2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Learning error models for graph SLAM

Christophe Reymann[1] and Simon Lacroix[1]

Abstract— **Following recent developments, this paper investigates the possibility to predict uncertainty models for monocular graph SLAM using topological features of the problem. An architecture to learn relative (i.e. inter-keyframe) uncertainty models using the resistance distance in the covisibility graph is presented. The proposed architecture is applied to simulated UAV coverage path planning trajectories and an analysis of the approaches strengths and shortcomings is provided.**

I. Introduction

We are aiming at developing an active mapping scheme in the context of large crop monitoring missions, or more generally for surface coverage missions with UAVs. Planning and adapting observation trajectories requires two abilities: compute a world model online, and estimate an associated error model from which the information content of future trajectories can be assessed.

The current operational solution for coverage mapping with UAVs is to feed a bundle adjustment (BA) technique with images acquired by an on-board camera: this requires heavy post-processing. Progresses in visual SLAM, and in particular in monocular graph SLAM approaches [19], [14], let seriously consider the possibility to achieve on-line mapping with a precision comparable to off-line BA techniques. Relying on such a mapping technique, one can develop *active SLAM* schemes, for which both an estimation and predictive error models are keys. Yet, defining such models remains a difficult problem, especially for graph SLAM approaches, where the extraction of a precise information matrix from the result of the optimization process is not straightforward.

This paper introduces an approach to learn a SLAM error model that explicits the errors on *all* the relative pose estimates of a Graph SLAM approach, so as to yield active information gathering strategies. Building on the seminal work of [9], we propose an architecture to learn relative error metrics between any pair of keyframes. The input of the learning architecture is a set of signatures of the structure of the covisibility graph maintained by the SLAM algorithm, as well as features computed from statistics on each edge of the graph. This error model also yields a prediction ability: new observations features can be inferred by a regression technique, from which covariance matrices can be predicted.

II. Related work

All active SLAM approaches have in common the need to compute the utility of an action, which has to rely on an uncertainty model of the robot pose and of the environment [1], [3], [2]. Fortunately, both Kalman filter and graph optimization based approaches share the property of being probabilistic frameworks and therefore maintain a representation of uncertainty, in the form of a covariance or information matrix. Computing the utility of new observations is done by using the current model as prior, and predicting the effect on the model by computing the posterior distribution after the integration of the new observations. In general, this is an intractable problem and therefore several approximations have to be made, such as assuming isotropic Gaussian noise and using a simple fixed-variance model for unknown locations.

Numerous issues hinder the precise computation of uncertainties. Indeed the uncertainties produced by the SLAM algorithms in real world scenarios are often optimistic: this is due to imprecisions and simplifications in the observation models, such as unmodeled correlations between observations of the same landmark (*e.g.* due to sensor calibration biases) and outliers. Some of these issues have been addressed in the literature, as in [20] for EKF, where adaptively "inflating" the prior uncertainty matrices with some hand-tuned parameters, and replacing EKF updates with a covariance intersection technique that integrates a correlation factor between measurements to lead to more conservative estimates.

Probabilistic reasoning on topological maps introduced in [17], [18] has shown to be helpful to maintain the global consistency of the graph and close loops. Active SLAM planning on topological maps has been developed [13], reasoning on the entropy of the produced topological graph. More recently, [12] proposes to plan on reasoning on the topological properties of the factor graph produced by the SLAM algorithm. It exploits the recent findings of [10], [11], that state that the topological properties of the factor graph is determinant of the accuracy of the estimation. It avoids any Bayesian reasoning, and the developed criteria depends only of the degree of the vertices is very easy to compute.

The seminal work of [9] explores the relationship between pose graph topology and the uncertainty estimate recovered from the information matrix of the maximum likelihood estimate in the 2D pose graph SLAM problem. Reasoning on the pose graph, with weight on edges taken as the measurement translation and rotational covariance, the authors find three interesting indicators linking graph topology with the SLAM information matrix. In particular, the weighted number of spanning trees is linked to the volume of confidence ellipsoids through the determinant of the Fisher information matrix. It is shown in [11] that the determinant converges to a pure function of the weighted number of spanning trees when a parameter δ converges to zero. This parameter depends only on the degree of the vertices, the sensing range and the precision of translation measurements. These results are proven in the case of planar SLAM, and empirical evidences seem to indicate a linear relationship between the log determinant of the Fisher information matrix and the tree connectivity for 3D pose graph SLAM.

[1] LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France

The results of this work are highly effective when dealing with pose graph SLAM with an accurate error model on the relative error measurements. However in the case of feature based SLAM with indirect measurements, such as in monocular SLAM, this information is not directly available – and applying this approach to the full graph of poses and landmarks is not possible, this graph being several orders of magnitude larger than a pose graph.

III. Learning the SLAM error model

In our UAV coverage context, we use a SLAM solution that exploits a landmark-based graphical model formulation solved by MAP estimation [14]. Landmarks are key points features that are tracked in the images, and a keyframe selection process selects the images which position constitute vertices in the factor graph. The estimation of the locations of landmarks and keyframes is posed as a non-linear least square problem, and solved by an optimization method.

A. The covisibility pose graph

The landmark based graphical model formulation of monocular SLAM builds a bipartite factor graph. Vertices either represent keyframe poses or landmark positions (Fig. 1), and factors encode the reprojection error on the image of landmarks, given the detected position of the landmark in the image, the estimated pose of the camera and the estimated position of the landmark.

We call *covisibility graph* the undirected graph G_α derived from the factor graph, keeping only the camera pose vertices and adding edges between two vertices if they share covisible landmarks (Fig. 1). Each edge (i, j) has a weight α_{ij}, which represents the tightness of the constraints linking i and j camera poses.

This covisibility graph is used in [14] to prune redundant observations while keeping the global topology intact, to select edges involved in local optimizations when adding a new keyframe, and to define the "essential graph" on which optimization is performed when closing loops.

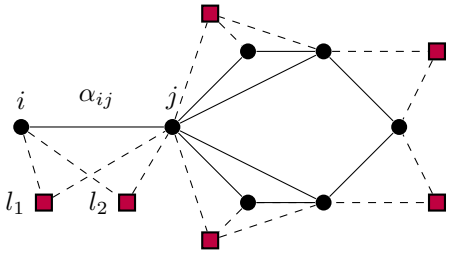


Fig. 1. A full landmark-pose constraint graph and the associated covisibility graph. ● correspond to camera poses, and ■ to landmark positions. All edges define the factor graph, whereas only solid edges define the covisibility graph (landmarks are not part of the covisibility graph).

B. The resistance distance

Relying on and expanding Khossoussi's work, we aim at instantiating the covisibility graph G_α so as to derive uncertainty estimates between any keyframe poses. In order to exploit this graph to plan further observations, the uncertainty estimates must be as close as possible to the actual error, and the graph must yield the possibility to assess the impact of future observations. The challenge is to generate meaningful α weights: we propose an architecture that enables the learning of these weights from errors measured with respect to the ground truth positions.

Khossoussi shows that the structure of the covisibility graph is correlated to the volume of the uncertainty ellipsoids through the (weighted) number of spanning trees in the graph. The resistance distance measure is a well grounded measure, used in various contexts such as Markov chains and networking problems (for a primer on graph resistance distance see [6], [5]). We postulate that the resistance distance between two vertices in the covisibility graph G_α is correlated to the relative error between the estimate of the corresponding keyframes.

The resistance distance can be computed directly from the weighted Laplacian matrix L_α, or conductance matrix, of the graph G_α:

$$L_\alpha = A\,\mathrm{diag}(\alpha)A^T \tag{1}$$

where $A \in \mathbb{R}^{n\times m}$ is the incidence matrix of G and $\mathrm{diag}(\alpha) \in \mathbb{R}^{m\times m}$ is the diagonal matrix composed from the edges weights, also called conductances.

Let Γ_α be the Moore-Penrose pseudoinverse of L_α. The effective resistance distance between vertices i and j can be computed using:

$$R_{ij} = (\Gamma_\alpha)_{ii} + (\Gamma_\alpha)_{jj} - 2(\Gamma_\alpha)_{ij} \tag{2}$$

The usage of the pseudoinverse in this definition allows for more robustness to ill-conditioning of the L_α matrix.

As its name denotes, the resistance distance R is a distance function, and therefore defines a metric on G_α. Intuitively, R_{ij} is small when there are many paths between vertices i and j with high conductance, and high when there are few paths with low conductance. Adding a new edge, *i.e.* a new edge always lowers the resistance distance between vertices. Thus it behaves as is expected of a relative error measure in SLAM: adding new measurements linking robot poses always lowers their relative localization error. Increasing the uncertainty of relative measures, thus decreasing the amount of information it encodes, increases the error.

C. Learning the relative error with the resistance distance

The resistance distance seems an interesting proxy for the relative error between vertices. To exploit it to precisely estimate errors, two questions must be answered: How to compute the weights α? How to derive the relative error from the resistance distance?

We propose an architecture that combines neural networks with the resistance distance to estimate the relative errors from the full graphical representation of SLAM, illustrated Fig. 2. The process consists of the following four steps:

1) From two keyframes i, j that share covisible landmarks (*i.e.* an edge of the covisibility graph G_α), we extract a feature vector X_{ij} that encodes how well the two keyframes would be colocalized from the matched landmarks.

2) We learn a function f mapping features to the weight edges of G_α.

3) From G_α we compute the resistance distance R_{kl} between any pair of vertices $(k, l) \in V$ (be they connected by edges in G_α or not).

4) Then a second learned function g maps the resistance distance R_{kl} to the relative error metric $\hat{e}_{kl}$.

If needed w independent metrics can be learned in parallel by using w outputs for f and inputs for g, thus computing the resistance distance on independent $G_{\alpha_0} \cdots G_{\alpha_w}$ graphs.

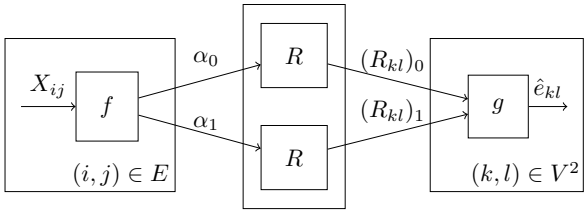


Fig. 2. Architecture of the function computing the SLAM error estimate $\hat{e}_{kl}$ for $(k,l) \in V^2$. $\alpha_{0..1}$ are the learned edge weight vectors associated to the covisibility graph $G = (V, E)$. The function f computes the weights $\alpha_{0..1}$ from the feature vectors $X_{ij} \in \mathbb{R}^k$, R_{kl} computes a scalar value for any two vertices in G_α, and g transforms the output of R_{kl} in a relative error metric between vertices k and l.

Because the resistance distance is differentiable, we can learn f and g together using neural networks. We simply need to define a cost function relating the estimated relative error $\hat{e}_{kl}$ to the ground truth error $\bar{e}_{kl}$, and use backpropagation to compute the gradients on the neural network parameters of f and g.

IV. Implementation of the learning architecture

A. Selecting informative features

We handpicked features capturing as best as we thought the nature of the constraints between keyframes, while remaining easy to compute. They represent geometric and image-content information and are defined on the basis of the covisible landmarks between the two keyframes:

- the number of covisible landmarks
- the total number of additional observations per landmark (*i.e.* the number of other keyframes in which they are visible)
- the parallax defined by the keyframes camera poses
- the average distance between the landmark and the keyframes positions
- the global saliency of the landmarks (defined in the dictionary used for the bag of words place recognition of ORB_SLAM2)

Histograms are computed with the parallax, distance, number of observations and global saliency features associated to all covisible landmarks of an image pair. These histograms, the number of covisible landmarks and the area of the overlap between the two considered images are aggregated in the feature vector X_{ij}.

B. Loss function

The definition of an appropriate loss function for the optimization is conditioned by the definition of the error model for $\bar{e}_{kl}$ that we are trying to learn. Ideally, one would want to produce a full six dimensional error model for the poses. However, our learning architecture rather explicits synthetic topological information than metric information and relationships between the variables estimated by the SLAM. It is therefore impossible with this model to predict the full error model defined by the covariances between the 6 pose parameters. Following the results and observations of [8], we can however hope to learn synthetic information about the pose uncertainty ellipsoids.

Because we can not differentiate the dimensions, we aim at learning the norm of the position error, not modeling the rotational error. Let $\bar{p}_k$ be the ground truth position for vertex k, and $\hat{p}_k$ the SLAM position estimate for the same vertex. We define the relative positional error norm as:

$$\bar{e}_{kl} = \parallel (\hat{p}_k - \hat{p}_l) - (\bar{p}_k - \bar{p}_l) \parallel \tag{3}$$

The probabilities of the relative errors are supposed independent of each other given the model, therefore for n graph samples with m vertices, we can write the joint probability of the concatenated error vector $\bar{\mathbf{e}}$ given the model:

$$p(\bar{\mathbf{e}} \mid \theta, X) = \prod_{i=1}^{n} \prod_{k=1}^{m_i - 1} \prod_{l=k+1}^{m_i} p(\bar{e}_{ikl} \mid \theta, X_i) \tag{4}$$

with θ the model parameters (in our case the network weights of f and g) and X the feature vector corresponding to $\bar{\mathbf{e}}$. The objective of the optimization is to find the parameters θ^* maximizing the joint probability of the data given the model:

$$\theta^* = \arg\max_{\theta} \; p(\bar{\mathbf{e}} \mid \theta, X) = \arg\max_{\theta} \; \log p(\bar{\mathbf{e}} \mid \theta, X) \tag{5}$$

Thus after substituting 4:

$$\theta^* = \arg\max_{\theta} \sum_{i=1}^{n} \sum_{k=1}^{m_i - 1} \sum_{l=k+1}^{m_i} \log \; p(\bar{e}_{ikl} \mid \theta, X_i) \tag{6}$$

Assuming an unbiased Gaussian error model on the poses, we compute the probability of observing an error as $\bar{e}_{ikl}$ given the learned error standard deviation $\sigma_{ikl} = \hat{e}_{kl}(\theta, X_i)$:

$$p(\bar{e}_{kl} \mid \theta, X) = \mathcal{N}(0, \sigma_{ikl}) = \frac{1}{\sqrt{2\pi\sigma_{ikl}^2}} e^{-\frac{1}{2}\left(\frac{\bar{e}_{ikl}}{\sigma_{ikl}}\right)^2} \tag{7}$$

Removing all terms not affecting the maximum, and switching to minimizing the negative log probability:

$$\begin{aligned} \theta^* &= \arg\min_{\theta} \; -\log p(\bar{\mathbf{e}} \mid \theta, X) \\ &= \arg\min_{\theta} \sum_{i=1}^{n} \sum_{k=1}^{m_i - 1} \sum_{l=i+1}^{m_i} \log(\sigma_{ikl}^2) + \left(\frac{\bar{e}_{ikl}}{\sigma_{ikl}}\right)^2 \end{aligned} \tag{8}$$

with predicted standard deviations $\sigma_{ikl} = \hat{e}_{kl}(\theta, X_i)$.

We use equation 8 to compute the loss function for the neural network architecture. Backpropagation allows the computation of the gradients of the loss function with respect to θ, as all intermediary operations are differentiable. The loss function can then be minimized using any gradient descent algorithm.

V. Results

We present results obtained by applying the learning architecture on a simulated dataset relating to coverage path planning of crop fields. The simulation setup allows to generate at will datasets with precise ground truth positions, while being faithful enough to evaluate the SLAM and error prediction architecture. An alternate to simulated runs would

be to exploit real datasets, resorting to a full BA to define the keyframe poses ground truth.

A. Simulation setup

The simulation setup instantiates a plan / execute / perceive loop in the context of UAV coverage. The environment is simulated with high resolution $20\,cm$ orthorectified image data[1] projected on a ground plane. The scenes are rural plains, with numerous crop parcels, tar and dirt roads, sparse trees and bushes... Three 12×12 kilometer tiles define three different environments.

A planning algorithm generates a Dubins trajectory covering the area to be mapped in a boustrophedon manner [7]. A value for the overlap of images on the ground is set: it defines the distance between parallel trajectory legs.

The plan is fed to a UAV flight simulator following the Dubins trajectory perfectly, but with added noise to introduce variability. The camera is simulated as if stabilized in roll and pitch, barring small perturbations of a few degrees. This model is not perfectly realistic, but it is good enough to evaluate the SLAM algorithm performances in the presence of moderate perturbations to the trajectory.

The UAV flies at a constant altitude of $150\,m$, and a constant $15\,m.s^{-1}$ airspeed. During flight, the position of the UAV is transmitted to the Morse simulator [4], where a $61°$ FoV 800×600 pixels camera observes the overflown scene at 32 Hz. The images are then processed by a modified version of the ORB_SLAM2 monocular SLAM software [15].

The dataset comprises of 5 areas to be mapped, with very distinct shapes. Every hundred new generated keyframes, the covisibility graph with computed features as well as the SLAM and ground truth poses are recorded. The overflown areas range from a half to a few km in length and width. 80 runs were performed with a ground image overlap varying from 30 to 90%, and a turn radius between 35 and $50m$.

B. Learning setup

Simulation results were aggregated and the dataset used to test the learning architecture, adding up to about four thousand SLAM results, with a large variety of trajectories, thanks to the variations on the overlap and turn radius. In addition, taking samples every hundred new keyframes ensures diversity in the graph sizes, all the more since ORB_SLAM2 has a keyframe culling routine, which induces strong modifications of the graph when closing loops, especially with back and forth boustrophedon trajectories.

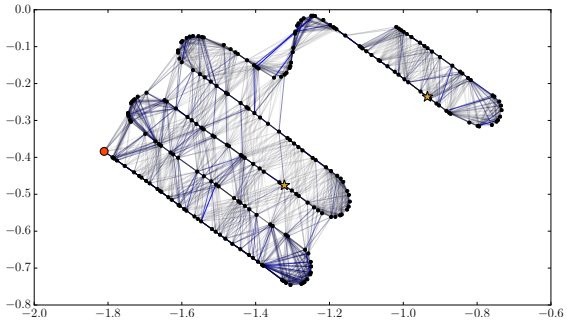
Fully connected neural network layers with rectified linear units were used for the f and g functions. A manual trial and error process was used to set the hyperparameters of the network, number of layers and units per layers. Function f has 4 hidden layers with 700, 100, 100 and 10 units respectively, and the g function has 2 hidden layers of 10 units each. 3 different weight sets $\alpha_0 \dots \alpha_2$ are learned in parallel and the 3 resistance distances computed for each edge kl are combined by g to produce the σ_{kl} output. To ensure that no arcs disappear in the covisibility graph, a minimum threshold is used on the weights, which is learned along with the other parameters of the network. Finally another fixed minimum threshold on the output σ_{kl} is set to 1e-5 to avoid division by 0 in the loss function (Equ. 8).

[1] from the french *National Geographic Institute* http://ign.fr

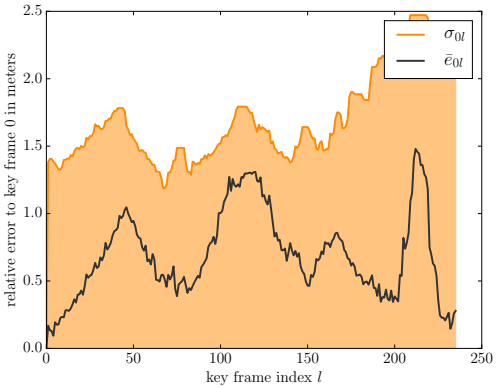
C. Qualitative analysis

We depict here 3 representative examples, using models learned on the whole dataset, excluding all the trajectories from the same mapping mission as the considered example.

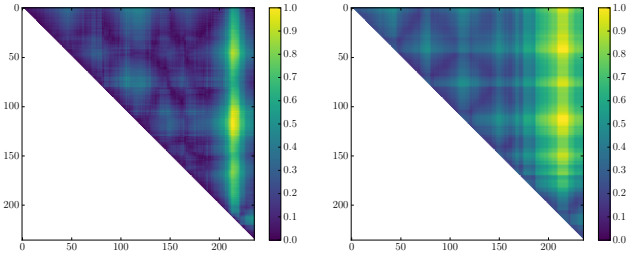
a) Trajectory A: This first example is a simple boustrophedon trajectory. Fig. 3(a) shows the covisibility graph: there are many loop closures between the trajectory first four legs, and a larger gap at the end.



(a) Covisibility graph (coordinates in km). Every keyframe is marked with •, while ● marks the reference keyframe and ★ marks every hundredth keyframe. Covisibility edges showing the learned weights are drawn in shades ranging from solid blue (highest weights) to pale grey (lowest weights.)



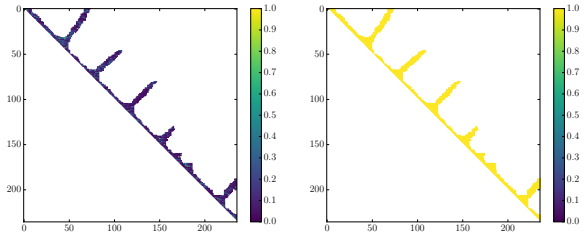
(b) Relative errors with respect to keyframe 0 (● above). Ground truth $\bar{e}_{0l}$ is drawn in black, the shaded orange area covers the predicted 1-σ standard deviation σ_{0l}.



(c) Matrices of the relative errors between keyframes k and l. Left: normalized ground truth relative error $\bar{e}_{kl}$, right: normalized predicted standard deviation σ_{kl}.

Fig. 3. Trajectory A results

Errors relative to the first keyframe are shown in Fig. 3(b), along with the 1-σ predicted uncertainty. One has to keep



(a) Normalized G_{α_0} (b) Normalized G_{α_1}

Fig. 4. Weighted adjacency matrices $G_{\alpha_{0,1}}$ for example A (normalized weight encoded in colors).

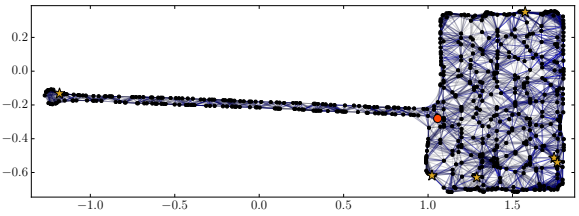
in mind that the observed error is only one realization of the predicted probability distribution, and is hence not necessarily representative of the quality of the prediction. One can however see interesting topological results. As expected, the error grows with the distance to the reference keyframe, and loop closures (near keyframes 75 and 150) bring the predicted error down. We can also observe the growth of the predicted uncertainty in the part weakly connected to the other parts (indices > 175), as well as an observed error peak in the same region. Here the predictions globally follow the same pattern as the actual errors – of course they are probabilistic in nature, and there may be deviations with the actual error, as we will see in other examples.

Fig. 3(c) shows the whole relative error matrices, measured and predicted (Fig. 3(b) actually plots the first line of these 2 matrices). We find the same global structure in the uncertainty estimate as in the ground truth error, with the prediction seemingly being more conservative in the error estimate. We find again the error peak in the band around index 215, which correspond to the last turn of the trajectory ($x = -0.7km$, $y = -0.3km$) that is topologically the farthest away from the well connected regions of the beginning. Local maximums and minimums seem to correlate well with the back and forth motions that close numerous local loops.

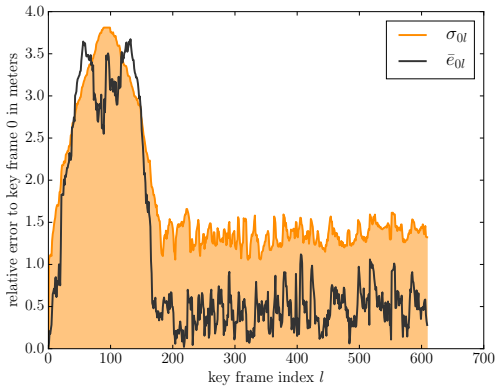
Finally we can observe the intermediate learned weighted adjacency matrices of the covisibility graph G_{α_0} and G_{α_1} in Fig. 4. In (a), we can clearly see that a variety of weights have been learned, however (b) shows for the second learned weight matrix a purely topological adjacency: all weights have the same value which corresponds to the lowest possible value given the threshold. Adding more intermediate G_α seem to always produce only one informative weight matrix, the other being (or very close to) a simple adjacency matrix multiplied by the value of the threshold. We conjecture this indicate that the structure of the problem as posed only needs one distance metric to perform the prediction without adding redundant information, at least given the input features that were used in our implementation. Indeed, using only one intermediate G_α produced very similar results in the values of the loss function, and this applies to all explored examples.

b) Trajectory B: This sample exhibits a single, long back and forth trajectory that links to a well connected area, resembling the shape of a hammer (Fig. 5(a)). Here we expect the largest relative errors between the hammer head and the base of the shaft.

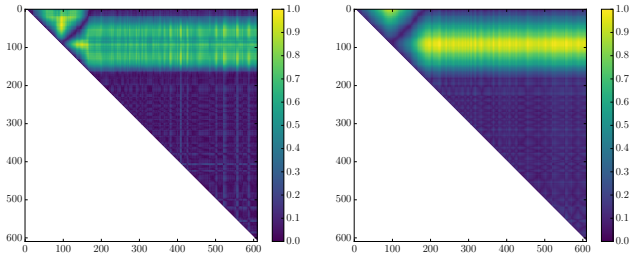
In this instance, the errors relative to the starting keyframe



(a) Covisibility graph (coordinates in km). The trajectory starts at the meeting point of the shaft and the head of the hammer, goes forth and back along the shaft, and then maps the head along back and forth motions.



(b) Relative errors with respect to keyframe 0 (above)



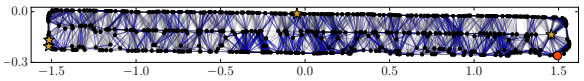
(c) Normalized relative error matrices (ground truth $\bar{e}_{kl}$ and predicted σ_{kl})

Fig. 5. Trajectory B results

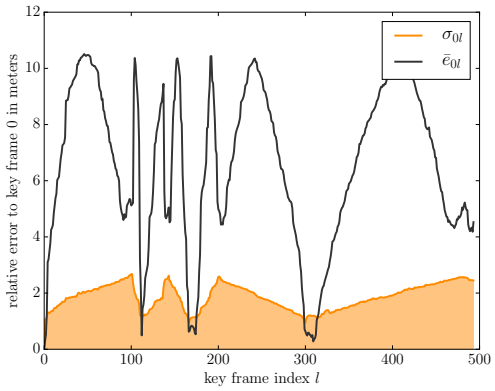
barely fit in the predicted 1-σ envelope, as shown Fig. 5(b). The predicted uncertainty is as expected at its maximum at the base of the shaft, around index 100. However, due to its random walk nature, the error exhibits two peaks corresponding to a maximum error in the middle of the shaft.

This structure can be found again in the relative error and uncertainty prediction matrices (Fig. 5(c)). We clearly see the bands corresponding to the shaft, as well as the dual peaks in the ground truth error.

c) Trajectory C: This example shows a case with very high errors, quite quickly accumulated, resulting in the prediction failing to produce probable results. The executed trajectory can be guessed by the covisibility graph Fig. 6(a). In this case the relative errors to the first keyframe attain higher than average values, with peaks around 10m due to a $3\,km$ long motion without loop closure, and with rapidly varying errors (Fig. 6(b)). In contrast, the prediction never rises above 2m, which results in errors deviating by more than 5σ from the model. Here the model seems to be unable to cope with very high and rapidly changing errors.

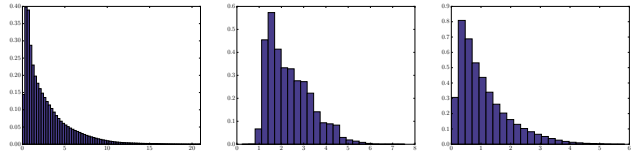


(a) Covisibility graph (coordinates in km)



(b) Relative errors with respect to keyframe 0 (● above). Prediction deviations wrt. the maximum ground truth error attains up to 5σ.

Fig. 6. Trajectory C results



(a) ground truth error $\bar{e}$ (b) predicted standard deviation σ (c) deviation from the prediction $\bar{e}/\sigma$

Fig. 7. Histograms of the ground truth error $\bar{e}$ (in meters), the predicted standard deviation σ (in meters) and the deviation of the ground truth error from the predicted standard deviation $\bar{e}/\sigma$ across 400 random samples, or about 10% of the dataset.

D. Quantitative results

We now examine statistical results computed on 400 trajectory samples, or about 10% of the dataset, for a total of about 59 millions data points. Three histograms of the ground truth error values $\bar{e}$, of the predicted standard deviation σ, and of the deviations of the error from the predicted standard deviation $\bar{e}/\sigma$ are shown Fig. 7.

As can be seen in Fig. 7(a) the maximum error lies around 20 meters, while the maximum predicted standard deviation is under 8 meters (7(b)). The peak of standard deviation prediction is around 1.5 meters and almost no predicted standard deviations lie under $1\,m$. Looking at the error, we see a slight bias: the peak is shifted around $25 - 50$cm. As well, the peak of the deviation $\bar{e}/\sigma$ (7(c)) is shifted around $0.25 - 0.5\sigma$. In the absence of bias the peak should be at zero in a normal distribution.

The maximum event is over five but under six sigma. Events over 5σ are expected (approximately 1 every 1.7 millions), but events over 6σ would not be (approximately 1 every 506 millions). Fractions of the population lying inside the n-σ range, as well as the expected fraction value in a normal distribution are compiled in table 8. These indicate a slight overconfidence tendency of the uncertainty model, as well as a distribution with heavier tails. Possible explanations include the aforementioned bias, as well as the failure to predict very large and rapid deviations, as in trajectory C.

Range	Realized fraction	Expected fraction
1σ	0.57	0.68
2σ	0.86	0.95
3σ	0.96	0.997
4σ	0.992	0.99993
5σ	0.9995	0.9999994

Fig. 8. Fraction of the errors lying in the n-σ range of the predicted distribution and expected fraction according to the normal distribution. Statistics computed across 400 samples (about 10% of the dataset), on 59 millions data points.

VI. Discussion

We have presented a novel approach to learn a relative error model for a monocular SLAM algorithm, using mostly topological information from the weighted covisibility graph through the resistance distance, which shows promising results. Although slightly overconfident, the learned model captures well the error variations caused by the graph topology, especially large scale topological features, whereas finer variations seem less precisely modeled.

This work is only a first step in the direction of mixing topological and metric information to predict SLAM uncertainties. Even with simplified simulations (no harsh rotations, flat ground), predictions are not perfect. The model fails to capture the larger uncertainties, especially with rapid variations, and sometimes smoothes too much the structure of the errors in the graph. As proven in [8] in the case of 2D SLAM, topological information and sensor precision alone are not sufficient to completely explain the volume of the uncertainty ellipsoids: metric information (*e.g.* the distance between keyframes) are also necessary. In our work, the few metric information included in the features are probably too rudimentary, and the reliance on the SLAM solution for the computations of these features introduces bias.

Besides the need to engineer better features, other factors should be considered. In particular, the observed bias of the ground truth error seems to indicate a deviation from the zero-mean Gaussian distributed error (which we use in our model), that grows with the relative distance: this may be related to the lever effect of angular errors on the pose errors, which our model does not account for.

An interesting direction for future work would be produce richer models, with a more complex architecture, integrating more metric information to predict a full error model (on each coordinate, including rotation angles). One of the difficulties is to engineer a loss function that enables to mix positional and rotational errors with widely different scales. A possibility is to drive the learning so as to minimize the errors on the landmarks poses. The rotation errors having a strong non linear effect, integrating the error projection in the loss function is however not trivial. An alternate solution is to exploit the loss function based on a matrix Lie groups approach, as introduced in [16].

REFERENCES

[1] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, Jose Neira, Ian Reid, and John J. Leonard. Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age. *IEEE Transactions on Robotics*, 32(6):1309–1332, December 2016.

[2] Henry Carrillo, Philip Dames, Vijay Kumar, and Jose A. Castellanos. Autonomous robotic exploration using occupancy grid maps and graph SLAM based on Shannon and Rényi Entropy. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 487–494, Seattle, WA, USA, May 2015. IEEE.

[3] Henry Carrillo, Ian Reid, and Jose A. Castellanos. On the comparison of uncertainty criteria for active SLAM. In *2012 IEEE International Conference on Robotics and Automation*, pages 2080–2087, St Paul, MN, USA, May 2012. IEEE.

[4] G. Echeverria, S. Lemaignan, A. Degroote, S. Lacroix, M. Karg, P. Koch, C. Lesire, and S. Stinckwich. Simulating Complex Robotic Scenarios with MORSE. In *3rd International Conference on Simulation, Modeling, and Programming for Autonomous Robots, Tsukuba (Japan)*, Nov. 2012.

[5] W. Ellens, F.M. Spieksma, P. Van Mieghem, A. Jamakovic, and R.E. Kooij. Effective graph resistance. *Linear Algebra and its Applications*, 435(10):2491–2506, November 2011.

[6] Arpita Ghosh, Stephen Boyd, and Amin Saberi. Minimizing Effective Resistance of a Graph. *SIAM Review*, 50(1):37–66, January 2008.

[7] Nicolas Holvoet. Planning for active mapping of crop fields using UAVs. Master's thesis, ENAC, September 2018.

[8] Kasra Khosoussi. *Exploiting the Intrinsic Structures of Simultaneous Localization and Mapping*. PhD thesis, UTS, 2017.

[9] Kasra Khosoussi, Matthew Giamou, Gaurav S Sukhatme, Shoudong Huang, Gamini Dissanayake, and Jonathan P How. Reliable graphs for slam. *The International Journal of Robotics Research*, 38(2-3):260–298, 2019.

[10] Kasra Khosoussi, Shoudong Huang, and Gamini Dissanayake. Novel insights into the impact of graph structure on SLAM. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference On*, pages 2707–2714. IEEE, 2014.

[11] Kasra Khosoussi, Shoudong Huang, and Gamini Dissanayake. Good, bad and ugly graphs for SLAM. In *RSS Workshop on the Problem of Mobile Sensors*, 2015.

[12] Andrej Kitanov and Vadim Indelman. Topological Multi-Robot Belief Space Planning in Unknown Environments. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7, Brisbane, QLD, May 2018. IEEE.

[13] Beipeng Mu, Matthew Giamou, Liam Paull, Ali-akbar Aghamohammadi, John Leonard, and Jonathan How. Information-based active SLAM via topological feature graphs. In *Decision and Control (CDC), 2016 IEEE 55th Conference On*, pages 5583–5590. IEEE, 2016.

[14] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5):1147–1163, October 2015.

[15] Raul Mur-Artal and Juan D. Tardos. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[16] V. Peretroukhin and J. Kelly. Dpc-net: Deep pose correction for visual localization. *IEEE Robotics and Automation Letters*, 3(3):2424–2431, July 2018.

[17] A. Ranganathan and F. Dellaert. Inference in the space of topological maps: An MCMC-based approach. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 2, pages 1518–1523, Sendai, Japan, 2004. IEEE.

[18] Ananth Ranganathan and Frank Dellaert. Online probabilistic topological mapping. *The International Journal of Robotics Research*, 30(6):755–771, May 2011.

[19] Georges Younes, Daniel Asmar, Elie Shammas, and John Zelek. Keyframe-based monocular SLAM: design, survey, and future directions. *Robotics and Autonomous Systems*, 98, 2017.

[20] Zhen Zhu and Clark Taylor. Conservative Uncertainty Estimation in Map-Based Vision-Aided Navigation. *IEEE Transactions on Aerospace and Electronic Systems*, 53(2):941–949, April 2017.