Benchmarking Visual Localization for Autonomous Navigation

Lauri Suomela Jussi Kalliola Atakan Dag Harry Edelman Joni-Kristian Kämäräinen
Tampere University, Finland
{lauri.a.suomela, jussi.kalliola, atakan.dag, harry.edelman, joni.kamarainen}@tuni.fi

Abstract

This work introduces a simulator-based benchmark for visual localization in the autonomous navigation context. The dynamic benchmark enables investigation of how variables such as the time of day, weather, and camera perspective affect the navigation performance of autonomous agents that utilize visual localization for closed-loop control. The experimental part of the paper studies the effects of four such variables by evaluating state-of-the-art visual localization methods as part of the motion planning module of an autonomous navigation stack. The results show major variation in the suitability of the different methods for vision-based navigation. To the authors' best knowledge, the proposed benchmark is the first to study modern visual localization methods as part of a complete navigation stack. We make the benchmark available at https://github.com/lasuomela/carla_vloc_benchmark.

1. Introduction

One of the most impressive capabilities of the human brain is the ability to take a look around, answer the question "Where am I?", and use a mental map of the environment to guide one to a place that has been visited before. A task that seems trivial to humans is notoriously difficult for robots. One promising approach to *autonomous navigation* is vision-based navigation that uses *visual localization* [54] to estimate the pose of an agent with respect to a metric map. The map is created prior to navigation by creating a 3D reconstruction from a "gallery set" of images representing the environment. The poses of new "query" images can be estimated by matching them to the gallery set. The pose information can then be utilized for navigating to different points on the map. This process is illustrated in Fig. 1.

Much of the ongoing research on visual localization focuses on developing methods that are more robust to viewpoint and appearance changes between the query and gallery images. In recent years, various benchmarking datasets have been published [20, 30, 54], and visual localization challenges have been hosted as part of the top-tier computer vision conferences. The new methods strive for even more accurate results on these benchmarks.

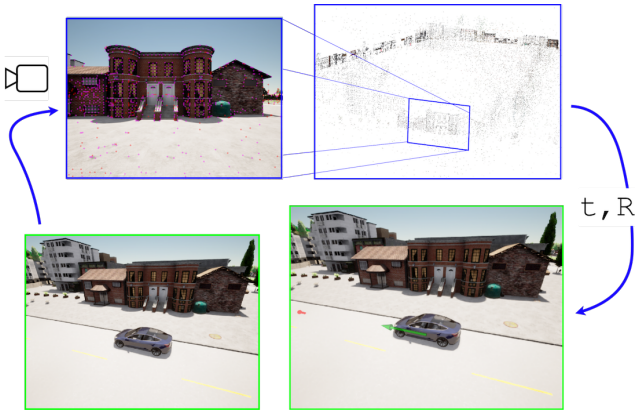


Figure 1: The vehicle finds its pose in a simulated environment by matching local features to a pre-built 3D model.

The most common applications of visual localization that are mentioned in the literature are autonomous driving and augmented reality [30, 54]. However, none of the new deep learning based localization methods have been demonstrated as part of a robot navigation stack. This raises the question: how relevant are the performance metrics used by the visual localization benchmarks and challenges for autonomous navigation? And how accurate do the localization methods actually have to be in order to enable autonomous robot navigation?

This paper seeks to address these concerns. We present a benchmark based on the Carla simulator [18] that enables testing visual localization methods for autonomous navigation. In the environment, the user can test how various state-of-the-art methods for visual localization perform when they are used for guiding the navigation of an autonomous car. This ability to directly test the performance of visual localization algorithms in their intended purpose enables the discovery of relevant new research problems, as compared to focusing on just measuring the algorithms' accuracy. Using a simulator also enables experiments that study the effects of factors such as illumination conditions, weather, and camera perspective on visual navigation. Furthermore, the use of a simulator enables comparing the output of the visual localization algorithms with the accurate

ground truth location of the autonomous vehicle, something that is not usually possible with real-world datasets [7]. As pointed out by Brachmann *et al*. [7], synthetic data seems "easier" for a visual localization method to handle when compared to real camera data. Because of this, the navigation results reported in our work are likely to be overly optimistic. Even so, we argue that the ability to test an end-to-end visual navigation stack provides important new knowledge to the discussion on visual localization methods.

Our main contributions are: **1)** A simulator benchmark that enables development and evaluation of visual localization methods for autonomous navigation tasks; **2)** An example use-case of the benchmark: evaluation of the state-of-the-art visual localization methods as part of a navigation stack; **3)** Novel findings that connect an established visual localization performance metric, recall rate, with a proposed new metric, failure rate. All results are fully reproducible and the benchmark is publicly available.

2. Related Work

This work focuses on application of visual localization to *autonomous navigation*, which can be used by mobile robots to handle various tasks such as delivery, inspection and people transportation [22, 62]. Vision-based navigation is useful in conditions where GPS or other sensors such as LiDAR [45], motion capture [38] or active localization beacons [33] are not available or fail. The advantage of vision-based methods is that they only require commodity RGB-cameras that are cheap and power-efficient.

Vision-based navigation. There are various approaches to vision-based navigation. One of the most important factors differentiating the methods is the kind of prior information the navigating agent has about the environment it is operating in. The environment can be completely unknown, or the agent can have access to a a map representing the environment [2].

In unknown environments a robot has to explore its surroundings. Its task can be to navigate to specific coordinates [13], find a certain object [12] or map the space [16]. For many applications, the ability to operate in known areas is sufficient [9, 23]. In such cases, no exploration is needed. The robot can use cameras to determine its pose on a map representing the environment. The pose in turn can be used to plan a route to the robot's goal. The map can be a topological collection of images along the robot's route [15], a full metric map of the environment [43], or even implicitly encoded in an action policy derived by reinforcement learning [41]. In this taxonomy, visual localization falls under the group of methods that utilize metric maps. Visual localization has been utilized for navigation purposes in planetary rovers [23], wheeled utility robots [44] and drones [43, 64], for example.

Visual localization. There are various approaches to visual localization, such as pose regression [29], scene coordinate regression [58] and direct image alignment [53, 61], but in recent years *hierarchical localization* [27, 51] methods have dominated the benchmarks.

Hierarchical localization consists of two stages. As a precondition, let's assume access to a "gallery" set of images representing the environment, from which a 3D reconstruction has been created by SLAM or SfM. At first stage of localization, the gallery images most similar to a new "query" image are retrieved using place recognition methods [37, 66]. Then, local features extracted from the query image and the most similar gallery images are matched [32]. The real-world locations of the gallery features are known from the 3D reconstruction, so the resulting 2D to 3D correspondences enable estimating the 6-DoF pose of the query image using Perspective-n-Point (PnP) methods [24]. The hierarchical visual localization approach has proven to be robust to changes in viewpoint and appearance, and is computationally feasible even for large-scale environments [51].

One of the characteristics of navigation which is relevant for visual localization is the sequential nature of the image data that robots' cameras capture. The continuous motion provides a strong prior that can be utilized in the prior retrieval stage by retrieving best matching image descriptor sequences instead of individual images [39, 42], by creating descriptors representing whole image sequences [25] or by using the current pose estimate as a prior for topological localization [60]. At the local feature matching stage, the generalized camera model [46] enables estimating the camera trajectory from multiple images simultaneously [60]. Kalman filters [8], particle filters [1] and graph-based methods [43, 64] can further process the pose estimates to enable sensor-fusion and outlier rejection.

Benchmarks. To the authors' best knowledge the work presented in this paper is the first dynamic benchmark where the components of visual localization based navigation stack can be individually developed and evaluated in fair and reproducible manner. Traditionally, performance of visual localization methods has been evaluated using static datasets of real images (*i.e*. Aachen Day-Night [54], Oxford RobotCar [34], CMU VL [5] and Visual Localization Benchmark [54]) and synthetic images (*i.e*. SimLocMatch [6], TartanAir [63] and V4RL [35]). While these datasets enable evaluating the accuracy of visual localization, they do not provide insights into how well the methods are suited for navigation tasks.

Simulators, on the other hand, enable reproducible experiments with sufficiently realistic interactions. Several simulated benchmarks for vision-based navigation exist. The annual challenges [21, 59] of the iGibson [65] and Habitat [55] simulators on tasks such as PointGoal and Ob-

jectGoal navigation [2] are a good example. While they provide good platforms for evaluating agents' navigation performance, the benchmarks aren't tailored for the analysis of visual localization methods: the focus is on operation in unknown environments. Our proposed Carla-based benchmark is specifically aimed for evaluation and investigation of the performance of visual localization when applied to the context of autonomous navigation.

3. Simulation Benchmark

Based on the discussion in Sec. 2, we identified a research gap in the application of visual localization for autonomous navigation. Visual localization is an active research topic in computer vision, but methods are evaluated using static datasets and it is unclear how well the methods work when the visual localization output is used for closed-loop control. As a solution we present a benchmark which enables easy experimentation with different visual localization methods as part of a navigation stack. The platform enables investigating various factors that affect visual localization and subsequent navigation performance, for example those listed in Table 1.

The benchmark is based on the Carla autonomous driving simulator [18] and our ROS2 [36] port of the Hloc visual localization toolbox [50]. Carla was chosen because of its simplicity of use, relatively high level of photorealism and ROS2 support via the Carla ROS bridge module. ROS2 enables easy integration of the demonstrated visual localization package with different robotic platforms. We want to emphasize that Carla was chosen independent of its automotive application. The insights of this paper concern autonomous robot navigation in general, not just autonomous driving.

3.1. ROS2 Visual localization package

In order to provide a generic visual localization interface to autonomous agents, we created the *ROS-Hloc Package*. It is a ROS2-wrapper for the Hloc toolbox [50] that is a collection of state-of-the-art visual localization methods and utility functions. The original toolbox is designed for static image collections, but our ROS-Hloc extends it to images arriving in a real-time stream.

Table 1: Factors that affect visual localization performance and whether they are supported by our benchmark and demonstrated in this paper.

F#	Factor	Possible	Reported
F1	Illumination	✓	✓
F2	Weather	✓	✓
F3	Viewpoint changes	✓	✓
F4	Scene structure	✓	✓
F5	Time of year (seasons)	✗	✗
F6	Camera placement (extr.)	✓	✗
F7	Camera parameters (intr.)	✓	✗
F8	Multiple cameras	✓	✗
F9	Dynamic objects	✓	✗
F10	Headlights	✓	✗

The ROS-Hloc workflow is as follows. First, a gallery set is collected for each test environment. Inside the simulator this is achieved by driving a reference run with Carla's built-in autopilot. Along the route, images are captured by a camera attached to the vehicle. The images are taken at steady intervals, and saved to disk along with the exact camera pose. After the gallery set has been captured, the images are processed to extract global and local feature descriptors. These are saved in a gallery database for queries. The 3D locations of the extracted local features are then estimated with the Colmap SfM library [56, 57]. Instead of running full SfM reconstruction we use point triangulation from known camera poses [26]. This produces higher quality 3D scene models than reconstructions from unordered collections of images. In the simulator, acquiring the exact camera poses is trivial. In the real world, LiDAR SLAM methods can be employed in the mapping phase to ensure high-fidelity 3D models [10, 30].

At inference time, the vehicle captures a query image which is sent to ROS-Hloc for pose estimation. First, the most similar gallery images are retrieved by place recognition. The retrieved gallery images are divided into spatial clusters using co-visibility clustering [51]. Then, local feature matching is used for establishing query-to-gallery 2D-3D correspondences. The correspondences are used as inputs to the Perspective-n-Point (PnP) [24] solver provided by Colmap to produce 6DoF pose estimates for each cluster. The pose estimate from the cluster with the highest number of inlier 2D-3D correspondences is chosen as the final output of the visual localization pipeline. This pose estimate is forwarded to the agent's *motion planning and control stack* where it is used for producing steering commands.

Hloc includes various localization method options. There are two global descriptor methods, NetVLAD [4] and Ap-GeM [48], which we test in conjunction with the four supported local feature extractors, SIFT [31], D2-net [19], R2D2 [49] and SuperPoint [17]. With all methods besides SuperPoint we conduct the local feature matching by nearest neighbor search with ratio test [32] (NN-ratio). With SuperPoint, we utilize the SuperGlue matcher [52] instead.

3.2. Vehicle motion planning & control components

In addition to ROS-Hloc for visual localization, the navigation stack needs two more components: a *motion planner* and a *controller*. Motion planner is sub-divided into *global* and *local planners*. Based on a route description, a global planner produces a set of waypoints from the vehicle's start position to its target. It is used in combination with a local planner that, at each timestep, finds the current closest

waypoint and passes it as a subgoal to the controller. The controller consists of two proportional-integral-derivative (PID) controllers [3], one for longitudinal and one for lateral control of the vehicle. Its purpose is to produce steering commands to move the vehicle towards the waypoint from local planner. We use the global planner and controller from Carla, the local planner we implemented ourselves.

Sensor fusion & sequential processing. Before being used for motion planning, the pose estimates from visual localization are first forwarded to a Kalman filter which fuses the estimates with measurements from a simulated wheel odometry sensor. The true values from the ideal odometry are injected with gaussian noise to make the sensor more realistic. The reason we fuse the visual pose estimates with wheel odometry data is that the PID-controller of the vehicle requires pose input at a high frequency, which is not achievable with the current state-of-the-art hierarchical visual localization systems. The wheel odometry also enables the vehicle to get estimates of its position when the environment is so degraded that the PnP solver of the visual localization pipeline cannot converge to a solution. The vehicle can navigate using pose information from just wheel odometry, but as the odometry measurements contain noise, the estimated pose accumulates error over time. This drift limits wheel odometry navigation only for short distances. Fusing the wheel odometry with visual localization effectively corrects the drift. We use the extended Kalman filter (EKF) implementation of the robot_localization ROS package [40]. To make the localization system more robust to outliers, visual pose estimates with more than 20 meters deviation from the filter's current state are discarded. This scheme addresses the sequential nature of visual localization in autonomous navigation context at the pose level, and introduces a degree of temporal stability to the pose estimates. We do not add sequential processing to the prior and local feature matching stages. These would be interesting research topics, but we argue that single-image localization is an important starting point for investigating the applicability of visual localization for autonomous navigation.

3.3. Evaluation scheme and performance metrics

Performance measurement methodology is an important part of any benchmark. In this work, we wanted to bring together visual localization and autonomous navigation, and therefore our metrics should be meaningful to both fields. Autonomous driving oriented visual localization performance metrics such as "probability distribution of certain distance driven without localization" [47] and "maximum open loop distance" [14] have been proposed, but whether they indicate success in autonomous navigation or have been invented to circumvent limitations of static datasets is unclear. Navigation performance is often measured by the *Success Rate (SR)* [63] or *Success weighed by Path Length (SPL)* [2]. In the context of our benchmark, SR would be measured by repeating N episodes of the test route, and calculating the ratio between successful navigations of the route and the total number of episodes. SPL extends SR by additionally measuring the deviation from the shortest path to goal. For the metrics used in our benchmark, we combine insights from the fields of visual localization [54], autonomous navigation [2] and visual object tracking [11] (VOT). The performance evaluation needs of VOT are similar to ours, and there the evaluation methodologies have been investigated rigorously. Inspired by the works of Kristan *et al*. [28] and Cehovin *et al*. [11], we adopt two metrics that describe localization *accuracy* and *robustness*.

Recall rate. The first aspect of performance is the accuracy of the visual localization method. This is an intuitive measure of how well a visual localization method performs under different conditions. For each experiment environment we conduct a test where a vehicle drives through the test route with visual localization running. To produce comparable measurements of the accuracy of the visual localization methods, we don't use the estimates for navigation, but only measure their accuracy. For navigation the vehicle uses ground truth pose information from the simulator. Navigating based on the visual localization estimates, which can contain large errors, would lead to the vehicle driving a bit different route on each test run, affecting the repeatability of the accuracy measurements. For each combination of experiment settings and localization methods we report the *localization recall*, which was adopted from Sattler *et al*. [54]. We report the proportion of correct poses within three error thresholds: < 0.25m, 2° (T1), < 0.5m, 5° (T2) and < 5m, 10° (T3). The distance of the estimated pose is compared to the pose of the vehicle at the moment the visual localization input image was taken.

Failure rate. The recall rate of a localization method does not fully describe its performance in autonomous navigation context. A high accuracy value can conceal infrequent but catastrophically large localization failures, which cause the vehicle to crash. The SR [63] and SPL [2] are common metrics for this kind of experiments. However, they do not fit ours' well. In SR and SPL methodologies, an attempt to navigate the test route ends after the first navigation failure along the route: an otherwise easy test route with one difficult segment will result in the same SR or SPL as a test route that is difficult in all its parts. As argued by Cehovin *et al*. [11], it is more informative to measure success with re-initializations. If localization error causes the vehicle to crash, the vehicle and its EKF state are re-initialized back to the point along the route before the failure. This enables evaluating the localization performance along all segments of the test route.

Similar to VOT [28, 11] we report the *average failure rate* of navigation

$$F = \frac{1}{N} \sum_{i=0}^{N} \frac{r_i}{L_{path}} \quad , \tag{1}$$

where N is the number of test episodes, r_i is the number of re-initializations required to complete the route on test episode i and L_{path} is the length of the route in kilometers.

To evaluate the failure rate, we conduct multiple episodes of the vehicle driving a predefined route while localizing based on sensor data. Since the autonomy stack of the vehicle is effectively using the visual localization to correct the wheel odometry drift *i.e.* to improve navigation performance from that of a wheel odometry only based system, we also measure the performance of a vehicle navigating using wheel odometry only. This defines a 'baseline' to which the navigation stack with visual localization should be compared. In easy conditions, the visual localization can be expected to improve navigation performance, while in degraded conditions the estimates can be very wrong and actually harm navigation performance.

4. Experiments

To demonstrate the capabilities of the proposed simulator benchmark we performed experiments to compare the ability of different visual localization methods to cope with gallery-to-query shifts in illumination, camera viewpoint and weather. We want to emphasize that our benchmark is not limited to these factors, and enables further experiments with e.g. those listed in Table 1.

We limit our experiments to the state-of-the-art hierarchical localization methods, and do not consider other approaches such as direct image alignment [61, 53] or sequence-based methods [39, 42]. Testing other method families would be an interesting research topic which we leave for future work.

Environments. Out of the 8 default maps in Carla we selected two that represent very different environments. Town01 is a model of a small town with densely packed buildings and a river in the center. We defined a 1.2-kilometer long test route, from which a gallery set of 615 images was produced. The route consists of straight road segments connected by five 90 degree turns. Town10 is a part of a bigger city with large buildings, skyscrapers and a beach. The route is approximately 0.5 kilometer long with six turns. The captured gallery set has 237 images. Both the gallery sets were gathered by driving around the towns in mid-day sunny conditions and capturing images every 2 meters by a camera pointed perpendicular to the right of the vehicle's direction of travel. These gallery sets were used in all the subsequent experiments.

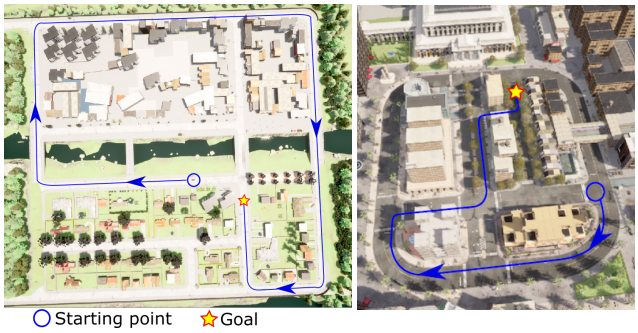


Figure 2: A bird's-eye view of the test routes in Town01 (left) and Town10 (right)

Methods and common parameters. Each of the localization methods included in Hloc (see Sec. 3.1) were tested in all the experiments. None of the methods were retrained with data from the simulator; pretrained model weights were acquired from the original Github repositories. The number of gallery images retrieved by place recognition for pose estimation was set to 5. Ratio threshold of NN-ratio matcher was set to 0.8, and SuperGlue was used with the default parameters. For all of the localization methods we used input image resolution of 800x600 pixels. The target speed of the vehicle and the localization frequency were set to $4m/s$ and $2Hz$, respectively. Magnitude of the wheel odometry noise was set to a level which causes the pose estimate to drift away from the true pose at a rate of 8.5% for the position and $0.4°/m$ for the orientation.

4.1. Illumination change

This experiment evaluated the performance of the state-of-the-art visual localization methods under query-to-gallery illumination change in "Town01" and "Town10" (Fig. 2). The autonomous vehicle was set to perform the test routes under multiple illumination conditions (Fig. 3). In the easiest test scenario the illumination corresponds to that of the gallery set, and in the most difficult scenarios there is almost complete darkness. We report the average failure rate for 5 repetitions of the test route for each localization method and illumination condition. For the recall rate, we only report the results from one run per test condition since the vehicle drives using an error-free controller and therefore the variance between the runs is negligible.

In Carla, illumination is controlled by two parameters: sun intensity and sun elevation angle. These two have to be adjusted jointly to produce a full range of illumination conditions from daylight to darkness. We start from sun intensity value of 1.0 and elevation angle $0.4\pi\ rad$, and for each successive illumination condition we halve the values from the previous condition by parameter k:

$$V_k = V_{base} * 0.5^k \tag{2}$$

where V_{base} is sun intensity or elevation angle value for the

gallery set, and V_k is the elevation angle or intensity for illumination condition k. $k \in [0, 1, ..., 10]$ leads to 11 distinct illumination conditions (see Fig. 3 for examples).

4.2. Viewpoint change

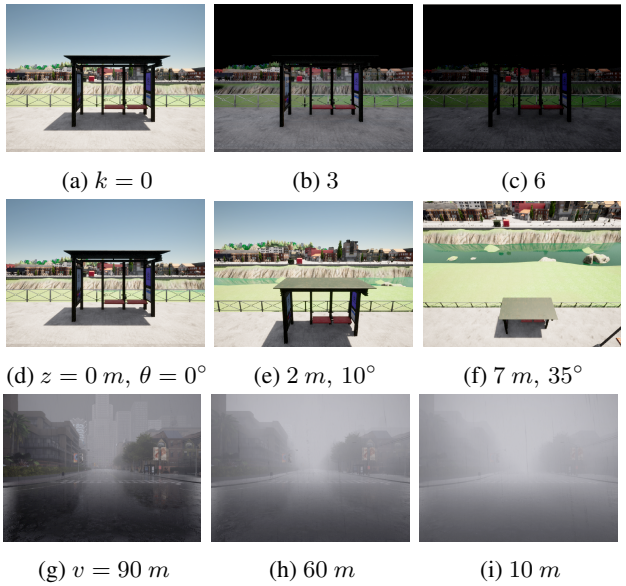
In this experiment, a viewpoint difference was introduced between the gallery and query images. The query images view the same scene content as the gallery, but from a different perspective. This was implemented by introducing a series of vertical offsets z and pitch angle decreases θ to the pose of the localization camera in the test runs (Fig. 3). We report average failure and recall rates of the different offsets, tested in Town01 under illumination level $k = 0$. Other settings were kept the same as in the illumination change experiments in Sec. 4.1.

4.3. Weather change

We conducted additional experiments on the effect of gallery-to-query weather change in Town10. Illumination level was set to $k = 0$ and rain was added into the environment. Then, we introduced progressively increasing amounts of fog. The amount is controlled by defining how close to the vehicle the fog begins. We created 4 weather conditions with visual ranges $v \in [90.0, 60.0, 30.0, 10.0]$ meters (Fig. 3). We report the failure and recall rates over 5 repetitions of each condition and method.

5. Results and Discussion

Here we provide a thorough analysis of the illumination experiment results, and for compact presentation show only the most important findings for the two other experiments: viewpoint and weather change. Full result tables and additional visualizations are provided in the appendix, available in the supplementary material.



(a) $k = 0$ (b) 3 (c) 6

(d) $z = 0\ m$, $\theta = 0°$ (e) 2 m, 10° (f) 7 m, 35°

(g) $v = 90\ m$ (h) 60 m (i) 10 m

Figure 3: Town01 under different illumination (a-c) and viewpoint (d-f) shifts, and weather changes in Town10 (g-i)

5.1. Illumination change

Failure rates. Table 2 shows the navigation failure rates for the illumination experiments. The same data is visualized in Fig. 4. As expected, the failure rates stay low when the gallery-to-query illumination shift is small. The rates rise with increasing severity of the shift. After exceeding the odometry failure rate, the rate for each method peaks and then starts to decrease. Around the peak, the gallery-to-query appearance change is large enough to cause big errors in the visual localization. However, they are not as large as to cause rejection by the EKF $20m$ outlier threshold. As the shift further grows, more pose estimates are rejected by the filter, and the vehicle starts mainly relying on wheel odometry. As result, the failure rate decreases and converges towards that of wheel odometry only.

Combinations using SuperPoint achieve the lowest failure rates, and by a clear margin. The ability of SuperPoint to improve navigation performance is remarkable. The method brings benefits over wheel odometry even at $k = 9$, when localizing the images is very difficult even for the human eye. The best performing combination is that of SuperPoint and NetVLAD, followed by SuperPoint with Ap-GeM. This follows a general pattern: The local feature method seems to have more effect on the performance than the place recognition method. Of the two place recognition methods, NetVLAD provides a slightly better performance. R2D2 and SIFT are consistently tied for the second best local feature. D2, a deep learning based feature, ranks the worst. The good performance of SIFT is interesting: published in 1999, it can still compete with the new methods.

Town comparison. The failure rates exhibit some differences between the two towns. In Town01, SuperPoint has significantly better performance than the other methods. In Town10, this gap is more narrow. This is likely caused by a higher degree of perceptual aliasing in the scene, such as buildings with repetitive textures. However, in both environments the overall order of method performance remains approximately the same - only R2D2 and SIFT switch places on some values of k.

Visual localization recall vs. failure rate. Table 4 presents the recall rates for the illumination experiments. The extent to which recall, a visual localization performance metric, measures navigation performance is an important question. Fig. 5 shows the correlation between the two metrics for the illumination experiments. As expected, the correlation is strong, but the plot also provides two

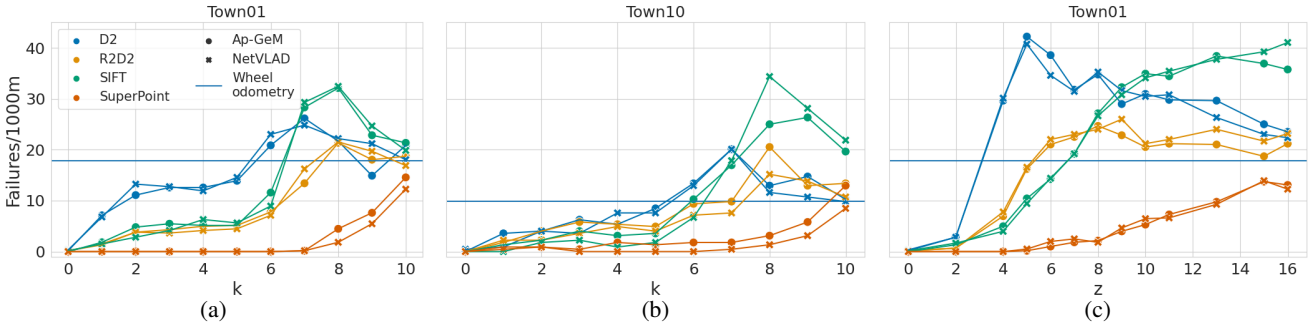


Figure 4: Relationship of failure rate with illumination (4a, 4b) and viewpoint change (4c). Marker color indicates type for local features, shape for global features.

Table 2: Navigation failure rates over 5 repeated runs of the same route in each daylight illumination level k conditions. Smaller is better. PR = place recognition method, LF = local feature type, CT = computation time (*ms*).

		Town01											Town10										
PR	LF	$k=0$	1	2	3	4	5	6	7	8	9	10	0	1	2	3	4	5	6	7	8	9	10	CT
Ap-GeM	Sift	**0.0**	1.8	4.8	5.5	5.1	5.1	11.6	28.3	32.1	22.8	21.4	**0.0**	1.3	2.2	4.0	3.1	3.6	10.3	17.0	25.0	26.3	19.6	169
	D2-net	0.2	7.1	11.1	12.6	12.6	13.9	20.9	26.2	21.7	14.9	20.9	**0.0**	3.6	4.0	6.2	5.4	8.5	13.4	20.1	12.9	14.7	10.3	165
	R2D2	0.2	1.5	3.8	4.3	5.0	5.1	7.8	13.4	21.4	18.0	18.7	**0.0**	1.8	4.0	5.8	5.4	4.9	9.4	9.8	20.5	12.9	13.4	194
	SuperPoint	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.2**	4.5	7.6	14.6	**0.0**	0.9	**0.9**	0.4	1.8	1.3	1.8	1.8	3.1	5.8	12.9	193
Net-VLAD	Sift	0.2	1.5	2.8	4.1	6.3	5.6	8.9	29.3	32.5	24.7	19.9	**0.0**	**0.0**	1.8	2.2	0.9	1.8	6.7	17.9	34.4	28.1	21.9	**134**
	D2-net	0.3	6.8	13.2	12.7	11.9	14.6	23.0	24.8	22.2	21.2	18.0	0.4	0.9	4.0	3.6	7.6	7.6	12.9	20.1	11.6	10.7	9.8	139
	R2D2	0.2	1.3	3.8	3.6	4.1	4.5	7.1	16.2	21.5	19.7	16.9	**0.0**	2.2	2.2	3.6	4.9	4.0	7.1	7.6	15.2	13.8	10.7	167
	SuperPoint	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.2**	**1.8**	**5.5**	**12.3**	**0.0**	0.4	**0.9**	**0.0**	**0.0**	**0.0**	**0.0**	**0.4**	**1.3**	**3.1**	**8.5**	166
Wheel odometry		17.9											9.8											

important findings: **1)** SuperPoint, that achieves the lowest failure rates in different illumination conditions, also achieves the lowest failure rate for a given recall rate; **2)** there is a certain operation point, determined by odometry drift, after which changes in the recall rate become meaningless for autonomous navigation. Especially the second finding is interesting as it shows that visual localization performance needs to be sufficiently good in order to improve over wheel odometry only. For Town01, the recall rate of a method at threshold T1 has to be above 60% to benefit navigation. In other words, improving recall from 40% to 50% is almost meaningless while improvement from 60% to 70% is clearly significant. This is intuitive, but not possible to observe from static datasets.

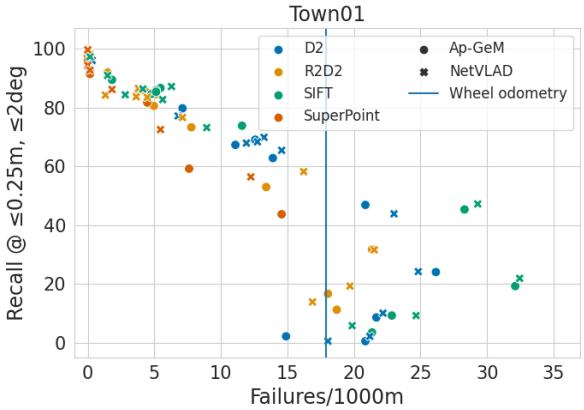


Figure 5: Relationship between the failure rate and recall rate T1. Marker color and shape indicate feature type.

Spatial distribution of navigation failures. The failure rate metric describes how often navigation failures happen, but it doesn't describe *where* the failures happen. The simulator enables visualizing the failure locations and identifying segments that are difficult for visual localization. Fig. 6 shows one example of such visualization.

Method runtimes. Table 2 also shows the average runtimes of each method, measured from Town10. These delays induce noise into the motion planning: a pose estimate is used for control after a small lag, during which the vehicle has moved a small distance from the point for which the pose estimate has been computed. At a target velocity of $4m/s$, the distances are in the range of $[0.54, 0.78]$ meters. This noise, however, is mostly in the longitudinal direction and doesn't seem to drastically affect navigation performance. During turns, the effect is more dominant.

5.2. Viewpoint and weather change

Viewpoint change The results for the viewpoint change are in Table 3, and visualized in Fig. 4c. The ranking of the methods is the same as in the previous illumination change experiment. Interestingly, the performance gap between SuperPoint and the other methods is even greater in the viewpoint experiments. At $z = 7$, none of the other methods help navigation, but even at $z = 16$ SuperPoint performance still hasn't deteriorated to level of wheel odometry. In the illumination experiments, this gap wasn't as wide. SuperPoint seems especially robust to viewpoint change, which is important for navigation applications.

Table 3: Navigation failure rates over 5 repetitions of the same route at each gallery-to-query camera pose (viewpoint) offset. z = elevation shift, θ = pitch shift.

		Town01												
PR	LF	$z=0$ $\theta=0$	2 10	4 22.5	5 27.5	6 32.5	7 35	8 37.5	9 40	10 40	11 40	13 40	15 40	16 40
Ap-GeM	Sift	**0.0**	1.3	5.0	10.4	14.2	19.2	27.2	32.3	34.9	34.4	38.4	36.9	35.8
	D2-net	0.2	2.8	29.6	42.2	38.6	31.8	34.8	29.0	31.0	29.8	29.6	25.0	23.5
	R2D2	0.2	0.7	7.0	16.2	21.0	22.5	24.7	22.8	20.5	21.2	21.0	18.7	21.2
	SuperPoint	**0.0**	**0.0**	**0.0**	**0.2**	**1.0**	**1.8**	2.2	**4.0**	**5.3**	7.3	9.8	**13.7**	13.1
Net-VLAD	Sift	0.2	1.7	4.0	9.4	14.4	19.2	26.7	30.8	34.1	35.4	37.7	39.2	41.1
	D2-net	0.3	2.8	30.1	40.7	34.6	31.5	35.3	31.6	30.5	30.8	26.3	23.0	22.4
	R2D2	0.2	0.7	7.8	16.6	22.0	23.0	24.0	26.0	21.2	22.0	24.0	21.7	23.2
	SuperPoint	**0.0**	**0.0**	**0.0**	0.5	2.0	2.5	**1.8**	4.6	6.5	**6.6**	**9.3**	13.9	**12.3**
Wheel odometry		17.9												

Table 4: The localization recall rates for the reference paths with thresholds T1 ($\leq$ 0.25m, $\leq 2°$), T2 ($\leq$0.50m,$\leq 5°$) and T3 ($\leq$5.00m, $\leq 10°$). Table with all values of k in the appendix.

	PR	LF	$k=0$ T1 / T2 / T3	2 T1 / T2 / T3	4 T1 / T2 / T3	6 T1 / T2 / T3	8 T1 / T2 / T3	10 T1 / T2 / T3
Town01	Ap-GeM	Sift	98.0 / 98.2 / 99.8	84.7 / 89.6 / 96.5	85.9 / 89.3 / 96.7	73.8 / 78.5 / 90.5	19.2 / 23.8 / 29.8	3.5 / 5.8 / 8.6
		D2-net	92.3 / 95.7 / 99.8	67.3 / 74.7 / 90.1	69.1 / 74.7 / 88.0	46.9 / 58.1 / 73.8	8.6 / 10.9 / 15.6	0.5 / 0.8 / 2.0
		R2D2	98.0 / 98.4 / **100.0**	85.5 / 90.4 / 97.7	80.6 / 88.2 / 97.0	73.3 / 79.7 / 93.6	31.7 / 37.5 / 50.5	11.2 / 12.9 / 15.0
		SuperPoint	**100.0 / 100.0 / 100.0**	**99.8 / 99.8** / 99.8	99.5 / **100.0 / 100.0**	**95.9** / 98.5 / 99.0	81.7 / 86.3 / 90.5	43.7 / 48.9 / 57.9
	Net-VLAD	Sift	97.4 / 98.5 / 99.8	84.4 / 87.7 / 97.5	87.2 / 89.8 / 97.4	73.2 / 79.6 / 92.3	21.9 / 27.0 / 34.4	5.8 / 8.2 / 13.0
		D2-net	96.1 / 96.9 / 99.7	69.9 / 75.8 / 90.1	67.9 / 75.7 / 87.3	43.8 / 52.4 / 75.0	10.0 / 13.0 / 17.9	0.5 / 1.2 / 1.8
		R2D2	98.4 / 98.5 / 99.7	86.5 / 91.0 / 97.5	86.2 / 89.5 / 97.5	76.6 / 82.4 / 93.1	31.6 / 37.7 / 49.2	13.8 / 15.3 / 18.3
		SuperPoint	**100.0 / 100.0 / 100.0**	99.7 / 99.7 / **100.0**	**99.8 / 100.0 / 100.0**	94.4 / **99.2 / 99.5**	**86.2 / 90.8 / 96.1**	**56.4 / 60.4 / 68.1**
Town10	Ap-GeM	Sift	95.2 / 96.4 / 99.6	89.5 / 90.3 / 92.7	87.1 / 87.9 / 91.5	76.6 / 79.0 / 83.9	12.5 / 16.5 / 33.5	0.0 / 0.4 / 3.6
		D2-net	94.7 / 98.4 / 99.2	84.2 / 88.7 / 91.9	81.9 / 87.1 / 92.7	46.8 / 53.6 / 65.3	0.0 / 0.0 / 1.2	0.0 / 0.0 / 0.0
		R2D2	93.1 / 94.7 / 99.6	88.3 / 90.7 / 92.3	85.5 / 87.9 / 90.3	75.8 / 78.2 / 83.5	15.7 / 21.4 / 31.5	0.4 / 0.8 / 2.8
		SuperPoint	99.6 / **100.0 / 100.0**	96.0 / 96.0 / 96.0	94.4 / 94.4 / 94.4	93.1 / 93.5 / 93.5	73.4 / 75.4 / 76.2	35.9 / 37.5 / 41.5
	Net-VLAD	Sift	96.4 / 97.2 / **100.0**	92.7 / 94.0 / 94.8	87.9 / 90.7 / 93.5	75.4 / 77.8 / 85.1	12.9 / 17.3 / 35.5	0.4 / 0.4 / 2.0
		D2-net	97.2 / 99.6 / **100.0**	86.7 / 89.5 / 94.0	83.1 / 88.3 / 92.7	45.2 / 58.5 / 69.4	0.0 / 0.0 / 0.8	0.0 / 0.0 / 0.0
		R2D2	91.6 / 92.8 / 99.6	88.7 / 91.1 / 92.7	85.9 / 89.1 / 90.7	76.6 / 80.6 / 87.1	14.1 / 20.2 / 33.9	0.4 / 0.8 / 2.0
		SuperPoint	**100.0 / 100.0 / 100.0**	**97.2 / 97.2 / 97.6**	**97.6 / 98.0 / 98.0**	**98.0 / 98.0 / 98.0**	**86.7 / 87.9 / 88.7**	**50.0 / 55.2 / 58.1**

Table 5: Failure rates at gallery-to-query weather (visibility) changes v.

		Town10			
PR	LF	$v=90$	60	30	10
Ap-GeM	Sift	8.5	25.4	25.4	21.9
	D2-net	12.1	13.8	10.3	12.9
	R2D2	10.3	12.5	10.3	10.3
	SuperPoint	**0.0**	0.4	1.8	2.2
Net-VLAD	Sift	7.6	25.9	29.5	25.9
	D2-net	12.5	13.4	10.7	9.4
	R2D2	9.4	8.0	7.6	8.5
	SuperPoint	**0.0**	**0.0**	**0.0**	**0.9**
Wheel odometry		9.8			

Figure 6: Crash locations (red) for NetVLAD + R2D2 in Town01.

Weather change. The failure rates of the weather experiments in Table 5 do not follow the same trend as the illumination and viewpoint experiments. Only SuperPoint shows correlation between the visual range and failure rate - it is also the only method to improve navigation performance at any value of v. Even the easiest condition $v = 90m$ is very difficult for the other methods. Investigating what creates robustness to this kind of gallery-to-query variation could be an interesting topic for future research.

5.3. Summary

The extensive experiments over illumination, viewpoint and weather change, and in two different maps, show that the best navigation performance is achieved by SuperPoint paired with either of the two place recognition methods, NetVLAD or Ap-GeM. SuperPoint performs well even in the tough conditions presented in Fig. 3 *(c), (f)* and *(i)* while the other methods do not improve navigation over wheel odometry at such severe gallery-to-query changes.

The in-depth analysis of the illumination experiment verifies the utility of the proposed new metric, failure rate. We believe that the proposed format of analysis proves useful to the development of navigation-oriented visual localization methods.

6. Conclusion

This paper introduced a simulator benchmark for testing and developing visual localization methods as a part of a vision-based autonomous navigation stack. To demonstrate the capabilities of the benchmark we evaluated popular visual localization methods under gallery-to-query appearance and viewpoint changes. The results show that the benchmark and the proposed navigation failure rate metric can reveal information about the visual localization methods that is not evident from the traditional static benchmarks. Substantial differences were observed in the performances of the methods, and it is evident that some are better suited to vision-based navigation than others.

In the future, the benchmark could be used for studying the performance gap between single-image and sequential visual localization, or for investigating the effect of factors such as camera placement. We hope that the research community finds the proposed benchmark useful for finding new, exciting research directions for vision-based autonomous navigation.

Acknowledgements. This research has received funding from the Technology Innovation Institute (TII) as part of the ARROWSMITH project.

References

[1] Naoki Akai, Luis Yoichi Morales, and Hiroshi Murase. Simultaneous pose and reliability estimation using convolutional neural network and Rao–Blackwellized particle filter. *Advanced Robotics*, 32(17):930–944, 2018.

[2] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir R. Zamir. On Evaluation of Embodied Navigation Agents. *arXiv:1807.06757 [cs]*, July 2018. arXiv: 1807.06757.

[3] Kiam Heong Ang, G. Chong, and Yun Li. PID control system analysis, design, and technology. *IEEE Transactions on Control Systems Technology*, 13(4):559–576, July 2005.

[4] Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1437–1451, June 2018.

[5] Hernán Badino, Daniel Huber, and Takeo Kanade. Real-time topometric localization. In *2012 IEEE International Conference on Robotics and Automation*, pages 1635–1642, May 2012. ISSN: 1050-4729.

[6] Vassileios Balntas and Dmytro Mishkin. simlocmatch-benchmark, Aug. 2021. original-date: 2021-05-15T20:52:18Z.

[7] Eric Brachmann, Martin Humenberger, Carsten Rother, and Torsten Sattler. On the Limits of Pseudo Ground Truth in Visual Camera Re-localisation. *arXiv:2109.00524 [cs]*, Sept. 2021. arXiv: 2109.00524.

[8] Eli Brosh, Matan Friedmann, Ilan Kadar, Lev Yitzhak Lavy, Elad Levi, Shmuel Rippa, Yair Lempert, Bruno Fernandez-Ruiz, Roei Herzig, and Trevor Darrell. Accurate Visual Localization for Automotive Applications. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1307–1316, Long Beach, CA, USA, June 2019. IEEE.

[9] Wolfram Burgard, Armin B. Cremers, Dieter Fox, Dirk Hähnel, Gerhard Lakemeyer, Dirk Schulz, Walter Steiner, and Sebastian Thrun. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114(1):3–55, Oct. 1999.

[10] Tim Caselitz, Bastian Steder, Michael Ruhnke, and Wolfram Burgard. Monocular camera localization in 3D LiDAR maps. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1926–1931, Daejeon, South Korea, Oct. 2016. IEEE.

[11] Luka Cehovin, Ales Leonardis, and Matej Kristan. Visual object tracking performance measures revisited. *IEEE Transactions on Image Processing*, pages 1–1, 2016.

[12] Devendra Singh Chaplot, Dhiraj Gandhi, Abhinav Gupta, and Ruslan Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, pages 4247–4258, Red Hook, NY, USA, 2020. Curran Associates Inc.

[13] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning To Explore Using Active Neural SLAM. In *Proceedings of the Eighth International Conference on Learning Representations (ICLR2020)*, Apr. 2020.

[14] Lee Clement, Mona Gridseth, Justin Tomasi, and Jonathan Kelly. Learning Matchable Image Transformations for Long-Term Metric Visual Localization. *IEEE Robotics and Automation Letters*, 5(2):1492–1499, Apr. 2020.

[15] Dominic Dall'Osto, Tobias Fischer, and Michael Milford. Fast and Robust Bio-inspired Teach and Repeat Navigation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 500–507, Sept. 2021. ISSN: 2153-0866.

[16] Tung Dang, Marco Tranzatto, Shehryar Khattak, Frank Mascarich, Kostas Alexis, and Marco Hutter. Graph-based subterranean exploration path planning using aerial and legged robots. *Journal of Field Robotics*, 37(8):1363–1388, 2020. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21993.

[17] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-Supervised Interest Point Detection and Description. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 337–33712, June 2018. ISSN: 2160-7516.

[18] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An Open Urban Driving Simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16. PMLR, Oct. 2017. ISSN: 2640-3498.

[19] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A Trainable CNN for Joint Description and Detection of Local Features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8092–8101, 2019.

[20] ETH Zurich Computer Vision Group and Microsoft Mixed Reality \& AI Lab Zurich. The ETH-Microsoft Localization Dataset, Oct. 2021. original-date: 2021-08-23T12:47:27Z.

[21] Facebook AI Research. Habitat Challenge 2021.

[22] Dario Floreano and Robert J. Wood. Science, technology and the future of small autonomous drones. *Nature*, 521(7553):460–466, May 2015.

[23] Paul Furgale and Tim Barfoot. Stereo mapping and localization for long-range path following on rough terrain. In *2010 IEEE International Conference on Robotics and Automation*, pages 4410–4416, May 2010. ISSN: 1050-4729.

[24] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):930–943, Aug. 2003.

[25] Sourav Garg and Michael Milford. SeqNet: Learning Descriptors for Sequence-Based Hierarchical Place Recognition. *IEEE Robotics and Automation Letters*, PP:1–1, 2021.

[26] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, 2 edition, 2004.

[27] Arnold Irschara, Christopher Zach, Jan-Michael Frahm, and Horst Bischof. From structure-from-motion point clouds to fast location recognition. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2599–2606, June 2009. ISSN: 1063-6919.

[28] M. Kristan, Jiri Matas, A. Leonardis, Tomás Vojír, R. Pflugfelder, G. Fernandez, G. Nebehay, F. Porikli, and Luka Cehovin. A Novel Performance Evaluation Methodology for Single-Target Trackers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[29] Zakaria Laskar, Iaroslav Melekhov, Surya Kalia, and Juho Kannala. Camera Relocalization by Computing Pairwise Relative Poses Using Convolutional Neural Network. In *2017 IEEE International Conference on Computer Vision Workshop (ICCVW)*, pages 920–929. IEEE Computer Society, Oct. 2017. ISSN: 2473-9944.

[30] Donghwan Lee, Soohyun Ryu, Suyong Yeon, Yonghan Lee, Deokhwa Kim, Cheolho Han, Yohann Cabon, Philippe Weinzaepfel, Nicolas Guerin, Gabriela Csurka, and Martin Humenberger. Large-scale Localization Datasets in Crowded Indoor Spaces. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3226–3235, Nashville, TN, USA, June 2021. IEEE.

[31] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov. 2004.

[32] Jiayi Ma, Xingyu Jiang, Aoxiang Fan, Junjun Jiang, and Junchi Yan. Image Matching from Handcrafted to Deep Features: A Survey. *International Journal of Computer Vision*, 129(1):23–79, Jan. 2021.

[33] Nicola Macoir, Jan Bauwens, Bart Jooris, Ben Van Herbruggen, Jen Rossey, Jeroen Hoebeke, and Eli De Poorter. UWB Localization with Battery-Powered Wireless Backbone for Drone-Based Inventory Management. *Sensors*, 19(3):467, Jan. 2019.

[34] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The Oxford RobotCar dataset. *International Journal of Robotics Research*, 36(1):3–15, Jan. 2017.

[35] Fabiola Maffra, Lucas Teixeira, Zetao Chen, and Margarita Chli. Real-Time Wide-Baseline Place Recognition Using Depth Completion. *IEEE Robotics and Automation Letters*, 4(2):1525–1532, Apr. 2019.

[36] Yuya Maruyama, Shinpei Kato, and Takuya Azumi. Exploring the performance of ROS2. In *Proceedings of the 13th International Conference on Embedded Software*, EMSOFT '16, pages 1–10, New York, NY, USA, Oct. 2016. Association for Computing Machinery.

[37] Carlo Masone and Barbara Caputo. A Survey on Deep Visual Place Recognition. *IEEE Access*, 9:19516–19547, 2021.

[38] Nathan Michael, Daniel Mellinger, Quentin Lindsey, and Vijay Kumar. The GRASP Multiple Micro-UAV Testbed. *IEEE Robotics Automation Magazine*, 17(3):56–65, Sept. 2010.

[39] Michael J. Milford and Gordon. F. Wyeth. SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights. In *2012 IEEE International Conference on Robotics and Automation*, pages 1643–1649, May 2012. ISSN: 1050-4729.

[40] Thomas Moore and Daniel Stouch. A Generalized Extended Kalman Filter Implementation for the Robot Operating System. In Emanuele Menegatti, Nathan Michael, Karsten Berns, and Hiroaki Yamaguchi, editors, *Intelligent Autonomous Systems 13*, Advances in Intelligent Systems and Computing, pages 335–348, Cham, 2016. Springer International Publishing.

[41] Steven D. Morad, Roberto Mecca, Rudra P. K. Poudel, Stephan Liwicki, and Roberto Cipolla. Embodied Visual Navigation With Automatic Curriculum Learning in Real Environments. *IEEE Robotics and Automation Letters*, 6(2):683–690, Apr. 2021. Conference Name: IEEE Robotics and Automation Letters.

[42] Tayyab Naseer, Luciano Spinello, Wolfram Burgard, and Cyrill Stachniss. Robust visual robot localization across seasons using network flows. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, pages 2564–2570, Québec City, Québec, Canada, 2014. AAAI Press.

[43] Helen Oleynikova, Michael Burri, Simon Lynen, and Roland Siegwart. Real-time visual-inertial localization for aerial and ground robots. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3079–3085, Sept. 2015.

[44] Michael Paton, Kirk MacTavish, Michael Warren, and Timothy D. Barfoot. Bridging the appearance gap: Multi-experience localization for long-term visual teach and repeat. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1918–1925, Oct. 2016. ISSN: 2153-0866.

[45] Andreas Pfrunder, Paulo V. K. Borges, Adrian R. Romero, Gavin Catt, and Alberto Elfes. Real-time autonomous ground vehicle navigation in heterogeneous environments using a 3D LiDAR. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2601–2608, Sept. 2017. ISSN: 2153-0866.

[46] R. Pless. Using many cameras as one. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pages II–587, June 2003. ISSN: 1063-6919.

[47] Horia Porav, Will Maddern, and Paul Newman. Adversarial Training for Adverse Conditions: Robust Metric Localisation Using Appearance Transfer. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1011–1018, May 2018. ISSN: 2577-087X.

[48] Jerome Revaud, Jon Almazan, Rafael Rezende, and Cesar De Souza. Learning With Average Precision: Training Image Retrieval With a Listwise Loss. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5106–5115, Seoul, Korea (South), Oct. 2019. IEEE.

[49] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2D2: Reliable and Repeatable Detector and Descriptor. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[50] Paul-Edouard Sarlin. hloc - the hierarchical localization toolbox, Nov. 2021. original-date: 2020-07-16T07:25:35Z.

[51] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12708–12717, Long Beach, CA, USA, June 2019. IEEE.

[52] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning Feature Matching With Graph Neural Networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4937–4946, Seattle, WA, USA, June 2020. IEEE.

[53] Paul-Edouard Sarlin, Ajaykumar Unagar, Mans Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, and Torsten Sattler. Back to the Feature: Learning Robust Camera Localization from Pixels to Pose. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3246–3256, Nashville, TN, USA, June 2021. IEEE.

[54] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomas Pajdla. Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8601–8610, June 2018. ISSN: 2575-7075.

[55] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9338–9346, Seoul, Korea (South), Oct. 2019. IEEE.

[56] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, Las Vegas, NV, USA, June 2016. IEEE.

[57] Johannes L. Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise View Selection for Unstructured Multi-View Stereo. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, Lecture Notes in Computer Science, pages 501–518, Cham, 2016. Springer International Publishing.

[58] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2930–2937, Portland, OR, USA, June 2013. IEEE.

[59] Stanford Vision and Learning Lab and Google Research. iGibson Challenge 2022.

[60] Erik Stenborg, Torsten Sattler, and Lars Hammarstrand. Using Image Sequences for Long-Term Visual Localization. In *2020 International Conference on 3D Vision (3DV)*, pages 938–948, Nov. 2020. ISSN: 2475-7888.

[61] Lukas von Stumberg, P. Wenzel, Q. Khan, and D. Cremers. GN-Net: The Gauss-Newton Loss for Multi-Weather Relocalization. *IEEE Robotics and Automation Letters*, 2020.

[62] Roni Utriainen and Markus Pöllänen. Review on mobility as a service in scientific publications. *Research in Transportation Business & Management*, 27:15–23, June 2018.

[63] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and S. Scherer. TartanAir: A Dataset to Push the Limits of Visual SLAM. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[64] Michael Warren, Melissa Greeff, Bhavit Patel, Jack Collier, Angela P. Schoellig, and Timothy D. Barfoot. There's No Place Like Home: Visual Teach and Repeat for Emergency Return of Multirotor UAVs During GPS Failure. *IEEE Robotics and Automation Letters*, 4(1):161–168, Jan. 2019.

[65] Fei Xia, William B. Shen, Chengshu Li, Priya Kasimbeg, Micael Edmond Tchapmi, Alexander Toshev, Roberto Martín-Martín, and Silvio Savarese. Interactive Gibson Benchmark: A Benchmark for Interactive Navigation in Cluttered Environments. *IEEE Robotics and Automation Letters*, 5(2):713–720, Apr. 2020. Conference Name: IEEE Robotics and Automation Letters.

[66] Xiwu Zhang, Lei Wang, and Yan Su. Visual place recognition: A survey from deep learning perspective. *Pattern Recognition*, 113:107760, May 2021.