2016 IEEE International Conference on Robotics and Automation (ICRA)
Stockholm, Sweden, May 16-21, 2016

Improving Accuracy of Feature-based RGB-D SLAM by Modeling Spatial Uncertainty of Point Features

Dominik Belter, Michał Nowicki, Piotr Skrzypczyński

Abstract— **Many recent solutions to the RGB-D SLAM problem use the pose-graph optimization approach, which marginalizes out the actual depth measurements. In this paper we employ the same type of factor graph optimization, but we investigate the gains coming from maintaining a map of RGB-D point features and modeling the spatial uncertainty of these features. We demonstrate that RGB-D SLAM accuracy can be increased by employing uncertainty models reflecting the actual errors introduced by measurements and image processing. The new approach is validated in simulations and in experiments involving publicly available data sets to ensure that our results are verifiable.**

I. Introduction

In the recent paper [1] we have presented a performance analysis of several variants of pose-based RGB-D SLAM (Simultaneous Localization and Mapping) and VO (Visual Odometry) systems employing point features for sensor motion estimation. However, the approach based on pose-graph optimization marginalizes the RGB-D point features in the front-end and doesn't include them in optimization. Thus, the pose-based approach cannot take advantage from the large number of pose-to-feature correspondences. These correspondences are essential in the Bundle Adjustment (BA) approach [20], widely employed by very accurate visual SLAM systems, such as PTAM [10] and ORB-SLAM [13]. While some recent works on factor graph-based RGB-D SLAM incorporate point features in the optimization framework [5], [12], [17], these systems do not attempt to model the spatial uncertainty of features or use a simplified approach to uncertainty modeling.

Therefore, we investigate how to efficiently incorporate the RGB-D point features in the factor graph-based SLAM framework, and how to model the spatial uncertainty of the features in order to improve the accuracy of trajectory estimation. The PUT SLAM system presented in this paper re-uses some techniques from our previous implementations [1], [15], but has a fundamentally different structure, building a persistent map of 3-D point features (Fig. 1). The main contribution of this work is a novel approach for modeling the spatial uncertainty of features. New spatial uncertainty models are developed on the basis of an experimental investigation of the location and spread of point features in RGB-D SLAM.

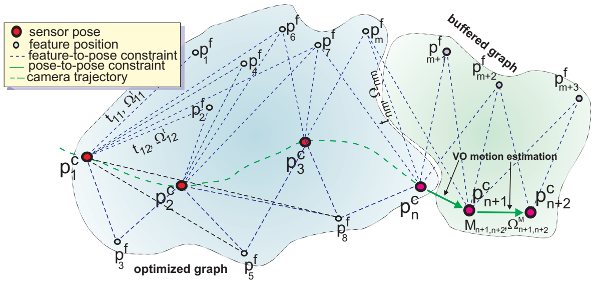

Fig. 1. Factor graph structure of the feature-based PUT SLAM

II. Related Work

Impressive accuracy in RGB-D SLAM was demonstrated by employing dense depth data, but this approach requires high-end GPGPU [22]. Thus, to run with no hardware acceleration many RGB-D SLAM systems follow the pose-graph approach discarding sensor measurements. They estimate the pose-to-pose (motion) constraints by using dense optical flow [8], sparse optical flow on keypoints [15], variants of the Iterative Closest Points (ICP) algorithm [7], or by matching directly sparse RGB keypoints [4].

In the BA approach to SLAM the sensor poses and feature locations are jointly optimized over sensor measurements, which results in the accuracy and robustness better than in the pose-based or dense/direct methods in visual SLAM [13]. Local BA was used to improve the pose-to-pose transformation estimation in RGB-D SLAM [7]. Also the global sparse BA can be augmented by depth measurements [17]. Recently, Maier *et al.* [12] defined local BA on submaps for real-time object reconstruction from RGB-D data, whereas the SlamDunk system [5] uses a global pool of 3-D point features and factor graph optimization. Although this system is similar to our approach, it directly matches the current perception to a large map of features, whereas PUT SLAM uses a fast VO algorithm to obtain a first guess of the camera pose, which improves matching robustness. Moreover, in [5] the uncertainty of point features is neglected. PUT SLAM shares also some concepts of the factor graph structure with [18]. However, [18] minimizes the re-projection error, whereas in the feature-based PUT SLAM the error is computed in the Euclidean space and the anisotropic spatial uncertainty of features is taken into account in optimization.

Khoshelham and Elberink [9] provided an analysis of the accuracy and resolution of depth data from Kinect sensor, while Park *et al.* [16] proposed a mathematical uncertainty model for the RGB-D features measured using Kinect. In spite of the availability of such physically-grounded sensor models, few papers tackle the issues of uncertainty in RGB-

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

D. Belter, M. Nowicki and P. Skrzypczyński are with the Institute of Control and Information Engineering, Poznan University of Technology (PUT), ul. Piotrowo 3A 60-965 Poznań, Poland, {name.surname}@put.poznan.pl

978-1-4673-8026-3/16/$31.00 ©2016 IEEE

D SLAM. Nguyen *et al.* [14] described a depth uncertainty model of Kinect, and applied it to a voxel-based RGB-D SLAM demonstrating improved accuracy of the recovered trajectories. Dryanovski *et al.* [3] formulated an uncertainty model for RGB-D features based on Gaussian mixture, and applied it to register a RGB-D camera pose against a map of sparse features updated by means of Kalman filtering. Endres *et al.* [4] applied the depth measurement model from [9] in their motion estimates verification procedure for RGB-D SLAM. Recently, using the model from [16], we have demonstrated in simulations that considering the anisotropic uncertainty in factor graph optimization significantly improves the accuracy in feature-based RGB-D SLAM [2].

III. RGB-D SLAM WITH A MAP OF FEATURES

The map contains poses of the camera and a set of visually salient features augmented by their 3-D positions computed from depth data. The factor graph for optimization in the back-end is constructed from the map data. Processing of the RGB-D frames (from Kinect or a similar sensor) in the front-end starts with a camera displacement guess obtained from the fast VO algorithm [1], which provides robust frame-to-frame tracking of the sensor pose, and is independent from the map. Two configurations of the VO pipeline in the front-end are considered. In the first one the associations between features in two frames are computed by detecting SURF keypoints in RGB images, describing them with local descriptors, and performing descriptor-based matching. Because the SURF descriptors are slow to compute and match the alternative VO pipeline has been implemented. It associates features employing fast sparse optical flow tracking [15]. In this variant ORB features are used instead of SURF, as the tracker requires corner-like keypoints. For the sake of computation efficiency the descriptors of features extracted in the VO are then re-used for matching between the new RGB-D frame and the map of features. From the set of associated features, the correct matches (inliers) are estimated using the preemptive RANSAC framework. The Umeyama algorithm [21] estimates the **SE**(3) transformation between frames, which is then verified using the remaining pairs of features. The next step involves matching of the features detected in the current frame to the predicted features from the map. PUT SLAM handles local, metric loop closures implicitly, by establishing pose-to-feature constraints to the map. To robustly determine feature-to-map correspondences the guided matching approach is applied – the nearest neighbor feature matches in descriptor space are accepted only in a small neighborhood of the predicted positions of the map features in the current image frame. With a set of candidate matches, the inliers are determined again by RANSAC.

The factor graph representation has two types of vertices: $\mathbf{p}^f$ representing the point features, and $\mathbf{p}^c$ representing the sensor poses. The edge $\mathbf{t}_{ij} \in \mathbb{R}^3$ represents a measurement constraint between the i-th pose and the j-th feature, whereas the edge $\mathbf{M}_{ik} \in \mathbf{SE}(3)$ is the rigid transformation constraint imposed by the estimated motion between poses i and k (cf. Fig. 1). Sensor poses are expressed in the global frame of the map, while the features are anchored in the poses from which they have been detected. To find the most plausible sequence of the camera poses (positions and orientations) $\mathbf{p}_1^c, ..., \mathbf{p}_n^c$ and feature positions $\mathbf{p}_1^f, ..., \mathbf{p}_m^f$ the following function has to be minimized:

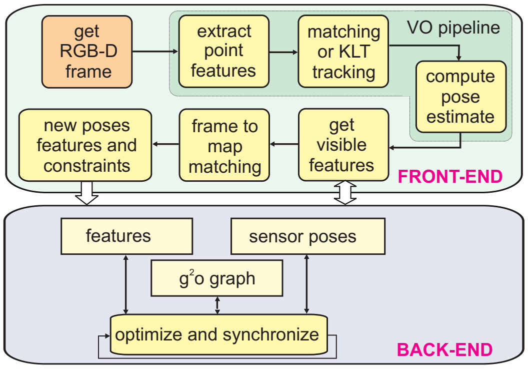

Fig. 2. Software architecture of the PUT SLAM system

$$\underset{\mathbf{p}}{\operatorname{argmin}}\; F = \sum_{i=1}^{n} \sum_{j=1}^{m} e(\mathbf{p}_i^c, \mathbf{p}_j^f, \mathbf{m}_{ij})^T \boldsymbol{\Omega}_{ij} e(\mathbf{p}_i^c, \mathbf{p}_j^f, \mathbf{m}_{ij}), \tag{1}$$

where $e(\mathbf{p}_i^c, \mathbf{p}_j^f, \mathbf{m}_{ij})$ is an error function computed for the estimated pose of the vertex and measured pose of the vertex which comes out from the measurement $\mathbf{m}_{ij}$, where $\mathbf{m}_{ij}$ is $\mathbf{t}_{ij} \in \mathbb{R}^3$ for pose-to-feature or $\mathbf{M}_{ij} \in \mathbf{SE}(3)$ for pose-to-pose constraints. The g^{2}o general graph optimization library [11] is used to solve (1).

The accuracy of each pose-to-feature measurement is represented by an information matrix $\boldsymbol{\Omega}^{\mathbf{t}}$ obtained by inverting the covariance matrix of this measurement. Thus, the information matrix is directly related to the spatial uncertainty of the feature. The pose-to-pose constraints obtained from the VO are added to the factor graph only if the number of map features matching the current perception is below a given threshold. These constraints stabilize the optimization in g^{2}o until a satisfactory overlapping with the existing map is achieved. For the pose-to-pose constraints the information matrix $\boldsymbol{\Omega}^{\mathbf{M}}$ is set to an identity matrix.

The front-end and the back-end work asynchronously, and get synchronized only on specific events (Fig. 2). Owing to this software architecture[1] the system efficiently uses a multi-core CPU with no GPGPU acceleration required. The map is synchronized with the optimized factor graph only when the back-end finishes the on-going optimization session. Therefore, new poses, features, and measurements are kept in a temporary graph (called “buffered”) during optimization of the main graph (cf. Fig. 1).

IV. SPATIAL UNCERTAINTY MODELING

A. Experiments in Simulation

The uncertainty of the Kinect sensor’s depth measurement increases with increasing distance to the observed surface (axial noise), and is influenced by the low resolution of

[1] Source code is at https://github.com/LRMPUT/PUTSLAM/tree/release

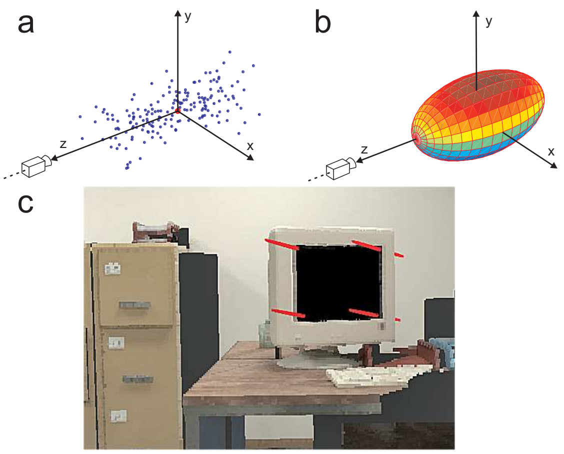

Fig. 3. View-dependent feature uncertainty: assumed distribution of measurements (a), visualization of the spatial uncertainty model (b), and visualization of selected features with their uncertainty ellipsoids

the measurements (lateral noise) [9], [14]. Moreover, the spatial uncertainty in the RGB-D point features depends on the noise introduced by the keypoint detector in the RGB images [16]. These factors result in the spatial uncertainty of features, which can be considerably anisotropic, and cannot be captured by the identity covariance matrices commonly applied for factor graph optimization in the back-ends of SLAM systems.

In order to demonstrate the influence of a more elaborated uncertainty model on the results of factor graph optimization we have developed a SLAM simulator that isolates the factors we wanted to investigate (the number, distribution and uncertainty model of features) from the data association errors usually introduced by a real front-end. The simulator describes the environment with a set of known 3-D points that correspond to visual features. Each point feature has an unique ID, so the simulated front-end can perfectly match features visible from various viewpoints. Positions of the points observed by the simulated RGB-D sensor are noisy – they are randomly drawn from the Gaussian pdfs of these features defined according to the assumed uncertainty model.

For the initial experiments we selected a model based on the approach proposed by Park *et al.* [16], with the depth uncertainty modeled according to the relation found in [9]. This model assumes that with 99.7% probability the feature measurements are located within an ellipsoid, accounting for both the axial and the lateral noise (Fig. 3a). The model is camera-view dependent. The major axis of the ellipsoid (in this case the z axis of the local coordinate system) is located on the line between the feature and the focal point of the camera (Fig. 3b). The covariance matrix $\boldsymbol{C}_p$ of a measured feature is computed by propagating the uncertainty of the pixel location in the (u, v) coordinates of the RGB image and the uncertainty of the corresponding depth measurement d:

$$\boldsymbol{C}_p = \mathbf{J}_f \cdot \boldsymbol{C}_f \cdot \mathbf{J}_f^T, \tag{2}$$

where $\mathbf{J}_f$ is the Jacobian of $f(u, v, d)$ – a function used to compute the Cartesian coordinates of the feature from the image coordinates (u,v) and depth d, with respect to these variables, whereas $\boldsymbol{C}_f$ is a 3×3 diagonal covariance matrix with σ_u^2, σ_v^2 and σ_d^2 on the diagonal. The uncertainty in keypoint location σ_u^2 and σ_v^2 is approximated by constant values determined experimentally. As shown by Park *et al.* [16] these values capture the worst-case uncertainty of a keypoint location for the SURF detector. The depth variance is computed as in [9].

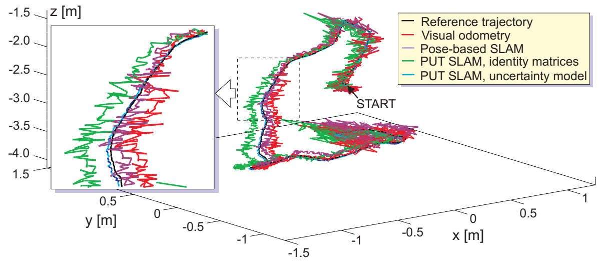

Fig. 4. Perfect front-end simulation results (ICL-NUIM `office/kt1`). The inset image shows an enlarged fragment of the trajectories

TABLE I

ATE AND POSITIONAL RPE FOR THE SIMULATED RGB-D SENSOR MOVING IN THE ICL-NUIM `office/kt1` ENVIRONMENT

method	ATE RMSE [m]		RPE RMSE [m]	
	mean	std.dev.	mean	std.dev.
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 mat.	0.0708	0.0049	0.0916	0.0040
PUT SLAM, $\mathbf{C}_p$ uncert.	0.0142	0.0011	0.0187	0.0016

In the simulation (Fig. 4) the `kt1` camera trajectory from the ICL-NUIM data set [6] was used. The point features observed by the simulated RGB-D sensor are obtained off-line from the PUT SLAM front-end processing the ICL-NUIM `office` sequence of RGB-D frames. The extracted features are then fixed in the simulator and augmented by unique IDs. In this experiment 86 features were used: min. distance between two features was 0.35 m, and 4 to 28 features were co-observed between two consecutive frames. The quantitative results were computed according to the well-established methodology introduced in [19], using the Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) metrics. For each investigated configuration we performed a series of 100 simulations to obtain statistics. From the results summarized in Tab. I it is evident that the feature-based PUT SLAM yields more accurate trajectory estimate than the pose-based SLAM implemented as in [1]. Moreover, taking into account the anisotropic feature uncertainty model allows PUT SLAM to achieve much smaller ATE than it was possible using identity matrices.

B. Experiments with the Reverse SLAM

Despite of the fact that the anisotropic uncertainty model brings a significant improvement in the simulated environment we found difficulties to use the approach proposed by Park *et al.* [16] in PUT SLAM running with the real front-end. If the $\mathbf{C}_p^{-1}$ information matrices are used, the optimization in g^{2}o often does not reach the convergence. This suggests that the $\mathbf{C}_p$ covariance matrices do not represent the actual uncertainty of the feature measurement errors.

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 SLAM the map and the trajectory of the robot are estimated taking into account noisy measurements. However, we are looking for the distribution of feature measurements that is obtained in the front-end when SLAM returns a perfect trajectory, i.e. the sensor noise and image processing errors accumulate in the representation of features, not in the trajectory error. In practice we run PUT SLAM and at each step we replace the estimated motion by ground truth motion. Providing that a precise ground truth trajectory is available a distribution of measurements (3-D positions of features) for each feature in the map can be computed. Unfortunately, data sets obtained using real sensors (e.g. [19]) introduce significant errors in the computed feature distributions due to the inaccuracy of the motion capture data and synchronization issues. Thus, the synthetic ICL-NUIM data set [6] is used in experiments with reverse SLAM. It provides perfect ground truth trajectories of the sensor and correspondences between the RGB-D frames and trajectory points.

Analysis of the measurements using the reverse SLAM tool allows us to create an uncertainty model of point features which captures the actual distribution of feature measurements in PUT SLAM. It is important that such a model captures not only the Kinect-type sensor noise, but also the uncertainty due to the limited precision of image processing and RANSAC-based estimation in the front-end.

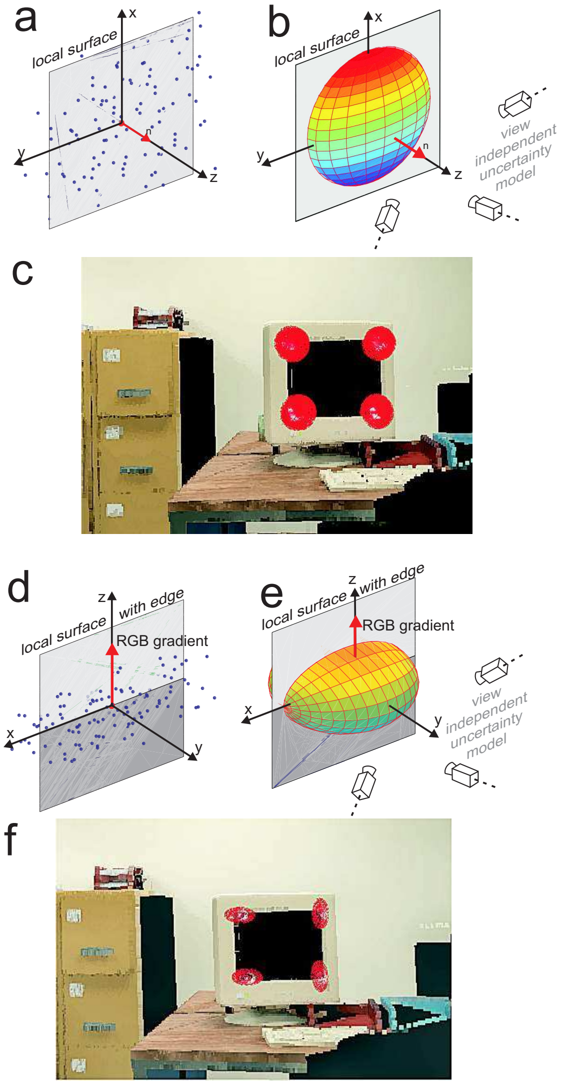

Fig. 5. View-independent feature uncertainty models – normal-based (a,b,c), and gradient-based (d,e,f): typical distribution of measurements (a,d), visualization of the spatial uncertainty model (b,e), and visualization of selected features with their uncertainty ellipsoids (c,f)

C. Normal-based Uncertainty Model

Initially, we analyzed the distributions of features for the ICL-NUIM sequences without depth noise, in order to determine the influence of the RGB image processing errors on the uncertainty of features. We have observed that for most features the measurements are spread on the object surfaces, forming rather flat "discs" (Fig. 5a). To capture this type of distribution the normal-based model ($\mathbf{C}_n$-model) is introduced, in which the minor axis of the ellipsoid is normal to the observed surface (Fig. 5b). This uncertainty model is view independent, defined in the local coordinates of the feature. To determine the normal vector (minor axis of the ellipsoid) we use the depth image only. The feature on the image coordinates $f_{u,v}$ and neighboring pixels (8-neighborhood) $f_{u+i,v+j}, i,j \in \{-1,0,1\}$ are transformed to the Cartesian coordinate system related to the camera. Then, we compute the set of vectors between the 3-D position of the feature $\mathbf{p}^c$ and the 3-D positions of the neighboring pixels on the image $\mathbf{p}_i$. To find the normal we compute cross products for pairs of neighboring vectors and compute the average:

$$n = \frac{1}{8}\sum_{i=1}^{8}(\mathbf{p}_i - \mathbf{p}^c) \times (\mathbf{p}_{(i+1)} - \mathbf{p}^c), \tag{3}$$

where i iterates clockwise on neighboring points and index $i+1$ is set to 0 if i =8. Finally, we create a local coordinate system represented by the rotation matrix $\mathbf{R}$. The z axis of the system $\mathbf{R}$ coincides with the surface normal n. The covariance matrix $\mathbf{C}_n$ is computed as:

$$\mathbf{C}_n = \mathbf{R} \cdot \mathbf{S} \cdot \mathbf{R}^{-1}, \tag{4}$$

where $\mathbf{S}$ is a diagonal scaling matrix. We scale the z axis which is related to the surface normal n. The scaling coefficient S_{33} is in the range (0,1). The remaining diagonal elements of the matrix S_{11} and S_{22} are set to 1. Although some keypoints located on geometric (3-D) corners do not have well defined normals, the graph optimization can combine all approximate measurement uncertainties provided for such a corner keypoint to better estimate the feature position. Uncertainty ellipsoids computed from $\mathbf{C}_n$ for selected features in the ICL-NUIM `office` sequence are shown in Fig. 5c.

[2] Hi-res video is available on http://lrm.cie.put.poznan.pl/icra2016.mp4

TABLE II

DISTRIBUTION OF MEASUREMENTS FOR FEATURES IN THE ICL-NUIM `office/kt1` SEQUENCE

config.	model	σ_x [m]	σ_y [m]	σ_z [m]	d_ε [%]
Matching no noise	$\mathbf{C}_n$	0.0137	0.0125	0.0081	38.09
	$\mathbf{C}_g$	0.0107	0.0106	0.0129	-21.26
Tracking no noise	$\mathbf{C}_n$	0.0097	0.0100	0.0080	19.12
	$\mathbf{C}_g$	0.0085	0.0092	0.0101	-14.44
Matching with noise	$\mathbf{C}_n$	0.0138	0.0134	0.0118	12.93
	$\mathbf{C}_g$	0.0124	0.0121	0.0138	-12.81
Tracking with noise	$\mathbf{C}_n$	0.0112	0.0119	0.0101	12.55
	$\mathbf{C}_g$	0.0111	0.0102	0.0127	-18.78

TABLE III

SLAM ACCURACY IMPROVEMENT BY USING THE NEW UNCERTAINTY MODELS (ICL-NUIM `office/kt1` SEQUENCE)

config.	model	identity mat.	uncert. mod.	η [%]
Matching no noise	$\mathbf{C}_n$	0.025±0.015	0.012±0.001	54.5
	$\mathbf{C}_g$		0.019±0.002	26.0
Tracking no noise	$\mathbf{C}_n$	0.030±0.014	0.020±0.003	35.7
	$\mathbf{C}_g$		0.024±0.007	21.6
Matching with noise	$\mathbf{C}_n$	0.027±0.003	0.026±0.002	4.3
	$\mathbf{C}_g$		0.026±0.002	5.2
Tracking with noise	$\mathbf{C}_n$	0.057±0.007	0.058±0.012	-1.4
	$\mathbf{C}_g$		0.046±0.008	19.0

D. Gradient-based Uncertainty Model

During the analysis of the reverse SLAM results we noticed that features located on edges present in the RGB images slide along these edges (Fig. 5d). Thus, we propose another spatial uncertainty model, in which the major axis of the ellipsoid is located on the detected edge (Fig. 5e). The procedure which determines the uncertainty matrix requires the rotation matrix $\mathbf{R}$ and scaling matrix $\mathbf{S}$. First, we detect the edge in RGB image on a 3×3 patch around the point feature using the Scharr kernel. We compute the direction of this edge in 3-D space using the depth data. Then, we construct the rotation matrix $\mathbf{R}$ representing the local coordinate system, which z axis coincides with the RGB gradient vector. The x axis of the coordinate system is located on the edge (note that it might be an edge of an object or a photometric edge on a flat surface). The covariance matrix $\mathbf{C}_g$ is computed according to (4). We scale the z axis of the ellipsoid. The value of S_{33} is in range (0,1), the elements S_{11} and S_{22} related to the x and y axes are set to 1. Uncertainty ellipsoids for selected features computed according to this $\mathbf{C}_g$-model are shown in Fig. 5f.

V. EVALUATION RESULTS

A. Tests with the Reverse SLAM Tool

We started to evaluate the influence of our new feature uncertainty models on the RGB-D SLAM performance using the reverse SLAM tool. Uncertainty ellipsoids from the actual distributions according to the $\mathbf{C}_n$-model and $\mathbf{C}_g$-models were computed for each feature in the map. To simplify the analysis we define the d_ε coefficient:

$$d_\varepsilon = \left(1 - \frac{\sigma_z}{(\sigma_x + \sigma_y) \cdot 0.5}\right) \cdot 100\%, \qquad (5)$$

where σ_x,σ_y, and σ_z are standard deviations of measurements along each axis. This coefficient describes the length of the major axis in relation to length of minor axes. If d_ε is close to 0% the uncertainty can be modeled by a sphere, and represented by an identity matrix in g^{2}o – it is not possible to take advantage from anisotropic uncertainty modeling. If d_ε is negative the expected major axis of the ellipsoid is shorter than the minor axes. Results for the ICL-NUIM data set are presented in Tab. II.

For the depth data without noise the obtained distribution is best captured by the $\mathbf{C}_n$-model. In this case the spatial uncertainty in features is introduced mainly by errors in keypoint detection. In contrast, for the ICL-NUIM sequence with noise the $\mathbf{C}_g$-model fits better. In this case the uncertainty is caused also by the depth noise that is particularly large on edges of objects.

B. Influence of the uncertainty models on SLAM accuracy

The reverse SLAM tool gave us some intuition how the proposed uncertainty models may behave in real SLAM. Then, we tested these two models in PUT SLAM on two publicly available data sets: the synthetic ICL-NUIM and the TUM RGB-D benchmark [19] obtained with real Kinect and Xtion sensors.

Results on the ICL-NUIM sequence (Tab. III) suggest that we properly identified the uncertainty models. The coefficient $\eta = (\text{ATE}_I - \text{ATE}_C)/\text{ATE}_I \cdot 100\%$, where ATE_I and ATE_C are the ATE RMSE errors for identity and covariance-based uncertainty, respectively, indicates the improvement due to the proposed model. For the data without depth noise the ATE RMSE is decreased significantly using the $\mathbf{C}_n$-model for both matching-VO and tracking-VO variants. By applying the uncertainty model the standard deviation of the resulting ATE metric decreases. For the data with depth noise the $\mathbf{C}_n$-model is inadequate, and the achieved accuracy improvement is much smaller.

For the matching-VO front-end the improvement is insignificant, but the tracking-based version benefits from the $\mathbf{C}_g$-model. This result shows that the sliding of features along edges is the dominant source of errors. The tracking-based version uses ORB keypoints, which are less repeatable than the SURF keypoints used in the matching-based variant, and are more prone to dislocation along edges. The depth noise amplifies these effects, as the depth measurements errors are bigger along the edges of objects. Feature measurements that were displaced on the RGB image could have much different depth values assigned, and thus have large spread.

On the basis of the previous results we selected the $\mathbf{C}_g$-model for application to the TUM RGB-D benchmark sequences. Because PUT SLAM front-end uses RANSAC and is not fully deterministic Tab. IV provides the ATE RMSE metric mean values and standard deviations obtained in 10 trials for five sequences. It is worth to note that in the literature usually only the best results are given. Results of using the $\mathbf{C}_g$-model are compared to the accuracy achieved by using identity matrices. The accuracy improvement due to the anisotropic uncertainty model varies from sequence to sequence, but is bigger for the tracking-VO variant. This is

TABLE IV

ATE RMSE VALUE IMPROVEMENT BY APPLICATION OF UNCERTAINTY MODELS IN FACTOR GRAPH OPTIMIZATION (TUM RGB-D BENCHMARK)

sequence	config.	identity mat.	$\mathbf{C}_g$-model	η [%]
fr1_desk	Matching	0.031±0.003	0.029±0.002	4.6
	Tracking	0.068±0.042	0.057±0.013	16.33
fr1_desk2	Matching	0.054±0.018	0.049±0.010	9.1
	Tracking	0.187±0.110	0.109±0.009	41.79
fr1_room	Matching	0.212±0.032	0.175±0.046	16.75
	Tracking	0.248±0.045	0.232±0.032	6.45
fr2_desk	Matching	0.067±0.005	0.066±0.003	1.87
	Tracking	0.120±0.033	0.111±0.010	7.02
fr3_long_office	Matching	0.030±0.008	0.025±0.017	16.22
	Tracking	0.230±0.094	0.203±0.049	11.74

TABLE V

COMPARISON OF SLAM ACCURACY (ATE RMSE)

sequence	PUT SLAM	best known result
fr1_desk	0.026	0.022 SubMap BA [12]
fr1_desk2	0.034	0.031 SubMap BA [12]
fr1_room	0.121	0.085 SubMap BA [12]
fr2_desk	0.060	0.076 SubMap BA [12]
fr3_long	0.021	0.023 SlamDunk-SIFT [5]

caused by the sliding of ORB keypoints, as in the noisy ICL-NUIM sequence. The accuracy improvement for the matching-VO version is smaller in most of the sequences. However, as far as we can tell from the literature, the ATE RMSE values obtained with this version of PUT SLAM are among the best results published so far for the tested sequences (Tab. V).

VI. CONCLUSIONS

While joint optimization of features and poses is already used in computer vision and visual SLAM, this paper contributes a novel approach to the modeling of spatial uncertainty in point features. We demonstrate via simulations that the anisotropic model of uncertainty can significantly improve the accuracy of feature-based RGB-D SLAM, providing that the model captures the actual spread of feature measurements. Then we introduce the 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 experimental results are the basis to formulate two new spatial uncertainty models for features. The first one (normal-based) captures mainly the spread of measurements due to errors in the RGB image processing, while the second one (gradient-based) captures the random sliding of features along edges, and the increased depth errors on these edges. As demonstrated experimentally on the TUM RGB-D benchmark, the gradient-based model is appropriate for real RGB-D data. The accuracy results for the variant of PUT SLAM applying the tracking-VO are less impressive, but the gain due to the anisotropic uncertainty model is significant. As the tracking-VO variant is very fast (20 to 40 Hz, depending on the sequence, with no GPGPU acceleration) it is a viable alternative to the much slower matching-based variant (3 to 5 Hz).

Currently we are working on a combined uncertainty model that captures several factors influencing the spread of feature measurements, and on selecting the uncertainty model individually for each feature, depending on its location in the environment.

REFERENCES

[1] D. Belter, M. Nowicki, P. Skrzypczyński, "On the performance of pose-based RGB-D visual navigation systems", Computer Vision – ACCV 2014 (D. Cremers et al., eds.), LNCS Vol. 9004, Springer, 2015, 407–423.

[2] D. Belter, P. Skrzypczyński, "The importance of measurement uncertainty modeling in the feature-based RGB-D SLAM", Proc. of the 10th Int. Workshop on Robot Motion and Control, Poznań, 2015, 308–313.

[3] I. Dryanovski, R. Valenti, J. Xiao, "Fast visual odometry and mapping from RGB-D data", Proc. IEEE Int. Conf. on Robotics & Automation, Karlsruhe, 2013, 2305-2310.

[4] F. Endres, J. Hess, J. Sturm, D. Cremers, W. Burgard, "3-D mapping with an RGB-D camera", IEEE Trans. on Robotics, 30(1), 2014, 177–187.

[5] N. Fioraio, L. Di Stefano, "SlamDunk: Affordable real-time RGB-D SLAM", Computer Vision – ECCV 2014 Workshops, LNCS 8925, Springer, 2014, 401-414.

[6] A. Handa, T. Whelan, J. B. McDonald, A. J. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM", IEEE Int. Conf. on Robotics & Automation, Hong Kong, 2014, 1524–1531.

[7] P. Henry, M. Krainin, E. Herbst, X. Ren, D. Fox, "RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments", Int. Journal of Robot. Res., 31(5), 2012, 647-663.

[8] C. Kerl, J. Sturm, D. Cremers, "Robust odometry estimation for RGB-D cameras", Proc. IEEE Int. Conf. on Robotics & Automation, Karlsruhe, 2013, 3748–3754.

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

[10] G. Klein, D. Murray, "Parallel tracking and mapping for small AR workspaces", Proc. Int. Symp. on Mixed and Augmented Reality, Nara, 2007, 225–234.

[11] R. Kümerle, G. Grisetti, H. Strasdat, K. Konolige, W. Burgard, "g^{2}o: A general framework for graph optimisation", Proc. IEEE Int. Conf. on Robotics & Automation, Shanghai, 2011, 3607–3613.

[12] R. Maier, J. Sturm, D. Cremers, "Submap-based bundle adjustment for 3D reconstruction from RGB-D Data", Pattern Recognition, LNCS 8753, Springer, 2014, 54–65.

[13] R. Mur-Artal, J. M. M. Montiel, J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system", IEEE Trans. on Robotics, 31(5), 2015, 1147–1163.

[14] C. Nguyen, S. Izadi, D. Lovell, "Modeling Kinect sensor noise for improved 3D reconstruction and tracking", Proc. Int. Conf. 3DIMPVT, Zurich, 2012, 524–530.

[15] M. Nowicki, P. Skrzypczyński, "Combining photometric and depth data for lightweight and robust visual odometry", Proc. European Conference on Mobile Robots, Barcelona, 2013, 125–130.

[16] J-H. Park, Y.-D. Shin, J.-H. Bae, M.-H. Baeg, "Spatial uncertainty model for visual features using a Kinect sensor", Sensors, 12(7), 2012, 8640–8662.

[17] S. Scherer, D. Dube, A. Zell, "Using depth in visual simultaneous localisation and mapping", Proc. IEEE Int. Conf. on Robotics & Automation, St. Paul, 2012.

[18] H. Strasdat, A. J. Davison, J. Montiel, K. Konolige, "Double window optimisation for constant time visual SLAM", Proc. Int. Conf. on Computer Vision, Los Alamitos, 2011, 2352-2359.

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

[20] B. Triggs, P. F. McLauchlan, R. I., Hartley, A. W. Fitzgibbon, "Bundle adjustment – a modern synthesis", Vision Algorithms: Theory and Practice, LNCS 1883, Springer, 2000, 298–372.

[21] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns", IEEE Trans. on Pattern Analysis & Machine Intelligence, 13(4), 1991, 376–380.

[22] T. Whelan, H. Johannsson, M. Kaess, J. J. Leonard, J. B. McDonald, "Robust real-time visual odometry for dense RGB-D mapping", In: Proc. IEEE Int. Conf. on Robotics & Automation, Karlsruhe, 2013, 5704–5711.