Long-Term Visual Localization Revisited

Long-Term Visual Localization Revisited

Carl Toft[1] Will Maddern[2] Akihiko Torii[3] Lars Hammarstrand[1]
Erik Stenborg[1] Daniel Safari[3,4] Masatoshi Okutomi[3] Marc Pollefeys[5,6]
Josef Sivic[7,8] Tomas Pajdla[8] Fredrik Kahl[1] Torsten Sattler[1,8]

[1] Chalmers University of Technology
[2] Oxford Robotics Institute, University of Oxford
[3]Tokyo Institute of Technology [4]Technical University of Denmark
[5]Department of Computer Science, ETH Zürich [6]Microsoft [7]Inria [8]CIIRC, CTU in Prague

Abstract—Visual localization enables autonomous vehicles to navigate in their surroundings and augmented reality applications to link virtual to real worlds. Practical visual localization approaches need to be robust to a wide variety of viewing conditions, including day-night changes, as well as weather and seasonal variations, while providing highly accurate six degree-of-freedom (6DOF) camera pose estimates. In this paper, we extend three publicly available datasets containing images captured under a wide variety of viewing conditions, but lacking camera pose information, with ground truth pose information, making evaluation of the impact of various factors on 6DOF camera pose estimation accuracy possible. We also discuss the performance of state-of-the-art localization approaches on these datasets. Additionally, we release around half of the poses for all conditions, and keep the remaining half private as a test set, in the hopes that this will stimulate research on long-term visual localization, learned local image features, and related research areas. Our datasets are available at visuallocalization.net, where we are also hosting a benchmarking server for automatic evaluation of results on the test set. The presented state-of-the-art results are to a large degree based on submissions to our server.

Index Terms—Visual localization, relocalization, 6DOF pose estimation, benchmark, long-term localization.

1 Introduction

Estimating the 6DOF camera pose of an image with respect to a 3D scene model is key for visual navigation of autonomous vehicles and augmented/mixed reality devices. Solutions to this *visual localization* problem can also be used to "close loops" in the context of SLAM or to register images to structure-from-motion (SfM) reconstructions.

Work on 3D structure-based visual localization has focused on increasing efficiency [42], [45], [52], [69], [86], improving scalability and robustness to ambiguous structures [44], [67], [84], [96], reducing memory requirements [14], [45], [67], and more flexible scene representations [71]. All these methods utilize local features to establish 2D-3D matches. These correspondences are in turn used to estimate the camera pose. This data association stage is critical as pose estimation fails without sufficiently many correct matches. There is a well-known trade-off between discriminative power and invariance for local descriptors. Thus, existing localization approaches will only find enough matches if both the query images and the images used to construct the 3D scene model are taken under similar viewing conditions.

Capturing a scene under all viewing conditions is prohibitive. Thus, the assumption that all relevant conditions are covered is too restrictive in practice. It is more realistic to expect that images of a scene are taken under a single or a few conditions. To be practically relevant, *e.g.*, for life-long localization for self-driving cars, visual localization algorithms need to be robust under varying conditions (*cf*. Fig. 1). Yet, there has been little work in the literature that actually measures the impact of varying conditions on 6DOF pose accuracy.

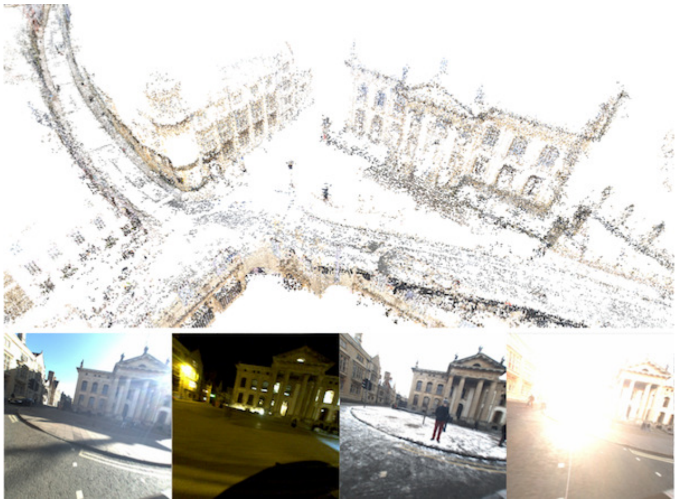


Fig. 1. Visual localization in changing urban conditions. We present three new datasets, *Aachen Day-Night*, *RobotCar Seasons* (shown) and *Extended CMU Seasons* for evaluating 6DOF localization against a prior 3D map (top) using registered query images taken from a wide variety of conditions (bottom), including day-night variation, weather, and seasonal changes over long periods of time.

One reason for this lack of work on visual localization under varying conditions was a lack of suitable benchmark datasets. The standard approach for obtaining ground truth 6DOF poses for query images is to use SfM. An SfM model containing both

- [7]*Inria, Departement dinformatique, Ecole normale superieure, CNRS, PSL Research University.*
- [8]*Czech Institute of Informatics, Robotics, and Cybernetics of the Czech Technical University in Prague.*

the database and query images is built and the resulting poses of the query images are used as ground truth [45], [71], [80]. Yet, this approach again relies on local feature matches and can only succeed if the query and database images are sufficiently similar [63]. The benchmark datasets constructed this way thus tend to only include images that are relatively easy to localize in the first place.

This paper is an extended version of our previous conference paper [70], where we presented three datasets for benchmarking localization methods in the long-term visual localization scenario. To create these datasets, we heavily relied on human work: We manually annotated matches between images captured under different conditions and verified the resulting ground truth poses. These three complimentary benchmark datasets are based on existing data [5], [54], [72]. All consist of a 3D model constructed under one condition and offer query images taken under different conditions: The *Aachen Day-Night* dataset focuses on localizing high-quality night-time images against a day-time 3D model. The *RobotCar Seasons* and *CMU Seasons* dataset both consider automotive scenarios and depict the same scene under varying seasonal and weather conditions. One challenge of the RobotCar Seasons dataset is to localize low-quality night-time images. The CMU Seasons dataset focuses on the impact of seasons on vegetation and thus the impact of scene geometry changes on localization.

In this paper, we present extended versions of the CMU Seasons and the RobotCar Seasons datasets. Since the original publication, we have also been very happy to see a large number of submissions to our evaluation server, and considerable progress has been made on the long-term localization problem and the benchmarks since then, allowing us to review the current state of the field, and to compare the best performing methods to better understand the common features that contribute to a well-performing method on this challenging problem.

Thus, in this paper we make the following contributions: *(i)* We present an outdoor benchmark complete with ground truth and metrics for evaluating 6DOF visual localization under changing conditions such as illumination (day/night), weather (sunny/rain/snow), and seasons (summer/winter). Two of the datasets presented here are extended versions of the datasets in [70]: the *Extended CMU Seasons* contains roughly 40% more images than the original CMU Seasons dataset (mostly of challenging vegetated areas). Additionally, while the original RobotCar Seasons dataset only released camera poses for one condition, we here release around half of the camera poses for all conditions. Our benchmark covers multiple scenarios, such as pedestrian and vehicle localization, and localization from single and multiple images as well as sequences. *(ii)* We provide an extensive summary of the current state-of-the-art algorithms from both the computer vision and robotics communities on our datasets, together with results from our own baseline methods, as well as a discussion that aims to provide insight into why these methods perform as they do. *(iii)* We show the value of querying with multiple images, rather than with individual photos, especially under challenging conditions. *(iv)* We have made our benchmarks publicly available at visuallocalization.net, where we are also hosting the benchmarking server for automatic evaluation of localization results on the hidden test set. We hope this will continue to stimulate research on long-term visual localization, local image feature learning, and related topics.

2 Related Work

Localization benchmarks. Tab. 1 compares our benchmark datasets with existing datasets for both visual localization and place recognition. Datasets for place recognition [17], [57], [82], [89], [90] often provide query images captured under different conditions compared to the database images. However, they neither provide 3D models nor 6DOF ground truth poses. Thus, they cannot be used to analyze the impact of changing conditions on pose estimation accuracy. In contrast, datasets for visual localization [16], [33], [37], [44], [45], [71], [72], [76], [80] often provide ground truth poses. However, they do not exhibit strong changes between query and database images due to relying on feature matching for ground truth generation. A notable exception is the Michigan North Campus Long-Term (NCLT) dataset [15], providing images captured over a long period of time and ground truth poses obtained via GPS and LIDAR-based SLAM. Yet, it does not cover all viewing conditions captured in our datasets, *e.g.*, it does not contain any images taken at night or during rain. To the best of our knowledge, ours are the first datasets providing both a wide range of changing conditions and accurate 6DOF ground truth camera poses.

Datasets such as KITTI [29], TorontoCity [93], or the Málaga Urban dataset [7] also provide street-level image sequences. Yet, they are less suitable for visual localization as only few places are visited multiple times.

2D image-based localization methods approximate the pose of a query image using the pose of the most similar photo retrieved from an image database. They are often used for place recognition [2], [17], [51], [68], [83], [89] and loop-closure detection [21], [28], [59]. They remain effective at scale [4], [68], [71], [90] and can be robust to changing conditions [2], [17], [60], [71], [83], [89]. As a baseline, we evaluate two compact VLAD-based [34] image-level representations: DenseVLAD [89] aggregates densely extracted SIFT descriptors [3], [50] while NetVLAD [2] uses learned features. Both are robust against day-night changes [2], [89] and work well at large-scale [71].

We also evaluate the de-facto standard approach for loop-closure detection in robotics, where robustness to changing conditions is critical for long-term autonomous navigation [17], [47], [57], [60], [83], [89]: FAB-MAP [21] is an image retrieval approach based on the Bag-of-Words (BoW) paradigm [78] that explicitly models the co-occurrence probability of different visual words.

3D structure-based localization methods [44], [45], [49], [67], [69], [84], [96] establish correspondences between 2D features in a query image and 3D points in an SfM point cloud via descriptor matching. These 2D-3D matches are then used to estimate the query's camera pose. Descriptor matching can be accelerated by prioritization [18], [45], [69] and efficient search algorithms [23], [52]. In large or complex scenes, descriptor matches become ambiguous due to locally similar structures found in different parts of the scene [44]. This results in high outlier ratios of up to 99%, which can be handled by exploiting co-visibility information [44], [49], [67], semantic verification [75], [87], [88] or via geometric outlier filtering [11], [25], [42], [84], [85], [96].

As baselines, we evaluate *Active Search* [69] and the *City-Scale Localization* approach [84], as representatives for efficient and scalable localization methods, respectively.

Hierarchical localization methods [30], [33], [66], [67], [72], [75] perform localization in a hierarchical fashion, combining

TABLE 1
Comparison with existing benchmarks for place recognition and visual localization. "Condition Changes" indicates that the viewing conditions of the query images and database images differ. For some datasets, images were captured from similar camera trajectories. If SfM 3D models are available, we report the number of sparse 3D points and the number of associated features. Only our datasets provide a diverse set of changing conditions, reference 3D models, and most importantly ground truth 6DOF poses for the query images.

		Image	3D SfM Model	# Images		Condition Changes			6DOF query
Dataset	Setting	Capture	(# Sub-Models)	Database	Query	Weather	Seasons	Day-Night	poses
Alderley Day/Night [57]	Suburban	Trajectory		14,607	16,960	✓		✓	
Nordland [82]	Outdoors	Trajectory		143k			✓		
Pittsburgh [90]	Urban	Trajectory		254k	24k				
SPED [17]	Outdoors	Static Webcams		1.27M	120k	✓	✓	✓	
Tokyo 24/7 [89]	Urban	Free Viewpoint		75,984	315			✓	
7 Scenes [76]	Indoor	Free Viewpoint		26,000	17,000				✓
Aachen [72]	Historic City	Free Viewpoint	1.54M / 7.28M (1)	3,047	369				
Cambridge [37]	Historic City	Free Viewpoint	1.89M / 17.68M (5)	6,848	4,081				✓(SfM)
Dubrovnik [45]	Historic City	Free Viewpoint	1.89M / 9.61M (1)	6,044	800				✓(SfM)
Landmarks [44]	Landmarks	Free Viewpoint	38.19M / 177.82M (1k)	204,626	10,000				
Mall [80]	Indoor	Free Viewpoint		682	2296				✓
NCLT [15]	Outdoors & Indoors	Trajectory		about 3.8M			✓		✓
Rome [45]	Landmarks	Free Viewpoint	4.07M / 21.52M (69)	15,179	1000				
San Francisco [16], [44], [71]	Urban	Free Viewpoint	30M / 149M (1)	610,773	442				✓(SfM)
Vienna [33]	Landmarks	Free Viewpoint	1.12M / 4.85M (3)	1,324	266				
Aachen Day-Night [70]	Historic City	Free Viewpoint	1.65M / 10.55M (1)	4,328	922			✓	✓
RobotCar Seasons (updated)	Urban	Trajectory	6.77M / 36.15M (49)	26,580	5,616	✓	✓	✓	✓
Extended CMU Seasons (new)	Suburban	Trajectory	3.37M / 17.17M (24)	60,937	56,613	✓	✓		✓

the above two approaches by using image retrieval as an initial step in a 3D-structure based approach in order to constrain the localization problem to a smaller 3D model. This typically makes the feature matching problem simpler, since it reduces the number of potentially distracting features present in the full model.

Sequence-based approaches for image retrieval are used for loop-closure detection in robotics [53], [57], [61]. Requiring a matched sequence of images in the correct order significantly reduces false positive rates compared to single-image retrieval approaches, producing impressive results including direct day-night matches with SeqSLAM [57]. We evaluate OpenSeqSLAM [82] on our benchmark.

Multiple cameras with known relative poses can be modelled as a generalized camera [62], *i.e.*, a camera with multiple centers of projection. Approaches for absolute pose estimation for both multi-camera systems [43] and camera trajectories [12] from 2D-3D matches exist. Yet, they have never been applied for localization in changing conditions. In this paper, we show that using multiple images can significantly improve performance in challenging scenarios.

Learning-based localization methods have been proposed to solve both loop-closure detection [17], [56], [81], [83] and pose estimation [19], [37], [92]. They learn features with stable appearance over time [17], [22], [24], [58], [60], [64], [74], [91], train classifiers for place recognition [13], [31], [36], [39], [47], [94], and train CNNs to regress 2D-3D matches [9], [10], [76] or camera poses [19], [37], [92]. In this paper, we evaluate approaches based on learned robust local features [22], [24], [64], which constitute the state-of-the-art on our benchmarks.

3 Benchmark Datasets for 6DOF Localization

This section describes the creation of our three new benchmark datasets. Each dataset is constructed from publicly available data, allowing our benchmarks to cover multiple geographic locations. We add ground truth poses for all query images and build reference 3D models (*cf*. Fig. 3) from images captured under a single reference condition. Note that database images with known pose for other conditions are provided as well for the RobotCar Seasons and Extended CMU Seasons datasets, but these do not overlap with any of the areas containing query images.

All three datasets present different challenges. The *Aachen Day-Night* dataset focuses on localizing night-time photos against a 3D model built from day-time imagery. The night-time images, taken with a mobile phone using HDR post-processing, are of high quality. The dataset represents a scenario where images are taken with hand-held cameras, *e.g.*, an augmented reality application.

Both the *RobotCar Seasons* and the *Extended CMU Seasons* datasets represent automotive scenarios, with images captured from a car. In contrast to the Aachen Day-Night dataset, both datasets exhibit less variability in viewpoints but a larger variance in viewing conditions. The night-time images from the RobotCar dataset were taken from a driving car with a consumer camera with auto-exposure. This results in significantly less well-lit images exhibiting motion blur, *i.e.*, images that are significantly harder to localize (*cf*. Fig. 2).

The RobotCar dataset depicts a mostly urban scene with rather static scene geometry. In contrast, the CMU dataset contains a significant amount of vegetation. The changing appearance and geometry of the vegetation, due to seasonal changes, is the main challenge of this dataset.

3.1 The Aachen Day-Night Dataset

Our Aachen Day-Night dataset is based on the Aachen localization dataset from [72]. The original dataset contains 4,479 reference and 369 query images taken in the old inner city of Aachen, Germany. It provides a 3D SfM model but does not have ground truth poses for the queries. We augmented the original dataset with day- and night-time queries captured using standard consumer phone cameras.

To obtain ground truth poses for the day-time queries, we used COLMAP [73] to create an intermediate 3D model from the reference and day-time query images. The scale of the reconstruction is recovered by aligning it with the geo-registered original Aachen model. As in [45], we obtain the reference model for the Aachen Day-Night dataset by removing the day-time query images. 3D points visible in only a single remaining camera were removed as well [45]. The resulting 3D model has 4,328 reference images and 1.65M 3D points triangulated from 10.55M features.

TABLE 2
Detailed statistics for the three benchmark datasets proposed in this paper. For each dataset, a reference 3D model was constructed using images taken under the same reference condition, *e.g.*, "overcast" for the RobotCar Seasons dataset. The column training images refers to additional images whose ground truth poses are provided, which were captured under different conditions from the reference condition.

	reference model				query images	training
	# images	# 3D points	# features	condition	conditions (# images)	images
Aachen Day-Night	4,328	1.65M	10.55M	day	day (824), night (98)	-/-
RobotCar Seasons	20,862	6.77M	36.15M	overcast (November)	dawn (681), dusk (591), night (678), night+rain (609), rain (615), overcast summer / winter (633 / 492), snow (645), sun (672)	5,718
Extended CMU Seasons	10,338	3.37M	17.17M	sun / no foliage (April)	sun (16,366), low sun (21,715), overcast (8,220), clouds (10,312), foliage (24,984), mixed foliage (20,932), no foliage (10,697) urban (18,373), suburban (21,599), park (16,641)	50,599

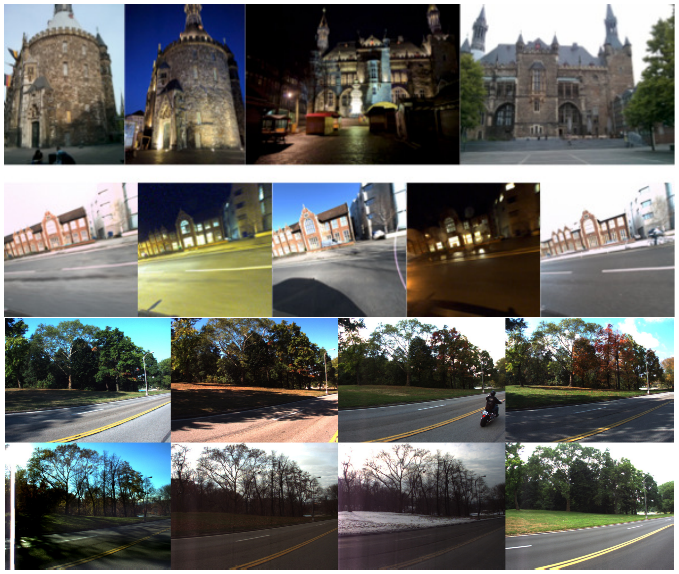


Fig. 2. Example query images for *Aachen Day-Night* (top), *RobotCar Seasons* (middle) and the *Extended CMU Seasons* (bottom) datasets.

Ground truth for night-time queries. We captured 98 night-time query images using a Google Nexus5X phone with software HDR enabled. Attempts to include them in the intermediate model resulted in highly inaccurate camera poses due to a lack of sufficient feature matches. To obtain ground truth poses for the night-time queries, we thus hand-labelled 2D-3D matches. We manually selected a day-time query image taken from a similar viewpoint for each night-time query. For each selected day-time query, we projected its visible 3D points from the intermediate model into it. Given these projections as reference, we manually labelled 10 to 30 corresponding pixel positions in the night-time query. Using the resulting 2D-3D matches and the known intrinsics of the camera, we estimate the camera poses using a 3-point solver [27], [38] and non-linear pose refinement.

To estimate the accuracy for these poses, we measure the mean reprojection error of our hand-labelled 2D-3D correspondences (4.33 pixels for 1600x1200 pixel images) and the pose uncertainty. For the latter, we compute multiple poses from a subset of the matches for each image and measure the difference in these poses to our ground truth poses. The mean median position and orientation errors are 36cm and 1°. The absolute pose accuracy that can be achieved by minimizing a reprojection error depends on the distance of the camera to the scene. Given that the images were typically taken 15 or more meters from the scene, we consider the ground truth poses to be reasonably accurate.

3.2 The RobotCar Seasons Dataset

Our RobotCar Seasons dataset is based on a subset of the publicly available Oxford RobotCar Dataset [54]. The original dataset contains over 20M images recorded from an autonomous vehicle platform over 12 months in Oxford, UK. Out of the 100 available traversals of the 10km route, we select one reference traversal in overcast conditions and nine query traversals that cover a wide range of conditions (*cf*. Tab. 2). All selected images were taken with the three synchronized global shutter Point Grey Grasshopper2 cameras mounted to the left, rear, and right of the car. Both the intrinsics of the cameras and their relative poses are known.

The reference traversal contains 26,121 images taken at 8,707 positions, with 1m between successive positions. Building a single consistent 3D model from this data is very challenging, both due to sheer size and the lack of visual overlap between the three cameras. We thus built 49 non-overlapping local submaps, each covering a 100m trajectory. For each submap, we initialized the database camera poses using vehicle positions reported by the inertial navigation system (INS) mounted on the RobotCar. We then iteratively triangulated 3D points, merged tracks, and refined both structure and poses using bundle adjustment. The scale of the reconstructions was recovered by registering them against the INS poses. The reference model contains all submaps and consists of 20,862 reference images and 6.77M 3D points triangulated from 36.15M features.

We obtained images from the other traversals by selecting reference positions inside the 49 submaps and gathering all images from the nine other traversals with INS poses within 10m of one of these positions. This resulted in 11,934 images in total, where triplets of images were captured at 3,978 distinct locations. These images were grouped into 460 temporal sequences based on the timestamps of the images.

Compared to [70], we have now, in addition to the images in the reference traversal, also publicly released around half of the camera poses for the images from the other traversals (and thus captured during different conditions), for a total of 26,580 images. The camera poses of the remaining 5,616 images are used as a hidden test set in the long-term visual localization benchmark for the RobotCar Seasons dataset.

Ground truth poses for the queries. Due to GPS drift, the INS poses for these other traversals cannot be directly used as ground truth. Again, there are not enough feature matches between day- and night-time images for SfM. We thus used the LIDAR scanners mounted to the vehicle to build local 3D point clouds for each of the 49 submaps under each condition. These models were then aligned to the LIDAR point clouds of the reference trajectory using ICP [6]. The camera trajectory is then obtained from the known relative pose between the camera and LIDAR. Many alignments needed to be manually adjusted to account for changes in scene

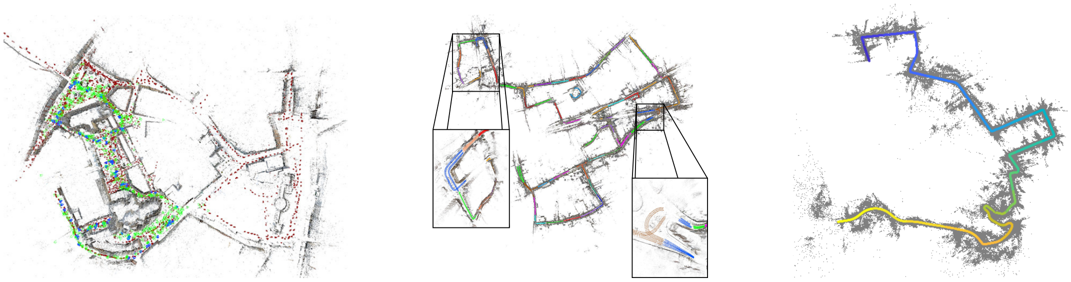


Fig. 3. 3D models of the *Aachen Day-Night* (left, showing database (red), day-time query (green), and night-time query images (blue)), *RobotCar Seasons* (middle), and *Extended CMU Seasons* (right) datasets. For RobotCar and CMU, the colors encode the individual submaps.

structure over time (often due to building construction and road layout changes). The final median RMS errors between aligned point clouds was under 0.10m in translation and 0.5° in rotation across all locations. The alignments provided ground truth poses for these images.

3.3 The Extended CMU Seasons Dataset

The Extended CMU Seasons Dataset is based on a subset of the CMU Visual Localization Dataset [5], which contains more than 100K images recorded by the Computer Vision Group at Carnegie Mellon University over a period of 12 months in Pittsburgh, PA, USA. The images were collected using a rig of two cameras mounted at approximately 45 degrees forward/left and forward/right angles on the roof of an SUV. The vehicle traversed an 8.5 km long route through central and suburban Pittsburgh 16 times with a spacing in time of between 2 weeks up to 2 months. Out of the 16 traversals, we selected the one from April 4 as the reference, and then 11 query traversals were selected such that they cover the range of variations in seasons and weather that the dataset contains.

As with the RobotCar Seasons dataset, we publicly release all images and corresponding ground truth poses for the reference traversal, in addition to around half of the ground truth poses from the other traversals, for a total of 60,937 images. The remaining half remain private as a test set for benchmarking purposes. Compared to the original CMU Seasons dataset from [70], the Extended CMU Seasons dataset is considerably larger: it contains around 42% more images, and the number of publicly available camera poses has been increased by a factor 8.5. More importantly, the publicly available poses now contain images taken under a wide range of environmental conditions.

Ground truth poses for the queries. As with the RobotCar dataset, the GPS is not accurate enough and the CMU dataset is also too large to build one 3D model from all the images. The full sequences were split up into 24 shorter sequences, each containing about 250 consecutive vehicle poses. For each short sequence, a 3D model was built using bundle adjustment of SIFT points tracked over several image frames using the system in [26]. The resulting submaps of the reference route were merged with the corresponding submaps from the other traversals by using global bundle adjustment and manually annotating image correspondences across sequences collected during different dates. Reprojection errors are within a few pixels for all 3D points and the distances between estimated camera positions and expected ones (based on neighbouring cameras) are under 0.10m. The resulting reference model consists of 3.37M 3D points triangulated from 17.17M features in 10,338 database images.

4 Benchmark Setup

The datasets are available for download from the benchmark website, where we are also hosting an evaluation server. To evaluate a method, a file containing the estimated 6 DoF poses for images in the test set is uploaded to the server, and the localization results are automatically computed, reported and ranked in the leaderboard. Since the original publication of this paper, many new submissions have arrived, and considerable progress has been made on the benchmarks. Below we present the evaluation measures used to benchmark performance, and in Sec. 6, results for the current top-performing methods are presented and discussed.

Evaluation measures. We measure the *pose accuracy* of a method by the deviation between the estimated and the ground truth pose. The *position error* is measured as the Euclidean distance $\|c_\text{est} - c_\text{gt}\|_2$ between the estimated c_est and the ground truth position c_gt. The absolute *orientation error* $|\alpha|$, measured as an angle in degrees, is computed from the estimated and ground truth camera rotation matrices $\mathtt{R}_\text{est}$ and $\mathtt{R}_\text{gt}$. We follow standard practice [32] and compute $|\alpha|$ as $2\cos(|\alpha|) = \text{trace}(\mathtt{R}_\text{gt}^{-1}\mathtt{R}_\text{est}) - 1$, *i.e.*, we measure the minimum rotation angle required to align both rotations [32].

We measure the percentage of query images localized within Xm and $Y°$ of their ground truth pose. We define three pose accuracy intervals by varying the thresholds: *High-precision* (0.25m, 2°), *medium-precision* (0.5m, 5°), and *coarse-precision* (5m, 10°). These thresholds were chosen to reflect the high accuracy required for autonomous driving. We use the intervals (0.5m, 2°), (1m, 5°), (5m, 10°) for the Aachen night-time queries to account for the higher uncertainty in our ground truth poses. Still, all regimes are more accurate than consumer-grade GPS systems.

Multi-camera methods and optimistic baselines. In order to measure the benefit of using multiple images for pose estimation, we have evaluated three approaches: *OpenSeqSLAM* [82] is based on image retrieval and enforces that the images in the sequence are matched in correct order. Knowing the relative poses between the query images, we can model them as a generalized camera [62]. Given 2D-3D matches per individual image (estimated via Active Search), we estimate the pose via a generalized absolute camera pose solver [43] inside a RANSAC loop. We denote this approach as *Active Search+GC* (AS+GC). We mostly use ground truth query poses to compute the relative poses that define the generalized cameras[1]. Thus, AS+GC provides an upper bound on the number of images that can be localized when querying with generalized cameras.

[1] Note that Active Search+GC only uses the relative poses between the query images to define the geometry of a generalized camera. It does *not* use any information about the absolute poses of the query images.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2020.3032010, IEEE Transactions on Pattern Analysis and Machine Intelligence

Additionally, we evaluate PFSL [79], a particle-filter based approach that performs localization by reasoning solely using semantic information. See Sec. 5 for details.

In order to measure how hard our datasets are, we also implemented two *optimistic baselines*. Both assume that a set of relevant database images is known for each query. Both perform pairwise image matching and use the known ground truth poses for the reference images to triangulate the scene structure. The feature matches between the query and reference images and the known intrinsic calibration are then used to estimate the query pose. The first optimistic baseline, *LocalSfM*, uses upright RootSIFT features [3], [50]. The second uses upright CNN features densely extracted on a regular grid. We use the same VGG-16 network [77] as NetVLAD. The *DenseSfM* method uses coarse-to-fine matching with conv4 and conv3 features.

We select the relevant reference images for the two baselines as follows: For Aachen, we use the manually selected day-time image (*cf*. Sec. 3.1) to select up to 20 reference images sharing the most 3D points with the selected day-time photo. For RobotCar and CMU, we use all reference images within 5m and 135° of the ground truth query pose.

We also evaluated *PoseNet* [37] but were not able to obtain competitive results. We also attempted to train *DSAC* [9] on KITTI but were not able to train it. Both PoseNet and DSAC were thus excluded from further evaluations.

5 Details on the Evaluated Algorithms

In this section we provide brief descriptions of each of the top performing methods, so that we can try to reason about what it is that makes a method perform well in the long-term visual localization scenario. Many methods are combinations of image retrieval methods and structure-based localization methods, so we start this section by briefly introducing some of the more common methods used as building blocks, and end with the composite approaches that combine these building blocks, referred to here as hierarchical approaches. In this way we may more clearly see what each individual method brings to the table.

5.1 2D Image-based Localization

By 2D image-based localization methods, or image retrieval methods, we here mean methods which do not employ any kind of 3D reasoning when computing the pose of the query image. Methods which fall into this category typically pose the localization problem as an image search problem. Given the query image, together with a set of database images whose camera pose is fully known (perhaps from GPS data or from a structure-from-motion reconstruction), the pose of the query image is typically approximated as the pose of the visually most similar image in the database set. Typically, the most similar image is found by computing a single global image descriptor for the query image and all database images, and then finding the nearest neighbour to the query descriptor in the set of database descriptors.

DenseVLAD. DenseVLAD [89] computes the global image descriptor by extracting SIFT features on a dense, regular grid on the image at different scales. This is done instead of extracting features at, for example, difference-of-Gaussian extrema, since it has been noted that the repeatability of feature detectors deteriorates with increasing viewpoint and lighting changes. The extracted descriptors are then clustered based on a vector of locally aggregated decriptors (VLAD) approach [35].

NetVLAD. NetVLAD [2] is a CNN which uses a differentiable VLAD layer to encode an image into a global image descriptor. The VLAD layer is placed at the final feature map of a CNN made for image classification, such as AlexNet or VGG 16, and performs pooling of the features in that layer. The network has been trained on Google Time Machine data in a weakly supervised manner based on a triplet ranking loss.

ToDayGAN. ToDayGAN [1] is a style-transfer network that performs night-to-day image translation for the purpose of image retrieval-based visual localization. Night-time images are translated to daytime images, followed by extraction of a DenseVLAD descriptor.

Localizing Visual Landmarks for Place Recognition. This learning based image retrieval method [95] is based on the idea that for the visual place recognition problem, not all parts of an image are useful for describing the image. Thus a *landmark localization network* which outputs dense local features as well as a heatmap showing the saliency (or distinctiveness) of each local feature is trained in a weakly supervised manner using only image level annotations (i.e. no pixel-level annotations of saliency). The similarity of two images may then be computed by matching features between the images, and summing the cosine-similarity of all matches, together with a weight depending on the spatial distribution of the matches.

5.2 Structure-based approaches

Unlike image-based localization, structure-based approaches employ a 3D representation constructed from the database images to perform visual localization. Typically, an SfM model is built from the database images, and the camera pose of the query image is computed by first establishing 2D-3D correspondences between the query image and the point cloud, and then performing camera pose estimation using these correspondences. These methods typically differ in which type of local image feature is used to build the model and establish matches, which matching strategy is used to establish matches, how the pose is computed from the matches, if outlier rejection is performed on the matches, and so on.

Active Search. Active Search [69] is a feature matching strategy which uses a quantization of the descriptor space in order to accelerate feature matching. For each feature in the image, matching is only performed to 3D points assigned to the same visual word [78], and image features are matched in ascending order, in the sense that image features assigned to a visual word with the fewest 3D points assigned to it, are matched first. When a match is found, the vicinity of that 3D point is examined for possible matches back into the image.

CityScaleLocalization. CityScaleLocalization [84] is a localization approach that employs an outlier rejection strategy to reject correspondences which cannot be a part of the optimal camera pose. The method is based on knowledge of the vertical direction, as well as an approximate height of the camera above ground. Under these circumstances, if one correspondence is assumed to be an inlier, an upper bound to the number of inliers for the best camera pose can be computed. If this is below the number of inliers found in the best pose candidate found so far, this correspondence can be permanently removed from consideration without affecting the solution.

Semantic Match Consistency. The Semantic Match Consistency approach [88] is a soft outlier rejection method, similar to the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2020.3032010, IEEE Transactions on Pattern Analysis and Machine Intelligence

CityScaleLocalization method but which incorporates scene semantics. Given knowledge about the gravity direction and camera height, assuming a 2D-3D correspondence is correct constrains the camera center to lie on a circle. This circle can then be traversed and the 3D structure projected down into the hypothesized camera centers along the circle. The consistency between the semantic labelling of the projected structure, and the actually observed labelling of the query image yields a soft inlier-outlier likelihood score for this correspondence.

Match Consistency with Fine-Grained Segmentations. The fine-grained segmentation match consistency method [41] is a structure-based method that computes the camera pose using PnP RANSAC on 2D-3D correspondences established by matching SIFT features, with an additional outlier filtering step based on semantics. Specifically, all 2D-3D matches whose 2D point has a semantic label which disagrees with its corresponding 3D point are discarded immediately after matching. This can be done using e.g. Cityscapes classes [20], or more fine-grained semantic classes learned in a self-supervised manner [41].

DGCNCCC. The Dense Geometric Correspondence Network (DGC-Net) [55] is a network for estimating dense correspondences between images. It takes as input two images and outputs the deformation field that maps one to the other. It is trained on both a synthetic dataset containing homographically warped images, as well as a real dataset containing true correspondences obtained via dense reconstruction.

5.3 Learned local image features

SuperPoint. SuperPoint [22] is a CNN which takes an image as an input and outputs local image features. The network consists of one shared encoder, and two decoder heads: one outputs a heatmap of detection scores, while the other outputs dense descriptors over the image. The network is trained in a self-supervised manner, where ground truth detections are generated for an unlabelled image by applying a detector, trained on synthetic data, to several warped versions of the image, and then warping back all detections into the original image.

D2-Net. D2-Net [24] is a learned feature detector and descriptor. It is a CNN which takes an image as input, and outputs a set of feature detections over the image, as well as a descriptor for each pixel in the image. The network consists of a single CNN which performs detection and description jointly; the descriptors correspond to the depth dimension of the tensor output by the network, and the detections correspond to those pixels which have a local maximum in one of their feature maps. The network is trained in a supervised manner on correspondences obtained from the MegaDepth [46] dataset.

5.4 Hierarchical Methods

The image retrieval and structure-based methods have different strengths and weaknesses, and stronger methods can of course be created by combining these two approaches. Under this section we have collected the methods that explicitly start with an image retrieval step to reduce the search space for a structure-based method.

Hierarchichal Localization. The Coarse-to-fine hierarchical localization method [66] combines image retrieval and structure-based approaches by first performing image retrieval to find a set of database images which are likely to be close to the query image. These shortlisted images are then clustered into a set of distinct places based on their covisibility graphs. For each place, a local 3D model is extracted from the full map, and localization is performed independently to each local model, keeping the best pose. Further, the local and global descriptors are distilled into a MobileNet [65] based CNN architecture, allowing efficient inference on mobile devices.

Visual Localization Using a Sparse Semantic 3D Map. This method [75] combines NetVLAD with structure-based pose estimation using SIFT features. Specifically, NetVLAD is used to retrieve a shortlist of database images. Local image features are then matched between the query image and each of the k retrieved images. For each of the k images, camera pose estimation is performed using PnP RANSAC, with the modification that the quality of the pose is based on the consistency of the semantic reprojection, as opposed to the traditional inlier counting. This gives a score for each database image. Lastly, all 2D-3D matches are pooled together, and PnP is performed to solve for the pose, where the scores for each database image is used to bias RANSAC's sampling to prefer correspondences from database images with higher semantic consistency score.

Asymmetric Hypercolumn Matching. Like the above two methods, this method [30] starts with an image retrieval step using NetVLAD descriptors to obtain a shortlist of images from the database that may contain visual overlap with the query image. Sparse-to-dense matching is then performed between the sparse features found in the database image (extracted using SuperPoint detections, and Hypercolumn features, i.e., features extracted from the feature layers of the VGG backbone of the NetVLAD network), and dense features extracted from the query image. In other words, each sparse feature in the retrieved database images (with known 3D position) is matched exhaustively to the dense features of the query. This bypasses the problem of non-repeatability of feature detectors during changing conditions [97].

5.5 Sequential and Multi-Camera Methods

By sequential methods, we refer to methods that use more than a single query image during pose estimation. This provides additional information, especially for the Extended CMU Seasons dataset, which consists of a sequence of timestamped images captured from a synchronized stereo rig. More accurate poses can here be obtained by exploiting the temporal consistency, making the results of these methods not directly comparable to methods which only utilize a single image.

PFSL. The PFSL method [79] is a particle filter-based semantic localization method, which performs filtering on a sequence of timestamped images and corresponding IMU data. It is only evaluated on the Extended CMU Seasons dataset, and uses a motion model based on simulated IMU data generated from (noisy versions of) the ground truth camera poses. The likelihood function for a given pose is based on how well the reprojection of a semantically labelled 3D point cloud matches the observed semantic labelling in the current query image. It is worth noting that explicit 2D-3D matching of local features is completely absent from this pipeline, and the method is thus not dependent on the invariance of local descriptors towards environmental changes to perform well.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2020.3032010, IEEE Transactions on Pattern Analysis and Machine Intelligence

SeqSLAM. SeqSLAM [57] is an image retrieval based approach that matches temporally coherent sequences of images using a similarity measure based on the correlation of the intensities in the images. The presented result are from the OpenSeqSLAM implementation from [82] with default parameters for template learning and trajectory uniqueness.

5.6 Optimistic Baselines

As explained in Sec. 5 of the paper, we implemented two *optimistic baselines*, which are provided some information about the ground truth camera pose for each image. Specifically, for each query image, we provide a small set of reference images depicting the same part of the model. The remaining problem is to establish sufficiently many correspondences between the query and the selected reference images to facilitate camera pose estimation. Thus, both approaches measure an upper bound on the pose quality that can be achieved with a given type of local feature.

LocalSfM. Given a query image and its relevant set of reference images, LocalSfM first extracts upright RootSIFT [3], [50] features. Next, LocalSfM performs exhaustive feature matching between the relevant reference images as well as between the query and the relevant reference images. While Active Search uses Lowe's ratio test[2], LocalSfM neither uses the ratio test nor a threshold on the maximum descriptor distance. Instead, it only requires matching features to be mutual nearest neighbors. Given the known poses and intrinsics for the reference images, LocalSfM triangulates the 3D structure of the scene using the previously established 2D-2D matches. Notice that the resulting 3D model is automatically constructed in the global coordinate system of the reference 3D model. Finally, we use the known intrinsics of the query image and the feature matches between the query and the reference images to estimate the camera pose of the query.

For each query image, the relevant set of reference images is selected as follows: For the RobotCar Seasons dataset, we use the ground truth pose of each query image to identify a relevant set of reference images. More precisely, we select all reference images whose camera centers are within 5m of the ground truth position of the query and whose orientations are within 135° of the orientation of the query image.

As explained in Sec. 3.2 of the paper, we manually select a day-time query image taken from a similar viewpoint for each night-time query photo in the Aachen Day-Night dataset. The day-time queries were included when constructing the intermediate model. Thus, their ground truth poses as well as a set of 3D points visible in each of them are obtained from the intermediate structure-from-motion model. For each day-time query, we select up to 20 reference images that observe the largest number of the 3D points visible in the day-time query. These reference images then form the set of relevant images for the corresponding night-time query photo.

LocalSfM is implemented using COLMAP [73]. It is rather straight-forward to replace upright RootSIFT features with other types of local features. Our implementaton is publicly available[3].

DenseSfM. DenseSfM modifies the LocalSfM approach by replacing RootSIFT [3] features extracted at DoG extrema [50] with features densely extracted from a regular grid [8], [48]. The goal of this approach is to increase the robustness of feature matching between day- and night-time images [89], [97]. We used convolutional layers (conv4 and conv3) from a VGG-16 network [77], which was pre-trained as part of the NetVLAD model (Pitts30k), as features. We generated tentative correspondences by matching the extracted features in a coarse-to-fine manner: We first match conv4 features and use the resulting matches to restrict the correspondence search for conv3 features. As for LocalSfM, we performed exhaustive pairwise image matching. The matches are verified by estimating up to two homographies between each image pair via RANSAC [27]. The resulting verified feature matches are then used as input for COLMAP [73]. The reconstruction process is the same as for LocalSfM, *i.e.*, we first triangulate the 3D points and then use them to estimate the pose of the night-time query. DenseSfM uses the same set of reference images for each query photo as LocalSfM.

6 Experimental Evaluation

This section presents the results of the current top performing methods for each of the three datasets, as well as a discussion where we attempt to identify the general strategies and pipelines that seem to perform well in the long-term localization scenario. In the benchmark, we have focused on pose accuracy, as described in Sec. 4. Only submitted methods with an accompanying article and methods that were presented at the *Long-Term Visual Localization under Changing Conditions* workshop during the Computer Vision and Pattern Recognition (CVPR) conference in 2019 are included.

The results on the three datasets are shown in Tab. 3, 4, and 7. Fig. 4 shows a summary of these results for a subset of the methods, in order to easier get an overview of the performance of the different methods. Note that the methods have been grouped into four different categories in order to get a better overview of the results. This categorization is based on the rough taxonomy of visual localization methods which was introduced in Sec. 5. These groupings correspond to *sequential methods*, *hierarchical methods*, *structure-based methods*, and *image-retrieval based* methods. Within each group, the methods are ranked in order of decreasing performance, with the best-performing method at the top. Grouping together the methods in this manner may of course change the global ordering of the methods, but it turned out that on these datasets these groupings are fairly natural, in the sense that methods in the same group tend to exhibit similar performance.

6.1 Evaluation on the Aachen Day-Night Dataset

The focus of the Aachen Day-Night dataset is on benchmarking the pose accuracy obtained by state-of-the-art methods when localizing night-time queries against a 3D model constructed from day-time imagery. In order to put the results obtained for the night-time queries into context, the methods are also evaluated on the 824 day-time queries. As shown in Tab. 3, the best-performing hierarchical and structure-based methods generally succeed in localizing most of the day-time query images, even in the high-precision regime. We conclude that the Aachen dataset is not particularly challenging for the day-time query images.

Night-time queries. Tab. 3 also reports the results obtained for the night-time queries. For the structure-based methods, we observe a significant drop in accuracy over all precision regimes when localizing night-time images. Given that the night-time queries were taken from similar viewpoints as the day-time queries, this drop is solely caused by the day-night change.

[2] Active Search uses a ratio test threshold of 0.7 for 2D-to-3D and a threshold of 0.6 for 3D-to-2D matching.

[3] https://github.com/tsattler/visuallocalizationbenchmark

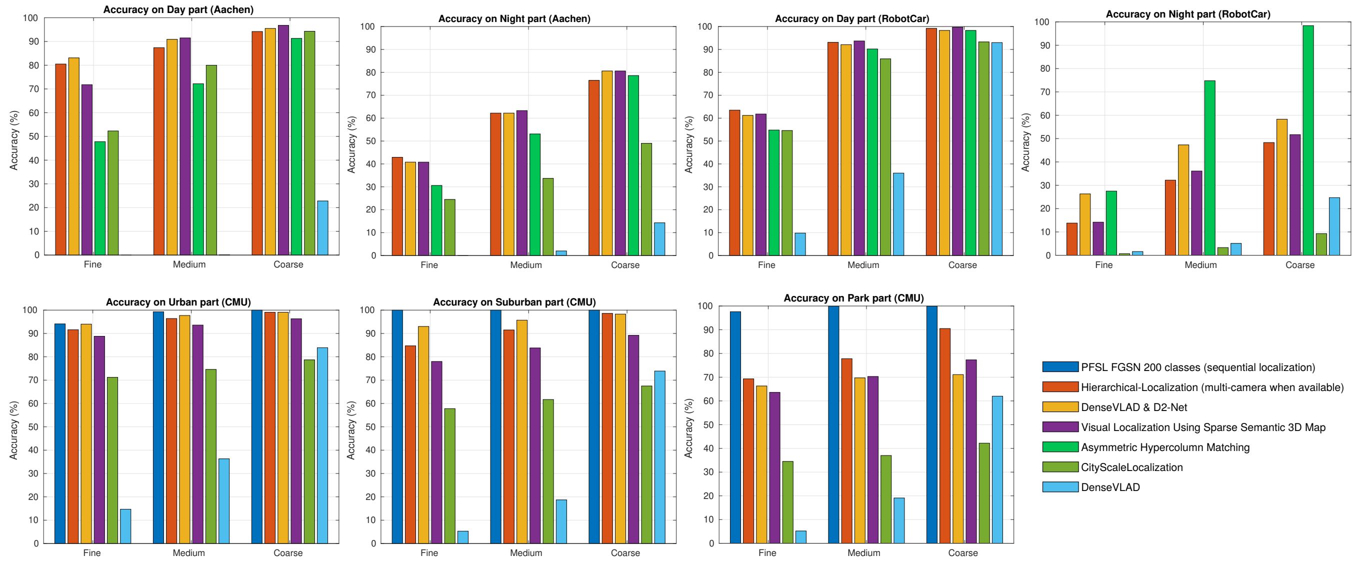


Fig. 4. Performance on the three datasets of some of the top performing methods. Results are shown for the three precision regimes fine, medium and coarse (see the individual tables for details on these). The CityScaleLocalization and DenseVLAD methods are included in the figures as well, as representatives of the structure-based and image-retrieval based methods, respectively.

TABLE 3
Performance of state-of-the-art methods on the Aachen Day-Night dataset.

	day	**night**
m	.25 / .50 / 5.0	.5 / .1 / 5.0
deg	2 / 5 / 10	2 / 5 / 10
Optimistic baselines		
DenseSfM		39.8 / 60.2 / 84.7
LocalSfM		36.7 / 54.1 / 72.4
Hierarchical methods		
NetVLAD & D2-Net - multi-scale [2], [24]	84.8 / 92.6 / 97.5	43.9 / 66.3 / 85.7
Vis. Loc. Using Sparse Semantic 3D Map [75]	71.8 / 91.5 / 96.8	40.8 / 63.3 / 80.6
DenseVLAD & D2-Net - multi-scale [24], [89]	83.1 / 90.9 / 95.5	40.8 / 62.2 / 80.6
Hierarchical-Loc. NetVLAD, SuperPoint [66]	80.5 / 87.4 / 94.2	42.9 / 62.2 / 76.5
Hierarchical-Loc. (multi-cam) [66]	80.5 / 87.4 / 94.2	42.9 / 62.2 / 76.5
Asymmetric Hypercolumn Matching [30]	47.8 / 72.2 / 91.3	30.6 / 53.1 / 78.6
Structure-based methods		
Active Search v1.1	85.3 / 92.2 / 97.9	27.6 / 38.8 / 56.1
CityScaleLocalization [84]	52.3 / 80.0 / 94.3	24.5 / 33.7 / 49.0
DGCNCCC [55]	22.9 / 49.8 / 84.7	19.4 / 37.8 / 68.4
Image-retrieval based methods		
Loc. Vis. Landmarks for 2D matching [95]	62.4 / 71.8 / 79.9	24.5 / 35.7 / 44.9
DenseVLAD [89]	0.0 / 0.1 / 22.8	0.0 / 2.0 / 14.3
NetVLAD [2]	0.0 / 0.2 / 18.9	0.0 / 2.0 / 12.2
Loc. Vis. Landmarks for Place Recognition [95]	0.0 / 0.2 / 20.8	0.0 / 1.0 / 10.2
FAB-MAP [21]	0.0 / 0.0 / 4.6	0.0 / 0.0 / 0.0

It is interesting to note that while the hierarchical methods experience a large drop for night-time images in the high-precision regime, the drop in accuracy for the coarse-precision regime is much less severe. This seems to indicate that the reduced search space caused by the initial image retrieval step significantly helps the feature matching problem by restricting the 2D-3D correspondence search to a much smaller space, reducing the impact of visually similar false positives.

However, the performance of the image retrieval methods alone, using no structure-based spatial reasoning to aid in the ranking, is quite poor. Note that these methods simply output the pose of the top-retrieved database image, so the poor performance in the high-precision regime should not be surprising. This dataset is a challenging one for pure image retrieval method due to the viewpoint differences between the database and query images, making it quite unlikely that the top retrieved database image should lie very close to the true pose of the query image.

If more than one database image is retrieved, the known 3D structure of the scene may be used to re-rank the candidates. There is thus a clear synergistic effect between the image retrieval and structure-based methods: the structure-based methods can be used to exclude visually similar but geometrically incompatible candidates, and the top retrieved candidates may then in turn be used to reduce the search space for the feature matching when establishing correspondences to be used for camera pose estimation. This effect is evident in the good performance of the hierarchical methods.

LocalSfM, despite knowing relevant reference images for each query, completely fails to localize about 20% of all queries. This is caused by a lack of correct feature matches for these queries, either due to failures of the feature detector or descriptor. DenseSfM skips feature detection and directly matches densely extracted CNN descriptors (which encode higher-level information compared to the gradient histograms used by RootSIFT). This enables DenseSfM to localize more images at a higher accuracy. Still, there is significant room for improvement, even in the coarse-precision regime (*cf*. Tab. 3). Also, extracting and matching dense descriptors is a time-consuming task.

6.2 Evaluation on the RobotCar Seasons Dataset

The focus of the RobotCar Seasons dataset is to measure the impact of different seasons and illumination conditions on pose estimation accuracy in an urban environment.

On the Aachen Day-Night dataset, the image retrieval-based methods performed overall considerably worse than structure-based methods and hierarchical methods. For the RobotCar dataset, the image retrieval methods essentially achieve the same coarse-precision performance as the structure-based and hierarchical methods, but with hierarchical methods being slightly better in this regime. This improved performance of image-retrieval methods is caused by the lower variation in viewpoints between database and query images as the car follows the same road. The poorer performance in the higher precision regimes should not be surprising, since these methods are constrained to outputting camera poses that coincide with one of the database images.

TABLE 4
Performance of state-of-the-art methods on the RobotCar Seasons dataset.

	day conditions							night conditions	
	dawn	dusk	OC-summer	OC-winter	rain	snow	sun	night	night-rain
m	.25 / .50 / 5.0	.25 / .50 / 5.0	.25 / .50 / 5.0	.25 / .50 / 5.0	.25 / .50 / 5.0	.25 / .50 / 5.0	.25 / .50 / 5.0	.25 / .50 / 5.0	.25 / .50 / 5.0
deg	2 / 5 / 10	2 / 5 / 10	2 / 5 / 10	2 / 5 / 10	2 / 5 / 10	2 / 5 / 10	2 / 5 / 10	2 / 5 / 10	2 / 5 / 10
Sequential methods									
SeqSLAM [57]	3.1 / 9.7 / 15.4	0.5 / 7.1 / 22.8	0.0 / 2.4 / 80.6	0.0 / 4.3 / 13.4	9.8 / 4.9 / 22.0	2.3 / 11.6 / 27.4	0.0 / 1.3 / 3.1	0.0 / 0.0 / 1.3	1.0 / 3.0 / 4.9
Hierarchical methods									
Vis. Loc. Using Sparse Semantic 3D Map [75]	58.1 / 93.8 / 99.1	76.6 / 95.4 / 100.0	47.4 / 94.3 / 100.0	51.2 / 98.8 / 100.0	78.5 / 94.6 / 100.0	65.1 / 97.7 / 100.0	55.4 / 83.0 / 99.1	13.3 / 32.3 / 51.8	15.3 / 40.4 / 51.7
Hierarchical-Localization (multi-cam) [66]	60.4 / 91.6 / 99.6	70.1 / 95.9 / 100.0	52.1 / 93.4 / 99.5	54.3 / 98.8 / 100.0	77.1 / 93.7 / 100.0	69.3 / 97.7 / 100.0	60.3 / 82.6 / 96.0	16.4 / 31.4 / 52.2	10.8 / 33.0 / 43.8
DenseVLAD & D2-Net [24]	56.4 / 93.8 / 99.6	77.2 / 95.9 / 100.0	43.6 / 91.0 / 98.1	53.7 / 96.3 / 100.0	77.1 / 94.6 / 100.0	64.7 / 95.8 / 99.1	56.2 / 79.0 / 92.4	27.0 / 48.7 / 59.7	25.6 / 45.8 / 56.7
ToDayGAN, NetVLAD, D2-Net	51.5 / 89.4 / 96.5	73.1 / 95.4 / 100.0	43.1 / 94.3 / 99.5	50.0 / 95.7 / 100.0	74.6 / 94.6 / 100.0	62.3 / 95.3 / 100.0	56.2 / 85.7 / 99.1	19.9 / 54.4 / 81.0	25.6 / 62.1 / 86.7
Hierarch.-Loc. NetVLAD+SuperPoint [66]	54.2 / 83.7 / 92.5	65.5 / 94.4 / 100.0	51.7 / 91.9 / 97.2	55.5 / 98.2 / 100.0	75.1 / 93.2 / 100.0	70.2 / 96.3 / 100.0	58.9 / 86.2 / 97.8	13.3 / 24.8 / 41.2	3.9 / 16.3 / 30.5
Asymmetric Hypercolumn Matching [30]	48.9 / 86.3 / 98.2	61.4 / 92.9 / 99.5	45.5 / 88.6 / 95.3	45.7 / 95.7 / 100.0	71.2 / 93.7 / 100.0	62.3 / 94.0 / 100.0	48.2 / 82.6 / 95.5	20.8 / 67.7 / 96.9	35.0 / 82.8 / 100.0
Structure-based methods									
Semantic Match Consistency [88]	56.4 / 94.7 / 100.0	72.6 / 94.9 / 100.0	44.5 / 93.8 / 100.0	47.6 / 95.7 / 100.0	78.0 / 94.6 / 100.0	60.5 / 97.7 / 100.0	52.2 / 80.8 / 100.0	10.2 / 26.5 / 50.9	7.4 / 33.5 / 48.8
Active Search v1.1 w/ pose prior	53.7 / 94.7 / 100.0	72.1 / 94.9 / 100.0	40.3 / 90.0 / 100.0	43.9 / 98.2 / 100.0	78.5 / 93.7 / 100.0	63.7 / 97.2 / 100.0	50.0 / 76.3 / 98.2	11.1 / 17.3 / 35.0	8.4 / 26.1 / 36.0
Act. Search on Seq. of Triplets (uses GT info.)	52.4 / 94.7 / 100.0	64.5 / 95.9 / 100.0	22.7 / 95.7 / 100.0	45.1 / 100.0 / 100.0	72.2 / 95.1 / 100.0	61.9 / 94.0 / 100.0	53.6 / 79.9 / 100.0	4.0 / 17.3 / 42.5	6.9 / 35.0 / 52.2
Active Search on Camera Triplets	55.5 / 93.4 / 100.0	58.9 / 89.3 / 100.0	37.9 / 92.4 / 100.0	48.2 / 95.1 / 100.0	68.3 / 93.7 / 100.0	58.6 / 95.8 / 100.0	52.2 / 77.7 / 91.5	1.3 / 4.9 / 10.2	2.0 / 11.3 / 14.3
Active Search v1.1	50.2 / 92.5 / 99.6	64.5 / 94.9 / 100.0	37.4 / 88.2 / 97.6	41.5 / 96.3 / 100.0	74.6 / 93.7 / 100.0	56.7 / 93.5 / 99.5	37.1 / 64.3 / 88.4	2.7 / 7.1 / 12.4	1.0 / 15.3 / 21.2
CityScaleLocalization [84]	54.2 / 89.4 / 96.9	75.1 / 95.4 / 100.0	37.4 / 82.9 / 91.5	48.2 / 96.3 / 100.0	73.7 / 94.6 / 100.0	61.4 / 94.9 / 97.2	33.9 / 52.7 / 71.0	0.4 / 1.3 / 6.2	1.0 / 5.4 / 12.8
DGCNCCC [55]	7.9 / 30.8 / 85.0	12.2 / 45.7 / 96.4	9.0 / 35.5 / 93.8	3.7 / 17.1 / 92.7	17.6 / 48.8 / 96.6	8.8 / 32.6 / 95.8	4.0 / 15.6 / 87.1	0.0 / 0.4 / 10.6	0.0 / 2.0 / 17.2
Image-retrieval based methods									
Localizing Vis. Landm. for 2D matching [95]	60.4 / 93.0 / 99.1	70.1 / 93.9 / 99.0	51.2 / 93.8 / 99.1	54.9 / 95.7 / 98.2	75.1 / 94.1 / 99.5	64.2 / 96.7 / 98.1	58.0 / 83.9 / 95.5	17.3 / 35.8 / 65.0	13.8 / 38.9 / 63.5
DenseVLAD (single-scale, top-1 interp.)	15.0 / 40.1 / 95.6	12.2 / 40.6 / 99.0	13.3 / 36.0 / 89.6	13.4 / 39.0 / 100.0	20.0 / 54.6 / 100.0	11.6 / 44.2 / 96.3	4.9 / 20.5 / 77.7	1.8 / 7.1 / 28.3	2.5 / 9.9 / 27.1
DenseVLAD (single-scale) [89]	13.7 / 41.0 / 95.6	8.6 / 37.1 / 99.0	9.5 / 35.5 / 89.1	1.8 / 31.1 / 100.0	14.1 / 49.3 / 100.0	13.0 / 40.5 / 96.3	7.1 / 19.2 / 77.7	2.2 / 6.6 / 28.3	1.5 / 6.9 / 26.6
ToDayGAN + DenseVLAD [1]	15.4 / 45.8 / 97.4	7.6 / 35.5 / 98.5	9.0 / 30.3 / 88.2	2.4 / 28.0 / 97.6	13.7 / 50.7 / 100.0	10.2 / 38.1 / 93.5	8.0 / 22.3 / 78.1	0.9 / 9.3 / 59.3	3.0 / 16.7 / 53.2
DenseVLAD [89]	15.4 / 45.8 / 97.4	7.6 / 35.5 / 98.5	9.0 / 30.3 / 88.2	2.4 / 28.0 / 97.6	13.7 / 50.7 / 100.0	10.2 / 38.1 / 93.5	8.0 / 22.3 / 78.1	0.9 / 4.4 / 24.3	2.5 / 5.9 / 25.1
Localizing Vis. Landm. for Place Rec. [95]	13.2 / 38.3 / 81.9	7.1 / 28.4 / 89.3	7.6 / 31.8 / 87.7	2.4 / 26.8 / 91.5	13.7 / 44.9 / 97.6	11.2 / 35.8 / 86.0	12.1 / 33.0 / 89.3	6.2 / 18.1 / 58.4	5.9 / 19.7 / 66.5
NetVLAD [2]	10.1 / 28.2 / 87.7	4.6 / 25.4 / 97.5	10.0 / 35.1 / 97.6	2.4 / 28.0 / 100.0	12.2 / 46.8 / 100.0	8.8 / 32.6 / 95.3	8.5 / 22.8 / 88.8	0.0 / 0.9 / 18.1	0.5 / 2.0 / 13.3
FAB-MAP [21]	2.2 / 5.3 / 10.6	2.5 / 18.3 / 57.4	0.9 / 9.5 / 30.8	0.6 / 14.0 / 46.3	12.2 / 40.5 / 92.2	3.3 / 8.8 / 28.4	0.0 / 0.0 / 0.9	0.0 / 0.0 / 0.0	0.0 / 0.0 / 0.0

TABLE 5
Using multiple images for pose estimation (ActiveSeach+GC) on the RobotCar Seasons dataset.

	all day	**all night**
m	.25 / .50 / 5.0	.25 / .50 / 5.0
deg	2 / 5 / 10	2 / 5 / 10
ActiveSearch v1.1	51.7 / 88.6 / 97.7	1.9 / 11.0 / 16.6
CSL	54.6 / 85.9 / 93.3	0.7 / 3.3 / 9.3
ActiveSearch+GC (triplet)	54.3 / 90.9 / 98.7	1.6 / 7.9 / 12.1
ActiveSearch+GC (sequence, GT)	53.3 / 93.3 / 100.0	5.4 / 25.6 / 47.1
SeqSLAM	1 / 6 / 15.9	0.5 / 1.4 / 3

TABLE 6
Using location priors to query only submodels rather than the full RobotCar Seasons dataset for night-time queries.

		RobotCar - all night
	m	.25 / .50 / 5.0
	deg	2 / 5 / 10
ActiveSearch	full model	0.9 / 3 / 4.9
	sub-model	4.4 / 11.7 / 16.6
CSL	full model	0.7 / 3.3 / 9.3
	sub-model	0.5 / 4.7 / 18.4
ActiveSearch+GC (triplet)	full model	1.6 / 7.9 / 12.1
	sub-model	9.3 / 21.2 / 29.4
ActiveSearch+GC (sequence, GT)	full model	5.4 / 25.6 / 47.1
	sub-model	17.7 / 42.7 / 64.1
ActiveSearch+GC (sequence, VO)	full model	1.4 / 11.2 / 24.2
	sub-model	3.7 / 16.1 / 48
LocalSfM	sub-model	20 / 35.9 / 49.9

Compared to Aachen Day-Night, there is an even stronger drop in pose accuracy between day and night for the RobotCar dataset. All methods fail to localize a significant number of queries for both the high- and medium-precision regimes.

The better performance of essentially all methods under ”night+rain” compared to ”night” comes from the autoexposure of the RobotCar’s cameras. A longer exposure is used for the ”night”, leading to significant motion blur.

For the night-time images, the *Asymmetric Hypercolumn Matching* [30] method stands out in particular, beating the other methods with a significant margin, localizing more than 90% of the images in the coarse-precision regime. This is likely due to the dense feature extraction strategy this method employs on the query images, bypassing the need for the feature detector to re-detect the same features seen during daytime from these blurry images.

Multi-image queries. The RobotCar is equipped with three synchronized cameras and captures sequences of images for each camera. Rather than querying with only a single image, we can thus also query with multiple photos. Tab. 5 shows the results obtained with SeqSLAM (which uses temporal sequences of all images captured by the three cameras) and Active Search+GC. For the latter, we query with triplets of images taken at the same time as well as with temporal sequences of triplets. For the triplets, we use the known extrinsic calibration between the three cameras mounted on the car. For the temporal sequences, we use relative poses obtained from the ground truth (GT) absolute poses. Because PFSL takes a couple of seconds to converge after initialization, and the sequences in the RobotCar Seasons Dataset are relatively short compared to that time, we did not run PFSL for this data set. For readability, we only show the results summarized for day- and night-conditions.

Tab. 5 shows that Active Search+GC consistently outperforms the corresponding single image methods in terms of pose accuracy. Active Search+GC is able to accumulate correct matches over multiple images. This enables Active Search+GC to succeed even if only a few matches are found for each individual image. Naturally, the largest gain can be observed when using multiple images in a sequence.

Location priors. In all previous experiments, we considered the full RobotCar 3D model for localization. However, it is not uncommon in outdoor settings to have a rough prior on the location at which the query image was taken. We simulate such a prior by only considering the sub-model relevant to a query rather than the full model. While we observe only a small improvement for day-time queries, localizing night-time queries significantly benefits from solving an easier matching problem (*cf*. Tab. 6). For completeness, we also report results for LocalSfM, which also considers only a small part of the model relevant to a query. Active Search+GC outperforms LocalSfM on this easier matching task when querying with sequences in the coarse regime. This is due to not relying on one single image to provide enough matches.

One drawback of sequence-based localization is that the relative poses between the images in a sequence need to be known quite accurately. Tab. 6 also reports results obtained when using our own multi-camera visual odometry (VO) system to compute the relative poses. The reasons for the performance drop are drift and collapsing trajectories due to degenerate configurations.

6.3 Evaluation on the Extended CMU Seasons Dataset

Compared to the urban scenes shown in the other datasets, significant parts of the Extended CMU Seasons dataset consist of suburban areas, as well as country road and park areas. Seasonal changes can drastically affect the appearance of such regions, making these the most challenging parts of this dataset. In the following, we thus focus on these conditions. Please refer to the benchmark website for results for all conditions of the dataset.

Tab. 7 evaluates the impact of changes in foliage and of different regions (urban, suburban and country road or park) on pose accuracy. The reference condition for the Extended CMU Seasons dataset does not contain foliage. Thus, other conditions for which foliage is also absent lead to the most accurate poses. Unsurprisingly, the images showing heavy vegetation (i.e., the regions labelled *park*) are the most challenging due to the dramatic changes in appearance here.

Similarly as in the RobotCar dataset, the image retrieval methods tend to be characterized by poor performance in the high-precision regime, but fairly good performance in the coarse precision regime. Note that the results for DenseVLAD and NetVLAD consists of the pose error obtained when using the pose of the top-ranked database image as the estimated pose of the query image. Even in the very challenging park regime it correctly retrieves a database image within five meters of the query image more than 60% of the time.

The structure-based methods exhibit an overall better performance than the pure image-retrieval based methods. This is especially true in the high-precision regimes, where structure-based methods perform significantly better. However, some of the image-retrieval based methods still outperform the structure-based methods in the coarse regime. This suggests that global image-level descriptors can still capture useful information even under severe environmental changes, at least in a self-driving car scenario where there tends to be very little viewpoint changes between query images and database images.

The hierarchical methods perform even better than the structure-based methods, bumping up the coarse-precision performance above that of the image retrieval based ones. This is natural since these combine the strengths of those two approaches. Since these methods can use the geometric structure to perform spatial verification of the image retrieval results, they are not limited to using only the top-retrieved database image, and thus the accuracy in the coarse regime is not limited by the corresponding results of the image retrieval methods. This yields a prior on the position, making the subsequent structure-based pose estimation easier.

Lastly, we have the sequential methods, which use all information gathered up until the current frame to perform localization. It should be noted that the top sequential method presented here (PFSL) had access to simulated (noisy) IMU measurements based on ground truth information, such that a particle filtering method employing motion and measurement updates could be evaluated. Still, it is interesting to note that the accuracy is almost 100% for all regimes, suggesting that the long-term visual localization problem is possible to solve robustly in robotics scenarios with a continuous feed of images and IMU data.

7 Conclusion & Lessons Learned

In this paper, we have introduced three challenging benchmark datasets for visual localization and reviewed the current state in the field of long-term visual localization. Significant progress has been made in the field since the original introduction of the datasets, but there still remains room for improvement. In particular, night-time images, and scenes containing little man-made structure remain challenging in the long-term scenario.

To summarize, our main takeaways from the review are the following: (i) Overall, hierarchical methods outperform pure structure-based methods, which in turn outperform pure image-retrieval based methods. However, image-retrieval based methods often perform quite well in the coarse-precision regime. (ii) The top performing methods are essentially the same for all three datasets, indicating that these methods generalize well. (iii) Learning-based local image features seem to outperform handcrafted ones in the long-term localization scenario. (iv) Hierarchical and structure-based methods are robust to most viewing conditions in urban environments, but their performance in heavily vegetated areas, and during night-time still has room for improvement. (v) Localizing night-time images against a database built from day-time photos is still a very challenging problem, even when a location prior is given. (vi) Scenes with a significant amount of vegetation are challenging, even when a location prior is given. (vii) SfM, typically used to obtain ground truth for localization benchmarks, does not fully handle problems (v) and (vi) due to limitations of existing local features. Dense CNN feature matching inside SfM improves pose estimation performance at high computational costs, but does not fully solve the problem. Novel (dense) features, *e.g.*, based on scene semantics [74], seem to be required to solve these problems. Our datasets readily provide a benchmark for such features through the LocalSfM and DenseSfM pipelines. (viii) Image-level descriptors such as DenseVLAD can succeed in scenarios where local feature matching fails. Using the image-retrieval results to reduce the search-space for feature matching is likely the source of the performance boost observed for hierarchical methods. (ix) Utilizing multiple images (in the case of a camera rig) or sequential information can greatly boost the localization performance. It is interesting to note that there still seems to be a large focus in the literature on single-image methods: most methods evaluated in this paper are single-image methods. Despite the large changes in scene appearance they achieve good performance in registering the images one at a time. We expect that substantial performance gains can likely be obtained by integrating these methods into a sequential localization pipeline, and believe this is a promising direction of future research. Other fruitful directions may be reducing memory consumption and increasing run-time efficiency, in order to enable the methods to run in real-time on resource constrained devices.

Acknowledgements. This work was partially supported by the European Regional Development Fund under the project IMPACT (reg. no. CZ.02.1.01/0.0/0.0/15_003/0000468), JSPS KAKENHI Grant Number 15H05313, EPSRC Programme Grant EP/M019918/1, the Swedish Research Council (grant no. 2016-04445), and the Swedish Foundation for Strategic Research (Semantic Mapping and Visual Navigation for Smart Robots).

References

[1] A. Anoosheh, T. Sattler, R. Timofte, M. Pollefeys, and L. Van Gool. Night-to-day image translation for retrieval-based localization. In *2019*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2020.3032010, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 7
Performance of state-of-the-art methods on the Extended CMU Seasons dataset.

	foliage	mixed foliage	no foliage	urban	suburban	park
m	.25 / .50 / 5.0	.25 / .50 / 5.0	.25 / .50 / 5.0	.25 / .50 / 5.0	.25 / .50 / 5.0	.25 / .50 / 5.0
deg	2 / 5 / 10	2 / 5 / 10	2 / 5 / 10	2 / 5 / 10	2 / 5 / 10	2 / 5 / 10
Sequential methods						
PFSL FGSN 200 classes [41], [79]	96.8 / 99.8 / 100.0	97.5 / 99.6 / 100.0	98.6 / 99.9 / 99.9	94.1 / 99.3 / 100.0	100.0 / 100.0 / 100.0	97.6 / 99.9 / 99.9
PFSL Cityscapes classes [40], [79]	89.6 / 97.0 / 100.0	89.6 / 96.7 / 100.0	90.6 / 97.2 / 99.9	88.1 / 93.8 / 100.0	97.4 / 99.2 / 100.0	81.8 / 97.2 / 99.9
SeqSLAM [57]	5.0 / 14.7 / 51.3	6.2 / 18.1 / 60.0	5.7 / 19.0 / 67.0	9.4 / 23.6 / 62.9	3.7 / 12.7 / 55.5	3.7 / 14.5 / 54.1
Hierarchical methods						
Hierarchical-Localization (multi-camera when available) [66]	78.4 / 85.6 / 94.9	82.5 / 89.7 / 96.7	91.7 / 96.0 / 99.1	91.6 / 96.4 / 99.1	84.7 / 91.5 / 98.6	69.3 / 77.8 / 90.5
DenseVLAD & D2-Net [24], [89]	81.5 / 85.9 / 88.9	87.7 / 90.2 / 91.6	90.7 / 92.1 / 92.5	94.0 / 97.7 / 99.1	93.0 / 95.7 / 98.3	66.3 / 69.7 / 71.1
Visual Localization Using Sparse Semantic 3D Map [75]	73.4 / 79.1 / 84.2	75.1 / 81.8 / 87.9	90.9 / 94.5 / 97.1	88.8 / 93.6 / 96.3	78.0 / 83.8 / 89.2	63.6 / 70.3 / 77.3
Hierarchical-Localization NetVLAD+SuperPoint [66]	69.2 / 75.5 / 88.3	75.2 / 81.7 / 90.8	88.7 / 92.8 / 96.4	89.5 / 94.2 / 97.9	76.5 / 82.7 / 92.7	57.4 / 64.4 / 80.4
Asymmetric Hypercolumn Matching [30]	58.5 / 75.7 / 87.8	62.9 / 79.6 / 89.4	72.0 / 87.7 / 94.5	65.7 / 82.7 / 91.0	66.5 / 82.6 / 92.9	54.3 / 71.6 / 84.1
Structure-based methods						
Match Consistency with Fine-Grained Segmentations (FGSN+SSMC) [41], [88]	65.5 / 71.6 / 77.7	66.0 / 73.3 / 81.0	84.6 / 89.6 / 93.5	85.3 / 91.0 / 94.6	69.5 / 76.4 / 83.7	51.4 / 57.6 / 65.5
Match Consistency with Fine-Grained Segmentations (GSMC-200) [88]	64.3 / 69.4 / 74.4	65.0 / 70.7 / 76.6	87.2 / 91.0 / 94.3	86.4 / 91.2 / 93.8	77.0 / 82.9 / 88.7	38.9 / 43.4 / 50.0
CityScaleLocalization [84]	47.0 / 50.2 / 55.3	52.4 / 56.1 / 62.0	80.3 / 83.2 / 86.6	71.2 / 74.6 / 78.7	57.8 / 61.7 / 67.5	34.5 / 37.0 / 42.2
DGCNCCC [55]	8.9 / 25.8 / 74.1	9.4 / 26.6 / 77.0	15.7 / 38.4 / 83.6	17.1 / 41.5 / 89.1	8.9 / 26.8 / 77.1	4.8 / 16.2 / 63.3
Image-retrieval based methods						
Localizing Visual Landmarks for 2D matching [95]	59.4 / 66.1 / 77.9	64.8 / 73.0 / 83.7	82.5 / 87.8 / 92.7	84.3 / 89.3 / 93.0	68.0 / 75.1 / 84.4	42.4 / 51.4 / 69.7
Localizing Visual Landmarks for Place Recognition [95]	9.5 / 26.7 / 77.4	10.3 / 28.4 / 79.0	9.4 / 30.3 / 84.6	17.3 / 42.5 / 89.0	5.8 / 19.4 / 76.1	6.6 / 23.1 / 73.0
DenseVLAD [89]	7.4 / 21.1 / 68.0	8.5 / 24.5 / 73.0	10.0 / 32.6 / 88.0	14.7 / 36.3 / 83.9	5.3 / 18.7 / 73.9	5.2 / 19.1 / 62.0
NetVLAD [2]	6.2 / 18.5 / 74.3	5.8 / 17.6 / 71.1	6.7 / 20.9 / 79.4	12.2 / 31.5 / 89.8	3.7 / 13.9 / 74.7	2.6 / 10.4 / 55.9

International Conference on Robotics and Automation (ICRA), pages 5958–5964. IEEE, 2019. 6, 10

[2] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proc. CVPR*, 2016. 2, 6, 9, 10, 12

[3] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *Proc. CVPR*, 2012. 2, 6, 8

[4] R. Arandjelović and A. Zisserman. DisLocation: Scalable descriptor distinctiveness for location recognition. In *Proc. ACCV*, 2014. 2

[5] H. Badino, D. Huber, and T. Kanade. Visual topometric localization. In *Proc. IV*, 2011. 2, 5

[6] P. J. Besl and N. D. McKay. Method for registration of 3-D shapes. *IEEE PAMI*, 14(2):239–256, 1992. 4

[7] J.-L. Blanco-Claraco, F.-Á. Moreno-Dueñas, and J. González-Jiménez. The Málaga urban dataset: High-rate stereo and LiDAR in a realistic urban scenario. *IJRR*, 33(2):207–214, 2014. 2

[8] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *Proc. ICCV*, 2007. 8

[9] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother. DSAC - Differentiable RANSAC for Camera Localization. In *Proc. CVPR*, 2017. 3, 6

[10] E. Brachmann, F. Michel, A. Krull, M. Y. Yang, S. Gumhold, and C. Rother. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In *Proc. CVPR*, 2016. 3

[11] F. Camposeco, T. Sattler, A. Cohen, A. Geiger, and M. Pollefeys. Toroidal Constraints for Two-Point Localization under High Outlier Ratios. In *Proc. CVPR*, 2017. 2

[12] F. Camposeco, T. Sattler, and M. Pollefeys. Minimal Solvers for Generalized Pose and Scale Estimation from Two Rays and One Point. In *Proc. ECCV*, 2016. 3

[13] S. Cao and N. Snavely. Graph-based discriminative learning for location recognition. In *Proc. CVPR*, 2013. 3

[14] S. Cao and N. Snavely. Minimal Scene Descriptions from Structure from Motion Models. In *Proc. CVPR*, 2014. 1

[15] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice. University of Michigan North Campus long-term vision and lidar dataset. *IJRR*, 35(9):1023–1035, 2016. 2, 3

[16] D. M. Chen, G. Baatz, K. Köser, S. S. Tsai, R. Vedantham, T. Pylvänäinen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, B. Girod, and R. Grzeszczuk. City-Scale Landmark Identification on Mobile Devices. In *Proc. CVPR*, 2011. 2, 3

[17] Z. Chen, A. Jacobson, N. Sünderhauf, B. Upcroft, L. Liu, C. Shen, I. D. Reid, and M. Milford. Deep learning features at scale for visual place recognition. In *Proc. ICRA*, 2017. 2, 3

[18] S. Choudhary and P. J. Narayanan. Visibility probability structure from sfm datasets and applications. In *Proc. ECCV*, 2012. 2

[19] R. Clark, S. Wang, A. Markham, N. Trigoni, and H. Wen. VidLoc: A Deep Spatio-Temporal Model for 6-DoF Video-Clip Relocalization. In *Proc. CVPR*, 2017. 3

[20] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. CVPR*, pages 3213–3223, 2016. 7

[21] M. Cummins and P. Newman. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *IJRR*, 27(6):647–665, 2008. 2, 9, 10

[22] D. DeTone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proc. CVPR Workshops*, pages 224–236, 2018. 3, 7

[23] M. Donoser and D. Schmalstieg. Discriminative Feature-to-Point Matching in Image-Based Locallization. In *Proc. CVPR*, 2014. 2

[24] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler. D2-net: A trainable cnn for joint description and detection of local features. In *Proc. CVPR*, pages 8092–8101, 2019. 3, 7, 9, 10, 12

[25] O. Enqvist and F. Kahl. Robust optimal pose estimation. In *Proc. ECCV*, pages 141–153, Marseille, France, 2008. 2

[26] O. Enqvist, C. Olsson, and F. Kahl. Non-sequential structure from motion. In *Workshop on Omnidirectional Vision, Camera Networks and Non-Classical Cameras (OMNIVIS)*, Barcelona, Spain, Nov. 2011. 5

[27] M. Fischler and R. Bolles. Random Sampling Consensus: A Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography. *Commun. ACM*, 24:381–395, 1981. 4, 8

[28] D. Gálvez-López and J. D. Tardos. Bags of binary words for fast place recognition in image sequences. *T-RO*, 28(5):1188–1197, 2012. 2

[29] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *IJRR*, 32(11):1231–1237, 2013. 2

[30] H. Germain, G. Bourmaud, and V. Lepetit. Sparse-to-dense hypercolumn matching for long-term visual localization. 2019. 2, 7, 9, 10, 12

[31] P. Gronat, J. Sivic, G. Obozinski, and T. Pajdla. Learning and calibrating per-location classifiers for visual place recognition. *IJCV*, 118(3):319–336, 2016. 3

[32] R. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation Averaging. *IJCV*, 103(3):267–305, 2013. 5

[33] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From Structure-from-Motion Point Clouds to Fast Location Recognition. In *Proc. CVPR*, 2009. 2, 3

[34] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Proc. CVPR*, 2010. 2

[35] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Proc. CVPR*, pages 3304–3311, jun 2010. 6

[36] T. Jenicek and O. Chum. No fear of the dark: Image retrieval under varying illumination conditions. In *Proc. ICCV*, pages 9696–9704, 2019. 3

[37] A. Kendall, M. Grimes, and R. Cipolla. PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. In *Proc. ICCV*, 2015. 2, 3, 6

[38] L. Kneip, D. Scaramuzza, and R. Siegwart. A Novel Parametrization of the Perspective-Three-Point Problem for a Direct Computation of Absolute Camera Position and Orientation. In *Proc. CVPR*, 2011. 4

[39] J. Knopp, J. Sivic, and T. Pajdla. Avoiding confusing features in place recognition. In *Proc. ECCV*, 2010. 3

[40] M. Larsson, E. Stenborg, L. Hammarstrand, M. Pollefeys, T. Sattler, and F. Kahl. A cross-season correspondence dataset for robust semantic segmentation. In *Proc. CVPR*, pages 9532–9542, 2019. 12

[41] M. Larsson, E. Stenborg, C. Toft, L. Hammarstrand, T. Sattler, and F. Kahl. Fine-grained segmentation networks: Self-supervised segmentation for improved long-term visual localization. In *Proc. ICCV*, pages 31–41, 2019. 7, 12

[42] V. Larsson, J. Fredriksson, C. Toft, and F. Kahl. Outlier rejection for absolute pose estimation with known orientation. In *Proc. BMVC*, 2016. 1, 2

[43] G. H. Lee, B. Li, M. Pollefeys, and F. Fraundorfer. Minimal solutions for the multi-camera pose estimation problem. *IJRR*, 34(7):837–848, 2015.

3, 5

[44] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. Worldwide Pose Estimation Using 3D Point Clouds. In *Proc. ECCV*, 2012. 1, 2, 3

[45] Y. Li, N. Snavely, and D. P. Huttenlocher. Location Recognition using Prioritized Feature Matching. In *Proc. ECCV*, 2010. 1, 2, 3

[46] Z. Li and N. Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proc. CVPR*, pages 2041–2050, 2018. 7

[47] C. Linegar, W. Churchill, and P. Newman. Made to measure: Bespoke landmarks for 24-hour, all-weather localisation with a camera. In *Proc. ICRA*, 2016. 2, 3

[48] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. SIFT Flow: Dense correspondence across different scenes. In *Proc. ECCV*, pages 28–42, 2008. 8

[49] L. Liu, H. Li, and Y. Dai. Efficient Global 2D-3D Matching for Camera Localization in a Large-Scale 3D Map. In *Proc. ICCV*, 2017. 2

[50] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60(2):91–110, 2004. 2, 6, 8

[51] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, 2016. 2

[52] S. Lynen, T. Sattler, M. Bosse, J. Hesch, M. Pollefeys, and R. Siegwart. Get Out of My Lab: Large-scale, Real-Time Visual-Inertial Localization. In *Proc. RSS*, 2015. 1, 2

[53] W. Maddern, M. Milford, and G. Wyeth. CAT-SLAM: probabilistic localisation and mapping using a continuous appearance-based trajectory. *IJRR*, 31(4):429–451, 2012. 3

[54] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *IJRR*, 36(1):3–15, 2017. 2, 4

[55] I. Melekhov, A. Tiulpin, T. Sattler, M. Pollefeys, E. Rahtu, and J. Kannala. DGC-Net: Dense Geometric Correspondence Network. *arXiv e-prints*, page arXiv:1810.08393, Oct 2018. 7, 9, 10, 12

[56] M. Milford, S. Lowry, N. Sunderhauf, S. Shirazi, E. Pepperell, B. Upcroft, C. Shen, G. Lin, F. Liu, C. Cadena, et al. Sequence searching with deep-learnt depth for condition-and viewpoint-invariant route-based place recognition. In *Proc. CVPR Workshops*, 2015. 3

[57] M. J. Milford and G. F. Wyeth. SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights. In *Proc. ICRA*, 2012. 2, 3, 8, 10, 12

[58] P. Mühlfellner, M. Bürki, M. Bosse, W. Derendarz, R. Philippsen, and P. Furgale. Summary maps for lifelong visual localization. *Journal of Field Robotics*, 2015. 3

[59] R. Mur-Artal, J. M. M. Montiel, and J. D. Tards. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *T-RO*, 31(5):1147–1163, 2015. 2

[60] T. Naseer, G. L. Oliveira, T. Brox, and W. Burgard. Semantics-aware visual localization under challenging perceptual conditions. In *ICRA*, 2017. 2, 3

[61] T. Naseer, L. Spinello, W. Burgard, and C. Stachniss. Robust visual robot localization across seasons using network flows. In *Proc. AAAI*, 2014. 3

[62] R. Pless. Using Many Cameras as One. In *Proc. CVPR*, 2003. 3, 5

[63] F. Radenović, J. L. Schönberger, D. Ji, J.-M. Frahm, O. Chum, and J. Matas. From Dusk till Dawn: Modeling in the Dark. In *Proc. CVPR*, 2016. 2

[64] J. Revaud, C. De Souza, M. Humenberger, and P. Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. In *Advances in Neural Information Processing Systems*, pages 12405–12415, 2019. 3

[65] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proc. CVPR*, pages 4510–4520, 2018. 7

[66] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12716–12725, 2019. 2, 7, 9, 10, 12

[67] T. Sattler, M. Havlena, F. Radenovic, K. Schindler, and M. Pollefeys. Hyperpoints and fine vocabularies for large-scale location recognition. In *Proc. ICCV*, 2015. 1, 2

[68] T. Sattler, M. Havlena, K. Schindler, and M. Pollefeys. Large-scale location recognition and the geometric burstiness problem. In *Proc. CVPR*, 2016. 2

[69] T. Sattler, B. Leibe, and L. Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE PAMI*, 39(9):1744–1756, 2017. 1, 2, 6

[70] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, F. Kahl, and T. Pajdla. Benchmarking 6DOF outdoor visual localization in changing conditions. In *Proc. CVPR*, pages 8601–8610, 2018. 2, 3, 4, 5

[71] T. Sattler, A. Torii, J. Sivic, M. Pollefeys, H. Taira, M. Okutomi, and T. Pajdla. Are Large-Scale 3D Models Really Necessary for Accurate Visual Localization? In *Proc. CVPR*, 2017. 1, 2, 3

[72] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt. Image Retrieval for Image-Based Localization Revisited. In *Proc. BMVC*, 2012. 2, 3

[73] J. L. Schönberger and J.-M. Frahm. Structure-From-Motion Revisited. In *Proc. CVPR*, June 2016. 3, 8

[74] J. L. Schönberger, M. Pollefeys, A. Geiger, and T. Sattler. Semantic Visual Localization. In *Proc. CVPR*, 2018. 3, 11

[75] T. Shi, S. Shen, X. Gao, and L. Zhu. Visual Localization Using Sparse Semantic 3D Map. *arXiv e-prints*, page arXiv:1904.03803, Apr 2019. 2, 7, 9, 10, 12

[76] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon. Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images. In *Proc. CVPR*, 2013. 2, 3

[77] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. ICLR*, 2015. 6, 8

[78] J. Sivic and A. Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *Proc. ICCV*, 2003. 2, 6

[79] E. Stenborg, C. Toft, and L. Hammarstrand. Long-term visual localization using semantically segmented images. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6484–6490. IEEE, 2018. 6, 7, 12

[80] X. Sun, Y. Xie, P. Luo, and L. Wang. A Dataset for Benchmarking Image-Based Localization. In *Proc. CVPR*, 2017. 2, 3

[81] N. Sünderhauf, F. Dayoub, S. Shirazi, B. Upcroft, and M. Milford. On the Performance of ConvNet Features for Place Recognition. In *Proc. IROS*, 2015. 3

[82] N. Sünderhauf, P. Neubert, and P. Protzel. Are we there yet? Challenging SeqSLAM on a 3000 km journey across all four seasons. In *Proc. ICRA Workshops*, 2013. 2, 3, 5, 8

[83] N. Sünderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford. Place Recognition with ConvNet Landmarks: Viewpoint-Robust, Condition-Robust, Training-Free. In *Proc. RSS*, 2015. 2, 3

[84] L. Svärm, O. Enqvist, F. Kahl, and M. Oskarsson. City-Scale Localization for Cameras with Known Vertical Direction. *IEEE PAMI*, 39(7):1455–1461, 2017. 1, 2, 6, 9, 10, 12

[85] L. Svärm, O. Enqvist, M. Oskarsson, and F. Kahl. Accurate Localization and Pose Estimation for Large 3D Models. In *Proc. CVPR*, 2014. 2

[86] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii. InLoc: Indoor Visual Localization with Dense Matching and View Synthesis. In *Proc. CVPR*, 2018. 1

[87] C. Toft, C. Olsson, and F. Kahl. Long-term 3D Localization and Pose from Semantic Labellings. In *ICCV Workshops*, 2017. 2

[88] C. Toft, E. Stenborg, L. Hammarstrand, L. Brynte, M. Pollefeys, T. Sattler, and F. Kahl. Semantic match consistency for long-term visual localization. In *Proc. ECCV*, pages 383–399, 2018. 2, 6, 10, 12

[89] A. Torii, R. Arandjelović, J. Sivic, M. Okutomi, and T. Pajdla. 24/7 Place Recognition by View Synthesis. In *Proc. CVPR*, 2015. 2, 3, 6, 8, 9, 10, 12

[90] A. Torii, J. Sivic, M. Okutomi, and T. Pajdla. Visual place recognition with repetitive structures. *IEEE PAMI*, 2015. 2, 3

[91] Y. Verdie, K. Yi, P. Fua, and V. Lepetit. Tilde: A temporally invariant learned detector. In *Proc. CVPR*, pages 5279–5288, 2015. 3

[92] F. Walch, C. Hazirbas, L. Leal-Taixé, T. Sattler, S. Hilsenbeck, and D. Cremers. Image-Based Localization Using LSTMs for Structured Feature Correlation. In *Proc. ICCV*, 2017. 3

[93] S. Wang, M. Bai, G. Mattyus, H. Chu, W. Luo, B. Yang, J. Liang, J. Cheverie, S. Fidler, and R. Urtasun. TorontoCity: Seeing the World With a Million Eyes. In *Proc. ICCV*, 2017. 2

[94] T. Weyand, I. Kostrikov, and J. Philbin. Planet - photo geolocation with convolutional neural networks. In *Proc. ECCV*, 2016. 3

[95] Z. Xin, Y. Cai, T. Lu, X. Xing, S. Cai, J. Zhang, Y. Yang, and Y. Wang. Localizing discriminative visual landmarks for place recognition. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5979–5985. IEEE, 2019. 6, 9, 10, 12

[96] B. Zeisl, T. Sattler, and M. Pollefeys. Camera pose voting for large-scale image-based localization. In *Proc. ICCV*, 2015. 1, 2

[97] H. Zhou, T. Sattler, and D. W. Jacobs. Evaluating Local Features for Day-Night Matching. In *Proc. ECCV Workshops*, 2016. 7, 8

Carl Toft received his M.Sc. in physics and astronomy from Chalmers University of Technology in 2014. Since 2016, he is a PhD student in the Computer Vision Group at Chalmers. He mainly works in visual localization and mapping, and is currently working on developing localization methods that are robust under large appearance variations in the environment due to weather, seasons and illumination changes.

Will Maddern leads the mapping and localisation efforts at Nuro (nuro.ai). Prior to joining Nuro, Will was a Senior Researcher with the Oxford Robotics Institute at the University of Oxford, and flagship lead for the Oxford RobotCar project (robotcar.org.uk). Will is responsible for the Oxford RobotCar Dataset, and has co-organized a workshop on deep visual SLAM at CVPR 2018 and a tutorial on vision for autonomous driving at ICCV 2015.

Akihiko Torii received a Master degree and PhD from Chiba University in 2003 and 2006. He then spent four years as a post-doctoral researcher in Czech Technical University in Prague. Since 2010, he has been with Tokyo Institute of Technology, where he is currently an assitant professor in the Department of Systems and Control Engineering, the School of Engineering.

Lars Hammarstrand is an Associate Professor at Chalmers University of Technology. He has previously worked at Volvo Car Group, where he was responsible for developing the sensor fusion platform for their active safety systems. His current research interests lie in the combination of Machine learning and Bayesian statistics in general and, in particular, with application to robust localization and perception for self-driving vehicles.

Erik Stenborg received a M.Sc. in computer engineering in 2005, and has since then worked with active safety systems at Volvo Car Corporation and Zenuity. He is currently employed by Zenuity as an industrial PhD student in the Signal processing research group at Chalmers University of Technology and is involved in a project that aims to improve vehicle positioning to the point where it is reliable enough for autonomous driving.

Daniel Safari received his Master degree in electrical engineering from the Technical University of Denmark in 2018; his thesis being on post-capture calibration of industrial underwater laser imaging systems. Since graduation, he has worked broadly with SLAM and visual odometry systems at Sony R&D in Tokyo. His academic interests lie primarily in vision algorithms and architectures suitable for hardware realization.

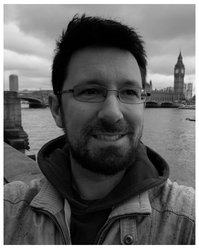
Masatoshi Okutomi received the B.Eng. degree from the University of Tokyo in 1981, and an M. Eng. degree from the Tokyo Institute of Technology in 1983. He joined the Canon Research Center in 1983. From 1987 to 1990, he was a Visiting Research Scientist at Carnegie Mellon University. He received the Dr. Eng. degree from the Tokyo Institute of Technology, in 1993 where hi is currently a professor at the Dept. of Systems and Control Engineering.

Marc Pollefeys is a Prof. of Computer Science at ETH Zurich and Director of Science at Microsoft. He is best known for his work in 3D computer vision, but also for works on robotics, graphics, machine learning, and camera-based self-driving cars and drones. He received a M.Sc. and a PhD from the KU Leuven in Belgium in 1994 and 1999, respectively. He became an assistant professor at the University of North Carolina in Chapel Hill in 2002 and joined ETH Zurich as a full professor in 2007.

Josef Sivic received his Ph.D. from the University of Oxford and Habilitation from Ecole Normale Superieure in Paris. He currently leads a team working on Intelligent Machine Perception at the Czech Institute of Robotics, Informatics and Cybernetics at CTU in Prague. He has been awarded the Longuet-Higgins prize (CVPR'07) and the Helmholtz prize (ICCV'03 and ICCV'05) for fundamental contributions to computer vision and served as area and program chair at major computer vision conferences.

Tomas Pajdla received the MSc and PhD degrees from the Czech Technical University, in Prague. He works in geometry and algebra of computer vision and robotics, with emphasis on nonclassical cameras, 3D reconstruction, and industrial vision. He contributed to introducing epipolar geometry of panoramic cameras, noncentral camera models generated by linear mappings, generalized epipolar geometries, and to developing solvers for minimal problems in structure from motion. He is a member of the IEEE.

Fredrik Kahl received his Ph.D. degree in mathematics in 2001 from Lund University. His thesis was awarded the Best Nordic Thesis Award in pattern recognition and image analysis 2001-2002 at the Scandinavian Conference on Image Analysis 2003, and in 2005, he received the ICCV Marr Prize. Since 2013, he is a professor at Chalmers University of Technology where he leads the Computer Vision Group. Research interests include machine learning, optimization, medical image analysis and computer vision.



Torsten Sattler received his PhD degree from RWTH Aachen University in 2014. He was then a postdoctoral and senior researcher at ETH Zurich in the Computer Vision and Geometry Group. Since 2019, he is an associate professor at Chalmers University of Technology and joined CIIRC as a senior researcher in July 2020. His research interests center around visual localization and 3D mapping and include local features, camera pose estimation, SLAM and scene understanding. He has organized workshops and tutorials on these topics at ECCV, ICCV, and CVPR.