Modeling spatial uncertainty of point features in feature-based RGB-D SLAM

Dominik Belter¹ · Michał Nowicki¹ · Piotr Skrzypczyński¹

Received: 29 May 2017 / Revised: 8 February 2018 / Accepted: 29 April 2018 / Published online: 21 May 2018
© The Author(s) 2018

Abstract

This paper deals with the problem of modeling spatial uncertainty of point features in feature-based RGB-D SLAM. Although the feature-based approach to SLAM is very popular, in the case of systems using RGB-D data the problem of explicit uncertainty modeling is largely neglected in the implementations. Therefore, we investigate the influence of the uncertainty models of point features on the accuracy of the estimated trajectory and map. We focus on the recent SLAM formulation employing factor graph optimization. Unlike some visual SLAM systems employing factor graph optimization that minimize the reprojection errors of features, we explicitly use depth measurements and minimize the errors in the 3-D space. The paper analyzes the impact of the information matrices used in factor graph optimization on the achieved accuracy. We introduce three different models of point feature spatial uncertainty. Then, applying the most simple model, we demonstrate in simulations how important is the influence of the spatial uncertainty model on the graph optimization results in an idealized SLAM system with perfect feature matching. A novel software tool allows us to visualize the statistical behavior of the features over time in a real SLAM system. This enables the analysis of the distribution of feature measurements employing synthetic RGB-D data processed in an actual SLAM pipeline. Finally, we show on publicly available real RGB-D datasets how an uncertainty model, which reflects the properties of the RGB-D sensor and the image processing pipeline, improves the accuracy of sensor trajectory estimation.

Keywords SLAM · Uncertainty model · Bundle adjustment · Factor graph optimization

1 Introduction

1.1 Motivation

The compact and affordable RGB-D sensors based on structured light, such as PrimeSense Carmine, Microsoft Kinect and Asus Xtion, fostered the progress in 3-D visual odometry (VO) and simultaneous localization and mapping (SLAM). Visual odometry computes the sensor motion between selected frames of the RGB-D input and recovers the trajectory [37]. However, the trajectory recovered using the frame-to-frame motion estimation has an unavoidable drift, as there are no constraints that enforce the global consistency of sensor motion. Therefore, the VO pipeline treated as a front-end for RGB-D data processing is often paired with an optimization engine (called back-end) to form a SLAM system, which yields globally consistent trajectories. Typically, the back-end post-processes a pose-graph, whose vertices correspond to the sensor poses, whereas its edges represent motion constraints between these poses. Point features

This work was supported by the Polish National Science Centre grant funded according to the decision DEC-2013/09/B/ST7/01583.

M. Nowicki received a doctoral scholarship from the Polish National Science Centre under the Grant 2016/20/T/ST7/00396.

Electronic supplementary material The online version of this article (<https://doi.org/10.1007/s00138-018-0936-9>) contains supplementary material, which is available to authorized users.

✉ Dominik Belter
dominik.belter@put.poznan.pl

Michał Nowicki
michal.nowicki@put.poznan.pl

Piotr Skrzypczyński
piotr.skrzypczynski@put.poznan.pl

¹ Institute of Control, Robotics and Information Engineering,
Poznań University of Technology, ul. Piotrowo 3A,
60-965 Poznań, Poland

are often employed for frame-to-frame motion estimation in pose-based RGB-D SLAM and VO systems [9,18].

Performance depends on the configuration of the image processing module (front-end) and constraints management strategy (back-end) [1]. The pose-based approach keeps only the relative pose-to-pose constraints, marginalizing out the actual 3-D point features; hence, it cannot improve the estimation of motion using the large number of feature-to-pose correspondences established by the front-end. On the other hand, keeping the point features and solving the Structure from Motion (SfM) problem defined as nonlinear least squares optimization allows obtaining very precise sensor trajectories [28]. Feature-to-pose measurements can be used to find coordinates of features and relative sensor motion using the Bundle Adjustment (BA) approach [42]. This approach is applied in the most successful visual SLAM systems: PTAM [21], ORB-SLAM [29] and ORB-SLAM2 [30] to obtain accurate trajectories of the camera. Also RGB-D SLAM systems use the graph-based optimization framework to integrate all feature-to-pose measurements [12,41].

In this research, we also employ factor graph optimization, but instead of minimization of the reprojection error onto images, as in PTAM [21] and ORB-SLAM [29], we directly define the feature position error in the 3-D space. Our approach to RGB-D SLAM exploits to a greater extent the depth measurements, and thus, it seems to be a better vehicle to demonstrate the role of modeling the uncertainty of RGB-D measurements than other systems of similar architecture, such as ORB-SLAM2 [30], which employs triangulation of feature positions and uses reprojection errors even for RGB-D data containing dense depth frames. Most of the published RGB-D graph-based SLAM research neglect the role of the measurement uncertainty model. The information matrices used to represent uncertainty in graph-based optimization are commonly set to identity, which means an equal importance of each measurement and isotropic spatial uncertainty [9,12,30]. From the literature, we know that the depth measurement accuracy depends on the measured distance [13,20]. Thus, the simplified approach with the uniform distribution of measurements is not justified by the underlying physics of the measurements in RGB-D cameras. Therefore, in this work, we investigate how to improve the accuracy of sensor trajectory estimation by explicitly modeling spatial uncertainty of the point features.

1.2 Problem statement

In the SLAM problem, we consider a RGB-D sensor moving freely in the environment and measuring the position of each detected point feature. The factor graph representation consists of vertices representing both the 3-D features and the sensor poses. Edges represent measurements between the poses and the features, or between two poses. The quality (or

importance) of each measurement is represented by an information matrix. The information matrix can be computed by inverting the covariance matrix of the measurement. Thus, more accurate measurements produce stronger constraints in graph optimization [24].

This paper presents methods to compute the information matrices on the basis of the measurement model of the RGB-D sensor, taking into account the additional uncertainty introduced by keypoint detection on the RGB images and dependencies between the geometric structure of the scene, and the resulting spatial uncertainty of the features. Hence, the elaborated uncertainty models we introduce try to capture the spatial uncertainty of point features resulting from the whole processing pipeline in the SLAM front-end.

1.3 Contribution

The contribution of this paper with respect to the state-of-the-art is threefold:

- we introduce a new uncertainty analysis methodology based on software tools that allow us to simulate RGB-D SLAM systems and to analyze the behavior of point features; these tools, in turn, make it possible to understand the nature of the spatial uncertainty of the point features in RGB-D SLAM;
- we propose mathematical uncertainty models for point features in RGB-D SLAM, which are based upon the investigations using the new methodology;
- we verify the suitability of the proposed uncertainty models on real RGB-D benchmark data.

Basic aspects of the new research methodology concerning the spatial uncertainty modeling of RGB-D point features have been introduced in two earlier papers: the workshop paper [5] demonstrated the importance of using the anisotropic spatial uncertainty models in BA-based SLAM exclusively on simulated examples, whereas the conference paper [3] introduced two new uncertainty models and demonstrated their feasibility mostly on synthetic RGB-D data. The aim of this journal article is to cover all aspects of our approach to spatial uncertainty modeling in a unified manner, to analyze in more detail some data processing steps that influence the uncertainty (such as RANSAC), and to extend the analysis of the behavior of features in the map on real RGB-D benchmark sequences (section “Application-oriented Evaluation”). Our ultimate aim is to demonstrate that considering the anisotropic uncertainty of features in factor graph optimization improves the accuracy of sensor trajectory estimation in applications of RGB-D SLAM. Comparing to the conference paper that introduced the new uncertainty models [3], we demonstrate here, using idealized simulations, how large could be the improvement in

Fig. 1 Feature-based SLAM problem represented by the factor graph structure. From each pose of the camera, sparse 3-D point features are observed. The constraints in the graph allow estimating the trajectory of the camera and positions of features simultaneously

trajectory accuracy due to proper models of the uncertainty in the feature-to-pose constraints in a BA-based SLAM system. This provides a motivation for investigating such models for a real SLAM system, and the journal paper demonstrates how particular building blocks of the processing pipeline in the SLAM front-end influence the uncertainty. In particular, we demonstrate this for the RANSAC procedure, which is commonly used in image processing, but as far as we know no one has shown before its influence on the spatial uncertainty of the produced point features. The journal article makes it possible to demonstrate with all the necessary details, that the spatial uncertainty of features should account not only for errors in RGB-D sensor measurements, but for the inaccuracy introduced in the whole process of feature extraction using the RGB images and depth images.

Although we also extend in this article the presentation of the PUT SLAM architecture, we do not consider this particular implementation of RGB-D SLAM a contribution, as it has been presented in detail in [2]. However, we use our own implementation of RGB-D SLAM as a convenient research material, upon which we can implement the software tools necessary for our current work on uncertainty, such as simulations and the reverse SLAM tool, having control on the implementation details.

As we argued in [1], there is a broad diversity of SLAM architectures and implementation details, even if we consider only feature-based systems. Therefore, it is impossible to address the features uncertainty modeling for a generic RGB-D SLAM architecture, as such an architecture hardly exists. We focus on the architectures employing the BA concept, with the structure of the factor graph (in the back-end) based on the feature-to-pose constraints (Fig. 1). In the recent literature [30], this approach is considered as superior to the (perhaps more popular) pose-based SLAM architecture [9]. Hence, the contributed methodology should be helpful for researchers that want to improve the performance of the

recent SLAM architectures. The PUT SLAM, which we use here, belongs to this family of BA-based systems, but was designed for Kinect-like RGB-D sensors, and its back-end optimization procedure minimizes the Euclidean distance errors in feature positions. This approach, different from the more commonly used feature reprojection error in the image plane, is motivated by the fact that the spatial uncertainty of RGB-D sensors can be modeled in the Euclidean space regardless of the depth measurement principle (i.e., active stereo vision or time-of-flight), which potentially makes the methodology proposed in this article universal with respect to the first and second generation of the RGB-D sensors [23].

The remainder of this paper is organized as follows: Section 2 presents the most relevant previous work in the areas of uncertainty modeling for RGB-D sensors and the uncertainty models in feature-based SLAM, while Sect. 3 details our PUT SLAM system, used through the paper as a reference architecture and a tool to investigate the uncertainty of features. The three approaches to modeling the uncertainty of point features in RGB-D SLAM are introduced in Sect. 4, and followed by the description of the simulation and visualization tools in Sect. 5. These tools are used to investigate the behavior of the point features depending on the uncertainty model and the noise characteristics of the RGB-D data. A quantitative evaluation of the results of applying the proposed uncertainty models in PUT SLAM on synthetic RGB-D data is provided in Sect. 6, while Sect. 7 demonstrates how the selected best model influences the accuracy of sensor trajectory estimation in PUT SLAM tested on real RGB-D data from benchmark sequences. Section 8 concludes the paper and sets outlook on the future research directions.

2 Related work

The uncertainty of depth and combined RGB-depth measurements from sensors based on the PrimeSense structured light technology was investigated in a number of recent papers. Khoshelham and Elberink [20] studied the accuracy and resolution of depth data from Kinect sensor. Gonzalez-Jorge et al. [14] demonstrated that the metrological characteristics in terms of accuracy and precision are almost independent on the type of sensor (Kinect or Xtion) due to the use of the same PrimeSense depth camera. Recently, research on the noise characteristics in Kinect v1 depth data was surveyed in [25]. The first-generation RGB-D sensors have been also compared to the Microsoft Kinect v2 based on the time-of-flight principle [13]. This research revealed that correlation used in the PrimeSense technology to compare the observed pattern of “speckles” to a reference pattern creates dependency in-between pixels in the depth images, which in turn causes errors in the range measurements. The papers dealing with uncertainty in RGB-D sensors focus mostly on appli-

cations outside of VO/SLAM, but Park et al. [36] proposed a mathematical uncertainty model for Kinect v1 sensor and RGB-D sparse point features. However, the approach of [36] was demonstrated without an application in real SLAM or VO. Although the structured light depth measurement principle in the most common RGB-D sensors may be considered as active stereo vision, the existing literature on uncertainty modeling in stereo vision considers mainly systems, in which the stereo matching is applied to discrete features, and no dense depth map is created [7,27]. An exception is the older work of Matthies and Shafer [26], who applied 3-D Gaussians to model the measurement errors in the discrete digital images, and demonstrated that propagating the uncertainty in the form of covariance matrices enables reduction in uncertainty in the localization task.

In spite of these results, up to now there is little work being done on the utilization of physical characteristics of RGB-D sensors in VO/SLAM. In contrast, the feature uncertainty modeling is widely used in SLAM research employing the extended Kalman filter. In these frameworks, the covariance from the sensor measurement model is propagated to feature model for both 2-D laser sensors [39] and 3-D vision [6]. In the work of Oskiper et al. [34] the uncertainty of the observed features (called “landmarks”) is explicitly modeled taking into account the stereo imagery processing pipeline and using the method from [26]. However, as far as we know, only Dryanovski et al. [8] formulated an uncertainty model of point features used to register a Kinect sensor pose in relation to a map of features which are estimated using Kalman filter. This model, based on the Gaussian mixture, was motivated by experimental assessment of the Kinect sensor depth measurements uncertainty. In graph-based SLAM, Endres et al. [9] applied the depth measurement model from [20] in the motion estimates verification procedure for their feature-based pose-graph RGB-D SLAM. The possibility of using the Mahalanobis distance that takes into account the uncertainty instead of the Euclidean distance in the pose-to-pose motion estimate computation is also mentioned in [9], but the paper provides no clear description of the uncertainty model being used. Conversely, the feature-based RGB-D visual odometry system presented in [18] minimizes feature reprojection error in the image space in order to compute the pose-to-pose motion estimates. This approach implicitly takes into account the fact that the uncertainty increases with range. Nguyen et al. [31] applied a depth uncertainty model of Kinect in a VO system that employs dense depth data. Although the improved accuracy of the recovered trajectories was shown in [31], and the dense depth-based approach generally achieves impressive results in terms of environment map reconstruction [44], it cannot model uncertainty of all the individual range measurements and then propagate this uncertainty to the dense map.

The RGB-D SLAM formulation used in this paper is similar to the structure from motion problem in computer vision, which is commonly solved applying the Bundle Adjustment (BA) [42]. Advanced, keyframe-based BA variants, such like the one implemented in PTAM [21], may be used for online mapping and motion estimation in robotics [4]. However, while there were some efforts to incorporate the uncertainty of feature points in the BA, much of the computer vision literature simply assumes one-pixel Gaussian noise in the location of features [17]. For example, Konolige and Agrawal [22] use such uncertainty model in the pose-based SLAM utilizing visual imagery. Ozog and Eustice [35] demonstrate that accounting for the uncertainty of the relative-pose transformation, computed using the Haralick’s method [16], in the two-view sparse BA improves the relative motion estimation between two image frames. Our approach can be considered similar to BA because the trajectory of the sensor and feature position are simultaneously optimized over sensor measurements. Such an approach is not only superior to the pose-based RGB-D SLAM formulation [9], but also gives better accuracy and robustness than dense/direct methods in visual SLAM [11] and RGB-D SLAM [19]. The direct methods are more prone to image distortions and artifacts due to such factors as the rolling shutter or automatic white balance because they need to model the whole image acquisition process that influences the pixel intensities [10]. Conversely, the feature-based approach employed in PUT SLAM enables this system to take great advantage from modeling of the uncertainty, because the uncertainty of features directly influences the strength of the constraints in optimization. As so far, other feature-based RGB-D SLAM systems of similar architecture, such as [12] and [41], did not attempt to model this uncertainty.

3 RGB-D SLAM with a map of features

We consider spatial uncertainty models of point features treating the SLAM algorithm itself as a “black box” that processes the measurements (constraints) and depends on the provided description of the “importance” of these measurements in the form of uncertainty model. Although the general structure of the BA-based SLAM algorithms is similar with respect to the main data processing components, there is no SLAM standard, generic architecture. Therefore, we describe here the architecture of our PUT SLAM system [2,3]. This brief description should make it easier to understand some of the mechanisms that are responsible for the spatial uncertainty of the features (e.g., matching and RANSAC), but is also necessary to introduce the software used to investigate the behavior of features, which is based on PUT SLAM (see Sect. 5).

Fig. 2 Architecture of the RGB-D PUT SLAM system: front-end and back-end module for image processing, graph optimization and map management, respectively

The architecture of the PUT SLAM system is presented in Fig. 2. PUT SLAM is a BA-based RGB-D SLAM that maintains a map of the environment. The map consists of a sequence of camera poses and a set of 3-D features. The so-called factor graph used for optimization and camera state estimation is formulated from the map data. The vertices in the graph are camera poses and observed features. These vertices are connected by measurements (feature-to-pose constraints). The graph optimization is running in a separate thread. The map and factor graph are synchronized after each iteration of the optimization process. This architecture requires more attention on implementation, but allows to efficiently use a multi-core CPU.¹

The point features $\mathbf{f}_j$ and camera poses $\mathbf{c}_i$ are represented by vertices in the factor graph. The constraints (measurements) are represented by edges in the graph. The measurement $\mathbf{m}_{ij}$ between the i -th pose and the j -th feature is represented by the 3-D edge $\mathbf{t}_{ij} \in \mathbb{R}^3$. The rigid body transformation (odometry) between two camera poses denoted as i and k is represented by the edge $\mathbf{o}_{ik} \in \mathbf{SE}(3)$, where $\mathbf{SE}(3)$ is the special Euclidean group that defines rotation and translation of a rigid body with respect to six degrees of freedom. To find the optimal sequence of the sensor poses $\mathbf{c}_1, \dots, \mathbf{c}_n$ and feature positions $\mathbf{f}_1, \dots, \mathbf{f}_m$ the following cost function is minimized:

$$\argmin_{\mathbf{f}, \mathbf{c}} F = \sum_{i=1}^n \sum_{j=1}^m \mathbf{e}(\mathbf{c}_i, \mathbf{f}_j, \mathbf{m}_{ij})^T \boldsymbol{\Omega}_{ij}^t \mathbf{e}(\mathbf{c}_i, \mathbf{f}_j, \mathbf{m}_{ij}), \quad (1)$$

¹ Source code is at <https://github.com/LRMPUT/PUTSLAM/tree/release>.

where $\mathbf{e}(\mathbf{c}_i, \mathbf{f}_j, \mathbf{m}_{ij})$ is an error function computed for the estimated and measured pose of the vertex. The measurement $\mathbf{m}_{ij}$ is 3-D transformation $\mathbf{t}_{ij}$ for feature-to-pose or $\mathbf{SE}(3)$ transformation $\mathbf{o}_{ij}$ for pose-to-pose constraints. However, the pose-to-pose constraints are added to the graph only if the number of matches between map features and features from the current frame is below a given threshold [2]. Pose-to-pose constraints stabilize the factor graph optimization process in case of an insufficient number of feature-to-pose constraints. The g^2o general graph optimization library [24] with the implementation of Preconditioned Conjugate Gradient method (PCG) is used to solve (1).

The accuracy of each feature-to-pose is represented in the factor graph by an information matrix $\boldsymbol{\Omega}_{i,j}^t$. The information matrix can be obtained by inverting the covariance matrix of the measurement. The information matrix $\boldsymbol{\Omega}_{i,j}^o$ for the pose-to-pose edge in the factor graph is set to an identity matrix, as in [9].

3.1 SLAM front-end

We implemented a front-end of the SLAM algorithm to verify our method on real-life RGB-D sequences. We use standard procedures for feature detection, description [38], matching, and tracking [1]. The front-end starts from detection of salient features on the RGB-D frame. The set of detected features is used to estimate frame-to-frame sensor displacement. The PCG solver used in the back-end requires a good initial guess. Therefore, we provide a reliable sensor displacement guess from the VO pipeline. We implemented a fast VO algorithm [33], which is independent of the map structure. In our investigations, we consider two configurations of the VO pipeline. In the first one, the associations between two consecutive RGB frames are found using SURF descriptors. However, the SURF descriptors are slow to compute and match [38]. Thus, we alternatively implemented fast sparse optical flow tracking with the Kanade–Lucas–Tomasi (KLT) algorithm [33]. In this case, ORB keypoint detector is used. Regardless of the method used to establish associations between features belonging to two frames in the sequence, the $\mathbf{SE}(3)$ transformation is computed from the paired 3-D points. To estimate the camera motion from this set of paired features, we apply the Umeyama algorithm [43] and pre-emptive RANSAC to remove outliers. The camera pose estimated from VO pipeline is used as an initial guess for the camera pose in the graph optimization. The constraints between new camera pose and features in the map are obtained by matching of features from the last frame and features projected from the map. Again, we use RANSAC, to determine the set of inliers. The features that are observed, but cannot be associated with the features projected from the map are added to the map, extending the environment model to newly discovered areas.

Fig. 3 2-D example which shows the results of graph optimization with an anisotropic uncertainty model of the measurements. Optimization with anisotropic uncertainty model allows to minimize Mahalanobis distance and reduce the error between estimated and real position of the feature

4 Spatial uncertainty modeling

In the BA-based approach to RGB-D SLAM, the importance of constraints defined by the measurements of feature positions is defined by their information matrices, that are directly related to the spatial uncertainty of these features. To show the advantages of using anisotropic uncertainty model in the BA-based formulation of SLAM, we consider a simplified 2-D case presented in Fig. 3. The position of a single feature is measured from two different positions of the camera. For each measurement, we draw the real uncertainty ellipsoid (filled-in with a gradient color). The measured position of the feature is inside the error ellipse for each measurement. To find the position of features from noisy measurements t_{11} and t_{21} , we construct a graph. The information matrices of the links in the graph Ω_{11}^t and Ω_{21}^t are computed from the inverse of the covariance matrix. The computed uncertainty (dashed line ellipses) differs to the real uncertainty of measurements. However, the optimization with anisotropic uncertainty model gives the results at least one order of magnitude better than minimization with identity information matrices [5]. In the first case, the Mahalanobis distance is minimized (1), while in the second example the distance in (1) degrades to the Euclidean distance, which is then minimized.

The classic approach to capture the spatial uncertainty in features used by SLAM systems is to propagate the uncertainty of sensor measurements through the data processing pipeline into the covariance matrices of the features [39]. This approach was also used with RGB-D sensors [8] and was the first one we attempted to use in our BA-based PUT SLAM to test whether an anisotropic uncertainty model improves the accuracy of estimation. However, realizing that this approach is insufficient in a system that uses RGB-D data processed in a complicated and non-deterministic (due to RANSAC) pipeline, we propose to develop new uncertainty models that accumulate the spatial uncertainty introduced in the front-end. The new models are implemented by analytically modeling the distribution of features observed a posteriori, rather than by propagating the uncertainty from the sensor model.

Fig. 4 Influence of the RANSAC outliers rejection threshold on the uncertainty of features: 2-D features observed from two different positions and their uncertainty ellipses (a, b), and the circle-shaped distributions (thick dashed lines) imposed by using RANSAC (c)

This new idea allows us to develop feasible uncertainty models even without the knowledge of the processing pipeline details. We also do not need to approximate the uncertainty propagation through non-differentiable or even non-analytical processing stages, as in the classic method [16].

For instance, the distribution of feature measurements is significantly influenced by the RANSAC procedure, which is used in the front-end to remove outlier matches. The influence of RANSAC on the spatial uncertainty of measurements is presented in Fig. 4. The set of features is observed from two different sensor poses (Fig. 4a, b). The uncertainty of measurements is modeled using an anisotropic model. However, if RANSAC is used to remove outliers, as in the front-end of PUT SLAM, the distribution of measurements is changed (Fig. 4c). In RANSAC, the Umeyama algorithm [43] is used to find a transformation between two sensor poses. Outlier measurements which are not consistent with the found transformation are removed from the set of inliers (feature f_3 in Fig. 4c). The RANSAC outlier threshold is defined here as the Euclidean distance between the expected and measured position of the feature. Thus, all the inlier measurements are inside a sphere defined by the RANSAC outlier threshold. In this case, the uncertainty can be modeled also as a sphere, and the information matrix for each feature should be set to the identity matrix in the graph.

In these investigations, we use the synthetic ICL-NUIM data set [15], as we need perfect ground truth sensor trajectories to isolate the spatial uncertainty introduced by processing the RGB-D frames in the front-end. Moreover, the ICL-

Fig. 5 View-dependent uncertainty model C_p of feature measurement: ellipsoid which represents the spatial uncertainty model (a), distribution of measured feature positions (b), and visualization of uncertainty ellipsoids (red color) for selected 3-D point features (c)

NUIM offers an insight into the nature of the noise introduced into the sensory data. Hence, using this dataset it is possible to draw clear conclusions as to the dependencies between the characteristics of the sensory data uncertainty and the behavior of the uncertainty model.

4.1 Uncertainty propagation from the sensor model

The error of depth measurements based on the PrimeSense technology increases with the distance to the observed object (axial noise) [20]. The uncertainty of measurements is also influenced by the lateral resolution of the sensor (lateral noise) [25,31]. The uncertainty model of depth measurements can be generalized for both Kinect and Xtion devices [14]. Moreover, the spatial uncertainty in the location of the RGB-D point feature depends on the inaccuracy introduced by the keypoint detector used in the front-end of the SLAM system [36]. Thus, the spatial uncertainty model of measurements is anisotropic and cannot be captured by the identity information matrices commonly applied for graph optimization in the back-end of SLAM systems.

To compute the information matrix Ω_{ij}^t of a feature-to-pose constraint, we have to determine the covariance matrix of the RGB-D point feature. The feature uncertainty model based on the concept of propagating the uncertainty from the sensor measurements is based on the approach proposed by Park et al. [36]. The sensor model defines the relation between the position of a feature in the image, and position of this feature in the 3-D space:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = d \cdot \begin{bmatrix} \frac{1}{f_x} & 0 & -\frac{x_c}{f_x} \\ 0 & \frac{1}{f_y} & -\frac{y_c}{f_y} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \quad (2)$$

where x_c, y_c define the position of the optical axis on the image plane, and f_x, f_y are focal lengths of the camera. The covariance matrix $C_{p(3 \times 3)}$ of each discovered feature is computed as follows:

$$C_p = J_p \cdot C_k \cdot J_p^T, \quad (3)$$

where $J_{p(3 \times 3)}$ is Jacobian of (2) computed with respect to the intrinsic coordinates of the point feature u, v and d , while $C_{k(3 \times 3)}$ is the covariance matrix of feature measurement u, v and depth d . Following the approach common in computer vision, we assume that the measurements of u and v coordinates in the camera are independent. This assumption and the fact that the uncertainty in d is caused by a physically different measurement channel allow us to write C_k as a diagonal matrix:

$$C_k = \begin{bmatrix} \sigma_u & 0 & 0 \\ 0 & \sigma_v & 0 \\ 0 & 0 & \sigma_d \end{bmatrix}. \quad (4)$$

As demonstrated by Park et al. [36], the variance values σ_u, σ_v can be considered constant for the given camera parameters and the chosen feature detection algorithm. However, the variance σ_d of depth d measurement increases with the distance from the camera. We use the approximation of σ_d found experimentally by Khoshelham and Elberink [20]:

$$\sigma_d = k_1 \cdot d^3 + k_2 \cdot d^2 + k_3 \cdot d + k_4, \quad (5)$$

where $k_1, \dots, k_4$ are constants ($k_1 = 0.57, k_2 = 0.89, k_3 = 0.42, k_4 = 0.96$). The components given by equations (3), (4) and (5) sum up to a simple C_p -model of the spatial uncertainty of a point feature. This model is view-dependent, i.e., the shape of the uncertainty ellipsoid for a given feature changes when it is seen by the sensor from a different viewpoint. A 3-D visualization of the C_p -model is presented in Fig. 5.

4.2 Normal-based uncertainty model

Realizing that propagating the uncertainty of the sensor measurements to the uncertainty of features is not enough to capture the uncertainty sources in the SLAM front-end, we analyzed the distribution of point features produced by PUT SLAM using the reverse SLAM tool. In order to isolate the effects of imperfect detection of point features, we started with the ICL-NUIM sequences without depth noise. We noticed that most measurements of the feature positions are located on the object surfaces and form flat ellipsoids (Fig. 6b and compare Fig. 10a). The minor axis of the ellipsoids is correlated with the normal vector to the surface. To capture

Fig. 6 Normal-based feature uncertainty model: ellipsoid which represents the spatial uncertainty model (a), distribution of measured feature positions (b), and visualization of uncertainty ellipsoids (red color) for selected 3-D point features (c)

this distribution of measurements, we propose the normal-based uncertainty model C_n . The normal-based uncertainty model is view-independent. It means that the global shape of the uncertainty ellipsoid does not depend on the position of the camera.

To determine the ellipsoid of the uncertainty model, we compute a normal vector to the point located in the feature coordinates $f_{u,v}$. To this end, we used depth image only [3]. Then, we compute the rotation matrix R . The z axis of the coordinate system R coincides with the surface normal n (Fig. 6a). The x and y axes are selected to form a right-handed coordinate system (Fig. 6b). The covariance matrix C_n is defined as:

$$C_n = R \cdot S \cdot R^{-1}, \quad S = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & S_z \end{bmatrix}, \quad (6)$$

where S is a scaling matrix. In this model, we scale the minor axis of the ellipsoid z , which is related to the surface normal n . The scaling coefficient S_z is in the range $(0,1)$. The diagonal elements of the matrix S_x and S_y are set to 1. Example uncertainty ellipsoids of the normal-based uncertainty model C_n for selected 3-D point features are presented in Fig. 6c.

4.3 Gradient-based uncertainty model

The C_n uncertainty model leverages the role of visual localization of the point features in the RGB frames. However,

Fig. 7 Gradient-based feature uncertainty model: ellipsoid which represents the spatial uncertainty model (a), distribution of measured feature positions (b), and visualization of uncertainty ellipsoids (red color) for selected 3-D point features (c)

the uncertainty of the feature location depends not only on the performance of the features detector implemented in the SLAM front-end pipeline and the quality of the RGB images, but also on the local structure of the scene. During visual analysis of the distribution of point features, we observed that features located in the vicinity of strong intensity gradients (photometric edges) slide along lines defined by these gradients. Thus, we propose another uncertainty model, which includes the observed behavior. In the proposed uncertainty model, the major axis of the ellipsoid is located along a photometric edge defined by strong intensity gradient in the RGB image (Fig. 7b and compare Fig. 10b). To compute the uncertainty matrix, we detect the RGB edge using a 3×3 Scharr kernel. The direction of the edge in 3-D space is computed using the depth data. Then, the procedure is similar to the procedure presented for the normal-based uncertainty model. We construct the rotation matrix R representing the local coordinate system. The z axis is related to the RGB gradient vector (Fig. 7a). The x axis of the coordinate system is located on the RGB edge (Fig. 7a and b). The RGB edge might be related to the edge of an object or a photometric edge on a flat surface. The covariance matrix C_g is computed using (6). We scale the x , y and z axes of the ellipsoid using the scaling matrix S . Gradient-based uncertainty ellipsoids (according to the C_g -model) for selected 3-D point features are presented in Fig. 7c.

5 Tools that give insight into the uncertainty of features

5.1 Experiments in Simulation

This section describes two software environments (programs) that make it possible to observe and measure some variables and quantities that are hard, or even impossible, to be observed and measured in a typical SLAM system. The common idea behind these tools is to replace particular components of the investigated system (here PUT SLAM) by equivalent modules that use synthetic or ground-truth-based measurements. This idea allows us to investigate particular aspects of the spatial uncertainty in SLAM, ignoring those aspects, that normally are out of control in fully experimental work. As demonstrated in our previous work [5], simulation-based experiments allow us to isolate errors introduced by wrong feature correspondences or multiplicated features, and to focus on the analysis of spatial uncertainty introduced by RGB-D measurements. The simulation environment replaces the real PUT SLAM front-end, providing the back-end with features and measurements (constraints) that are free from qualitative errors introduced by a real front-end [2]. In the simulator, we define the environment with a set of 3-D point features described by unique identifiers. The identifiers are used in the matching of the features observed from various viewpoints. The position measurement of a selected feature is generated by randomly drawing a point from the uncertainty distribution ellipsoid defined around the nominal position of a feature according to the chosen uncertainty model. To generate a sequence of measurements for graph optimization, we simulate the motion of the sensor along the given trajectory. Then, we compute the position of each feature in the camera frame ${}^c\mathbf{f}_j$:

$${}^c\mathbf{f}_j = (\mathbf{c}_i)^{-1}\mathbf{f}_j, \quad (7)$$

where $\mathbf{c}_i$ is the global sensor pose and $\mathbf{f}_j$ is the global position of the j -th feature. To check whether the feature is within the range of the sensor, we compute the projection of the 3-D feature on the image plane $[u, v, d]^T$:

$$\begin{bmatrix} u \\ v \\ d \end{bmatrix} = \begin{bmatrix} \frac{x f_x}{z} + x_c \\ \frac{y f_y}{z} + y_c \\ z \end{bmatrix}. \quad (8)$$

If the projection of the 3-D point lies on the image plane (640×480) and within the range of the sensor, the covariance matrix is computed using (3). In the simulation experiments, we use the $\mathbf{C}_p$ -model to define the uncertainty.

In the first experiment, we simulated a box-like environment, which represents a room ($5.5 \times 5.5 \times 5.5$ m). We randomly generate 1000 point features on the surface of each

Fig. 8 Trajectories obtained in the simulation in a box-like empty room. The sensor is facing the direction of motion and rotates in the corners

wall. The camera moves along a rectangular reference trajectory inside the room (Fig. 8). At each corner of the room, the sensor is rotated by 90° in small increments so the optical axis of the sensor (z axis) is in the direction of forward motion. This configuration of the sensor is most common, but at the same time, it is the most challenging one for a SLAM system [5]. This is caused by the fact that the usually large axial uncertainty of RGB-D sensor measurements is, in this case, directed most of the time along the estimated trajectory of the robot. Moreover, when the robot rotates it observes significantly less common features on consecutive images. As we have observed in [5], a ceiling-facing camera is more advantageous, if appropriate features are available. We have presented a more thorough analysis of the features observation scenarios in the earlier research [5].

The trajectories obtained in simulation in the room environment are shown in Fig. 8. When VO method is used, the position drift is not canceled by a loop closure procedure. The error accumulates and the estimated trajectory is far from the trajectory of the sensor. The application of SLAM paradigm (i.e., loop closure) allows canceling the drift when the features from the beginning of the experiment are re-observed (loop closure).

An experiment in simulation allows us not only to assess the accuracy of the trajectories but also to compare the trajectories and feature-based maps obtained with and without the uncertainty model. We use simulations to confirm the hypothesis, that modeling of spatial uncertainty in feature-based PUT SLAM improves the accuracy. To obtain statistics, we run the simulation 100 times for each investigated configuration. The quantitative results are computed according to the methodology introduced in [41], using the Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) metrics. The distance between corresponding points of the estimated and ground truth trajectories is measured by ATE, whereas the RPE metric reveals the local drift of the trajectory. We can obtain the translational or rotational RPE, taking, respectively, the translational or rotational part of the homogeneous matrix computed when comparing the sensor poses along the

Table 1 ATE and RPE for the front-view sensor experiment

Method	ATE RMSE (m)		RPE RMSE (m)	
	Mean	SD	Mean	SD
VO [1]	0.1370	0.0500	0.0247	0.0027
Pose-based SLAM [1]	0.1113	0.0450	0.0231	0.0023
PUT SLAM identity matrix	0.0172	0.0014	0.0222	0.0022
PUT SLAM C_p uncertainty	0.0043	0.0164	0.0049	0.0237

Fig. 9 Simulation results in the environment and trajectory from ICL-NUIM office/kt1 sequence

trajectories. As ATE compares absolute distances between two rigid sets of points representing the estimated and the ground truth trajectory, it is always expressed as a metric distance. With the uncertainty model, the ATE error is 4 times smaller while the RPE error is about 4.5 times smaller (Table 1). Details of the simpler localization systems: VO and pose-based SLAM included in Table 1 for comparison have been described in [1], while the approach used to run these methods in the simulation was the same as for PUT SLAM.

It is worthwhile noticing that even though the error is smaller for PUT SLAM with uncertainty model in comparison with PUT SLAM without uncertainty modeling, the standard deviation is greatly higher. In the series of 100 trials, the one solution from 100 optimization procedures is incorrect. One camera pose from the whole trajectory optimized

by g^2o is incorrect, and the mean ATE error of the trajectory is significantly higher [5]. We notice this behavior when the camera observes only a flat wall and the number of pose-to-features measurements is small. In this case, the graph-based back-end sometimes finds a solution, which is incorrect but satisfies measurements from the given camera pose. This situation is difficult to detect in practice. Thus, whenever the number of measurements is smaller than a threshold an additional pose-to-pose constraint is added to the graph to stabilize the optimization process in the back-end.

In the next simulation, we use the `kt1` sensor trajectory from the ICL-NUIM dataset (Fig. 9) to confirm the results in a more realistic scenario. The positions of features are obtained offline from the PUT SLAM front-end processing the ICL-NUIM office sequence of RGB-D frames. The 86 extracted features are fixed in the simulator and augmented by unique IDs that enable the simulator to perfectly associate the observed features to the map. The results summarized in Table 2 confirm that the feature-based PUT SLAM yields more accurate trajectory estimate than the pose-based SLAM implemented as in [1]. Moreover, the anisotropic feature uncertainty model allows PUT SLAM to achieve much smaller ATE than it was possible using identity matrices.

The simulation experiments give us also the possibility to verify the accuracy of the obtained map (positions of features). Unfortunately, the available benchmark datasets produced using real RGB-D sensors contain ground truth sensor trajectories, but no ground truth for the map. Hence, a quantitative comparison of the map is not possible, and to demonstrate the accuracy gains in the location of map features, we also use the synthetic ICL-NUIM dataset. In the simulation, we compute the mean squared error (MSE) for (known and fixed) features in the map. The results are presented in Table 3. VO and pose-based SLAM give similar

Table 2 Numerical results (ATE and positional RPE) obtained in the ICL-NUIM office/kt1 environment

Method	ATE RMSE (m)		RPE RMSE (m)	
	Mean	SD	Mean	SD
VO [1]	0.1388	0.0403	0.0924	0.0041
Pose-based SLAM [1]	0.1356	0.0400	0.0918	0.0043
PUT SLAM identity matrix	0.0708	0.0049	0.0916	0.0040
PUT SLAM C_p uncertainty	0.0142	0.0011	0.0187	0.0016

Table 3 Mean squared error (MSE) for the map in the ICL-NUIM office/kt1 environment

Method	MSE (m^2)	
	Mean	SD
VO [1]	0.2080	0.2272
Pose-based SLAM [1]	0.2012	0.2162
PUT SLAM, identity matrix	0.0240	0.0242
PUT SLAM, C_p uncertainty	0.0006	0.0001

results. The MSE value is greater than the noise introduced by individual measurements. This comes from the fact that a feature position is measured in the current sensor frame and the sensor pose error accumulates. Because the sensor pose drifts, also the estimated positions of features differ significantly from their real positions. However, in the BA-based PUT SLAM, the positions of features are jointly optimized with positions of the sensor. This allows us to reduce the MSE errors for features and improve the quality of the map. Hence, by introducing the spatial uncertainty model of features the feature-based PUT SLAM achieves much smaller estimation errors not only for the sensor trajectory but also for the map of features.

5.2 Reverse SLAM: looking closer at the features

We demonstrated in the simulation that an anisotropic spatial uncertainty model of features improves significantly the accuracy of estimation in factor graph-based SLAM. However, we found it difficult to apply the approach to uncertainty modeling proposed by Park et al. [36] in the actual PUT SLAM working with real data. When the C_p -model is used, the optimization in g^2o becomes unstable or returns worse results than the optimization with identity matrices. This suggests that the C_p covariance matrix does not represent correctly the uncertainty of real measurements.

In practice, the uncertainty of features is also influenced by the tracking/matching algorithm [33]. Therefore, in order to investigate how the proposed uncertainty model fits to the distribution of real measurements we have developed the *reverse SLAM* tool.² In the reverse SLAM tool, we are looking for the distribution of feature measurements that is obtained in the front-end when SLAM returns a perfect trajectory. Thus, we move the sensor along the known trajectory and run the front-end of the SLAM. At each frame, we provide the real position of the camera instead of its estimate. Finally, the sensor noise and image processing errors accumulate in the representation of features, not in the trajectory error. Assuming that an accurate ground truth trajectory is available, a distribution of

Fig. 10 Illustration of measurement distributions observed in the *reverse SLAM* tool: measured position of features is located on the surface of the objects (a) or along RGB edge (b). The enlarged area presents 3-D distribution of measurements and surface of the object (the cabinet and the desk)

measurements (3-D positions of features) for each feature in the map can be computed. Unfortunately, datasets obtained using real RGB-D sensors and external motion capture techniques typically suffer from synchronization issues between the RGB-D data and the ground truth trajectory. For instance, in the TUM RGB-D Benchmark [41] data from the Vicon motion capture system, and data from the Kinect sensor are not well synchronized due to the different sampling frequencies [23]. This dataset cannot be used in reverse SLAM tool because errors that usually manifest themselves only when computing benchmarking values for the particular trajectory [30], in reverse SLAM may change significantly the distribution of the features, hence leading to wrong conclusions as to the uncertainty model. Therefore, we use the synthetic ICL-NUIM dataset [15] in experiments with reverse SLAM. This dataset provides not only perfect ground truth trajectory

² Video is available at <https://www.youtube.com/watch?v=tP5I5IDBYLw>.

Fig. 11 Distribution of measurements on the image plane for the first frame of the ICL-NUIM *office/kt1* sequence extracted by the tracking-based version of PUT SLAM front-end. Measured positions of different features projected on the image plane are indicated by points in different colors

ries of the sensor, but also perfect synchronization between the RGB-D frames and trajectory points.

The output from the reverse SLAM tool gives us the information about the real distribution of feature measurements in PUT SLAM. The uncertainty model obtained from the reverse SLAM tool contains not only information about the sensor noise but also about the accuracy of image processing and RANSAC-based outliers rejection in the front-end.

Using the reverse SLAM tool, we can analyze the distribution of measurements in the 3-D space. The example distributions of measurements are presented in Fig. 10. Using the visualization tool, we can analyze the distribution of measurements for each feature in the map. We noticed that some measurements are located on the surfaces of the objects (Fig. 10a). We can also compute statistical properties of the distribution. The standard deviation of measurements in z axis (normal to the surface) is 0.005 m which is 62% smaller than in y axis and 130% smaller than in x axis. Another type of distribution is presented in Fig. 10b. In this case, the measured positions of the selected feature are located along the computer cable. The standard deviation of measurements along the y axis is 39% larger than in x axis and 96% larger than in the z axis.

It is also possible to verify distribution visually by projecting measurements on the image plane. In Fig. 11, we present distribution of measurements projected on the first image from the ICL-NUIM *office/kt1* sequence. It is visible that the measurements slide along RGB (photometric) edges forming small clusters. This suggests that despite the spatial uncertainty re-shaping by the RANSAC outlier rejection mechanism we can still find a beneficial anisotropic uncertainty model of the point features.

Table 4 Statistical parameters of feature measurements in the ICL-NUIM *office/kt1* sequence

Config.	Model	σ_x (m)	σ_y (m)	σ_z (m)	d_ε
Matching no noise	C_n	0.0137	0.0125	0.0081	0.38
	C_g	0.0107	0.0106	0.0129	-0.21
Tracking no noise	C_n	0.0097	0.0100	0.0080	0.19
	C_g	0.0085	0.0092	0.0101	-0.14
Matching with noise	C_n	0.0138	0.0134	0.0118	0.13
	C_g	0.0124	0.0121	0.0138	-0.13
Tracking with noise	C_n	0.0112	0.0119	0.0101	0.13
	C_g	0.0111	0.0102	0.0127	-0.19

6 Analysis of quantitative results

6.1 Evaluation of the Uncertainty Models

The reverse SLAM tool that we have developed allows us to compute the distributions of point features depending on the assumed uncertainty model. The reverse SLAM computes a spatial uncertainty ellipsoid from the actual distribution for each feature included in the map. To facilitate the analysis of these ellipsoids, we introduce the d_ε coefficient:

$$d_\varepsilon = \left(1 - \frac{\sigma_z}{(\sigma_x + \sigma_y) \cdot 0.5} \right), \quad (9)$$

where σ_x , σ_y , and σ_z stand for the standard deviations of measurements along the local x , y and z axis of the uncertainty model, respectively. The coefficient defined by (9) gives the relation between the length of the uncertainty ellipsoid major axis and the length of its minor axes. Larger positive values of d_ε indicate that the uncertainty is anisotropic and should be properly captured by the Mahalanobis distance used in (1). However, if d_ε approaches 0 the uncertainty is almost isotropic. Thus, it can be sufficiently modeled as a sphere, which is then represented by an identity matrix in (1). A negative value of d_ε means that the expected major axis of the uncertainty ellipsoid is shorter than its minor axes.

Table 4 gives exemplary numerical results of applying the reverse SLAM tool to the ICL-NUIM *office* environment with the *kt1* trajectory. Uncertainty ellipsoids were computed for the C_n -model and the C_g -model. The ICL-NUIM dataset contains two variants of the depth data: noiseless (ideal) and with simulated Kinect-like noise. The results obtained from our reverse SLAM tool were considerably different for these two variants of the RGB-D data. For the noiseless depth data, the distributions of feature points were best captured by the C_n -model. This is explained by the fact that in the absence of depth errors the spatial uncertainty of point features is caused mainly by errors in the

Table 5 Improvement of SLAM accuracy by using proposed uncertainty models (ICL-NUIM office/kt1 sequence)

Config.	Identity mat.	Model	Uncert. mod.	η (%)
Matching no noise	0.025±0.015	C_n	0.012±0.001	54.5
		C_g	0.019±0.002	26.0
Tracking no noise	0.030±0.014	C_n	0.020±0.003	35.7
		C_g	0.024±0.007	21.6
Matching with noise	0.027±0.003	C_n	0.026±0.002	4.3
		C_g	0.026±0.002	5.2
Tracking with noise	0.057±0.007	C_n	0.058±0.012	−1.4
		C_g	0.046±0.008	19.0

The results obtained for the C_p model are not in the table because the graph optimization does not converge

location of keypoints. In contrast, the C_g -model was better for the ICL-NUIM data with synthetic noise in-depth measurements. Apparently, in this case, the spatial uncertainty of features depends also on the depth noise. The fact that the gradient-based uncertainty model fits best, in this case, is explained by the increased depth errors on edges of objects. In a real sensor based on the PrimeSense technology, this effect is related to the dependency in-between pixels in the depth image [13].

6.2 Impact of the uncertainty models on the SLAM accuracy

The analysis of the spread of point features using the reverse SLAM tool allowed us to gain some intuition about the usefulness of our uncertainty models. However, this intuition had to be verified by investigating how the C_n -model and C_g -model behave in real SLAM. To this end, we tested these two uncertainty models in the regular PUT SLAM software, which computed the ATE and RPE metrics that allowed us to assess the impact of the uncertainty models on SLAM accuracy. Both variants of PUT SLAM were used in these tests—the one with the matching-based VO, and the other one based on tracking for VO implementation.

To facilitate the analysis of the ATE- and RPE-based numerical results, we introduce another coefficient, which indicates the relative trajectory accuracy improvement due to the proposed model:

$$\eta = (E_I - E_C)/E_I \cdot 100\%, \quad (10)$$

where E_I and E_C are the error values (ATE RMSE or RPE RMSE) for the identity matrices and the uncertainty-based matrices, respectively.

Results obtained for the ICL-NUIM sequence are provided in Table 5. For both variants of PUT SLAM (the matching-VO and tracking-VO), the ATE RMSE is decreased

significantly by using the C_n -model on the sequence with noiseless depth data. Moreover, the standard deviation of the resulting ATE metric decreased by applying the uncertainty model. However, when a sequence with simulated depth noise was used, the achieved accuracy improvement was much smaller. Apparently, the C_n -model is inadequate in this case. The application of the C_g -model resulted in an improvement of the ATE metric for the tracking-VO variant, while the accuracy improvement for the matching-VO PUT SLAM was rather insignificant. These results are explainable in light of our observations made using the reverse SLAM tool. When noise is present in the depth data, the dominant source of errors in the location of point features is sliding of these points along edges. The tracking-VO variant uses ORB keypoints that are more prone to dislocation along edges because the detection in ORB is less repeatable than in the SURF keypoints used in the matching-VO variant [38]. Hence, the sliding effect is more profound in the tracking-based version and can be to some extent compensated by a proper model of the anisotropic uncertainty.

7 Application-oriented evaluation

Finally, we have investigated how the proposed approach to uncertainty modeling influences the accuracy of SLAM trajectory estimation using five real RGB-D data sequences from the TUM RGB-D Benchmark, recorded with Kinect or Xtion sensors. As mentioned before, the reverse SLAM tool cannot be used to determine the spatial uncertainty of measurements from such a dataset. Therefore, the C_g -model was applied in these tests, as the previous results strongly suggested that it is the only uncertainty model that provides beneficial improvements in g^2o optimization in the presence of depth measurements noise, which is unavoidable in real Kinect/Xtion data. However, this model should be properly parametrized for the new data type. To this end, we found the relation between the parameters of the uncertainty model and the mean accuracy achieved in a series of experiments with PUT SLAM. We modified the standard deviations along local axes of the C_g uncertainty model and registered errors of the obtained sensor trajectories. For each parameter value, we performed 100 experiments to compute statistics (Fig. 12).

Moreover, we extended the applied uncertainty model considering the fact that the variance of Kinect depth measurements increases with the distance from the camera. Thus, we scale the C_g uncertainty ellipsoid by the factor d_{scale} , which depends on the distance from the camera d (depth measurement) and a scaling parameter S_u :

$$d_{\text{scale}} = (S_u \cdot d)^2 + 1.0. \quad (11)$$

Fig. 12 Relation between SLAM accuracy (RPE RMSE) and parameter of the gradient-based uncertainty model: scale S_z (RGB gradient) (a, b), scale S_y (c, d), and depth-dependent scale factor S_u (e, f) for matching-based (a, c, e) and tracking-based (b, d, f) version of PUT SLAM. The mean value and standard deviation are computed from series of 100 SLAM runs

Finding optimal parameters for such an uncertainty model is extremely time-consuming. Hence, we did not perform an exhaustive search of the uncertainty model parameters. Instead, we applied a simplified, hierarchical procedure. At first, we found the optimal value of the scale along the RGB gradient of the model (S_z) (cf. Fig. 12a and b). Then, we modified the scale along the y axis (S_y) to find the best value of the parameter. The S_x value is set to 1. Finally, the dependence between the S_u scale factor and the SLAM

RPE trajectory accuracy was determined (Fig. 12e and f). We searched for the best parameters of the model using the TUM *fr1_desk* sequence. Then, the parametrized uncertainty model was verified on several other sequences, that we considered as representative for various SLAM tasks and environment types. The parameters of the obtained model are: $S_z = 0.8$, $S_y = 1.25$. The S_u value is set to 5.0 for matching-based and to 1.8 for the tracking-based version of PUT SLAM.

In the experiments on the real RGB-D data, we use the RPE RMSE metric [41]. In our implementation of the feature-based RGB-D SLAM, the position of the camera is connected (through point features) only with a small set of the previous camera poses (keyframes). The probability that distant camera poses is included in the graph used by the back-end decreases with the distance from the current sensor pose. This comes from the fact that our system closes loops only when the sensor drift is small, as the PUT SLAM does not contain appearance-based loop closure implementation. Thus, very long trajectories can drift regardless of the accurate local measurements, and the influence of the feature uncertainty model on the trajectory accuracy is best represented by the RPE RMSE value.

Table 6 summarizes the RPE RMSE metric mean values and standard deviations yielded by the SLAM for five sequences from the TUM dataset. Results obtained using the C_g -model are compared to the accuracy achieved by using the default identity matrices in g^2o . It is worth noting that in the literature often only the best results are provided, without any statistics. However, as the SLAM front-end employs RANSAC, thus being not fully deterministic, we computed the statistics for 100 trials, in order to provide convincing numerical results.

The accuracy improvement due to employing the anisotropic uncertainty model varies from sequence to sequence. It is worth mentioning that the feature position error for single Kinect measurement can be higher than 10 cm

Table 6 RPE RMSE value [mm] improvement by application of uncertainty models in factor graph optimization (TUM RGB-D benchmark)

Sequence	Config.	Identity mat.	C_g -model	η (%)
fr1_xyz	Matching	6.90±0.12	5.85±0.08	15.2
	Tracking	34.97±1.14	28.34±0.72	19.0
fr1_desk	Matching	14.50±0.48	11.92±0.42	17.8
	Tracking	21.08±2.09	17.80±0.71	15.6
fr1_desk2	Matching	14.52±0.46	12.79±0.36	11.9
	Tracking	26.21±3.02	18.67±1.02	28.8
fr2_desk	Matching	17.97±0.54	5.41±0.06	69.9
	Tracking	24.52±2.58	11.36±0.96	53.6
fr3_long_office	Matching	10.69±0.30	7.89±0.13	26.1
	Tracking	14.51±2.46	13.81±2.25	4.83

The optimization results for C_p and C_n models are not included, as g^2o optimization does not converge for these models on real data

Fig. 13 Trajectories obtained for the *fr2_desk* sequence using matching-based (a) and tracking-based (b) version of the SLAM without (identity matrices) and with measurements uncertainty modeling

and our implementation of SLAM reaches few centimeters accuracy on trajectories of the length of about 10 m. Hence, the trajectory error of PUT SLAM is almost ten times smaller than the errors of individual depth measurements. Additional improvement of the accuracy is difficult to obtain. However, by application of the uncertainty model in the back-end optimization the RPE RMSE error can be reduced by more than 50% (*fr2_desk* sequence in Table 6). For other trajectories, the improvement of accuracy varies from 4.83 to 28.8%. The worst improvement is obtained for *fr3_long_office* sequence with the tracking-based version of the SLAM (4.83%). The sparse optical flow tracking with the Kanade–Lucas–Tomasi (KLT) algorithm and ORB point detector is very fast, but sensitive to rapid camera motions due to motion blur in RGB images. When the tracked keypoints drift from their real positions, also the 3-D features in the map accumulate drift. In PUT SLAM, this drift can be compensated only locally, because of the lack of appearance-based loop closures. Such loop closures are implemented in some of the state-of-the-art SLAM systems, such as ORB-SLAM2 [30], and allow these systems to achieve better trajectory esti-

Fig. 14 Example reconstruction of a dense environment map for the *fr3_long_office* sequence using FastFusion algorithm [40]

mation accuracy than the accuracy obtained in PUT SLAM on looped trajectories. However, these problems are more specific to the particular SLAM architecture [2], than to the role of the spatial uncertainty model in the optimization back-end, and are beyond the scope of this paper.

To show the improvement of the trajectory estimation using uncertainty modeling, we enlarge some region of the trajectories obtained on the *fr2_desk* sequence (Fig. 13). Without uncertainty modeling, the noisy measurements of the feature positions propagate to the sensor position. As a result, the camera trajectory is noisy. When the proposed uncertainty model is applied, the optimization-based back-end can utilize measurements with low uncertainties and ignore uncertain measurements. The obtained trajectory is smooth for both, tracking and matching-based PUT SLAM (Fig. 13a and b).

Whereas we focused on the local improvement of the trajectory, PUT SLAM with the proposed uncertainty model C_g provides also globally consistent trajectories, which is proved by small ATE RMSE values obtained using the TUM RGB-D benchmark. We do not attempt to compare PUT SLAM trajectory estimation accuracy against other SLAM approaches, as the ATE values computed for the entire benchmark sequences depend on a number of factors not related to the uncertainty model of the features (or to a lack of such a model). Among these factors, the loop closure mechanism seems to be most important. As we recently have shown elsewhere [32], the loop closure and relocalization function of a SLAM system can compensate the trajectory drift that mounted because of inaccurate matching of the point features. Hence, we only show on few example sequences from the TUM RGB-D Benchmark that PUT SLAM, despite its rather simple architecture, yields globally consistent and accurate trajectories. The best ATE RMSE error for *fr3_long_office* sequence is 0.022, which is better than ORB-SLAM2 [30] and Slam-Dunk-SIFT [12] and

the best ATE RMSE error for `fr1_desk2` sequence is 0.030 (0.031 SubMap BA [41], 0.048 ElasticFusion [45]).

Moreover, in order to demonstrate that with the self-localization accuracy achieved by PUT SLAM using the new uncertainty model accurate reconstruction of a dense environment map is possible, and we produced a surfel-based visualization using the FastFusion algorithm [40] (Fig. 14). The mapping algorithm is integrated with our software and can run in real time along with PUT SLAM, as demonstrated for the `fr3_long_office` sequence in the video material accompanying this paper.

8 Conclusions

This article contributed a novel methodology for the modeling of spatial uncertainty of point features in feature-based RGB-D SLAM. Considering the feature-based approach employing a factor graph as the state of the art in visual and RGB-D SLAM, our work contributed:

- the reverse SLAM tool concept and implementation; this tool is useful to analyze the distribution of feature measurements in both the 3-D and image space and to identify the most suitable uncertainty model.
- new uncertainty models of point features, which incorporate not only the axial and lateral spatial uncertainty caused by the PrimeSense technology RGB-D sensors, but take into account image processing uncertainties in the whole SLAM front-end pipeline;
- an in-depth analysis of the application of feature uncertainty models in the factor graph-based formulation of SLAM, which demonstrates accuracy gains on real RGB-D data sequences due to using more elaborated spatial uncertainty models;

We demonstrated in simulation that the accuracy of feature-based RGB-D SLAM can be improved by accurate identification of anisotropic uncertainty model of measurements. Then, we proposed the application of reverse SLAM tool to analyze the distribution of uncertainty measurements in actual RGB-D SLAM working on synthetic data. The identified uncertainty models are used to compute information matrices of constraints in factor graph optimization of PUT SLAM, which serves as an example implementation for the family of factor graph-based SLAM systems that is targeted in our research. The use of the reverse SLAM revealed that the spatial uncertainty in point features is influenced not only by the sensor properties, but also by the RGB image processing and the iterative procedure for estimation of $SE(3)$ transformation (RANSAC). We consider this an important conclusion from the presented research, as it explains why the uncertainty model based on the sensor noise (C_p -model [36])

works well in simulation, but eventually fails in tests with the actual PUT SLAM. In the simulations, there is no actual image processing involved, and the C_p -model based upon Kinect sensor characteristics improves the results at least one order of magnitude, compared to the same BA-based SLAM algorithm without the uncertainty model. However, this approach does not work with the actual PUT SLAM, that introduces additional uncertainty in the front-end data processing. The model of the uncertainty used in the back-end graph optimization is device specific and specific to the front-end pipeline. However, we introduce the general reverse SLAM tool, which allows us to determine the spread of feature measurements taking into account not only the sensor model, but the whole feature processing pipeline in the SLAM front-end. The proposed procedure can be used for other sensors and front-ends of the SLAM. The visual and numerical analysis of measurements distribution using the reverse SLAM tool allowed us to propose two uncertainty models that catch the noise resulting from the whole data processing in the front-end. The normal-based uncertainty model assumes that the measured positions of features are scattered on the surface of the objects. The gradient-based uncertainty model explains the increased errors along edges.

Finally, we demonstrated experimentally on the TUM RGB-D Benchmark that the proposed gradient-based model is appropriate for real RGB-D data. The accuracy improvement is higher for the tracking-VO variant of the PUT SLAM front-end. The accuracy improvement for this variant of PUT SLAM is important, as the tracking-based variant is much faster (20 to 40 frames per second) than the matching-based variant, but was less accurate in most tests. As the two variants of our front-end differ in the type of point features detector and differ in the details of processing, we were able to demonstrate that the internal architecture of the SLAM front-end indeed influences the uncertainty of features. The open-source PUT SLAM developed in our earlier research served here only as our research vehicle and basis for the reverse SLAM tool, but the developed models and research methodology should be suitable for other factor graph-based RGB-D SLAM systems, as long as they rely on depth data accuracy and use a similar back-end architecture.

As the available RGB-D sensors based on structured light are very similar with respect to performance and uncertainty characteristics [14], the presented uncertainty models should suit all of them: Kinect, Xtion, Carmine, and the recent RealSense. Although some characteristics differ for RGB-D sensors using the time-of-flight principle, which may influence the uncertainty of features (e.g., smaller depth noise along sharp edges [23]), the methodology of developing the uncertainty model with the use of the reverse SLAM tool is still valid for those sensors. Hence, we believe that the simulation/experimental framework based on the reverse SLAM tool will enable us to quickly develop uncertainty

models suitable for the next generation of RGB-D sensors.

Among our further research directions, there is also the work on uncertainty models for other types of constraints, including pose-to-pose and pose-to-plane measurements in factor graph optimization. We are also interested in developing a procedure that determines a suitable uncertainty model of each individual feature according to the local characteristics of the RGB-D data.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Belter, D., Nowicki, M., Skrzypczyński, P.: On the performance of pose-based RGB-D visual navigation systems. In: Cremers, D., Reid, I., Saito, I., Yang, M.-H. (eds.) *Computer Vision—ACCV 2014*, Volume 9004 of the Series Lecture Notes in Computer Science, pp. 407–423. Springer, Cham (2015)
2. Belter, D., Nowicki, M., Skrzypczyński, P.: Accurate map-based RGB-D SLAM for mobile robots. In: Reis, L.P., Moreira, A.P., Lima, P.U., Montano, L., Muñoz-Martinez, V. (eds.) *Robot 2015: Second Iberian Robotics Conference*, Advances in Robotics, vol. 2, pp. 533–545. Springer, Cham (2016)
3. Belter, D., Nowicki, M., Skrzypczyński, P.: Improving accuracy of feature-based RGB-D SLAM by modeling spatial uncertainty of point features. In: *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1279–1284. Stockholm, Sweden (2016)
4. Belter, D., Skrzypczyński, P.: Precise self-localization of a walking robot on rough terrain using parallel tracking and mapping. *Ind. Robot Int. J.* **40**(3), 229–237 (2013)
5. Belter, D., Skrzypczyński, P.: The importance of measurement uncertainty modeling in the feature-based RGB-D SLAM. In: *Proceedings of the 10th International Workshop on Robot Motion and Control*, pp. 308–313. Poznań, Poland (2015)
6. Davison, A.J., Reid, I., Molton, N., Stasse, O.: MonoSLAM: real-time single camera SLAM. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(6), 1052–1067 (2007)
7. Di Leo, G., Liguori, C., Paolillo, A.: Covariance propagation for the uncertainty estimation in stereo vision. *IEEE Trans. Instrum. Meas.* **60**(5), 1664–1673 (2011)
8. Dryanovski, I., Valenti, R., Xiao, J.: Fast visual odometry and mapping from RGB-D data. In: *Proceedings of IEEE International Conference on Robotics & Automation*, pp. 2305–2310. Karlsruhe, Germany (2013)
9. Endres, F., Hess, J., Sturm, J., Cremers, D., Burgard, W.: 3-D mapping with an RGB-D camera. *IEEE Trans. Robot.* **30**(1), 177–187 (2014)
10. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(3), 611–625 (2018)
11. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: large-scale direct monocular SLAM. In: *Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision—ECCV 2014*, pp. 834–849. Springer, Cham (2014)
12. Fioraio, N., Di Stefano, L.: SlamDunk: affordable real-time RGB-D SLAM. In: *Agapito, L., Bronstein, M.M., Rother, C. (eds.) Computer Vision—ECCV 2014 Workshops*, pp. 401–414. Springer, Cham (2015)
13. Gesto Diaz, M., Tombari, F., Rodriguez-Gonzalez, P., Gonzalez-Aguilera, D.: Analysis and evaluation between the first and the second generation of RGB-D sensors. *IEEE Sens. J.* **15**(11), 6507–6516 (2015)
14. Gonzalez-Jorge, H., Riveiro, B., Vazquez-Fernandez, E., Martínez-Sánchez, J., Arias, P.: Metrological evaluation of Microsoft Kinect and Asus Xtion sensors. *Measurement* **46**(6), 1800–1806 (2013)
15. Handa, A., Whelan, T., McDonald, J.B., Davison, A.J.: A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In: *Proceedings of the IEEE International Conference on Robotics & Automation*, pp. 1524–1531. Hong Kong (2014)
16. Haralick, R.: Propagating covariance in computer vision. In: *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, pp. 493–498. Jerusalem, Israel (1994)
17. Hartley, R.I., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge (2004)
18. Henry, P., Krainin, M., Herbst, E., Ren, X., Fox, D.: RGB-D mapping: using kinect-style depth cameras for dense 3D modeling of indoor environments. *Int. J. Robot. Res.* **31**(5), 647–663 (2012)
19. Kerl, C., Sturm, J., Cremers, D.: Robust odometry estimation for RGB-D cameras. In: *Proceedings of the IEEE International Conference on Robotics & Automation*, pp. 3748–3754. Karlsruhe, Germany (2013)
20. Khoshelham, K., Elberink, S.: Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors* **12**(2), 1437–1454 (2012)
21. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: *Proceedings of the International Symposium on Mixed and Augmented Reality*, pp. 225–234. Nara, Japan (2007)
22. Konolige, K., Agrawal, M.: FrameSLAM: from bundle adjustment to realtime visual mapping. *IEEE Trans. Robot.* **24**(5), 1066–1077 (2008)
23. Kraft, M., Nowicki, M., Schmidt, A., Fularz, M., Skrzypczyński, P.: Toward evaluation of visual navigation algorithms on rgb-d data from the first- and second-generation kinect. *Mach. Vis. Appl.* **28**(1), 61–74 (2017)
24. Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., Burgard, W.: g²o: a general framework for graph optimization. In: *Proceedings of the IEEE International Conference on Robotics & Automation*, pp. 3607–3613. Shanghai, China (2011)
25. Mallick, T., Das, P., Majumdar, A.: Characterizations of noise in kinect depth images: a review. *IEEE Sens. J.* **14**(6), 1731–1740 (2014)
26. Matthies, L., Shafer, S.A.: Error modeling in stereo navigation. *IEEE J. Robot. Autom.* **3**(3), 1255–1262 (1987)
27. Miura, J., Shirai, Y.: An uncertainty model of stereo vision and its applications to vision-motion planning of robot. In: *13th International Joint Conference on Artificial Intelligence*, pp. 1618–1623. Chambery (1993)
28. Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F., Sayd, P.: Generic and real-time structure from motion using local bundle adjustment. *Image Vis. Comput.* **27**(8), 1178–1193 (2014)
29. Mur-Artal, R., Montiel, J.M.M., Tardós, J.D.: ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Trans. Robot.* **31**(5), 1147–1163 (2015)
30. Mur-Artal, R., Tardós, J.D.: ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *IEEE Trans. Robot.* **33**(5), 1255–1262 (2017)
31. Nguyen, C., Izadi, S., Lovell, D.: Modeling kinect sensor noise for improved 3D reconstruction and tracking. In: *Proceedings of the Second International Conference on 3D Imaging, Modeling,*

- Processing, Visualization & Transmission, pp. 524–530. Zürich, Switzerland (2012)
32. Nowicki, M., Belter, D., Kostusiak, A., Čížek, P., Faigl, J., Skrzypczyński, P.: An experimental study on feature-based SLAM for multi-legged robots with RGB-D sensors. *Ind. Robot Int. J.* **44**(4), 428–441 (2017)
 33. Nowicki, M., Skrzypczyński, P.: Combining photometric and depth data for lightweight and robust visual odometry. In: *Proceedings of the European Conference on Mobile Robots*, pp. 125–130. Barcelona, Spain (2013)
 34. Oskiper, T., Chiu, H.P., Zhu, Z., Samaresekera, S., Kumar, R.: Stable vision-aided navigation for large-area augmented reality. In: *Proceedings of the IEEE Virtual Reality Conference*, pp. 63–70. Singapore (2011)
 35. Ozog, P., Eustice, R.: On the importance of modeling camera calibration uncertainty in visual SLAM. In: *Proceedings of the IEEE International Conference on Robotics & Automation*, pp. 3777–3784. Karlsruhe, Germany (2013)
 36. Park, J.H., Shin, Y.D., Bae, J.H., Baeg, M.H.: Spatial uncertainty model for visual features using a kinect sensor. *Sensors* **12**(7), 8640–8662 (2012)
 37. Scaramuzza, D., Fraundorfer, F.: Visual odometry: Part I the first 30 years and fundamentals. *IEEE Robot. Autom. Mag.* **18**(4), 80–92 (2011)
 38. Schmidt, A., Kraft, M., Fularz, M., Domagala, Z.: Comparative assessment of point feature detectors and descriptors in the context of robot navigation. *J. Autom. Mobile Robot. Intell. Syst.* **7**(1), 11–20 (2013)
 39. Skrzypczyński, P.: Spatial uncertainty management for simultaneous localization and mapping. In: *Proceedings of the IEEE International Conference on Robotics & Automation*, pp. 4050–4055. Rome, Italy (2007)
 40. Steinbrücker, F., Sturm, J., Cremers, D.: Volumetric 3D mapping in real-time on a CPU. In: *Proceedings of the IEEE International Conference on Robotics & Automation*, pp. 2021–2028. Hong Kong (2014)
 41. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of RGB-D SLAM systems. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems*, pp. 573–580. Vilamoura, Portugal (2012)
 42. Triggs, B., McLauchlan, P., Hartley, R., Fitzgibbon, A.: Bundle adjustment—a modern synthesis. In: Triggs, B., Zisserman, S., Szeliski, R. (eds.) *Vision Algorithms: Theory and Practice. Lecture Notes in Computer Science*, vol. 1883, pp. 298–372. Springer, Cham (2000)
 43. Umeyama, S.: Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **13**(4), 80–92 (1991)
 44. Whelan, T., Johannsson, H., Kaess, M., Leonard, J., McDonald, J.: Robust real-time visual odometry for dense RGB-D mapping. In: *Proceedings of the IEEE International Conference on Robotics & Automation*, pp. 5704–5711. Karlsruhe, Germany (2013)
 45. Whelan, T., Salas-Moreno, R.F., Glocker, B., Davison, A.J., Leutenegger, S.: ElasticFusion: real-time dense slam and light source estimation. *Int. J. Robot. Res.* **35**(14), 1697–1716 (2016)

Dominik Belter graduated from the Poznan University of Technology (2007). He received Ph.D. degree in robotics from the same University in 2012. Since 2012, he has been an Assistant Professor at the Institute of Control, Robotics and Information Engineering of the Poznan University of Technology. Dominik Belter is the author or co-author of over 60 technical papers in the fields of robotics and computer science. His research interests include walking robots, machine learning, and computer vision.

Michał Nowicki graduated from Poznan University of Technology in 2014 receiving B.Sc. and M.Sc. in automatic control and robotics from the Poznan University of Technology (PUT) in 2013 and 2014, respectively, and B.Sc. in computer science from the same university in 2014. Currently, he is a Research Assistant and Ph.D. student at the Institute of Control, Robotics and Information Engineering. His research interests include computer vision, simultaneous localization and mapping, and Android-based navigation.

Piotr Skrzypczyński received the Ph.D. and D.Sc. degrees in robotics from Poznań University of Technology (PUT) in 1997 and 2007, respectively. Since 2010, he is an Associate Professor at the Institute of Control, Robotics and Information Engineering (ICRIE) of PUT and Head of the Mobile Robotics Laboratory at ICRIE. He is author or co-author of over 160 technical papers in robotics and computer science. His current research interests include: autonomous mobile robots, simultaneous localization and mapping, multisensor fusion, machine learning, and computational intelligence in robotics.