2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Daejeon Convention Center
October 9-14, 2016, Daejeon, Korea

Erasing bad memories: agent-side summarization for long-term mapping

Marcin Dymczyk, Thomas Schneider, Igor Gilitschenski, Roland Siegwart, and Elena Stumm
Autonomous Systems Lab, ETH Zurich

***Abstract*— Precisely estimating the pose of an agent in a global reference frame is a crucial goal that unlocks a multitude of robotic applications, including autonomous navigation and collaboration. In order to achieve this, current state-of-the-art localization approaches collect data provided by one or more agents and create a single, consistent localization map, maintained over time. However, with the introduction of lengthier sorties and the growing size of the environments, data transfers between the backend server where the global map is stored and the agents are becoming prohibitively large. While some existing methods partially address this issue by building compact *summary maps*, the data transfer from the agents to the backend can still easily become unmanageable.**

In this paper, we propose a method that is designed to reduce the amount of data that needs to be transferred from the agent to the backend, functioning in large-scale, multi-session mapping scenarios. Our approach is based upon a landmark selection method that exploits information coming from multiple, possibly weak and correlated, landmark utility predictors; fused using learned feature coefficients. Such a selection yields a drastic reduction in data transfer while maintaining localization performance and the ability to efficiently summarize environments over time. We evaluate our approach on a data set that was autonomously collected in a dynamic indoor environment over a period of several months.

I. Introduction

Recent developments in robotics have pushed forward numerous new applications, where robots can provide invaluable help, e.g. by reaching previously inaccessible places or performing tasks more precisely than humans. A vital requirement for these capabilities is to accurately estimate the robot's pose with respect to the global environment, as well as other agents. Often, robots need to operate in GPS-denied areas, without any embedded localization beacons. This motivates the need for precise, high-frequency pose estimation made possible by using visual-inertial sensors paired with a localization framework.

Many mapping and localization scenarios consist of repeated visits to the same environment by one or more agents. In a typical setting, a mapping agent creates a map of the current session locally [4, 2]. Such a map is then transferred to a mapping backend which collects data from all the agents

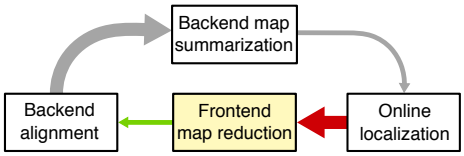

Fig. 1: We propose a frontend map reduction method that can be used to reduce the data flow from the mapping agent to the mapping backend.

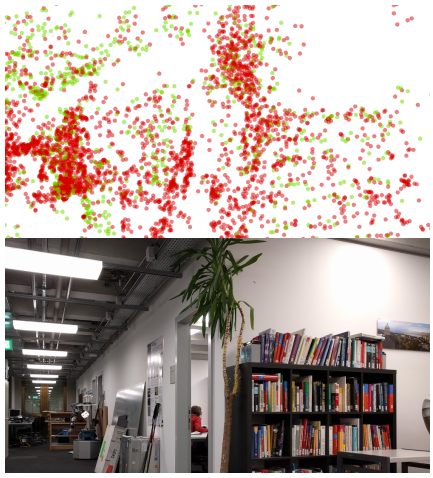

Fig. 2: Above: the result of the proposed method – features expected to be stable are marked in green and feature expected to be volatile and hard to redetect are marked in red. It is worth to note that the algorithm considers the ceiling structure as a source reliable landmarks, but rejects objects on the floor and the plant, as visible in the photo of the same area (below).

that is used to build and maintain a single consistent map of the environment [4, 5]. When a certain place is revisited, new information may be observed and used to refine the map quality or fill possible gaps. Once an agent needs to localize itself against the existing global map, it can request the information from the backend and place recognition can be used to retrieve a current pose [6, 1, 3]. In the case of mobile robotics, we can assume that the information is exchanged using a wireless transmission system. Thus, there is a considerable limitation in the network's bandwidth, which affects both the backend-to-agent transmission of the localization map as well as the agent-to-backend transmission of the raw, local map.

To overcome these bandwidth limitations, we would like to reduce the data flow between the backend and the agent (and vice versa). Existing work has mostly aimed at reducing the size of the localization map, by creating a so called Summary Map, which is then sent to the agents [7, 8, 9] (illustrated by the top block in Fig. 1). These approaches take advantage of repeated visits to the same location and derive statistics out of the collected data in an offline processing step. Their goal is to select a set of landmarks that permits reliable place recognition while ensuring coverage of the entire environment and keeping the transfer size as small as possible.

In this paper, our primary goal is to reduce the data flow from the mapping agent to the mapping backend (illustrated

978-1-5090-3761-2/16/$31.00 ©2016 IEEE

by the bottom half of Fig. 1). When not reduced, those transfers can reach sizes of over 50MB per 100m trajectory in a system using relatively compact 512-bit visual descriptors. To limit this bandwidth burden, we propose a method that only requires data from the current mapping session to predict and select those landmarks which are most likely to be useful for localization in the long term. To perform the prediction, we have come up with several metrics, called predictors, that we deem to describe aspects of this long-term consistency. The method then learns the relative influence of each predictor based on training data, therefore combining numerous (sometimes weak) sources of information in order to infer the overall quality of each landmark.

The proposed methodology is evaluated using a dataset recorded by an autonomous robot in a highly-dynamic office space (depicted in Fig. 2). The data was collected over a period of several months, with varying lighting conditions and numerous objects appearing and disappearing from the mapped area. The evaluation not only shows improvement over the state-of-the-art, but also provides insight into the relative merit of a variety of proposed landmark quality predictors (based on properties such as geometry, persistency, and appearance). By using the method, we are able to reduce the agent-backend data flow by 80% with a marginal localization recall drop of about 5%.

II. Related Work

Vision-based place recognition systems enable retrieval of precise 6-DoF pose information using relatively low-cost sensors, in GPS-denied environments, and without using any external beacons. Current state-of-the-art place recognition approaches range from using holistic image retrieval [12], over features obtained using deep-learning techniques [13], to local point features [14]. Working with local point features (landmarks), has the advantage of providing an intuitive framework for building up a 3D model of the environment, and these are therefore used in this work. This map can then serve as a database to localize from [15],[16].

In general, this process has a strong dependence on the quality of the underlying landmark detection and description. As a result, a range of work has been done to investigate and quantify the quality of visual descriptors. For example, in [17], the stability of feature descriptors over viewpoint and lighting changes is investigated, providing a measure of robustness. Similarly, the work of [18] focuses on learning repeatable detectors over drastic illumination changes. This work relies on training images capturing the same view-point over multiple illumination conditions.

Furthermore, in relation to localization, existing work has explored similar ideas of evaluating the quality of features (or landmarks) with respect to the stability and reliability of being detected repeatedly. For example, in [19] and [20] a subset of landmarks is selected based on the uniqueness of the descriptors. Additionally, the work of [10] evaluates the likelihood of a SIFT descriptor to be matched in subsequent observations, in a localization framework similar to the one presented in this paper.

In the context of robotics, and generally within the applications where the computational power is limited, binary descriptors such as BRISK [21] have gained popularity. While these descriptors are more compact, they also contain less information which reduces their discriminative power required for a robust matching. This motivates the need for methods that score the information content of such keypoints for localization, as proposed in this paper.

Perhaps the most similar work to that presented here is by Buoncompagni et al. [11], where image features are scored based on distinctiveness, repeatability, and detectability; then combined to give an overall saliency score. However, the weights used therein to combine these scores are hand-tuned, rather than learned automatically as we do here. Moreover, the work presented in [11] is applied to the task of object detection and recognition, and is therefore missing metrics specific to long-term localization.

Applications to long-term mapping scenarios certainly promote the need for map maintenance and selection of only the most relevant landmarks. Existing work has mostly aimed at maintaining the map over long time-horizons [8], removing outdated information by using a change-detection metric [22] or aggregating the experiences to eventually encompass all possible conditions [23]. As state-of-the-art systems push to cover larger and larger areas over longer time-scales, there exists a growing interest in reducing the amount of data that needs to be stored or transferred [24]. One of the ideas is to compress the global model into compact representations, so-called Summary Maps [7], [9], effectively subsampled versions of the centrally maintained map that are useful for localization. These approaches, however, assume access to all past sessions and history, which is generally only possible on a centralized mapping server.

Our goal is to take inspiration from existing summarization methods and deploy them in an online fashion on the agent. As a consequence, we would only send the most valuable parts of the local map to the backend, reducing the required transmission bandwidth and processing time. The main contributions of the paper are as follows:

- A presentation and evaluation of novel metrics for predicting a landmark's relevance.
- An extension of the matchability prediction approach presented in [10] to binary descriptors.
- A general framework, which learns the relative influence of proposed landmark quality predictors, in order to rank sets of landmarks in an online fashion.
- An evaluation of the entire approach in a realistic mapping scenario using data collected autonomously, spanning over several months.
- An analysis of the possible compression ratios that can be used on the agent-side while avoiding a significant loss in localization performance.
- A study of the proposed method used as a part of the system presented in [8] to confirm its advantageous long-term properties – getting better over time.

III. Methodology

Our approach to reducing the number of landmarks while keeping the most relevant ones is twofold. On the agent-side, we obtain a ranking of current landmarks, which serves as a decision criteria for selecting the ones that are most suitable for being sent to the server. This approach is described in the first subsection. Afterwards, we revisit an Integer Linear Programming based approach for keeping informative landmarks on the backend-side. This approach is then described in the second subsection.

A. Summarization on the agent-side

The merit of a given landmark for localization depends on various aspects. This work aims at exploring some of these by considering different features related to landmark observations in order to infer the respective significance of each landmark. Furthermore, by evaluating such features in conjunction, even relatively weak correlations in the data can be exploited to boost results. Thus, several evaluation metrics are proposed, and later combined in a regression framework. Using the results thereof, a landmark ranking policy can be defined such that only a subset of the most relevant landmarks is selected while still maintaining relocalization quality during future sessions.

Before evaluating the ranking features of each landmark i, we first define the following notation (further illustrated in Fig. 3):

- $p_{GL,i} = (p_{Gi,x}, p_{Gi,y}, p_{Gi,z})$ denotes the position of the landmark i in the global frame G,
- $p_{GV,j}$ denotes the position of the keyframe j in the global frame G,
- $b_{i,j} = \dfrac{p_{GLi} - p_{GVj}}{\|p_{GLi} - p_{GVj}\|_2}$ denotes a unit-length bearing vector corresponding to the ray between the feature i and the keyframe j,
- $\mathcal{S}_i$ is an ordered set of keyframes observing the feature i, i.e. if $j \in \mathcal{S}_i$ then j observes i,
- $m_{i,j}$ is the the keypoint measurement of the keyframe j, which got associated with the landmark i,
- $m'_{i,j} = (m_{x,i,j}; m_{y,i,j})$ represents the reprojection of a landmark i into the keyframe j, using the corresponding projection and distortion models.

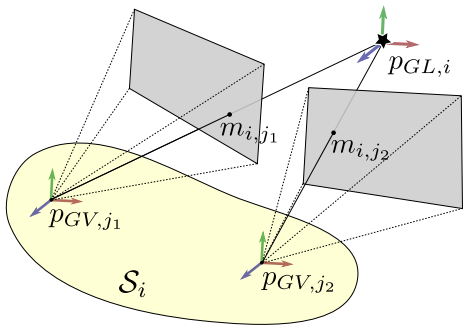

Fig. 3: A landmark located at $p_{GL,i}$ is observed from a set $\mathcal{S}_i$ of keyframes located at $p_{GV,j}$. For each keyframe, the location of the landmark in the image plane is given as $m_{i,j}$.

1) Predictor features: We will now present candidate features related to the observation of each landmark, which may aid in the prediction of its relevance with respect to localization. The goal is to model the probability of consistently redetecting and matching a landmark based on the local information, such as geometry and the descriptor pattern, from a single visit to its environment. We assume any long-term mapping data, such as previous landmark observations, is not available to us. The actual relationship between the proposed predictor variables and the empirical probability of redetecting the landmark will be evaluated in Section IV. The list of candidate features proposed in this paper is by no means complete, but we believe that it captures a range of relevant metrics, and demonstrates the value of combining several sources of information (even weak ones) for landmark selection.

Track length: We will begin by looking at properties related to local tracking information about each landmark during the current traversal. Track length is one of the simplest and most intuitive features, widely used by existing map reduction methods [24, 7, 9]. Its underlying assumption is: the more frames landmark i is reobserved in, the more probable it is that it can be observed from a large area and that it can easily be redetected. It is given by

$$\phi_i^l = |\mathcal{S}_i| \tag{1}$$

Distance traveled while observing the landmark: This is a variant of the track length predictor, which accounts for effects such as varying velocity or keyframe selection and is obtained as

$$\phi_i^d = \sum_{j \in \mathcal{S}_i} \|p_{GV,j+1} - p_{GV,j}\|_2 \tag{2}$$

Distance traveled between the two most distant keyframes on a track: Similarly, this brings robustness against trajectories that are meandering and have a large length even though they only cover a relatively small area. We believe this measure is complementary to the ones presented above. Its computation requires maximization over all keyframes observing the same landmark, i.e.

$$\phi_i^\Delta = \max_{j,j' \in \mathcal{S}_i} \|p_{GV,j'} - p_{GV,j}\|_2 \tag{3}$$

Maximum angle between observation rays: Additionally, we propose to not only use the distance along the track, but also the maximum angle spanned by all observation rays from the keyframes to the landmark. If this angle is relatively wide, the area where the landmark can be observed is likely to be large as well. As every $b_{i,j}$ is of unit length, the predictor can be efficiently calculated as a dot product:

$$\phi_i^b = \max_{j,j' \in \mathcal{S}_i} \cos^{-1} b_{i,j} \cdot b_{i,j'} \tag{4}$$

Mean reprojection error: Furthermore, apart from the landmark track geometry, it is also worth to consider the consistency of the map in the landmark's locality. The mean reprojection error of the landmark into all of the observing keyframes might be considered as a metric that contains

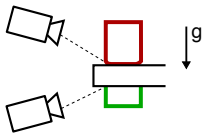

Fig. 4: The idea behind the gravity constraint predictor is that clutter cannot fly. Objects seen from below (the green rectangle in the image) must be fixed or anchored, and are therefore expected to be relatively persistent. On the other hand, features seen from above (the red cup) are not necessarily anchored and can be easily displaced or removed from the environment.

additional information about the utility of a landmark. A landmark that is triangulated from many observers with distinct positions (given that it has no incorrect keyframe associations) is likely to have a small reprojection error.

$$\phi_i^{\epsilon} = \frac{\sum_{j \in \mathcal{S}_i} \|m_{i,j} - m'_{i,j}\|_2}{|\mathcal{S}_i|} \tag{5}$$

Gravity constraint: Our aim is to select not only those landmarks which have a high chance of being redetected and matched, but those which also tend to be stable over longer periods of time. We therefore propose a predictor based on what we refer to as the gravity constraint. It disambiguates between landmarks that belong to objects which are located *on* something (and not necessarily anchored) from objects which are *hanging* from something and must be anchored, otherwise they would fall down. The basic concept is illustrated in Fig. 4. We believe the objects that are not anchored, such as objects placed on a desk, will prove to be less reliable for long-term localization. Assuming g is the gravity vector pointing down:

$$\begin{aligned} g &= (0, 0, -1) \ , \quad \|g\|_2 = 1 \ , \\ \phi_i^g &= \frac{\sum_{j \in \mathcal{S}_i} b_{i,j} \cdot g}{|\mathcal{S}_i|} \end{aligned} \tag{6}$$

Vertical coordinate: In the same vein, we also include a predictor that relates to the specific position of each landmark in the mapped environment. The vertical coordinate of the landmark's position may provide additional information about its long-term stability. For example, in our office environment, objects that are either on the floor (e.g. parcels) or within a person's reach (e.g. things on the table) change frequently, while objects located higher up tend to be more stable. As this quantity might not be linear with the landmark quality, we expect to use some nonlinear transformation f before using this feature in our model (see Section IV-C).

$$\phi_i^h = f(p_{Gi,z}) \tag{7}$$

Descriptor appearance classification: Following the methodology presented in [10] for SIFT descriptors, we propose a similar method for classifying the BRISK descriptors using Random Forests. We believe that information about the appearance pattern will be complimentary to the other predictor variables that are based on map geometry. Instead of the binary classification proposed by Hartmann et al., we have formed 5 classes corresponding to the landmark quality. We used Gini impurity as a split criterion and 100 trees in total. To prevent overfitting, which is very likely with the binary data, we limited the depth of a tree to 14. The output of the Random Forest is transformed as follows to form a single scalar that can be fed to the regression framework:

$$\phi_i^{rf} = \sum_{c=1}^{5} c \cdot P(\text{class} = c) \tag{8}$$

2) Regression: We would like to combine all of the features presented above into a single, scalar landmark relevance score, that can be used to rank the landmarks according to the predicted quality and select a subset of them. One way of achieving this is by using a regression algorithm and fitting the coefficients based on a labeled training set.

The training set labeling: We propose to label the training set based on the past evidence of observing a particular landmark in multiple datasets covering the same trajectory over longer periods of time. Calculating the ratio of the number of datasets in which the landmark ℓ_i was observed over the total number of datasets $|\mathcal{D}|$ approximates the empirical probability of observing the landmark in a new dataset:

$$P(\ell_i|\mathcal{D}) \approx \frac{\#\text{ of datasets observing the landmark}}{\text{total }\#\text{ of datasets}} \tag{9}$$

The landmark correspondences between the datasets can be established using an appearance-based feature matching algorithm. It is worth noting that since all the datasets were recorded when following the same trajectory, each portion of the environment is equally represented, and therefore persistent and reliable landmarks should be redetected in each traverse.

Feature selection: To confirm the true significance of the predictors presented above, we apply the Lasso regression method (see (10)). Depending on the value of λ, the method penalizes the absolute value of the coefficients β and might eventually set them to 0. We therefore cross-validate over the values of λ, look for the model with the highest prediction accuracy and observe which coefficients are not equal to zero. Assuming X is a matrix of predictors, y is the dependent variable and N is the number of samples, the Lasso regression can be formalized as:

$$\min_{\beta} \left\{ \frac{1}{N} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \right\} \tag{10}$$

Regression model: After performing the feature selection, we fit our final model using ridge regression. This technique penalizes the L-2 norm (instead of L-1 in the Lasso method) of the coefficients and is suitable for handling collinearity in the predictor variables. This is beneficial here, as many of the proposed predictor variables are strongly correlated (e.g. the distance along the track with the maximum angle between the rays).

B. Summarization on the backend-side

While the summarization on the frontend-side mostly focuses on information immediately available to the agent, the backend-side may additionally incorporate information from multiple agents and earlier traversals through the same environment. For the backend landmark selection we used

the method proposed in [9] which uses an Integer Linear Programming optimization to find the desired subset of landmarks. The problem can be described as:

$$\begin{aligned} &\text{minimize } \mathbf{q}^T\mathbf{x} + \lambda \mathbf{1}^T\boldsymbol{\zeta} \\ &\text{subject to } \mathbf{A}\mathbf{x} + \boldsymbol{\zeta} \geq b\mathbf{1} \\ &\sum_{i=1}^{N} \mathbf{x}_i = n_{desired} \end{aligned} \tag{11}$$

where $\mathbf{A}$ is the covisibility matrix, b is the desired number of landmarks per keyframe and $\boldsymbol{\zeta}$ is a slack variable. Each landmark is represented as a binary variable x_i and the goal is to find a subset of $n_{desired}$ landmarks. The landmark score $\mathbf{q}$ is based on the number of observing datasets, descriptor stability, as well as the within-dataset track length.

IV. Experiments

In order to evaluate our methodology, we used a Turtlebot[1] robot, autonomously collecting datasets in an office environment. We first present the experimental setup (Sec. IV-A) followed by a discussion on the preparation of the training set (Sec. IV-B), and a subsequent evaluation of the regression model (Sec. IV-C). We compare the presented approach to other selection methods (Sec. IV-D) and evaluate the long-term performance of the integrated summarization system (Sec. IV-E).

A. Experimental setup

The data for all the experiments and evaluation has been collected using a Turtlebot robot, depicted in Fig. 5. The navigation and local obstacle avoidance used during data collection was based on the Hokuyo 2D laser rangefinder and the stock ROS navigation stack [25]. Additionally, the robot was equipped with a VI-Sensor [26] to build visual-inertial maps for the long-term map maintenance experiments. Mapping data was collected over a 150 m long trajectory, twice a day, over a period of several months. Since the datasets are recorded within a functioning office environment, there are many examples of dynamic objects that are often moving, appearing, or disappearing on a daily basis (see Fig. 2). In addition, the lighting conditions vary significantly for an indoor environment, depending greatly on the weather conditions and the artificial light inside the building.

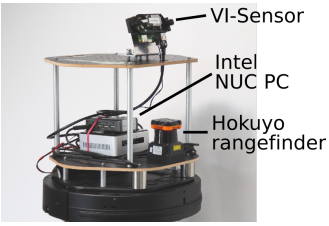

Fig. 5: The Turtlebot robot used to acquire the datasets.

The datasets were then fed into the mapping system described in [2], to perform evaluations and verify how the proposed agent-side map reduction method affects the short-term and long-term place recognition performance. For all of the following evaluations, we used a threshold of 40 cm when computing the precision of the pose retrieval. The ground-truth poses were obtained using a full-batch visual-inertial bundle adjustment of the trajectories, as is often done when no external motion tracking system data is available [3, 9].

[1]www.turtlebot.com

Predictor	Coefficient	R^2 change if excluded
Track length (frames), ϕ_i^l	0.0050	−0.0011
Total distance, ϕ_i^d	0.0095	−0.0043
Max. distance, ϕ_i^Δ	0.0101	−0.0061
Max. angle, ϕ_i^b	0.1109	−0.0019
Mean reproj. error, ϕ_i^ϵ	−0.0040	−0.0001
The gravity constraint, ϕ_i^g	0.1954	−0.0334
Z-coordinate, ϕ_i^h	0.0174	−0.0307
BRISK random forest, ϕ_i^{rf}	0.1165	−0.0232

TABLE I: The table shows the influence of the predictor variables in our regression model. As all predictors are normalized, large absolute values of coefficients imply stronger impact of the corresponding predictor. The change in the coefficient of determination (R^2) when excluding a specific predictor, provides an idea about the unique contribution of that predictor to the entire regression model. Thus, the larger the drop, the more unique the predictor's contribution is.

B. Training dataset

The proposed method is a data-driven approach, and therefore we need a training set which consists of multiple mapping sessions. We have used 31 maps recorded by the Turtlebot, aligned them together, identified commonly observed landmarks (using [16]), and jointly refined the maps using our visual-inertial least-squares optimization. Eventually, the landmarks in the dataset could be automatically labeled using the empirical probability measure described in Section III.

Furthermore, the datasets were split into two pieces: a training part and an evaluation part. The sets of landmarks used for training and evaluation are therefore completely disjoint, but they still come from a similar environment (illustrated in Fig. 6). In this way, we can test how the proposed method generalizes e.g. within a single building.

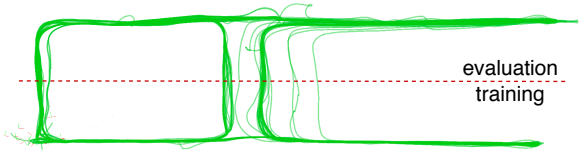

Fig. 6: Split of the collected maps between the part used for fitting the regression parameters and for all the evaluations. The complete dataset consists of 31 trajectories, each about 150 m long.

C. Regression model

First, we examine the relationship between each feature and the landmark observation count, as depicted in Fig. 7. It can be seen that all the predictor variables are correlated with the landmark labeling and only the z-coordinate of the landmark exposes a strong non-monotonic relationship. This observation was confirmed by the cross-validation procedure of the Lasso regression, where only the coefficient for the landmark's z-coordinate variable was set to zero (at $\lambda = 4.249 \cdot 10^{-4}$).

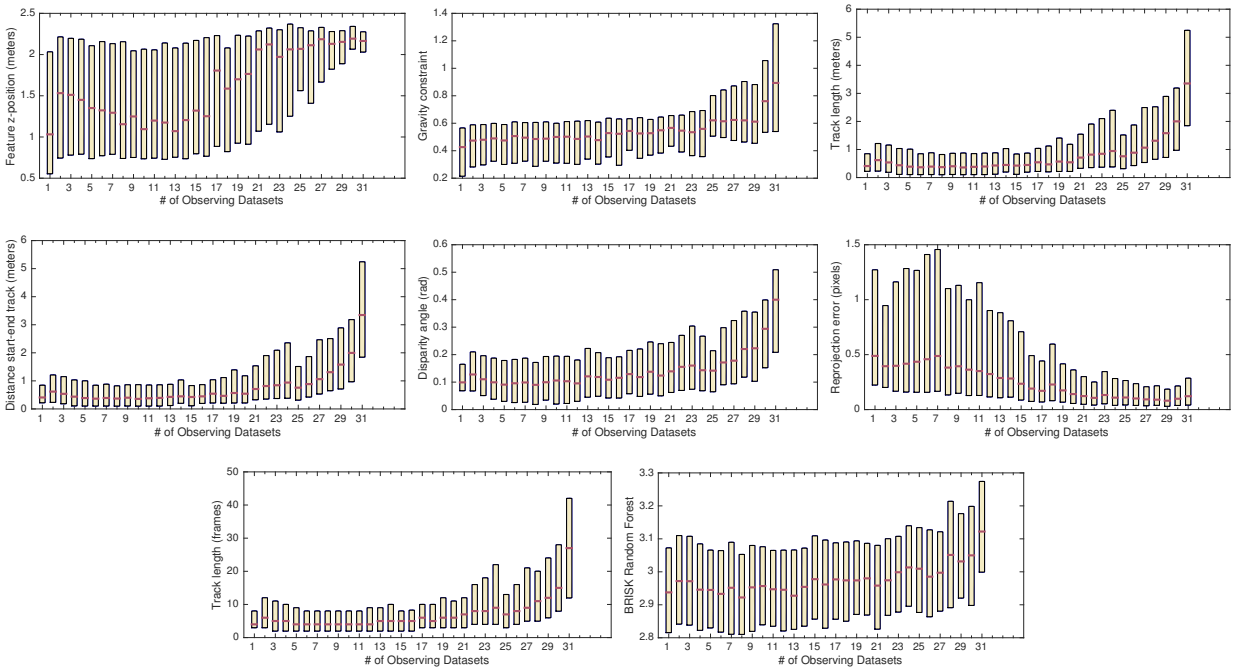

Fig. 7: The relationship between the proposed predictor variables and the the training set labeling. Yellow bars represent the interquartile range and the violet marker is the median value. The only predictor variable that is showing a strong non-monotonic relationship with the labeling is the z-coordinate of the landmark position. We therefore propose to use a non-linear transformation of this predictor before fitting the regression coefficients.

Instead of just removing this predictor variable, we decided to add a non-linear transformation that will make the relationship between the landmark label and its z-coordinate monotonic. We used $\phi_i^h = \max(p_{GLi,z}, \theta_z)$, with $\theta_z = 1.5$ m. While this might be perceived as a strongly handcrafted feature, it actually has an interpretation: The landmarks located around $h = 1.5$ m often belong to dynamic objects, which is rarely true for the landmarks close to the ceiling. After applying the transformation, the Lasso regression did not reject any of the predictors and we were able to obtain the model coefficients using Ridge regression.

The Ridge regression algorithm was able to fit to the training model and reported a coefficient of determination $R^2 = 0.1265$, meaning that over 12% of the variance is explained by the predictor variables. While this result may seem low, we need to take into account that we are predicting a very complex phenomenon using only weakly correlated variables. Table I, displays the values of each coefficient for the normalized predictors. We also include the change in the R^2 value after excluding each of the predictor variables – the larger the change is, the more information was brought by the predictor over all the other ones. We can notice that excluding the track length-related predictors is not causing a large change in R^2, as there is always another measure that captures similar properties. On the other hand, removing the Random Forest prediction or the gravity constraint reduces the R^2 significantly, suggesting these features provide an orthogonal source of information.

Finally, we also present the regression result against the evaluation set labeling in Fig. 8. The results suggest that we should be able to filter out the best landmarks with good precision (the first quartile value of the 31st class is larger than the third quartile value of the first 15 classes).

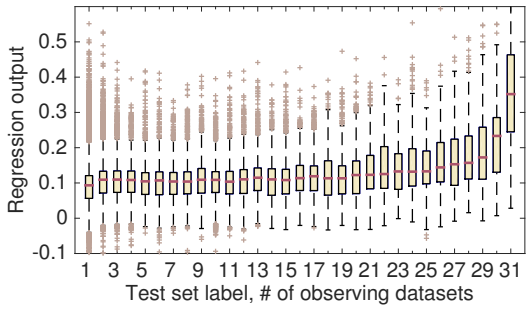

Fig. 8: Regression results over the evaluation dataset. Yellow bars represent the interquartile range, the violet line is the median value, and beige crosses denote outliers. The regression model output, which corresponds to the predicted landmark quality, shows a significant correlation with the evaluation set labeling. Moreover, the distributions of the very good and very bad landmark score are separable – the Q1 value of the best landmarks is larger than the Q3 value of the worst.

D. Comparison with other selection methods

The regression output can be used to rank the landmarks of the map and select a suitable subset for localization. We furthermore want to analyze how the place recognition performance is affected by the map reduction using the following approaches: the proposed method, random selection and the method based solely on the track length (which was used in [8]). The results of this evaluation are presented in Fig. 9. While all three methods perform similarly up to a 60% reduction, the differences are significant at reduction ratios of 80% and higher. Our proposed method outperforms both the random selection and the track length based methods, having an edge in the F1-score values of 10% for 90%

landmark reduction and 30% for 98% landmark reduction when compared to the latter.

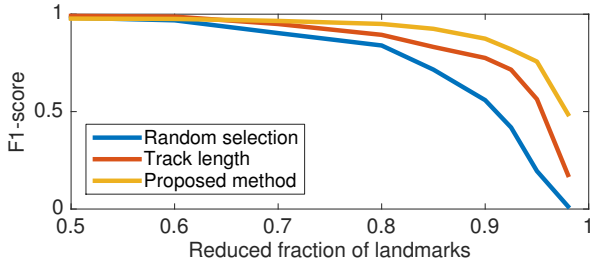

Fig. 9: Comparison of place recognition performance based on a map from a single session. The landmarks of the map were reduced using different methods and up to varying degrees of compression. We use the F1-score to measure the performance, which is a typical data retrieval measure combining precision and recall.

E. Iterative mapping and summarization

A backend designed for long-term mapping needs to incorporate new data from agents over time and iteratively re-summarize the global map ensuring its size to stay bounded. In this context, it is interesting to investigate an optimal ratio between agent-side and backend-side summarization. In other words, we seek the optimal trade-off between minimal data upload to the backend while still providing enough information to create accurate localization maps of the environment.

For this reason, we have built a series of maps with varying agent-side and backend-side summarization levels from a single dataset. A second dataset, recorded in the same environment, is used to localize against the built maps. An evaluation of the F1-score against the "fraction of retained landmarks", as shown in Fig. 10, indicates that an agent-side data reduction of up to 50% is feasible while only marginally affecting the place recognition recall. Even a reduction of 80%, which vastly reduces the required transmission bandwidth, only causes a drop of 5-10% in the F1-score, which may still be acceptable in many applications.

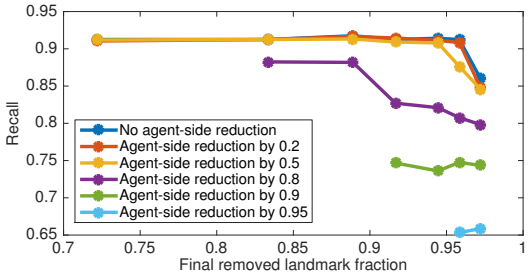

Fig. 10: Place recognition recall values for different levels of summarization on the agent and in the backend. We can see that the agent-side summarization causes almost no recall drop when reducing the map by 50%. Even a more aggressive reduction, by 80%, causes a mere drop of 5% of the recall. This indicates that agent-side summarization can significantly reduce data uploads to the backend while maintaining similar place recognition quality.

One fundamental question still remains unanswered. Can we incrementally build and localize from a map that was built by repeatedly merging in new agent data followed by a resummarization? To investigate this, we first apply agent-side summarization to a local agent map, then merge it into the existing global map of the backend. Next, the map is resummarized in the backend. This process is repeated for a set of 30 datasets as shown in Fig. 1.

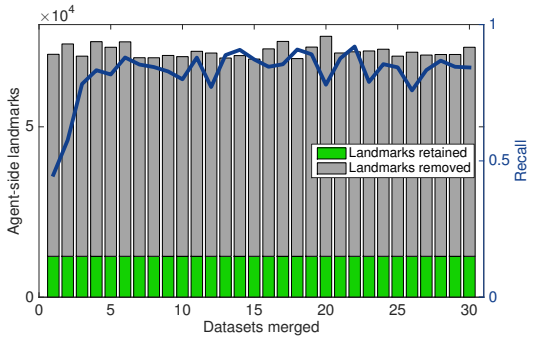

Fig. 11: The proposed agent-side summarization combined with backend summarization yields good place recognition performance at reduced data transfer. The map on the agent-side is reduced to 12,000 landmarks (about 20%) after each session and then used to build the global map in the backend, limited to 6,000 landmarks. Initially, the *experience* of the mapping sessions is summarized and the recall results are improving to reach the 75-90% range and stabilize.

Again, we evaluate the recall by localizing each mission against the current global map before merging it in. The results are presented in Fig. 11. It can be seen that the recall values stabilize within the region of 75-90% after gathering a sufficient set of stable landmarks. With this experiment, we validate that agent-side summarization can significantly reduce transferred data while still maintaining good localization performance.

V. Conclusions

In this paper we have presented an algorithm that selects a subset of landmarks which are more likely to be consistently redetected during subsequent localization attempts. The landmark selection procedure is based solely on locally available map data, rather than requiring information gained only after several visits to an area. By relying on this reduced subset of landmarks, the amount of data transferred between agents and the map backend can be throttled without corrupting localization. The proposed method scores landmarks by fusing a combination of novel and existing predictor variables using coefficients provided by a regression framework. The approach was evaluated in a long-term, iterative mapping scenario, using data collected by our autonomous office mapping platform. We show that we can drastically reduce the data transfers (by about 80% for agent to backend) while maintaining comparable localization results. In addition, we show that our method performs well in an iterative mapping process, leveraging long-term experience, and integrating easily with the backend summarization methods of [9]. This data-driven approach brings a benefit over hand-crafted variable fusion and also allows us to learn which variables are the most informative. In our experiments, we show that the newly introduced gravity constraint feature and the maximum angle between keyframe-landmark rays, as well as the proposed binary descriptor classification have the largest influence. While some of the proposed predictor variables

may be environment specific, the presented methodology is generalizable to different environments.

In future work we would like to identify additional landmark quality predictors and deploy the system in a large-scale, multi-agent mapping application. We also believe that modern machine learning techniques such as convolutional neural networks employed directly on the image data, may provide complementary predictors to the currently proposed features. We can furthermore imagine learning and maintaining a set of place-specific regression coefficients.

VI. Acknowledgments

We would like to thank Mathias Gehrig for the preparation of the Turtlebot, our autonomous mapping platform. The research leading to these results has received funding from Google's project Tango.

References

[1] S. Middelberg, T. Sattler, O. Untzelmann, and L. Kobbelt, "Scalable 6-dof localization on mobile devices," in *European Conf. on Computer Vision*, 2014.

[2] T. Cieslewski, S. Lynen, M. Dymczyk, S. Magnenat, and R. Siegwart, "Map API - scalable decentralized map building for robots," in *IEEE Int. Conf. on Robotics and Automation*, 2015.

[3] S. Lynen, T. Sattler, M. Bosse, J. Hesch, M. Pollefeys, and R. Siegwart, "Get out of my lab: Large-scale, real-time visual-inertial localization," in *Robotics: Science and Systems*, 2015.

[4] L. Riazuelo, J. Civera, and J. Montiel, "C2tam: A cloud framework for cooperative tracking and mapping," *Robotics and Autonomous Systems*, 2013.

[5] C. Forster, S. Lynen, L. Kneip, and D. Scaramuzza, "Collaborative monocular slam with multiple micro aerial vehicles," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2013.

[6] C. Arth, M. Klopschitz, G. Reitmayr, and D. Schmalstieg, "Real-time self-localization from panoramic images on mobile devices," in *IEEE Int. Symp. On Mixed and Augmented Reality*, 2011.

[7] P. Mühlfellner, M. Bürki, M. Bosse, W. Derendarz, R. Philippsen, and P. Furgale, "Summary maps for lifelong visual localization," *Journal of Field Robotics*, 2015.

[8] M. Dymczyk, S. Lynen, T. Cieslewski, M. Bosse, R. Siegwart, and P. Furgale, "The gist of maps-summarizing experience for lifelong localization," in *IEEE Int. Conf. on Robotics and Automation*, 2015.

[9] M. Dymczyk, S. Lynen, M. Bosse, and R. Siegwart, "Keep it brief: Scalable creation of compressed localization maps," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2015.

[10] W. Hartmann, M. Havlena, and K. Schindler, "Predicting matchability," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2014.

[11] S. Buoncompagni, D. Maio, D. Maltoni, and S. Papi, "Saliency-based keypoint selection for fast object detection and matching," *Pattern Recognition Letters*, vol. 62, pp. 32–40, 2015.

[12] M. J. Milford and G. F. Wyeth, "Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights," in *IEEE Int. Conf. on Robotics and Automation*, 2012.

[13] N. Sünderhauf, S. Shirazi, A. Jacobson, E. Pepperell, F. Dayoub, B. Upcroft, and M. Milford, "Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free," in *Robotics: Science and Systems*, 2015.

[14] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, 2012.

[15] T. Sattler, B. Leibe, and L. Kobbelt, "Fast image-based localization using direct 2d-to-3d matching," in *Int. Conf. on Computer Vision*, 2011.

[16] S. Lynen, M. Bosse, P. Furgale, and R. Siegwart, "Placeless place-recognition," in *3DV*, 2014.

[17] G. Carneiro and A. D. Jepson, "The quantitative characterization of the distinctiveness and robustness of local image descriptors," *Image and Vision Computing*, vol. 27, no. 8, pp. 1143–1156, 2009.

[18] Y. Verdie, K. M. Yi, P. Fua, and V. Lepetit, "TILDE: A temporally invariant learned detector," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2015.

[19] W. Zhang and J. Košecká, "Hierarchical building recognition," *Image and Vision Computing*, vol. 25, no. 5, pp. 704–716, 2007.

[20] J. Knopp, J. Sivic, and T. Pajdla, "Avoiding confusing features in place recognition," in *European Conf. on Computer Vision*. Springer, 2010.

[21] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Int. Conf. on Computer Vision*, 2011.

[22] A. Walcott-Bryant, M. Kaess, H. Johannsson, and J. J. Leonard, "Dynamic pose graph slam: Long-term mapping in low dynamic environments," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2012.

[23] W. Churchill and P. Newman, "Experience-based Navigation for Long-term Localisation," *The International Journal of Robotics Research (IJRR)*, 2013.

[24] H. S. Park, Y. Wang, E. Nurvitadhi, J. C. Hoe, Y. Sheikh, and M. Chen, "3d point cloud reduction using mixed-integer quadratic programming," in *IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, 2013.

[25] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, "The office marathon: Robust navigation in an indoor office environment," in *IEEE Int. Conf. on Robotics and Automation*, 2010.

[26] J. Nikolic, J. Rehder, M. Burri, P. Gohl, S. Leutenegger, P. T. Furgale, and R. Siegwart, "A synchronized visual-inertial sensor system with fpga pre-processing for accurate real-time slam," in *IEEE Int. Conf. on Robotics and Automation*, 2014.