Map Quality Evaluation for Visual Localization

Hamza Merzić, Elena Stumm, Marcin Dymczyk, Roland Siegwart, and Igor Gilitschenski

***Abstract*— A variety of end-user devices involving keypoint-based mapping systems are about to hit the market e.g. as part of smartphones, cars, robotic platforms, or virtual and augmented reality applications. Thus, the generated map data requires automated evaluation procedures that do not require experienced personnel or ground truth knowledge of the underlying environment. A particularly important question enabling commercial applications is whether a given map is of sufficient quality for localization.**

This paper proposes a framework for predicting localization performance in the context of visual landmark-based mapping. Specifically, we propose an algorithm for predicting performance of vision-based localization systems from different poses within the map. To achieve this, a metric is defined that assigns a score to a given query pose based on the underlying map structure. The algorithm is evaluated on two challenging datasets involving indoor data generated using a handheld device and outdoor data from an autonomous fixed-wing unmanned aerial vehicle (UAV). Using these, we are able to show that the score provided by our method is highly correlated to the true localization performance. Furthermore, we demonstrate how the predicted map quality can be used within a belief based path planning framework in order to provide reliable trajectories through high-quality areas of the map.

I. INTRODUCTION

Understanding a map's structure and evaluating map quality is useful for a wide range of applications, particularly for applications where constant localization or feature tracking is crucial. Emerging technologies such as Augmented Reality (AR) and automated mapping frameworks rely heavily on accurate localization in order to properly overlay 3D objects and consistently reconstruct the environment, respectively.

As a result, the goal of this work is to provide a simple and general framework for predicting localization performance for poses within the map. Using this framework we aim to exploit the map's structure by providing a means of discriminating between places where localization will succeed or fail. Such information can then allow higher level applications to incorporate the likelihood of robust localization in tasks such as navigation and decision making.

However, the key challenge is defining the map quality metric and its representation. One way to look at map quality is through the accuracy of representation of the mapped area. Creating reliable maps in this sense is accomplished using approaches such as optimization over entropy [1] or mutual information [2]. On the other hand, map quality can be viewed as the information quality of parts of the map's structure, where information-rich parts should be easier to localize against [3]. Using this approach to discriminate between the good and bad parts of the map depends to a large extent on the observer's pose, since different views of the same landmark can produce entirely different visual descriptors. Therefore, in the context of this paper, we shall refer to map quality as the localization performance of poses within the map, simply called query poses. In this setting, high scoring parts of the map represent poses from which localization is expected to succeed.

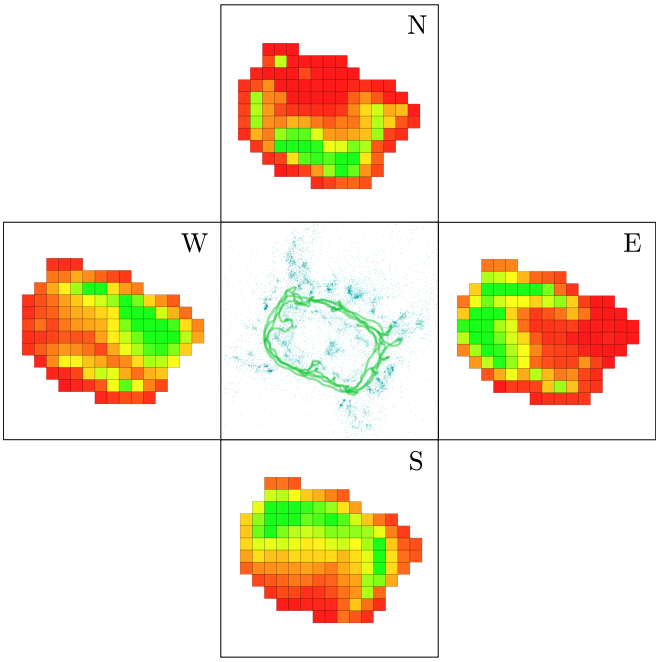


Fig. 1: An illustration of the predicted map quality using the proposed algorithm. The landmarks and trajectory from the underlying map are shown in the middle frame, with each surrounding frame representing the expected localization quality throughout the environment, based on the corresponding pose orientation. Green cells indicate a high likelihood of localization from the given pose, while red indicates a low score.

The presented notion of map quality is highly relevant for various applications, particularly in the following areas. Firstly, active localization and active SLAM applications rely on revisiting well mapped areas for detecting and closing loops, i.e. recognizing a previously visited place and using the new information to correct for accumulated errors [4]. Secondly, knowing map quality plays an important role for multi-agent mapping, since it enables more efficient map

Authors are from the Autonomous Systems Lab, ETH Zurich, Zurich, Switzerland. hmerzic@student.ethz.ch, estumm@ethz.ch, marcin.dymczyk@mavt.ethz.ch, rsiegwart@ethz.ch, igilitschenski@ethz.ch

construction and could help save computational resources by avoiding well mapped areas during mapping while focusing on poorly mapped ones instead. Thirdly, map quality can be used in path planning and navigation in order to enforce paths to follow well mapped areas. Lastly, the map's structure can be sparsified while maintaining a desired quality, hence enabling better map compression [5].

The algorithm that we propose is simple, computationally inexpensive, and general within visual landmark-based mapping frameworks. In order to obtain a map quality score, we split the algorithm into two steps. In the first step we use the neighborhood of the query pose to estimate which landmarks will be observed from it. We subsequently calculate each landmarks' contribution to the final score based on the location of the query pose, as well as information about the mapped poses which have observed the landmark.

Overall, this paper includes the following contributions:

- An efficient algorithm for evaluating map quality.
- A way of visualizing map quality.
- Evaluations performed on both indoor and outdoor environments through the use of ground and flying agents.
- A demonstration of the algorithm applied to generating beliefs for path planning under uncertainty.

II. RELATED WORK

Active mapping and localization scenarios rely on map quality in order to construct maps which ensure the success of future navigation queries, e.g. finding paths within the map. A certain notion of map quality is used in order to satisfy the navigation requirements. A common strategy for active mapping is to focus on parts of the map with high expected information gain. For this purpose Davison et al. [6] used a moving camera head, controlled to reduce expected future measurement uncertainty. Fairfield et al. [7] demonstrated a framework for heuristically estimating the entropy of the map, which they use as a measure of map quality thereafter.

Applications in robot navigation different notions of map quality in order to represent the belief space for path planning. Each point in this space corresponds to the expected uncertainty of a robot's state. Navigation queries are performed with the goal of minimizing the average state uncertainty or the uncertainty at the goal state. Valencia et al. [8] extend the pose SLAM algorithm to directly construct the belief space for path planning. A drawback to the presented approach is that there is no extrapolation in the belief space, i.e. only the visited nodes are used for planning. Bry et al. [9] adapt the RRT* planning algorithm, in order to incorporate belief space information in the planning framework. They represent the belief space as uncertainty covariances assigned to each state. For the unvisited states, they use predefined covariances based on the expected measurements. Chaves et al. [10] go a step further and include stochastic measurement acquisition variables that aim to model the probability of taking certain measurements. They use this to define a risk measure associated with a low probability of taking a measurement and incorporate that into their planner. They demonstrate their algorithm on a visual mapping application, based on the work of Kim et al. [11], which uses visual saliency to generate the belief space. The drawback with this method is limited extrapolation performance, since images are required in order to calculate visual saliency. Costante et al. [3] overcome these issues by using an information based approach for defining the belief space. The map quality measure of their approach is encoded in the covariance obtained from the dense image-to-model alignment. By synthesizing images from the environment model, they are able to extrapolate the uncertainty to unvisited states. However, as the focus of the work is not on global localization, preference is given only to feature rich areas, with no notion of their suitability for relocalization (i.e. redetection and matching).

Dequaire et al. [12] use a different notion of map quality. Through the use of Gaussian Processes, they predict localization performance in a *teach and repeat* [13] framework. This framework consists of two phases. The *teach* phase, where the robot is operated through the desired path. During this phase the robot builds a map of the world using a similar procedure to the one used in our experiments. The goal of *repeat* phases is to track the previously taught path. Since a change in the environment might greatly influence the localization performance, even a slight deviation from the initial path might cause a loss of localization. This is an important issue, since loss of localization can lead to controller instability. Dequaire et al. thus designed a framework which reliably estimates the localization envelope, which gives an estimation of map quality around previously visited paths. However, this approach fails to extrapolate orientation information, and must be computed offline.

Our work provides a framework for evaluating map quality, which is general in the sense that localization quality can be extrapolated to previously unvisited poses. In addition, the approach incorporates 6DOF pose information in the assessment of map quality. Aspects such as redetectability of landmarks or landmark covisibility, which are often left out of other work, are implicitly included in our algorithm. Furthermore, the algorithm is simple in the sense that it allows for a sparse map representation, and relies only on the underlying map structure.

III. MAP QUALITY EVALUATION

For the proposed map quality evaluation method, we assume a keypoint based visual map as generated by state-of-the-art SLAM systems during potentially multiple runs through the same environment. Localization is performed using 2D-3D matching [14]. Our proposed method assesses map quality by predicting whether localization is likely to be successful from given 6DOF query poses. As no prior observation typically exists from these query poses, we must estimate which landmarks will be visible from each pose. Therefore, we first compute the landmarks that have been visible from nearby poses; and subsequently compute which of these landmarks are also expected to be visible from the

query pose itself. Finally, this information is used to score the localization quality of the query pose.

A. Retrieving Relevant Landmarks

In the first step, we compute a candidate list $\mathcal{L}_v$, consisting of landmarks that have been observed from neighboring poses and a landmark specific weight. This is achieved by considering the query pose, retrieving its K nearest neighbor poses from previous mapping traversals and the landmarks observed by them. Assigning weights to landmarks based on a metric of usefulness or quality has been a topic of extensive previous work. These landmark selection policies have generally been information theoretic [15] incorporating both location and feature uncertainty. But, feature uncertainty is not necessarily a good measure of redetectability, which is our main focus. By considering landmarks with repetition, we implicitly include redetectability information and increase the prior for landmarks observed from multiple nearest neighbors. For this retrieval task, we define nearest neighbors not merely by the proximity of observer positions, but also incorporate their orientation by using the metric presented in [16] where a design parameter controls the trade-off between emphasizing position or orientation. Although this parameter is mission-specific, a certain robustness in its choice was observed in practice. For all of the evaluations presented, the same metric emphasizing orientation proximity, was used. This way of defining the distance avoids selecting observing poses that point in an entirely different direction.

The weight of each landmark represents the number of observers within the K neighbors. For better robustness, we extend the combinatorial landmark selection policy used in [14] by also removing landmarks that were rarely observed (in our case less than 6 observations from distinct poses within the map). The resulting method is depicted in Algorithm 1. It takes the existing map $\mathfrak{M}$ and the query pose v_q as parameters. There, the functions GetNearestNeighbours, LandmarksObservedFromPose, UpdateLandmarkList and RemoveRarelyObservedLandmarks implement the functionality described above.

Algorithm 1 GetLandmarkCandidateList($v_q, \mathfrak{M}$)

// Initialize empty list of landmarks.
$\mathcal{L}_v \leftarrow \emptyset$
// Find nearest neighbors.
$nn\_poses \leftarrow$ GetNearestNeighbors($v_q, \mathfrak{M}$)
for $p \in nn\_poses$ **do**
  $l\_observed \leftarrow$ LandmarksObservedFromPose($p, \mathfrak{M}$)
  // Increment weights for landmarks in $\mathcal{L}_v$.
  $\mathcal{L}_v \leftarrow$ UpdateLandmarkList($l\_observed, \mathcal{L}_v$)
end for
$\mathcal{L}_v \leftarrow$ RemoveRarelyObservedLandmarks($\mathcal{L}_v, \mathfrak{M}$)
return $\mathcal{L}_v$

B. Predicting Visible Landmarks

For each landmark in the candidate list we predict its visibility from the query pose. This is carried out by constructing a convex hull over all positions from which a given landmark was observed, inflating the convex hull, and then checking whether the query pose lies within the resulting polygon. Convex hull construction is visualized in Figure 4. Use of convex hulls is based on the assumption that if a landmark is observed from a number of given positions, it can also be observed from within the convex hull of these positions. For most cases this holds well, and since the algorithm essentially defines well-localizable poses to be inside many hulls, our approach shows satisfying robustness to this assumption.

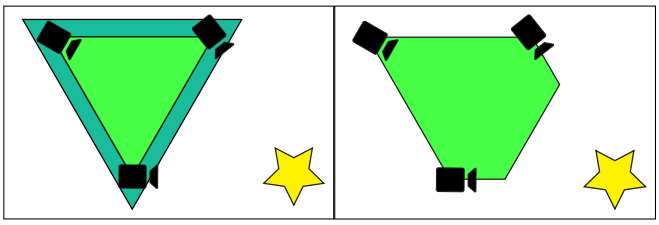


Fig. 2: The left image shows a regular (light green) and inflated (dark green addition) convex hull. The right image shows the convex hull extended toward a landmark.

In this work, we assume that the variance of local movements in the z-axis remains small. In most of the single-agent applications, this is a reasonable assumption since the height of the mapping agent, either a person or a robot, typically remains constant throughout the mission. This allows us to simplify the implementation by projecting all poses into the x, y-plane and using 2D convex hulls.

In order to obtain better results, convex hulls are slightly inflated. The convex hull is extended by creating a copy of all observer poses and placing them even closer to the landmark and then generating a joint convex hull from both the old and newly created poses. In the implementation, this is carried out before the actual convex hull computation. This inflation is motivated by the assumption that if we can detect a landmark from a pose, then we should also be able to detect it by moving a bit closer. The resulting polygon is then subsequently inflated in all directions, increasing its area by a predefined share, which is specific to the platform and the typical operational environment. These operations play an important role in some highly constrained environments such as corridors, where the ordinary convex hull ends up being almost a mere line, and extrapolation becomes impossible. The convex hull inflation process is visualized in Figure 2.

C. Pose scoring

The entire resulting map quality procedure is visualized in Algorithm 2. Iterating through the list of candidate landmarks, the function ConvexHull computes the inflated convex hull according to the methodology described in the previous subsection, whereas the function Inside simply checks if v_q lies inside the inflated convex hull. Note that each convex hull is weighted according to the corresponding landmark's observers (as described in section III-A). Essentially, the more strongly weighted convex hulls a query pose is contained within, the higher the score it will receive, as this

indicates that more landmarks are likely to be observable. Lastly, we use the Normalize function to map the convex hull counter to a specific range or set, e.g. $[0, 5]$, $\{0, 1\}$, etc. We use the obtained score to represent our localization estimate.

Algorithm 2 EvaluateMapQuality($v_q, \mathfrak{M}$)

```
// Initialize convex hull counter.
score ← 0
L_v ← GetLandmarkCandidateList(v_q, M)
for (l, w) ∈ L_v do
    observers ← PosesObservingLandmark(l)
    C_l ← ConvexHull(observers)
    if Inside(v_q, C_l) then
        score ← score + w
    end if
end for
Normalize(score, M)
return score
```

Our implementation of the Normalize function remaps the predicted number of visible landmarks to the range $[-1, 1]$ as can be seen in the x-axes of figures 3 and 5. The scaling is done using a platform specific parameter which defines the crossover score, i.e. the minimum score for which we expect successful localization. The obtained score is then limited from above by 2 times the crossover score, and linearly scaled to $[-1, 1]$, so that -1 corresponds to score of 0, and 0 then corresponds to the crossover score. Thus, in terms of binary prediction, any score below 0 is expected to result in a failed localization attempt, while any score above is expected to succeed. This scaling allows the scores to fit a unified interface for higher level applications which rely on map quality, e.g. path planning algorithms. Selecting the crossover parameter is performed using cross-validation on the aligned map. Alternatively, the list of all potentially visible landmarks, for further processing and higher-level decision making, can be obtained.

The computational complexity of the entire procedure is highly dependent on the underlying implementation of the map data structure. In most cases however, the driving factors behind computational cost are the nearest neighbor search when selecting relevant landmark candidates, along with convex hull construction for each of the landmarks. Using k-d trees, this search can be implemented in $\mathcal{O}(\log n)$ in the number of candidate poses [17]. Convex hull construction performance is extensively studied in IV-A.

IV. Empirical Results

In what follows, we first present benchmarks for convex hull construction and discuss how the algorithm can be adapted for online use (IV-A). We proceed by presenting evaluations performed on indoor ground data and on outdoor flight data (IV-B). After subsequently discussing map quality visualization (IV-C), we demonstrate the use of the algorithm for belief based path planning (IV-D).

In order to construct the maps, the procedure presented in [14] was used. To demonstrate the generality of our algorithm, we used different hardware setups and landmark descriptors in each of the presented scenarios. In general, the hardware used is composed of a visual and an inertial sensor, i.e. a camera and an IMU, which enable localization without the need of GPS or WiFi. Additionally, RANSAC was used for outlier rejection [18]. Maps have been optimized and aligned offline, using nonlinear least squares optimization [19].

A. Convex hull construction

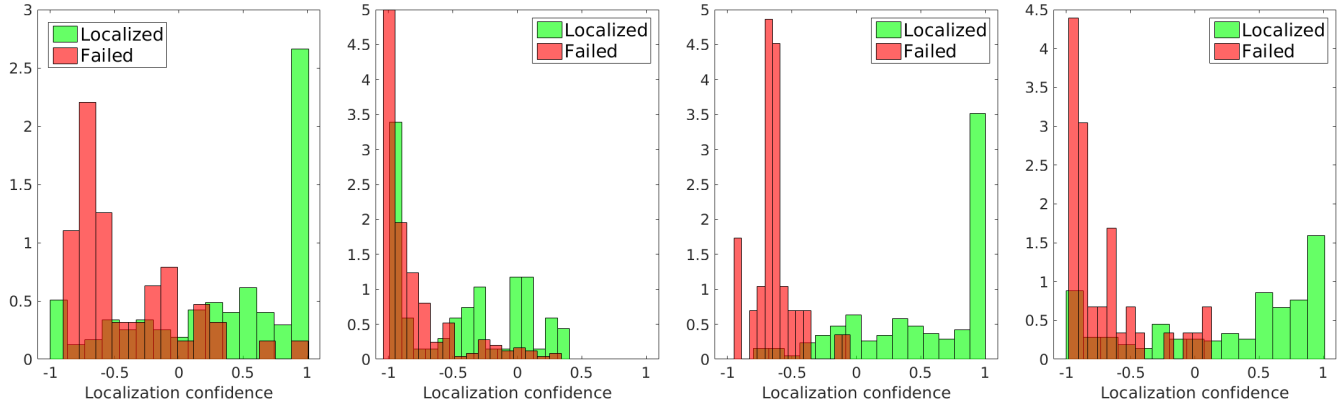
As previously mentioned, convex hull construction presents one of the most computationally expensive parts of the algorithm, along with the nearest neighbor search. For that reason we additionally perform an evaluation of the convex hull construction process. The map being considered contains 76144 visual landmarks and 4125 camera poses, and convex hull creation uses the popular Graham scan algorithm [20]. Results are shown in Table I. Rows represent ordinary convex hulls, convex hulls extended toward the landmark, and convex hulls both extended and inflated, respectively. The columns represents the total time in seconds needed to construct all 76144 convex hulls, the average time in microseconds per landmark, the average number of points used to construct a hull and the average number of points in a hull, respectively. The difference in time between the methods is negligible, even when the convex hulls differ substantially in size.

TABLE I: Convex hull construction benchmark.

CH	Tot $t[s]$	Avg $t[\mu s]$	Avg $\#v$	Avg CH $\#v$
Normal	1.408	18.495	9.17	3.509
Extended	1.439	18.898	18.34	4.837
+ Inflated	1.449	19.032	18.34	4.837

The benchmark has been performed on a single thread of a 8 GB RAM and Intel Core i7 quad @2GHz per core machine running Linux Ubuntu 14.04 LTS.

In an online application of the algorithm, we are able to evaluate map quality at e.g. a dense grid sampling of 6DOF poses throughout the mapped area (see Figure 1 for example). In this case, after adding each new measurement, we begin by generating convex hulls of the observed landmarks. Then, we iterate through potential grid cells, and check if the new camera pose is closer than some of the corresponding nearest poses. If that is the case, we update the grid cell with the new camera pose and its score contribution. Efficient implementations of nearest neighbor search [17] and point in polygon check [21] provide negligible complexity compared to convex hull construction. Therefore, assuming that on average from every new camera pose 100 landmarks can be observed, the entire update step takes around 2 ms, according to Table I. It is important to note that even better performance could be obtained by using a more efficient implementation for convex hull construction, caching the created hulls, or using multithreading.



(a) Regular run. (b) Intentionally difficult run. (c) Intentionally good run. (d) Regular run.

Fig. 3: Evaluation of the map quality prediction on the indoor ground data. Two distributions can be observed, which correspond to successful and failed localizations, i.e. green and red points, respectively. The data is normalized, so that the distributions can be observed. For all positive points, successful localization is predicted.

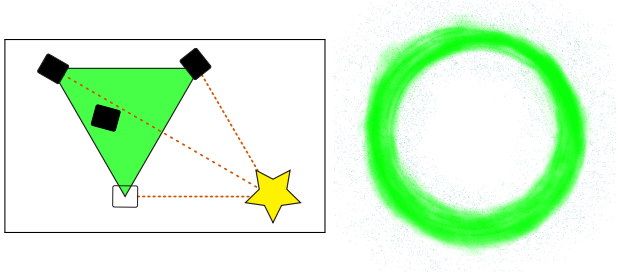


Fig. 4: The left image shows an example of convex hull construction, while the image to the right shows overlayed convex hulls from all the landmarks within a flight map.

B. Map quality evaluation

We carry out two experiments in order to validate our algorithm; one performed on indoor data recorded by a ground agent and another one on outdoor data obtained from an UAV. For each of the experiments, a detailed map consisting of multiple passes through the same area was recorded to serve as the test map. This map was thoroughly optimized, such that it can be assumed that it accurately represents the mapped area. Apart from this test map, a number of query trajectories were recorded in order to evaluate the localization predictions.

For each of the query trajectories, a localization histogram over map quality scores, as in Figure 3, was created. The data is normalized, such that it is easier to examine the distributions. For positive values in x we predict successful localization, while for negative we predict failure. Points in the bins correspond to actual localization outcomes. Green points indicate success and red points indicate failure in localization.

The evaluation metrics *specificity* and *accuracy* are used to investigate the prediction performance. *Specificity* represents the ratio of true negatives (negative red points), and the sum of true negatives and false positives (all red points). By maximizing this score, we ensure that we correctly predict as many failed localizations as possible. This measure is meaningful since we give preference to maintaining localization at all times, and therefore favor false negatives over false positives. In order to balance our predictions, we also introduce an accuracy score. *Accuracy* simply measures the ratio of correct predictions.

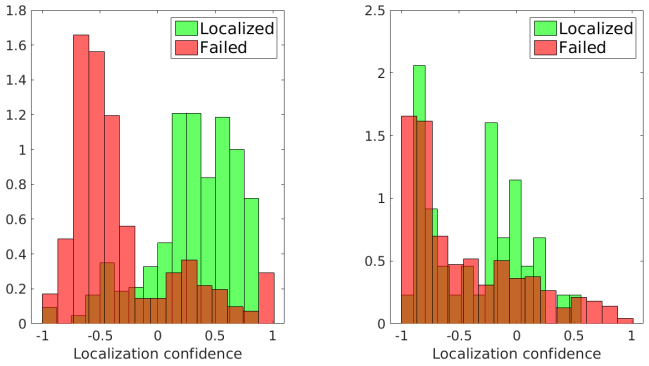
In order to run the algorithm, parameters mentioned in section III have to be defined. Constant values for number of nearest neighbors (10), orientation weight for distance metric (5.0), and ratio of extension toward the landmark (0.125) have been used across all evaluations. The convex hull inflation ratio and the crossover score were different across different evaluations, but constant within the same setups.

1) Indoor ground data: For the indoor ground data evaluation we constructed a detailed map of a single room of the size $5\,\text{m} \times 3\,\text{m}$, along with four different query trajectories. Data was acquired using Google's Tango visual-inertial odometry estimator, which uses FREAK [22] descriptors for landmark representation. For every point in a single query trajectory, we compared our expected map quality to the actual localization result. A localization attempt is considered to be successful if the resulting position error of the estimate is less than 30 cm and the crossover score of 750 was used for localization predictions. Histograms for all four evaluations are shown in Figure 3, and the corresponding specificity and accuracy scores are presented in Table II. We divide the query trajectories into three categories: intentionally difficult, regular, and intentionally good runs. These different categories are used to evaluate the map quality prediction in varying conditions. For the intentionally difficult run, an effort was made to fail to localize. This was achieved through larger movements and changing camera heights. The opposite was done for the intentionally good runs. Regular runs represent trajectories where the motion attempted to simulate standard handheld use-cases.

TABLE II: Indoor ground mission results.

	a) Regular	b) Difficult	c) Good	d) Regular
Specificity	0.84	0.97794	1	0.925
Accuracy	0.74194	0.82609	0.80165	0.68362

The results indicate good performance of our prediction. In the histograms, we can observe two distributions for localized and failed points, showing a clear discrimination between the two in most of the runs. The specificity and accuracy metrics displayed in Table II also indicate good performance. In our experiments, convex hull inflation did not show a significant improvement to the overall results.



(a) Regular run. (b) Difficult run.

Fig. 5: Evaluation of the map quality prediction on the outdoor flight data.

2) Outdoor flight data: In the case of the outdoor flight evaluation, the data was acquired using a fixed wing UAV flying six loops with a 50 m radius at 100 m above ground. The six loops in the data were split into four runs which were optimized thoroughly and used as the ground truth test map, and two runs which were used for evaluation. We used a different visual inertial sensor [23], combined with BRISK [24] descriptors. A localization was considered successful if the position error was less than 10 m. In this setting successful localizations were predicted for scores above 600. The histogram of results is displayed in Figure 5, while the numerical results are shown in Table III. In Figure 5a a regular run is presented. We can again see the two distributions with clear discrimination between them. For Figure 5b we observe slightly worse results. It is easy to notice that a multitude of points which successfully localized, were assigned a low quality. This is due to the fact that the difficult run was tested for extrapolation capabilities of our algorithm, with deviations from the ground truth map exceeding 10 m. Due to lack of structure in those parts, the algorithm predicted bad quality even though the system managed to localize.

TABLE III: Flight mission results.

	a) Regular	b) Difficult
Specificity	0.78431	0.78918
Accuracy	0.80769	0.74567

We observed that convex hull inflation is particularly important in this case. This is intuitive since movements in the x, y-plane do not influence the angle of observation of the landmark on the ground by a large factor. As a result, visual descriptors will remain similar in that case, and the likelihood of redetectability increases. In this specific application a 40% inflation showed good results. Nevertheless, the results are inferior to those from ground data evaluation. The reasons for this are that the test map was obtained using fewer runs over the same area. Also, landmarks are farther away, and therefore the triangulation errors are larger which makes the localization cutoff of 10m potentially too conservative.

C. Map quality visualization

To demonstrate map quality visualization, we use the test map from IV-B.1. The end result of our algorithm is shown in Figure 1. Four different map quality grids consisting of cells of size $0.5\,\text{m} \times 0.5\,\text{m}$ are displayed. Each grid corresponds to a specific orientation of query poses. For the presented visualization we use only four different orientations labeled north, east, south and west. The color of the cell represents the quality assigned to the corresponding pose, which consists of a grid cell center position and the corresponding orientation. The green and red regions represent poses with high and low localization confidence, respectively. Yellow parts represent lack of confidence and are more prone to errors. These four map quality grids can be merged to a single grid, as shown in Figure 6.

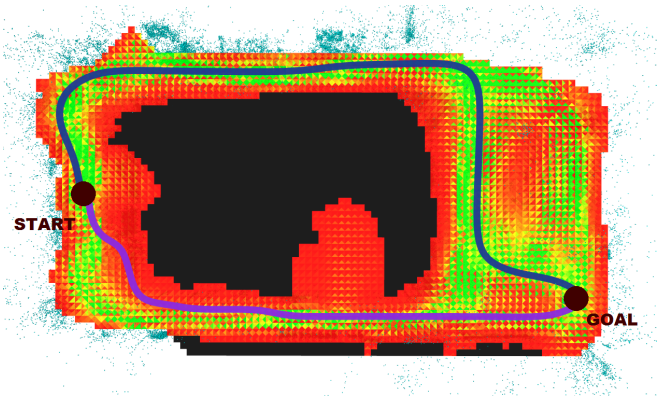


Fig. 6: Path planning under uncertainty, using map quality as the belief measure. Each cell is divided into four directions, corresponding to the quality prediction for a pose oriented in that direction. The black parts indicate unexplored areas of the map. The blue and purple paths were obtained for $\alpha = 1$ and $\alpha = 0.1$, respectively. One can see that the blue path follows a route which remains in parts of the map with a high probability of successful localization.

Adjusting the cell size allows for increasing the resolution of map quality predictions. Another example of predicted map quality can be seen in Figure 6, where the grid resolution is $0.2\,\text{m} \times 0.2\,\text{m}$.

D. Map quality as belief for path planning

In this section, we demonstrate how our algorithm can be used natively as a measure of belief for path planning under uncertainty. In a similar manner to Figure 1, the mapped area was discretized into cells of size $0.2\,\mathrm{m} \times 0.2\,\mathrm{m}$, and four poses were evaluated at each cell location (corresponding to four distinct orientations north, east, south, west). We define the transition cost from state s to s' as:

$$q(s, s') = \text{distance(s, s')} + \alpha(1 - \text{quality}(s, s')),$$

where $\text{quality}(s, s')$ is the quality of pose centered at s facing s'. This way we penalized both path length and accumulated quality from start to goal. Different paths are demonstrated in Figure 6 for $\alpha = 1$ and $\alpha = 0.1$.

We can see that in the case of higher penalty on the accumulated quality, the path changes to follow the more feature rich areas of map, where localization is more likely to succeed. Path planning is done using the popular A* algorithm, with transition cost as defined above, and Euclidean distance as the heuristic.

V. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a novel algorithm for evaluating and visualizing map quality in terms of localization performance. We have evaluated the algorithm on a variety of data and the results show satisfying performance given the assumptions. The approach generalizes well to different environments and conditions, as demonstrated by evaluation on both indoor ground data, outdoor aerial data, as well as variations in the underlying visual features and hardware setup. Furthermore, the approach is able to extrapolate localization information to previously unseen poses in the map. Results indicate good predictive performance of the algorithm, with both specificity and accuracy being above 75% for most of the evaluations.

A potential improvement to the algorithm would be to remove the assumption of low variance along the z-axis, which will be the focus of our future work. Furthermore, our future work will include refinements to the confidence measure of map quality estimates, improvements to observed landmark predictions, e.g. by leveraging geometric relations of landmarks within landmark constellations [25], and extensive evaluations on real robots.

REFERENCES

[1] W. Burgard, D. Fox, and S. Thrun, "Active mobile robot localization by entropy minimization," in *Advanced Mobile Robots, 1997. Proceedings., Second EUROMICRO workshop on*. IEEE, 1997, pp. 155–162.

[2] B. J. Julian, S. Karaman, and D. Rus, "On mutual information-based control of range sensing robots for mapping applications," *The International Journal of Robotics Research*, p. 0278364914526288, 2014.

[3] G. Costante, C. Forster, J. Delmerico, P. Valigi, and D. Scaramuzza, "Perception-aware path planning," *arXiv preprint arXiv:1605.04151*, 2016.

[4] A. Gil, O. M. Mozos, M. Ballesta, and O. Reinoso, "A comparative evaluation of interest point detectors and local descriptors for visual slam," *Machine Vision and Applications*, vol. 21, no. 6, pp. 905–920, 2010.

[5] M. Dymczyk, S. Lynen, M. Bosse, and R. Siegwart, "Keep it brief: Scalable creation of compressed localization maps," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 2536–2542.

[6] A. J. Davison and D. W. Murray, "Simultaneous localization and map-building using active vision," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 865–880, 2002.

[7] N. Fairfield and D. Wettergreen, "Active slam and loop prediction with the segmented map using simplified models," in *Field and Service Robotics*. Springer, 2010, pp. 173–182.

[8] R. Valencia, J. Andrade-Cetto, and J. M. Porta, "Path planning in belief space with pose slam," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 78–83.

[9] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 723–730.

[10] S. M. Chaves, J. M. Walls, E. Galceran, and R. M. Eustice, "Risk aversion in belief-space planning under measurement acquisition uncertainty," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 2079–2086.

[11] A. Kim and R. M. Eustice, "Real-time visual slam for autonomous underwater hull inspection using visual saliency," *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 719–733, 2013.

[12] J. Dequaire, C. H. Tong, W. Churchill, and I. Posner, "Off the beaten track: Predicting localisation performance in visual teach and repeat," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 795–800.

[13] P. Furgale and T. D. Barfoot, "Visual teach and repeat for long-range rover autonomy," *Journal of Field Robotics*, vol. 27, no. 5, pp. 534–560, 2010.

[14] 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.

[15] H. Kretzschmar and C. Stachniss, "Information-theoretic compression of pose graphs for laser-based slam," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1219–1230, 2012.

[16] R. Di Gregorio, "A novel point of view to define the distance between two rigid-body poses," in *Advances in Robot Kinematics: Analysis and Design*. Springer, 2008, pp. 361–369.

[17] J. Ichnowski and R. Alterovitz, "Fast nearest neighbor search in se (3) for sampling-based motion planning," in *Algorithmic Foundations of Robotics XI*. Springer, 2015, pp. 197–214.

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

[19] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[20] A. M. Andrew, "Another efficient algorithm for convex hulls in two dimensions," *Information Processing Letters*, vol. 9, no. 5, pp. 216–219, 1979.

[21] K. Hormann and A. Agathos, "The point in polygon problem for arbitrary polygons," *Computational Geometry*, vol. 20, no. 3, pp. 131–144, 2001.

[22] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *Computer vision and pattern recognition (CVPR), 2012 IEEE conference on*. Ieee, 2012, pp. 510–517.

[23] 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 *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 431–437.

[24] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *2011 International conference on computer vision*. IEEE, 2011, pp. 2548–2555.

[25] E. Stumm, C. Mei, and S. Lacroix, "Probabilistic place recognition with covisibility maps," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 4158–4163.