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

Pit30M: A Benchmark for Global Localization in the Age of Self-Driving Cars

Julieta Martinez[1], Sasha Doubov[1,2], Jack Fan[1],
Ioan Andrei Bârsan[1,3], Shenlong Wang[1,3], Gellért Máttyus[1], Raquel Urtasun[1,3]

***Abstract*— We are interested in understanding whether retrieval-based localization approaches are good enough in the context of self-driving vehicles. Towards this goal, we introduce Pit30M, a new image and LiDAR dataset with over 30 million frames, which is 10 to 100 times larger than those used in previous work. Pit30M is captured under diverse conditions (i.e., season, weather, time of the day, traffic), and provides accurate localization ground truth. We also automatically annotate our dataset with historical weather and astronomical data, as well as with image and LiDAR semantic segmentation as a proxy measure for occlusion. We benchmark multiple existing methods for image and LiDAR retrieval and, in the process, introduce a simple, yet effective convolutional network-based LiDAR retrieval method that is competitive with the state of the art. Our work provides, for the first time, a benchmark for sub-metre retrieval-based localization at city scale.**

The dataset, additional experimental results, as well as more information about the sensors, calibration, and metadata, are available on the project website:

`https://uber.com/atg/datasets/pit30m`

I. Introduction

Localizing an autonomous agent accurately and in real time is a fundamental problem in robotics. In the context of autonomous driving, it allows the self-driving vehicles (SDVs), to navigate to their destinations. Furthermore, accurate localization enables the use of HD maps, which boosts and contextualizes downstream autonomy tasks such as perception, motion forecasting, and motion planning.

Localization tasks can be divided into two broad categories: online localization and global localization. Online localization assumes that the pose at the previous time step is known, and is tasked with propagating that information over time, as well as combining it with current sensory measurements. However, small errors may accumulate during online localization, making the pose estimate drift over time or even fail altogether. Global localization aims to overcome this issue, as it re-estimates the global pose without assumptions on previous steps. Hence, global localization is an important fail-safe module that allows self-driving vehicles to recover from temporary online localization failures.

Autonomous driving faces unique challenges when it comes to global localization. To systematically evaluate various global localization approaches in this context, we need a benchmark that reflects the setting and its particular challenges. Ideally, the dataset employed to carry out this study should be diverse, large-scale, and have accurate ground truth over a variety of environments and traffic scenarios. Furthermore, since autonomous driving platforms typically carry a variety of sensors that provide complementary information (such as LiDAR, camera, GPS, and IMU), the benchmark should contain multi-sensory data to enable researchers to exploit multi-modal inputs for the global localization task. Unfortunately, no existing dataset fulfills all these criteria.

In this paper we introduce Pit30M, a dataset that spans over a year of driving in Pittsburgh, PA, USA, comprising over 1 000 trips, 25 000 km, and 1 500 hours driven under different times of day, seasons, and diverse weather conditions. Moreover, we provide accurate ground truth poses (under 10 cm of error) for all our data. With over 30 million images and LiDAR sweeps, our dataset is one to two orders of magnitude larger than the biggest publicly-available dataset for this task. We also provide metadata such as time of day, weather, and approximate occlusion by leveraging image and LiDAR segmentation, which allow us to formally quantify the diversity in our dataset and understand localization errors. We give an overview of the geographical and temporal extent of our dataset in Figure 1.

We investigate both image- and LiDAR-based approaches to retrieval localization. For visual localization, and with our dataset's scale, diversity and density, we find that a modern convolutional backbone with a simple pooling scheme perform on par with state-of-the-art architectures specifically designed for this task, such as NetVLAD [1]. For LiDAR-based localization, we investigate both the latest network architectures and suitable pointcloud representations. We show that bird's-eye view voxelization coupled with a strong convolutional backbone is competitive with the best previously proposed pointcloud representations and architectures for this task – which also rely on NetVLAD pooling. Finally, we provide an analysis of the failure modes and complementarity of LiDAR- and camera-based localization.

II. Related work

Structure- and retrieval-based localization have been studied in the computer vision community for decades. We summarize these two areas of research, as well as regression-based and hybrid approaches to localization. We also briefly review datasets typically used for this task.

[1] Uber Advanced Technologies Group
[2] University of Waterloo
[3] University of Toronto

978-1-7281-6211-9/20/$31.00 ©2020 IEEE

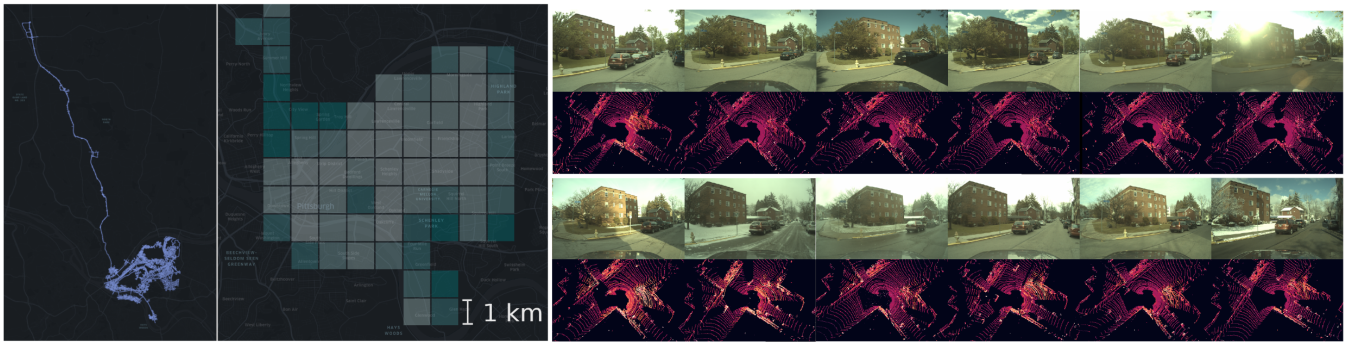


Fig. 1: **Our new localization dataset, Pit30M.** Left: Each square is 1 km^2, for a total area of about 50 km^2 plus over 20 km of highway in the Pittsburgh Metropolitan Area. Right: Examples of image and LiDAR point clouds taken in the same place at different times.

A. Structure-based localization

Image structure-based localization: Cameras perceive a 2d projection of the 3d scene, so research in this area has focused on building consistent 3d maps of the world [2]. Given a query image it is possible to recover its pose by matching its 2d features against the 3d model, and then solving the perspective-n-point (PnP) problem. Towards this goal, researchers have explored a variety of 2d-to-3d descriptor matching techniques [3, 4, 5, 6, 7, 8].

While these approaches can be very accurate, several drawbacks remain. Scalability (maintaining a very large 3d database), for example, remains challenging; while fine vocabularies [7, 9] and model compression [10, 11] provide ways to accelerate matching in large scenes, accuracy suffers, and building and storing 3d models at city scale requires large engineering efforts. Building systems that are robust to long-term changes [12] also remains an active area of research.

LiDAR structure-based localization: LiDARs provide a 3d point cloud that can be aggregated over time to provide a dense 3d reference scan of a scene. This scan can later be used for localization by aligning new observations using, e.g., local registration methods such as iterative closest point [13, 14], or 2d template matching [15, 16, 17, 18, 19].

A substantial portion of work on LiDAR-based localization focuses on online localization. Levinson *et al.* [15] proposed one of the first LiDAR-based online localization systems capable of centimetre-accurate localization, and subsequent work has improved the robustness of such systems by using probabilistic maps [16, 17], leveraging deep learning to bypass the need for calibrated intensity [18], or incorporating real-time kinematic (RTK) information [20]. Nevertheless, given their online nature, such systems require highly accurate initialization (assumed to be provided by another system), and rely on dense high-definition maps which can be prohibitively expensive to collect and build at scale.

B. Retrieval-based localization

Retrieval approaches to localization do not rely on a prebuilt map, but assume access to a database with localized sensor observations. The pose of a query can thus be estimated by finding the nearest observation in the database. Since database entries may be represented by a single vector, these approaches tend to be more scalable; however, their accuracy is limited by the density and coverage of the underlying database, and finding compact yet discriminative representations remains difficult.

Image retrieval-based localization: Classical methods extract local invariant features [21, 22], and aggregate them into a global descriptor such as visual bag-of-words [23] or VLAD [24, 25]. Candidate re-ranking and geometric verification are sometimes used as a second stage to further boost performance [26, 27]. Recent work has used deep convolutional neural networks (CNNs) to learn compact visual representations [1, 28, 29]. For instance, NetVLAD [1] uses a CNN and differentiable VLAD pooling to learn global image representations for retrieval in an end-to-end manner, and RMAC [29] builds a compact deep feature vector with Region of Interest (RoI) pooling.

LiDAR retrieval-based localization: Compared to images, LiDAR-based retrieval methods remain relatively unexplored. While handcrafted 3d descriptors have been used for 3d registration and recognition tasks [30, 31], we are not aware of classical global pooling techniques applied to LiDAR retrieval-based localization.

Recently, work has concentrated on learning deep descriptors from 3d point clouds [32, 33, 34]. PointNetVLAD [35] uses PointNet [36] to generate local per-point features, which are then aggregated by a VLAD [1] layer. PCAN [37] improves upon PointNetVLAD by learning an attention map for aggregation, using an architecture inspired by PointNet++ [38]. Finally, LPD-Net [39] achieves state-of-the-art retrieval results using a graph neural network to leverage local structure when learning global descriptors. While these methods yield excellent results, they operate directly on raw point clouds, which is computationally expensive in general.

C. Other localization approaches

Regression-based localization: In these approaches, the model directly outputs a position in a known scene [40, 41, 42, 43]. The most significant advantage of these methods is that they allow for localization without access to external databases, leading to low memory usage and fast inference.

While promising, these methods have been shown to not generalize well to city-scale localization [12, 44].

Hybrid Localization: Other methods take components from both retrieval- and map-based localization, or incorporate non-traditional elements in their map representation and inference. For instance, recent work has explored the complementarity of semantics, temporal information, and global pose regression [45, 46, 47], or local and global localization features [48] in a single model. Another recent line of work has explored ways of learning map representations that are better suited for visual localization tasks [49, 50].

D. *Current localization datasets*

Datasets are a key component of research in large-scale localization. On one hand, datasets that span large city areas such as SFO-Landmarks [51] and Tokyo/Pittsburgh Street view [25, 52] often provide only GPS readings as reference poses (SFO-Landmarks also considers visual overlap to compute ground truth). Unfortunately, GPS can be inaccurate by several metres, making it hard to quantify the error of localization methods that aim for sub-metre accuracy. This is evident in the evaluation protocol of most previous work in large-scale retrieval-based localization, where a database match is considered correct if it is within 20 or 25 metres of the query [1, 25, 35, 37, 39].

Other datasets such as Cambridge [41] and Aachen [12] derive ground truth from SfM models, for which the error is hard to quantify and, due to the computational cost of SfM, remain hard to extend to city-scale. Urban driving datasets such as KITTI [53], Oxford RobotCar [54], DeepLoc [46] and NCLT [55] use robotic platforms for data collection. However, they either only cover multiple areas of the environment just once, or focus on revisiting the same route up to 100 times, limiting geographic extent. Finally, these datasets often derive ground truth localization from GPS and inertial filters, which do not achieve centimetre-level accuracy.

Recent work has used manual annotations to provide more accurate ground truth localization, either via manual verification of 2d-3d matches with existing SfM models [27], or by manually aligning LiDAR and SfM point clouds in publicly-available datasets [12]. However, these annotations have remained relatively small-scale efforts, providing around 100 000 localized images in the largest case. In contrast, we aim for a dataset that provides millions of accurately-localized images and LiDAR point clouds.

III. Pit30M: Global localization at city scale

We assume that the region where our SDV is located has previously been covered with an appearance database. This dataset should ideally have three characteristics:

1) **Diversity** in appearance is necessary to train models that learn to recognize the same site under changes due to weather, seasons, illumination, construction, occlusion, and dynamic objects in the scene.
2) **Scale** refers to the area spanned by the dataset. We want our dataset to cover an entire city, as this is the typical operational domain of a self-driving car.

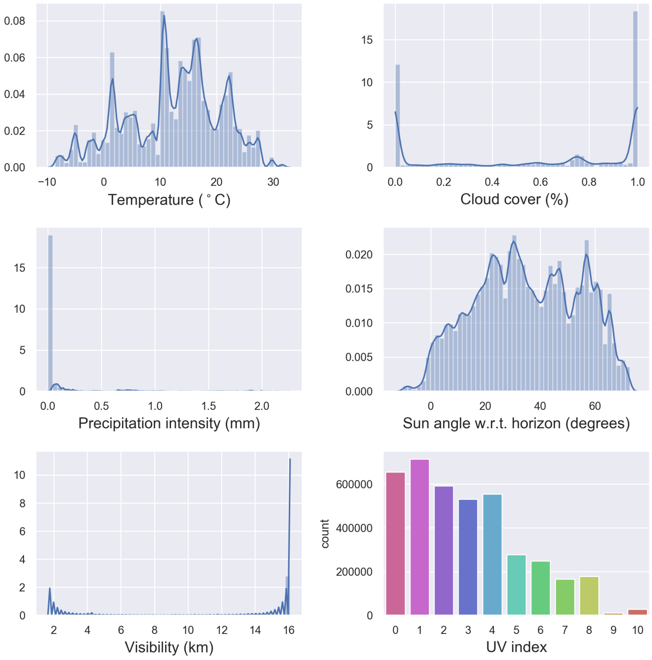


Fig. 2: **Probability density functions (PDFs) for metadata in Pit30M.** For a complete description of these tags, please refer to the supplementary material on the project website.

3) **Accurate ground truth** provides a clear evaluation benchmark for methods that achieve sub-metre accuracy. We can also use this ground truth as a supervisory signal to improve retrieval-based localization.

We used our self-driving fleet to collect a dataset of 1 343 trips and 30 million images and LiDAR point clouds. Our data was collected from Jan 2017 to Feb 2018 in the Metropolitan area of Pittsburgh, PA, USA. Our vehicles carry a Velodyne HDL-64 LiDAR sensor, a wheel odometer and an IMU, which we use to localize offline using vehicle dynamics and LiDAR registration against a pre-existing dense 3d scan of the scene geometry. These measurements are all fed to a commercial batch optimization system that has been validated to yield under 10cm localization error. We use an HD, global-shutter, colour camera located in the roof of the vehicle, facing forward at all times, which provides images at a resolution of 1 920 × 1 200 pixels. The horizontal and vertical fields of view are 78.6° and 52.5°, respectively. The intrinsic and extrinsic calibration parameters of the cameras and LiDAR (e.g. the LiDAR-to-camera rigid transformation) are computed and validated a priori using a standard setup consisting in fiducial targets and non-linear optimization. We also carry a consumer-grade GPS sensor. The continuous stream of points produced by the LiDAR is broken up into 100ms partitions and motion-compensated. The corresponding camera image is selected such that it is as close as possible to the moment that the LiDAR's rolling shutter passed through the middle of its FoV The synchronization is within a few milliseconds.

Pit30M is, to the best of our knowledge, the largest benchmark for large-scale localization to date both in terms of images, LiDAR readings, and accurate ground truth informa-

	Distance (km)	†Images	†Accurate GT	Geo span (km^2)	Time span	Sessions	(type) LiDAR
Pittsburgh 250k [56]	–	250	–	∼ 16	–	–	–
Tokyo 24/7 [25]	–	*600	**1	2.56	–	–	–
SFO Landmarks [51]	–	1 700	1 700	∼ 18	–	–	(Unspecified) ✓
DeepLoc [46]	4	2	2	0.015	1 day	10	(Unspecified) ✓
NCLT [55]	147	630	630	∼ 1	15 months	27	(Velodyne 32) ✓
Aachen [57]	–	5	[12] 5	∼ 1.5	–	–	–
CMU [58]	98.7	82	[12] 82	–	3 months	12	–
Oxford robotcar [54]	1 000	‡8 500	[12] 38	∼ 10	18 months	133	(2x SICK LMS) ✓
Pit30M (Ours)	25 000	30 000	30 000	∼ 50	14 months	1343	(Velodyne 64) ✓

TABLE I: **Comparison of datasets for large-scale visual localization.** †In thousands. ‡The dataset has over 20M images in total, but we consider only the frontal camera to make it comparable to our dataset. *Including synthesized views. **The number of query images localized manually.

tion. Table I provides summary statistics of existing datasets (described in the previous section) as well as ours, and Figure 1 shows the extent of our data. The proposed dataset includes over 25 000 km and 1 500 hours of driving, resulting in a benchmark that is one to two orders of magnitude larger than those used in previous work. Moreover, our dataset spans all seasons, diverse weather conditions (including rain, sleet, and snow), multiple times of day, including images taken at night and with low natural lighting, as well as construction and changes in buildings and pavement.

Large-scale metadata: Previous datasets have provided manual, trip-level metadata, typically with the goal of identifying challenging conditions for localization. For example, the Oxford dataset [54] provides 11 different tags including "sun", "clouds", "dusk", and "snow", and the CMU seasons dataset [12] includes tags for "park", "urban", "foliage', and "low sun", among others.

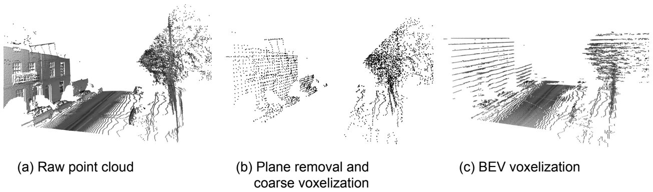
Unfortunately, trip-level tags can be ambiguous; e.g., the same trip may be sunny and cloudy at different times. Instead, we have collected more granular metadata using historical weather and astronomical data, which can be obtained at scale. In particular, we have collected weather via the `darksky.net` public API, and estimated the angle of the sun in the sky using the `skyfield` library. We have also used state-of-the-art LiDAR [59] and image [60] semantic segmentation to estimate the degree of background occlusion in our dataset. We showcase our labels by analyzing the results of our preliminary benchmark in Section V. Figure 2 shows probability density functions of some of our tags.

IV. Benchmarking Large-Scale Localization

We turn our attention to benchmarking retrieval-based localization approaches. Formally, let $\mathcal{Z}$ be either an image or LiDAR sweep point cloud, let $\mathcal{G}$ be the GPS pose and $\mathbf{y}$ the position of the SDV we are trying to infer.

In the retrieval localization setting, we start from a dataset of pre-localized sensor observations, and represent each full sensor reading as a vector $\mathbf{z} = f(\mathcal{Z})$, to obtain a database of vector-pose pairs $\mathcal{D} = \{(\mathbf{z}_1, \mathbf{y}_1), (\mathbf{z}_2, \mathbf{y}_2), \ldots, (\mathbf{z}_n, \mathbf{y}_n)\}$. Given an online sensor reading $\mathcal{Z}_q$, we first compute the global feature representation $\mathbf{z}_q = f(\mathcal{Z}_q)$ and infer the current pose via (high-dimensional) nearest neighbour search:

$$\hat{\mathbf{z}} = \underset{\mathbf{z}_i}{\operatorname{argmin}} \|\mathbf{z}_q - \mathbf{z}_i\|_2^2, \tag{1}$$



(a) Raw point cloud

(b) Plane removal and coarse voxelization

(c) BEV voxelization

Fig. 3: **LiDAR representations benchmarked in this work.** (a) Raw point cloud (not used by any method). (b) Point cloud after ground plane removal and downsampling to 4 096 points [35, 37, 39]. (c) BEV voxelization with intensities. We use the latter as input to CNNs.

and output the pose associated with the nearest dataset descriptor $\hat{\mathbf{z}}$. Since each observation is associated with a single vector $\mathbf{z}_i$, the obtained pose is expressed in a global coordinate frame.

We now introduce a couple of simple, yet effective retrieval convolutional networks for large-scale localization. By leveraging the supervision provided by our dataset (which is typical in an industrial setting), we show that strong convolutional backbones with simple pooling schemes can match the state of the art in image and LiDAR retrieval. This allows us to showcase the importance of and gains that are possible with our data, and gives us insights into the state of the art of retrieval-based localization in the context of self-driving.

Learning for retrieval: We rely on deep convolutional networks trained with a standard triplet loss, common in the retrieval literature:

$$\mathcal{L}_{\text{retrieve}} = \max\{d(\mathbf{a}, \mathbf{p}) - d(\mathbf{a}, \mathbf{n}) + m, 0\}, \tag{2}$$

where $d(\cdot, \cdot)$ is the Euclidean distance function, a triplet $(\mathbf{a}, \mathbf{p}, \mathbf{n})$ consists of three latent, ℓ_2-normalized embeddings produced by a network $f(\mathcal{Z})$. In this context, $\mathbf{a}$ is an "anchor" descriptor, $\mathbf{p}$ is a "positive" descriptor and $\mathbf{n}$ is a "negative" descriptor. We build our triplets such that the geo-location of the positive image is closer to the anchor than the negative image by a pre-defined margin of at least $m = 0.5$ in embedding feature space. In our experiment, we consider sensor readings within 1 metre to be positives, and within 2 and 4 metres to be negatives; notice that this fine-grained distinction is enabled by the accurate ground truth provided by our dataset.

We also make sure that the heading angle of the three images is within a range of 30°, so the images have over-

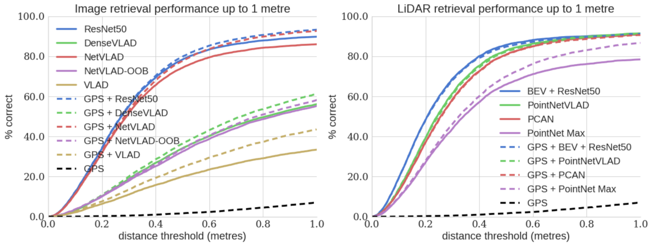


Fig. 4: **Performance of retrieval-based methods.** Left: Image retrieval results. Right: LiDAR retrieval results.

lapping fields of view. Finally, we ensure that the positive and negative samples do not come from the same trip as the anchor, which encourages the learned representations to be invariant to factors such as time of day, weather, and dynamic objects in the scene. We collect a triplet for each image in the dataset, and learn $f(\cdot)$ (in practice implemented as a Resnet-50 [61]) via backpropagation.

GPS + retrieval: We also consider using GPS to restrict the search area. This is a more realistic scenario in the context of self-driving, yet is less studied in the literature. Here, we collect a set of embeddings located within τ metres from the GPS reading, where τ is a tunable hyperparameter that we set based on the empirical error of our GPS measurements ($\tau = 20$m in our main results). We then perform retrieval as in the global case, only restricted to this region.

Bird's-Eye View (BEV) representation: Although it is straightforward to use CNNs with images in the above formulation, it is not immediately clear how that can be achieved when the input is a point cloud. For this, we introduce a representation that is conceptually simple yet achieves state-of-the-art results on publicly available benchmarks. (Please refer to our project website for detailed experiments on the Oxford Robotcar [54] dataset.) We preprocess the raw LiDAR point cloud as a BEV multi-channel representation by discretizing 3d space into an evenly-spaced voxelization of size $l \times w \times h$. Crucially, we treat the resulting voxelization as a 2d image by discretizing the z axis into c channels, which can be plugged directly into standard 2d CNNs. This representation has proven useful for efficient real-time LiDAR object detectors [62, 63], but here we show that it also produces competitive results on LiDAR retrieval. In contrast, previous retrieval work has operated directly on the point clouds, which are heavily downsampled for computational reasons [35, 37, 39]. We visualize different LiDAR representations in Figure 3.

V. Experiments

A. Evaluation protocol

We split Pit30M in terms of different trips. This partition protocol suits the self-driving scenario well, where a fleet typically maps the drivable areas beforehand, but new trips are faced with changes in visual appearance and new dynamic objects on the road. We randomly select 941 trips for training, 134 for validation and 268 for testing. We further select 10 000 random query sensor readings from the test partition to report our final localization metrics. We report the percentage of correctly localized queries for increasing distance ranges. This results in a monotonically increasing curve for increasing distance, with a hypothetical perfect localizer having a performance of 100 for every distance value.

% within (metres)	0.25m	0.50m	1.0m	5.0m	average	median
GPS	0.4	1.7	6.2	73.7	4.20	3.40
Image-based methods						
VLAD	8.59	20.01	33.44	51.40	2401.33	3.95
DenseVLAD	14.50	34.12	56.15	77.82	843.98	0.81
NetVLAD-OOB	13.85	32.47	55.27	82.38	476.33	0.84
NetVLAD	42.57	74.38	86.07	87.60	577.69	0.29
Resnet50 (Img)	45.40	78.51	90.39	91.87	418.64	**0.27**
GPS + VLAD	10.44	24.78	43.58	78.80	3.53	1.26
GPS + DenseVLAD	15.33	36.47	61.33	90.78	2.05	0.72
GPS + NetVLAD-OOB	14.47	33.88	58.24	90.87	2.05	0.77
GPS + NetVLAD	43.53	77.78	92.73	96.80	0.92	0.28
GPS + Resnet50 (Img)	45.74	79.79	93.49	97.06	0.86	**0.27**
LiDAR-based methods						
PointNet Max	36.53	66.46	78.82	80.77	944.28	0.34
PointNetVLAD	51.60	83.29	91.89	93.16	322.85	0.24
PCAN	48.95	81.88	91.51	92.47	339.96	0.26
BEV + Resnet50	60.17	86.08	91.39	92.56	353.27	**0.20**
GPS + PointNet Max	37.73	70.03	86.70	91.95	1.78	0.32
GPS + PointnetVLAD	50.77	82.43	91.27	94.35	1.42	0.25
GPS + PCAN	48.43	80.77	90.63	93.39	1.55	0.26
GPS + BEV + Resnet50	59.38	84.88	91.25	93.74	1.51	**0.21**

TABLE II: **Detailed localization results for retrieval-based approaches.** We report the percent of correct predictions within different distance thresholds, and mean and median over the entire query set. Top: Image-based methods. Bottom: LiDAR-based methods.

B. Benchmarked methods

Camera-based methods: From classical methods, we consider SIFT-based VLAD [24] and DenseVLAD [25], a variant of VLAD specifically designed for very dense datasets with high visual variability. For these two methods, we learn the visual vocabulary on a subset of 5M images from the training set, and use 128 SIFT clusters. We also consider NetVLAD [1], a deep learning approach that uses VGG-16 convolutional feature maps as local features, and adds a learnable VLAD-based pooling layer. We use the TensorFlow implementation of Cieslewski *et al.*[1] [64]. Since previous work [48] has relied on the best model from [1] trained on Pittsburgh[2] for global localization, we also benchmark this pre-trained network out-of-the-box on our dataset, so as to provide context about the performance of a strong baseline in the field. We call this method NetVLAD-OOB.

LiDAR-based baselines: We consider PointNet-Max [36], PointNetVLAD [35], and PCAN [37] and train them on the Pit30M dataset. We use the publicly-available implementations of PointNetVLAD[3] [35]. We also implemented our own version of PCAN [37]. Note that we do not evaluate the state-of-the-art LPD-Net [39] on Pit30M due to the lack of a public implementation. However, as shown on the project website,

[1] https://github.com/uzh-rpg/netvlad_tf_open
[2] https://www.di.ens.fr/willow/research/netvlad/
[3] https://github.com/mikacuy/pointnetvlad

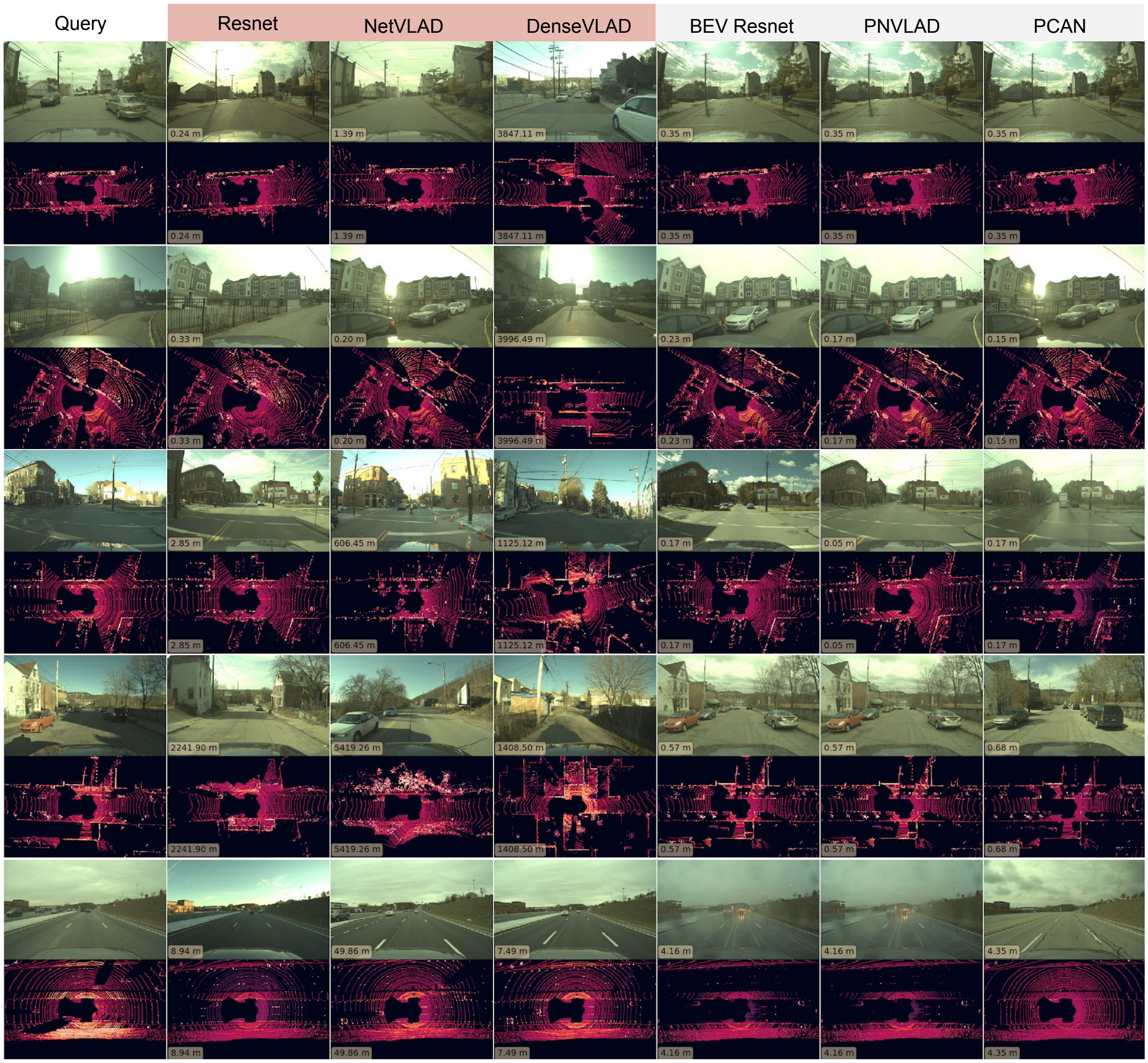


Fig. 5: **Qualitative results under exhaustive search.** Left: Query. Middle: Image retrieval method. Right: LiDAR retrieval methods.

benchmarking on the Oxford datasets shows that our BEV + Resnet50 method is competitive with this strong baseline.

C. Results

Quantitative results: We show the results of our retrieval-based benchmark in Figure 4, and detailed results in Table II. Regarding image retrieval, while VLAD is outperformed by both DenseVLAD and NetVLAD–OOB, the differences between the hand-crafted DenseVLAD and deep NetVLAD are rather small, and no clear winner emerges. Our image-based Resnet50 baseline performs on par with NetVLAD. In LiDAR retrieval, BEV emerges as the best method overall, outperforming all its image counterparts and other LiDAR methods.

Qualitative results: We show some qualitative results for the retrieval-based methods on Pit30m in Figure 5. We focus on challenging queries with glare, low sun angle, snow and rain. For example, the second row shows a query with glare, where both ResNet and NetVLAD manage to localize correctly, but DenseVLAD fails. The bottom row shows an interesting example where snow makes LiDAR highly reflective; however all LiDAR methods are able to localize within 5m.

D. Analysis

We use the annotations provided by Pit30M to analyse the results of our benchmarking. We focus on understanding our best-performing methods, GPS+Resnet-50 (images) and GPS+BEV+Resnet-50 (LiDAR).

First, we observe in Figure 6 (left), that large GPS error tends to cause large overall localization errors for both images and LiDAR. This is not surprising, since we limit the search radius to 20m around the GPS prediction. In Figure 7, we show the pairwise Pearson correlation coefficient between our labels and failure cases of image and LiDAR retrieval (a query is considered a "failure" if its error exceeds 2m), after removing cases where GPS error is above 20m. While we observe that, for example, image error is correlated with image occlusion, and image and LiDAR errors are highly

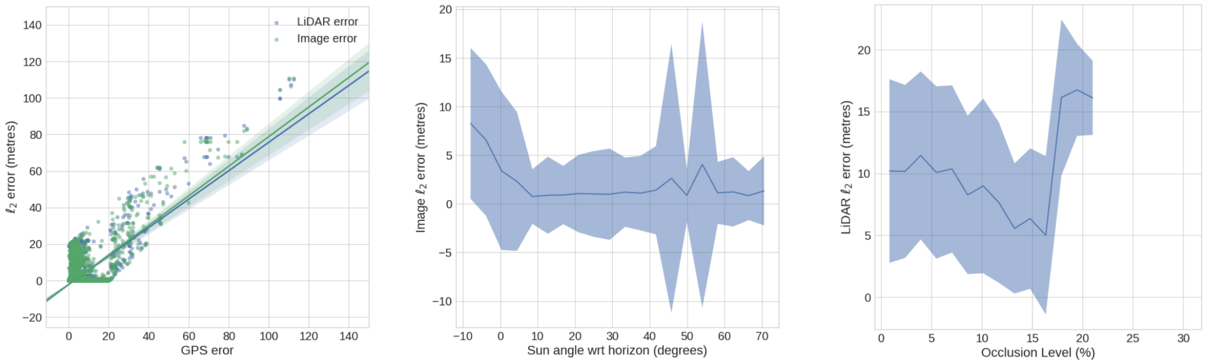


Fig. 6: **Examples of analysis enabled by the Pit30M metadata.** Left: GPS error is correlated with both image and LiDAR localization error. Middle: Image localization error vs. sun angle in the horizon (altitude angle). We observe a smooth error increase as the sun gets closer to the horizon. Right: We plot LiDAR queries with more than 1 metre of error (failure cases) against LiDAR occlusion. We observe a sharp spike in error when between 15 and 20% of points are assigned to dynamic objects.

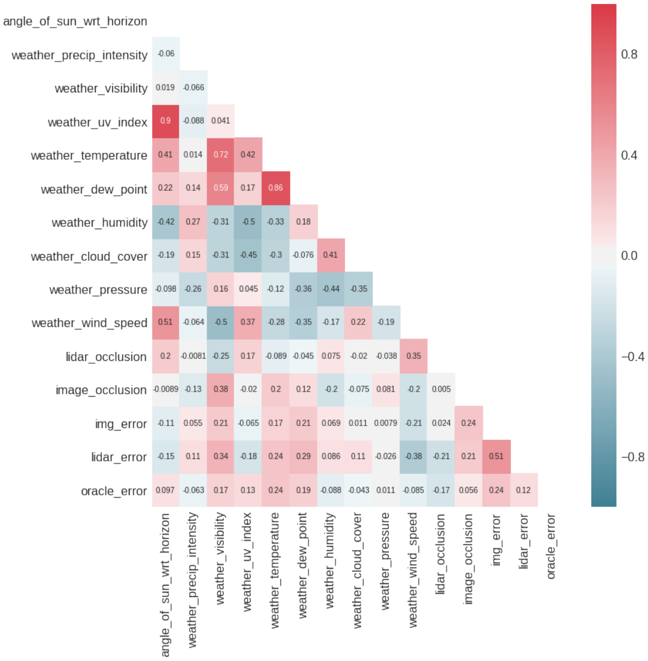


Fig. 7: **Pairwise correlations between metadata in Pit30M and error of different methods.** ”Oracle error” stands for a hypothetical method that is able to pick the best of either image or LiDAR prediction for each query.

correlated, this analysis is somewhat limited, as it is only able to capture linear correlations. Thus, we further examine image error vs. sun angle in Fig. 6 (middle); here, we observe increased errors during dawn and twilight, and no effect during daytime. We also examine LiDAR error vs. occlusion, and observe a spike in errors when 15-20% of points are assigned to dynamic objects.

VI. Conclusions

We have introduced Pit30M, a novel large-scale dataset for image and LiDAR localization, and studied retrieval-based methods in the context of self-driving cars. Our dataset provides extensive metadata and sub-metre ground truth, and allows researchers to study accurate global localization at city-scale. We have also provided an initial benchmark with multiple methods for visual and LiDAR localization, and in the process shown that strong modern convolutional backbones perform remarkably well in this scenario. Our analysis also hints at future research directions using multi-sensor fusion, and highlights challenging scenarios for localization. Our dataset and metadata are available on the Pit30M project website. Additionally, we plan to set up an evaluation server to track the progress made by the community in this field.

References

[1] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, “NetVLAD: CNN architecture for weakly supervised place recognition,” in *CVPR*, 2016.

[2] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski, “Building Rome in a day,” in *ICCV*, 2009.

[3] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, “D2-Net: A Trainable CNN for Joint Detection and Description of Local Features,” in *CVPR*, 2019.

[4] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof, “From structure-from-motion point clouds to fast location recognition,” in *CVPR*, 2009.

[5] T. Sattler, B. Leibe, and L. Kobbelt, “Fast image-based localization using direct 2d-to-3d matching,” in *ICCV*, 2011.

[6] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua, “Worldwide pose estimation using 3d point clouds,” in *ECCV*, 2012.

[7] T. Sattler, M. Havlena, F. Radenovic, K. Schindler, and M. Pollefeys, “Hyperpoints and fine vocabularies for large-scale location recognition,” in *ICCV*, 2015.

[8] L. Liu, H. Li, and Y. Dai, “Efficient global 2d-3d matching for camera localization in a large-scale 3d map,” in *ICCV*, 2017.

[9] M. Havlena and K. Schindler, “Vocmatch: Efficient multiview correspondence for structure from motion,” in *ECCV*, 2014.

[10] F. Camposeco, A. Cohen, M. Pollefeys, and T. Sattler, “Hybrid scene compression for visual localization,” in *CVPR*, 2019.

[11] S. Lynen, B. Zeisl, D. Aiger, M. Bosse, J. Hesch, M. Pollefeys, R. Siegwart, and T. Sattler, “Large-scale, real-time visual-inertial localization revisited,” *arXiv*, 2019.

[12] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, *et al.*, “Benchmarking 6dof outdoor visual localization in changing conditions,” in *CVPR*, 2018.

[13] K. Yoneda, H. Tehrani, T. Ogawa, N. Hukuyama, and S. Mita, “Lidar scan feature for localization with highly precise 3-d map,” in *IV*, 2014.

[14] F. Pomerleau, F. Colas, R. Siegwart, *et al.*, “A review of point

cloud registration algorithms for mobile robotics," *Foundations and Trends in Robotics*, 2015.

[15] J. Levinson, M. Montemerlo, and S. Thrun, "Map-Based Precision Vehicle Localization in Urban Environments," *RSS*, 2007.

[16] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," *ICRA*, 2010.

[17] W. Ryan W. and E. Ryan M., "Visual localization within LIDAR maps for automated urban driving," *IROS*, 2014.

[18] I. A. Barsan, S. Wang, A. Pokrovsky, and R. Urtasun, "Learning to localize using a lidar intensity map," in *CoRL*, 2018.

[19] X. Wei, I. A. Barsan, S. Wang, J. Martinez, and R. Urtasun, "Learning to Localize Through Compressed Binary Maps," in *CVPR*, 2019.

[20] G. Wan, X. Yang, R. Cai, H. Li, H. Wang, and S. Song, "Robust and Precise Vehicle Localization based on Multisensor Fusion in Diverse City Scenes," *ICRA*, 2018.

[21] D. G. Lowe *et al.*, "Object recognition from local scale-invariant features." in *ICCV*, 1999.

[22] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *ECCV*, 2006.

[23] D. Filliat, "A visual bag of words method for interactive qualitative localization and mapping," in *ICRA*, 2007.

[24] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *CVPR*, 2010.

[25] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 place recognition by view synthesis," in *CVPR*, 2015.

[26] J. Knopp, J. Sivic, and T. Pajdla, "Avoiding confusing features in place recognition," in *ECCV*, 2010.

[27] 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 *CVPR*, 2017.

[28] F. Radenović, G. Tolias, and O. Chum, "Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples," in *ECCV*, 2016.

[29] G. Tolias, R. Sicre, and H. Jégou, "Particular object retrieval with integral max-pooling of cnn activations," in *ICLR*, 2016.

[30] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *ECCV*, 2010.

[31] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *ICRA*, 2009.

[32] L. He, X. Wang, and H. Zhang, "M2DP: A novel 3D point cloud descriptor and its application in loop closure detection," in *IROS*, 2016.

[33] A. Dewan, T. Caselitz, and W. Burgard, "Learning a local feature descriptor for 3d lidar scans," in *IROS*, 2018.

[34] R. Klokov and V. Lempitsky, "Escape from cells: Deep kd-networks for the recognition of 3d point cloud models," in *ICCV*, 2017.

[35] M. A. Uy and G. H. Lee, "PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition," in *CVPR*, 2018.

[36] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3d classification and segmentation," *arXiv preprint arXiv:1612.00593*, 2016.

[37] W. Zhang and C. Xiao, "PCAN: 3d attention map learning using contextual information for point cloud based retrieval," in *CVPR*, 2019.

[38] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *NeurIPS*, 2017.

[39] Z. Liu, S. Zhou, C. Suo, P. Yin, W. Chen, H. Wang, H. Li, and Y.-H. Liu, "LPD-Net: 3d point cloud learning for large-scale place recognition and environment analysis," in *CVPR*, 2019.

[40] 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 *CVPR*, 2013.

[41] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *ICCV*, 2015.

[42] E. Brachmann, F. Michel, A. Krull, M. Ying Yang, S. Gumhold, *et al.*, "Uncertainty-driven 6d pose estimation of objects and scenes from a single RGB image," in *CVPR*, 2016.

[43] E. Brachmann and C. Rother, "Learning less is more-6d camera localization via 3d surface regression," in *CVPR*, 2018.

[44] T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixe, "Understanding the limitations of cnn-based absolute camera pose regression," in *CVPR*, 2019.

[45] A. Valada, N. Radwan, and W. Burgard, "Deep auxiliary learning for visual localization and odometry," *arXiv preprint arXiv:1803.03642*, 2018.

[46] N. Radwan, A. Valada, and W. Burgard, "Vlocnet++: Deep multitask learning for semantic visual localization and odometry," *arXiv preprint arXiv:1804.08366*, 2018.

[47] J. L. Schönberger, M. Pollefeys, A. Geiger, and T. Sattler, "Semantic visual localization," *JPRS*, 2018.

[48] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *CVPR*, 2019.

[49] J. F. Henriques and A. Vedaldi, "Mapnet: An allocentric spatial memory for mapping environments," in *CVPR*, 2018.

[50] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz, "Geometry-aware learning of maps for camera localization," in *CVPR*, 2018.

[51] D. M. Chen, G. Baatz, B. Girod, R. Grzeszczuk, K. Koser, S. Tsai, R. Vedantham, T. Pylvanainen, K. Roimela, X. Chen, *et al.*, "City-scale landmark identification on mobile devices," in *CVPR*, 2011.

[52] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi, "Visual place recognition with repetitive structures," in *CVPR*, 2013.

[53] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *CVPR*, 2012.

[54] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *The International Journal of Robotics Research*, vol. 36, 2017.

[55] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, "University of Michigan North Campus long-term vision and lidar dataset," *IJRR*, 2016.

[56] A. Torii, J. Sivic, M. Okutomi, and T. Pajdla, "Visual place recognition with repetitive structures," *TPAMI*, 2015.

[57] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt, "Image retrieval for image-based localization revisited." in *BMVC*, 2012.

[58] A. Bansal, H. Badino, and D. Huber, "Understanding how camera configuration and environmental conditions affect appearance-based localization," in *IV*, 2014.

[59] C. Zhang, W. Luo, and R. Urtasun, "Efficient convolutions for real-time semantic segmentation of 3d point clouds," in *3DV*, 2018.

[60] M. Bai and R. Urtasun, "Deep watershed transform for instance segmentation," in *CVPR*, 2017.

[61] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[62] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *CVPR*, 2017.

[63] W. Luo, B. Yang, and R. Urtasun, "Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net," in *CVPR*, 2018.

[64] T. Cieslewski, S. Choudhary, and D. Scaramuzza, "Data-efficient decentralized visual slam," in *ICRA*, 2018.