotics. Fast plane detection algorithms based on RANSAC [30] are not well suitable for processing of unordered point clouds, because non-deterministic fitting of the model can lead to suboptimal results depending on the order of picking the data points from the cloud. Whereas the Hough transform is a voting scheme that enables extraction of multiple planes from unordered point clouds [31], it is computationally expensive due to the high-dimensional Hough space. Recent research on 3-D scene segmentation and object detection resulted in many approaches that extract planar primitives by region growing and clustering, such as the agglomerative hierarchical clustering algorithm [32]. The extraction of planar segments has been implemented according to this idea in the preliminary version of our global localization system [33]. The inspiration was taken from the algorithm described in [34]. We have noticed that the original algorithm from [34], aimed at segmentation of RGB-D data for object recognition, produced plane-based models that were considerably over-segmented for our purposes.

Only a few researchers proposed to use features at a higher level of geometric abstraction for global localization. Pathak et al. [35] proposed a fast, global method for registration of noisy planes. Cupec et al. [36] demonstrated the use of their planar segments registration algorithm in a global localization system, employing a multi-hypothesis EKF to handle correspondences. In contrast, the work by Fernàndez-Moral et al. [37] uses sub-graphs of plane adjacency to find a corresponding location in a global map. The method uses a series of unary and binary tests to check if the geometry of scenes matches. Unfortunately, due to examining all combinations of matches it is possible to check only sub-graphs that are composed of direct neighbors of the chosen plane. An interesting choice for global localization is complete object features. They can be easily distinguished, and their number in a typical scene is much smaller than the number of geometric features of any type. However, perception systems are not yet robust enough and object detection requires some sort of *a priori* object models (e.g. CAD models) [38,39].

1.3. Contribution

We propose a novel approach to the problem of global localization using a probabilistic framework based on a mixture of weighted Gaussian kernels that represent local and partial localization cues. Though this framework is able to use different types of localization cues, it is applied here to handle the constraints that are imposed on the six degrees of freedom (6-d.o.f.) pose of a free-floating RGB-D sensor by matching the locally observed planar segments to a global map of planar features.

While we demonstrate the PlaneLoc system[1] in generic global localization tasks, using an *a priori* prepared map of the whole environment, it is possible to apply our solution for both re-localization and direct (metric) loop closing with any RGB-D SLAM algorithm. To facilitate the use of PlaneLoc with online learned maps we show also a new and efficient algorithm for extracting planar segments from RGB-D point clouds, and a method that incrementally integrates the resulting features into a global map.

This journal article builds upon our recent conference paper [33] that introduced the idea of metric global localization with planar features and demonstrated some preliminary results. These results were obtained using maps of planar segments extracted from unordered global point clouds integrated using the ElasticFusion software [24]. In this research, we no longer need the ElasticFusion or any other software that integrates the RGB-D points in an off-line fashion. The new version of PlaneLoc integrates planar segments extracted online from a number of consecutive RGB-D frames into a local view map. Although we use the ORB-SLAM2 system to obtain sensor pose estimates while building the local map, PlaneLoc can be integrated with any other RGB-D SLAM or visual odometry software for that purpose. Hence, major contributions of this article with respect to the conference paper are new and improved methods for the representation and matching of the segments that enable the use of our global localization framework in the context of online navigation with learned maps. In summary, the contributions of this work are as follows:

- a novel global localization framework employing Gaussian kernel approximation to integrate local and partial localization cues to represent the likelihood of the agent pose,
- a particular instantiation of this framework that uses sets of planar segments to establish the localization cues,
- new algorithms that extract planar segments from the RGB-D frames and integrate them into either local or global segment-based maps,
- experimental evaluation of the new version of PlaneLoc on publicly available data sets, including a comparison to the state-of-the-art ORB-SLAM2 system in the re-localization task.

The remainder of the paper has the following structure: The global localization framework is introduced in Section 2. In Section 3 the algorithm used to obtain the global and local maps of planar segments is presented. Next, the practical global localization system employing the planar segments is detailed in Section 4. Section 5 presents our experimental methodology and the results of testing the PlaneLoc system on two publicly available RGB-D data sets. Finally, a brief discussion on the advantages and limitations of the proposed approach to localization, and conclusions with an outlook of future research are provided in Section 6.

2. Localization framework

A fundamental problem in global localization is to associate the local perception (scene view) to a unique location in the known map of the environment. If both the scene view and the global map are represented by planar segments, this problem transforms into finding associations between the locally observed segments and the segments in the global map. We can try to discover the associations between planar segments employing appearance (e.g. color), or size and global pose of the segments. However, many ambiguities remain when using such criteria, which make it necessary to examine constraints imposed by the geometric relations between the potentially matching features. Since infinite plane primitives provide only partial constraints on 6-d.o.f. poses of the associated

[1] Open source code: https://github.com/LRMPUT/PlaneLoc.

segments, at least three non-parallel pairs of matching segments are required to obtain a SE(3) transformation. Note that we do not use directly the shapes of the segments (i.e. their hulls) in matching, as a shape extracted from a noisy point cloud cannot be considered a reliable feature, neither we are aware whether the whole object has been already observed. Therefore, we propose the process of associating the planar segments to the map as illustrated in Fig. 2. Two sets of planar segments representing the same place seen from two different viewpoints are shown in Fig. 2a. These sets can be associated in a number of ways, as demonstrated in Fig. 2b and 2c. However, if both sets indeed represent the same place, only one association is valid.

In the global localization procedure, one of these sets represents the local view at the current pose of the agent, while the other one represents a location picked from the available global map. Thus, examining all the possible matching combinations becomes intractable even for room-size maps. Moreover, even combining the pairwise relations between the sets of planar segments and the unary constraints (appearance, size) as criteria in the matching procedure we can obtain some wrong associations due to local similarities of the environment geometry and visual appearance similarities. These associations act as outliers in recursive estimation procedures, e.g. involving Kalman filtering [36], and degenerate the results. A common solution to this problem is to embed the estimation into a RANSAC scheme, which however spawns other issues, such as extensive hypothesis evaluation and setting proper thresholds for outlier rejection. Therefore, our idea introduced in [33] is to build a probability distribution of the agent pose using triplets of associated planar segments as localization cues. We use triplets as it is a minimal number of associations that fully constrains SE(3) pose. Employing many potentially corresponding planar segments combinations, we generate a kernel-based global PDF that describes the likelihood of the agent pose.

The triplets consist of three pairs of associated plane features (cf. Fig. 2b). Each triplet is evaluated if it induces a plausible transformation by projecting planar segments from the coordinate frame of the current sensor view into the coordinate frame of the global map (Fig. 2c). But even the plausible triplets should not contribute equally to the final pose hypothesis, as some triplets provide more reliable information. Thus, the contribution has to be weighted, which is illustrated in a simplified form in Fig. 2d. Then, we construct a PDF to find the transformation that best explains the associations of the triplets observed from the current agent pose. A triplet supports the transformation by introducing a weighted Gaussian kernel that adds to the PDF. The kernels are placed in the location space, i.e. each point in that space corresponds to a possible transformation between the sensor view and the map. Hence, if many kernels are placed in some area, the probability density in that area is high. During localization, we seek the maximum of the PDF and finally test it for being the correct transformation.

The data processing steps used in generation and evaluation of the triplets of planar segments for global localization are summarized in Fig. 3. The process begins with plane segmentation that isolates planar surfaces from the acquired RGB-D frames. The extracted segments are merged into a local map of segments that plays the role of the local 3-D view. Individual planar segments from the local map are then matched to segments in the global *a priori* map, and the outcome pairs are used to form triplets of pairs representing possible transformations. Each transformation is evaluated to test if it is plausible and then passed to the probabilistic localization framework that computes the agent pose PDF and finds the most plausible pose with respect to the global coordinate system.

3. Representing a scene using planar segments

A proper representation of the scene is essential for our system to achieve good performance. A scene should be represented with a sufficient detail level and there should be incorporated as few false-positive observations as possible. Therefore, the preliminary implementation of our system described in [33] was extended with a method of extracting planar segments directly from individual RGB-D frames and a mechanism for managing the map of planar features.

Thus, we no longer use an external system to fuse the acquired RGB-D data from a number of consecutive sensor frames into larger point clouds, that were afterward segmented into planar patches. Instead, the frame processing pipeline starts with the extraction of planar segments directly from the incoming RGB-D images. The extracted segments are then merged with the planar features already kept in the local map. However, individual planar segments do not impose enough constraints to obtain frame-to-frame sensor motion while building the local map. Because knowing sensor motion is crucial for map accumulation from consecutive frames, we use the ORB-SLAM2 system to provide visual odometry that runs along the local map update process.

3.1. Extracting planar segments

We extract planar surfaces from RGB-D images (Fig. 4) and exploit the fact that both the RGB image (Fig. 5a) and the depth data (Fig. 5b) yielded by an RGB-D sensor are stored in dense, two-dimensional arrays. Processing dense 2-D data is less computationally expensive than sparse 3-D clouds because the adjacency relations between points are straightforward. The most time-consuming step in the previous version of the system, namely supervoxel clustering [33], is now replaced by 2-D image segmentation, enhanced with the depth information. The segmentation algorithm merges individual pixels into larger patches that are internally consistent and is based on the work by Felzenszwalb and Huttenlocher [40]. It is a graph-based method that assigns a weight to each edge of the graph. This weight depends on the similarity between the neighboring pixels it connects.

The regions are then merged using a criterion concerning edge weight and the internal variability of the region, starting with edges of the lowest weight (Fig. 5c). The weight between the ith and jth pixel is given by the following equation:

$$\omega_{(i,j)} = 0.5|\mathcal{I}_i - \mathcal{I}_j| + 0.5c|\mathcal{D}_i - \mathcal{D}_j|, \tag{1}$$

where $\mathcal{I}_i$ is the ith pixel intensity value, $\mathcal{D}_i$ is the ith pixel depth value, and $c = 64$ is a scaling factor that assures the same ranges for pixel intensities and depth values. Since (1) accounts also for the variability of depth values, our region merging criterion takes into account the geometry of the scene, unlike the criterion in [40]. Finally, only planar patches satisfying the following criteria are taken into further consideration:

- The number of points in a patch has to be above the threshold τ_{eppts} to ensure that sufficient statistics have been collected. The threshold depends on the accuracy of depth measurements in the given depth range. The value of 50 is used for Kinect/Xtion, because it eliminates small patches whose shapes cannot be determined when observing exemplary depth images. This value should be increased for sensors of lower accuracy of depth measurements.
- The curvature of a segment has to be below the threshold τ_{ecurv}, so it is indeed a planar segment. The curvature is computed as a ratio of the smallest eigenvalue to the sum of all eigenvalues: $\frac{\lambda_3}{\lambda_1+\lambda_2+\lambda_3}$. The value of the threshold is related to the accuracy of depth measurements and is set to 0.06

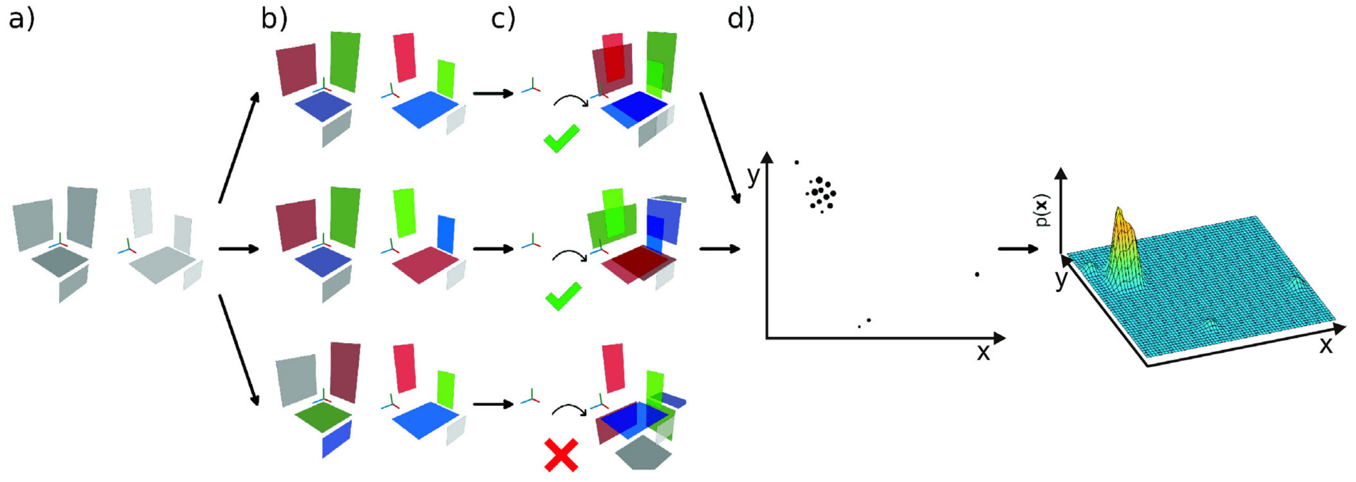

Fig. 2. Conceptual illustration of how the agent global pose PDF is generated by matching sets of planar segments.

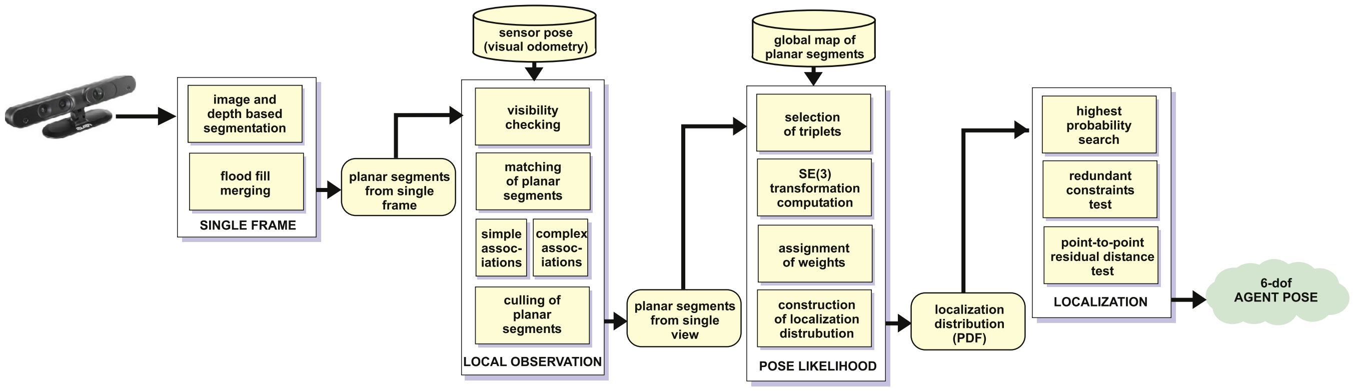
image and depth based segmentation
flood fill merging
SINGLE FRAME
planar segments from single frame
sensor pose (visual odometry)
visibility checking
matching of planar segments
simple assoc- iations
complex assoc- iations
culling of planar segments
LOCAL OBSERVATION
planar segments from single view
global map of planar segments
selection of triplets
SE(3) transformation computation
assignment of weights
construction of localization distrubution
POSE LIKELIHOOD
localization distribution (PDF)
highest probability search
redundant constraints test
point-to-point residual distance test
LOCALIZATION
6-dof AGENT POSE

Fig. 3. Simplified block scheme of the PlaneLoc global localization system. Rounded blocks represent data structures in the processing pipeline.

for Kinect/Xtion. This value eliminates patches that are not considered as planar when observing point clouds generated from exemplary frames.

- The ratio of the smallest eigenvalue and both the second and the third smallest has to be below the threshold τ_{eeig}. Those ratios are the measures of how precisely the normal vector is estimated in the two directions spanning the plane. Same as for the curvature, the threshold is related to the accuracy of the depth measurement, and the value of 0.4 is set to eliminate patches that cannot be unambiguously recognized as planar.

Planar patches are then merged by the flood fill algorithm (Fig. 5d). The algorithm, using patches adjacency list, merges all patches that are sufficiently flat, their normals are approximately parallel to each other, and there are no steps between them. Merged patches are considered as planar segments if they contain at least $\tau_{\text{espts}} = 1500$ points to avoid further processing of poorly estimated segments. The value of τ_{espts} has been adjusted to eliminate small, non-planar objects that could be observed in point clouds generated from exemplary frames. Using a threshold depending upon the number of points, instead of area, makes it possible to accept small, but well-estimated segments, e.g. those observed from a small distance. The last operation is to estimate the equation of the infinite plane supporting the planar segment and the hull embracing all data points belonging to the segment (see Section 3.3).

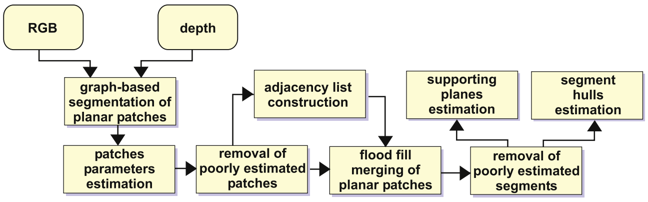
RGB
depth
graph-based segmentation of planar patches
patches parameters estimation
removal of poorly estimated patches
adjacency list construction
flood fill merging of planar patches
removal of poorly estimated segments
supporting planes estimation
segment hulls estimation

Fig. 4. Schematic overview of the planar segments extraction algorithm. Only the processing blocks are shown in the pipeline.

3.2. Managing the map of planar features

To build a map of planar segments that can be used in a localization task, it is necessary to carefully manage the process of merging information from consecutive frames. The model of each plane should be refined when new observations are available, avoiding the fusion of the data coming from wrong associations at the same time. To maintain good performance in terms of speed and efficiency of localization, it is also mandatory to remove planar segments originating from wrong measurements or faulty extraction. To assure that our system follows this strategy, we have introduced two mechanisms: *end of life value* l_s for all planar segments in the map, and *delayed merging* of complex matches (Fig. 6). The value of l_s reflects the certainty that the feature is a valid planar segment, based on the available observations. Each new planar segment has a value $l_{\text{init}} = 4$ assigned. This value is then decreased by $l_{\text{dec}} = 1$, whenever the segment should be observed from the current pose

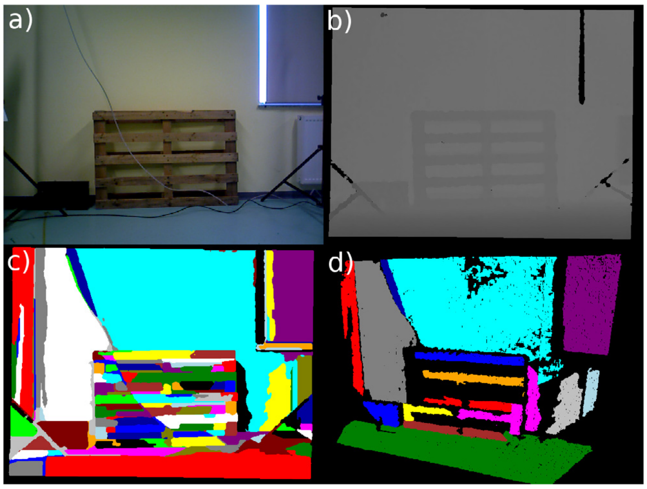

Fig. 5. Extraction of planar surfaces: RGB image (a), depth image (b), planar patches segmented using RGB and depth information (c), and merged segments (d).

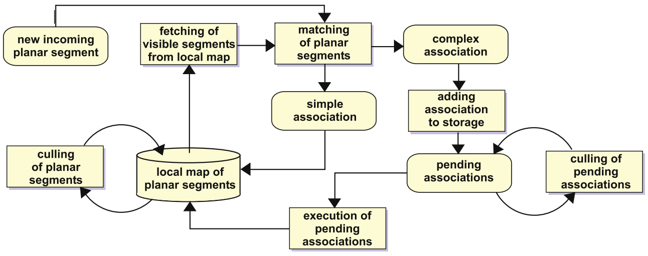
new incoming planar segment
fetching of visible segments from local map
matching of planar segments
complex association
simple association
adding association to storage
culling of planar segments
local map of planar segments
pending associations
culling of pending associations
execution of pending associations

Fig. 6. Block scheme of the local map update and management process. Rounded blocks represent data structures in the processing pipeline.

Algorithm 1 Updating the map of planar features

```
Let PA be short for pending association
Let $\mathcal{S}^s$ be the set of local scene segments
Get visible map segments $\mathcal{S}^m$
for all $\mathbf{s}^s \in \mathcal{S}^s$ do
    Initialize matches set $\mathcal{M} = \emptyset$
    for all $\mathbf{s}^m \in \mathcal{S}^m$ do
        if $\mathbf{s}^s$ and $\mathbf{s}^m$ match then
            $\mathcal{M} = \mathcal{M} \cup \mathbf{s}^m$
        end if
    end for
    if $|\mathcal{M}| = 0$ then
        Add new segment $\mathbf{s}^s$ to map with $l_s = l_{\mathrm{init}}$
    else if $|\mathcal{M}| = 1$ then
        Merge $\mathbf{s}^s$ and $\mathbf{s}^m$
    else
        if $\exists$ PA for $\mathcal{M}$ then
            $l_a \leftarrow l_a + l_{\mathrm{inc}}$
        else
            Add new PA with $l_a = l_{\mathrm{init}}$
        end if
    end if
end for
Merge PAs with $l_a > l_{\mathrm{exe}}$
Decrease PAs' $l_a$ by $l_{\mathrm{dec}}$
Remove PAs with $l_a \leq 0$
for all $\mathbf{s}^m \in \mathcal{S}^m$ do
    if $\mathbf{s}^m$ has not been matched then
        Decrease segment's $l_s$ by $l_{\mathrm{dec}}$
    end if
end for
Remove segments with $l_s \leq 0$
```

of the sensor, but it is not. Conversely, this value is increased by $l_{\mathrm{inc}} = 2$ whenever an observation is matched with the segment. When l_s reaches 0, the planar segment is removed from the map.

The *delayed merging* mechanism is used when a new planar segment is matching two or more map segments at once. Since associating the new planar segment with multiple map segments would effectively merge all these segments, it has to be taken with care because the merging cannot be undone. Thus, we postpone such complex merges until enough evidence is available that they are indeed observations of the same planar segment. All potential complex merges are kept in the storage of pending associations, and each pending association, similarly to planar segments, has its own *end of life value* l_a. The value is decreased by l_{dec} every time a new frame is processed and increased by l_{inc} when an observation that matches the same set of planar segments occurs. Eventually, pending associations are executed when their l_a reaches $l_{\mathrm{exe}} = 6$ or removed from the storage when the value drops down to 0. The whole map management process is summarized by Algorithm 1.

3.3. *Representation and merging*

In the presented system, the measurements have a form of sets of points in 3-D. Having determined which points belong to which planar segment, all other properties of segments are obtained from the analysis of spatial and visual attributes of those points. Therefore, in the following section, we describe how to compute the necessary values.

A pose of an infinite plane in 3-D space can be obtained from analysis of the scattering of the points belonging to the plane. The points should be scattered mainly in two orthogonal directions, while the variation in the third orthogonal direction should be relatively small. This analysis can be accomplished by means of the Singular Value Decomposition (SVD) on a covariance matrix of the positions of points. The eigenvector corresponding to the smallest eigenvalue determines the normal vector **n** of the plane and the position of the points centroid can be used to compute the shortest distance d from the plane to the origin, therefore defining the plane equation:

$$\mathbf{n}\mathbf{q} - d = 0, \tag{2}$$

which holds for all points **q** laying on the plane.

Other parameters describing the planar segment are then derived from the centroid position and the covariance matrix. Therefore, we maintain those values during merging. Having two planar segments with the centroid positions $\boldsymbol{\mu}_i$, $\boldsymbol{\mu}_j$, the covariance matrices $\mathbf{S}_i$, $\mathbf{S}_j$, and the numbers of points n_i, n_j belonging to the segment, we compute the combined centroid position as in [41]:

$$\boldsymbol{\mu} = \frac{n_i\boldsymbol{\mu}_i + n_j\boldsymbol{\mu}_j}{n_i + n_j}, \tag{3}$$

while the covariance is given by:

$$\mathbf{S} = \mathbf{S}_i + \mathbf{S}_j + \frac{n_i n_j}{n_i + n_j}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T. \tag{4}$$

Since every planar segment has some boundaries, another important description of a segment is its hull. Planar segments can have arbitrary shapes, therefore we use non-convex hulls. To compute the hulls, we utilize alpha shapes, which are further processed to obtain 2-D polygons representing boundaries of the planar segment. For computations related to hulls on a plane, we have used the CGAL library.[2] The polygons are then simplified to reduce

[2] https://www.cgal.org/.

the number of points necessary to represent them to speed up the process of finding intersections of two hulls. The number of points for each polygon rarely exceeds 20.

During merging, the hull is re-computed using all data points from both planar segments. To avoid excessive accumulation of the points in segments merged from many observations, the points are filtered using a voxel grid filter after each merging operation. The data points from both merged segments are projected onto the plane of the merged segment. Then, the filter averages all points that were projected into a single voxel. In result, the points belonging to a planar segment retain approximately constant spatial density through the lifetime of this feature.

3.4. Matching

The procedure of determining which planar segment representations are actually observations of the same planar segment is crucial for the map maintenance. It starts with selecting the map segments that should be visible from the current pose of the camera. The planar segments are considered to be one-sided and the normal vectors point towards the camera. The one-side assumption is justified by the fact that we want to use planar surfaces of objects that have some volume. Thus, only these segments from the map that are facing the same direction as the currently perceived ones are considered as visible.

To check, considering occlusions, whether a visible segment could be actually observed, we build a depth buffer for every pixel. For every pixel, we construct a list of planar segments that a ray cast from the camera origin intersects by projecting hulls of every segment onto the image plane. By employing hulls of segments, the projection operation is computationally inexpensive, as the hulls contain a small number of points, which is below 20 for the most cases. Given a ray, we test if it intersects the hull, thus avoiding testing every pixel against every segment. Due to a relatively small number of planes in the map and a limited depth range of the sensor, this process is fast and considerably limits the number of segments involved in further computations. Once the depth buffer is established, we count the number of pixels that a segment could be observed at. We consider only the intersections closest to the RGB-D sensor, and additionally the intersections found within 0.2 m behind those closest because the error of Kinect/Xtion depth measurements rarely exceeds 0.2 m in the assumed depth range [42]. A planar segment is considered visible if it is detected at more than $\tau_{\mathrm{mpix}} = 1000$ pixels. The threshold depends on the accuracy of visual odometry (frame of reference transformation) and was adjusted to eliminate the cases when a correct map segment is considered visible but not matched to any of the detected segments. The threshold prevents from processing segments for which only a small patch is visible and it is not obvious that this segment is indeed in the field of view of the sensor. We use a pixel-based threshold, rather than area-based one because it better reflects the amount of evidence provided by the depth sensor. Furthermore, it does not affect localization itself, because it only prevents merging segments, but does not discard them. This choice follows the policy to avoid merging when there is no strong evidence that the segments are parts of the same structure.

Measuring how similar are the two considered planar segments is not as trivial as in the case of a pair of points. One can measure how similar are the parameters of the infinite planes supporting those segments, but the parameter values depend strongly on the choice of the reference frame. The d value in (2) is the distance from the origin of the coordinate system. If the plane equations are estimated from point clouds located close to this origin, the uncertainty of the estimated directions of the normal vectors does not influence their d values significantly. However, if the planar segments are far from the origin, even a small change in the normal directions changes significantly the d value, as shown in Fig. 7. Therefore, we have decided to refer back to the primary source of information about the plane and measure how much the sets of RGB-D points can differ to still represent the same plane. Assuming that the first planar segment has a centroid $\boldsymbol{\mu}_j$ and normal vector $\mathbf{n}_j$, and the compared planar segment has a centroid $\boldsymbol{\mu}_i$ and a covariance $\mathbf{S}_i$, we define the similarity metric $f_{(i,j)}$ between the ith and jth segment. This metric is computed as a variance of points positions of the segment i in a direction of the normal $\mathbf{n}_j$, but represented with respect to the centroid $\boldsymbol{\mu}_j$:

$$f_{(i,j)} = \frac{1}{n_i}\mathbf{n}_j^T\mathbf{S}_{i\to j}\mathbf{n}_j, \tag{5}$$

where $\mathbf{S}_{i\to j}$ is a covariance of the planar segment i with respect the centroid of the planar segment j:

$$\begin{aligned}\mathbf{S}_{i\to j} &= \sum_{k=1}^{n_i}(\mathbf{q}_k-\boldsymbol{\mu}_j)(\mathbf{q}_k-\boldsymbol{\mu}_j)^T\\ &= \sum_{k=1}^{n_i}(\mathbf{q}_k-\boldsymbol{\mu}_i+\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)(\mathbf{q}_k-\boldsymbol{\mu}_i+\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)^T\\ &= \sum_{k=1}^{n_i}(\mathbf{q}_k-\boldsymbol{\mu}_i)(\mathbf{q}_k-\boldsymbol{\mu}_i)^T+(\mathbf{q}_k-\boldsymbol{\mu}_i)(\mathbf{q}_k-\boldsymbol{\mu}_j)^T\\ &\quad+(\mathbf{q}_k-\boldsymbol{\mu}_j)(\mathbf{q}_k-\boldsymbol{\mu}_i)^T+(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)^T\\ &= \mathbf{S}_i+n_i(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)^T,\end{aligned} \tag{6}$$

where the last step follows from the fact that $\sum_{k=1}^{n_i}\mathbf{q}_k-\boldsymbol{\mu}_i$ is equal to 0.

To recognize two planar segments as the same they have to fulfill the following criteria:

- The angle between the directions of their normal vectors has to be below the threshold $\tau_{\mathrm{mnorm}} = 45°$, which is only a rough test if the segments are faced in the same direction.
- Both, the $f_{(i,j)}$ and $f_{(i,j)}$ metrics have to be below the threshold $\tau_{\mathrm{mmet}} = 0.02$. The value of the threshold depends on the accuracy of the depth measurement and was adjusted to the highest value that did not cause incorrect merging on exemplary segments.
- The difference between their color histograms $h_{(i,j)}$ has to be below the threshold $\tau_{\mathrm{mhist}} - 4.5$, which is a coarse test for appearance similarity. The threshold is adjusted to accept correct matches in exemplary sequences.
- Their hulls have to overlap, which is assumed to be true if a ratio of an area of intersection and an area of the hull is above the threshold $\tau_{\mathrm{marea}} = 0.3$ for one of the segments. This threshold helps to avoid associating segments that are actually disjoint in spite of being located on the same infinite plane and takes into account possible inaccuracy in the sensor pose estimation.

4. Localization with planar segments

4.1. Selecting triplets

Having the current view and the global map represented as planar segments, we pick pairs of segments, one segment from the current view, and another one from the map. Those pairs are potential matches and each of them may be either correct, if the two planar segments indeed represent the same planar surface, or incorrect if they do not. To limit the number of pairs only segments that are visually similar are considered. The appearance of each plane is represented as a histogram of the Hue and Saturation

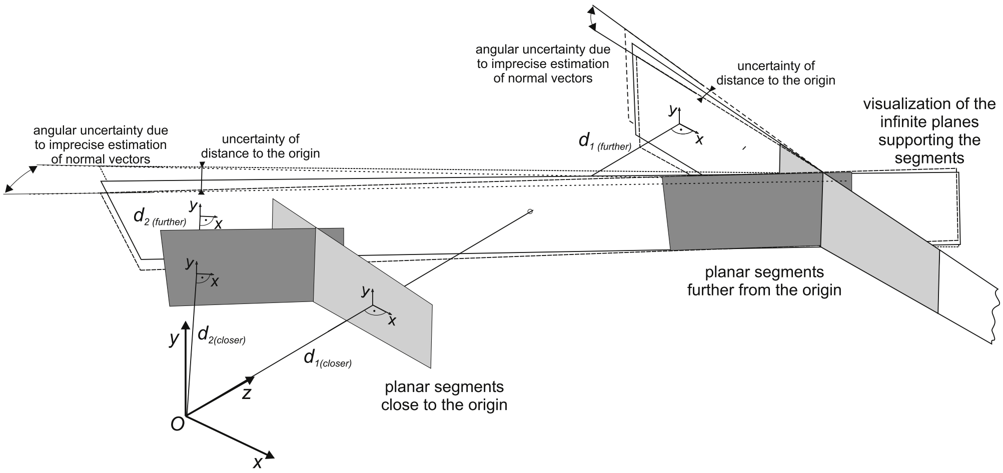
angular uncertainty due to imprecise estimation of normal vectors
uncertainty of distance to the origin
visualization of the infinite planes supporting the segments
angular uncertainty due to imprecise estimation of normal vectors
uncertainty of distance to the origin
$d_{1\,(further)}$
$d_{2\,(further)}$
planar segments further from the origin
$d_{2(closer)}$
$d_{1(closer)}$
planar segments close to the origin
y
x
z
O

Fig. 7. Simplified illustration of the influence the location of the planar segments with respect to the origin of the global coordinate frame on the uncertainty of the plane parameters.

components of the HSV color model and is embedded into a vector $\mathbf{h}_i^s$ for the ith plane from the current view and into a vector $\mathbf{h}_j^m$ for the jth plane from the map. Planar segments i and j are considered similar if the difference between their histograms does not exceed the threshold:

$$h_{(i,j)} = |\mathbf{h}_i^s - \mathbf{h}_j^m| < \tau_{\text{lhist}}. \tag{7}$$

The HSV model, that closely aligns with the way human vision perceives colors, enables the algorithm to distinguish differently colored surfaces, even using a consumer-grade camera with auto-exposure and auto-white balance, such as the one in Microsoft Kinect. The threshold τ_{lhist} is dependent upon the environment's characteristics and its default value is set to 2.5. However, if the number of produced pairs exceeds 275, we dynamically trim this value to select only 275 most similar pairs.

As three pairs allow to compute an SE(3) transformation, we form triplets of pairs that represent a valid transformation if all three matches are correct. Again, to limit the size of the search space, each triplet can be early rejected if it does not fulfill one of the following conditions:

- Each plane i and j has to appear in at most one of three pairs, as the same plane segment cannot be matched more than once.
- The pairwise relations between planar segments have to be the same in the set of map segments and scene segments. An easily verified relation between planar segments is an angle between the induced infinite planes, therefore we check if angles between the planes are approximately equal in both sets (with the margin of $\tau_{\text{lang}} = 15°$). This threshold was adjusted to the minimal value that accepts correct matches and should be increased if a sensor of lower depth measurements accuracy is applied.
- The map planar segments must not be further than $\tau_{\text{ldist}} = 5$ m each from the other. The map can be large in comparison to the current view, therefore if two planar segments are far from each other, they cannot be visible in the same view. The value of τ_{ldist} is correlated with the range and the field of view of the sensor.

4.2. Computing transformations

To evaluate the correctness of the established triplets it is necessary to calculate the alleged SE(3) transformations between the frames of reference of the local view and the global map induced by those triplets. We use a general method that takes as input $n \geq 3$ pairs of planar segments and outputs a transformation given by the translation vector $\mathbf{t} = [t_x \quad t_y \quad t_z]^T$ and the rotation quaternion $\mathbf{r} = [r_x \quad r_y \quad r_z \quad r_w]^T$. The method consists of two steps. At first, it calculates the rotation using normal vectors of the planes $\mathbf{n}_i^s$ and $\mathbf{n}_j^m$, then the translation is obtained using distances from origins d_i^s and d_j^m. A derivation of the rotation calculation algorithm is based upon the method of Walker et al. [43]. The process tries to minimize the differences between current view's normal vectors and transformed map's normal vectors:

$$\begin{aligned}\mathbf{e}_{(i,j)} &= |\mathbf{W}(\mathbf{r})^T\mathbf{Q}(\mathbf{r})\mathbf{n}_j^m - \mathbf{n}_i^s|^2\\ &= \left[\mathbf{W}(\mathbf{r})^T\mathbf{Q}(\mathbf{r})\mathbf{n}_j^m - \mathbf{n}_i^s\right]^T\left[\mathbf{W}(\mathbf{r})^T\mathbf{Q}(\mathbf{r})\mathbf{n}_j^m - \mathbf{n}_i^s\right]\\ &= (\mathbf{n}_j^m)^T\mathbf{Q}(\mathbf{r})^T\mathbf{W}(\mathbf{r})\mathbf{W}(\mathbf{r})^T\mathbf{Q}(\mathbf{r})\mathbf{n}_j^m\\ &\quad - 2\mathbf{n}_i^s\mathbf{W}(\mathbf{r})^T\mathbf{Q}(\mathbf{r})\mathbf{n}_j^m + (\mathbf{n}_i^s)^T\mathbf{n}_i^s\\ &= 2\left[1 - \mathbf{r}^T\mathbf{Q}(\mathbf{n}_i^s)^T\mathbf{W}(\mathbf{n}_j^m)\mathbf{r}\right].\end{aligned} \tag{8}$$

The total energy to minimize is given by the equation:

$$E_r = \sum_{(i,j)} 2\left[1 - \mathbf{r}^T\mathbf{Q}(\mathbf{n}_i^s)^T\mathbf{W}(\mathbf{n}_j^m)\mathbf{r}\right] = \mathbf{r}^T\mathbf{C}\mathbf{r} + \text{const.}, \tag{9}$$

where: $\mathbf{C} = -2\sum_{(i,j)}\mathbf{Q}(\mathbf{n}_i^s)^T\mathbf{W}(\mathbf{n}_j^m)$. To constrain rotation quaternion to the length 1, a Lagrangian multiplier λ was introduced to the equation:

$$\tilde{E}_r = \mathbf{r}^T\mathbf{C}\mathbf{r} + \lambda(\mathbf{r}^T\mathbf{r} - 1), \tag{10}$$

which can be solved by taking the derivative with respect to $\mathbf{r}$:

$$\frac{\partial\tilde{E}_r}{\partial\mathbf{r}} = (\mathbf{C} + \mathbf{C}^T)\mathbf{r} + 2\lambda\mathbf{r} = 0. \tag{11}$$

After rearrangement we obtain:

$$\underbrace{-\frac{1}{2}(\mathbf{C} + \mathbf{C}^T)}_{\mathbf{D}}\mathbf{r} = \lambda\mathbf{r} \tag{12}$$

where the solution $\mathbf{r}^*$ is given by the eigenvector of matrix $\mathbf{D}$ that corresponds to the largest eigenvalue.

An important question is whether the computed rotation is the only solution to Eq. (12) or there are many possible solutions because the planes do not impose enough constraints. It turns out that the solution to (12) is unambiguous if the largest eigenvalue has no counterpart with a negative sign. This additional criterion is induced by the assumption that the segments are one-sided. Only equations for a fully constrained rotation do not have a solution where normal vectors are pointing in the opposite directions, maximizing the energy, instead of minimizing it.

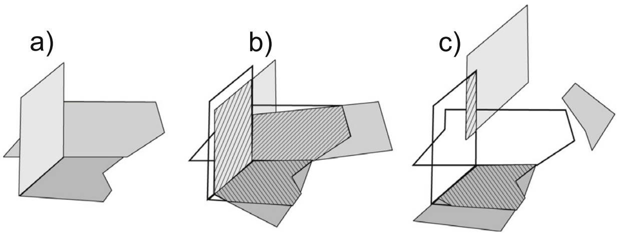

Fig. 8. Using simplified hulls of the planar segments in matching: triplet of planar segments from the local view (a), aligned local and global planar segments with overlapping hulls — slashed areas are the overlapping ones (b), and aligned local and global planar segments with hulls that are not overlapping enough for an acceptable match (c).

Computing the translation between frames of reference is done by minimizing a square errors between current view's plane distances and the transformed global map's plane distances [44]:

$$E_t = \sum_{(i,j)} (d_i^s - d_j^m - (\mathbf{n}_j^m)^T \mathbf{t})^2 = |\mathbf{b} - \mathbf{A}\mathbf{t}|^2, \tag{13}$$

where:

$$\mathbf{A} = \left[\mathbf{n}^m_{j,1}, \dots, \mathbf{n}^m_{j,n}\right]^T, \tag{14}$$

$$\mathbf{b} = \left[d^s_{i,1} - d^m_{j,1}, \dots, d^s_{i,n} - d^m_{j,n}\right]^T. \tag{15}$$

The solution is obtained using SVD decomposition with an additional assertion of the **A** matrix rank to determine if enough constraints are present to compute the translation.

4.3. Evaluating transformations

Unfortunately, as the transformation computation is only minimizing the errors, the solution may actually be implausible in case of a mismatch between the segments. It is the case when the system of equations imposed by the planar constraints is inconsistent. Therefore, each calculated transformation has to be tested whether it is valid or not. This is a preliminary step, before selecting the final correct transformation, that eliminates impossible transformations. This step consists of evaluating two criteria:

- the plane representation projection criterion,
- the overlapping of the hulls criterion.

If the transformation is valid, a representation of map's plane, transformed to the current view's frame, should be approximately equal to the actual one of the current view's plane for each pair of segments. The difference between those representations is computed using the similarity metric (5) and has to be below a threshold $\tau_{\text{lmet}} = 0.02$:

$$f_{(i,j)} < \tau_{\text{lmet}} \wedge f_{(j,i)} < \tau_{\text{lmet}}. \tag{16}$$

Same as τ_{mmet}, the value of the threshold depends on the accuracy of the depth measurement.

Planes are infinite and the computed transformation can be still implausible, despite similar plane representations. It is often the case that the solution point to a distant location, far away from the investigated environment. Therefore, it is also necessary to check if the transformation is justified by the available observations. We assume the solution to be plausible when the hulls of the points belonging to the map planar segments, denoted by $\eta(\mathcal{P}^m_j)$, after transformation to the current view frame of reference, overlap with the hulls of the points belonging to the current view segments, denoted by $\eta(\mathcal{P}^s_i)$:

$$\mathcal{G}_{(i,j)} = \left[\mathbf{T}^{-1}_{m,s}\eta(\mathcal{P}^m_j)\right] \cap \eta(\mathcal{P}^s_i), \tag{17}$$

where $\mathbf{T}_{m,s}$ denotes a homogeneous transformation matrix from the map frame to the current view frame. In other words, we check if for each pair of infinite planes the same segment of the plane is observed and represented in the global map and the current scene (Fig. 8). The area of the co-observed part has to be a certain fraction of the smaller convex hull to take into account inaccurate location estimation:

$$\frac{\alpha\left(\mathcal{G}_{(i,j)}\right)}{\min\left\{\alpha\left(\eta(\mathcal{P}^m_j)\right), \alpha\left(\eta(\mathcal{P}^s_i)\right)\right\}} > \tau_{\text{larea}}, \tag{18}$$

where $\alpha(.)$ is a function returning the area of the hull and $\tau_{\text{larea}} = 0.3$.

4.4. Assigning weights

The set of triplets $\mathcal{T}$ generated in the previous processing steps is then included in the probabilistic framework. Each triplet is scored and the computed scores are treated as weighting factors used to build a PDF. The final outcome of the method is the SE(3) transformation, which is the most probable according to the supporting evidence.

To this processing stage, we have at our disposal a set $\mathcal{T}$ of triplets along with corresponding transformations that exclude implausible ones. But not every triplet has the same value in terms of a correct transformation inference. Due to this fact, we have to assign a weight to each triplet, before we will be able to construct a probability distribution that will reflect our whereabouts knowledge. The triplet weight takes into consideration the appearance difference $h_{(i,j)}$ (7) and the area of the convex hulls intersection $\alpha(\mathcal{G}_{(i,j)})$ (17). Moreover, the weight depends on how frequently the respective plane segment was employed, as the same plane can be a part of multiple triplets and therefore can introduce a bias to the PDF. We handle it by calculating an occurrence factor for each triplet a and each pair $(i, j) \in \mathcal{S}_a$ within the triplet. The more triplets including the same plane in a vicinity of induced transformation, the smaller the weight:

$$w_{a,(i,j)} = \left[\sum_{b\in\mathcal{T}}\sum_{(k,l)\in\mathcal{S}_b} \mathbb{I}_{i=k}\exp(-y_{(a,b)})\right]^{-1}, \tag{19}$$

where $\mathbb{I}_{i=k}$ is an indicator function equal to 1 whenever $i = k$ and 0 otherwise, and $y_{(a,b)}$ is a squared norm of an SE(3) logarithm map of a difference between transformations $\mathbf{T}$ for the triplets a and b:

$$y_{(a,b)} = \left|\log(\mathbf{T}_a^{-1}\mathbf{T}_b)\right|^2. \tag{20}$$

The overall weight for the triplet a is expressed by the equation:

$$w_a = \sum_{(i,j)\in\mathcal{S}_a} \alpha(\mathcal{G}_{(i,j)}) w_{a,(i,j)} \exp(-h_{(i,j)}). \tag{21}$$

4.5. Constructing localization distribution

The SE(3) transformations induced by triplets are represented as points in a 6-D space. These points should form clusters near plausible locations and the largest cluster should be near the correct location. We consider each transformation as a hint that the correct localization is in the vicinity of this transformation. The strength of the contribution is controlled by the weight given by (21) and we convert this contribution to the probabilistic domain by placing a weighted Gaussian kernel in each transformation point. The final PDF is, therefore, a sum of all kernels:

$$p(\mathbf{x}) = \frac{1}{Z}\tilde{p}(\mathbf{x}) = \frac{1}{Z}\sum_{a\in\mathcal{T}} K_a(\mathbf{x}), \tag{22}$$

where Z is a normalizing constant, and the kernel is:

$$K_a(\mathbf{x}) = w_a \exp\left\{-\log(\mathbf{T}_x^{-1}\mathbf{T}_a)^T \mathbf{I}_a \log(\mathbf{T}_x^{-1}\mathbf{T}_a)\right\}. \tag{23}$$

The distance between the kernel's center, defined by the 6-D coordinates $\mathbf{x}$ and represented by homogeneous matrix $\mathbf{T}_x$, and the 6-D point for triplet a, represented by the homogeneous matrix $\mathbf{T}_a$, is computed using logarithm map. The logarithm map is embedded in the square form of the multidimensional Gaussian distribution with the information matrix $\mathbf{I}_a$. The role of the information matrix is to scale the kernel along the dimensions of the 6-D space, i.e. with respect to the particular degrees of freedom. Since the triplets of planar segments we use for localization always provide constraints in all six degrees of freedom, we use identity matrices as $\mathbf{I}_a$. However, this can change if we introduce partial localization cues, e.g. from inertial sensors that yield only an orientation of the agent, or from WiFi fingerprinting that provides its 2-D position.

Having a probability distribution, we seek the point with the highest probability. Inference in general distributions can be complicated and to avoid this, we exploit the fact that we already have a list of possible solutions. For each triplet, we evaluate the transformation induced by this candidate and choose the one with the highest probability, denoted further as $\mathbf{x}^*$. To decide that the transformation $\mathbf{x}^*$ is correct it has to pass three tests that ultimately verify the recognition. First of all, its probability has to exceed a certain threshold:

$$\tilde{p}(\mathbf{x}^*) > \tau_{\text{lprob}} \tag{24}$$

This condition checks if there is enough evidence to be confident about the location. The value of the threshold was adjusted on a subset of local scenes to eliminate false positives.

The second test controls whether there are redundant constraints that the transformation was computed from. Three pairs of infinite planes (represented by their respective segments) are sufficient to compute a transformation, but no information is left to verify that transformation. Therefore, we check how many distinct planes can be matched using the transformation $\mathbf{x}^*$. We transform current view's planar segments, perform matching with map segments and examine the sets of matched segments. For both, the map and the current view, we check the number of distinct planes using the criterion for $f_{(i,j)}$ values. Note that we do not count distinct planar segments, but distinct planes to avoid counting multiple times e.g. a wall split into a few planar segments. The number of distinct planes has to be above a threshold τ_{ldiv}. According to our experience, the τ_{ldiv} value of 6 rejects nearly all incorrect recognitions that have not been eliminated by the first criterion.

The last test is the fitness score test. It refers back to the point clouds and determines if the current view's point cloud transformed to the map's frame of reference is aligned with the actual map's one. This test involves computation of the sum of squared distances from each point of the transformed current view's point cloud $\mathcal{P}^s$ to its closest neighbor in the point cloud of the global map $\mathcal{P}^m$. This residual distance has to be below a threshold:

$$\sum_{\mathbf{q}_l^s \in \mathcal{P}^s} (\mathbf{T}_{\mathbf{x}^*}\mathbf{q}_l^s - \hat{\mathbf{q}}_l^m)^2 < \tau_{\text{lres}}, \tag{25}$$

where $\hat{\mathbf{q}}_l^m$ is the nearest neighbor in map's point cloud. The threshold was adjusted on a subset of local scenes used for initial tests to produce no false positives. It is worth noting that the fitness score test examines only points belonging to planar segments, and is much faster than the popular Iterative Closest Points (ICP) algorithm, because (25) is equivalent to a single iteration of ICP on a reduced size point cloud.

If the transformation fulfills all three conditions, it is assumed to be the correct one.

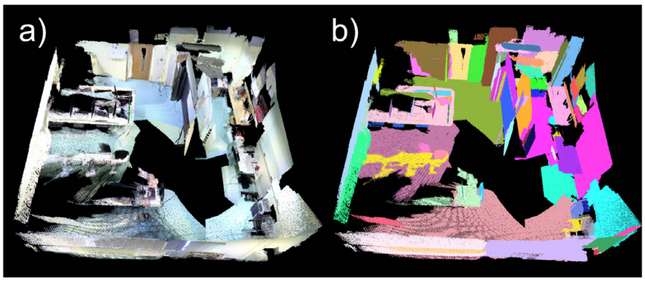

Fig. 9. Visualization of the PUT RGB-D/Workshop environment: colored point cloud made from three RGB-D sequences registered in a common frame (a), and the global map of planar features built from these three sequences (b). Note that individual planar segments are represented by the measured points that belong to these segments, shown in random colors.

5. Experimental evaluation

The PlaneLoc system has been evaluated in the global localization task with an *a priori* known map. Because PlaneLoc is not yet integrated within an RGB-D SLAM system (e.g. PUT SLAM [5]), we have employed the state-of-the-art ORB-SLAM2 [6] software to obtain reliable visual odometry for the incremental construction of local, segment-based maps.

Besides a representation of the local view in terms of planar segments, to perform global localization, the PlaneLoc system requires a global map, also composed of planar segments. This requirement makes it difficult to evaluate PlaneLoc on a publicly available data set that was already used by other localization or SLAM systems. Unfortunately, RGB-D data sets obtained experimentally usually do not contain ground truth for the observed objects, thus it is not possible to prepare a "ground truth" global map of planar segments. Such a perfect *a priori* map is in principle not available also if the localization system works online with an RGB-D sensor. Even a map generated from an up-to-date CAD model of the environment can be imperfect due to unmodeled or non-static objects. Therefore, in the presented experiments we build the global maps off-line using the described planar segments extraction and map management methods. This approach is analogous to a situation when a SLAM system is used to build a global map and then this map is re-used for localization. However, when using a data set that contains ground truth trajectories for the RGB-D sensor (which is common), we use these ground truth trajectories instead of the SLAM-estimated ones to increase the accuracy of the global map.

Regarding local maps, using a single RGB-D frame to build them is not enough, because a sufficient number of planar segments has to be detected. Hence, to widen the local view context, we use the same procedure as for global map building to obtain a set of planar segments from 50 consecutive RGB-D frames. In this case, the necessary visual odometry is delivered by the ORB-SLAM2 system. Considering the Kinect frame rate, the short sequence used in local mapping takes less than 2 s and in most cases, the trajectory does not accumulate significant drift, despite enforcing no loop closures within this short time window. Creating both the global segment-based maps and the local sets of planar segments we use only depth measurements acquired in the ranges between 0.2 and 4 m. This is motivated by the depth noise characteristics of the commonly used RGB-D sensors based on structured light technology [42]. These sensors cannot measure very close objects, whereas errors in depth measurements increase with distance from the sensor. Hence, measurements in the range above 4 meters would rather corrupt the perceived geometry than contribute to correct localization.

For evaluation, we have employed two data sets: the publicly available NYUv2 [45], and the PUT RGB-D/Workshop,[3] which has

[3] Available at http://lrm.put.poznan.pl/rgbdw/.

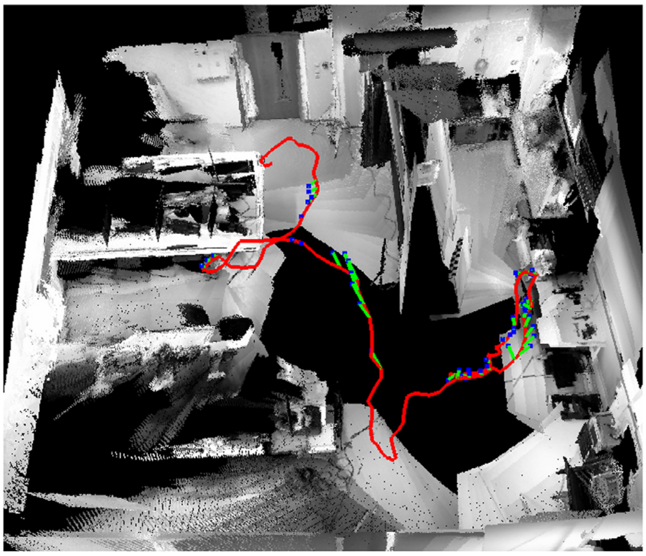

Fig. 10. Global localization results for the PUT RGB-D/Workshop data set *seq2* for $\tau_{\text{lprob}} = 1.0$, $\tau_{\text{lres}} = 0.07$, and $\tau_{\text{ldiv}} = 6$. Test trajectories are marked with red lines, recognized places with blue dots, while green lines connect the recognized locations to their respective ground truth poses (for interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

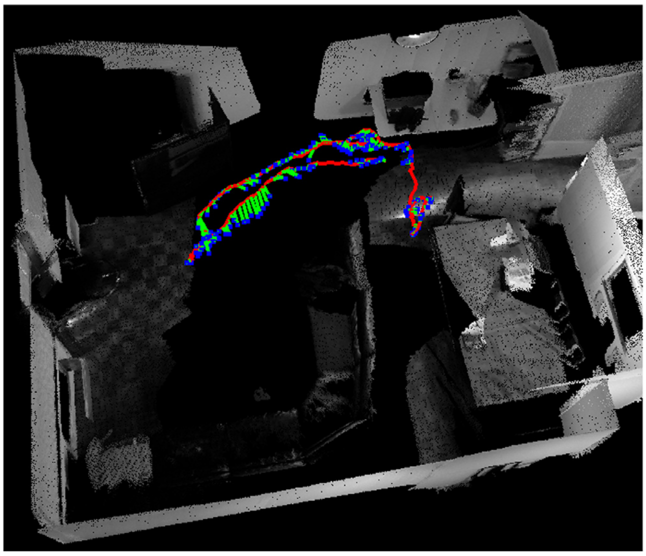

Fig. 11. Global localization results for the NYUv2 data set for $\tau_{\text{lprob}} = 1.0$, $\tau_{\text{lres}} = 0.07$, and $\tau_{\text{ldiv}} = 6$. Test trajectories are marked with red lines, recognized places with blue dots, while green lines connect the recognized locations to their respective ground truth poses (for interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

been prepared specifically to support this research. The NYUv2 data set was acquired in a typical household environment and is aimed at evaluating semantic segmentation. Since no ground truth for the sensor trajectory is provided in this data set, we used the sensor trajectory computed by ORB-SLAM2 to build a global map of segments from these data. The map contained 55 planar segments. The PUT RGB-D/Workshop data set was acquired in an 8 × 8 meters robotic workshop at PUT (Fig. 9a), with the ground truth sensor trajectory obtained using the OptiTrack motion capture system. This data set consists of 10 sequences (*seq1* to *seq10*). Three non-overlapping ground truth trajectories and their corresponding RGB-D sequences (*seq5*, *seq6*, and *seq7*) were used to build a global map that contains 93 segments (Fig. 9b).

Our motivation for using a motion capture system in the global map building process was the need for a good quality *a priori* map to test global localization only, which was the main focus of the presented research. However, we are aware that ground truth sensor trajectories (e.g. from OptiTrack) rarely are available in practical applications of indoor localization systems. Thus, if SLAM is used to produce the global map, the relative pose errors (RPE in the sense of the localization quality metrics defined in [46]) in the trajectory would have to be bound, to avoid wrong alignment of the planar segments extracted from consecutive RGB-D frames in the matching procedure (see Section 3.4). Specifically, the translational RPE should have to be smaller than the 0.2 m tolerance defined in the depth buffer, as the error in sensor translation adds here to the depth sensor error. The rotational RPE must have not exceeded 5° to ensure that tests based on the normal vectors of the segments work properly. It should be noticed that in our experiments the RPE values achieved by ORB-SLAM2 were much smaller than these thresholds, for example on the *seq1* trajectory of the PUT RGB-D/Workshop data set the translational RPE MRSE was 0.01 m, while the rotational RPE MRSE was only 0.52°.

In all presented experiments the localization performance was measured by counting the locations that were correctly recognized (true positives — TP), those that were recognized incorrectly (false positives — FP), and those not recognized at all (negatives — N). For the performance tests, we applied PlaneLoc to every 10-th pose of the investigated sequence, attempting to localize the sensor with respect to the entire global map. The metrics given by Eq. (20) was used to tell if the estimated sensor pose is acceptable with respect to the ground truth pose. For all sequences sensor poses that were distant at most 0.16 from the respective ground truth pose have been considered as correct. This particular threshold was chosen because our experience with ORB-SLAM2 and PUT SLAM [5] shows that re-localization within this range usually enables to recover tracking. We believe that this value should be suitable also for other similar SLAM systems.

Additionally, mean Euclidean $\bar{d}$ and angular $\bar{\alpha}$ distances between the estimated sensor poses and the ground truth trajectory were computed in all experiments. Results are summarized in Table 1, where three sets of parameters were evaluated: with all consistency tests, with the point cloud fitness test (25) switched off, and with the number of distinct planes test switched off. Visualizations of the recognized places are presented in Figs. 10 and 11, for the PUT RGB-D/Workshop and NYUv2 data sets, respectively.

The results obtained using two different data sets indicate that the proposed method is reliable, finding a large number of locations along the test trajectories. In contrast to our preliminary results [33], the cloud alignment test is no longer the main fail-safe against incorrect recognitions. The introduced test for distinct planes took over most of this functionality. If the algorithm is parametrized with the τ thresholds chosen according to the guidelines given in the paper, it produces no false positives, which is of pivotal importance in the localization task. The main cause for the number of places (local views) that remained unrecognized was the absence of sufficient planar segments to constrain the pose in all 6 degrees of freedom. This suggests that planar segments should be further augmented with other features, such as line segments (edges).

The mean computing time for a single global localization act in the experiments was 19 s on a Core i5-3230M 2.6 GHz laptop. A more detailed breakdown of the computation time with respect to the four main blocks of the PlaneLoc architecture (cf. Fig. 3) yields the following results (averaged over all experiments): single frame processing — 736 ms, generating the local observation — 518 ms, computing the pose likelihood — 16 439 ms, and final localization — 2255 ms. From these figures, it is clear that the

Table 1
Global localization results for three RGB-D sequences and three different parameter sets. Setting $\tau_{\mathrm{lres}} = \infty$ or $\tau_{\mathrm{ldiv}} = 0$ means that the respective tests are switched off.

Localization	Sequence	Parameters			Statistics					
system		τ_{lprob}	τ_{lres}	τ_{ldiv}	TP	FP	N	$\bar{d}$ [m]	$\bar{\alpha}$ [°]	$l_{\max}$
PlaneLoc	PUT RGB-D/W *seq1*	1.0	0.07	6	32	0	184	0.136	2.55	370
		1.0	∞	6	32	0	184	0.136	2.55	370
		1.0	0.07	0	57	14	145	0.638	18.64	330
	PUT RGB-D/W *seq2*	1.0	0.07	6	48	0	118	0.125	2.09	264
		1.0	∞	6	48	1	117	0.199	3.845	264
		1.0	0.07	0	60	7	99	0.342	12.680	264
	NYUv2 *living room 3*	1.0	0.07	6	176	0	47	0.090	1.59	–
		1.0	∞	6	176	0	47	0.090	1.59	–
		1.0	0.07	0	200	1	22	0.100	2.03	–
ORB-SLAM2	PUT RGB-D/W *seq1*	–	–	–	66	0	155	0.145	1.65	570
	PUT RGB-D/W *seq2*	–	–	–	74	0	97	0.139	1.86	300

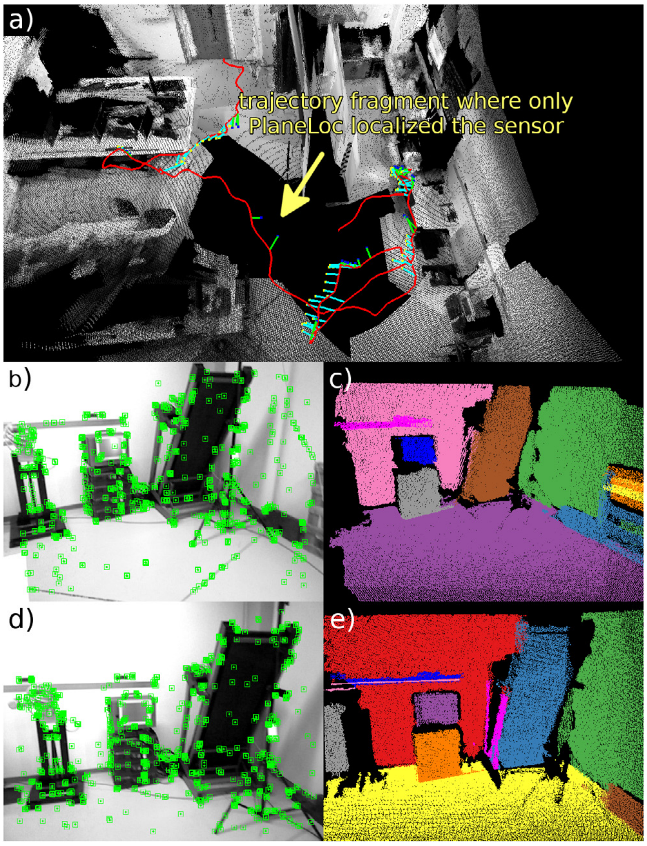

Fig. 12. Example locations that ORB-SLAM2 was not able to re-localize in (neither in their wider neighborhood): visualized trajectory with PlaneLoc (blue dots) and ORB-SLAM2 (yellow dots) recognitions (a), RGB frames with ORB features marked as green rectangles (b), (d), and their counterpart local maps of planar segments that PlaneLoc successfully localized within the global map (c), (e) (for interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

most time-consuming step (16 s in average, 35 s maximum when many similar combinations of planar segments were present in an environment with repetitive geometric structures) is the generation of candidate triplets and computation of transformations for these triplets. Fortunately, those steps can be made much faster by parallelization on a GPU, as these computations are independent. They can be scheduled to separate threads without any synchronization between them, e.q. using CUDA framework. With Nvidia GTX Titan it is possible to run simultaneously up to 2688 threads, which should enable the algorithm to run fast enough to be used as a relocalization or loop-closing module. It should be noticed that although PlaneLoc can generate a large number of individual localization cues that contribute to the final PDF, we have observed that the average number of triplets included in the final PDF for correct recognitions was only 111.

To demonstrate the advantages of PlaneLoc over the classic approach to re-localization in SLAM, we have processed the PUT RGB-D/Workshop sequences with ORB-SLAM2. Since ORB-SLAM2 uses its own map of point features, such a map has been built using the *seq5*, *seq6*, and *seq7* sequences. To ensure a fair comparison to PlaneLoc that uses a global map obtained from ground truth trajectories, we have used the same ground truth data to correct poses of keyframes in ORB-SLAM2.

Then, we have investigated if ORB-SLAM2 is able to re-localize at every 10-th frame employing this known map. Sequences *seq1* and *seq2* were used for re-localization evaluation. The estimated sensor poses were accepted or rejected using the same metrics and the threshold value of 0.16 as for PlaneLoc. In general, the re-localization in ORB-SLAM2 performed well in the feature-rich workshop environment, yielding accurate sensor pose estimates at the same locations as PlaneLoc. However, there were few areas, particularly in the *seq1* sequence, that ORB-SLAM2 could not re-localize in, whereas PlaneLoc managed to produce acceptable sensor poses in these parts of the trajectory (Fig. 12). For *seq1* ORB-SLAM2 did not recognize 155 frames and among them were 14 frames recognized by PlaneLoc. For those frames, the maximal error, measured using (20), was 0.104 and average error was 0.039, where the threshold was 0.16. In case of *seq2* there were 97 frames unrecognized by ORB-SLAM2, 18 of them were recognized by PlaneLoc with the maximum error of 0.111 and the average error of 0.035. Although the total number of recognized locations (TP) was bigger for ORB-SLAM2 in both trajectories considered for this comparison, the trajectory segments in which the systems could not localize were shorter for PlaneLoc. This is shown by the $l_{\max}$ metrics, that stands for the maximum number of consecutive frames without a correct recognition.

6. Discussion and conclusions

This paper considers the global localization problem in 6 d.o.f. applying a novel probabilistic approach that creates a PDF describing the likelihood of the agent pose from local and weak cues. The proposed implementation, called PlaneLoc, uses sets of local planar segments extracted from RGB-D data as the cues.

In opposition to many RGB-D and visual SLAM systems that are available as open source and can be used for performance comparison, no comparable RGB-D global localization software was accessible to us during the presented experiments. Therefore, it was not possible to directly compare PlaneLoc to other global localization solutions on the same data sets. However, the experimental results demonstrated that PlaneLoc performs well in room-sized environments, yielding correct and accurate pose estimates. In particular, the comparison to ORB-SLAM2 in the re-localization task demonstrated that PlaneLoc to a less extent depends on the local appearance of the environment than a typical approach to re-localization in SLAM, based on a combination of visual place recognition and pose estimation using point features. In practice, PlaneLoc in its current variant cannot use a depth-only sensor, because the color data are necessary to improve the efficiency of segmentation into planar patches (described by Eq. (1)), and are essential for early rejection of incompatible pairs of planar segments (using the color histograms). However, the similarity of visual appearance between the *a priori* map and the images acquired during localization is not at the core of the computations.

Contrarily, our approach depends more on the geometry of the environment that should have enough planar structures to build an *a priori* map and to extract planar segments at local views. For example, in a long corridor without any distinguishable local structures, PlaneLoc cannot find triplets of planar segments that constraint the pose hypothesis in all six degrees of freedom. When encountering geometric symmetries in the environment, particularly combined with a limited number of detected triplets, our approach cannot distinguish between a number of the agent pose hypotheses having equal probability. These situations can be to some extent circumvented by enlarging the perceived local environment, e.g. using a sensor with a bigger range of the depth measurements, that can see features outside the area of degenerate geometry. Additionally, localization cues obtained from inertial or WiFi sensors can substitute the agent pose constraints not available in the degenerate cases. On the other hand, visual appearance similarity tests that are more discriminative than the simple color histograms could help to distinguish between objects of similar geometry. The current implementation of PlaneLoc, using only a Kinect/Xtion RGB-D sensor is most suitable for man-made environments, such as offices, labs, and industrial shop floors, which are rich in local geometric structures.

A more general advantage of the proposed probabilistic framework is that not only paired planar segments can contribute to the PDF. The framework can easily accommodate localization cues coming from other geometric features, or even from other sensing modalities, e.g. an orientation sensor like Inertial Measurements Unit (IMU). Although the application of different sensors is not a unique feature among the localization algorithms, an advantage of our approach is that sensors of very different modality can be used together, as long as they generate localization cues (also partial) in the common probabilistic framework. Our approach favors modularity, as the localization mechanism remains exactly the same, no matter what types of sensors or processing algorithms are applied. Although a number of parameters should be set properly to obtain useful localization cues, most of these parameters are specific to the particular localization cues and they depend on the characteristics of the sensory data. In this article the parameters of a module that generates 6 d.o.f. cues from RGB-D data using planar segments have been discussed and explained in relation to the depth data characteristics.

Also, the applications of the presented approach are not limited to global localization and re-localization in SLAM. The methods developed for extraction and merging of planar segments can be easily adapted for matching planar features in a graph-based SLAM framework, where poses of planar segments in the global map are updated by optimizing parameters of the infinite planes supporting these segments, together with the optimized sensor poses [47]. Note that this 6 d.o.f. localization method can be applied to a broad range of robots and autonomous agents, from smartphones with compact RGB-D sensors (Intel RealSense), through different legged robots, to quadcopters operating indoors.

Further research focuses on decreasing the localization time to a practically acceptable value (less than 5 s) by optimizing the code and implementing selected operations on the GPU. Then, we plan to extend PlaneLoc adding edge features, and finally, integrating IMU localization cues. The independent orientation cues should enable pose estimation from local views that do not contain geometric features constraining all six degrees of freedom.

Acknowledgment

This work was supported by the National Centre for Research and Development (NCBiR), Poland, Contract No LIDER/33/0176/L-8/16/NCBR/2017.

References

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, J.J. Leonard, Past, present, and future of simultaneous localization and mapping: Towards the robust-perception age, IEEE Trans. Robot. 32 (6) (2013) 1309–1332.

[2] S. Lowry, N. Sünderhauf, P. Newman, J.J. Leonard, D. Cox, P. Corke, M.J. Milford, Visual place recognition: A survey, IEEE Trans. Robot. 32 (1) (2016) 1–19.

[3] R. Mur-Artal, J.M.M. Montiel, J.D. Tardós, ORB-SLAM: a versatile and accurate monocular SLAM system, IEEE Trans. Robot. 31 (5) (2015) 1147–1163.

[4] M. Kraft, M. Nowicki, A. Schmidt, M. Fularz, P. Skrzypczyński, Toward evaluation of visual navigation algorithms on RGB-D data from the first- and second-generation kinect, Mach. Vis. Appl. 28 (1) (2017) 61–74.

[5] D. Belter, M. Nowicki, P. Skrzypczyński, Improving accuracy of feature-based RGB-D SLAM by modeling spatial uncertainty of point features, in: Proc. IEEE International Conference on Robotics and Automation, Stockholm, 2016, pp. 1279–1284.

[6] R. Mur-Artal, J.D. Tardós, ORB-SLAM2: An open-source SLAM system for monocular, stereo and RGB-D cameras, IEEE Trans. Robot. 33 (5) (2017) 1255–1262.

[7] M. Kopicki, R. Detry, M. Adjigble, R. Stolkin, A. Leonardis, J.L. Wyatt, One-shot learning and generation of dexterous grasps for novel objects, Int. J. Robot. Res. 35 (8) (2016) 959–976.

[8] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, J. Tardós, A comparison of loop closing techniques in monocular SLAM, Robot. Auton. Syst. 57 (12) (2009) 1188–1197.

[9] M. Cummins, P. Newman, FAB-MAP: Probabilistic localization and mapping in the space of appearance, Int. J. Robot. Res. 27 (6) (2008) 647–665.

[10] J. Sivic, A. Zisserman, Video Google: a text retrieval approach to object matching in videos, in: IEEE International Conference on Computer Vision, vol. 2, Nice, 2003, pp. 1470–1477.

[11] M. Cummins, P. Newman, Appearance-only SLAM at large scale with FAB-MAP 2.0, Int. J. Robot. Res. 30 (2011) 1100–1123.

[12] M.R. Nowicki, J. Wietrzykowski, P. Skrzypczyński, Real-time visual place recognition for personal localization on a mobile device, Wirel. Pers. Commun. 97 (1) (2017) 213–244.

[13] D. Galvez-López, J.D. Tardós, Bags of binary words for fast place recognition in image sequences, IEEE Trans. Robot. 28 (5) (2012) 1188–1197.

[14] B. Glocker, J. Shotton, A. Criminisi, S. Izadi, Real-time RGB-D camera relocalization via randomized ferns for keyframe encoding, IEEE Trans. Vis. Comput. Graphics 21 (5) (2015) 571–583.

[15] H. Zhang, Y. Liu, J. Tan, Loop closing detection in RGB-D SLAM combining appearance and geometric constraints, Sensors 15 (6) (2015) 14639–14660.

[16] Z. Chen, O. Lam, A. Jacobson, M. Milford, Convolutional neural network-based place recognition, in: Australasian Conference on Robotics and Automation, Melbourne, 2014, pp. 8–14.

[17] H. Yin, Y. Wang, L. Tang, X. Ding, R. Xiong, LocNet: Global localization in 3D point clouds for mobile robots, 2017, arXiv preprint, CoRR abs/1712.02165.

[18] R. Mur-Artal, J.D. Tardós, Fast relocalisation and loop closing in keyframe-based SLAM, in: Proc. IEEE International Conference on Robotics and Automation, Hong-Kong, 2014, pp. 846–853.

[19] K.O. Arras, J.A. Castellanos, R. Siegwart, Feature-based multi-hypothesis localization and tracking for mobile robots using geometric constraints, in: Proc, IEEE International Conference on Robotics and Automation, vol. 2, 2002, pp. 1371–1377.

[20] D. Fox, W. Burgard, S. Thrun, Active Markov localization for mobile robots, Robot. Auton. Syst. 25 (3–4) (1998) 195–207.

[21] S. Thrun, W. Burgard, D. Fox, Probabilistic Robotics (Intelligent Robotics and Autonomous Agents), The MIT Press, 2005.

[22] R.C. Luo, K.C. Yeh, K.H. Huang, Resume navigation and re-localization of an autonomous mobile robot after being kidnapped, in: IEEE International Symposium on Robotic and Sensors Environments, Washington, 2013.

[23] S. Ito, F. Endres, M. Kuderer, G.D. Tipaldi, C. Stachniss, W. Burgard, W-RGB-D: Floor-plan-based indoor global localization using a depth camera and WiFi, in: Proc. IEEE International Conference on Robotics and Automation, Hong-Kong, 2014, pp. 417–422.

[24] T. Whelan, R.F. Salas-Moreno, B. Glocker, A.J. Davison, S. Leutenegger, ElasticFusion: Real-time dense SLAM and light source estimation, Int. J. Robot. Res. 35 (2016) 1697–1716.

[25] M. Heredia, F. Endres, W. Burgard, R. Sanz, Fast and robust feature matching for RGB-D based localization, 2015, arXiv, CoRR abs/1502.00500.

[26] J. Weingarten, R. Siegwart, 3D SLAM using planar segments, in: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Beijing, 2006, pp. 3062–3067.

[27] M. Kaess, Simultaneous localization and mapping with infinite planes, in: Proc. IEEE International Conference on Robotics and Automation, Seattle, 2015, pp. 4605–4611.

[28] Y. Taguchi, Y.D. Jian, S. Ramalingam, C. Feng, Point-plane SLAM for hand-held 3D sensors, in: Proc. IEEE International Conference on Robotics and Automation, Karlsruhe, 2013, pp. 5182–5189.

[29] L. Ma, C. Kerl, J. Stückler, D. Cremers, CPA-SLAM: Consistent plane-model alignment for direct RGB-D SLAM, in: Proc. IEEE International Conference on Robotics and Automation, Stockholm, 2016, pp. 1285–1291.

[30] J. Biswas, M. Veloso, Depth camera based indoor mobile robot localization and navigation, in: Proc. IEEE International Conference on Robotics and Automation, Saint Paul, 2012, pp. 1697–1702.

[31] M. Dou, L. Guan, J.M. Frahm, H. Fuchs, Exploring high-level plane primitives for indoor 3D reconstruction with a hand-held RGB-D camera, in: Computer Vision – ACCV Workshops, Springer, 2012, pp. 94–108.

[32] C. Feng, Y. Taguchi, V.R. Kamat, Fast plane extraction in organized point clouds using agglomerative hierarchical clustering, in: Proc, IEEE International Conference on Robotics and Automation, Hong-Kong, 2014, pp. 6218–6225.

[33] J. Wietrzykowski, P. Skrzypczyński, A probabilistic framework for global localization with segmented planes, in: Proc. European Conference on Mobile Robots, Paris, 2017, pp. 1–6.

[34] T.T. Pham, M. Eich, I. Reid, G. Wyeth, Geometrically consistent plane extraction for dense indoor 3D maps segmentation, in: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Daejeon, 2016, pp. 4199–4204.

[35] K. Pathak, A. Birk, N. Vaskevicius, J. Poppinga, Fast registration based on noisy planes with unknown correspondences for 3D mapping, IEEE Trans. Robot. 26 (3) (2010) 424–441.

[36] R. Cupec, E.K. Nyarko, D. Filko, A. Kitanov, I. Petrović, Global localization based on 3D planar surface segments detected by a 3D camera, in: Proc. of the Croatian Computer Vision Workshop, Zagreb, 2013, pp. 31–36.

[37] E. Fernández-Moral, W. Mayol-Cuevas, V. Arévalo, J. González-Jiménez, Fast place recognition with plane-based maps, in: Proc. IEEE International Conference on Robotics and Automation, Karlsruhe, 2013, pp. 2719–2724.

[38] F. Ribeiro, S. Brandão, J.P. Costeira, M. Veloso, Global localization by soft object recognition from 3D partial views, in: Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, Hamburg, 2015, pp. 3709–3714.

[39] R.F. Salas-Moreno, B. Glocken, P.H.J. Kelly, A.J. Davison, Dense planar SLAM, in: IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 2014, pp. 157–164.

[40] P.F. Felzenszwalb, D.P. Huttenlocher, Efficient graph-based image segmentation, Int. J. Comput. Vis. 59 (2) (2004) 167–181.

[41] J. Saarinen, H. Andreasson, T. Stoyanov, A.J. Lilienthal, 3D normal distributions transform occupancy maps: An efficient representation for mapping in dynamic environments, Int. J. Robot. Res. 32 (14) (2013) 1627–1644.

[42] K. Khoshelham, S. Elberink, Accuracy and resolution of kinect depth data for indoor mapping applications, Sensors 12 (2) (2012) 1437–1454.

[43] M.W. Walker, L. Shao, R.A. Volz, Estimating 3-D location parameters using dual number quaternions, CVGIP: Image Understanding 54 (3) (1991) 358–367.

[44] O.D. Faugeras, M. Hebert, A 3-D recognition and positioning algorithm using geometrical matching between primitive surfaces, in: Proc. of the Eighth International Joint Conference on Artificial Intelligence - Volume 2, 1983, pp. 996–1002.

[45] N. Silberman, D. Hoiem, P. Kohli, R. Fergus, Indoor segmentation and support inference from RGBD Images, in: Proceedings of the 12th European Conference on Computer Vision, Springer, Berlin, 2012, pp. 746–760.

[46] J. Sturm, N. Engelhard, F. Endres, W. Burgard, D. Cremers, A benchmark for the evaluation of RGB-D SLAM systems, in: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots & Systems, Vilamoura, Portugal, 2012, pp. 573–580.

[47] J. Wietrzykowski, On the representation of planes for efficient graph-based SLAM with high-level features, J. Autom. Mobile Robot. Intell. Syst. 10 (3) (2016) 3–11.

Jan Wietrzykowski graduated from Poznan University of Technology in 2015. He received B.Sc. and M.Sc. in Automatic Control and Robotics from the same university in 2014 and 2015, respectively. Since 2015 he is a Ph.D. student at the Faculty of Electrical Engineering. In 2016 he became a research assistant at the Institute of Control, Robotics, and Information Engineering. He is author or coauthor of 13 technical papers in the area of robotics and machine learning. His current research interests include robotic global localization, machine learning, and simultaneous localization and mapping.

Piotr Skrzypczyński graduated from the Poznan University of Technology (PUT) in 1993. He received Ph.D. and D.Sc. degrees in Robotics from the same university in 1997 and 2007, respectively. Since 2010, he is an associate professor at the Institute of Control, Robotics, and Information Engineering (ICRIE) of PUT, and since 2012 head of the Automation and Robotics Division at ICRIE. He leads the Mobile Robotics Laboratory at ICIE. Prof. Skrzypczyński is author or coauthor of more than 160 technical papers in the areas of robotics and computer science. His current research interests include autonomous mobile robots, navigation, localization and mapping, multisensor fusion, and computational intelligence in robotics.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2022.3150841

Stereo Plane R-CNN: Accurate scene geometry reconstruction using planar segments and camera-agnostic representation

Jan Wietrzykowski[1] and Dominik Belter[1]

***Abstract*—The article introduces a novel method for planar segments detection and description from a stereo pair of images. The existing systems for planes detection utilize single RGB images and have accuracy- and scale-related problems regarding 3D reconstruction with the obtained planar segments. The proposed approach draws inspiration from deep-learning-based systems for plane detection and depth reconstruction. Firstly, we improve the planes detection in the image. Secondly, we enhance geometry reconstruction accuracy using a stereo setup. To achieve the 3D model of the observed planes, we introduce a novel neural network architecture and training strategy that jointly optimizes the prediction of disparity, normal vectors, and plane parameters. Moreover, the proposed approach utilizes an efficient camera-agnostic representation of the problem. Finally, we show that our system outperforms existing approaches to planar segments detection and parameters estimation and improves the reconstruction accuracy of indoor environments.**

***Index Terms*—Deep Learning for Visual Perception; Mapping; Semantic Scene Understanding**

I. Introduction

A KEY factor in introducing camera-based localization systems to everyday life is their robustness. One way to improve the robustness is to include a relocalization mechanism that uses higher-abstraction-level objects as matched features [1], [2]. Viable alternatives to point features are planar segments because they can be reliably detected and are common in man-made environments [2]. However, precise 3D pose estimation of those segments is crucial, not only for camera localization [3] but also for scene reconstruction [4]. Unfortunately, geometry reconstruction of planar segments using a monocular camera is a difficult task [2], [5] due to problems with metric scale estimation and ambiguity in the orientation of planes. A solution can be to use RGB-D sensors that became widely used in the last years. However, their effective range is limited, which leads to discarding a lot of beneficial information about distant regions of the scene [1]. Promising alternatives are stereo cameras that have a larger effective range than the available RGB-D cameras and enable more accurate geometry reconstruction than a monocular camera [6]. However, most of the stereo-based reconstruction research focuses on dense depth estimation from a pair of images [6], [7] and neglects the role of position and orientation of higher-level planar features, useful in the localization and scene reconstruction.

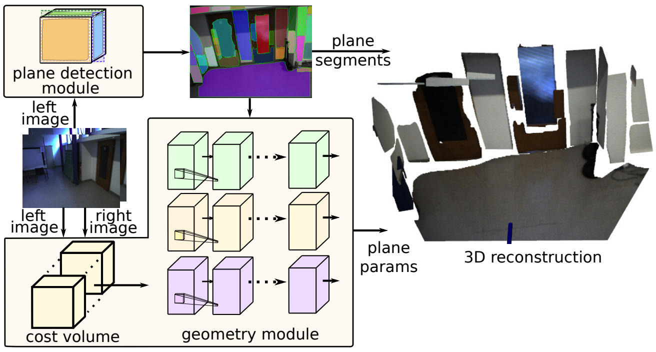

Fig. 1. Architecture of the proposed system. The plane detection module detects planes on a single RGB image, and a novel geometry module utilizes stereo pair of images to estimate the 3D poses of the detected planes.

Almost all state-of-the-art research on planar segments detection is focused on the application of Deep Neural Networks (DNNs) to images from monocular cameras [2], [3], [4]. This group includes the Plane R-CNN [5] that uses an architecture based on Mask R-CNN [8] and achieves state-of-the-art detection performance. However, our tests indicate that the performance of this method on a different dataset drops significantly, especially for distant planes. Also, geometry reconstruction is still unsatisfactory due to problems stemming from using a monocular camera, namely, inaccurate estimates of distances to planes and normal vectors.

Considering the limitations of the existing methods, we propose a new DNN-based system that detects planes and utilizes depth information encoded in a stereo pair of images to estimate 3D plane parameters. We bridge a gap between RGB-based plane detectors and systems that densely estimate depth from stereo pairs of images to obtain an accurate set of planar segments describing the scene (Fig. 1). Our contribution can be summarized as follows[1]:

- An improved plane segmentation method that deals with the problem of suppressed plane segments in the detection

Manuscript received: September 24, 2021; Revised January 12, 2022; Accepted February 3, 2022.

This paper was recommended for publication by Editor Cesar Cadena upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the National Science Centre (NCN), contract no. UMO-2018/31/N/ST6/00941.

[1] Jan Wietrzykowski and Dominik Belter are with Institute of Robotics and Machine Intelligence, Poznan University of Technology, 60-965 Poznań, Poland name.surname@put.poznan.pl

Digital Object Identifier (DOI): see top of this page.

[1] Implementation and dataset are available at https://github.com/LRMPUT/sprcnn

methods based on Regions Of Interest (ROIs) and Non-Maximum Suppression (NMS).

- A novel neural network architecture that leverages disparity information from a stereo camera to accurately reconstruct scene geometry.
- A camera-agnostic normal vector representation that improves the robustness of the neural network to changes of the camera parameters that naturally arise when deploying a system in a real-life scenario.
- A training procedure that simultaneously utilizes global parameters of planes, pixel-wise normal vectors, and disparity prediction to improve the accuracy of plane parameters estimation.
- A fully automatically generated photorealistic synthetic dataset containing stereo images annotated with planar segments.

II. Related work

A. Pixel-wise depth and normal estimation

Research on scene geometry estimation focuses on recovering pixel-wise depth information from a single image [9], [10]. However, the work by Smolyanskiy *et al.* [6] argues that a 3D precise geometric reconstruction requires a stereo camera. They also propose a semi-supervised method for learning depth prediction. The ground truth data utilizes 3D laser scanner measurements and is augmented by unsupervised photoconsistency evaluation between stereo images. Convolutional neural networks (CNNs) have also been proven to be efficient in estimating normal vectors for each pixel of an RGB image [11]. However, recent work suggests that a coupled estimation of normal vectors and depth values provides more consistent and accurate estimates [12]. Also, Kusupati *et al.* [13] show that joint learning depth and normal vectors and enforcing consistency give significantly better results than separate learning. A generalized approach to consistency learning demonstrated on normal and depth estimation is presented in [14]. Unfortunately, knowledge about depth and surface normals could be only supplementary information for the generation of geometric features. Nonetheless, in this article, we follow the joint learning approach and optimize simultaneously losses related to disparity, pixel-wise normal vectors, and plane parameters estimation to provide more accurate results.

B. Detection of planar segments

Planar segments are a promising alternative to pixel-wise geometry reconstruction. Indoor environments are rich in planar segments and can be described just by a few of these geometric primitives. Another advantage is that the geometric properties of the underlying infinite planes can be easily described by a linear equation with only 4 parameters. Our initial experiments [15] prove that planes are also suitable for the global localization methods. An end-to-end approach to recover 3D planes from a single image is presented in [16], where supervision of learning is done indirectly by using depth ground truth. The parameters of the detected planes are estimated from values extracted from the latent space of the neural network. The limitation of their method is that only five planar segments can be detected in the scene, and learning requires a complete depth map for every training image. A limited number of planar segments can be also processed by the PlaneNet method [17]. The architecture of the neural network proposed in [17] contains separate branches for plane segmentation and for estimation of plane parameters. In our research, we also utilize a separate branch for estimation of plane parameters, but, at the same time, we train dense branches for disparity and normal vector estimation and show that this approach provides more accurate results. The SVPNet method [18] focuses on the binary classification of pixels into planar and non-planar segments and the extraction of embeddings where the same plane instances are close to each other. The planar segments are extracted using the mean-shift algorithm, and the plane parameters are estimated for each pixel in the first processing stage. In [5] the proposed Plane R-CNN method detects planar regions and reconstructs a piecewise planar depth map from a single RGB image. The plane instances are detected using a network based on the Mask R-CNN [8]. Then, a segmentation refinement network improves the consistency of the detected planar segments. The depth image is estimated directly from the RGB image by the decoder connected to the feature pyramid network (FPN) [19]. Estimation of the normal vectors consists of two components. The classifier picks one of the seven anchor normal directions and separately estimates the residual 3D vector. We, however, use a direct normal estimation because it provides better results.

C. 3D reconstruction and applications

Detected planar segments can be used in further scene reconstruction. Park and Yoon [20] show that stereo matching and disparity estimation can be improved by plane hypotheses generation and global optimization with plane hypotheses. In [21], information about planar segments from a single omnidirectional camera image is used to design plane-aware loss that improves normal vectors' predictions accuracy. The Plane R-CNN has been already used to detect and reconstruct planes that are occluded by other objects on a single camera view [22]. Ye *et al.* [2] added a plane description network which is later used to match detected planes between images and estimate the motion of the camera. Also, Xi and Chen [3] show that multi-view regularization of planar segments improves the reconstruction of indoor scenes. Another approach is presented in [4] where the neural networks predict if planes are orthogonal, parallel and if two planes are in contact. In our method, the stereo pair of images is used instead of multiple views and the regularization of two views is embedded in a new DNN architecture designed by us to recover 3D geometry. We focus on the accurate estimation of 3D geometry because the correct poses of planes are crucial for camera localization [2] and scene reconstruction [21].

III. Detection and reconstruction of planar segments

The proposed network consists of two main components: a plane detection module and a geometry module (Fig. 1).

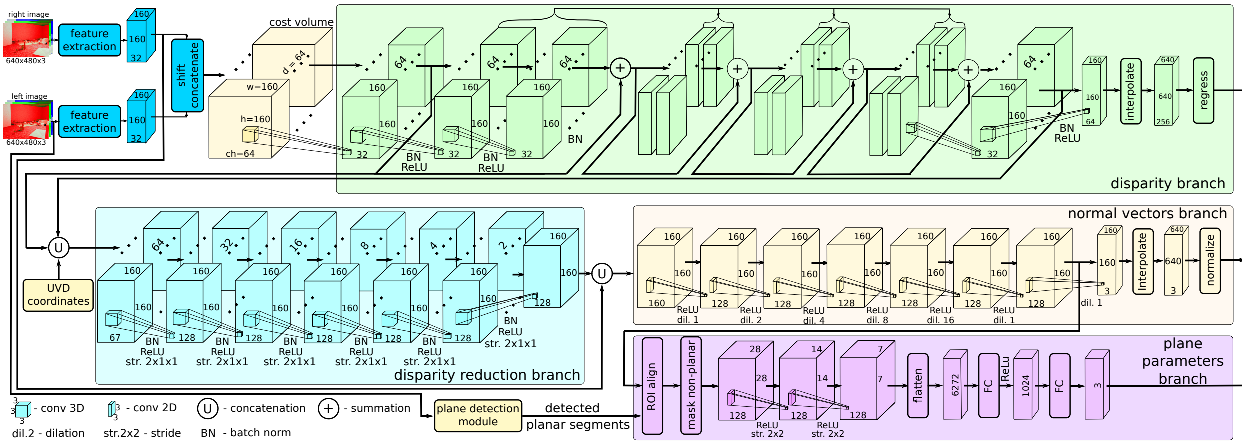

Fig. 2. Architecture of the geometry module in the Stereo Plane R-CNN. Each 3D convolution is 3×3×3 and 2D convolution is 3×3

The plane detection module is inspired by the Plane R-CNN [5] architecture that detects planar segments on a monocular image (the left one in our system). We improved the detection quality of planar segments by applying ROI-aware segmentation during training and by learning on a properly labeled dataset. The geometry module is inspired by the work of Kusupati *et al.* [13] that exploits stereo setup to infer about the geometry of the scene. This module builds a cost volume for a 3D space observed by the sensor and processes neural network embeddings to estimate pixel-wise disparities, normal vectors, and plane parameters for the detected segments. We jointly minimize losses related to all estimated values and use a camera-agnostic normal representation to improve geometry reconstruction performance.

A. Geometry module architecture

Although 3D coordinates of points computed from estimated disparity do not guarantee a precise fitting of a plane model, the geometry module (Fig. 2) utilizes features produced during disparity estimation to estimate normals and plane parameters. The module is based on a cost volume created in a proposed $\widetilde{UVD}$ space (explained in Sec. III-C), where features from the left and right image are concatenated for every point in that space. A disparity branch (green block in Fig. 2) is based on the Pyramid Stereo Matching Network [7] that uses 3D convolutions to process concatenated features and produce probability distributions of disparities for each pixel. Expected values are computed from those distributions to regress final disparity values. Features from the beginning and the end of the disparity estimation branch are concatenated with $\widetilde{UVD}$ coordinates and used in a disparity reduction branch (light blue). Using 3D convolutions with stride 2 in the disparity dimension, that halves this dimension's size after each operation, we reduce this dimension of the feature maps to 1. This step effectively removes the disparity dimension, leaving rich 2D normal features. The 2D normal features are concatenated with visual features from the left image and processed using 2D convolutions with various dilations to return three parameters of normals for every pixel. The visual features help to smooth the estimates by providing visual cues about the surface. Moreover, 2D features are also used in a plane parameters subbranch (purple in Fig. 2) that samples them using ROI align according to detected ROIs. Sampled features that do not belong to the segment but are inside the ROI are masked by zeroing their values and such feature map is processed using two convolutional and two fully connected layers to estimate a plane normal.

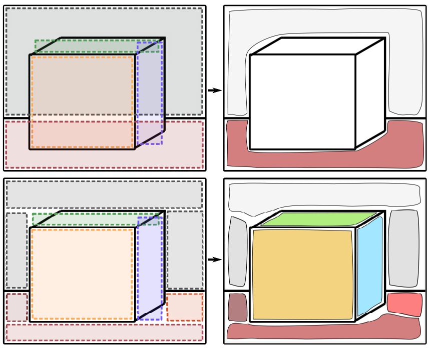

Fig. 3. ROI-aware segmentation: NMS removes some detections for complex scenes where bounding boxes overlap (top). Oversegmentation of the complex shapes (bottom) produces a larger number of smaller detections, but preserves planes that are crucial for scene reconstruction

B. ROI-aware detection and segmentation

We have observed that Plane R-CNN has problems with planar segments occupying a large area of the image, especially the ones also interleaving with other segments (illustrated in Fig. 3). We noticed that it was due to ROI boxes containing multiple segments. Boxes for different segments are overlapping with each other and got suppressed in the Non-Maximum Suppression (NMS) step. The same problem exists for prolonged segments and in general for all segments whose shape is far from square. Therefore, we propose to divide target segments during training into smaller ones with more square-like shapes. It is worth noting that segmentation into planar segments is often arbitrary and can be done in many correct ways. For example, the front of the cupboard can be segmented as one segment or as separate segments for each door. Both segmentations are correct regarding the plane-based localization or navigation [1]. We employed a simple algorithm we call ROI-aware segmentation based on flood fill, that can be executed on the fly when loading training

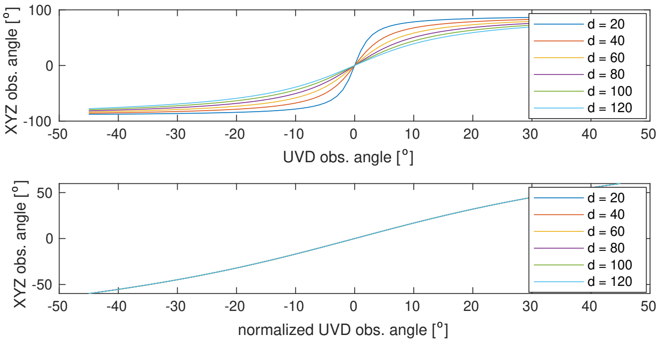

Fig. 4. Angle of observation of a normal in the XYZ space as a function of angle of observation in the UVD space (top) and $\widetilde{\text{UVD}}$ space (bottom) for different disparities, $f_x = f_y = 550$, $c_x = o_x = 320$, $c_y = o_y = 240$, baseline $b = 0.2$, and scale constant $a = 320$. Note values on vertical axes. For $\widetilde{\text{UVD}}$ all lines overlap.

samples. The algorithm starts at a random pixel and floods the segment as long as the ratio of the area of the grown region to the area of the smallest bounding box is above 0.5. If the ratio is below this threshold, the bounding box is much larger than the segment itself and is mostly empty. This implies that the segment's shape is deviating from being square-like and that the ROI of this segment can be overlapping with ROIs of the neighboring segments. Although it is a greedy algorithm that does not guarantee optimal segmentation, we found it works well in practice with an acceptable level of over-segmentation. Therefore, instead of a refining module proposed in [5], we use carefully segmented target masks during learning to obtain good quality detections when testing. Note that it is not necessary to use this mechanism during the inference because a neural network has already learned to produce proposals that are ROI-aware.

C. Scene geometry from stereo camera

A mapping between 3D coordinates of points or normal vectors and pixel coordinates relies heavily on the camera intrinsic parameters. If a black box model (e.g., neural network) is applied to the estimation of 3D coordinates from an image it has to capture this relation. Thus, we make the normal representation camera-agnostic to simplify this problem and to avoid unnecessary transformations that DNN would have to learn. If an input to the DNN is a pair of stereo images, data structures containing those images are organized according to image coordinates (u, v) and disparities (d). Hence, u, v, and d are known to the network for every processed point. What the network does not know, are XYZ coordinates of points because camera parameters are necessary to compute them. Therefore, instead of performing estimation in XYZ space associated with a camera frame and physical dimensions, we exploit disparity-normalized UVD ($\widetilde{\text{UVD}}$) space associated with pixel coordinates and disparity. This mitigates problems with deployment in real-life scenarios that stem from differences between available training datasets and target hardware. The transformation between XYZ space and $\widetilde{\text{UVD}}$ space is linear, so planes remain planes under this transformation. To derive this transformation, let us consider equations of a 3D world point (x, y, z) projection for a calibrated stereo camera with a baseline b:

$$\begin{cases} u = \dfrac{f_x x}{z} + c_x - o_x \\ v = \dfrac{f_y y}{z} + c_y - o_y \\ d = \dfrac{f_x b}{z}, \end{cases} \tag{1}$$

where $(x, y, z)^T$ is a 3D position of a point in a camera frame, f_x, f_y, c_x, c_y are intrinsic camera parameters, and o_x, o_y is an origin of the UVD coordinate frame, which can be chosen arbitrarily to move it from the upper left corner of the image. Transformation of point $\mathbf{p} = \begin{pmatrix} x, & y, & z, & 1\end{pmatrix}^T$ from XYZ to a point $\mathbf{p}_D$ in UVD using homogeneous coordinates can be written as:

$$\mathbf{p}_D = \mathbf{G}_{D,C}\mathbf{p}, \tag{2}$$

where $\mathbf{G}_{D,C}$ is a matrix following (1). Therefore, plane parameters in UVD $\boldsymbol{\pi}_D = \begin{pmatrix} n_u, & n_v, & n_d, & -r_D\end{pmatrix}^T$ that satisfy $\boldsymbol{\pi}_D \cdot \mathbf{p}_D = 0$ can be transformed to XYZ using:

$$\boldsymbol{\pi} = \mathbf{G}_{D,C}^T \boldsymbol{\pi}_D = \mathbf{G}_{C,D}^{-T} \boldsymbol{\pi}_D, \tag{3}$$

derived using (2), where $\boldsymbol{\pi} = \begin{pmatrix} n_x, & n_y, & n_z, & -r\end{pmatrix}^T$. This transformation is also linear, however has an undesired property that for small disparities (distant objects), a relatively small angular error in normal estimation in UVD propagates as a large error in XYZ. To illustrate this, consider a plane observed at different horizontal angles (rotation around the Y-axis of the camera) in front of the camera. In the top part of Fig. 4 an angle of observation in the XYZ space was plotted as a function of an angle of observation in the UVD space for example camera parameters. It is visible that the smaller the disparity, the sharper the transition and thus the larger the derivative, which is a multiplicative factor in the error propagation. To overcome this problem, we normalize the coordinates in the UVD space with the disparity:

$$\begin{cases} \tilde{u} = \dfrac{u}{d} = \dfrac{f_x}{f_x b} x + \dfrac{c_x - o_x}{f_x b} z \\ \tilde{v} = \dfrac{v}{d} = \dfrac{f_y}{f_x b} y + \dfrac{c_y - o_y}{f_x b} z \\ \tilde{d} = \dfrac{a}{d} = \dfrac{a}{f_x b} z, \end{cases} \tag{4}$$

where a (320 in the experiments) is an arbitrary constant assuring uniform scaling of the space and forcing values in $\widetilde{\text{UVD}}$ to be of the same magnitude. By virtue of this normalization, and using values of o_x, o_y close to c_x, c_y (usually optical centers of cameras do not vary much and are close to image center), relation of observation angles in XYZ and $\widetilde{\text{UVD}}$ is approximately linear and does not depend on d (see bottom part of Fig. 4). Using homogeneous coordinates it can be written as:

$$\mathbf{p}_{\tilde{D}} = \begin{pmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{d} \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{f_x}{f_x b} & 0 & \frac{c_x - o_x}{f_x b} & 0 \\ 0 & \frac{f_y}{f_x b} & \frac{c_y - o_y}{f_x b} & 0 \\ 0 & 0 & \frac{a}{f_x b} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \mathbf{G}_{\tilde{D},C}\mathbf{p}. \tag{5}$$

This space is camera-agnostic, meaning that to calculate a plane equation in this space it is sufficient to know image coordinates and disparities of points forming this plane, without knowledge about focal lengths f_x, f_y, optical center c_x, c_y, and baseline b. Moreover, the space is scaled similarly to the XYZ space and therefore angular errors are not significantly magnified. To transform plane $\boldsymbol{\pi}_{\tilde{D}} = \left(n_{\tilde{u}}, \quad n_{\tilde{v}}, \quad n_{\tilde{d}}, \quad -r_{\tilde{D}}\right)^T$ from $\widetilde{\text{UVD}}$ space to XYZ space we use equation analogous to (3):

$$\boldsymbol{\pi} = \mathbf{G}_{C,\tilde{D}}^{-T}\boldsymbol{\pi}_{\tilde{D}}. \tag{6}$$

In the plane parameters branch of the DNN, we estimate only the normal vector of the segment. To estimate the distance to the origin r we use RANSAC and disparity estimates from the disparity branch. In the procedure, we seek the best set of inliers using RANSAC and a threshold on the relative distance $\frac{\mathbf{n}_h \cdot \operatorname{proj}(\mathbf{p})}{r_h} < 0.05$, where $\operatorname{proj}(\mathbf{p})$ is a 3D XYZ point expressed in inhomogeneous coordinates, $\mathbf{n}_h$ is a normal vector of the RANSAC hypothesis, and r_h is a distance to the origin of the RANSAC hypothesis. Finally, r is estimated using all inliers from the best hypothesis, leaving the DNN estimated normal unchanged (RANSAC estimated normal is ignored).

During detection, we use only two classes (planar and non-planar). We found that using anchors for normal directions and dividing planes into classes related to those directions, as in [5], does not improve normal estimation accuracy comparing to direct estimation of 3 normal parameters.

IV. Training

A. Dataset

To the best of our knowledge, there is no large real-world dataset with stereo images and ground truth depth information for the indoor environment. Moreover, the quality of depth information and created mesh models in existing monocular datasets, such as `ScanNet` [23], are insufficient. We examined the labeling of `ScanNet` used by Plane R-CNN [5], which turned out to be of poor quality due to noisy and inaccurate mesh models produced using an RGB-D sensor. Thus, we generated a synthetic dataset called `SceneNet Stereo` to train the neural network. To generate scenes and render photorealistic images, we exploit a method from the `SceneNet RGB-D` dataset [24] by adapting it to produce stereo images. As a result of having perfect information about the geometry of rendered scenes, the training set was very accurate, which is difficult to obtain on real-world images. We generated 200 random scenes with 300 images for each scene. Finally, we selected approximately 35k training images and 2k testing images.

B. Loss and parameters

TABLE I
STATISTICS ON DETECTED PLANES FOR TESTING DATASETS

	bin no.	1	2	3	4	5	6
	A [px]	0-50	50-100	100-150	150-200	200-250	250-640
`SceneNet Stereo`	no. of planes	19258	10283	3025	1612	1329	2450
	$\overline{\text{area}}$ [%]	3.9	7.8	7.1	7.6	10.5	39.7
`TERRINet`	no. of planes	138739	70501	16052	4670	3608	4286
	$\overline{\text{area}}$ [%]	6.4	12.2	8.3	5.0	6.6	13.8

We use weights pre-trained on the `Coco` dataset for the detection module and weights pre-trained on the `ScanNet` [13] dataset for feature extraction layers and disparity branch of the geometry module. We train the whole model simultaneously, using different loss functions for specific branches. We use a loss from [5], without plane parameters component, to supervise the detection module and denote it as $\mathcal{L}_r$. The disparity estimation is supervised using L_1 smooth loss for all pixels $\mathcal{P}$ that have a valid target depth:

$$\mathcal{L}_d = \frac{1}{|\mathcal{P}|}\sum_{p\in\mathcal{P}} f_1(d_p, d_p^*), \tag{7}$$

where f_1 is a smooth L_1 difference function, d_p is a disparity for pixel p, and d_p^* is a target disparity for pixel p. Because we are interested only in pixels belonging to planar segments, we exclude pixels near edges of objects during computation of pixel-wise normal vector loss $\mathcal{L}_n$:

$$\mathcal{L}_n = \frac{1}{|\mathcal{P}|}\sum_{p\in\mathcal{P}} f_1(\mathbf{n}_p, \mathbf{n}_p^*), \tag{8}$$

where $\mathbf{n}_p$ and $\mathbf{n}_p^*$ are an estimated and a reference normal vector, respectively. The plane parameters loss is computed for all detections $\mathcal{D}$ returned by the detection module:

$$\mathcal{L}_p = \frac{1}{|\mathcal{D}|}\sum_{d\in\mathcal{D}} f_1(\mathbf{n}_d, \mathbf{n}_d^*). \tag{9}$$

The final loss is a sum of weighted losses $\mathcal{L}_r$ and (7)–(9):

$$\mathcal{L} = \mathcal{L}_r + w_d\mathcal{L}_d + w_n\mathcal{L}_n + w_p\mathcal{L}_p, \tag{10}$$

where $w_d = 1$, $w_n = 100$, and $w_p = 100$ to accommodate for different scales of values. For a fair comparison, we use the same weights during training of the baseline Plane R-CNN model (note that it is different from the original setup because of the different value of w_p). The training takes 10 epochs using Adam optimizer with a learning rate equal to 10^{-5} and weight decay equal to 10^{-4}. We augment training examples using random color and sharpness manipulation, Gaussian noise, and random cropping. For the baseline solution (Plane R-CNN), we skip augmentation as it worsens results.

V. Experimental verification

We use three metrics that measure geometric aspects of segmentation that are important during localization [1]:

- Detection Error (DE) - measures how planar is the area labeled as one segment. It is computed as RMS of point-to-plane distance in 3D between points belonging to the segment and plane fit into those points using RANSAC. It also measures the quality of segmentation, like metrics evaluating the similarity of pixels clustering, while avoiding problems with the ambiguity of correct segmentation.
- Depth Reconstruction Error (DRE) - measures RMS of depth differences between 3D points belonging to the segment and depthmap induced by a plane estimated by the DNN. We use only points classified as inliers by RANSAC to accommodate for imperfect segmentation

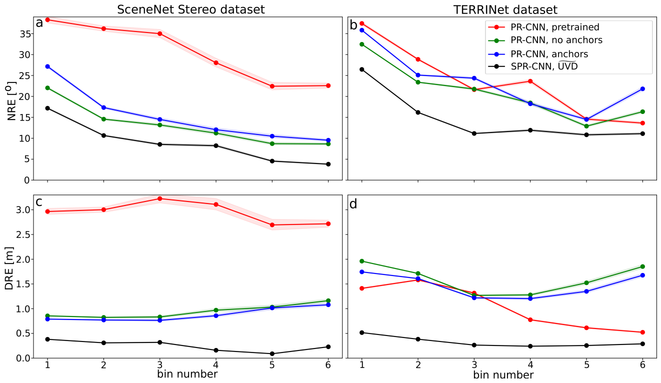

Fig. 5. Dependence between the NRE (a,b) and DRE (c,d) for the SceneNet Stereo (a,c) and TERRINet (b,d) datasets. Confidence intervals are marked with light-colored regions. Relations between the bin numbers and planar segments sizes are presented in Tab. I

and incorrect ground truth depth at the edges. We also crop plane induced depth to 15 m as we do not consider objects that are further away.

- Normal Reconstruction Error (NRE) - measures RMS of differences between normals found using RANSAC and ones estimated by the DNN.

In all metrics, we use ground truth depth information to precisely estimate plane equations. Robust estimation mitigates situations when segment masks spill over edges of surfaces or depth is incorrect at the edges.

Two different datasets were used during the evaluation to show various aspects of planar segments detection and geometry reconstruction. The first one is a testing part of the synthetic SceneNet Stereo dataset with nine different scenes and approximately 2k images. Note that the number of planes used to evaluate the accuracy of the methods is significantly higher. The second one is a real-world TERRINet dataset gathered in office and laboratory environments[2]. It contains several indoor sequences of stereo images, Velodyne VLP-16 lidar scans, and ground truth poses from a Qualisys motion capture system. By using ground truth poses and lidar measurements we built a precise point cloud representation of the scenes. Then, the point cloud was used to compute a depth map for every stereo image pair. We used approximately 8.5k images from 3 different environment settings.

To give more insight into the geometry reconstruction of various planar segments, we present results as a function of the segment's area expressed in pixels. We divide segments into six bins, depending on the square root of their area, denoted as A, which can be intuitively compared to the area of a square with a side length equal to A. Statistics regarding bins for used datasets are presented in Tab. I, where $\overline{\text{area}}$ denotes mean area per image.

A. Geometry reconstruction using stereo

The main experiment shows that our system, trained on a photorealistic synthetic dataset, can be used in a real-world scenario and has a superior performance over the baseline Plane R-CNN solution in terms of geometry reconstruction. The Plane R-CNN yields the best results in the literature with other systems only presenting functionalities added on the top of the Plane R-CNN [4], [22] or being closed-source [3]. Those functionalities can be also added on top of Stereo Plane R-CNN if necessary, but would obfuscate the results. The methods used for comparison are shortly characterized below:

- *Plane R-CNN (PR-CNN), pre-trained* - baseline version presented in [5] with anchors for normal estimation and refinement module, trained by the authors on the ScanNet dataset, using left image only.
- *PR-CNN, no anchors* - baseline version without anchors and the refinement module, trained on the SceneNet Stereo dataset, using left image only.
- *PR-CNN, anchors* - baseline version with anchors but without the refinement module, trained on the SceneNet Stereo dataset, using left image only.
- *RANSAC, SGBM depth* - the method that uses non-learned Semi-Global Block Matching stereo depth estimation and performs classic plane fitting using RANSAC.
- *RANSAC, DNN depth* - the method that uses learned stereo depth estimation architecture from [7] trained on our dataset and performs RANSAC plane fitting.
- *Stereo Plane R-CNN (SPR-CNN), $\widetilde{UVD}$* - the proposed method described in Sec. III.

The results of the experiment are presented in Tab. II. To eliminate the influence of segment detection on the results, we used the same detections for all tests. Detections were generated and saved by one version of the system and are loaded in all test cases, except the pre-trained Plane R-CNN due to the presence of the refinement module. Table II presents qualitative results, while detailed results on TERRINet dataset are visible in Fig. 5, where performance as a function of the square root of the area A is presented. Both learned stereo versions perform significantly better in terms of depth errors, which suggests that it is crucial to precise geometry reconstruction. However, the classic approach to stereo depth estimation does not provide enough valid points to precisely fit a plane. As for normal errors, Stereo Plane R-CNN outperforms other systems by a large margin. Results also indicate that using anchors does not improve normal estimation accuracy which is why we do not use this technique. Please also note that the pre-trained version of Plane R-CNN detects less distant segments (mean distance to points is 3.39 m, compared to 4.21 m for detections used for other versions), which explains better results of the pre-trained version compared to the version trained by us when testing on the TERRINet dataset. Figure 7 shows visualizations of example scenes for the baseline Plane R-CNN without ROI-aware segmentation and Stereo Plane R-CNN.

B. Detection using ROI-aware segmentation

This experiment aims at showing the effects of the proposed adaptation to the ROI-based processing. We use the DE score, which is designed to measure the quality of detection, to show the differences between methods. We employ the synthetic dataset only because it has precise depth information for all

[2] Dataset collection was supported by the TERRINet project funded by EU H2020 under GA No.730994

TABLE II
GEOMETRIC ACCURACY (DRE AND NRE) FOR BOTH DATASETS

	SceneNet Stereo		TERRINet	
	DRE [m]	NRE [°]	DRE [m]	NRE [°]
PR-CNN, pretrained [5]	2.82	25.64	1.00	20.75
PR-CNN, no anchors	1.05	11.13	1.66	21.34
PR-CNN, anchors	0.98	12.81	1.52	24.12
RANSAC, SGBM depth	0.44	11.28	2.17	30.84
RANSAC, DNN depth	**0.22**	10.13	0.38	22.88
SPR-CNN, $\widetilde{\text{UVD}}$ (full arch.)	0.24	**7.09**	**0.34**	**15.07**

TABLE III
DETECTION ERRORS FOR THE SceneNet Stereo DATASET

bin no.		1	2	3	4	5	6	all
SPR-CNN w/o ROI-aware	DE [m]	0.180	0.206	0.185	0.187	0.152	**0.118**	0.147
	$\overline{\text{area}}$ [%]	1.3	5.5	6.3	8.0	11.5	43.8	76.5
SPR-CNN w. ROI-aware	DE [m]	**0.148**	**0.127**	**0.123**	**0.166**	**0.102**	0.136	**0.134**
	$\overline{\text{area}}$ [%]	4.1	9.4	8.5	8.3	10.6	39.0	79.8

pixels. Quantitative results are presented in Tab. III, where despite a larger area of detected segments, DNN trained with ROI-aware segmentation performs significantly better. However, the most notable differences can be seen in Fig. 7, where visual comparison is presented.

C. Ablation study

To justify our design choices we conduct an ablation study comparing different versions of Stereo Plane R-CNN:

- *SPR-CNN normal vec. only* - version without plane parameters branch, plane normals are estimated by averaging pixel-wise values from the normal vector branch, using $\widetilde{\text{UVD}}$.
- *SPR-CNN, plane param. only* - version without the supervision of pixel-wise normals in the normal vector branch, using $\widetilde{\text{UVD}}$.
- *SPR-CNN, XYZ (full arch.)* - estimates normals in the XYZ space, instead of the $\widetilde{\text{UVD}}$ space.
- *SPR-CNN, UVD (full arch.)* - estimates normals in the UVD space, instead of the $\widetilde{\text{UVD}}$ space.
- *SPR-CNN, $\widetilde{\text{UVD}}$ (full arch.)* - proposed method.

Results are presented in Tab. IV and suggest that having a specialized branch for plane parameters estimation boosts performance significantly. However, supervision of normals at the level of pixels and $\widetilde{\text{UVD}}$ representation also contribute to the final result notably. Additionally, it is clearly visible that regular UVD space (without normalization) is not suitable for normal estimation as it yields the worst results as far as NRE is concerned.

D. Robustness to camera parameters change

The goal of the last experiment is to show that the proposed camera-agnostic representation performs well when camera parameters change. Because there is no real-world dataset that contains images from different cameras with different parameters, we use the synthetic SceneNet Stereo dataset in this experiment. Test sequences were rendered once more with fixed lighting and with different camera parameters. We were changing diagonal field of view (FoV, change of, both, f_x and f_y), vertical FoV (change of f_y), horizontal FoV (change of f_x), and baseline (b). We do not consider different c_x and c_y values, without a loss of generality, because their change only shifts image left/right and up/down. In this experiment, we compare monocular Plane R-CNN that estimates normals in XYZ, Stereo Plane R-CNN that estimates normals in XYZ, and the proposed method that estimates normals in $\widetilde{\text{UVD}}$. The results are summarized in Fig. 6. The increase of NRE for the version exploiting camera-agnostic representation is lower than for the version using XYZ representation, which supports the thesis that such a representation is beneficial to assure robustness to camera parameters change. However, despite using camera-agnostic representation, the whole model is not completely camera-agnostic because of changing incidence relations when f_x, f_y, or b change. The model seems to be more sensitive to f_x change (change of diagonal and horizontal FoV) than to f_y and b change (change of vertical FoV and baseline). It is worth noting that changing the diagonal FoV in the monocular system slightly lowers the normal estimation error. This phenomenon comes from the fact that changing, both, f_x and f_y only scales the image. Moreover, with the wider camera field of view, a broader context is captured and the normal estimation error decreases. Nonetheless, results for the monocular system are still worse than for the stereo ones. In terms of DRE, the performance of both stereo versions slightly deteriorates, which can be again attributed to changing incidence relations. However, changing camera parameters introduces the scale change and increases significantly the DRE value for the monocular version when the diagonal FoV changes.

TABLE IV
ABLATION STUDY OF DIFFERENT VERSIONS OF STEREO PLANE R-CNN

	SceneNet Stereo		TERRINet	
	DRE [m]	NRE [°]	DRE [m]	NRE [°]
SPR-CNN, normal vec. only	0.29	8.75	0.38	18.77
SPR-CNN, plane param. only	0.28	8.28	0.37	16.09
SPR-CNN, XYZ (full arch.)	0.36	7.21	0.71	16.07
SPR-CNN, UVD (full arch.)	0.25	10.41	0.48	21.30
SPR-CNN, $\widetilde{\text{UVD}}$ (full arch.)	**0.24**	**7.09**	**0.34**	**15.07**

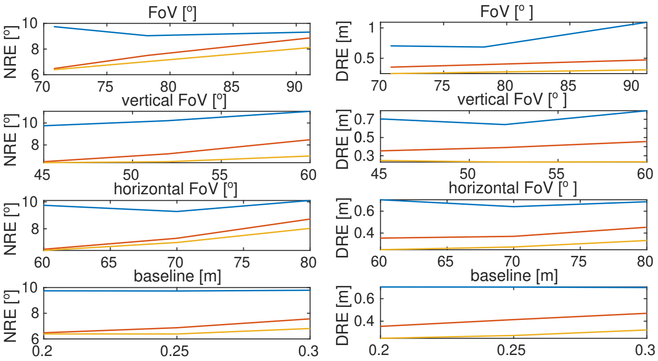

Fig. 6. Dependence between NRE (left) DRE (right) and camera parameters change for the PR-CNN (blue), SPR-CNN XYZ (red), and SPR-CNN, $\widetilde{\text{UVD}}$ (orange). The most left values on horizontal axes are used during learning.

VI. CONCLUSIONS

In this article, we propose the Stereo Plane R-CNN method that detects and computes the parameters of planar segments from stereo pairs of images. The system is trained on the

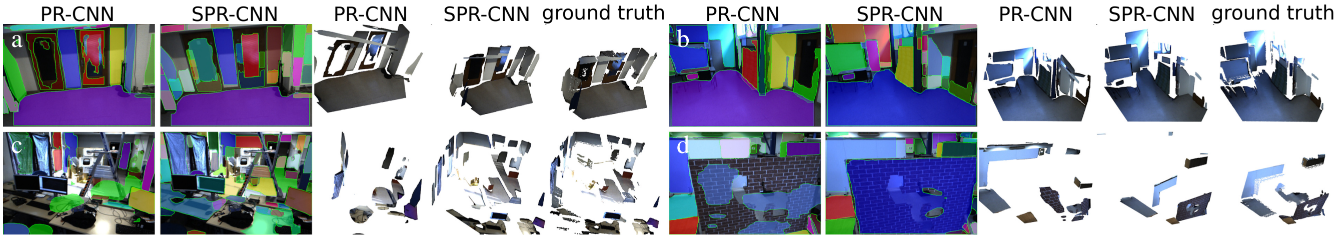

Fig. 7. Comparison of segment detection and scene geometry reconstruction performance on the four scenes (a,b,c,d) from the `TERRINet` dataset. Note scale problems of the monocular version for the first example (a).

synthetic dataset that provides accurate information about the depth of the scene, segmentation, and plane parameters. The system is verified on the dataset obtained in the indoor unstructured and challenging environment. The proposed method is compared to other the state-of-the-art methods. Moreover, we provide ablation studies on the contributions of main components in our system to justify our design choices and to show the performance of the proposed method. The results presented in Tab. II show that the proposed problem representation and neural network architecture outperform other approaches. The mean inference time for our solution is 0.419 s on RTX 3090 with batch size 1, which is approximately twice as much as Plane R-CNN and is caused by a larger amount of computations needed to process the cost volume. However, the obtained computation time is sufficient for the global localization task [1].

In particular, we show that improved image segmentation deals with the suppression problems of methods based on ROIs and NMS (results in Tab. III). We also propose a novel neural network architecture that leverages disparity information from a stereo camera to precisely reconstruct scene geometry (Fig. 7). The neural network uses the proposed camera-agnostic normal representation $\widetilde{\mathrm{UVD}}$ that improves robustness to camera parameters change (Fig. 6). Finally, we propose a training procedure that simultaneously utilizes parameters of planes, pixel-wise normal vectors, and disparity prediction to improve the accuracy of reconstruction (results in Tab. II and Fig. 5). In the future, we are going to integrate the Stereo Plane R-CNN with our global relocalization system [1] to improve localization in the indoor environment.

References

[1] J. Wietrzykowski and P. Skrzypczyński, "PlaneLoc: Probabilistic global localization in 3-D using local planar features," *Robotics and Autonomous Systems*, vol. 113, pp. 160–173, 2019.

[2] W. Ye, H. Li, T. Zhang, X. Zhou, H. Bao, and G. Zhang, "SuperPlane: 3D plane detection and description from a single image," in *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, 2021, pp. 207–215.

[3] W. Xi and X. Chen, "Reconstructing piecewise planar scenes with multi-view regularization," *Computational Visual Media*, vol. 5, p. 337–345, 2019.

[4] Y. Qian and Y. Furukawa, "Learning pairwise inter-plane relations for piecewise planar reconstruction," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 330–345.

[6] N. Smolyanskiy, A. Kamenev, and S. Birchfield, "On the importance of stereo for accurate depth estimation: An efficient semi-supervised deep neural network approach," in *CVPR 2018 Workshop on Autonomous Driving*, Salt Lake City, 2018.

[5] C. Liu, K. Kim, J. Gu, Y. Furukawa, and J. Kautz, "PlaneRCNN: 3D plane detection and reconstruction from a single image," in *IEEE Conf. on Comp. Vis. and Pattern Recog. (CVPR)*, 2019, pp. 4445–4454.

[7] J.-R. Chang and Y.-S. Chen, "Pyramid Stereo Matching Network," in *IEEE Conf. on Comp. Vis. and Pattern Recog. (CVPR)*, 2018, pp. 5410–5418.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *IEEE Int. Conf. on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[9] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep Ordinal Regression Network for Monocular Depth Estimation," in *IEEE Conf. on Comp. Vis. and Pattern Recog. (CVPR)*, 2018.

[10] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," *ArXiv preprint*, 2021.

[11] X. Wang, D. F. Fouhey, and A. Gupta, "Designing deep networks for surface normal estimation," in *IEEE Conf. on Comp. Vis. and Pattern Recog. (CVPR)*, 2015, pp. 539–547.

[12] X. Qi, R. Liao, Z. Liu, R. Urtasun, and J. Jia, "Geonet: Geometric neural network for joint depth and surface normal estimation," in *IEEE Conf. on Comp. Vis. and Pattern Recog. (CVPR)*, 2018, pp. 283–291.

[13] U. Kusupati, S. Cheng, R. Chen, and H. Su, "Normal assisted stereo depth estimation," in *IEEE Conf. on Comp. Vis. and Pattern Recog. (CVPR)*, 2020, pp. 2186–2196.

[14] A. R. Zamir, A. Sax, N. Cheerla, R. Suri, Z. Cao, J. Malik, and L. J. Guibas, "Robust learning through cross-task consistency," in *IEEE Conf. on Comp. Vis. and Pattern Recog. (CVPR)*, 2020, pp. 11 194–11 203.

[15] J. Wietrzykowski and P. Skrzypczyński, "A probabilistic framework for global localization with segmented planes," in *2017 European Conference on Mobile Robots (ECMR)*, 2017, pp. 1–6.

[16] F. Yang and Z. Zhou, "Recovering 3D Planes from a Single Image via Convolutional Neural Networks," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 87–103.

[17] C. Liu, J. Yang, D. Ceylan, E. Yumer, and Y. Furukawa, "PlaneNet: Piece-wise planar reconstruction from a single RGB image," in *IEEE Conf. on Comp. Vis. and Pattern Recog. (CVPR)*, 2018, pp. 2579–2588.

[18] Z. Yu, J. Zheng, D. Lian, Z. Zhou, and S. Gao, "Single-image piece-wise planar 3D reconstruction via associative embedding," in *IEEE Conf. on Comp. Vis. and Pattern Recog. (CVPR)*, 2019, pp. 1029–1037.

[19] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *IEEE Conf. on Comp. Vis. and Pattern Recog. (CVPR)*, 2017, pp. 936–944.

[20] M.-G. Park and K.-J. Yoon, "As-planar-as-possible depth map estimation," *Comp. Vision and Image Underst.*, vol. 181, pp. 50–59, 2019.

[21] M. Eder, P. Moulon, and L. Guan, "Pano popups: Indoor 3D reconstruction with a plane-aware network," in *2019 International Conference on 3D Vision (3DV)*, 2019, pp. 76–84.

[22] Z. Jiang, B. Liu, S. Schulter, Z. Wang, and M. Chandraker, "Peek-a-boo: Occlusion reasoning in indoor scenes with plane representations," in *IEEE Conf. on Comp. Vis. and Pattern Recog. (CVPR)*, 2020, pp. 110–118.

[23] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3D reconstructions of indoor scenes," in *IEEE Conf. on Comp. Vis. and Pattern Recog. (CVPR)*, 2017.

[24] J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison, "SceneNet RGB-D: Can 5M synthetic images beat generic ImageNet pre-training on indoor segmentation?" in *IEEE Int. Conf. on Computer Vision (ICCV)*, 2017, pp. 2697–2706.

On the descriptive power of LiDAR intensity images for segment-based loop closing in 3-D SLAM

Jan Wietrzykowski[1] and Piotr Skrzypczyński[1]

Abstract— **We propose an extension to the segment-based global localization method for LiDAR SLAM using descriptors learned considering the visual context of the segments. A new architecture of the deep neural network is presented that learns the visual context acquired from synthetic LiDAR intensity images. This approach allows a single multi-beam LiDAR to produce rich and highly descriptive location signatures. The method is tested on two public datasets, demonstrating an improved descriptiveness of the new descriptors, and more reliable loop closure detection in SLAM. Attention analysis of the network is used to show the importance of focusing on the broader context rather than only on the 3-D segment.**

I. INTRODUCTION

3-D laser SLAM methods recently became one of the key components of autonomous vehicles. A majority of them are based on variants of the ICP (Iterative Closest Points) concept, i.e. on matching laser points to other points or ad-hoc created local structures, such as planes and line segments [1], [2], [3]. The accuracy of such methods is sufficient for most navigational tasks, thus the development is focused on making them more reliable and robust by giving an ability to recover from failures and to correct drift when the same location is revisited. Unfortunately, working on the level of points makes the tasks of loop closure and re-localization difficult due to a lack of discriminative features that could be matched between temporarily distant observations. On the other hand, point cloud retrieval, like in [4], provides only place recognition, not metric localization. A different approach to laser SLAM is to cluster the point clouds into larger segments representing meaningful objects or their parts, e.g. cars, trees, parts of buildings [5]. Those segments can be then matched between the current observation and the map, enabling metric global localization used for loop closing and re-localization. However, relying solely on the geometry of isolated objects leaves a lot of uncertainty, because there could be many objects with similar shapes e.g. trees, walls. We conjecture that it would be beneficial for the descriptiveness of those segments to include also information about the texture and the surroundings, which together constitute a broader (visual) context.

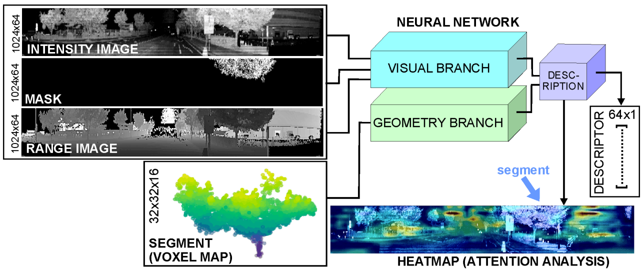

Fig. 1. Concept of enhancing descriptors of 3-D segments by exploiting the broader visual context acquired from LiDAR intensity images. The included attention map shows that the network focuses on the context (warmer colors) rather than on the masked segment (pointed by the blue arrow).

Using a single sensor is practical in applications, as it does not require accurate calibration that is necessary if a LiDAR-camera pair is employed [6]. However, it is problematic to obtain a visual context in LiDAR-only perception. Fortunately, modern LiDARs provide also information about the intensity of the reflected beam. Recent results [7] suggest, that LiDAR intensity is more reliable for place recognition than regular camera images in some scenarios. The nearer the wavelength to the visible light spectrum, the closer the readout to a passive camera image [8]. This comes in handy when multi-beam LiDARs are used, which provide a relatively dense scan of the scene. By arranging intensity readouts in a regular grid on a cylinder around the LiDAR, this information can be treated similarly to a camera image. The intensity image carries a lot of information, not only about the appearance of the segment it may depict but also about the segment's surroundings. Thus, it can be used to retrieve the visual context.

This paper attempts to bridge the gap that exists between the 3-D segment-based approach to loop closing, which turned out to be very practical in LiDAR SLAM [3], and the appearance-based approach, which is commonly applied in visual SLAM. To our knowledge, we are the first to use intensity images to enhance the learned descriptors of 3-D segments. We are also not aware of any work that researched learning description of segments visible in images. Most papers tackle the problem of describing points and their local surrounding or the problem of global image descriptor computation. Our solution falls in-between these two extremes, learning to describe segments that occupy part of the image, while also including the context in the description (Fig. 1). The contribution of this paper is as follows [1]:

- A method using intensity images to enhance segment

*This work was supported by the National Science Centre (NCN), contract no. UMO-2018/31/N/ST6/00941. The work of P. Skrzypczyński was supported by TAILOR, a project funded by EU Horizon 2020 research and innovation program under GA No. 952215

[1] Jan Wietrzykowski and Piotr Skrzypczyński are with Institute of Robotics and Machine Intelligence, Poznan University of Technology, Piotrowo 3A, 60-965 Poznań, Poland name.surname@put.poznan.pl

[1] Code available here: https://github.com/LRMPUT/segmap_vis_views

descriptors (Section IV-A).

- A novel architecture and training methodology for learning descriptors (Sections IV-B and IV-C).
- A procedure for attention analysis of the neural network in a case where output is a descriptor for in-depth analysis of our method (Section IV-D).
- Experimental verification of these methods (Section V).

II. RELATED WORK

In the field of LiDAR-based global localization, the use of segments is not broadly explored. SegMatch, the predecessor of SegMap, introduced incremental segment growing and used hand-crafted features based on eigenvalues during matching [9]. Tinchev *et al.* [10] modified SegMap to use different descriptors learned by a lightweight network with X-Conv operations.

A significantly different approach to global localization using LiDAR measurements was presented by Chen *et al.* [11]. They used images of cylindrical projections in a similar way to our solution, but estimated overlap and yaw angle between a pair of views. Contrary to our segment descriptor, they use a global descriptor of the whole image as an input to regression branches. As this method outputs only similarity measure and relative yaw, ICP registration is needed to compute the 2-D pose. Handcrafted global descriptors of LiDAR scans were used in LocNet [12], but machine learning was applied to compare those descriptors.

Most of the work on describing appearance comes from the computer vision community and focuses on camera images. SuperPoint [13] is one of the recent examples of learned detectors and descriptors. The description is learned using warped images by minimizing hinge loss between all pairs of pixels. On the other hand, in [14] descriptor projections are learned using triplet loss with L2 distance and hard negative mining. The authors of [14] also show that including the context in a form of descriptors of larger portions of the image around the keypoint is beneficial for matching. However, they provide only limited experiments with whole images to describe the context. When it comes to range data, PointNetVLAD [4] was the first DNN-based method producing a discriminative global descriptor for the localization task cast as large scale 3-D point cloud retrieval. Using a similar approach to global localization with 3-D point clouds, PCAN [15] introduced an attention mechanism that predicts significance of each point using its local geometrical context, but not the visual information.

The concept of using intensity information in global localization was explored only in a few studies. Synthesized intensity images compared favorably in [7] to regular camera images for DNN-based place recognition under varying weather conditions. If handcrafted features are employed, as in [16], where histograms of intensity were used, the geometric information is not embedded in the descriptors and used only for hypothesis verification. Histograms were also used by Guo *et al.* [17] in descriptors called ISHOT, which combined the SHOT descriptors and histograms of intensity differences around the keypoints. When only local displacement is estimated, it is possible to exhaustively search the space of possible solutions. L3Net [18] exploits cost volume spanning (x,y,yaw) space and directly minimizes the displacement error during training. However, the description is done by detecting keypoints and describing local patches using coordinates and intensity values by a simple multi-layer perceptron.

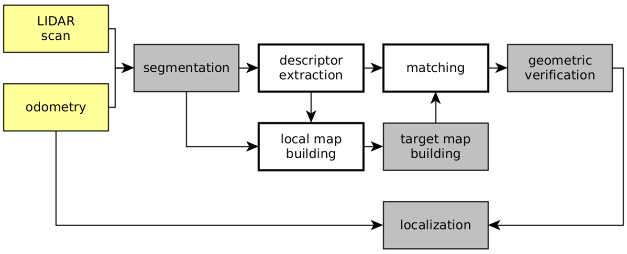

Fig. 2. An overview of the localization pipeline in SegMap with marked modified modules (bolded, white rectangles) and unmodified modules (gray rectangles).

III. LOCALIZATION FRAMEWORK

We demonstrate our novel approach to the context-aware description of 3-D segments extending the open-source, modular SegMap framework [5] for mapping and global localization. Our solution inherits from SegMap the processing pipeline of the segment's geometry (clustering into segments) and the general structure of the global map. In the framework, we plug-in the new procedure of learning the descriptors, together with a modified deep neural network (DNN) architecture, and a new segment matching procedure, which better exploits the enhanced descriptors (Fig. 2).

The processing starts with an acquisition of a new LiDAR scan. The scan is then matched to the previous one to estimate a sensor's displacement. The displacement is integrated with the previous pose estimate and produces odometry, but is also used to compensate the sensor motion by transforming all points to the common frame of reference [2]. The local map of segments is obtained using the SegMap clustering procedures (incremental Euclidean segmenter) and the odometry estimates. In comparison to the original SegMap, we also store synthesized intensity and range images in the local map. Next, segments in the local map are described, which is the main focus of this paper. The sensor pose estimate with respect to the target map is computed by finding match candidates for each segment's descriptor in the local map. The candidates are being found using kNN search in the descriptor space among the descriptors of segments from the target map. From all match candidates, the best subset is selected through consistency clustering and the final pose is computed using this subset. As in SegMap, the target map can be either loaded from the disk or built along the trajectory by accumulating local map segments with their descriptors using current pose estimates.

IV. DESCRIPTION OF SEGMENTS

Segments in the proposed solution are described using only LiDAR data, however, they owe their descriptiveness to the use of range readouts along with intensity data. A local voxel grid representation of the segment is built from the range data, like in SegMap, whereas the intensity data is converted to an image that provides a camera-like view of a segment. An intensity image can be synthesized by directly exploiting the arrangement of measurement directions, as in the case of Ouster LiDAR, or by a projection of the acquired point cloud onto a cylinder surrounding the sensor if the arrangement of the used readouts is not grid-like (e.g. from a Velodyne sensor).

A. *Visual input from a LiDAR*

To describe the segment, an intensity image with the largest area of the projected segment is selected from available visual views and fed to the DNN along with the voxel grid representation of the segment. Special care has to be taken to ensure that binary masks, denoting positions of segments in images, are aligned with objects in those images. It is not feasible to track the origin of every point in a point cloud representing a segment, because of multiple filtering operations and the associated high computational and memory requirements. Thus, we decided to compute masks by projecting point clouds onto images. The simplest approach would be to project points onto a cylinder around the scanner as follows:

$$r = \arg\min_{r'} |\alpha_{r'} - \alpha|, \quad c = \operatorname{round}\left(\frac{\beta}{2\pi} \cdot 1024\right), \qquad (1)$$

where (r, c) are resultant pixel coordinates, r' iterate over LiDAR's scanning rings, α is an inclination angle of the point, β is an azimuth angle of the point, and $\alpha_{r'}$ is an inclination angle for the ring r'. However, a mask computed this way would be inaccurate, because measurements of pixels were not taken at the same time like in a global shutter camera. Moreover, the time it takes a LiDAR to do a full scan is considerably longer than in a regular rolling shutter camera. When this fact is ignored and points are projected onto the image surface using only one pose of the LiDAR, masks can be misaligned by a large margin, depending on a velocity (Fig. 3). We deal with this problem by keeping directions of rays coming out from the LiDAR, transformed to a common frame of reference by using displacement estimated by the odometry procedure. Then, during projection, for every point $\mathbf{p}$, we choose the pixel (i, j) with the closest direction $\mathbf{n}_{ij}$ of the scanning ray. To speed up computations, we search only in a vicinity of the pixel (r, c) that would be selected by the simple projection:

$$(i^*, j^*) = \underset{\substack{r-r_m \leq i < r+r_m \\ c-c_m \leq j < c+c_m}}{\arg\min} \; 1 - \arccos \frac{\mathbf{n}_{ij} \cdot \mathbf{p}}{|\mathbf{p}|}, \qquad (2)$$

where (i^*, j^*) are pixel coordinates of the pixel with the closest direction, $r_m = 16$ and $c_m = 32$ are margins in which we search, set experimentally to values that let always find the globally optimal pixel in the training dataset. We also check if the distance to the point is consistent with the range measurement to account for possible occlusions stemming from the motion compensation. The final mask is an image with pixels equal to 0 where no points were projected and 1 otherwise.

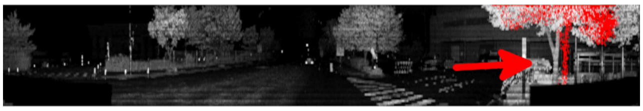

Fig. 3. Exemplary mask misalignment caused by ignoring motion compensation of LiDAR scans.

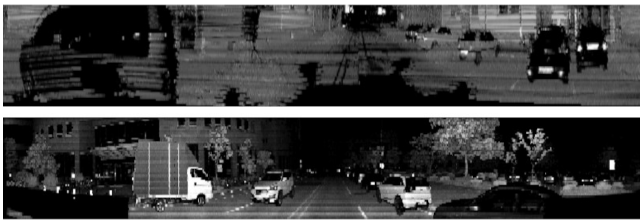

Fig. 4. Visual comparison of intensity images quality from Velodyne HDL-64E (top) and Ouster OS1-64 (bottom) showing that using intensity data from the KITTI dataset might not bring significant benefits.

Laser scans from the KITTI [19] dataset required some additional processing, as in this case only motion compensated point clouds are available. Thus, we perform angular bilinear interpolation to form an intensity image from measurements in directions not forming a regular grid. For every direction on the regular grid, we select 4 nearest neighbors that fall into bins spanning from the current direction on the grid to the nearest directions on the grid. Every neighbor is selected from a different quadrant of the image plane around the point. Then, a horizontal angular interpolation is performed on the pair of upper and the pair of lower points separately to compute two points with the same horizontal angle as the points on the grid. Finally, a vertical interpolation is done to compute range and intensity for the target direction. Unfortunately, the longer laser wavelength in HDL-64E, the lack of raw measurements, and necessary interpolations resulted in a degraded quality of the intensity images from KITTI (Fig. 4) that might be hindering their use to produce visual descriptors.

B. *DNN architecture*

The architecture of the DNN used to produce segment descriptors is depicted in Fig. 5. It consists of two branches that are merged down-stream the computations: a geometry branch processing voxels that is the same as in SegMap, and a proposed visual branch processing intensity images. An input to the visual branch is composed of three layers: intensity image, range image, and segment mask concatenated into a single 3 channel tensor. The intensity image is normalized to have mean value 0 and standard deviation 1 across the whole training dataset, range image is normalized to have mean value 0 for all pixels belonging to the segment and standard deviation equal to 1, and mask values are 1 for pixels belonging to the segments and 0 otherwise. The mask

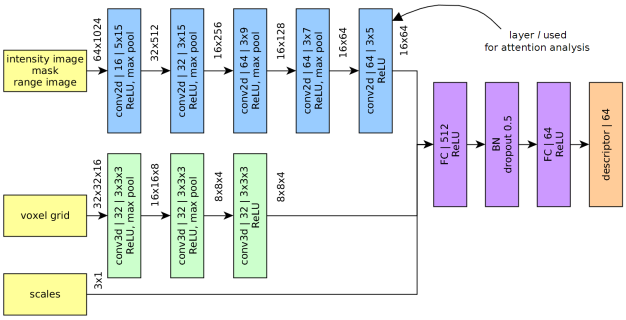

Fig. 5. DNN architecture that combines visual (LiDAR intensity) and geometrical information to produce a segment descriptor. Two convolutional branches are merged using a fully connected layer.

tells the DNN what it should describe and range information gives additional hints about the boundaries of objects. By feeding the whole image instead of only a segment part, we enable the DNN to leverage the context of the segment, because its neighborhood is visible. Typically to image processing DNNs, convolutions compress the information into higher-level features. Finally, outputs from both branches are concatenated and the descriptor of 64×1 size is computed after a fully connected layer.

C. Learning

The same as in [5], we cast the description task as a classification problem, where each segment represents a different class. Due to a large number of classes comparing to the number of training examples, the DNN produced useful descriptors without overfitting to the specific features of the classes. We augmented training intensity images by exploiting their circular nature and randomly rotating the image around the vertical axis. To maximize the number of distinct training examples, each segment observation was assigned an intensity image with the same timestamp during training. Whereas during testing, we always selected the view where the segment was the most visible so far, i.e. the mask was the largest among already collected scans. In both, training and testing, we reject images whose mask area is smaller than 50 pixels.

For DNN's parameter optimization we used Adam optimizer, batch size of 8, learning rate 0.0001, and trained for 256 epochs, selecting a model with the highest validation accuracy.

D. Attention analysis

There is a number of papers describing algorithms for DNN attention analysis for the classification problem [20], but the authors were unable to find a method that is suitable when the network's output is a descriptor. For the classification problem, ScoreCam [20] is a viable solution that does not exhibit problems with visually noisy results as the gradient-based methods [21] and is relatively easy to implement. ScoreCam computes attention heatmaps by analyzing the last layer of the DNN that has spatial dimensions (layer l, c.f. Fig. 5), where the information is compressed the most, but the spatial structure is conserved. For each channel in this layer, it calculates its importance by evaluating a score for the target class c. The weight of the k-th channel is a softmax output for the target class computed by doing a forward pass with a masked input image, thus the higher the probability of the target class, the higher the weight of the channel:

$$w_k^l = f(X \circ M_k^l)[c], \tag{3}$$

where $[c]$ is a result for the target class c, $\circ$ is the element-wise product, $f(\cdot)$ denotes the forward pass, and X is the input image. The mask M_k^l is calculated by upsampling activations in this channel to the size of the network's input and normalizing them:

$$M_k^l = \frac{\mathrm{up}(A_k^l) - \min\left(\mathrm{up}(A_k^l)\right)}{\max\left(\mathrm{up}(A_k^l)\right) - \min\left(\mathrm{up}(A_k^l)\right)}, \tag{4}$$

where A_k^l denotes the k-th activation map for the layer l and $\mathrm{up}(\cdot)$ is the operator of upsampling. This way the mask highlights image parts that were important for the activations in this channel while suppressing other parts. The final heatmap H^l is produced by multiplying channel activations with corresponding weights, upsampling to the size of the input, summing, removing negative values, and normalizing:

$$\begin{aligned} \tilde{H}^l &= \max\left(\sum_k w_k^l \cdot \mathrm{up}(A_k^l), 0\right), \\ H^l &= \frac{\tilde{H}^l - \min\left(\tilde{H}^l\right)}{\max\left(\tilde{H}^l\right) - \min\left(\tilde{H}^l\right)} \end{aligned} \tag{5}$$

With descriptors as output of the DNN the problem stems from the lack of a target class. To deal with this problem, we decided to weight channels on the basis of similarity to the descriptor computed using non-masked input. This way we measure how much the considered channel contributes to the descriptor and the smaller the difference the bigger the weight. Denoting $\mathbf{d} = f(X)$ the descriptor for unmodified input, we compute the weights as:

$$w_k^l = \frac{a}{|f(X \circ M_k^l) - \mathbf{d}|}, \tag{6}$$

where a normalizes weights to sum up to 1.

V. PERFORMANCE ANALYSIS

We tested our solution using two datasets with different types of environments and sensors. The first one is MulRan [22] recorded in a lower density urban area near Daejeon in Korea with Ouster OS1-64 LiDAR, and the second one is KITTI [19] recorded in a residential area near Karlsruhe in Germany with Velodyne HDL-64E. In both cases, we used two sequences for training purposes, namely DCC01 and KAIST01 for MulRan, and 05 and 06 for KITTI. We wanted to compare different types of sensors, hence our choice was MulRan that used Ouster sensor that operates in a different IR wavelength than most of the other scanners, and KITTI to enable comparison with SegMap and SegMatch.

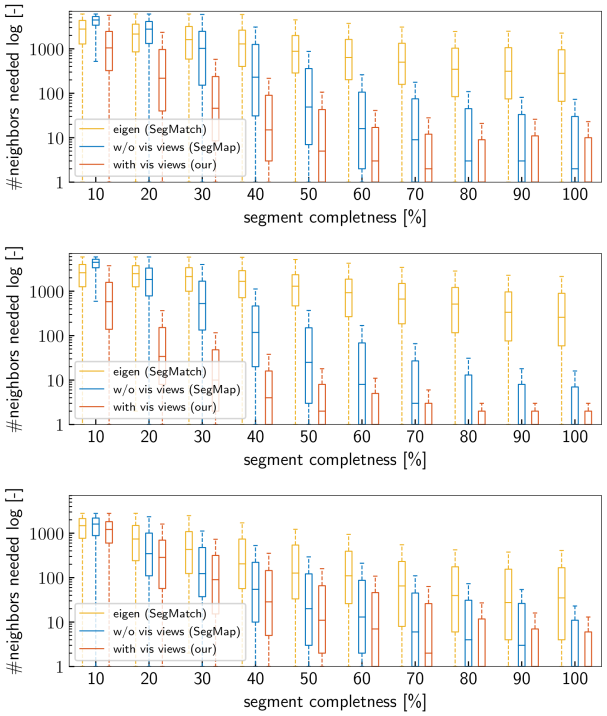

Fig. 6. Analysis of the descriptiveness of the proposed solution comparing to geometrical descriptors and eigen-based features on MulRan DCC03 (top), MulRan KAIST02 (middle), and KITTI 00 (bottom).

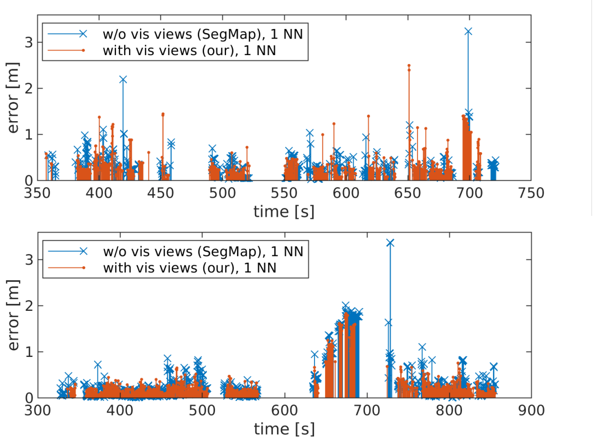

Fig. 7. Errors of relative positions (translational errors) computed for the recognized loop closures along the DCC03 (top) and KAIST02 (bottom) trajectories. Loop closures are uniformly distributed along the entire trajectories for both methods but maximum errors are smaller for our.

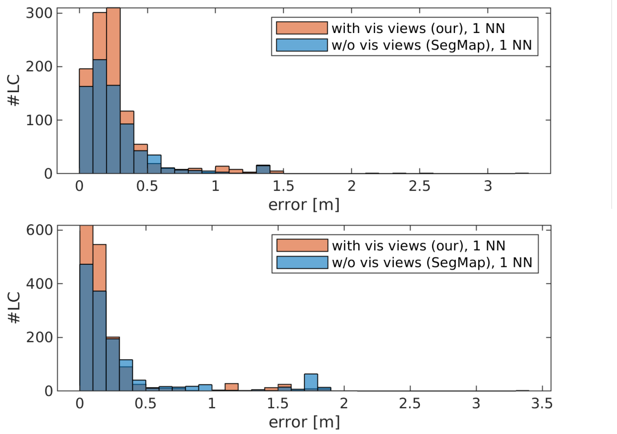

Fig. 8. Histograms of errors in the estimates position (translational errors) computed for the recognized loop closures along the DCC03 (top) and KAIST02 (bottom) trajectories. Darker blue color denotes the overlapping bars of both methods. It is clearly visible that our method produces more loop closures of high position accuracy.

We evaluated the performance of the proposed solution using complete sequences from both datasets that were not used during training. For MulRan it were DCC03 and KAIST02 that have multiple loops and no large ground truth pose gaps (which appear in e.g. DCC02). For KITTI it was the 00 sequence, that is long enough, has multiple loops, and the ground truth trajectory is accurate (as opposed to e.g. 08). It gave us 5376 (52589 views), 5289 (54991 views), and 2659 (32073 views) testing segments, respectively, for DCC03, KAIST02, and KITTI 00. Segments from the target map are retrieved for matching using nearest neighbors in the space of descriptors. Thus, to evaluate the performance of our descriptors, we used the same measure as used in [5], [9], calculating how many nearest neighbors are necessary to retrieve a positive match, i.e. another observation of the same segment, excluding observations from the same sequence of observations. We call this value segment's *rank* and investigate it as a function of segment's *completeness*, i.e. ratio of the size of a point cloud representing segment at a given time instance to the size after all observations were merged. Segment completeness changes naturally when the vehicle moves in the environment and new measurements are incorporated into the segment representation. Additionally, we compare our results to hand-crafted features based on the eigendecomposition of point clouds, used in SegMatch (denoted as *eigen*). Except for SegMatch and the baseline SegMap, we were unable to directly compare with other systems, either because they are not segment-based (e.g. [4], [11], [17]), or there is no open source code available, as in [10]. Figure 6 depicts results for different stages of segment completeness for the DCC03, KAIST02, and KITTI 00 sequence. They indicate that descriptors with visual views outperform their purely geometric counterparts in all intervals. It is worth noting that from 80% of completeness, more than half of the descriptors have a positive match as the first nearest neighbor. This fact is especially important because, during on-line operation, usually complete or almost complete segments are used. As expected, the lowest gain is observed for the KITTI dataset containing scans from Velodyne HDL-64E which provides intensity images of considerably worse quality than those from Ouster OS1-64.

To show the application potential of the new descriptors, we analyzed the number and quality of loop closures produced using them. In most cases, including even one incorrectly recognized loop closure in SLAM optimization

TABLE I
RESULTS OF LOOP CLOSURES. NUMBERS IN RED INDICATE INCORRECTLY DETECTED LOOP CLOSURES THAT ARE UNACCEPTABLE IN SLAM.

seq.	descriptor	#corr.	#incorr.	error [m]
MulRan DCC03	w/o vv (SegMap), 25 NN	1405	3	0.36
	with vv (our), 25 NN	1330	135	4.97
	w/o vv (SegMap), 1 NN	764	0	0.27
	with vv (our), 1 NN	1076	0	0.27
MulRan KAIST02	w/o vv (SegMap), 25 NN	2081	141	14.69
	with vv (our), 25 NN	1906	163	15.65
	w/o vv (SegMap), 1 NN	1402	0	0.33
	with vv (our), 1 NN	1620	0	0.23
KITTI 00	w/o vv (SegMap), 25 NN	423	0	0.75
	with vv (our), 25 NN	469	0	0.75
	w/o vv (SegMap), 1 NN	256	0	0.69
	with vv (our), 1 NN	205	0	0.67

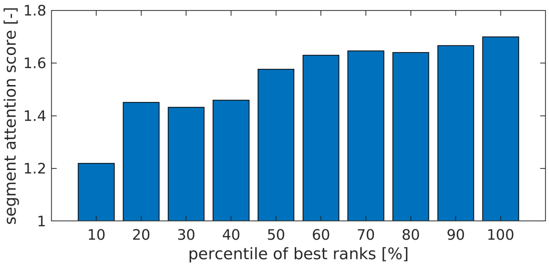

Fig. 9. Segment attention score as a function of performance of correct segment retrieval. The better the rank, the less attention is placed on the segment itself and the more on its context.

has a strong negative impact on the estimated trajectory [1]. Therefore, place recognition systems should avoid those, even at a cost of fewer correctly recognized places. During initial experiments, with 25 nearest neighbors fetched for every segment in the local map (denoted *25 NN*), it turned out that both solutions, without and with visual views, produced such incorrect recognitions. However, we observed that in most cases it is sufficient to get just the first nearest neighbor for our descriptor. Inspired by Lowe's criterion on matching SIFT descriptors [23], we proposed to accept only match candidates which are the first nearest neighbor and for which the distance of descriptors multiplied by 1.2 is smaller than the distance to the second nearest neighbor (denoted *1 NN*). Using this criterion we eliminated incorrect recognitions while preserving a high number of correct ones. Plots of translational errors in time are presented in Fig. 7, while a histogram of these errors is shown in Fig. 8. The mean position errors and numbers of recognized locations for all considered sequences are gathered in Tab. I. For this analysis we assumed that recognitions with a translational error greater than 5 m are incorrect. The time plots of translational error qualitatively show that loop closures for our version are approximately uniformly distributed along the entire trajectories and are not focused in one part. They also depict that maximum errors are smaller for our version. The histograms show that for the MulRan sequences our method yields a higher number of loop closures that are very accurate, which is beneficial to SLAM. The quantitative results demonstrate also that for SLAM applications the *25 NN* version is not suitable because of the incorrect recognitions. For the *1 NN* version and Ouster OS-64 (MulRan) our solution yields better results in terms of the number of correct recognitions while being better or comparable in terms of the mean position error. The results for Velodyne HDL-64E (KITTI) are inconclusive, which we attribute to the considerably lower quality of the synthesized intensity images.

In terms of inference time, our solution is slightly faster than the original one (143 ms vs 155 ms on average using GTX 1080 Ti) thanks to a smaller number of described segments due to rejection of segments with a too small mask. The most time-consuming operation is the insertion of new visual views to the local map that includes finding the best visual view for every segment and takes 313 ms on average.

We use the attention analysis mechanism to demonstrate how important it is to take into consideration the visual context surrounding the segment on the image. To show this effect quantitatively, we compute every segment's attention score as a ratio of the mean attention of pixels belonging to the mask and its nearest neighborhood in the intensity image, to the mean attention for all other pixels. Figure 9 plots the attention score for 10 bins of segments, sorted according to their rank (the same rank as used during the performance analysis, c.f. Fig. 6). There is a trend showing that the better the rank, the less attention is placed on the segment itself and the more on its context. Figure 10 visualizes attention heatmaps for two segments from the top 10% ranks and thus being correctly associated (top rows), and two others from the bottom 10% ranks that were mismatched (bottom rows). There is a visible shift of attention in the DNN from the segment to the surrounding context for the correct associations, whereas in the incorrectly associated examples the DNN focused on particular objects.

VI. CONCLUSIONS

We presented research aimed at improving loop closing based on the concept of geometric segments, making it possible to consider the visual context that surrounds these segments. This context gives the segment descriptors much more descriptive power. While there are many objects of similar shapes in real-world outdoor scenes (e.g. cars), a combination of the segment's geometry, its texture, and other textures in the neighborhood is intuitively much more unique. We verified this conjecture in two ways: (i) by demonstrating quantitatively on the MulRan and KITTI datasets that our new descriptors are more robust in matching than the purely geometric ones from SegMap, (ii) by showing through DNN attention analysis that the visual context indeed matters when learning the descriptors. Moreover, we demonstrated the processing pipeline that exploits the intensity readouts of a modern LiDAR in a way similar to passive camera images. The new DNN architecture combines geometric and intensity data at the feature level, producing compact descriptors. Having SLAM applications in mind and exploiting better descriptive power of our solution, we proposed a

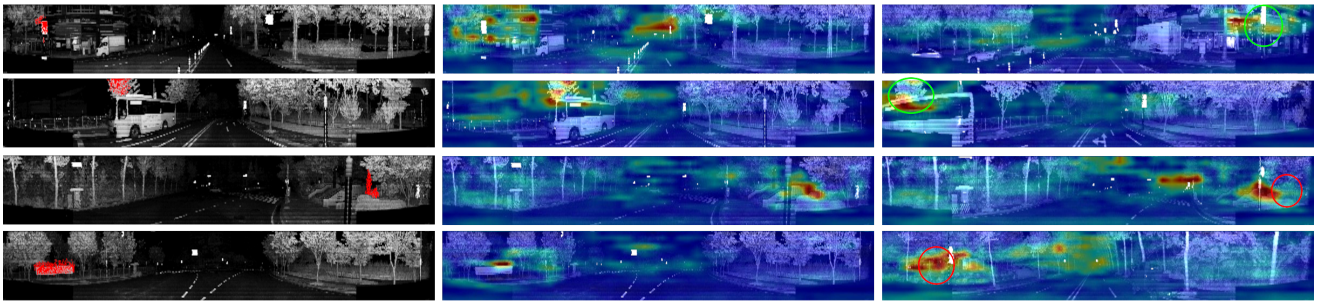

Fig. 10. Examples of first nearest neighbors retrieved from the target map as query seqment views (left), attention heatmaps for query segments (middle), and attention heatmaps for matched segments (right). Two first examples depict correctly matched segments (circled in green) from top 10% ranks, whereas two last depict incorrectly matched ones (circled in red) with correct segments being distant in the descriptor space from bottom 10% ranks.

different method for selecting potential segment matches that eliminates incorrect loop closure detections that could deteriorate pose and map estimates. The presented results are considerably better for Ouster OS1-64 due to good quality intensity images and could be further improved using the latest technology LiDAR with 128 beams [8] and ambient images instead of the intensity images. This possibility will be investigated in our future work.

REFERENCES

[1] K. Ćwian, M. R. Nowicki, J. Wietrzykowski, and P. Skrzypczyński, "Large-scale LiDAR SLAM with factor graph optimization on high-level geometric features," *Sensors*, vol. 21, no. 10, p. 3445, 2021.

[2] J. Zhang and S. Singh, "Low-drift and real-time LiDAR odometry and mapping," *Autonomous Robots*, vol. 41, no. 2, pp. 401–416, 2017.

[3] X. Liu, L. Zhang, S. Qin, D. Tian, S. Ouyang, and C. Chen, "Optimized LOAM using ground plane constraints and SegMatch-based loop detection," *Sensors*, vol. 19, no. 24, p. 5419, 2019.

[4] M. A. Uy and G. H. Lee, "PointNetVLAD: deep point cloud based retrieval for large-scale place recognition," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4470–4479.

[5] R. Dubé, A. Cramariuc, D. Dugas, H. Sommer, M. Dymczyk, J. Nieto, R. Siegwart, and C. Cadena, "SegMap: Segment-based mapping and localization using data-driven descriptors," *International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 339–355, 2020.

[6] M. R. Nowicki, "Spatiotemporal calibration of camera and 3D laser scanner," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6451–6458, 2020.

[7] K. Żywanowski, A. Banaszczyk, and M. R. Nowicki, "Comparison of camera-based and 3d lidar-based place recognition across weather conditions," in *16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2020, pp. 886–891.

[8] Ouster, "OS1 mid-range high-resolution imaging Lidar," 2020. [Online]. Available: https://data.ouster.io/downloads/datasheets/datasheet-revd-v2p0-os1.pdf

[9] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "Segmatch: Segment based place recognition in 3D point clouds," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 5266–5272.

[10] G. Tinchev, A. Penate-Sanchez, and M. Fallon, "Learning to see the wood for the trees: Deep laser localization in urban and natural environments on a CPU," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1327–1334, 2019.

[11] X. Chen, T. Läbe, A. Milioto, T. Röhling, O. Vysotska, A. Haag, J. Behley, and C. Stachniss, "OverlapNet: Loop closing for LiDAR-based SLAM," in *Proceedings of Robotics: Science and Systems*, Corvalis, USA, 2020.

[12] H. Yin, Y. Wang, X. Ding, L. Tang, S. Huang, and R. Xiong, "3D LiDAR-based global localization using Siamese neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1380–1392, 2020.

[13] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: self-supervised interest point detection and description," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 337–033 712.

[14] A. Loquercio, M. Dymczyk, B. Zeisl, S. Lynen, I. Gilitschenski, and R. Siegwart, "Efficient descriptor learning for large scale localization," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3170–3177.

[15] W. Zhang and C. Xiao, "PCAN: 3D attention map learning using contextual information for point cloud based retrieval," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12 428–12 437.

[16] K. P. Cop, P. V. K. Borges, and R. Dubé, "Delight: An efficient descriptor for global localisation using LiDAR intensities," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3653–3660.

[17] J. Guo, P. V. K. Borges, C. Park, and A. Gawel, "Local descriptor for robust place recognition using LiDAR intensity," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1470–1477, 2019.

[18] W. Lu, Y. Zhou, G. Wan, S. Hou, and S. Song, "L3-Net: towards learning based LiDAR localization for autonomous driving," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 6382–6391.

[19] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition*, Rhode Island, 2012, pp. 3354–3361.

[20] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, "Score-CAM: score-weighted visual explanations for convolutional neural networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[21] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[22] G. Kim, Y. S. Park, Y. Cho, J. Jeong, and A. Kim, "MulRan: Multimodal range dataset for urban place recognition," in *IEEE International Conference on Robotics and Automation*, 2020, pp. 6246–6253.

[23] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2022.3185732

PlaneLoc2: Indoor global localization using planar segments and passive stereo camera

JAN WIETRZYKOWSKI[1]

[1]Institute of Robotics and Machine Intelligence, Poznan University of Technology, 60-965 Poznan, Poland (e-mail: jan.wietrzykowski@put.poznan.pl)

Corresponding author: Jan Wietrzykowski (e-mail: jan.wietrzykowski@put.poznan.pl).

This work was supported by the Polish National Science Centre (NCN), contract no. UMO-2018/31/N/ST6/00941.

ABSTRACT This paper introduces PlaneLoc2 - a novel indoor global localization system designed to harness the potential of stereo cameras. A need for robust global localization that does not produce incorrect results (false positives) is present in almost every life-long autonomy task. We show that planar segments extracted from stereo vision data by a neural network enable such robust localization. Planar segments are easier to discriminate than keypoint features and provide easy-to-use geometric constraints. We propose an architecture that exploits a single deep neural network (DNN) to detect planar segments, produce appearance descriptors, and estimate segment geometry. Moreover, we introduce a novel view-based segment map and a novel pose retrieval procedure that considers the uncertainty of features to efficiently use the geometric constraints provided by them. We also show that the new learned descriptor provides better discrimination than the hand-crafted one. Finally, we present experimental results that show that our solution outperforms other state-of-the-art global localization methods and does not produce incorrect agent poses. For both test scenes it recognizes at least 15% more poses than the second best method without incorrect recognitions.

INDEX TERMS Simultaneous localization and mapping, Artificial neural networks, Stereo image processing

I. INTRODUCTION

ACCURACY of modern simultaneous localization and mapping (SLAM) systems over the last years has improved significantly, yet they are still not applicable to many real-world tasks. The main reason is that to work for a prolonged time these systems have to be able to recover from failures and have to correct localization drift that inevitably accumulates over time. When no external source of positioning is available, e.g. in indoor environments where there is no Global Positioning System (GPS) signal, global localization becomes essential. Global localization is a problem of localizing an agent with respect to a known map without knowledge of its previous poses [1]. In the case of metric global localization, the pose (translation and rotation) is expressed in a frame of reference of the map using appropriate representation, e.g. translation vector and rotation matrix. Metric global localization is a vital component of solutions to problems such as recovery after loosing pose tracking due to occlusion or other external factors, or loop closing when a robot arrives at a previously visited scene after traversing a long loop and drift has to be rectified. In order to compute the pose in this situation, it is necessary to match a selected type of features or objects between a local view and the global map. The more discriminative the features or objects, the better, because it is easier to avoid incorrect associations. However, a nontrivial problem is to reliably and repeatedly detect such objects and to exploit geometric constraints provided by their associations. One possibility is to use planar segments that are common in indoor environments. They are not so easily detected as keypoint features and geometric constraints are more complex than point-to-point constraints, nonetheless, they are more discriminative and there are usually fewer of them, which reduces the number of possible association combinations. Therefore, to build a global localization system that will benefit from planar segments, it is necessary to develop proper detection and pose retrieval algorithms. The detection of planar segments is usually done using RGB-D sensors because of the availability of depth information that helps to segment the scene and enables geometry estimation, i.e. plane equations supporting segments. Unfortunately, RGB-D sensors have limited effective range, and other sensors providing depth information, such as LiDARs (Light Detection and Ranging), are expensive. An interesting alternative is a passive stereo camera that also facilitates unambiguous geometry recovery, but has a longer effective range than RGB-D sensors and is cheaper than LiDARs. However, to harness the full potential of stereo cameras, special care has to be taken because stereo estimated depth is not as accurate as the one from RGB-D sensors or LiDARs. Whereas multiple papers discuss planar segment detection without explicit depth information [2], localization using planar segments [3], and some systems allow localization using stereo sensors [4], no significant prior work exists that combines those topics to propose a robust global localization system. This paper closes this gap by introducing the PlaneLoc2 (Sec. III), depicted in Fig. 1. The goal of the presented research is to develop a system that delivers a metric pose of the agent with respect to a known map, using a passive stereo camera, and exploiting planar segments as reference objects. The contribution of the paper can be summarized as follows[1]:

- Extending Stereo Plane R-CNN planar segment detection network with a module to extract the geometry and uncertainty of geometry of planar segments. This enables application of this network architecture to the real-world problem of global localization (Sec. IV-A).
- Developing a planar segment appearance description method that is embedded in the segment detection network. The enhanced descriptor significantly limits the number of potential matches considered during localization (Sec. IV-B).
- Proposing a novel view-based map and a novel pose retrieval method that better suit the characteristic of passive stereo cameras (Sec. V).

The rest of the article is structured as follows. In Sec. II we survey other papers and compare them with our approach. Sec. III is dedicated to the overview of the global localization pipeline. In Sec. IV we describe the planar segment extraction mechanism, while the view-based approach to global localization is presented in Sec. V. The proposed methods are extensively evaluated and compared to other state-of-the-art systems in Sec. VI. Finally, conclusions are drawn in Sec. VII.

This work builds on results from our previous articles. A planar segment detection DNN that enables accurate geometry retrieval was introduced in [5]. We use this network in the PlaneLoc2, but add a segment geometry extraction mechanism that can be used in global localization. The extracted information include the uncertainty that is a vital part of the description of geometry. The segment appearance description learning is inspired by our previous successful loop closing method [6], where descriptors of general (not necessarily planar) segments were computed from LiDAR data. The general idea of inference by building a probability density function (PDF) describing agent pose is borrowed from the PlaneLoc system that uses RGB-D data [7]. However, a completely new mapping approach and pose retrieval procedure are introduced in this article to handle a stereo sensor.

II. RELATED WORK

In this section we describe other papers related with our work. The description is divided into three subsections concerning different aspects of global localization: sensors, features, and methods in general.

A. SENSORS

Rapid development of RGB-D sensors that followed the introduction of Kinect, brought a variety of sensors that use different measurement techniques, such as structured light, time of flight (ToF), and active stereo. However, all those solutions have a limited effective range of 4-6 m [8], even Kinect v2 that is especially vulnerable to reflective surfaces [8]. Therefore, modern RGB-D sensors often resort to passive stereo for larger distances, which increases the effective range [9]. The limited range poses problems for many real-world applications and makes a stereo camera the preferable sensor. The applications include, but are not limited to, tracking human motion [10], SLAM [11], and scene reconstruction [12]. Moreover, depth information is sometimes used to simulate a view-based stereo measurement to achieve better results [4]. Also, when significant scene sizes are considered, stereo is the only viable option [13] with monocular cameras struggling with scale ambiguity [14]. Aware of those results in related areas, we resort to a passive stereo camera to increase the effective range of perception of planar segments with respect to our earlier PlaneLoc system from [7].

B. FEATURES IN GLOBAL LOCALIZATION

One of the key aspects of global localization is a choice of features to be matched. Algorithms like DBoW2 [15], used in ORB-SLAM3, resort to classical, non-learned keypoint features, such as BRIEF (Binary Robust Independent Elementary Features) or ORB (ORiented FAST and Rotated BRIEF). A more recent approach is to use a trained keypoint detector and descriptor, as in [16] where finding dense pixelwise correspondences between two images is enabled by a pyramid of coarse-to-fine features. Learned features are oftentimes combined with learned matching methods, such as SuperPoint detector and descriptor [17] and SuperGlue [18] matcher that uses a graph neural network to aggregate global context. A more localization-oriented feature learning was proposed in [19], where supervision at the level of pose was applied to train a multiscale feature generator. However, the pose estimation is left to a principled algorithm and the method requires a coarse initialization of pose, therefore being not suitable for global localization. In our work, we adopt a different approach and instead of resorting to a complex description and matching methods, we use planar segments that are easier to describe and match.

[1] Implementation and dataset are available at https://github.com/LRMPUT/plane_loc_2

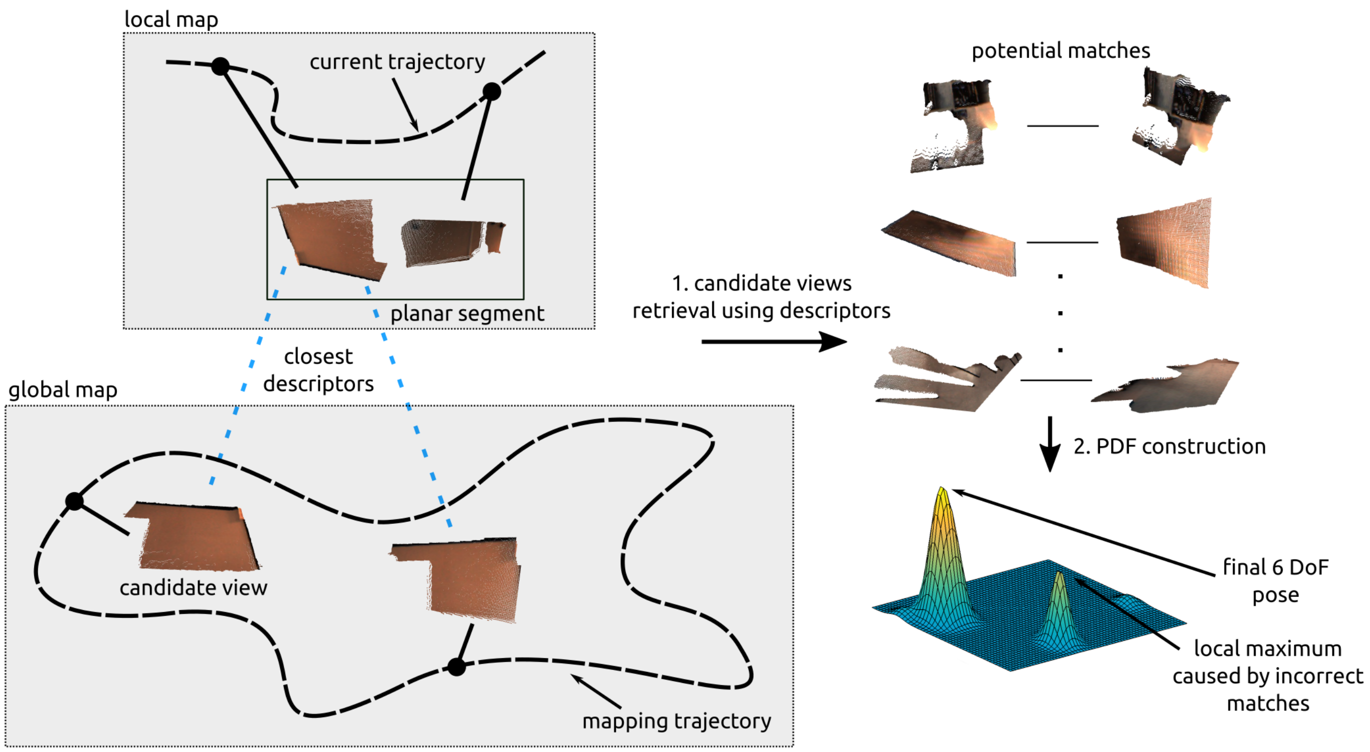

FIGURE 1. PlaneLoc2 retrieves candidate views using appearance descriptors and builds a PDF of pose using all potential matches. The final pose is a maximum of the PDF, verified by the fail-safe checks.

Planar segments are not as commonly used as reference objects, compared to keypoint features, mainly because difficulties with their detection, and with exploiting the geometric constraints they provide. Nonetheless, there are SLAM systems that use planar segments, such as the one presented in [20], where planar segments enabled loop closures in a LiDAR-based system. LiDAR measurements facilitates accurate estimation of planar segments' geometry, therefore the solution cannot be directly applied to a camera-based system, such as ours. In camera-based SLAM, planar segments were used in [3], [21], however, planar constraints were used only during incremental localization and loop closing was based on keypoint features. Contrarily, PlaneLoc2 uses planar segments to recover global pose, which is a part of loop closing procedure. A demonstration of global registration of camera pose with planar segments was presented in [22], but no quantitative localization results were provided. Global localization was also considered in [23], where graphs of incidence of planar segments were used to compare their sets. However, the method was tested only in a small environment, where objects were close to a sensor and their geometry could be accurately estimated using RGB-D data. In opposition, in this paper, we quantitatively evaluate the proposed solution in a workshop-sized environment to enable a fair comparison with other systems.

C. GLOBAL LOCALIZATION METHODS

Most of the global localization methods use associations between keypoint features to recover a pose. Loop closing and relocalization mechanisms in ORB-SLAM3 [4], based on DBoW2 [15], use sparse ORB features and hierarchical tree to quickly retrieve candidate images to match against. The pose is computed by point-to-point correspondences and later verified by tracking a local map. A solution using learned descriptors is presented in [24], where candidate images are found using NetVLAD [25] descriptor, followed by dense matching and pose verification using view synthesis. Unfortunately, view synthesis requires the database images to contain dense depth maps, which can be troublesome to obtain. A conceptually similar approach was described by Sarlin *et. al* [26], where localization is done in two steps: global candidate images retrieval, followed by local feature matching. Our solution follows a different strategy than those algorithms, matching directly objects of reference and including context description in the appearance descriptor of those objects. A data-driven approach could alleviate the need to choose a specific strategy and combine benefits of both solution. However, despite the enormous capabilities of DNNs, they have been applied mainly to feature generation and incremental localization [27], whereas global pose retrieval is done using principled algorithms, as in the aforementioned papers.

Uncertainty in global localization is not easy to capture and has been discussed only in a few articles. In [28] a place recognition method was proposed that uses Bayesian filtering with simple motion and sensor models. The model is used in prediction and resampling steps of a particle filter, but the computed place gives only a coarse pose. Another example of Bayesian localization is presented in [29], where authors integrated LiDAR and camera measurements and proposed an efficient inference method with a decomposition of the global map into local places. Those two methods maintain a probability distribution of poses and constrain transitions between locations using a motion model. Such an approach differs from the one presented in this article, because we assume that visual odometry in a short horizon is precise enough to neglect its uncertainty and represent the pose distribution using kernels.

III. GLOBAL LOCALIZATION USING PLANAR SEGMENTS

The RGB-D based PlaneLoc, despite achieving good results in terms of precision and recall, had a few issues that were identified during the research and hindered further development:

- Ignoring planar segments further than 4 m due to a limited effective range of RGB-D sensors. During global localization, using only the part of the image that is close to the sensor significantly limits the context and limits the number of geometrical constraints.
- Using poorly discriminating appearance descriptors based on color histograms. They were dependent on illumination and did not include context, therefore their comparison produced many spurious potential matches.
- Using pose retrieval optimization based on infinite planes. It did not include information about the boundaries of planar segments and produced implausible solutions that had to be additionally verified.

In the new approach, PlaneLoc2, the above-mentioned issues were addressed to improve robustness and recall. Nonetheless, the inference procedure is based on the previous version [7] in which all plausible pose hypotheses are generated and a PDF representing knowledge about the pose is built. In the PDF the maximum is sought and additional asserts are performed to ensure that the returned pose is correct. The same idea is applied here, although most of the other components had to be redesigned to benefit from a stereo sensor.

The processing pipeline (see Fig. 2) starts with planar segments detection and description using a DNN. To maximize computation sharing during this stage, we use a single DNN that extracts all information necessary for further processing, including segments' 3-D geometry. The geometry and visual odometry are used to match segments from the current frame to those present in the local map. Information from the current frame is then used to either update segments in the local map or to add new ones, depending on the matching results. Both maps, the local and the global one, do not merge segments explicitly to get a single representation but rather store information about views of the segments.

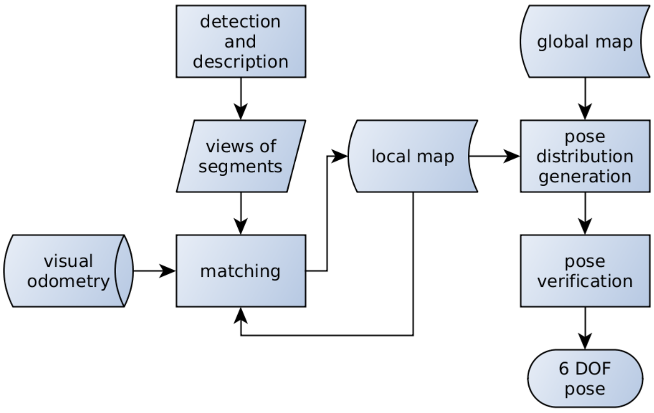

FIGURE 2. Processing pipeline of PlaneLoc2.

After updating the local map, a localization procedure is performed, that associates segments between the local and the global map and builds the PDF. The procedure starts with the retrieval of candidate global map views using appearance descriptors. As a result of using deep learned descriptors that provide good discrimination, only 2 candidate views have to be retrieved to get a high probability of including a correct match. Using retrieved views, all plausible pose hypotheses are generated by examining triplets of matched segments and every hypothesis is inserted into the 6-D pose PDF as a kernel:

$$p(\mathbf{q}) = \frac{1}{Z}\tilde{p}(\mathbf{q}) = \frac{1}{Z}\sum_{a} w_a K_a(\mathbf{q}), \qquad (1)$$

where $\mathbf{q}$ is a 6 element pose vector (a logarithm of the $SE(3)$ transformation matrix), Z is a normalizing factor, K_a is a kernel function for hypothesis a, and w_a is a weight of the kernel a. The weights are computed as follows:

$$w_a = \sum_{b} \alpha_b, \qquad (2)$$

where b ranges over all local segment views used in the hypothesis a, and α_b is an area of the segment view b. A novel procedure to retrieve a pose hypothesis based on a set of matches is used to exploit view-based representation and provide as many geometric constraints as possible. The pose retrieval procedure is critical during the pose hypothesis generation and the final pose computation. When the PDF maximum is found and the final pose $\mathbf{q}^*$ is computed, three fail-safe checks are performed to ensure that the pose is correct:

- $\tilde{p}(\mathbf{q}^*) > \tau_p$ - the value of the unnormalized PDF $\tilde{p}(\mathbf{q}^*)$ for the final pose $\mathbf{q}^*$ has to be above a threshold τ_p to assert that enough positive evidence was collected.
- $\min\left(\frac{\alpha_m^l}{\alpha_t^l}, \frac{\alpha_m^g}{\alpha_t^g}\right) > \tau_r$ - the ratio of the area of segment views that were matched α_m to the total area of visible segment views α_t has to be above a threshold τ_r for, both, the local map (denoted by a superscript l) and the

global map (denoted by a superscript g) to verify that there is no significant amount of negative evidence.

- $|\mathcal{M}| > \tau_d$ - the number of distinct matched pairs of segments has to be above a threshold to make sure that the positive evidence is diverse enough.

Our system has three main threads that can be executed concurrently. The first one is responsible for detecting planar segments and creating views – its processing takes 557 ms per frame on average on RTX 3090 GPU. The second one builds and manages the local map, using approximately 11 ms of i5-8250U CPU time for each frame. The last thread is a pose inference thread that returns results every 2883 ms on average using CPU only. The execution time allows to update the local map with a frequency of approximately 2 Hz, which is enough for global localization, since consecutive frames usually do not contain significant amount of new information. Although the local map can be updated with a frequency of 2 Hz, the agent pose cannot be retrieved after each update due to the longer processing time of the inference thread. Nonetheless, in the considered scenarios, information about the global pose yielded every 3 s is enough to recover from loosing pose tracking or to correct the drift.

IV. PLANAR SEGMENTS EXTRACTION

As mentioned in Sec. I, reliable and repeatable object detection is essential if they are to be used during localization. Drawing from development in the object detection field, where DNNs achieve the best results, outperforming classical methods by a large margin, we also use DNN to detect reference objects in the form of planar segments. The DNN, introduced in our recent work [5] (see Fig. 3), simultaneously produces image masks of individual planar segments, their appearance descriptors used to preliminarily match segments between the local map and the global map, and retrieves the 3-D geometry of the segments. It was trained on a photo-realistic synthetic `SceneNet Stereo` dataset containing approximately 35k images from 200 different scenes. The training was started from weights pretrained on the real-world `Coco` and `ScanNet` datasets, same as in [5]. We trained the network for 10 epochs using Adam optimizer with a learning rate equal to 10^{-5} and weight decay equal to 10^{-4}. Training examples were augmented using random color and sharpness manipulation, Gaussian noise, and random cropping. Despite using only a synthetic dataset for the final training, the network performs well on real-world data, as evaluated in Sec. VI.

A. DETECTION

To exploit more information about the scene by including also distant segments, we use a stereo camera instead of an RGB-D sensor. However, stereo estimated depth is not accurate enough to reliably segment an image into planar segments and to fit supporting 3-D planes for those segments. Nonetheless, a pair of stereo images is still a valuable source of information regarding the geometry of the scene and can be used without explicit depth reconstruction. In the PlaneLoc2 a DNN is used to segment image into planar segments and to estimate segments' supporting planes. The Stereo Plane R-CNN architecture detailed in [5] uses camera-agnostic geometry representation to provide robustness to camera parameters change and to enhance the results. To use this network for localization purposes, an export mechanism had to be added that handles the depth uncertainty. Besides a plane equation and a hull denoting the boundary of the segment, we also store a mean value and a covariance matrix of 3-D points forming this segment. The points are calculated using the estimated depth and the uncertainty of their estimation is extracted from the disparity estimation branch of the geometry module of the DNN. In this branch, a cost volume is created that holds the probability distribution over disparity values for each pixel. It is straightforward to compute the standard deviation of disparity σ_d from this distribution:

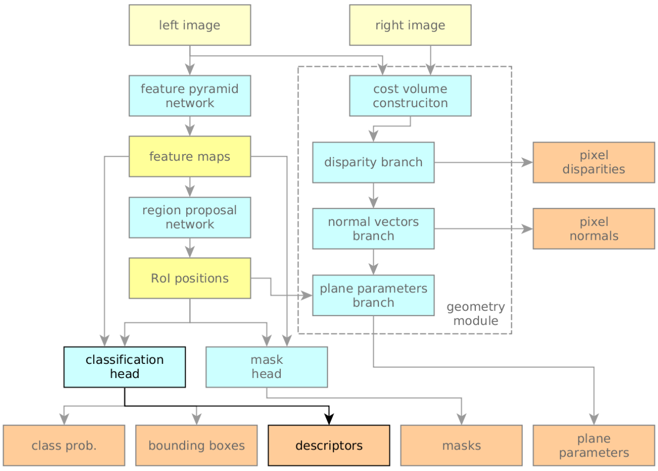

FIGURE 3. An overview of DNN used to detect and describe planar segments. Gray blocks and connections were not modified.

$$\sigma_d = \sqrt{\sum_d p(d)(d - \overline{d})}, \tag{3}$$

where $p(d)$ is a probability that d is a disparity for this pixel, and $\overline{d}$ is an expected value of the disparity. Then, a standard deviation of depth σ_z can be calculated using a camera model as follows:

$$\sigma_z = \sigma_d \frac{z}{f_x b}, \tag{4}$$

where z is a depth value, f_x is a focal length for X axis of the camera, and b is a baseline of the stereo setup. Finally, a covariance of 3-D point $\mathbf{x}_i$ in a camera frame of reference can be approximated as:

$$\mathbf{S}_i = \begin{pmatrix} 0.05^2 & & \\ & 0.05^2 & \\ & & \sigma_z^2 \end{pmatrix}. \tag{5}$$

A small, constant value of uncertainty of 0.05 m was used for the X and Y axes because uncertainty in those directions can be neglected compared to uncertainty in the Z axis. The uncertainties of individual points are aggregated to obtain a covariance matrix of the whole point cloud as follows:

$$\mathbf{S} = \sum_i \mathbf{S}_i + \mathbf{x}_i \mathbf{x}_i^T - \boldsymbol{\mu}\boldsymbol{\mu}^T, \tag{6}$$

where $\boldsymbol{\mu}$ is a centroid of the point cloud. This uncertainty is necessary to accommodate for inaccurate geometry estimation during the association check and the pose retrieval.

B. DESCRIPTION

The DNN also helped resolve another issue of the previous version of PlaneLoc system, namely poorly discriminating appearance descriptors. We added additional layers in a classification head (location of this classification head in the entire structure of the DNN is presented in Fig. 3) of the DNN that produce descriptors as presented in Fig. 4. Features that are used to compute class probabilities and bounding box refinements are processed by a fully connected layer to output a descriptor. However, the most troublesome part of training a DNN that computes descriptors is the way of supervision. Inspired by [6], we also formulate this problem as a classification task. During training, every instance of a planar segment in the training dataset is a separate class and all observations of this segment should be classified as this class. To increase the robustness of the descriptor, a dropout layer is added between the descriptor and fully connected and softmax layers that output segment instance probabilities. The segment instance probabilities are used to compute a cross entropy loss by comparing with target annotations. Correspondences between observations and instances of planar segments that serve as the target annotations are computed using 3-D mesh models, eliminating the need for tedious manual labeling. Such a modification adds little overhead to the Stereo Plane R-CNN model from [5], while producing discriminative descriptors.

V. VIEW-BASED APPROACH TO GLOBAL LOCALIZATION

Distant planar segments, even if not useful to constrain how far the sensor is from the segment because of problems with accurate depth estimation, still provide good orientation constraints. To exploit those constraints, we proposed a novel, view-based map and a pose retrieval procedure that takes into consideration the uncertainty of depth estimation. Moreover, the new pose retrieval procedure treats planar segments as spatially bounded, providing more constraints as opposed to the previous approach that treated them as infinite planes.

A. PLANAR SEGMENT MAP

The new map structure, instead of explicitly merging different observations of the same planar segment to produce a single representation in the form of a point cloud, stores information about separate views of segments. The structure of the map is depicted in Fig. 5, showing the following information is stored for each view:

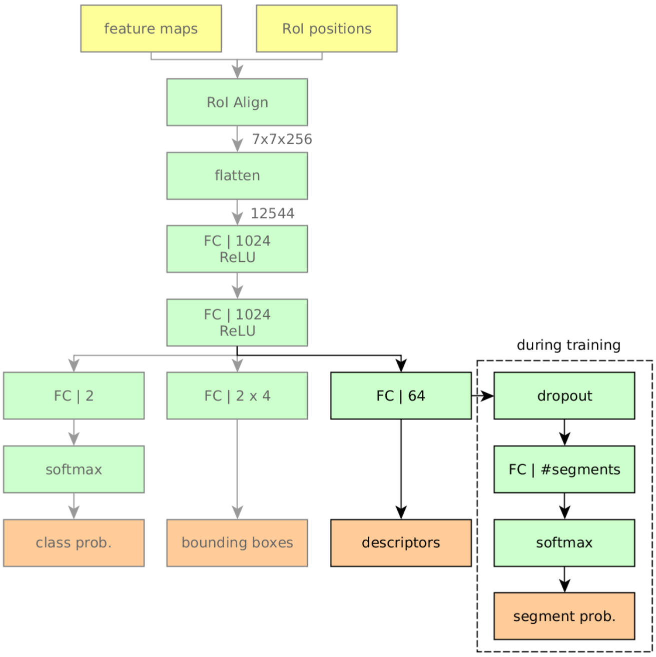

FIGURE 4. A modified classification head from Stereo Plane R-CNN that produces descriptors. Gray blocks and connections were not modified. Location of this classification head in the entire structure of the DNN is presented in Fig. 3.

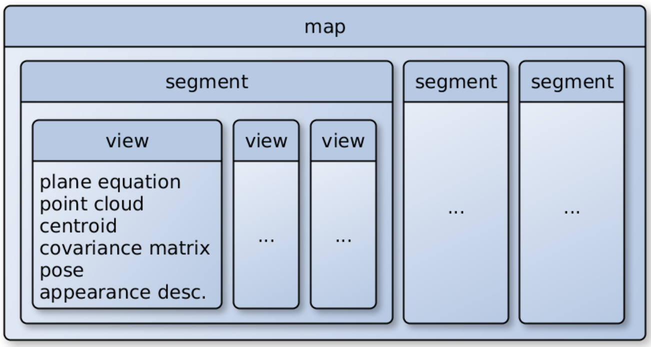

FIGURE 5. The map in PlaneLoc2 contains planar segments, whereas segments store information about their views.

- Plane equation ($\boldsymbol{\pi}$) - estimated by the plane parameters branch of the geometry module.
- Point cloud - 3-D points constituting the segment. Points are reprojected using the stereo estimated depth from the disparity branch of the DNN. To limit storage requirements, they are downsampled using a voxel grid filter with a raster of 0.05 m.
- Centroid and covariance matrix ($\boldsymbol{\mu}, \mathbf{S}$) - computed from the point cloud.
- Pose - a visual odometry pose from which the segment was observed.
- Appearance descriptor - produced by the DNN and used to retrieve global map view candidates.

By avoiding merging, we circumvent the problem of, usually computationally costly, information merging and uncertainty propagation from different views. When a new frame is processed, a depth buffer is built to check which segments from the local map can be visible. During the buffer construction, for every segment we select a view with the observation pose $\mathbf{T}^v$ (expressed as a $SE(3)$ transformation matrix) closest to the current pose $\mathbf{T}^c$, according to the following metric, that is a weighted sum of translational and rotational differences:

$$d\left(\mathbf{T}^c, \mathbf{T}^v\right) = d_t\left(\left(\mathbf{T}^v\right)^{-1}\mathbf{T}_1\right) + w_r d_r\left(\left(\mathbf{T}^v\right)^{-1}\mathbf{T}_1\right), \quad (7)$$

where w_r is a weight of the rotational difference, $d_t(\cdot)$ is a function returning translation of the transformation, and $d_r(\cdot)$ is a function returning rotation of the transformation. The weight $w_r = 5$ is set to make an error of approximately $5°$ equal to an error of 0.5 m. The new views are matched against the potentially visible segments by a geometry test that employs the same error function as the pose retrieval procedure:

$$g\left(\mathcal{P}^c, \mathcal{N}(\boldsymbol{\mu}^v, \mathbf{S}^v)\right) = \sqrt{e_{c,v}\left(\mathbf{I}, \mathbf{0}\right)} < \tau_g, \quad (8)$$

where $\mathcal{P}^c$ is a set of points representing the currently considered new view, $\mathcal{N}(\boldsymbol{\mu}^v, \mathbf{S}^v)$ is a distribution representing the local map view, $e_{c,v}\left(\mathbf{I}, \mathbf{0}\right)$ is the error function defined in (9) for identity transformation, and τ_g is a threshold. Depending on the results of this test, views are either added to existing segments or create new ones. Additionally, we store an end-of-life (EOL) counter for every segment. It is initialized with a value of 4 and increased by 2 whenever a new view is added and decreased by 1 whenever the segment is potentially visible but no new view was added. Segments with EOL higher than 8 are treated as mature and their counter is not decreased anymore. When EOL drops to 0, the segment is considered an invalid observation and is removed from the map.

The local map has a limited time horizon of 2 seconds. Such a horizon prevents accumulation of the drift from the visual odometry, yet includes a broader context of a scene than a single frame. As a result of the view-based approach, older information can be easily removed by dropping information about outdated views.

B. POSE RETRIEVAL

The aim of pose retrieval is to compute a pose of the sensor with respect to the global map, given a set of matches between views of planar segments in the local map and ones in the global map. The novel pose retrieval used in this work does so by minimizing an error of fitting virtual points of the first planar segment to a distribution describing the second planar segment. Such formulation allows exploiting uncertainty of depth estimation while also providing a system of linear equations that can be quickly solved. Consider a planar segment from the local map (denoted by a superscript l) and a planar segment from the global map (denoted by a superscript g) described by their centroids $\boldsymbol{\mu}$, covariance matrices $\mathbf{S}$, and plane equations $\boldsymbol{\pi}$. To assess how N transformed points $\mathbf{R}\mathbf{x}_i^l + \mathbf{t}$ forming the local segment distribution fit the global segment distribution $\mathcal{N}(\boldsymbol{\mu}^g, \mathbf{S}^g)$, one can use a squared Mahalanobis distance:

$$\begin{aligned}
&e_{l,g}\left(\mathbf{R}, \mathbf{t}\right) = \\
&= \frac{1}{N}\sum_i (\mathbf{R}\mathbf{x}_i^l + \mathbf{t} - \boldsymbol{\mu}^g)^T (\mathbf{S}^g)^{-1} (\mathbf{R}\mathbf{x}_i^l + \mathbf{t} - \boldsymbol{\mu}^g) \\
&= \frac{1}{N}\sum_i (\mathbf{R}\mathbf{x}_i^l + \mathbf{t} - \boldsymbol{\mu}^g)^T (\mathbf{V}^g \boldsymbol{\Lambda}^g (\mathbf{V}^g)^T)^{-1} \\
&\quad (\mathbf{R}\mathbf{x}_i^l + \mathbf{t} - \boldsymbol{\mu}^g) \\
&= \frac{1}{N}\sum_i (\mathbf{R}\mathbf{x}_i^l + \mathbf{t} - \boldsymbol{\mu}^g)^T (\mathbf{V}^g \boldsymbol{\Lambda}_s^g (\boldsymbol{\Lambda}_s^g)^T (\mathbf{V}^g)^T) \\
&\quad (\mathbf{R}\mathbf{x}_i^l + \mathbf{t} - \boldsymbol{\mu}^g) \\
&= \frac{1}{N}\sum_i \sum_k \left((\mathbf{R}\mathbf{x}_i^l + \mathbf{t} - \boldsymbol{\mu}^g)^T \mathbf{v}_k^g \frac{1}{\sqrt{\lambda_k^g}} \right)^2, \qquad (9)
\end{aligned}$$

where $\mathbf{V}^g \boldsymbol{\Lambda}^g (\mathbf{V}^g)^T$ is an eigen decomposition of the covariance matrix $\mathbf{S}^g$, $\boldsymbol{\Lambda}_s^g$ is a matrix with inverses of square roots of eigenvalues $\frac{1}{\sqrt{\lambda_k^g}}$ on the diagonal, and $\mathbf{v}_k^g$ are columns of the matrix $\mathbf{V}^g$ and eigenvectors of the covariance matrix $\mathbf{S}^g$. To minimize $e_{l,g}\left(\mathbf{R}, \mathbf{t}\right)$, a set of linear equations can be build in the form:

$$(\mathbf{R}\mathbf{x}_i^l + \mathbf{t} - \boldsymbol{\mu}^g)^T \mathbf{v}_k^g \frac{1}{\sqrt{\lambda_k^g}} = 0, \quad (10)$$

and then solved using SVD-based least squares algorithm. Unfortunately, this gives $3N$ equations when using all points from the local segment distribution. Hence, instead of all points, we use virtual points that subsume the distribution:

$$\mathbf{x}_{\pm k}^l = \boldsymbol{\mu}^l \pm K\sqrt{\lambda_k}^l \mathbf{v}_k^l, \quad (11)$$

where K is a number of dimensions used. We use 4 virtual points that correspond to two principal directions ($K = 2$) of the distribution $\mathcal{N}(\boldsymbol{\mu}^l, \mathbf{S}^l)$ projected onto plane $\boldsymbol{\pi}^l$. Those points lay on the plane and are in a distance of two standard deviations from the centroid. Using only points on the plane from the local segment distribution, instead of using 6 points that would represent the distribution before the projection, is of utmost importance to conserve the planar nature of those constraints. If all 6 points were used, and the uncertainty of estimation would be high in a direction of a normal vector of the local segment (i.e. due to poor depth estimation), the fitting error would be high if the global distribution was mainly planar (see Fig. 6). This high fitting error could cause minimization to favor undesired rotations. Moreover, using only 4 points further reduce the number of equations by exploiting the planarity of the segments. Additionally, the centroid and the covariance matrix are computed using stereo estimated depth and give a less accurate description of the geometry than the plane equation from the specialized branch of the DNN. Hence, by projecting distribution on the plane, the accuracy is increased. However, centroids and covariance matrices are still used because they are the only source of uncertainty measures.

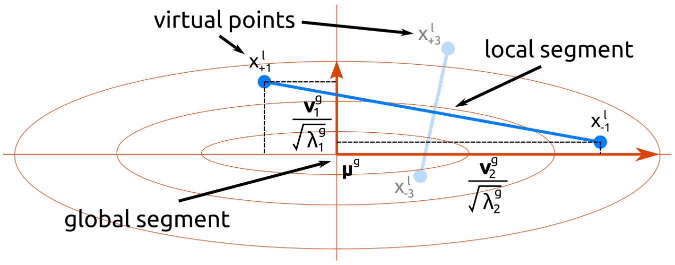

FIGURE 6. Schematic illustration of fitting error of local planar segment (blue) to global planar segment (orange) using 2-D section. Lengths of $\frac{v^g}{\lambda^g}$ vectors correspond to a unit of error. The error is denoted using a dashed line. Using virtual points perpendicular to the plane $\mathbf{x}^l_{-3}$ and $\mathbf{x}^l_{+3}$ could yield high errors and undesired behavior during optimization (see text).

After solving a system of 36 equations (3 pairs of matched segments, 3 dimensions, 4 virtual points) in the form of Eq. (10), we get values of the matrix $\mathbf{R}$ and the vector $\mathbf{t}$. Unfortunately, there are no constraints on the orthonormality of the values in $\mathbf{R}$, so it might not be a valid rotation matrix. To obtain a proper rotation matrix, we perform orthonormalization using the SVD decomposition:

$$\mathbf{R}' = \mathbf{U}\mathbf{V}^T, \tag{12}$$

where $\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T = \mathbf{R}$ is the SVD decomposition. To refine the transformation, a Gauss Newton optimization in the Lie algebra is performed by minimizing a sum of squares of the following residuals:

$$r_{i,k} = (\mathbf{R}\exp(\boldsymbol{\omega})\mathbf{x}^l_i + \mathbf{t} - \boldsymbol{\mu}^g)^T \mathbf{v}^g_k \frac{1}{\sqrt{\lambda^g_k}}, \tag{13}$$

where $\boldsymbol{\omega}$ is a rotation increment. The Jacobians of the residuals are as follows:

$$\frac{\partial r_{i,k}}{\partial \mathbf{t}} = (\mathbf{v}^g_k)^T \frac{1}{\sqrt{\lambda^g_k}} \tag{14}$$

$$\left.\frac{\partial r_{i,k}}{\partial \boldsymbol{\omega}}\right|_{\boldsymbol{\omega}=\mathbf{0}} = (\mathbf{v}^g_k)^T \frac{1}{\sqrt{\lambda^g_k}} \mathbf{R} \left[\mathbf{x}^l_i\right]_\times, \tag{15}$$

where $\left[\mathbf{x}^l_i\right]_\times$ is skew symmetric matrix formed from elements of $\mathbf{x}^l_i$. We can assume that $\boldsymbol{\omega}$ is close to $\mathbf{0}$ because it is an increment. By empirical examination, the number of iterations was set to a constant value of 5. In a vast majority of cases, further iterations do not alter the transformation, whereas using a constant value bounds the execution time. The result is a transformation $(\mathbf{R}, \mathbf{t})$ that stems from the geometric constraints imposed by a set of matched planar segments and is used later to build the PDF of the agent pose. The same procedure is also used to compute the final pose, after the maximum of the PDF was found and all matches were established.

VI. EXPERIMENTAL VERIFICATION

We use a real-world TERRINet dataset[2] to evaluate the proposed solution. The dataset contains trajectories from 3 different scenes with reference poses from Qualisys motion capture system. We recorded stereo images along with Velodyne VLP-16 LiDAR scans that were later used to generate ground truth depth maps for every image. The ground truth depth maps enabled the computation of correspondences between planar segments detected in different image frames.

A. DESCRIPTION

The aim of the first experiment is to show the effectiveness of our new learned descriptors. We compare them with descriptors based on color histograms used in the previous version of PlaneLoc. For every detected planar segment we compute its rank, i.e. the number of nearest neighbors necessary to fetch from the database of all descriptors to include a correct match. We exclude segments from the same trajectory, as images containing them could be very similar to the image of the query segment. To give more insight on the characteristic of descriptors, we present the rank as a function of the size of detected segments. We divided segments based on the square root of their area in pixels, denoted as A, into 6 bins (see Fig. 7). It is clearly visible that the learned descriptors outperform the histogram-based ones by a large margin for all sizes. It is also worth noting that from A equal to 100, the first neighbor is almost always the correct one (values on the box plot further than 1.5 inter-quartile range from the box were treated as outliers and removed).

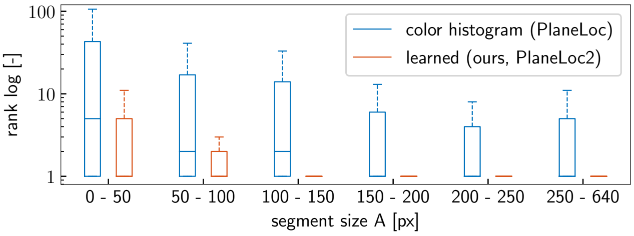

FIGURE 7. Statistics on ranks as a function of the square root of segment size A. Values on the box plot further than 1.5 inter-quartile range from the box were treated as outliers and removed.

B. LOCALIZATION

The second experiment compares the proposed solution with other state-of-the-art global localization systems. As avoiding an incorrect loop closure or relocalization is of utmost importance to the precision of most SLAM systems, we report a percentage of correct and incorrect localization acts (called recognitions hereinafter) and their precision. We compare the pose computed by a considered method with the reference pose and compute the translational and the rotational error. The threshold for assuming a recognition correct is 0.5 m and 10° as an error within such bounds usually enables resuming tracking in SLAM systems [7]. If a method returns no result, we do not compute the errors and treat such outcome as an unknown pose. For each scene, we use one trajectory to build a map and a different one to evaluate localization with a known map. The map is built using the reference poses for

[2] This dataset was collected during the author's visit to LAAS-CNRS in Touluse, within the TERRINet project funded by EU H2020 under GA No.730994

all tested solutions to exclude the factor of map precision. We tested the following solutions:

- `OS3/r` - relocalization mechanism from ORB-SLAM3. The system was forced to relocalize every frame and pose after local map tracking was evaluated if the relocalization was successful. Localization is performed every frame.
- `OS3/m` - map merging mechanism from ORB-SLAM3. A new map was being build for the test trajectory and a transformation between the current map and the prebuilt map was evaluated if a merge was successful. Localization is performed every time a keyframe is inserted into the map.
- `NV+SP` - hierarchical localization [26] with SuperGlue [18] and NetVLAD [25] was evaluated with a global map constructed using COLMAP software[3]. Localization is performed every frame.
- `PL2` (ours) - the solution presented in this paper. The local map is updated every 15 frames because consecutive frames are similar to each other and do not provide diverse views, therefore localization is performed every 15 frames.

Setting proper values of parameters is a troublesome task, especially in complex systems. To facilitate this task in the PlaneLoc2, we follow a data-driven paradigm and use the first scene to perform statistical analysis and compute the values of parameters:

- τ_d - a maximum distance between descriptors that is considered during candidate segment views retrieval. It is set to include 90% of all correct matches.
- $\tau_{svd,t}$ and $\tau_{svd,r}$ - a minimum value of a singular value for translational and rotational part Jacobians in the gradient descent optimization of the pose to assume that the pose is constrained in all dimensions. It is set to include 90% of all correct triplets.
- τ_e - a maximum value of residual error to consider a fitting of planar segments as correct during the pose retrieval. It is set to include 75% of all correct triplets. Value of 75% was used instead of 90% to limit the number of considered triplets and to reduce the computational burden.
- τ_p, τ_r, and τ_d - thresholds that are used during the final safe-checks. They are set to maximize the number of correct matches, while keeping the number of incorrect matches equal to 0. Multiplied by a factor of 1.2, inspired by the Lowe's ratio test [30], to add a safety margin.
- τ_g - threshold used to determine whether two segment observations should be merged (see Eq. (8)) in a map. Empirically set to a value of 2 that prevents most of the incorrect data associations.

To enable a fair comparison, for ORB-SLAM3 we used the parameter setting designed by the authors and used in the EuRoC indoor experiments [4]. Likewise, for NV+SP we used parameters set for the InLoc dataset [26] that is similar in characteristic to the TERRINet dataset.

TABLE 1. Results of global localization on TERRINet dataset. Cases with incorrect recognitions are colored red. The best correct recognitions rates for cases without incorrect recognitions are emboldened.

scene	measure	method			
		`OS3/r`	`OS3/m`	`NV+SP`	`PL2`
02	correct [%]	58.1	40.3	95.0	**73.8**
	incorrect [%]	0.0	0.0	5.0	0.0
	unknown [%]	41.9	59.6	0.0	26.2
	mean error lin. [m]	0.08	0.09	0.10	0.09
	mean error ang. [°]	0.7	0.7	1.9	1.1
	max. error lin. [m]	0.27	0.17	18.35	0.37
	max error ang. [°]	5.3	1.7	156.3	3.2
03	correct [%]	49.4	18.2	94.3	**47.9**
	incorrect [%]	0.2	0.0	5.7	0.0
	unknown [%]	50.4	81.8	0.0	52.1
	mean error lin. [m]	0.06	0.04	0.20	0.10
	mean error ang. [°]	1.5	0.7	2.9	1.1
	max. error lin. [m]	0.81	0.09	12.88	0.38
	max error ang. [°]	10.5	1.2	174.8	3.2

Quantitative results are gathered in Tab. 1, while visualization of results for scene 02 are presented in Fig. 8. Both ORB-SLAM3 mechanisms, relocalization and map merging, recognize a lower percentage of poses than our solution. An exception is scene 02, where relocalization recognized slightly more poses, but also yielded incorrect ones. The NV+SP recognized a higher percentage of poses but also produced many incorrect ones, some of which were distant more than 18 m from the reference pose. Such behavior can be attributed to a lack of fail-safe checks that inevitably reject some of the correct recognitions, but also prevent incorrect ones. Thus, our system recognized the highest percentage of poses among cases where no incorrect results were produced. Moreover, our system did not produce any incorrect recognitions in all test cases.

The accuracy of all tested methods is similar, with mean error values varying slightly on different scenes. Maximum errors depend mainly on incorrect recognitions and are the lowest for the ORB-SLAM3 map merging mechanism, while being below 0.4 m and 3.5° for the PlaneLoc2.

VII. CONCLUSION

In this article, we present the PlaneLoc2 global localization method that utilizes a passive stereo camera to detect planar segments and compute a PDF of the 6-D pose. The method uses a DNN that jointly detects planar segments, describes their appearance, and estimates their geometry. The detected segments are used to build view-based local and global maps, that are easily manageable and store information about the uncertainty of geometry of planar segments. The uncertainty is exploited in a novel pose retrieval procedure that is designed with stereo sensors in mind. In the experimental section, we show that the new learned appearance descriptor outperforms the classic, based on color histograms one. We also tested the global localization performance of our system and show that it achieves the best percentage of

[3] https://colmap.github.io

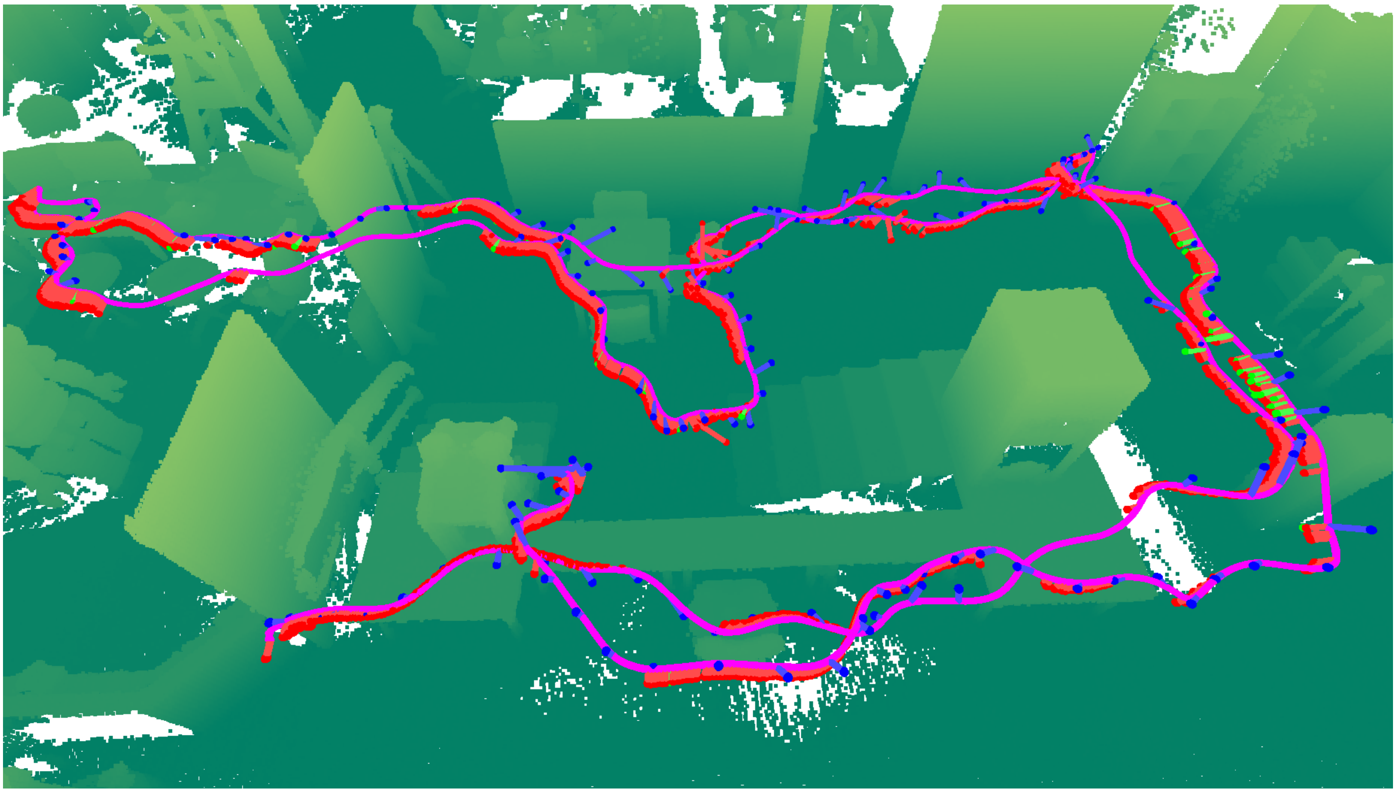

FIGURE 8. Visualization of results with reference trajectory (magenta line), ORB-SLAM3 relocalization poses (red points), ORB-SLAM3 map merge poses (green points), and PlaneLoc2 poses (blue points). Lines of corresponding colors connect reference poses with computed poses. Points in the point cloud are colored according to their height above the ground. Results for NV+SP were omitted for clarity.

recognized poses, when cases without incorrect recognitions are considered (15.7% more poses in the first scene than the second best solution and 29.7% more poses in the second scene). Moreover, the PlaneLoc2 did not produce incorrect recognitions in all cases, which is of pivotal importance in navigation and SLAM systems, proving its suitability as a global localization system.

The most important changes, with respect to the previous version of PlaneLoc, that helped achieve good results include the new appearance descriptor. Results in Sec. VI-A suggest that it significantly limits the number of incorrect potential matches. Additionally, considering geometric constraints from distant segments enabled correct pose retrieval in higher percentage of situations. The new pose retrieval procedure that accommodates the spatial boundaries of planar segments further increases the number of geometric constraints available. All those factors facilitate a high correct recognition rate without incorrect recognitions.

As a part of the future work, we plan to expand the system with other types of geometric features, such as edges. Edges could provide additional constraints that are unused in this version of the system.

REFERENCES

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*, ser. Intelligent robotics and autonomous agents. Cambridge: MIT Press, 2006.

[2] C. Liu, K. Kim, J. Gu, Y. Furukawa, and J. Kautz, “PlaneRCNN: 3D plane detection and reconstruction from a single image,” in *IEEE Conf. on Comp. Vis. and Pattern Recog. (CVPR)*, 2019, pp. 4445–4454.

[3] S. Yang and S. Scherer, “Monocular Object and Plane SLAM in Structured Environments,” *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3145–3152, Oct. 2019.

[4] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, “ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM,” *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[5] J. Wietrzykowski and D. Belter, “Stereo Plane R-CNN: Accurate Scene Geometry Reconstruction Using Planar Segments and Camera-Agnostic Representation,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4345–4352, 2022.

[6] J. Wietrzykowski and P. Skrzypczyński, “On the descriptive power of LiDAR intensity images for segment-based loop closing in 3-D SLAM,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 79–85.

[7] ——, “PlaneLoc: Probabilistic global localization in 3-D using local planar features,” *Robotics and Autonomous Systems*, vol. 113, pp. 160–173, 2019.

[8] G. Halmetschlager-Funek, M. Suchi, M. Kampel, and M. Vincze, “An Empirical Evaluation of Ten Depth Cameras: Bias, Precision, Lateral Noise, Different Lighting Conditions and Materials, and Multiple Sensor Setups in Indoor Environments,” *IEEE Robotics & Automation Magazine*, vol. 26, no. 1, pp. 67–77, Mar. 2019.

[9] I. C. Condotta, T. M. Brown-Brandl, S. K. Pitla, J. P. Stinn, and K. O. Silva-Miranda, “Evaluation of low-cost depth cameras for agricultural applications,” *Computers and Electronics in Agriculture*, vol. 173, p. 105394, 2020.

[10] P. Hausamann, C. B. Sinnott, M. Daumer, and P. R. MacNeilage, “Evaluation of the Intel RealSense T265 for tracking natural human head motion,” *Scientific Reports*, vol. 11, no. 1, p. 12486, Dec. 2021.

[11] K. Chappellet, G. Caron, F. Kanehiro, K. Sakurada, and A. Kheddar, “Benchmarking Cameras for Open VSLAM Indoors,” in *2020 25th International Conference on Pattern Recognition (ICPR)*. Milan, Italy: IEEE, Jan. 2021, pp. 4857–4864.

[12] M. Senthilvel, R. K. Soman, and K. Varghese, "Comparison of handheld devices for 3d reconstruction in construction," in *Proceedings of the 34th International Symposium on Automation and Robotics in Construction (ISARC)*, M.-Y. N. T. U. o. S. Cheng, Technology), H.-M. N. T. U. o. S. Chen, Technology), K. C. N. T. U. o. S. Chiu, and Technology), Eds. Taipei, Taiwan: Tribun EU, s.r.o., Brno, July 2017, pp. 698–705.

[13] I. Cvišić, J. Ćesić, I. Marković, and I. Petrović, "SOFT-SLAM: Computationally efficient stereo visual simultaneous localization and mapping for autonomous unmanned aerial vehicles," *Journal of Field Robotics*, vol. 35, no. 4, pp. 578–595, 2018.

[14] N. Smolyanskiy, A. Kamenev, and S. Birchfield, "On the importance of stereo for accurate depth estimation: An efficient semi-supervised deep neural network approach," in *CVPR 2018 Workshop on Autonomous Driving*, Salt Lake City, 2018.

[15] D. Gálvez-López and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, October 2012.

[16] X. Li, K. Han, S. Li, and V. Prisacariu, "Dual-Resolution Correspondence Networks," in *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[17] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-Supervised Interest Point Detection and Description," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Salt Lake City, UT, USA: IEEE, Jun. 2018, pp. 337–33 712.

[18] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning Feature Matching With Graph Neural Networks," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, Jun. 2020, pp. 4937–4946.

[19] P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl, and T. Sattler, "Back to the Feature: Learning Robust Camera Localization from Pixels to Pose," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, pp. 3246–3256, iSSN: 2575-7075.

[20] K. Ćwian, M. R. Nowicki, J. Wietrzykowski, and P. Skrzypczyński, "Large-Scale LiDAR SLAM with Factor Graph Optimization on High-Level Geometric Features," *Sensors*, vol. 21, no. 10, p. 3445, May 2021.

[21] X. Zhang, W. Wang, X. Qi, Z. Liao, and R. Wei, "Point-Plane SLAM Using Supposed Planes for Indoor Environments," *Sensors*, vol. 19, no. 17, p. 3795, Sep. 2019.

[22] Y. Taguchi, Y.-D. Jian, S. Ramalingam, and C. Feng, "Point-plane SLAM for hand-held 3D sensors," in *2013 IEEE International Conference on Robotics and Automation*. Karlsruhe, Germany: IEEE, May 2013, pp. 5182–5189.

[23] E. Fernandez-Moral, W. Mayol-Cuevas, V. Arevalo, and J. Gonzalez-Jimenez, "Fast place recognition with plane-based maps," in *2013 IEEE International Conference on Robotics and Automation*. Karlsruhe, Germany: IEEE, May 2013, pp. 2719–2724.

[24] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii, "InLoc: Indoor Visual Localization with Dense Matching and View Synthesis," *arXiv:1803.10368 [cs]*, Apr. 2018, arXiv: 1803.10368.

[25] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN Architecture for Weakly Supervised Place Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1437–1451, 2018.

[26] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From Coarse to Fine: Robust Hierarchical Localization at Large Scale," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, Jun. 2019, pp. 12 708–12 717.

[27] Z. Teed and J. Deng, "DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras," *Advances in Neural Information Processing Systems*, 2021.

[28] M. Xu, N. Snderhauf, and M. Milford, "Probabilistic Visual Place Recognition for Hierarchical Localization," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 311–318, Apr. 2021, conference Name: IEEE Robotics and Automation Letters.

[29] R. Steiner, M. Cox, P. V. K. Borges, L. Bernreiter, and J. Nieto, "Certainty Aware Global Localisation Using 3D Point Correspondences," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8710–8717, 2021.

[30] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, p. 91–110, nov 2004.

JAN WIETRZYKOWSKI graduated from Poznan University of Technology in 2015. He received BSc and MSc in Automatic Control and Robotics from the same university in 2014 and 2015, respectively. Since 2015 he is a PhD student at the Faculty of Control, Robotics, and Electrical Engineering. In 2016 he became a research assistant at the Institute of Robotics and Machine Intelligence. He is the author or coauthor of multiple technical papers in the area of robotics and machine learning, including ICRA and IROS conference papers, and RAS and RAL journal papers. His current research interests include robotic global localization, machine learning, and simultaneous localization and mapping.

• • •

Chapter 3

Conclusions

3.1 Summary

This thesis presents a research on global localization methods that are suitable for indoor environments and use planar segments as objects of reference. The author proposed novel methods for solving tasks in the processing pipeline of a global localization system that enabled outperforming the existing state-of-the-art systems. The novel methods are incorporated in the PlaneLoc2 – a complete global localization system that is the main contribution of this work. Innovations and improvements in particular areas are summarized in the following subsections.

3.1.1 Planar segment representation

This dissertation considers multiple ways of representing the geometry of planar segments. The first one is a representation using infinite planes that support the segment. Infinite planes are not spatially bounded, as opposed to segments themselves, therefore they do not reflect all properties of segments. Nonetheless, they provide straightforward means to constrain the pose of the agent using current observations of planar segments and observations of planar segments stored in a global map. This work evaluates two representations of infinite planes in the context of pose optimization [Wietrzykowski 2016]. The first one is a $SE(3)$ transformation with a properly designed covariance matrix that delivers information about non-constrained directions. The second one is a quaternion-based minimal representation that encodes all parameters of the plane. Experiments suggest that the minimal representation performs better when constraints are weak, but otherwise it performs similarly. However, further research revealed that the representation using infinite planes is not stable, i.e. the values can change significantly when 3-D points move due to noise, especially when the frame of reference is far from the observed segment [Wietrzykowski and Skrzypczyński 2019]. To mitigate this issue, a new metric of distance between planar segments is proposed in [Wietrzykowski and Skrzypczyński 2019] (see Eq. (5) in that article). The metric is based on point-to-plane distance and utilizes a covariance matrix and a centroid vector of the point cloud forming the segment. The covariance matrix and the centroid vector are efficiently updated when new observations are available, making it computationally feasible. Unfortunately, this metric is not suitable for global pose retrieval, because minimization equations are fourth degree polynomials and cannot be easily solved. Therefore, a new metric is proposed in [Wietrzykowski 2022] (see Eq. (9) in that article) that avoids this problem by using virtual points. The virtual points subsume the distribution represented in the previous metric by a covariance matrix and a centroid vector, making it possible to form quadratic equations. Moreover, this metric accommodates the uncertainty of depth measurements and spatial bounds, making it possible to exploit more constraints. It is used in the final version of the global localization system [Wietrzykowski 2022] to recover the pose and to assess whether two observations are observations of the same planar segment.

3.1.2 Planar segment detection

Methods of planar segment detection vary depending on the sensor used, and this thesis describes methods suitable for two different sensors. Earlier experiments employed an RGB-D sensor because of the readily available depth measurements that enable accurate geometry estimation. To increase the field of view during localization, a number of consecutive RGB-D frames are fused together using the ElasticFusion [Whelan et al. 2015] in [Wietrzykowski and Skrzypczyński 2017]. The result is an unorganized point cloud representing the current local view. To extract planar segments from such a point cloud, a purely geometric method based on supervoxel clustering and flood fill algorithm is proposed. However, this approach is dependent on an external system, is slow, and ignores the cues contained in images. Hence, a new plane detection method is presented in [Wietrzykowski and Skrzypczyński 2019] that detects segments in an organized point cloud built from a single RGB-D frame. In this way, the incidence relations of an organized point cloud can be exploited to speed up the computations and image cues can be used to enhance the segmentation. To maintain a broader context, planar segments detected in a number of frames are merged to build a local map. Despite providing valuable information about depth, RGB-D sensors have a limited effective range. Therefore, a stereo camera is proposed to detect planar segments in [Wietrzykowski and Belter 2022]. Because stereo-estimated depth is not accurate enough to use the aforementioned methods to detect planar segments, a DNN-based solution is introduced. The solution detects planar segments in the image using a module based on the Plane R-CNN [Liu et al. 2019] and recovers the geometry of planar segments using a novel geometry module. As a result of modifying the segmentation procedure used to label the training examples to better suit the DNN architecture, the detection results are improved compared to the baseline solution. Moreover, geometry reconstruction of the detected planar segments is enhanced by introduction of a geometry module that builds a cost volume from a pair of stereo images and estimates the normal vectors using a camera-agnostic representation. The camera-agnostic representation facilitates training and improves robustness to camera parameter changes. This solution is used in the PlaneLoc2 [Wietrzykowski 2022] system that concludes this research.

3.1.3 Map building and map management

Two versions of map building and map management methods are presented in this work. The first one is introduced in [Wietrzykowski and Skrzypczyński 2019] and designed to work with planar segments detected using an RGB-D sensor. When newly detected segments are processed, they are either added as new segments, merged with the existing segments, or placed in a delayed merge queue. The delayed merge queue prevents the incorrect merging of two or more existing segments when a newly detected segment is matched to more than one segment. When enough evidence is available, delayed merges are executed. The merging itself creates a single representation of a segment by fusing the point clouds of the merged segments, updating the covariance matrix and the centroid vector, updating the plane equation, and recomputing the hull. Additionally, a mechanism for removing incorrect observations using an end-of-life counter is introduced that keeps the number of planar segments in the map at a reasonable level. This mechanism creates a depth buffer for every new frame to check which segments should be visible. When there are multiple times a segment should be visible but no new observations of this segment are available, it is removed from the map. Unfortunately, the PlaneLoc2 uses a passive stereo camera instead of an RGB-D sensor, therefore a new map building method is presented in [Wietrzykowski 2022]. Point clouds from stereo estimated depth are less accurate than those from an RGB-D sensor and explicit merging would cause quick degeneration. Hence, a view-based map is introduced that stores information about different views of planar segments, instead of explicitly merging their representations. Additionally, the view-based map preserves uncertainty estimates that are otherwise lost in the merging process. It is worth noting that the uncertainty estimates are essential for the new pose retrieval algorithm in the PlaneLoc2. These view-based map building and map management methods are incorporated in the final version of the global localization system [Wietrzykowski 2022].

3.1.4 Appearance descriptor

This thesis also makes an effort to research the appearance descriptors that are used to find candidate matches from the global map of planar segments. The PlaneLoc [Wietrzykowski and Skrzypczyński 2019] uses simple appearance descriptors based on color histograms. The descriptor is a normalized and concatenated histogram of H and S color components from the HSV space. It reduces significantly the number of potential match candidates, but is susceptible to changing lighting conditions and does not include information about surrounding of the segment. Therefore, a new, learned descriptor is presented in [Wietrzykowski 2022] that leverages the potential of DNNs. By a proper modification of the DNN used to detect planar segments [Wietrzykowski and Belter 2022], a descriptor is produced with a little overhead during the detection phase. The new descriptor is compared with the histogram-based one, and the experimental results show it is significantly more discriminative. For planar segments bigger than 10000 pixels, the nearest neighbor in the descriptor space is almost always a correct match. As a result, it is enough to consider only two match candidates during global localization in the PlaneLoc2 [Wietrzykowski 2022] when these descriptors are used, which reduces the computational burden.

3.1.5 Pose retrieval

Pose retrieval is a vital part of a global localization system. It computes the pose of the agent given a set of observations matched between the local view and the global map. Two different pose retrieval algorithms are described in this thesis. An algorithm exploiting the equations of planes supporting planar segments is introduced in [Wietrzykowski and Skrzypczyński 2017]. It is a two-phase algorithm, where orientation is computed in the first phase using normal vectors, and translation is computed in the second phase assuming a fixed rotation. If all matches are correct and the pose is constrained in all dimensions, the algorithm produces accurate results. However, in the presence of noise, when plane equations are not accurately estimated, it may produce implausible solutions due to the assumption that planes are infinite. The assumption about infinite planes also leads to rejection of constraints stemming from boundaries of planar segments. These issues are addressed in [Wietrzykowski 2022], where a new pose retrieval method is introduced. The new method fits the virtual points representing the distribution of points of planar segments from the global map to the distribution of planar segments from the local map. After the initial phase of solving a system of linear equations, an orthonormalization is performed, followed by a gradient descent optimization to refine the pose. This method is also capable of accommodating the uncertainty of depth estimation, making it suitable for global localization using a passive stereo camera. It significantly contributes to the performance of the PlaneLoc2 [Wietrzykowski 2022] that is the final version of the global localization solution presented in this thesis.

3.1.6 Inference

This dissertation presents a novel pose inference method introduced in [Wietrzykowski and Skrzypczyński 2017]. As a result of using planar segments as reference objects, it is possible to consider all plausible minimal sets of matches. The final pose is the one supported by a weighted majority of Gaussian kernels representing the pose solutions stemming from minimal sets of matches. All kernels form a 6-D PDF describing the belief about the agent's whereabouts. Thanks to considering all plausible hypotheses, this inference method avoids the problems with stopping criteria occurring in RANSAC. Moreover, having a PDF incorporating all cues, it is possible to detect whether two distinct locations have high probability and reject such inconclusive results to avoid recognizing an incorrect pose [Wietrzykowski and Skrzypczyński 2017]. The same inference algorithm is used in the PlaneLoc2 [Wietrzykowski 2022], modified to exploit planar segments detected using a passive stereo camera.

3.1.7 Datasets

Three datasets have been developed during the research presented in this thesis that enable experimental verification of the solutions and facilitate training of the DNN. The datasets are made publicly available to benefit the community and to enable reproduction of the presented results[1]. The `PUT RGB-D/Workshop` is a real-world dataset collected in a workshop that includes RGB-D frames, AHRS measurements, and reference pose information from the Optitrack motion capture system. It is used in [Wietrzykowski and Skrzypczyński 2017, 2019] to evaluate the pose recognition performance. Another real-world dataset is the `TERRINet` dataset that presents a larger workshop environment with three different wall and furniture settings. It contains 16k images from a passive stereo camera, AHRS measurements, scans from Velodyne VLP-16 LiDAR, and reference poses from the Qualisys motion capture system. Using the reference poses and the LiDAR scans, ground truth depth maps are generated for every image. This dataset is used to evaluate the geometry reconstruction performance of Stereo Plane R-CNN [Wietrzykowski and Belter 2022] and the pose recognition performance of the PlaneLoc2 [Wietrzykowski 2022]. The last one is the synthetic SceneNet Stereo dataset generated using modified SceneNet RGB-D framework [McCormac et al. 2017]. The dataset includes stereo camera images, reference depth maps, reference normal vector maps, and reference pose information. Being the outcome of a rendering process, the reference information is accurate and complete. As a result, this synthetic dataset is used to train the DNN in [Wietrzykowski and Belter 2022; Wietrzykowski 2022]. It also enabled evaluating the detection performance and the robustness to camera parameter changes of the Stereo Plane R-CNN.

3.2 Impact

The thesis of this dissertation is stated in Ch. 1 and is restated here for convenience of the reader: *Local, partial, and uncertain cues from planar segment features allow to build a probability density function describing the global metric pose of an agent in a man-made environment.* Auxiliary theses are as follows:

- By considering many small sets of local geometric features in kernel density estimation it is possible to build a function with the maximum corresponding to the global pose of an agent.
- Observations of planar segments in man-made environments enable to determine the pose of an agent with six degrees of freedom with respect to a predefined map of planar segments.
- Deep neural network facilitates detection and description of planar segments using a passive stereo camera.

The research presented in this dissertation is concluded by the PlaneLoc2 system [Wietrzykowski 2022] that is experimentally shown to be capable of building a PDF describing the global metric pose of an agent in a man-made environment. The system uses small sets of planar segments matched between the local map and the global map, that can be viewed as local, partial, and uncertain cues. They are local, because they occupy only a fragment of the scene. They are also partial, because not all available segments are matched. And finally, they are uncertain, because geometry estimation is subject to uncertainty, which proves the main thesis. The PDF

[1] Implementation of PlaneLoc2 and datasets are available at `https://github.com/LRMPUT/plane_loc_2` and `http://lrm.put.poznan.pl/rgbdw/`

is constructed using kernel density estimation and searched to find a maximum that becomes a candidate for the 6-D pose of an agent, expressed with respect to a predefined map of planar segments. If the candidate pose passes fail-safe checks, it is assumed to be the true pose of an agent. The experiments showed that the PlaneLoc2 has a high recognition rate and does not return incorrect poses, which proves the first and the second auxiliary thesis. Moreover, planar segments are detected using a novel DNN that suits the characteristic of passive stereo cameras and achieves good detection and geometry reconstruction performance, which proves the third auxiliary thesis.

The global localization system presented here can be used in multiple scenarios. The scenarios include, but are not limited to, medical care assistant robots, shopping mall customer service robots, industrial facility inspection robots, and indoor mapping gear. In all life-long autonomy scenarios the global localization ability is essential to solve loop closures and the kidnapped robot problem [Thrun et al. 2005]. The PlaneLoc2 can be integrated with the existing SLAM and navigation solutions to provide global 6-D pose that facilitates these tasks.

3.3 Future work

The presented solution can be extended and enhanced in many ways. The author plans on investigating the possibility of exploiting other types of geometric features, such as edges. Edges can add valuable pose constraints that are unavailable when using planar segments only. Another interesting direction of research is the use of sets of matches that do not fully constrain the pose. An example could be when an agent is in a long corridor without doors and there are no features that constrain the pose along the corridor. As a result, a PDF in these cases could also be built and partial global localization could be performed. Also partial information about orientation from an AHRS sensor could be a valuable extension to the system. The AHRS information can be incorporated in the PDF and additionally constrain the pose. It can be useful when the orientation is weakly constrained by planar segments or when there are two similar locations that differ in orientation.

Bibliography

G. Bresson, Z. Alsayed, L. Yu, and S. Glaser. Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving. *IEEE Transactions on Intelligent Vehicles*, 2(3): 194–220, 2017. doi: 10.1109/TIV.2017.2749181.

C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021. doi: 10.1109/TRO.2021.3075644.

D. DeTone, T. Malisiewicz, and A. Rabinovich. SuperPoint: Self-Supervised Interest Point Detection and Description. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 337–33712, Salt Lake City, UT, USA, June 2018. IEEE. ISBN 978-1-5386-6100-0. doi: 10.1109/CVPRW.2018.00060.

R. Dubé, A. Cramariuc, D. Dugas, H. Sommer, M. Dymczyk, J. Nieto, R. Siegwart, and C. Cadena. SegMap: Segment-based mapping and localization using data-driven descriptors. *International Journal of Robotics Research*, 39(2-3):339–355, 2020.

E. Fernandez-Moral, W. Mayol-Cuevas, V. Arevalo, and J. Gonzalez-Jimenez. Fast place recognition with plane-based maps. In *2013 IEEE International Conference on Robotics and Automation*, pages 2719–2724, Karlsruhe, Germany, May 2013. IEEE. ISBN 978-1-4673-5643-5 978-1-4673-5641-1. doi: 10.1109/ICRA.2013.6630951.

D. Gálvez-López and J. D. Tardós. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, October 2012. ISSN 1552-3098. doi: 10.1109/TRO.2012.2197158.

G. Halmetschlager-Funek, M. Suchi, M. Kampel, and M. Vincze. An Empirical Evaluation of Ten Depth Cameras: Bias, Precision, Lateral Noise, Different Lighting Conditions and Materials, and Multiple Sensor Setups in Indoor Environments. *IEEE Robotics & Automation Magazine*, 26(1):67–77, Mar. 2019. ISSN 1070-9932, 1558-223X. doi: 10.1109/MRA.2018.2852795.

K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. doi: 10.1109/ICCV.2017.322.

M. Kaess. Simultaneous localization and mapping with infinite planes. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4605–4611, 2015. doi: 10.1109/ICRA.2015.7139837.

U. Kusupati, S. Cheng, R. Chen, and H. Su. Normal assisted stereo depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2186–2196, 2020. doi: 10.1109/CVPR42600.2020.00226.

R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. G^2o: A general framework for graph optimization. In *2011 IEEE International Conference on Robotics and Automation*, pages 3607–3613, 2011. doi: 10.1109/ICRA.2011.5979949.

X. Li, K. Han, S. Li, and V. Prisacariu. Dual-Resolution Correspondence Networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

C. Liu, K. Kim, J. Gu, Y. Furukawa, and J. Kautz. PlaneRCNN: 3D plane detection and reconstruction from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4445–4454, 2019. doi: 10.1109/CVPR.2019.00458.

D. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2, 1999. doi: 10.1109/ICCV.1999.790410.

J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison. SceneNet RGB-D: Can 5M Synthetic Images Beat Generic ImageNet Pre-Training on Indoor Segmentation? In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

R. Mur-Artal and J. D. Tardós. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017. doi: 10.1109/TRO.2017.2705103.

E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An efficient alternative to SIFT or SURF. In *2011 International Conference on Computer Vision*, pages 2564–2571, 2011. doi: 10.1109/ICCV.2011.6126544.

R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison. SLAM++: Simultaneous Localisation and Mapping at the Level of Objects. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1352–1359, 2013. doi: 10.1109/CVPR.2013.178.

P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk. From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12708–12717, Long Beach, CA, USA, June 2019. IEEE. ISBN 978-1-72813-293-8. doi: 10.1109/CVPR.2019.01300.

F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015. doi: 10.1109/CVPR.2015.7298682.

N. Smolyanskiy, A. Kamenev, and S. Birchfield. On the Importance of Stereo for Accurate Depth Estimation: An Efficient Semi-Supervised Deep Neural Network Approach. In *CVPR 2018 Workshop on Autonomous Driving*, Salt Lake City, 2018.

N. Sünderhauf, O. Brock, W. Scheirer, R. Hadsell, D. Fox, J. Leitner, B. Upcroft, P. Abbeel, W. Burgard, M. Milford, and P. Corke. The limits and potentials of deep learning for robotics. *The International Journal of Robotics Research*, 37(4-5):405–420, 2018.

Y. Taguchi, Y.-D. Jian, S. Ramalingam, and C. Feng. Point-plane SLAM for hand-held 3D sensors. In *2013 IEEE International Conference on Robotics and Automation*, pages 5182–5189, Karlsruhe, Germany, May 2013. IEEE. ISBN 978-1-4673-5643-5 978-1-4673-5641-1. doi: 10.1109/ICRA.2013.6631318.

S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. ISBN 0262201623.

T. Whelan, S. Leutenegger, R. S. Moreno, B. Glocker, and A. Davison. ElasticFusion: Dense SLAM Without A Pose Graph. In *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015. doi: 10.15607/RSS.2015.XI.001.

J. Wietrzykowski. On the Representation of Planes for Efficient Graph-based SLAM with High-level Features. *Journal of Automation, Mobile Robotics and Intelligent Systems*, 10(03):3–11, 2016. doi: DOI:10.14313/JAMRIS_3-2016/18.

J. Wietrzykowski. PlaneLoc2: Indoor global localization using planar segments and passive stereo camera. *IEEE Access*, 2022. doi: 10.1109/ACCESS.2022.3185732. © 2022 IEEE. Reprinted with permission.

J. Wietrzykowski and D. Belter. Stereo Plane R-CNN: Accurate Scene Geometry Reconstruction Using Planar Segments and Camera-Agnostic Representation. *IEEE Robotics and Automation Letters*, 7(2):4345–4352, 2022. doi: 10.1109/LRA.2022.3150841. © 2022 IEEE. Reprinted with permission.

J. Wietrzykowski and P. Skrzypczyński. A probabilistic framework for global localization with segmented planes. In *2017 European Conference on Mobile Robots (ECMR)*, pages 1–6, 2017. doi: 10.1109/ECMR.2017.8098672. © 2017 IEEE. Reprinted with permission.

J. Wietrzykowski and P. Skrzypczyński. PlaneLoc: Probabilistic global localization in 3-D using local planar features. *Robotics and Autonomous Systems*, 113:160–173, 2019. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2019.01.008.

J. Wietrzykowski and P. Skrzypczyński. On the descriptive power of LiDAR intensity images for segment-based loop closing in 3-D SLAM. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 79–85, 2021. doi: 10.1109/IROS51168.2021.9636698. © 2021 IEEE. Reprinted with permission.

World Health Organization. *Global perspectives on assistive technology: proceedings of the GReAT Consultation 2019, World Health Organization, Geneva, Switzerland, 22–23 August 2019. Volume 2.* World Health Organization, 2019.