Preprint of a publication at Towards Autonomous Robotic Systems (TAROS) 2017. The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-64107-2_49

Towards camera based navigation in 3D maps by synthesizing depth images

Stefan Schubert, Peer Neubert, and Peter Protzel

TU Chemnitz, 09126 Chemnitz, Germany
{stefan.schubert,peer.neubert,peter.protzel}@etit.tu-chemnitz.de

Abstract. This paper presents a novel approach to localize a robot equipped with an omnidirectional camera within a given 3D map. The pose estimate builds upon the synthesis of panoramic depth images, which are compared to the current view of the camera. We present an algorithmic approach to compute the similarity between these synthetic depth images and visual images, and show how to utilize this image matching for mobile robot navigation tasks, i.e. heading estimation, global localization, and navigation towards a target position. The presented method requires neither additional colour nor laser intensity information in the map. We provide a first evaluation of the involved image processing pipeline and a set of proof-of-concept experiments on a mobile robot. The presented approach supports different use cases like map sharing for heterogeneous robotics teams, or the usage of external sources of 3D maps like extruded floor plans.

Keywords: camera-based localization, visual compass, visual homing, 3D map, omnidirectional camera

1 Introduction

In this paper, we present a novel approach to combine the advantages of powerful sensors that create 3D maps and cheap and lightweight cameras for mobile robot localization. Prerequisite is a 3D map of the world which has to be given or built in advance, and a camera which is moving around in this world/map (in our experiments, we use a panoramic camera). The map could be built by a 3D laserscanner, which makes this world representation very accurate, and gives metric information for an exact and global localization, however, the map could also be given by other sources like computer models, etc.. Once the map is available, a camera which is cheap, light, and with potentially high frame rate moves around in this world. The 3D map is used as a world model which enables the system to synthesize images at arbitrary positions. Subsequently, in order to determine the current position in the world, the synthesized images are compared to the current visual camera view. We want to emphasize that our approach does not require additional information in the 3D map, i.e., there is no need to add colour or intensity information to the 3D points.

Our proposed system borrows a biological inspiration from a theory about the navigation mechanisms of the dessert ant [1]. Since the heat in the dessert

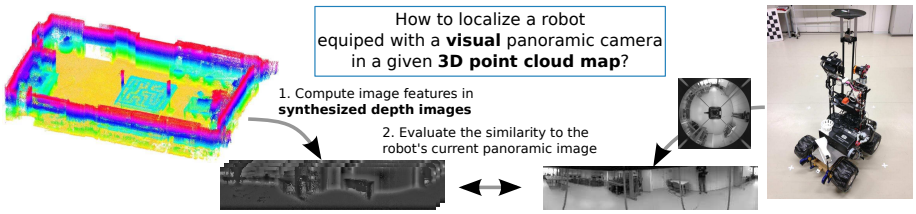

Fig. 1. We present an approach to localize a robot with a panoramic camera in a known 3D point cloud map based on synthesized depth images.

prevents the ant from leaving a pheromone trace, desert ants developed an alternative method to find back to their nest: Before they leave their nest's location, they first take a 360° snapshot of their surroundings as a *home view*. Then, they walk around for foraging. When they want to find back to their nest, they start to acquire new 360° views of their current position. Such current views are then compared to their *home view* which they still remember. By relating both images' content, they can then finally determine a rough home direction. Details on a technical realization of the *ant algorithm* on mobile robots are provided by Möller [2]. His algorithm can be used to implement, e.g., a visual compass, visual homing, visual teach & repeat, or exploration.

Fig. 1 illustrates our key idea to adapt this biologically plausible theory of visual navigation for cross sensor modality localization. We replace the ant's *home views* with depth images that are synthesized from a 3D map. We further provide an image processing pipeline, that can match these depth images to the current visual camera view of the robot. Based on this, we are able to solve navigation tasks, e.g., to determine the motion direction ("home direction") to arbitrary positions in the 3D map. In contrast to the dessert ant's approach, there is no need that we visited this place beforehand - all we need is the 3D information.

There are several use-cases for such a system: For instance, a heterogeneous robotics team consisting of one big robot equipped with a heavy (and expensive) 3D laserscanner, and one or more small robots (solely) equipped with cameras. The big robot maps the world and thus defines a reference frame for all small robots which can then manoeuvre afterwards or in parallel to the big robot. Moreover, if the scan rate of the big robot is low (i.e., if the robot has to stop to acquire a new scan [3]), the proposed system can be used to localize the robot between consecutive scans. In contrast to visual odometry, our approach is anchored to the previous laserscan which prevents drift. Another use-case of the proposed system is to preset the 3D map from other sources like CAD models, floor plans, or even from an extruded hand-drawn sketch.

In this paper, we

– present a novel approach for a camera-based localization in 3D maps that might be useful for a variety of navigation problems. As the approach builds upon synthesized depth images, an additional data augmentation with colour or intensity is not necessary.

– show a proof-of-concept implementation for a set of robot navigation tasks.
– provide preliminary experiments which first evaluate the visual pipeline for the comparison of colour and depth images, and second shows its potential for navigation tasks.

In the next section, we give an overview over the related work. The subsequent section 3 gives then a detailed explanation about how to realize our proposed approach from an algorithmic point of view. Then, proof-of-concept experiments are shown in section 4 which prove the applicability of our proposed approach. Finally, a conclusion and a discussion of future work are given in section 5.

2 Related Work

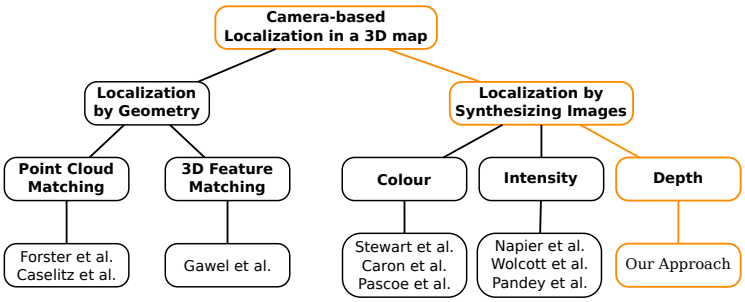

Fig. 2. Taxonomy of the camera-based localization approach in 3D maps including the corresponding related work.

Fig. 2 provides a coarse taxonomy of approaches to camera based localization in 3D maps. A prerequisite for the presented problem is a given or previously built map. In [4] the map is given as a textured model, however, in most cases this map is rather generated in advance by active depth sensors like laserscanners or RGB-D cameras [5–13]. Furthermore, this geometrical data can be enhanced either by intensity or colour information. For the most common case of using a laserscanner for map generation, intensity data can be acquired directly with the range measurements as reflectance information [12–14], whereas colour information has to be added by additional sensors like mono- or omni-cameras [4, 9–11]. It should be mentioned that RGB-D or stereo-camera could also be used as they measure the range and colour of an obstacle concurrently, however, this approach is less suited due to their low range and bad accuracy compared to laserscanners.

As soon as an initial map is present, the goal is to localize a system equipped with a camera within this 3D map. Basically, there are two ways to achieve a localization: either the geometrical information of the map is used directly, or the map is used to generate synthetic views in the world with a suited projection/camera model. In the first case, the vision information is used to reconstruct the environment's geometrical structure. Therefore, [5] and [8] are using the Visual SLAM system ORB-SLAM [15] to build a semi-dense point cloud of their

environment, whereas [7] builds a dense reconstruction. With both 3D representations given by the map and the visual reconstruction, the position of the camera can be determined either directly with a point cloud registration approach like ICP (Iterative Closest Point) [5–7], or with a 3D feature matching approach like in [8].

The second approach uses the map information to synthesize views close to the actual camera. Subsequently, the synthetic and the real images can be used to compute a transformation between both images. In [4, 9–11], they synthesize the images with colour information, which requires a more expensive map generation in advance. In contrast, [12, 14] showed that the intensity information of a laserscanner is sufficient to determine the current camera pose in the map. Finally, a transformation between both images has to be computed. The most common approach to compute this transformation is a Mutual Information Maximization based approach (e.g. see [4]) which maximizes the Shannon entropy between both images.

Napier et al. [13] presented a system which is similar to our approach but requires additionally intensity data. They first compute edges in both images with a subsequent patch normalization. Then as a brute force search, they simply sample synthetic images around an initial pose guess of the actual camera pose in order to find a best match which corresponds to the actual camera pose.

3 Algorithmic Approach

We aim at solving navigation tasks using an RGB or grey level camera and a given 3D map that is used for synthesizing depth images. Due to the challenges that raise when comparing images across such different modalities (visual and depth images), we focus on a holistic image comparison in contrast to feature-based methods (e.g., building on local keypoints). The following section 3.1 presents our applied depth image synthesis followed by an explanation of the image processing part of our approach in section 3.2 and a description of how this can be applied to mobile robot navigation tasks in section 3.3.

3.1 Synthesizing depth images

Since we focus on the comparison between depth and visual image, we implemented a straightforward approach to generate a synthetic spherical depth image at an requested camera pose in the map. Given the map as point cloud and a requested pose, each point's distance is projected onto a unit sphere centred at this pose. The azimuthal and polar angles are discretised to the target image resolution. By keeping only the minimal distance values for each direction, this spherical grid corresponds to the depth image. Pixels without projected 3D points are set to NaN values. This preliminary approach is easy to implement but its runtime is linear in the number of 3D points. Presumably, the runtime could be improved by the usage of computer graphics techniques including ray tracing algorithms and efficient data structures like k-d trees or octrees.

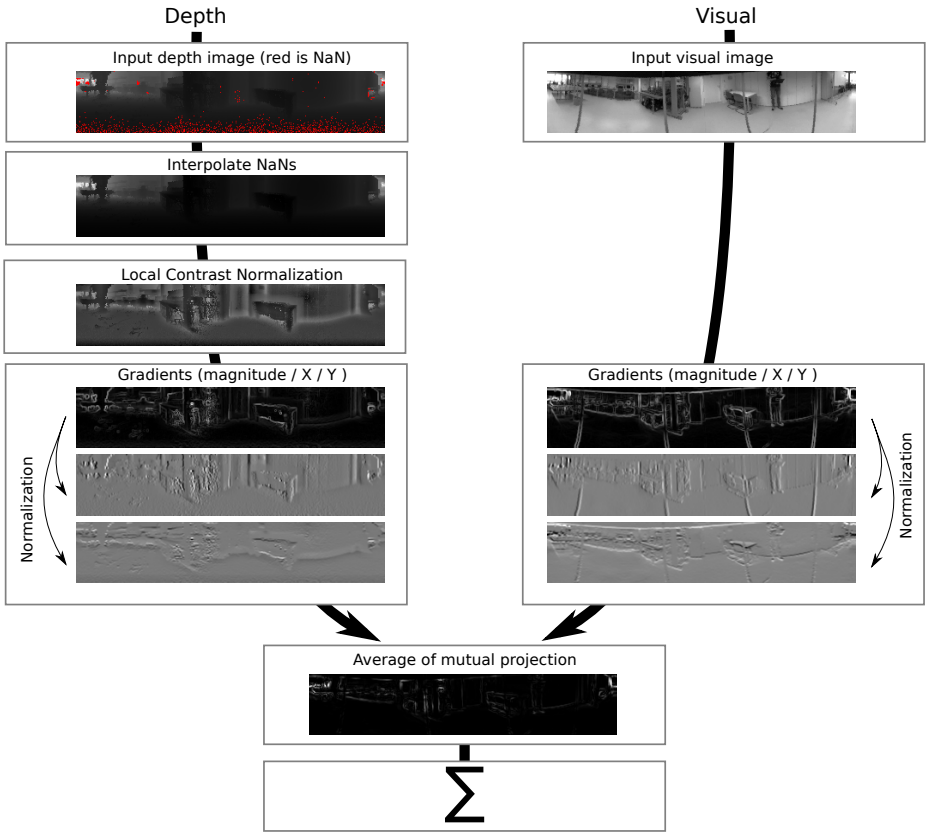

Fig. 3. Overview of the image processing pipeline to compute the similarity between depth and visual images. See text for details.

3.2 Comparing visual and depth images

A key component of the proposed system is the ability to compare visual grey level or colour images with depth images. Fig. 3 provides an overview of the algorithmic steps. The input depth and visual images are first processed independently to obtain gradient based image features for both modalities which are combined in the final stage of the algorithm. The processing of the **depth image** involves the following steps:

1. **Interpolate not-a-number (NaN) values** Our synthesized depth images, and also depth images from other sources like RGB-D cameras, include a considerable number of NaN pixels for which no depth information is provided. In the input depth image in Fig. 3, they are shown in red. To reduce their influence on subsequent processing steps, they are interpolated from their surrounding non-NaN values.

2. **Local contrast normalization** The underlying thesis for matching depth and visual images is that those parts in the scene that create depth changes are also likely to create visual features. However, this likelihood is not directly proportional to the magnitude of the depth change: E.g., think of the depth change between a door frame and the wall where it is mounted. Although the absolute change in depth is rather small, it might be as clearly visible in the camera image as the depth edge from the frame to the room behind the door when the door is open. Both sides of the door frame might provide useful features in both modalities.
   To utilize small and large depth steps in an image, we perform a local contrast normalization. The goal is to scale depth changes dependent on the amount of change in their local surrounding. The first step is to smooth the image with a Gaussian kernel (e.g., with standard deviation 2). Then, for each depth pixel $D(i,j)$, we collect the depth information in a $[(2k+1)\times(2k+1)]$ image neighbourhood centralized at this pixel (e.g., k being 2.5% of the image width). Finally, we compute mean μ and standard deviation σ of this data and modify the initial central pixel by
   $$D(i,j) \leftarrow \frac{D(i,j)-\mu}{\sigma} \tag{1}$$
   To avoid spurious edges at patch boundaries, we do not apply a patch normalization to zero mean and unit standard deviation to the whole $[(2k+1)\times(2k+1)]$ neighbourhood in one step, but process each pixel independently with its own neighbourhood. To keep this computational feasible, we utilize integral images and the Steiner translation theorem to compute the means and standard deviations.
3. **Gradients** To detect depth changes, we compute image gradients on the output of the local contrast normalization. Vertical and horizontal central differences are computed by convolving with Sobel filters. The resulting oriented gradients $\overrightarrow{G}_{depth}$ are additionally normalized to have a total magnitude sum of one for the whole image.

For the **visual image**, we directly compute the gradients $\overrightarrow{G}_{visual}$ by smoothing and central difference computation using Sobel filters and also normalize to a total gradient magnitude sum of one.

To obtain the similarity s between visual and depth images, we evaluate the mutual projections of these gradients by

$$s = \sum_{pixels} \left\| \frac{\overrightarrow{G}_{visual} \cdot \overrightarrow{G}_{depth}}{\|\overrightarrow{G}_{depth}\|^2} \cdot \overrightarrow{G}_{depth} \right\| + \sum_{pixels} \left\| \frac{\overrightarrow{G}_{depth} \cdot \overrightarrow{G}_{visual}}{\|\overrightarrow{G}_{visual}\|^2} \cdot \overrightarrow{G}_{visual} \right\| \tag{2}$$

The later section 4.1 will provide experimental results on this similarity computation including an evaluation of the influence of individual parts of the processing pipeline. While this pipeline can be used with standard cameras (as is done in section 4.1), the following section applies this approach to panoramic images from an omnidirectional camera to solve navigation tasks.

3.3 How to apply this approach to navigation tasks?

The application to navigation tasks is inspired by the theory of the dessert ant's navigation mechanisms [2] outlined in the introduction. Since we are not able to estimate the underlying camera motion directly from the comparison of depth and visual images, we require to sample possible transformations to find reasonable transformations. In the following, we want to discuss how this can be feasibly applied to three mobile robotics navigation tasks: visual compass, global localization, and navigation towards a goal location.

Visual compass Given a local 3D map (e.g., a single 3D laserscan from a nearby location) or a global 3D map and a rough estimate of the current position of the robot, we want to estimate the heading direction of the robot relative to this map based on a panoramic image from the current position.

Based on the comparison of visual and depth images from the previous section, we propose the following approach:

1. Inputs are a panoramic image I and a 3D map.
2. Synthesize a depth image D from the 3D map at the given estimated position. The orientation for synthesis is aligned to the coordinate system (i.e., yaw equals zero).
3. For a discrete set of possible orientations α (e.g., each 10 degrees), create the rotated panoramic image I_α. This can be efficiently done by circularly shifting the columns.
4. Compute the similarity between each rotated image I_α and D using equation 2. For an efficient implementation, equation 2 allows to precompute the image gradients and then rotate this feature image instead of computing features for each rotated input image.
5. The estimated robot orientation corresponds to the rotation α^* of the most similar image I_{α^*}.

The evaluation in section 4.2 will show that the quality of the heading estimation is robust to errors in the initial pose estimate.

Global localization Given a global 3D map, we want to estimate the absolute pose of the robot in the map. The image similarity computation of section 3.2 seamlessly integrates into Monte Carlo localization. For each sample robot position, a depth image is synthesized and compared to the current panoramic image based on the above described scheme for heading estimation. If only a single panoramic image is given, we can sample all possible robot locations and estimate their likelihood directly from the image similarities (example results are given in Fig. 6). If a sequence of images is available, the similarity from equation 2 can be used for computing the resampling weights in a particle filter for successive pose estimation.

Navigating towards a goal location Given a global 3D map, a rough initial guess of the current robot pose and a nearby target location X, we want to estimate the motion direction towards X. If the rough estimate of the current pose is far away from the target location, it may be sufficient to compute a (accordingly rough estimated) motion direction directly from the geometric relation between the initial guess for the current pose and the target. However, the closer we are to the target location, the higher is the relative error due to the only roughly known current robot pose.

In such cases, we can estimate the motion direction towards the target based on similarities between the current visual panoramic image of the robot and synthesized depth images around the target location. Therefore, we sample possible motion directions ψ and motion distances d. The sample values for ψ and d should be selected based on the geometric relation of the robot pose estimate and the target location. The more uncertain the pose estimate is, the more samples are required. In the results presented in section 4.2, we sample ψ each 10 degrees and use $d \in \{0.5m, 1m\}$.

Each sample is used to create a transformed *target location* $X_{\psi,d}$ and to synthesize a depth image from the 3D map at this pose. Again, the above scheme for heading estimation can be used to evaluate the accordance of this sample pose with the current image. The best motion direction can be obtained from the sample motion direction ψ^* that created the most similar synthetic depth image.

4 Experimental Results

This section is divided into two parts: an evaluation of the image processing pipeline and a proof-of-concept experiment on a navigation task with our mobile robot depicted in Fig. 5.

4.1 Image processing pipeline evaluation

Experimental setup To answer the question whether we are able to compute a reasonable similarity measure between depth and visual images at all, we first evaluate the image processing pipeline of section 3.2 on an RGB-D dataset. We collected a sequence of 300 image pairs with a hand held Asus Xtion sensor. This RGB-D camera provides pairs of visual and depth images that are (almost) pixel aligned. The sequence was captured in several rooms, a staircase and a hallway of our university building. A typical distance between consecutive images is one meter walking distance and/or 25 degree rotation mainly around vertical axis. Example images can be seen in Fig. 4.

Based on this dataset, we pose the following place recognition problem: Given a visual image from this dataset, decide which depth images show the same place. This experiment is evaluated by computing precision-recall curves. We use the algorithm of section 3.2 to compute similarities between all possible pairs of visual and depth images, and apply a threshold t to this similarity to obtain

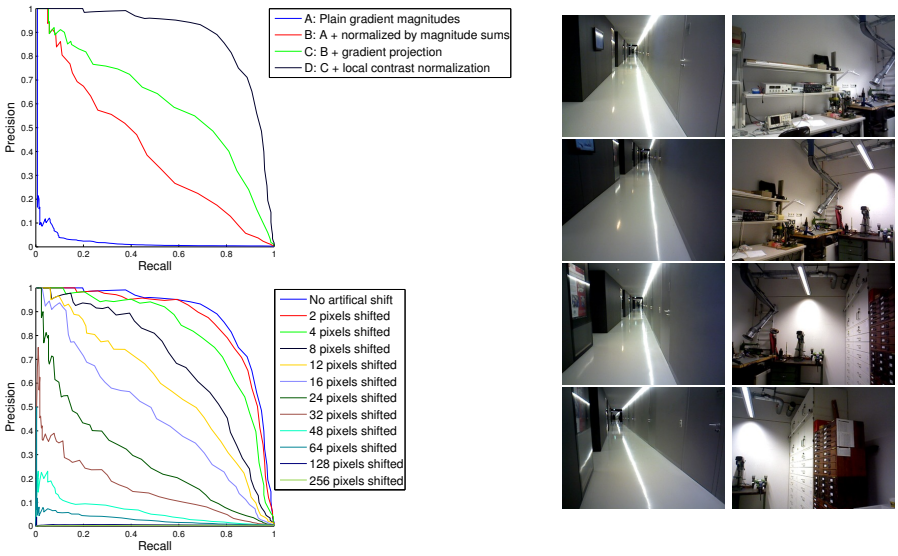

Fig. 4. (*left*) Evaluation results of the image processing pipeline on the RGB-D dataset. See text for details. (*right*) Two example sequence parts of the RGB-D dataset.

binary decisions about matchings. A true-positive (TP) matching is a visual image that is matched to the single correct depth image. A false-positive (FP) is every matching between non corresponding images. There can be one TP and multiple FP matchings for each image. All pairs of corresponding images that are not recognized as matchings are counted for false-negatives (FN). From these values, we compute:

$$recall = \frac{TP}{TP + FN} \qquad precision = \frac{TP}{TP + FP} \tag{3}$$

The threshold t is varied to compute curves in the precision-recall graph.

Results Fig. 4 provides results of the image processing pipeline of section 3.2 on this task. The similarities computed from the whole described pipeline (the black curve D) showed to be well suited to approach this task. A simple image evaluation, whether there are high gradients at the same parts of the images or not (by element-wise multiplication of the gradient magnitude images), is not sufficient to solve this task (the blue curve A). Normalizing the total sum of gradients in each image before the element-wise multiplication yields the improved red curve (B). Incorporating the direction of the gradients by mutually projecting them on each other further improves the results (the green curve C). Finally, the application of the described local contrast normalization results in an additional significant improvement.

While the images from the RGB-D camera are reasonably well pixel-aligned, for the target robotics application we want the similarity measure to be robust

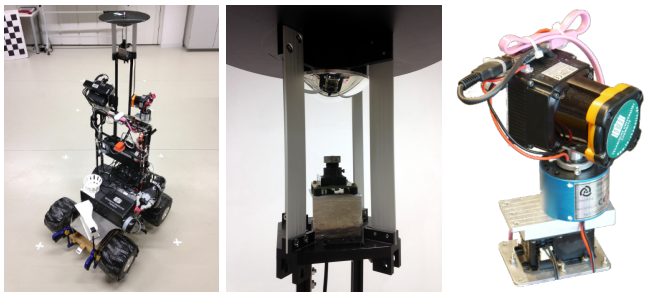

Fig. 5. The used hardware for the proof-of-concept real-world experiment. Our robot (see [3] for a system overview) is equipped with an omni-camera (middle), and a custom-made 3D laserscanner [16] (right).

against small deviations. The second graph in Fig. 4 shows an evaluation of the influence of a misalignment between pixels in the visual and depth images. Therefore, we artificially applied a horizontal offset on the depth images before computation of the similarity (we shift the image columns and refill with NaNs). The resulting precision-recall curves show that the performance gradually decreases with increasing amount of misalignment. This behaviour is well suited for application in the navigation approaches described in section 3.3: We don't want the similarity measure to be invariant against such misalignments since this would, e.g., prevent from determining the motion direction close to a target or even the application for a visual compass at all. On the other hand, robustness to reasonable misalignments reduces the required number of pose samples and increases robustness to small errors in the 3D map or the camera calibration.

Although there are significant differences between these RGB-D image pairs and the comparison of panoramic images and synthesized depth image, the good performance of the image processing pipeline on this experiment encourages its application for the navigation task of section 3.3.

4.2 Proof-of-concept real-world experiment

Experimental setup This proof-of-concept real-world experiment is conducted for an investigation of the performance of our proposed algorithm for a camera-based localization in a 3D world. As experimental environment we choose a lab (see Fig. 6) in which we mark eight points **A**-**H** in a 3×1 metres grid for which our robot captures omnidirectional camera snapshots.

We use a skid-steering mobile robot[1] which is equipped with both a 3D laserscanner and an omni-camera (see Fig. 5). Our custom-made 3D laserscanner [16] consists of a spinning *Hokuyo UTM-30LX* 2D laserscanner, which achieves higher resolutions than typical ready-to-use 3D laserscanners. It provides 0.25°

[1] A full description of the robot including hardware and software setup can be found in [3].

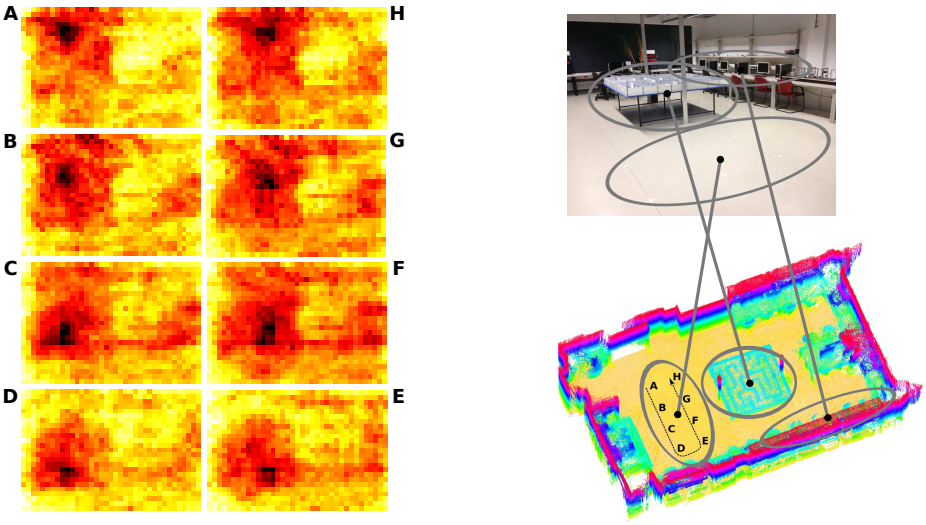

Fig. 6. Visualization of the result of our proof-of-concept real-world experiment. The left side shows heat maps (top view of the room) representing the similarity of depth images synthesized all over the room to the eight omni-camera images at the positions **A**-**H** (darker colour corresponds to higher similarity). The right 3D map is the model of the room (shown above), and was used to generate depth images at arbitrary positions.

vertical resolution which enables us to acquire dense laserscans of an environment. For the navigation task, we create a 3D Point Cloud map of our room by matching multiple 360° laserscans with the point cloud registration algorithm ICP (Iterative Closest Point). Fig. 6 shows the resulting 3D map; note that the ceiling was removed for visualization.

The used omnidirectional camera consists of a wide-angle camera which points at a curved concentric mirror. The omni-camera has an aperture angle of approx. 185°, and acquires images with a resolution of 480×752 pixels. The returned images are rectified with the *OCamCalib Toolbox* by Scaramuzza et al. [17] in order to get 360° panoramic images.

Given the full 3D map of our experimental environment, our robot performs a run along the points **A**-**H** (see Fig. 6). Meanwhile, it records omnidirectional images at the locations **A** to **H** for the subsequent evaluation (see results below).

Results To evaluate the localization performance of our proposed approach we synthesize depth images in a regular grid all over the room, and use the algorithmic approach of section 3 to compute their similarity to all eight real omni-images at the positions **A**-**H** separately.

Fig. 6 shows the result of this evaluation: Our algorithm performs quite well as it shows the highest similarities for positions which are close to the actual positions. All maxima approximately correspond to the actual positions of the respective omni-image; this illustrates that our algorithm is able to perform a global localization, this is to estimate global positions in the world. As can be

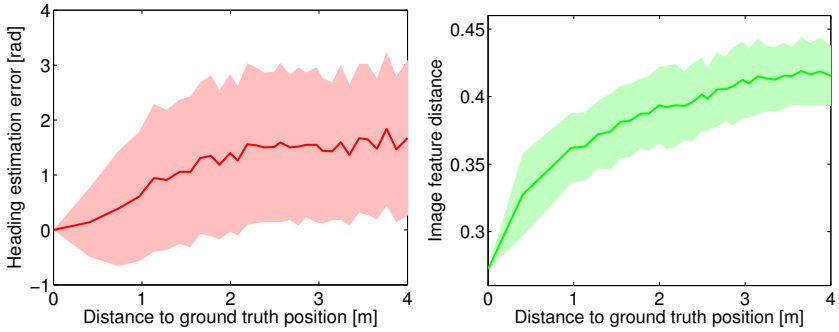

Fig. 7. *(left)* Evaluation of the heading estimation error between omni-image and synthesized depth image as a function of the distance between both images. *(right)* Evaluation of the image feature distance $\frac{1}{1+s}$ between omni-image and synthesized depth images as a function of the distance between both images.

seen, positions close to the maximum show also higher values, and their similarity decreases with higher range. Such a continuous decrease with higher distance to the actual position is advantageous for tasks in which we want to navigate to a target position, as we could simply follow close high-similarity values until we reach our actual target position.

Fig. 7 shows the result of a performance measure which investigates the heading estimation error as well as the image feature distance as a function of the range to the actual position. The heading estimation error (Fig. 7, left) represents the difference between the actual orientation of the omni-image and the estimated orientation of the synthesized depth images in the room. An ideal curve would be a horizontal line which remains at zero, however, such a behaviour is impossible since the amount of occlusions and perspective differences increases with a higher range. Therefore, the behaviour of the heading estimation error with increasing range is reasonable: for the first part, it remains relatively flat; this indicates the heading estimation works fine even for a difference to the actual position. The continuous increase for higher ranges is also desirable; it shows that the heading estimation performance does not rapidly drop for small occlusions.

The interpretation of the behaviour of the image distance with respect to the actual position distance in the world (Fig. 7, right) depends on the use case: A very rapid increase of the error value, even for small distances to the ground-truth location, might be advantageous for global localization tasks as we could determine the actual position more exactly, whereas a gradual increase might be more desirable for navigation tasks as we could follow the gradient to reach a target position. Since the curve shows a higher increase for lower values and a more continuous increase for medium distances, it seems to be a good combination for both a global localization and navigation.

Fig. 8 shows a vector field which is intended to investigate the performance of an actual navigation task. Here, we use a localization approach as described in

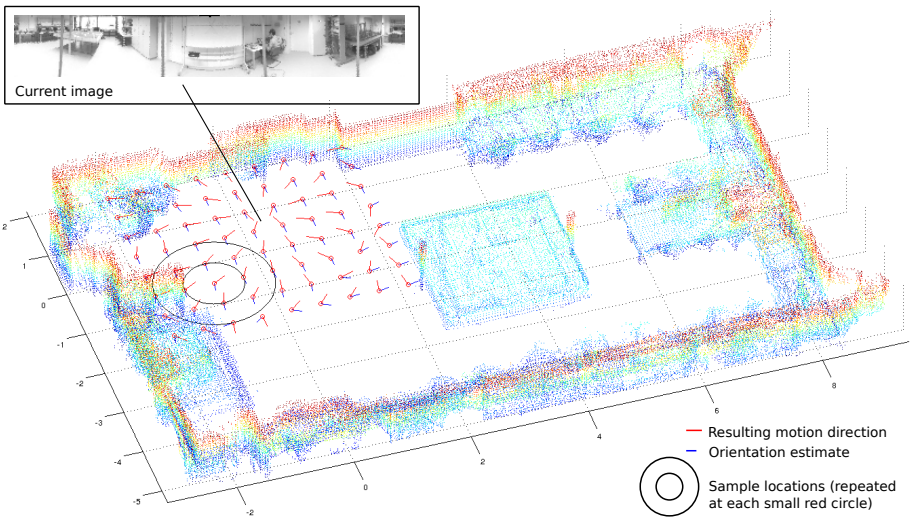

Fig. 8. Estimation of motion direction from the current pose to each of the target locations shown as red circles. For each of the targets, we sample depth images from the local neighbourhood at angles ψ on two circles with radius d of 0.5 and 1.0 metres that are centred at the target. For one example target, these circles are shown in black. The red lines indicate the estimated motion direction between the current pose and target. The short blue lines show the estimated orientation of the current view (this should point parallel to the short dimension of the room).

section 3 which compares synthesized depth images with the actual omni-image in order to estimate the camera's orientation in the world, and the required movement direction to get to the target position. Given a current panoramic view, we illustrate the estimated motion direction (red lines) for a set of possible target locations (red circles; in a practical application, we would expect only a single target at a time). Each target is evaluated by sampling and evaluating synthetic depth images at angle ψ and motion distance d. The two concentric black circles illustrate the two resulting circles for one of the targets. As can be seen, for most of the target locations, the resulting motion directions connect the target pose and the current view location. In particular, for many target locations whose distance is larger than d from the current image (i.e., the current image location is not on the the black circles), there are reasonable motion direction estimates as well.

5 Conclusion and Future Work

In this paper, we dealt with the problem of camera-based localization in a given 3D map. Our presented approach builds upon synthesized depth images which are compared to real omnidirectional images in order to determine the current pose of the camera. We discussed how this image matching can be applied for navigation tasks like visual compass, global localization, or navigation towards

a target position. The evaluation of the image processing part on an RGB-D dataset showed that the algorithm is able to provide reasonable similarity measures between visual and depth images. The final set of proof-of-concept experiments on mobile robot navigation tasks also showed promising results and revealed plenty of open questions for future work.

Our aim is to investigate the applicability of this algorithm for real-world indoor scenes like office environments, but also for outdoor scenarios like in the SpaceBot Cup; a German national contest on a moon-like surface which we attended in 2013 and 2015 (see [3] for details). For this outdoor navigation task, we intend to use a heterogeneous robot team which consists of a bigger robot, like in our real-world experiment in this work, and a smaller robot, which is not equipped with an on-board 3D laserscanner, so that it has to use the camera to localize itself in the bigger robot's 3D map which is acquired with a 3D laserscanner.

During our experiments we encountered problems with the calibration of our omni-camera. The camera-mirror alignment and consequently our intrinsic calibration is sensitive to mechanical strains. Hence, the applicability of our approach has to be investigated for fish-eye cameras, or even standard field-of-view cameras, which are less sensitive for mechanical strain. Furthermore, the system is also sensitive to its extrinsic calibration, i.e. the orientation of the vertical camera axis in the map has to be known. In uneven outdoor terrain, this could be addressed by combination with an IMU.

The current system is not runtime optimized and completely implemented in Matlab, which slows down the computation time. Beside a more efficient computational implementation, algorithmic improvements can particularly address the number of required depth-image samplings. This could be achieved by different techniques: First, a particle filter could be applied which reduces the number of possible locations in the world. Second, more sophisticated techniques for a reduction of depth-image sampling could be developed like the usage of image warping techniques which are applied in the mentioned *ant algorithm* by Möller [2].

Currently, the features of our synthesized depth images are computed in the 2D image space. In our future work, we want to compare this approach to a projection of 3D features onto our synthesized images. Our current colour-depth-image comparison is a hand-designed approach; we believe that the usage of learning techniques could contribute to a performance improvement of our approach as it could learn more sophisticated features.

In future work, we will also evaluate the benefit from enhancing the synthesized depth images with intensity and/or colour information and include a a comparisons of our system to other existing approaches which encounter the problem of a camera-based localization in 3D maps.

References

1. Ralf Möller. A model of ant navigation based on visual prediction. *Journal of Theoretical Biology*, 305:118 – 130, 2012.

2. Ralf Möller. Local visual homing by warping of two-dimensional images. *Robotics and Autonomous Systems*, 57(1):87 – 101, 2009.
3. S. Lange, D. Wunschel, S. Schubert, T. Pfeifer, P. Weissig, A. Uhlig, M. Truschzinski, and P. Protzel. Two autonomous robots for the dlr spacebot cup - lessons learned from 60 minutes on the moon. In *Proceedings of ISR 2016: 47st International Symposium on Robotics*, pages 1–8, June 2016.
4. Guillaume Caron, Amaury Dame, and Eric Marchand. Direct model based visual tracking and pose estimation using mutual information. *Image and Vision Computing*, 32(1):54 – 63, 2014.
5. T. Caselitz, B. Steder, M. Ruhnke, and W. Burgard. Monocular camera localization in 3d lidar maps. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1926–1931, Oct 2016.
6. T. Caselitz, B. Steder, M. Ruhnke, and W. Burgard. Matching geometry for long-term monocular camera localization. In *ICRA Workshop: AI for long-term Autonomy*, 2016.
7. C. Forster, M. Pizzoli, and D. Scaramuzza. Air-ground localization and map augmentation using monocular dense reconstruction. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3971–3978, Nov 2013.
8. A. Gawel, T. Cieslewski, R. Dub, M. Bosse, R. Siegwart, and J. Nieto. Structure-based vision-laser matching. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 182–188, Oct 2016.
9. G. Pascoe, W. Maddern, and P. Newman. Direct visual localisation and calibration for road vehicles in changing city environments. In *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, pages 98–105, Dec 2015.
10. G. Pascoe, W. Maddern, A. D. Stewart, and P. Newman. Farlap: Fast robust localisation using appearance priors. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6366–6373, May 2015.
11. A. D. Stewart and P. Newman. Laps - localisation using appearance of prior structure: 6-dof monocular camera localisation using prior pointclouds. In *IEEE International Conference on Robotics and Automation*, pages 2625–2632, May 2012.
12. R. W. Wolcott and R. M. Eustice. Visual localization within lidar maps for automated urban driving. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 176–183, Sept 2014.
13. A. Napier, P. Corke, and P. Newman. Cross-calibration of push-broom 2d lidars and cameras in natural scenes. In *2013 IEEE International Conference on Robotics and Automation*, pages 3679–3684, May 2013.
14. Gaurav Pandey, James R. McBride, Silvio Savarese, and Ryan M. Eustice. Automatic extrinsic calibration of vision and lidar by maximizing mutual information. *Journal of Field Robotics*, 32(5):696–722, 2015.
15. R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Trans. on Robotics*, 2015.
16. Stefan Schubert, Peer Neubert, and Peter Protzel. How to build and customize a high-resolution 3d laserscanner using off-the-shelf components. In Lyuba Alboul, Dana Damian, and Jonathan M. Aitken, editors, *Towards Autonomous Robotic Systems: 17th Annual Conference, TAROS 2016, Sheffield, UK, June 26–July 1, 2016, Proceedings*, pages 314–326, Cham, 2016. Springer International Publishing.
17. D. Scaramuzza, A. Martinelli, and R. Siegwart. A toolbox for easily calibrating omnidirectional cameras. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5695–5701, Oct 2006.