Scale-Preserving Long-Term Visual Odometry for Indoor Navigation

Sebastian Hilsenbeck, Andreas Möller, Robert Huitl, Georg Schroth, Matthias Kranz [1], and Eckehard Steinbach
Institute for Media Technology
Technische Universität München
Munich, Germany
Email: s.hilsenbeck@tum.de, andreas.moeller@tum.de, huitl@tum.de,
schroth@tum.de, matthias.kranz@ltu.se, eckehard.steinbach@tum.de

***Abstract*—We present a visual odometry system for indoor navigation with a focus on long-term robustness and consistency. As our work is targeting mobile phones, we employ monocular SLAM to jointly estimate a local map and the device's trajectory. We specifically address the problem of estimating the scale factor of both, the map and the trajectory. State-of-the-art solutions approach this problem with an Extended Kalman Filter (EKF), which estimates the scale by fusing inertial and visual data, but strongly relies on good initialization and takes time to converge. Each visual tracking failure introduces a new arbitrary scale factor, forcing the filter to re-converge. We propose a fast and robust method for scale initialization that exploits basic geometric properties of the learned local map. Using random projections, we efficiently compute geometric properties from the feature point cloud produced by the visual SLAM system. From these properties (e.g., corridor width or height) we estimate scale changes caused by tracking failures and update the EKF accordingly. As a result, previously achieved convergence is preserved despite re-initializations of the map. To minimize the time required to continue tracking after failure, we perform recovery and re-initialization in parallel. This increases the time available for recovery and hence the likelihood for success, thus allowing almost seamless tracking. Moreover, fewer re-initializations are necessary. We evaluate our approach using extensive and diverse indoor datasets. Results demonstrate that errors and convergence times for scale estimation are considerably reduced, thus ensuring consistent and accurate scale estimation. This enables long-term odometry despite of tracking failures which are inevitable in realistic scenarios.**

I. Introduction

The ability to retrieve a user's location represents a fundamental enabling technology for a wide range of applications, above all, navigation. In contrast to outdoor scenarios, where GPS is established as the universal solution, indoors, no such standard exists as of today.

Indoor positioning methods comprise a wide variety of technologies [1], [2]. An overview on vision-based approaches can be found in comprehensive surveys of recent [3] and older systems [4]. Active systems rely on a priori knowledge about the environment, such as a building model, reference images or fiducial markers [5], [6]. Passive systems, in contrast, are localized using pre-installed hardware, e.g., by multiple fixed cameras [7].

Since passive localization does not scale well for large environments, active systems need to be employed. In order to improve localization accuracy, the relative movement between reference points (i.e. where absolute location estimation is possible) can be estimated using odometry. To this end, different solutions have been provided, ranging from pedometers [8] and inertial sensors [9] to tracking using monocular or stereo cameras [10], or a combination of multiple techniques [11]. Camera-based (visual) odometry can be understood as a special case of visual SLAM (Simultaneous Localization and Mapping). SLAM refers to the problem of estimating the pose of a robot in an initially unknown environment, while at the same time building a map of the environment. Visual odometry does not aim at creating a globally consistent map of the environment, but uses local maps instead.

A. Related Work

In this paper, we concentrate on the special case of monocular SLAM, where images from a single camera are used for localization and mapping. In order to compute depth information, salient image points are tracked over time while the camera is in motion. Due to the projective nature of a camera, the depth can be determined only up to scale, i.e., the unit is unknown and has to be estimated by other means. Further, if no reference data is available to fix the scale, a problem called *scale drift* arises, where the scale is not only unknown, but deviates over time. Recent work like Strasdat et al. [12] models the scale in the state vector of a Kalman Filter and corrects the scale drift when loop closures are detected. The same formulation allows for the propagation of the metric scale over the map as soon as the scale factor is known.

Various approaches to resolve the scale ambiguity and to establish a metric coordinate system have been proposed. For example, Davison et al. [13] add a calibration object with known size to the scene, or Munguia et al. [14] require manual selection of three scene points with known distance. Likewise, if objects with known size can be identified, they can be used to determine the metric scale. In this paper, we follow the latter approach and propose to estimate the (non-metric) size

[1]now with Department of Computer Science, Electrical and Space Engineering; Luleå University of Technology; Luleå, Sweden

978-1-4673-1954-6/12/$31.00

of corridors from point clouds and relate it to the (metric) size determined from a building model, where available.

When no information about the physical dimensions of the scene is available (e.g., through a building model along with a coarse position estimate), the metric scale is obtained by fusing inertial measurements from accelerometers and gyroscopes with visual data. There is a huge body of previous work on visual-inertial data fusion (an introduction is given by Corke et al. [15]). Most researchers focus on improving the accuracy of pose estimates, e.g., by disambiguating translational and rotational motion which is difficult using visual information only, or by stabilizing inertial measurements which are prone to drift by adding visual information. Furthermore, the relative pose between camera and inertial sensors can be auto-calibrated, and a more robust estimate of the gravity vector can be obtained [16].

More recent work exploits inertial data to perform metric reconstruction, i.e., the unknown visual scale is determined by incorporating inertial measurements [17], [18]. Kneip et al. describe an approach of this nature in [19]. They analyze delta-velocities obtained from both visual pose estimates and inertial measurements to solve for the unknown scale factor.

B. Contributions

While this technique allows estimation of scale, it strongly relies on a good initialization of the filter and takes time to converge. Until convergence of the filter, the odometry estimates are of limited use. Inaccurate initialization prolongs this phase or may even result in divergence. In realistic indoor scenarios, visual tracking is likely to fail frequently as buildings often exhibit sparse or ambiguous texture. As a result, re-initialization of the map becomes necessary, which introduces an unknown new scale.

We perform recovery and re-initialization in parallel. This increases the time available for recovery and hence the likelihood for success. At the same time, tracking interruptions are minimized and fewer re-initializations are required.

Further, we propose a fast and robust method for scale initialization that exploits basic geometric properties of the learned local map. Using random projections, we efficiently compute geometric properties from the feature point cloud produced by the visual SLAM system. Since these properties, e.g., hallway width or height, remain locally constant, the scale ratio between consecutive maps can be determined to recover the scale estimate before re-initialization. Thus, the correct scale is preserved given that the filter has converged once (possibly over several re-initializations).

The remainder of this paper is structured as follows: In Section II, we introduce the parallel tracking and mapping algorithm our approach is based on, while Section III explains the parallelization of recovery from a tracking failure and re-initialization of a new map. In Section IV, we discuss the robust estimation of relative scale changes between consecutive maps based on geometric building properties, and augment a Kalman Filter by directly feeding these estimates into its state update in case of a map re-initialization. In Section V, we describe the proposed efficient extraction of a room's dimensions from the point cloud produced by the visual SLAM system, and Section VI presents an evaluation of the proposed methods. The experiments demonstrate that our approach can reduce the number of re-initializations and tracking failures compared to traditional parallel tracking and mapping, and that our system aids in estimating the scale between local maps.

II. Monocular Visual Odometry

We use *Parallel Tracking and Mapping* (PTAM) by Klein and Murray [20] as the basis of our monocular visual odometry system. In contrast to conventional SLAM algorithms, PTAM separates the tracking of image features for pose updates from the mapping part, where trackable features are collected to build a three-dimensional map of the local environment. This allows for an update of the camera pose at frame-rate, while expensive optimization techniques for the mapping can be executed at a lower rate and when computational resources are available. As a result, PTAM is eligible for mobile applications, whereas conventional SLAM approaches are prohibitively expensive due to their (at least) quadratic complexity in the number of observations.

A. Parallel Tracking and Mapping

PTAM detects feature points in the camera image using the FAST [21] keypoint detector. Observations in the current frame are associated with those in the previous frame by the tracking routine which performs an epipolar search and patch-based cross-correlation. The successful feature associations are used to compute the new 6D pose for the current frame. An m-estimator with outlier rejection further increases robustness.

The mapping part of PTAM operates keyframe-based, i.e., whenever a frame contains a large number of new observations that have been tracked, it is inserted into the map. Hence, every feature is stored within a keyframe and the map comprises a sparse set of keyframes. This allows for the use of bundle adjustment to jointly optimize the three-dimensional positions of observations and the six-dimensional poses of keyframes. The optimized 3D positions of the features are, in turn, used during tracking for subsequent pose updates.

The initialization of the map is performed using the 5-point algorithm [22] on a pair of images at the beginning of the sequence. A set of features from the first image is tracked over several frames in order to generate the required baseline. As this step assumes a non-zero camera velocity, it may have to be repeated until a map is successfully instantiated.

B. Discussion of PTAM

PTAM provides a robust basis for a monocular visual odometry system running on mobile devices. The structure of the algorithm leads to comparatively low computational complexity while generating accurate position and orientation estimates with fairly small errors due to drift. However, an inherent disadvantage of monocular sensor systems is the

inability to observe the true scale of position data relative to a metric coordinate system.

To enable anytime localization and navigation in indoor environments, it is therefore necessary to include additional information in order to scale the visual odometry to metric coordinates. We propose to use basic geometric properties of the surrounding building structure to retrieve the unknown scale factor, exploiting the fact that PTAM already estimates the 3D structure of the local environment.

Further challenges are rapid movements of either the camera or objects in the scene that cause motion blur and thus decimate the number of visible features to a degree that the system loses tracking. PTAM incorporates a basic recovery mechanism, i.e., it stops adding new keyframes to the map when tracking quality is considered poor and tries to match the scene with previous keyframes instead. However, recovery takes time and might only make sense in some cases, for instance, when large objects occlude the camera only temporarily (e.g., opening doors, people walking by). By contrast, if also the camera has moved in the meantime, it is unlikely that recovery succeeds. Here, it would be beneficial to create a new map immediately in order not to lose time with recovery and reduce the non-tracking time.

III. Instantaneous Recovery from Tracking Failures

A. Autonomous Operation

In large-scale, real-world environments, it can occur that the available features are not sufficient for tracking. In this case, PTAM's recovery implementation fails with high probability. If the user moves away from the existing map, PTAM gets locked in an infinite loop as it is not able to find the current video frame's features in the map. Since we are interested in odometry, it is not a conceptual problem that tracking is only possible in local maps. However, the creation of a new map requires an initialization based on a stereo pair, using the 5-point algorithm [23]. For this, a candidate stereo pair is chosen autonomously and, if insufficient feature correspondences were found and no map can be created, the procedure is repeated until the initialization is successful.

B. Parallel Recovery and Re-Initialization

In Sec. II, we discussed the problems of PTAM with relation to rapid movement caused, for instance, by persons in the scene or quick camera motion, which is likely to occur with a handheld device. In order to make the system more robust with respect to occlusions and moving objects, we propose a combined approach of recovery and re-initialization, using the following basic principle:

- As soon as the system is no longer able to detect enough features for tracking (e.g., because of an obstacle), it enters recovery mode. At the same time, initialization of a new map is started in the background.
- If the system cannot recover quickly, it switches to the newly initialized map, so that it can immediately continue with tracking.
- If recovery is successful (e.g., when the obstacle disappears and the previous features become visible again), the newly initialized map is discarded and tracking continues with the old map.

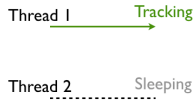
To realize this functionality, two tracker threads are used, which will in the following be referred to as T1 and T2 (see Fig. 1). In the normal state, only T1 is active while T2 sleeps, which is symbolized by a continuous and a dotted line in Fig. 1(a). As soon as tracking is interrupted (the tracker enters the *Lost* state), T1 sends a *wake* signal to T2 (Fig. 1(b)). T2 now starts a new stereo initialization and attempts to create a new map. At the same time, T1 enters recovery mode and tries to estimate the camera pose in all existing keyframes (Fig. 1(c)). Depending on the success of the recovery step, the system can proceed in two different ways. If T2 created a new map before recovery in T1 succeeded (case 1, Fig. 1(d)), T2 sends a *sleep* signal to T1 and tracking continues with the newly created map. The original map is discarded, but since we are interested in relative positioning, we do not require that we can resume tracking in the old map if we ever come back to that location. If the recovery in T1 succeeded before T2 created the new map (Fig. 1(e)), tracking continues with T1, and the creation of the new map is interrupted. This corresponds to the 'original' single-tracker behavior, since the second map is not used.

When comparing these two alternatives presented in Fig. 1(d) and Fig. 1(e), the latter one is preferential, since we want to use one single map as long as possible to avoid re-initializations (and thereby scale loss). Therefore, the original tracker (here T1) is prioritized over the second initialization. At the time when T2 has built its new map and continues tracking, it sends the *sleep* signal to T1 with a grace period parameter (set to four seconds). During this grace period, T1 has time to pursue recovery, as shown in Fig. 1(e). If it succeeds, the system switches back to T1 and discards the newly created map of T2 (including the features and keyframes that already might have been added to it). If T1 was not able to recover within the grace period, it is finally suspended and its map is deleted.

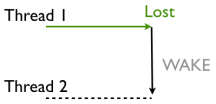
In case of a tracking failure, this solution is able to follow up tracking quickly, since re-initialization starts immediately after entering the *lost* state. This is particularly important for longer periods of motion where recovery is likely to fail. When the obstacle disappears within the grace period, the system is able to recover almost instantly, since only a switch to the previous tracker thread is required.

IV. Scale Estimation Based on Building Geometry

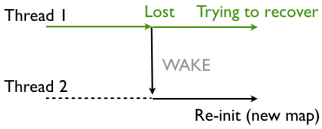
For most buildings, geometric properties such as height, width, and length of a room or corridor remain locally constant until interrupted or ended by another part of the building. Especially in public buildings (e.g., hospitals, universities,



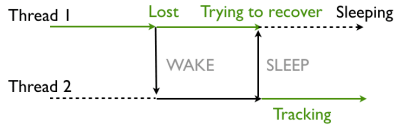
(a) When Thread 1 (T1) is tracking, Thread 2 (T2) sleeps.



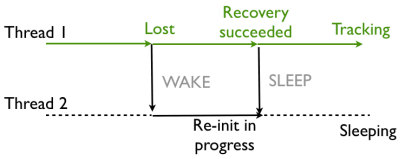
(b) In case of lost tracking, T1 sends a wake signal to T2.



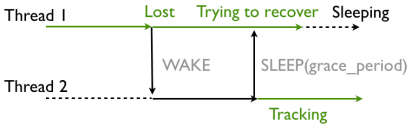
(c) While T1 tries to recover, T2 initializes a new map.



(d) Case 1: When the new map is ready, tracking continues with T2. The original thread T1 sleeps.



(e) Case 2: If recovery succeeds before T2 completes the re-initialization, T1 continues and the new map is discarded.



(f) In an extended version, a grace period is added for T1 to recover. Even if T2 is already tracking with the new map, T1 still has time to recover.

Fig. 1. Schematic overview of the parallel tracker implementation for increased robustness to moving objects.

administration buildings), this assumption is mostly valid. We observe that these properties are usually defined as the distance between opposite dominant planes within the interior building structure. We demonstrate in the following how properties of this kind can be estimated efficiently and used to aid in scale estimation for monocular visual odometry.

In general, there are two scenarios to distinguish: One where there is no prior knowledge available, and one where the true building dimensions are known a priori (e.g., from a floor plan or an initial survey). In both cases, geometric properties provide valuable information to improve scale estimation.

A. Case 1: Unknown Building Dimensions

In the general case, information on the dimensions of rooms and corridors is not available and the standard approach to scale estimation is the incorporation of inertial measurements using a Kalman Filter [11]. The filter takes metric acceleration and angular velocity readings as input and propagates a kinematic model to infer metric position and orientation of the device for each time step. These predictions are correlated with scaled position data as provided by the monocular visual odometry algorithm. This way, the scale factor is estimated in real-time as part of the filter's state vector and even occasional scale drift can be accommodated for.

The estimation process takes time to converge, though, and its duration depends to a great degree on the accuracy of the initialization of the filter state. As long as convergence is incomplete, the visual odometry data are not reliable. Therefore, it is important that this phase takes as little time as possible and occurs only once at the beginning of operation. In real indoor environments, however, frequent challenges occur for a vision-based system such as dynamic objects and sparsely textured regions as explained in the introduction. PTAM is no exception in this context and relies strongly on the presence of a sufficient number of features to track in order to deliver contiguous pose updates. If this is not the case and recovery methods as described in Sec. III fail, a new map has to be initialized from a new pair of images. Again, due to the arbitrary baseline, the scale of the map is unknown and most likely different from the one before the tracking failure. This entails that the Kalman Filter has to start over and so far achieved convergence is lost. In the worst case, this happens repeatedly with the result that convergence of the filter and thus scale estimation become impossible.

We exploit the observation that local building geometry can be assumed constant to alleviate the effect of frequent failures of the visual odometry system. We estimate properties such as corridor height while the Kalman filter converges. In case a new map needs to be initialized, a simple comparison of those properties to the estimates from before the re-initialization allows us to directly compute the scale factor between the current and the previous map. This factor is multiplied to the Kalman Filter's scale variable which, therefore, remains unaffected by the re-initialization of the local map and previously completed convergence is not lost.

B. Case 2: A Priori Known Building Dimensions

For certain buildings, it is possible to extract information on basic dimensions of rooms and corridors in advance, for instance, from floor plans or by conducting an initial survey. Once the true width of, e.g., a corridor is known, a direct comparison with the estimated width in the vision coordinate system reveals the scale factor between the visual odometry and metric coordinates. Thus, the odometry data become at once useful for navigation and other location-based services.

In this scenario, the mobile device requires a very coarse estimate of its absolute position within the building to identify the current room or corridor. Existing mobile localization approaches based on Wi-Fi using fingerprinting have been demonstrated to achieve room-accurate positioning while sub-

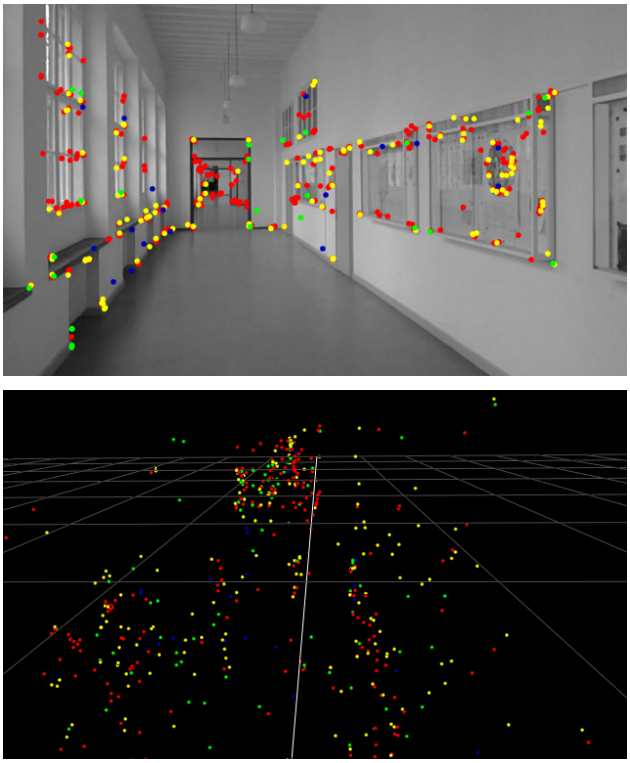


Fig. 2. Top: A frame with feature points that are tracked by PTAM. The color of a point indicates the size of the feature in the image. Bottom: The corresponding three-dimensional point cloud as it appears in the reconstructed map. The 3D positions of the points are very noisy. The estimated ground plane is indicated by the grid lines.

meter accuracy still remains out of reach [24]. Likewise, systems that employ content-based image retrieval have been shown to yield accuracies at meter-level but are unable to produce contiguous pose information for the mobile device [25]. Both approaches complement each other and enable direct scale computation using known geometric building properties. This way, the sub-meter accurate visual odometry readings become available and no convergence behavior needs to be taken into consideration.

V. Estimation of Geometric Building Properties

As the user navigates through the building, PTAM continuously tracks feature points and inserts their 3D positions into the map. Hence, the map is represented by a point cloud along the trajectory of the mobile device. In order to reliably extract geometric properties that relate to the building structure, we develop a statistical technique that applies a cascade of projections to the point cloud. The most basic geometric properties of buildings typically refer to distances between opposite (parallel) dominant planes in the building structure. As these features remain visible despite being sampled into a point cloud, we specifically address the problem of retrieving this type of properties from a sparse and local point cloud.

A. Relating Point Cloud Features to Building Properties

In contrast to state-of-the-art approaches that use plane fitting in combination with RANSAC [26] or full map matching (i.e., point cloud matching using ICP [27]), we employ a combination of steered and random projections to achieve efficiency while retaining robustness. Steered projections are based on additional information (e.g., gravity) and allow us to exploit this knowledge for reducing computational complexity as early as possible during the estimation process. Random projections, in contrast, ensure robustness as the data are typically subject to considerable noise.

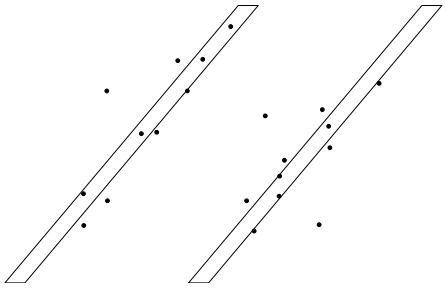
Using the inertial measurement unit (IMU) of the mobile device, the direction of gravity is retrieved in order to identify the orientation of the building ground plane relative to the PTAM point cloud. In general, this orientation is arbitrary and depends on the device orientation at the time when the map was first initialized. The objective ultimately is to identify the orientation of an orthogonal coordinate system that maximizes the correspondence of the point cloud with up to three orthogonal pairs of parallel planes that define the dimensions of the current room or corridor (usually four walls, the ceiling, and the floor). To identify the walls, we project the point cloud onto the ground plane. Within the resulting two-dimensional point cloud, the model we search for is a pair of dominant parallel lines as explained in Sec. V-B. The uncertainty for every hypothesis is measured by the spread of the point cloud around the model. In general, this approach leads us to find the current corridor's longer (i.e., dominant) pair of walls.

For the floor and the ceiling, we project the three-dimensional point cloud onto the gravity vector, yielding a one-dimensional distribution. We search for the two dominant clusters as described in Sec. V-B. The cluster centers correspond to the intersections of the projection line with the floor plane and the ceiling plane, respectively, and thus reveal their position within the point cloud.

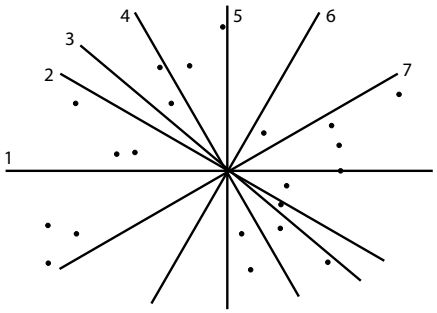
Both the projection onto the ground plane as well as the projection onto the gravity vector have the purpose of identifying perpendicular structures. Hence, in a two-step process, we use the position estimates of the dominant planes that result from the first iteration to re-sort the point cloud. For the second iteration, the points that agree well with the model for the floor and the ceiling are removed from the projection onto the ground plane and, vice versa, points that fit to the estimated model for the walls are ignored when projecting onto the gravity vector. This approach sorts the point cloud into parts that correspond to the floor or the ceiling and parts that belong to the walls. As a result, the two complementary projection techniques tend to focus on perpendicular structures only, and thus converge more precisely.

B. Extracting Point Cloud Features

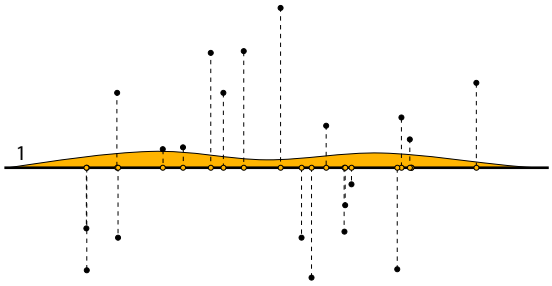
The central problem is to identify the pair of dominant parallel lines (planes before the projection) within a two-dimensional point cloud. We propose to use a statistical



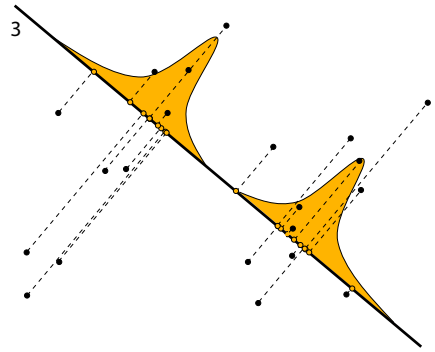
(a) The camera tracks feature points on the walls of a corridor (seen from above) and estimates the 3D-positions of the points. Due to noise, the resulting point cloud is scattered along the true corridor walls.



(b) After projecting the point cloud onto the ground plane, we create a bundle of projection lines at the camera position in order to find the normal to the corridor walls.



(c) To identify the optimal normal, we project the 2D point cloud onto each projection line in the bundle and fit a bimodal Gaussian to the resulting distribution (here projection line 1). The dotted lines connect the original points with their respective projections. The resulting bimodal distribution (orange area) exhibits large spread.



(d) The estimated bimodal distribution for projection line 3 (orange area) exhibits the lowest spread (combined within-class variance) and thus is considered optimal. The two modes identify the true positions of the corridor walls (compare to (a)).

Fig. 3. Projection-based geometric feature extraction from a point cloud as described in Sec. V-B.

approach rather than direct model fitting (see Fig. 3). This allows us to reduce the data by one more degree of freedom before extracting the desired statistical features and, thus, further increases speed. We generate a bundle of projection lines centered at the device's position, each with a random orientation. We project the 2D point cloud onto each line and only then apply statistical model fitting on the now one-dimensional datasets.

For each line of projection, we fit a bimodal Gaussian distribution to the projected data points. We employ a technique known as Otsu's Method [28] developed for thresholding images. This allows us to compute the maximum likelihood estimate for a bimodal distribution, i.e., to find the optimal separating threshold between the two modes, without having to explicitly recompute the within-class variances for each possible threshold which would be prohibitively expensive. The maximum likelihood estimate for Gaussian distributions corresponds to the solution that minimizes the combined within-class variance, but can be as well computed using just the cluster means and cardinalities [28]. This way, only very little complexity is required to identify the optimal distribution for each projection line and only at this point we compute the actual within-class variances for each projection line.

The line with the sharpest distribution, i.e., the minimal combined within-class variance, gives the best estimate for the normal to the dominant lines in the point cloud. Along this normal, the two modes of the bimodal distribution mark the positions of the two dominant lines. Further, the distance of the modes defines the distance between the two corresponding opposite dominant planes in the original point cloud, thus representing one of the room's dimensions in vision coordinates. This estimate corresponds to the least-squares solution and, thus, gives the maximum likelihood estimate under the assumption of additive Gaussian noise on the data.

This procedure is iterated as new sensor readings are collected. After the first iteration, the bundle of projection lines is divided into one part that continues to be created with random orientations, and another part that re-attempts the orientation (angle) that was found to be optimal in the previous iteration. More precisely, the second part comprises a narrow fan of projection lines that is centered at the previously optimal angle in order to allow the system to converge to a stable estimate

Fig. 4. Images from the dataset used in our experiment. The indoor environment exhibits various corridor widths, architectural differences and lighting conditions. Feature-rich areas alternate with monotonous and repetitive structures. Motion blur as well as people walking by may perturb tracking, a problem that needs to be considered in real-world conditions.

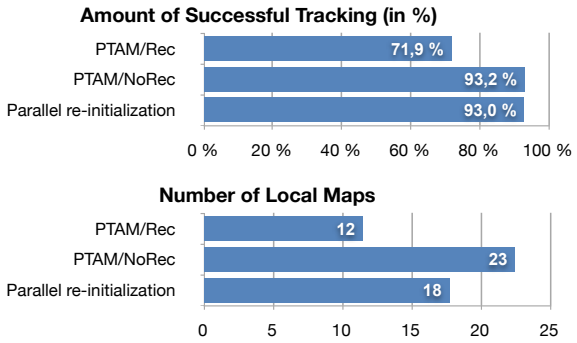


Fig. 5. Averaged results of tracking with four trajectory videos with a total length of 34:31 minutes. The parallel recovery and re-initialization approach needs fewer local maps at the same amount of tracking, compared to conventional PTAM without recovery (PTAM/NoRec).

as new points are being inserted into the map. This mixture of steered and randomly chosen projection lines reduces the number of lines that have to be sampled at each time step considerably, thus lowering computational complexity, while attaining robustness in the presence of severe noise. At each time step, the combined within-class variance of the optimal estimate (i.e., of the distribution on the optimal projection line) defines a measure of uncertainty. Hence, when modifying the state of a Kalman Filter, we change the scale estimate and correctly adapt its uncertainty.

VI. Evaluation

A. Parallel Recovery and Re-Initialization

In order to evaluate the performance of our approach with respect to map recovery and re-initializations (see Sec. III-B), we conduct an experiment with four trajectories using videos from our indoor dataset [29]. The videos have a total length of 34:31 minutes along a track of 680 meters length. The dataset is not trimmed towards "ideal" conditions, but intentionally contains diverse lighting situations, reflections, architectural changes and people walking by (see Fig. 4).

We compare our parallel re-initialization approach with two single-tracking versions of PTAM: The first enters recovery mode in case of lost tracking and has up to four seconds time to recover to the old map before a new map is initialized (grace period). In the following, we call this version PTAM/Rec. The second immediately initializes a new map when tracking is lost (in the following called PTAM/NoRec). For each system, we measure the fraction of time where tracking was successful. Tracking is considered unsuccessful during recovery or the initialization of a new map. We also measure the total number of local maps created.

Results show the expected difference between the two single-tracking PTAM versions (see Fig. 5). PTAM/Rec is in *tracking* state only for 71.9% of the time (standard deviation of individual videos = 0.11). This is due to the system always trying to apply recovery in case of tracking failure. This often fails when the camera has moved on too far and only features disjunct to the previous ones are visible. PTAM/NoRec manages to track the map during 93.2% of the time (standard deviation = 0.02) since in case of tracking failures a new map is initialized immediately and thus no time is lost for recovery attempts. However, this higher tracking time of PTAM/NoRec has the drawback of producing a larger number of local maps: PTAM/NoRec creates 23 maps, while PTAM/Rec only needs 12 since recovery sometimes allows to use the old map. Hence, for single-tracking PTAM, there is a conflict between the competing goals of high tracking time and a low number of maps.

Our approach of parallel re-initialization performs equivalent to PTAM/NoRec in terms of tracking time. It tracks the map for 93.0% of the time (standard deviation = 0.02). However, it needs only 18 local maps, since it has more time for recovery compared to PTAM/Rec. These results indicate that parallel re-initialization combines the advantages of being able to apply a recovery strategy while maintaining a high total tracking time: On the one hand, the system can revert to the old map if necessary and the total number of maps can be reduced. This can be useful, e.g., after motion blur due to a person in front of the camera. On the other hand, when a new map is inevitable, no time for recovery is lost and the no-tracking periods are minimized.

B. Scale Estimation

We evaluate our proposed scale estimation techniques according to three different aspects. We refer to the example of estimating the width of rooms and corridors as we have these data available in our data set. In addition, however, we demonstrate the case where ground truth data are not accessible by the system. First, we show the precision of absolute position estimation when computing the scale by comparing the estimated corridor width to the true building dimensions. Second, we show the reliability of detecting and estimating relative scale jumps that occur due to re-initializations of the visual odometry system. Finally, we use the estimated relative scale factors between consecutive maps to augment a Kalman

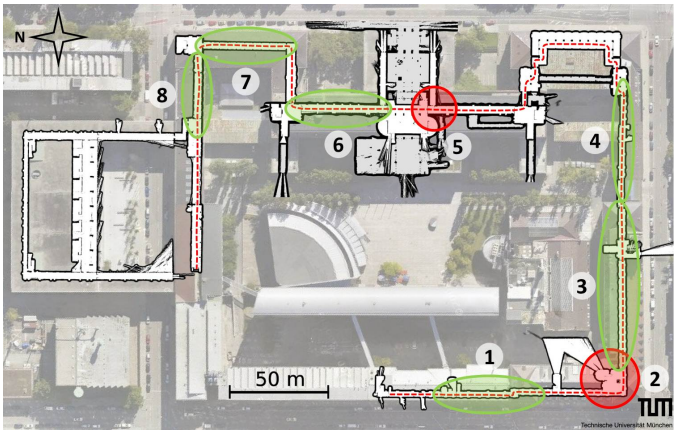


Fig. 6. Our experiments are conducted within the extensive TUMindoor dataset [29] which comprises several floors of a university main building. Videos were recorded along the red trajectory. The numbered areas correspond to the data shown in detail in Figs. 7, 8 9, and 10.

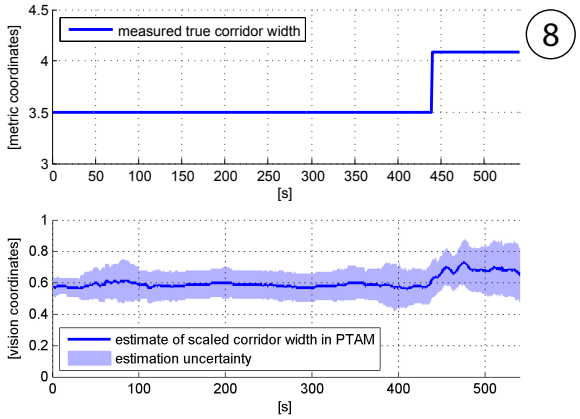


Fig. 7. Top: True corridor width taken from a two-dimensional floor plan. Bottom: Estimated corridor width within the vision coordinate system (PTAM map). Around second 440, the first corridor ends and a larger one begins. The change in width is reliably estimated (cf. Area 8 in Fig. 6).

Filter that tracks the scale as part of its state vector by fusing inertial and visual data.

Fig. 7 shows at the bottom the continuous estimate of the current corridor's width in vision coordinates. At the top, the corresponding true corridor width is given in metric coordinates. Their ratio defines the absolute scale factor between vision and metric coordinates and is used to rescale (point by point) the estimated position trajectory of PTAM. Fig. 8 shows at the top the three components of the device's true trajectory overlaid with the visual odometry data from PTAM after rescaling. The remaining relative position error is given at the bottom. On average, it is reduced from above 80% to below 10%. For this experiment, we tracked the true pose of the device by attaching it rigidly to a mapping trolley as described in [29]. The trolley is equipped with laser range finders to compute its exact pose up to centimeter-precision. In Fig. 6, the corridor's location within our dataset is indicated as Area 8.

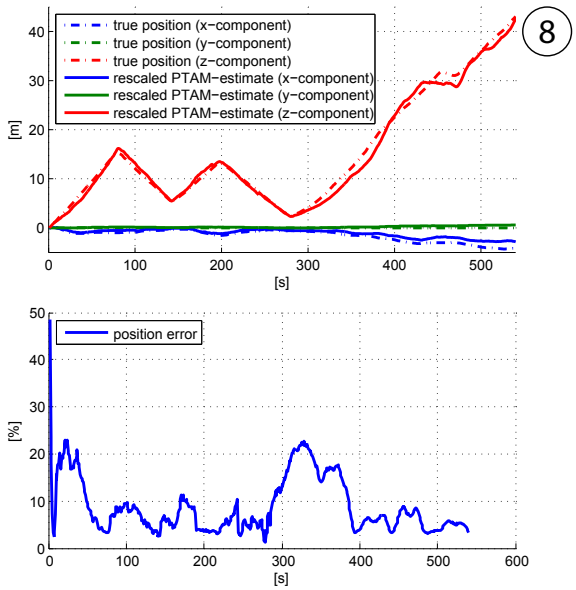


Fig. 8. Top: Rescaled position odometry and ground truth data. The scale factor is estimated from the ratio of the estimated corridor width and the true one. A priori known building dimensions derived from a floor plan are used. Bottom: Successful scale estimation reduces the relative position error from above 80% to below 10% on average (cf. Area 8 in Fig. 6).

Fig. 9 shows several parts of the visual odometry trajectory where re-initializations of the map were necessary. The graphs depict the continuous estimate of the current corridor's width. Green squares denote successful initializations of a new map. Red dots denote unresolvable tracking failures.

For the most part, the width estimation proves very stable and converges fast once the vision system is initialized. Map re-initializations, in contrast, cause significant discontinuities as the scale may change arbitrarily. Due to the fast reaction time and robustness of the width estimation, however, the change in scale can be directly observed as the ratio between the width estimate before and after the re-initialization.

As the proposed method relies on the assumption that the local building structure is mainly regular, it happens that unconventional structure causes the geometric property estimation to fail. A comparison of Figs. 10 and 6 shows that the width estimation failed in the case of a semicircular room (cf. Area 2 in Fig. 6), and where the map re-initialization coincides with the transition between two very dissimilar building parts (cf. Area 5 in Fig. 6). In particular, the second case does not satisfy the assumption that geometric building properties remain locally constant.

For the general case, where there is no a priori knowledge of a building's structure, we employ a state-of-the-art Extended Kalman Filter (EKF) as described in [18]. By fusing inertial and visual data the filter estimates the scale as part of its state vector. Fig. 11 shows the EKF's scale estimate and estimation error over a duration of 10 minutes. For the evaluation, we artificially introduced scale jumps by a factor of two and three

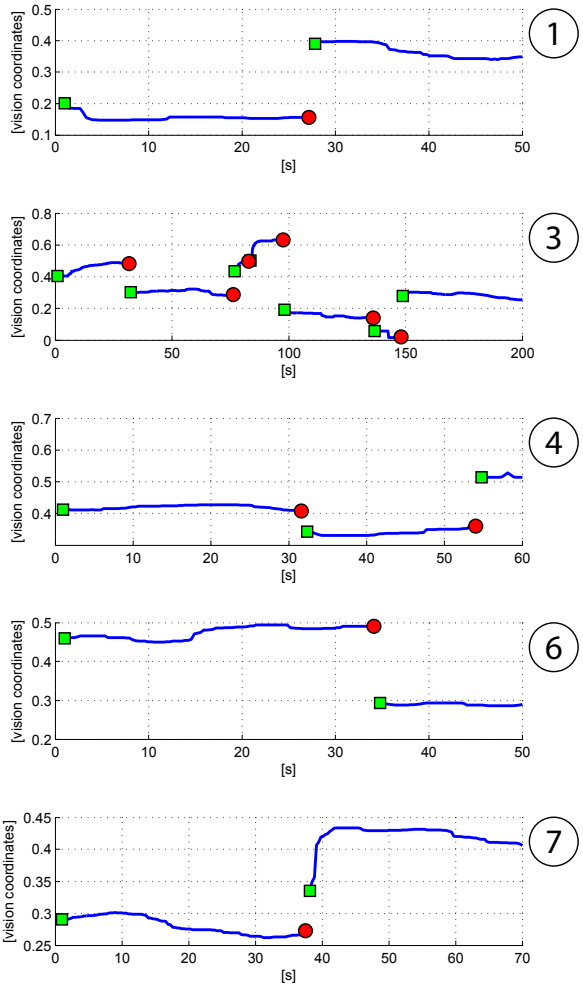


Fig. 9. Continuous estimation of the corridor width (in vision coordinates). Green squares denote successful initializations of a new map; red dots denote unresolvable tracking failures. From top to bottom, the graphs correspond to Areas 1, 3, 4, 6, and 7 in Fig. 6. The reliable estimation allows us to detect scale jumps that occur due to re-initializations of the vision system. At these points, the relative scale factor between consecutive maps can be well observed from the estimated corridor width.

at seconds 200 and 400, respectively.

The graphs 11(a) and 11(b) demonstrate the strong impact of scale changes as the filter expects a nearly stationary scale trajectory. The result is additional convergence time and estimation errors above 20% for prolonged durations. Graphs 11(c) and 11(d), on the other hand, show the augmented EKF's behavior with instant scale update derived from corridor width estimates. Even though the relative scale change is only approximated, the adaptation suffices to prevent significant additional convergence time. The estimation error remains below 20% throughout the experiment.

VII. Conclusion

We present a visual odometry system for long-term robustness and consistency. Using a single camera, we employ visual SLAM to estimate the device's trajectory. As the odometry

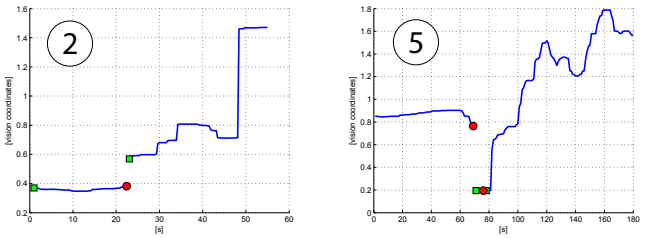


Fig. 10. Two situations where the estimation of the corridor width is not reliable due to locally irregular building structure. Green squares denote successful initializations of a new map; red dots denote unresolvable tracking failures. Left: The map re-initialization occurs in a large stairwell with a partly circular building outline (cf. Area 2 in Fig. 6). Right: The map re-initialization coincides with the transition from a fairly narrow corridor into a large hall with columns (cf. Area 5 in Fig. 6). However, in these cases, the uncertainty measure indicates the temporarily unreliable corridor width estimate (cf. Fig. 7, bottom).

data can only be observed up to scale, an Extended Kalman Filter estimates the scale factor to metric coordinates by fusing inertial and visual data. As tracking failures of the visual SLAM system entail the re-initialization of the local map, they also introduce an arbitrary scale change. In order to minimize the number of re-initializations, we devise a parallel re-initialization and recovery strategy. We demonstrate a substantial reduction of the number of required new maps while maintaining constantly high tracking time in an experiment with extensive real-world indoor data.

For the case where re-initialization is inevitable, we propose to exploit geometric building properties (basic dimensions of corridors or rooms) to determine the resulting relative scale change. We present an efficient and robust projection-based technique to extract these geometric features from a point cloud, i.e., the local map of the vision system. By incorporating these estimated relative scale changes into the state update of a Kalman filter, additional convergence times are effectively eliminated.

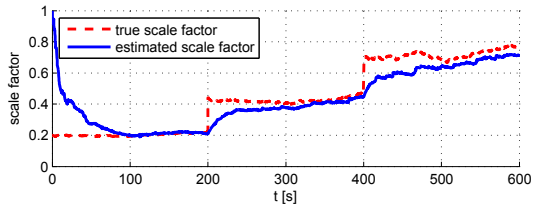
Experimental evaluation using video recordings from our extensive indoor dataset demonstrates the stable and robust extraction of geometric properties. Further, comparing our augmented Kalman Filter to a conventional implementation, the average scale estimation error of the filter is shown to be considerably reduced despite re-initializations of the vision system.

Acknowledgment

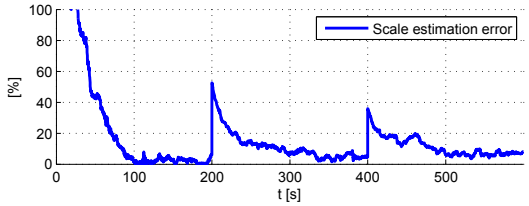
This research project has been supported by the space agency of the German Aerospace Center with funds from the Federal Ministry of Economics and Technology on the basis of a resolution of the German Bundestag under the reference 50NA1107.

References

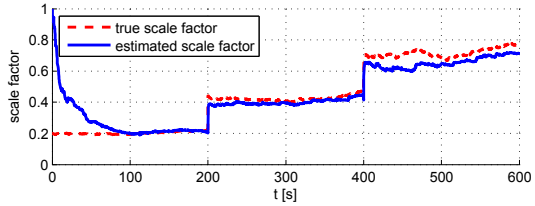
[1] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1067–1080, 2007.



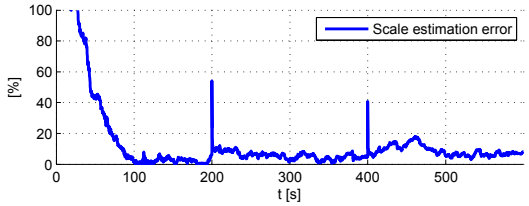
(a) Scale changes due to re-initializations of the vision system require the filter to re-converge which takes time.



(b) The additional convergence time introduces considerable estimation errors.



(c) By updating the filter's scale variable based on geometric building properties the impact of scale changes is drastically diminished.



(d) Previously achieved convergence is preserved despite re-initializations of the vision system.

Fig. 11. Scale estimation where no a priori knowledge on building dimensions is available. Using a Kalman Filter, the scale is estimated by fusing inertial and visual data. Re-initializations of the vision system cause arbitrary scale jumps that are difficult for the filter to follow (a,b) and introduce further convergence time. By computing the scale factor between consecutive maps from geometric building properties, the Kalman Filter can be updated directly and prior convergence is preserved (c,d).

[2] A. Möller, C. Kray, L. Roalter, S. Diewald, R. Huitl, and M. Kranz, "Tool Support for Prototyping Interfaces for Vision-Based Indoor Navigation," in *Proceedings of the Workshop on Mobile Vision and HCI (MobiVis). Held in Conjunction with Mobile HCI*, Sep 2012.

[3] R. Mautz and S. Tilch, "Optical indoor positioning systems," in *Int'l Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2011.

[4] G. DeSouza and A. Kak, "Vision for mobile robot navigation: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 237–267, 2002.

[5] J. Kim and H. Jun, "Vision-based location positioning using augmented reality for indoor navigation," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 3, pp. 954–962, 2008.

[6] A. Mulloni, D. Wagner, I. Barakonyi, and D. Schmalstieg, "Indoor positioning and navigation with camera phones," *Pervasive Computing, IEEE*, vol. 8, no. 2, pp. 22–31, 2009.

[7] F. Boochs, R. Schutze, C. Simon, F. Marzani, H. Wirth, and J. Meier, "Increasing the accuracy of untaught robot positions by means of a multi-camera system," in *Int'l Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2010, pp. 1–9.

[8] S. Cho and C. Park, "MEMS based pedestrian navigation system," *Journal of Navigation*, vol. 59, no. 01, pp. 135–153, 2006.

[9] P. Robertson, M. Angermann, and B. Krach, "Simultaneous localization and mapping for pedestrians using only foot-mounted inertial sensors," in *Int'l conference on Ubiquitous computing (UbiComp)*. ACM, 2009, pp. 93–96.

[10] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, vol. 1. IEEE, 2004.

[11] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart, "Fusion of IMU and vision for absolute scale estimation in monocular SLAM," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1, pp. 287–299, 2011.

[12] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Scale drift-aware large scale monocular SLAM," in *Robotics: Science and Systems*. The MIT Press, 2010.

[13] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Int'l conference on Computer Vision (ICCV)*, 2003, pp. 1403—1410.

[14] R. Munguia and A. Grau, "Monocular SLAM for visual odometry," in *IEEE Int'l Symposium on Intelligent Signal Processing WISP 2007*, Oct. 2007, pp. 1–6.

[15] P. Corke, J. Lobo, and J. Dias, "An introduction to inertial and visual sensing," *The Int'l Journal of Robotics Research*, vol. 26, no. 6, pp. 519—535, 2007.

[16] E. Jones, A. Vedaldi, and S. Soatto, "Inertial structure from motion with autocalibration," in *Workshop on Dynamical Vision*, 2007.

[17] J. Kelly and G. S. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *The Int'l Journal of Robotics Research*, vol. 30, no. 1, pp. 56—79, 2011.

[18] S. Weiss and R. Siegwart, "Real-time metric state estimation for modular vision-inertial systems," in *2011 IEEE Int'l Conference on Robotics and Automation (ICRA)*, May 2011, pp. 4531–4537.

[19] L. Kneip, S. Weiss, and R. Siegwart, "Deterministic initialization of metric state estimation filters for loosely-coupled monocular vision-inertial systems," in *2011 IEEE/RSJ Int'l Conference on Intelligent Robots and Systems (IROS)*, Sep. 2011, pp. 2235–2241.

[20] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Int'l Symposium on Mixed and Augmented Reality*, Japan, November 2007, pp. 1–10.

[21] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European Conference on Computer Vision (ECCV)*, 2006, pp. 430–443.

[22] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–777, Jun. 2004.

[23] H. Stewénius, C. Engels, and D. Nistér, "Recent developments on direct relative orientation," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 60, no. 4, pp. 284–294, 2006.

[24] M. Kranz, C. Fischer, and A. Schmidt, "A comparative study of DECT and WLAN signals for indoor localization," in *8th Annual IEEE Int'l Conference on Pervasive Computing and Communications (PerCom 2010)*. Mannheim, Germany: IEEE, March 2010, pp. 235–243.

[25] R. Huitl, G. Schroth, S. Hilsenbeck, F. Schweiger, and E. Steinbach, "Virtual reference view generation for CBIR-based visual pose estimation," in *ACM Multimedia 2012*, 2012.

[26] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[27] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in *Proceedings of the IEEE Int'l Conference on Robotics and Automation*, vol. 3, apr 1991, pp. 2724–2729.

[28] N. Otsu, "A threshold selection method from gray-level histograms," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 9, no. 1, pp. 62–66, Jan. 1979.

[29] R. Huitl, G. Schroth, S. Hilsenbeck, F. Schweiger, and E. Steinbach, "TUMindoor: an extensive image and point cloud dataset for visual indoor localization and mapping," in *IEEE Int'l Conference on Image Processing (ICIP 2012)*, Orlando, FL, USA, Sep 2012.