2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Ground Texture Based Localization: Do We Need to Detect Keypoints?

Jan Fabian Schmid[1,2], Stephan F. Simon[1], Rudolf Mester[2,3]

***Abstract*— Localization using ground texture images recorded with a downward-facing camera is a promising approach to achieve reliable high-accuracy vehicle positioning. A common way to accomplish the task is to focus on prominent features of the ground texture such as stones and cracks. Our results indicate that with an approximately known camera pose it is sufficient to use arbitrary ground regions, i.e. extracting features at random positions without significant loss in localization performance. Additionally, we propose a real-time capable CPU-only localization method based on this idea, and suggest possible improvements for further research.**

I. INTRODUCTION

Precise localization is a key capability of autonomous vehicles. It is necessary to follow a given path without deviation and to deliver goods to a specified position. Previous work has shown that ground texture based localization, using images from downward-facing cameras, is suitable for this task [1], [2], [3]. It is an infrastructure-free method that works in areas without landmarks, and in scenarios where the vehicle is surrounded by elements blocking its sight or radio connection. It works regardless of external lighting if the recording area is covered and illuminated artificially.

State-of-the-art ground texture based localizers estimate the camera pose using correspondences of recognized visual features [1], [2], [3], [4]. These methods have been shown to enable reliable, centimeter precise positioning [3], [4], but they induce a significant computational load.

Features represent characteristic image regions. They consist of two parts, the *keypoint object* and the *feature descriptor*. The keypoint object specifies an image patch, based on its image coordinates (the keypoint), its orientation, and scale. The image patch content is summarized by the feature descriptor. Feature-based localization can be divided into 5 subtasks: (1) *keypoint detection*, extracting similar image regions in a query image (image that is used for localization) and reference images; (2) *keypoint selection*, selecting a subset of keypoint objects for further processing; (3) *feature description*, computing descriptors that take similar values for corresponding keypoint objects; (4) *feature matching*, proposing feature correspondences; (5) *pose estimation*, estimating the query image pose w.r.t. the reference image poses.

Zhang et al. identified matching as one of the most time-consuming steps of the task, so they propose to use an approximate nearest neighbor (ANN) search index [3]. In our previous work, matching is further sped up with the identity matching approach, which matches features only if their descriptors are identical [4]. This also allows to take advantage of localization estimates (in the following called prior), which can not be done with ANN matching, because the search index is built globally for all reference images. Using this prior, our method was shown to have better localization success rates, while being faster to compute [4].

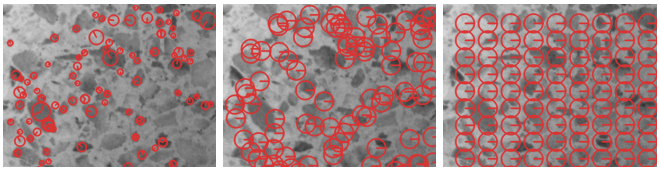

Fig. 1: Illustration of keypoint objects. Left to right: SIFT, random, uniform. Images are taken from the *tiles* dataset [3].

One remaining bottleneck, preventing the real-time applicability of ground texture based localization, is keypoint detection. Previously, we found the SIFT detector [5] to be among the best detectors for the application [6], [4]. However, it is a costly detector that dominates the computation time of our method. One approach is to use a GPU for feature extraction, but for the application in low-cost robots it is desirable to avoid the use of dedicated hardware accelerators.

We hypothesize, that for ground texture based localization with available prior, proper keypoint detection can be disregarded. Instead, keypoints can be sampled randomly, mainly because pose estimation with a downward-facing camera is with good approximation a 2D problem. The camera pose can be estimated as an Euclidean transformation of rotation and translation in two dimensions. Therefore, keypoint object scale is constant for corresponding image regions in different images; and with an available pose prior, the relative orientation to the starting pose can be used as keypoint orientation, reducing the keypoint object properties to be determined to its image coordinates.

The contribution of this paper is twofold. First, we show that keypoint sampling is a valid alternative to keypoint detection for ground texture based localization with available prior. For three state-of-the-art feature-based localization methods, we achieve comparable success rates using sampled keypoints. This shows that it is not necessary to recognize prominent features of the texture. Second, we present a method based on keypoint sampling with more than 90% success rate, which takes less than half as long as the next fastest one. Also, further improvements are evaluated.

II. RELATED WORK

We review ground texture based localization approaches, fast-to-compute detectors, and work on keypoint sampling.

[1]Robert Bosch GmbH, Hildesheim, Germany
SchmidJanFabian@gmail.com
[2]VSI Lab, CS Dept., Goethe University, Frankfurt am Main, Germany
[3]Norwegian Open AI Lab, CS Dept., NTNU Trondheim, Norway

A. Ground texture based localization

Incremental ground texture based localization (e.g. [7]) uses overlapping images to estimate the camera pose relative to the previous one. *Absolute* localization methods estimate the pose using a pre-recorded map of reference images.

Some approaches are designed for absolute *global* localization without available prior pose estimate (Micro-GPS [3], StreetMap [2] variant with bag of words image retrieval). Here, we are concerned with absolute localization *with* available prior, which means that localization is initialized differently (e.g. using GPS). *Ranger* [1] is one of the best performing methods for this task [4]. It is a feature-based approach, using CenSurE [8] keypoints and rotated BRIEF [9] feature descriptors. For feature matching, Ranger performs nearest neighbor (NN) matching with cross check. Based on the prior, the method selects the closest reference image to start with. If it has at least 25 matches with the query image, they are used for RANSAC pose estimation, otherwise Ranger considers the next closest reference image. Another state-of-the-art approach that takes advantage of an available prior is *StreetMap* [2]. It considers a set of closest reference images to the prior. SURF [10] features are extracted from these images and from the query image. Matches are computed using NN matching with ratio test constraint [5], and used to estimate the camera pose with RANSAC. There are StreetMap variants for global localization and for tiled ground textures, which we do not consider.

In our previous work [4], we introduced the identity feature matching approach for ground texture based absolute localization with and without prior. This feature matching approach considers features as matches only if they have identical descriptors. An efficient implementation of identity matching is to employ a lookup-table, where features are sorted into the table based on the value of their descriptor. Such a table is build for each reference image individually. We suggested to employ the first 15 bits of LATCH [11] as a compact binary feature descriptor. LATCH computes descriptors through comparisons of a center image patch from the keypoint position with two surrounding image patches. For the further procedure, key design choices of *Micro-GPS* [3] were adopted: keypoint objects are extracted with SIFT, in a voting procedure each feature match votes on a grid map for its corresponding camera position, and the pose is estimated in a RANSAC procedure, Our method was shown to achieve localization performance competitive with the state-of-the-art. An advantages of it is that it allows to exploit prior pose estimates, which was not possible with Micro-GPS's pre-computed approximate nearest neighbor search index. Our method uses the prior to decide which look-up tables, i.e. which reference images, are taken into consideration. Therefore, identity matching can significantly decrease localization time. This leaves the use of SIFT as the bottleneck of this system in terms of computation speed.

B. Speed-optimized keypoint detection

Some keypoint detectors are designed with computation speed in mind. Among the fastest ones is FAST [12], a corner detector that considers a pixel to be part of a corner, if multiple contiguous surrounding pixels are significantly darker or brighter. Another fast-to-compute corner detector is Good Features To Track (GFTT) [13]. It simplifies the corner-score function of the Harris detector.

A second type of detectors use image pyramids to find scale-invariant keypoint objects. One of the most successful ones is SIFT [5]. It detects blobs as local intensity extrema in a Difference-of-Gaussian (DoG) pyramid. SURF [10] and CenSurE [8] approximate the DoG, using the faster to compute Difference-of-Boxes or Difference-of-Octagons.

C. Keypoint sampling

Keypoint detection typically identifies image regions that fulfill a keypoint criterion or that maximize a keypoint score function. *Keypoint sampling*, on the other hand, determines keypoints independent of the image content. Keypoints can be sampled uniformly [14] or randomly [15]. It has been used for image understanding tasks, where it is not necessary to retrieve corresponding image regions.

III. PROBLEM STATEMENT

The task, we are dealing with consists of two parts: map generation and localization. To generate the map, we assume that a set of reference images with globally optimized (ground truth) poses are available. The map stores reference features, which can be used to find correspondences with the query image. For localization, a query image is recorded independently of the reference images, i.e. at a different time, with independent camera orientation, and potentially under different illumination conditions. We assume that a prior pose estimate with the approximate position and orientation is available. Subsequently, the map is used to estimate the query image pose relative to the reference images. We adopt the specification of Zhang et al. [3] for correct pose estimates: an estimate is correct if the absolute difference to the true pose is less than $4.8\,\text{mm}$ in position and 1.5 degrees in orientation.

IV. FEATURE REPEATABILITY

A quality measure of keypoint extractors is their *keypoint repeatability* [16], i.e. the ratio of keypoint objects that are found in an image as well as in another overlapping image.

Whether keypoint objects are suited for a task also depends on the employed feature descriptor and the matching strategy. Therefore, we introduce *feature repeatability*, which considers the whole feature instead of only the keypoint object. Feature repeatability is computed as the ratio of features from the overlap of two images that are correctly matched with each other. With a correct match it should be possible to determine the query image pose. Therefore, we evaluate correctness of a match by evaluating the correctness of its pose estimate (with the same thresholds used to determine the correctness of a localization attempt).

Another quality of interest for our method is what we refer to as *descriptor repeatability*. It measures the ratio of corresponding keypoint objects that are evaluated to the same

feature descriptor value. To evaluate this, we determine corresponding keypoint objects by projecting keypoint objects from an image a into an overlapping image b, and compare the descriptor values assigned to them in the two images.

V. METHOD

We build on our previous method [4], but keypoint objects are not detected with SIFT, they are sampled randomly or uniformly. Figure 1 presents SIFT, random, and uniform keypoints. Random keypoints are retrieved as a simple random sample of image coordinates without replacement, which has greater computational cost than uniform sampling. However, using uniform sampling for query and reference images could lead to a situation where all keypoints are misaligned. Therefore, we use random keypoints during mapping and uniform keypoints during localization. This does not affect the success rate, compared to a setting in which we also use random keypoints during localization. The keypoint object orientation is given by the relative camera orientation to the map coordinate system, which is known during mapping and estimated during localization using the prior. The constant keypoint object scale is fixed by the descriptor. Keypoint sampling leads to poor keypoint repeatability, and consequently to poor feature repeatability. However, with more reference features, feature repeatability increases (Figure 2).

Extensions: Three extensions to our method are evaluated.

1) *Repeatability constraint* for reference features (illustrated in Figure 3): only reference features with repeatable descriptors are stored during mapping. This can be done, if there are overlapping reference images. We project keypoint objects into the overlapping image to check if they are stable, i.e. if they are evaluated to the same descriptor, only then features are stored.
2) ***M**ulti-**m**ap (MM) approach*: create multiple maps using the same reference images, but different sets of extracted features, and localize with each of them. The final pose estimation is determined by the localization attempt with most RANSAC inliers. This strategy is interesting for random keypoint sampling, as each map will store a different set of features.
3) ***M**ulti-**m**ap approach with varying **o**rientations (MMO)*: for each map we apply a slightly different orientation to the keypoint objects (default is to use the ground truth orientation). Using additional maps with deviating orientations (e.g. with ± 5 degrees) increases the independence of the localization attempts and the robustness to orientation.

VI. EXPERIMENTS

We evaluate on six ground texture datasets (fine and coarse asphalt, carpet, concrete, tiles, and wood) from the database of Zhang et al. [3]. The images have a resolution of 1288 by 964 pixels, which corresponds to an area of about 0.2 m by 0.15 m. For each texture there are about 2000 to 4000 images for mapping; for localization there are additional independently recorded image sequences.

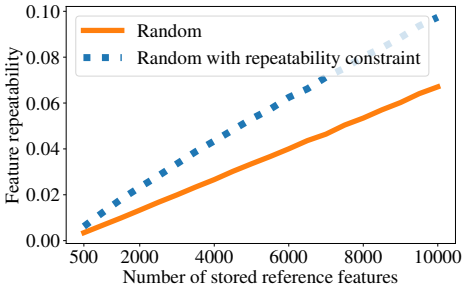

Fig. 2: Feature repeatability for varying numbers of stored reference features, using random keypoint sampling, and identity matching with 15-bit LATCH descriptors.

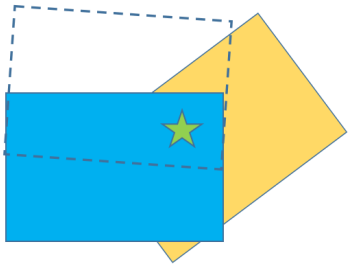

Fig. 3: Repeatability constraint: a feature (green star) is only stored for the reference image (blue rectangle) if the corresponding keypoint object in an overlapping reference image (dashed rectangle) is evaluated to the same feature descriptor. We expect this to increase the chance of corresponding keypoint objects from reference and query image (yellow rectangle) being evaluated to the same feature descriptor.

Besides random and uniform keypoint sampling, we examine SIFT and SURF, as well as the three detectors, which we found to be the fastest [6] among the implemented methods in OpenCV 4.0 [17]: FAST, GFTT, and CenSurE. Since these detectors do not provide keypoint object orientation, we apply the same strategy as for keypoint sampling, using the orientation of the localization prior. Parameters are optimized for keypoint repeatability and computation speed [6]. For all series of experiments involving random keypoint sampling, results are averaged over three repetitions.

In addition to our localization method, we examine Ranger [1] and StreetMap [2], which we reimplemented, using the details provided by the authors. All methods are evaluated with the same query and reference images, and the same random seeds to generate the localization prior.

In order to generate the prior pose estimates, we take the ground truth poses, shift them with a fixed distance into a randomly sampled direction, and rotate it with an orientation sampled from a zero-mean normal distribution.

How many reference images are considered during localization attempts depends on the shift distance. This number should be chosen so that overlapping reference images are included with high likelihood. We empirically determine the numbers that enable successful localization, they range from 5 images for experiments with zero shift distance, over 20 images for distances of 0.1 m, to 250 images for 0.5 m.

TABLE I: Mean descriptor repeatability values with and without applied repeatability constraint.

Repeatability constraint	Random	Uniform	FAST	GFTT	CenSurE	SIFT	SURF
No	4.2%	4.0%	5.0%	5.0%	5.1%	4.6%	6.1%
Yes	7.1%	7.0%	7.5%	8.0%	7.0%	6.3%	7.4%

TABLE II: Mean feature repeatability values with and without applied repeatability constraint.

Keypoint selection	Repeatability constraint	Random	Uniform	FAST	GFTT	CenSurE	SIFT	SURF
Based on keypoint score	No	0.7%[1]	0.7%[1]	8.0%	9.8%	12.7%	13.0%	15.6%
Based on keypoint score	Yes	1.2%[1]	1.3%[1]	1.7%	4.4%	3.0%	6.3%	1.0%
Random selection	No	0.7%	0.7%	0.7%	9.8%	1.2%	13.0%	1.3%
Random selection	Yes	1.2%	1.3%	1.3%	5.4%	2.3%	6.3%	2.5%

During localization, reference images are chosen based on proximity to the prior position estimate, using a k-d-tree.

Implementations: We describe the implementation details of the evaluated localization methods. Some aspects, such as the employed keypoint extraction method, are changed for individual experiments. For all keypoint detectors and feature descriptors the OpenCV 4.0 implementations are used.

For our method, keypoint objects are either detected using SIFT (we call this variant *Ours*) or sampled randomly for reference images and uniformly for query images (*Ours (Random)*). For the SIFT variant, up to 850 keypoint objects are extracted with the following parameters: 11 layers per pyramid octave, a contrast threshold of 0.005, an edge threshold of 13, and a Gaussian smoothing with sigma of 8.5. For Ours (Random), 5000 keypoints per reference image are randomly sampled, and for query images 2000 keypoints are sampled uniformly. Features are described using the first 15-bit of the oriented version of the LATCH descriptor. The half-size is set to 8, and the sigma of the Gaussian smoothing to 2.2. For localization, query image features and reference image features are matched with the identity feature matching approach [4]. Each match is used to cast a vote for the camera position on a voting map. Finally, the camera pose is estimated in a RANSAC process using the matches that voted for the voting map cell with most votes.

For StreetMap, we extract up to 768 SURF features, using an image pyramid with 5 octaves with 4 layers each, and a Hessian threshold of 20. Feature matching is performed using the L2 norm, the OpenCV brute force matcher for nearest neighbor (NN) matching, and a ratio test with threshold of 0.9. The remaining matches between query and reference images are used for RANSAC based pose estimation.

For Ranger, we extract up to 1250 keypoint objects with CenSurE (in OpenCV called StarDetector), with a maximum patch size of 14, a response threshold of 0, a projected line threshold value of 29, a binarized threshold value of 22, and a non-maximum suppression size of 2. Keypoint objects are described using the rotation invariant 64-byte BRIEF descriptor. For localization, Ranger selects the closest reference image to the prior and matches its features with the query image features, using the OpenCV brute force matcher with Hamming norm and cross check. Subsequently, the camera pose is estimated in a RANSAC procedure. If there are at least 25 RANSAC inliers, the localization terminates; otherwise, matching and pose estimation are repeated with the next closest reference image. In case that none of the considered reference images satisfies the condition, we use the attempt that had the most RANSAC inliers.

VII. RESULTS

First, we evaluate the descriptor and feature repeatability; then, localization performance of the introduced methods.

A. Evaluation of descriptor and feature repeatability

Descriptor and feature repeatability are examined for different keypoint extraction approaches. The evaluation is done for our method, i.e. the first 15-bit of LATCH are used as descriptors, and, using identity matching, matches are proposed for all pairs of features with the same descriptor.

Additionally, the repeatability constraint is examined. For this purpose, overlapping reference images are required, which are not available in the employed dataset. Therefore, we use the image sequences intended for localization, which have significant intersection area (22.7% on average), as reference images, and images intended for mapping as query images. The results, presented in Table I and II, are averaged over 600 image pairs (100 per texture type) of reference and query image pairs with an intersection of at least 20%. From each image of these pairs up to 1000 features are extracted.

The descriptor repeatability is similar for most keypoint extractors. Thus, the probability of evaluating corresponding keypoint objects to the same 15-bit LATCH descriptor is not strongly dependent on the keypoint extractor. The repeatability constraint increases the descriptor repeatability all over.

In order to evaluate feature repeatability, it is necessary to specify the keypoint object orientation. Here, we use the ground truth image orientation, effectively eliminating the need for robustness against orientation. For SIFT and SURF, which could provide orientation information on their own, this improves the performance. The results (Table II first row) confirm previous findings that SIFT, SURF, and CenSurE are among the best keypoint detectors for ground texture images [6]. Since descriptor repeatability is similar for all keypoint extraction methods, variances in feature repeatability result from variances in keypoint repeatability.

[1]Keypoint score not available; therefore, equivalent to random selection.

TABLE III: Evaluation with position error of 0.1 m, and standard deviation of the orientation error of 5.0 degrees.

Method	Keypoint detector	Feature descriptor	Matching strategy	Success rate	Computation time (ms) Overall	Detection	Desc.	Matching	Pose est.
Ranger	CenSurE	BRIEF	NN + cross check	**99.9%**	55.7	30.4	9.5	14.7	< 0.1
Ranger	FAST	BRIEF	NN + cross check	**99.9%**	61.8	14.1	9.2	34.0	4.1
Ranger	Random + Uniform	BRIEF	NN + cross check	99.6%	187.6	< 0.1	9.2	147.4	30.6
StreetMap	SURF	SURF	NN + ratio test	99.2%	153.3	95.6	18.4	38.8	< 0.1
StreetMap	CenSurE	SURF	NN + ratio test	83.7%	74.2	28.6	4.4	40.3	< 0.1
StreetMap	Random + Uniform	SURF	NN + ratio test	99.1%	116.9	< 0.1	21.4	94.7	< 0.1
Ours	SIFT	15-bit LATCH	Identity matching	99.1%	730.1	716.9	10.1	2.1	< 0.1
Ours	CenSurE	15-bit LATCH	Identity matching	95.2%	40.4	29.0	8.3	2.3	< 0.1
Ours	Random + Uniform	15-bit LATCH	Identity matching	93.5%	**25.4**	< 0.1	17.5	5.0	2.3
Ours MMO-2	Random + Uniform	15-bit LATCH	Identity matching	97.9%	33.2	< 0.1	18.0	10.4	4.7
Ours MMO-4	Random + Uniform	15-bit LATCH	Identity matching	99.5%	47.9	< 0.1	17.8	20.6	9.4

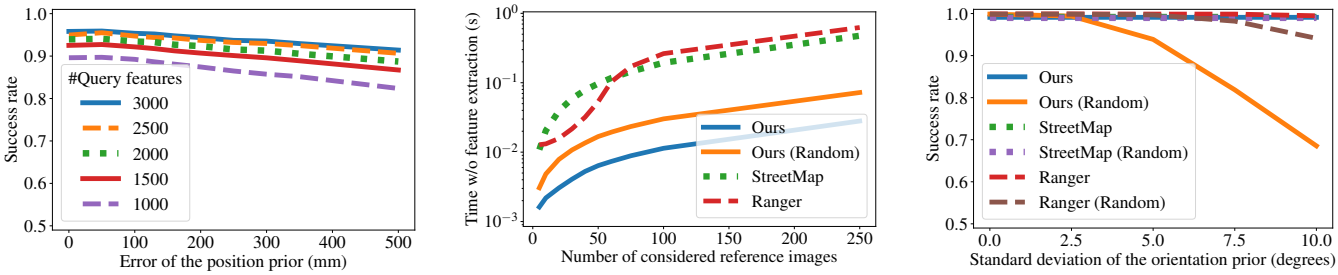

Fig. 4: **Left:** success rate of our method with varying numbers of query image features for varying position prior accuracies. **Center:** localization time without keypoint detection and feature description for varying numbers of considered reference images. **Right:** success rate for increasing standard deviation of the orientation prior.

This explains the poor performance of random and uniform keypoints. If the number of stored reference features is increased to 5000, the score of random keypoints improves from 0.7% to 3.3%. Furthermore, the repeatability constraint increases feature repeatability of sampled keypoints (Table II second row). Again, using 5000 instead of 1000 reference features further improves the result (from 1.2% to 5.3% for random keypoints). Interestingly to us, the feature repeatability of keypoint detectors worsens with applied repeatability constraint. This can be explained with the keypoint selection method. The evaluated keypoint detectors provide a keypoint score, which can be used to select the *best* 1000 features per image. Selecting a subset of the detected keypoints randomly instead, decreases the feature repeatability (Table II third row). This is not the case for GFTT and SIFT, because, different to the other methods, they find less than 1000 keypoints on average (SIFT 392.5, GFTT 933.9). If we apply the repeatability constraint to randomly selected keypoints, the feature repeatability is increased, as for randomly and uniformly sampled ones (Table II fourth row). Again, this is not the case for GFTT and SIFT, because their already small number of extracted features is further decreased with applied repeatability constraint. Overall, only sampled keypoints benefit from the repeatability constraint. For keypoint detectors it is better to rely on score based selection.

B. Evaluation of localization performance

For the following experiments, we evaluated 500 test sequence images per texture type. Figure 4 (left) shows localization success rates for our method with different numbers of query features and with increasing position prior error, the applied orientation prior has a standard deviation of 5.0 degrees. We notice, using more than our default value of 2000 query features does not significantly increase performance. A prior with error of up to 0.2 m decreases the success rate of our method only slightly: with 2000 query features it decreases from 94.1% without error to 92.2% with an error of 0.2 m. For larger errors, it decreases more significantly (88.7% for an error of 0.5 m). Success rates of Ranger and StreetMap are less affected by the position error, with an error of 0.5 m they still reach success rates of 99.8% and 97.6%, but they become slow with larger numbers of considered reference images (Figure 4 (center)), due to the use of nearest neighbor matching. Timings are measured on a PC with E3-1270 Intel Xeon CPU. Ranger is particularly fast for small position errors (and therefore small numbers of considered reference images), because it terminates as soon as a well matching reference image is found. In the following, we fix the position prior error to 0.1 m.

A key parameter for localization with keypoint sampling is the number of features stored per reference image. For Ranger, we find that with 750 sampled keypoints it already reaches a similar success rate as with CenSurE (99.6% to 99.9%). StreetMap benefits from larger numbers of reference features; with 2000 sampled keypoints it reaches 99.1% success rate. Using more keypoints improves performance slightly, but significantly increases computational cost. Our method, due to the use of identity matching, requires more

reference features to reach good performance, good performance is reached with 5000 reference features.

The evaluation of the localization success rate for varying orientation prior accuracies (Figure 4 (right)) demonstrates the limits of our method with sampled keypoints. The method relies on corresponding keypoint objects being evaluated to the same descriptor value, which requires a good estimate of keypoint object orientation. StreetMap is not affected by the orientation prior, because during feature description SURF estimates keypoint orientation itself. Ranger (Random) has decreased success rates only for large errors in the orientation prior. The following experiments are evaluated with an orientation prior with a standard deviation of 5.0 degrees.

We evaluate the localization methods, using their default keypoint detectors, the fast-to-compute detectors CenSurE, FAST, and GFTT, and keypoint sampling. Table III presents the results; for the fast-to-compute detectors, we only present the one with highest success rate. With GFTT keypoints, Ranger has a success rate of 95.0%, StreetMap 39.7%, and our method 85.7%. StreetMap with FAST keypoints reaches 63.7% success rate, and our method 88.1%.

Ranger is the most successful method with 99.9% success rate using CenSurE or FAST keypoints. Our method with keypoint sampling is with 25.4 ms the fastest method, taking less than half as long as Ranger, but with 93.5% it has a lower success rate. Ranger and StreetMap reach similar success rates with keypoint sampling as with their default detector, but it increases their computation time, mainly due to the use of nearest neighbor matching. This is particularly devastating for Ranger, because it matches with multiple reference images individually.

Evaluation of the multi-map approach: For this approach, multiple maps with independently sampled random keypoints are created. Using two maps increases the success rate to 95.1% (from 93.5% with a single map), with four maps it increases to 95.6%. The multi-map approach with applied orientation deviations further improves performance. Using two maps with orientation deviations of ± 2.5 degrees increases the success rate to 97.9%, with 8 ms longer computing time. (Ours MMO-2 in Table III). With four maps and orientation deviations of ± 6.0 and ± 2.0 degrees (Ours MMO-4 in Table III), the method has a success rate of 99.5%, still being 8 ms faster to compute than Ranger with CenSurE. Also, with multiple localization attempts, we can perform a consistency check. If we require that at least two of the pose estimations are close to each other (closer than 4.8 mm and with less than 1.5 degrees orientation difference), we can reject 63 out of 64 unsuccessful localization attempts for MMO-2, while also rejecting 14.0% (412) of successful attempts. With MMO-4 this would lead to a rejection of 11 of 14 unsuccessful attempts, and 1.84% (55) successful ones. Instead of storing multiple maps with independently sampled features, one could store multiple times as many reference features. However, storing more than 5000 features per reference image does not significantly increase performance. The advantage of the multi-map approach lies in its ability to perform multiple *independent* localization attempts.

VIII. CONCLUSION

We have shown that it is not necessary to perform proper keypoint detection for feature-based localization with a downward-facing camera. The evaluated methods can achieve similar localization success with detected and (randomly) sampled keypoints. Ranger and StreetMap are slowed down, however, due to the use of nearest neighbor matching. Our method, using identity matching, is significantly sped up. This acceleration is paid for by a lower success rate. We suggest three possible solutions to this: (1) the proposed method has a success rate of 99.8% if the error of the orientation prior is smaller than 4 degrees. Because of the short computation time of the method, it can localize in short intervals. Therefore, an orientation prior with sufficient accuracy might be available in practice; (2) we show that the multi-map approach with applied orientation deviations increases the success rate, it can be used to scale the computational effort according to the expected prior accuracy; (3) for further research, we propose the repeatability constraint. It increases the number of correct matches for sampled keypoints, but requires to store overlapping reference images.

REFERENCES

[1] K. C. Kozak and M. Alban, "Ranger: A ground-facing camera-based localization system for ground vehicles," in *PLANS*, April 2016, pp. 170–178.

[2] X. Chen, A. S. Vempati, and P. Beardsley, "StreetMap - mapping and localization on ground planes using a downward facing camera," in *IROS*, Oct 2018, pp. 1672–1679.

[3] L. Zhang, A. Finkelstein, and S. Rusinkiewicz, "High-precision localization using ground texture," in *ICRA*, 2019, pp. 6381–6387.

[4] J. F. Schmid, S. F. Simon, and R. Mester, "Ground texture based localization using compact binary descriptors," in *ICRA*, 2020, pp. 1315–1321.

[5] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, Nov 2004.

[6] J. F. Schmid, S. F. Simon, and R. Mester, "Features for ground texture based localization - a survey," in *BMVC*, 2019.

[7] N. Nourani-Vatani, J. Roberts, and M. V. Srinivasan, "Practical visual odometry for car-like vehicles," in *ICRA*, May 2009, pp. 3551–3557.

[8] M. Agrawal, K. Konolige, and M. R. Blas, "CenSurE: Center surround extremas for realtime feature detection and matching," in *ECCV*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 102–115.

[9] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in *ECCV*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 778–792.

[10] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *ECCV*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417.

[11] G. Levi and T. Hassner, "LATCH: Learned arrangements of three patch codes," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016, pp. 1–9.

[12] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *ECCV*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 430–443.

[13] J. Shi and C. Tomasi, "Good features to track," in *CVPR*, January 1994, pp. 593 – 600.

[14] H. Chatoux, F. Lecellier, and C. Fernandez-Maloigne, "Comparative study of descriptors with dense key points," in *ICPR*, Dec 2016, pp. 1988–1993.

[15] E. Nowak, F. Jurie, and B. Triggs, "Sampling strategies for bag-of-features image classification," in *ECCV*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 490–503.

[16] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool, "A comparison of affine region detectors," *IJCV*, vol. 65, no. 1, pp. 43–72, Nov 2005.

[17] G. Bradski, "The OpenCV library," *Dr. Dobb's Journal of Software Tools*, 2000.