2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 24-27 September 2018, Nantes, France

Evaluation Criteria for Inside-Out Indoor Positioning Systems based on Machine Learning

Christoffer Löffler[*], Sascha Riechel[*], Janina Fischer[*], and Christopher Mutschler[*†]
{ christoffer.loeffler | riechesa | janina.fischer | christopher.mutschler } @iis.fraunhofer.de

[*]Machine Learning and Information Fusion Group
Fraunhofer Institute for Integrated Circuits IIS

[†]Machine Learning and Data Analytics Lab
Friedrich-Alexander University Erlangen-Nürnberg (FAU)

***Abstract*—Real-time tracking allows to trace goods and enables the optimization of logistics processes in many application areas. Camera-based inside-out tracking that uses an infrastructure of fixed and known markers is costly as the markers need to be installed and maintained in the environment. Instead, systems that use natural markers suffer from changes in the physical environment. Recently a number of approaches based on machine learning (ML) aim to address such issues.**

This paper proposes evaluation criteria that consider algorithmic properties of ML-based positioning schemes and introduces a dataset from an indoor warehouse scenario to evaluate for them. Our dataset consists of images labeled with millimeter precise positions that allows for a better development and performance evaluation of learning algorithms. This allows an evaluation of machine learning algorithms for monocular optical positioning in a realistic indoor position application for the first time. We also show the feasibility of ML-based positioning schemes for an industrial deployment.

I. Introduction

Real-time tracking is used to trace goods and optimize logistics processes in many applications such as large indoor warehouses. Mobile, monocular optical positioning approaches meet the requirements of exact real-time positioning and large-scale tracking Their cheaper cost and robustness surpasses other, e.g. RF- and LIDAR-based systems.

Camera-based inside-out tracking, i.e., self-positioning, often uses an infrastructure of fixed, known markers installed and maintained in the environment [1]–[5]. Besides time-consuming solutions that use manually installed hand-crafted markers, there are also methods that use natural features, e.g. edges and walls [6], [7]. However, both approaches require expensive feature detection [8] and matching with a reference database at runtime which not only consumes much CPU time. Additionally, changes in the physical environments result in a poor position quality if we use natural markers.

Many optical indoor positioning approaches are limited in their usability as they rely on the assistance from additional knowledge or sensors. This includes location references such as 3D models, depth information in RGB-D images, detailed pre-recorded image dictionaries, synthetic markers, projected targets (e.g. laser grid), or additional sensors such as IMUs [9]. Such combinations suffer from high setup effort and cost, high computational load at runtime (e.g. image dictionary), limitation by distance (e.g. projection) or the need for additional sensors (RGB-D). While there are methods that may help, e.g. direct [10] and sparse [11] SLAM (Simultaneous Localization and Mapping), the basic problems of occlusion, installation, processing costs, and degeneration remain. The construction of scene models through dense and sub-pixel multi-view stereo reconstruction [12], [13] using direct image alignment, or generative models [14], i.e., models that exploit combinations of appearance, have lower CPU requirements [15]. But they need a lot of memory to store the map information.

Recent advances in machine learning, such as regression forests [16]–[18] and deep convolutional neural networks [8], [19]–[22], may become an alternative. Regression forests directly learn the positioning from pixels to world points, using feature detection methods (like SIFT and ORB) and RANSAC for camera pose estimation. Instead, deep learning (DL) delegates feature extraction and matching to a deep neural network that uses a cascade of convolutional operators, i.e., convolutional layers, to extract meaningful information, e.g. features and estimated positions. Machine learning approaches need thousands of images from the target area, each labeled with its pose, i.e., position and orientation. During a training phase we capture significant features and learn their relation to positions. This enables the prediction of poses from previously unseen images in a subsequent navigation phase.

However, the evaluations that are proposed in these approaches often lack completeness from an indoor positioning point of view as many aspects (e.g. changing environments and lighting conditions) remain open. Hence, it is difficult to judge the real-world applicability of such approaches. This paper makes the first attempt to transfer this ML research into a real localization application. We analyze the general requirements of training datasets for image-based location estimation with regard to homogeneity and precision for an indoor localization task from the logistics context.

The remainder of the paper is structured as follows. Sec. II gives an overview of related evaluation datasets and commonly used evaluation methods. In the following, we present our main contributions. First, we provide a common ground to properly evaluate ML-based positioning schemes in practice in Sec. III. Second, while available datasets are either outdated, not comprehensive or compile datasets based on a specific topic (e.g. landmark retrieval) we introduce the *Warehouse*

978-1-5386-5635-8/18/$31.00 ©2018 IEEE

dataset, which models an industrial localization use-case and that includes highly precise reference labels in Sec. IV. Third, we evaluate state of the art ML-based positioning approaches based on our dataset using our evaluation criteria in Sec. V. Sec. VI concludes.

II. Related Work

We first discuss publicly available datasets (Sec. II-A) before we discuss how ML-based positioning schemes are usually evaluated in practice (Sec. II-B).

A. Datasets

We can categorize the available datasets for optical localization in outdoor, small indoor, and large indoor areas.

Outdoor datasets are widely adopted in the DL literature [8], [16], [20], [23], [24]. The *Cambridge Landmark* dataset [8] uses five large-scale urban outdoor scenes (areas from 875 m^2 to 50,000 m^2) with considerable clutter, e.g. pedestrians and vehicles, and under various weather conditions. Structure from Motion (SfM) calculates 3D models from down-sampled videos (which results in a sparse spacing of about 1m between camera positions) to estimate position labels. However, while such datasets are optimal for outdoor localization, e.g. navigating in urban environments, they are inappropriate to evaluate indoor localization in industrial environments due to their different lighting, volatility and feature conditions.

Small scale indoor datasets range from small rooms to multi-floor areas, connecting rooms by corridors, and feature offices. The *7 Scenes* dataset [18] includes seven different small-scale (3D scans up to 6 m^3) indoor scenes. The SLAM-based KinectFusion system [25] takes images with a resolution of 640 × 480 and depth information. However, the scenes mostly include highly textured areas, e.g. offices and kitchens, that lack global ambiguities [24]. The *University* dataset [26] includes five different scenes, e.g. conference rooms and offices, connected by corridors, resulting in a total of 9,694 images for training and 5,068 for testing. However, the ground truth is obtained by manual walks through the scenes using *Google Tango*, which offers a poor accuracy of 6cm in small scenes to 3m in large scenes [27]. Moreover, the pose-graph optimization framework that obtains a globally consistent map introduces location constraints.

Large indoor datasets not only cover rooms and offices but also complete university buildings and shopping malls.

The Baidu [28] dataset covers 5,000 m^2 of a shopping mall with 682 training images that either capture the store fronts or the corridor, and 2,296 query images at random positions. As the images have been captured at different times there are also moderate appearance changes. Ground truth is obtained using a highly precise LiDAR. To create 3D models 20% of the training images were annotated to guide the labeling of other images. However, the authors recorded only one route through the mall with a low number of training images.

The Matterport3D [29] dataset that covers large indoor scenes, e.g. apartments and offices, is used for scene understanding and semantic segmentation. The Wijmans [30] dataset includes large indoor areas (up to 34,000 m^2) recorded with a 3D scanner and consists of 277 RGB-D panoramic images captured on five floors of a university building. It includes significant challenges, such as repetitive patterns (stairs), textures (walls), building structures (windows), and moved furniture or moving people. However, both datasets lack recordings with bigger environmental changes for evaluating optical localization, e.g. many moved objects, and changed illumination.

The InLoc [31] dataset extends [30] with 329 images from two of the five floors. The images were matched to the most similar reference RGB-D image from the previous dataset. Then their poses were calculated (using P3P-RANSAC and bundle adjustment). As the new images were recorded months later at different times of day they include further changes, e.g. moved furniture, occluders (people), and different illumination. However, the data not only stems largely from an office environment, which is not applicable to industrial applications. ML, and DL in particular, needs considerably more training data to learn important features from changed environments.

Another large-scale indoor dataset [32] includes two separate recordings for 4 small indoor scenes (12 rooms), e.g. offices and apartments, combined to a larger area. Ground truth position labels were obtained from the RGB-D information with methods of reconstruction and global bundle adjustment. However, the indoor locations are comparably small and the dataset lacks global ambiguities.

The *TU Munich Large-Scale Indoor* [24] dataset includes 1,314 images, spaced 1m apart, covering a university floor of 5,575 m^2. *NavVis M3* provides sub-cm accuracy for ground truth positions. While the dataset is challenging as it contains repetitive structures and textureless walls (on which methods based on feature detectors suffer), it lacks dynamic appearances changes, e.g. moved structures and occlusion.

Our proposed dataset includes specially crafted test trajectories that help to answer specific questions in the evaluation of novel algorithms, e.g. generalization to unseen images, areas with varying proximities to static objects and robustness in both dynamic and homogenic environments, and focuses on industrial applications.

B. Evaluation Criteria

Not only ML-based approaches but any fingerprinting-based locating scheme determines the position of mobile tags after it initially has seen a training dataset with ground truth positions. However, as commonly applied evaluation strategies vary strongly it is hard to compare existing implementations based on their evaluation results.

Concrete quantifications for the dimensions of the data are, e.g. generalizable criteria for sampling test data from a training dataset, i.e., point or blockwise sampling and gaps, different spacings of separate test recordings in and around a given training dataset, i.e., for interpolation and generalization analysis, or specific tests with illumination and dynamic environments, i.e., for evaluation of the positioning robustness.

Some datasets provide different data for training and test. While [8] uses separate, partially overlapping paths for testing and training, it does not introduce a systematic test approach to gain insights on particular aspects. Training data in [28] was recorded methodically on straight paths at equidistant intervals, while the query dataset simply uses images from random places in the target area. [31] uses test images from mobile phones that cover two out of five areas from the training dataset. However, all of those approaches differ in their nature and none of them introduces a systematic data selection based on localization requirements.

ML-based fingerprinting approaches on RF-signals, e.g. WiFi or cellular, often record data in varyingly spaced grids [33], based on floor plans with (sub) room-level position accuracy [34], [35], randomly, e.g. on street corners [36] or on random walks. They also split the data in training and test sets differently. Test data is either sampled randomly [35], [36] or along parallel lines through the grid [33], [37]. [34] allows spatial (room precision) and temporal (months apart) sampling of test data. However, none of the above approaches describes the test data selection criteria sufficiently or derives a systematic strategy that could be applied for optical localization. As they are not equivalent the results also differ.

Furthermore, also the applied accuracy metrics differ. Commonly used is the percentage of predictions below an error threshold [18], e.g. for small-scale datasets <5cm for the position and <5° for the orientation. This is problematic as it is less meaningful than, e.g. the median error. Others [24] [8] use the median position and orientation error. This reduces the result to one value, leaving out errors over time, error distribution, axis-specific error values, and location-specific behavior, i.e., problematic untextured areas.

Our contribution aims at providing test criteria that enable unified performance evaluations of fingerprinting- and ML-based locating schemes. We provide definitions on relevant trajectorial data and concise methods to sample data for training and test from recorded data sets.

III. Evaluation Criteria

A. Properties

Training and test data sets can be sampled specifically to evaluate certain algorithmic properties of ML-based positioning schemes. We define such properties as follows.

Generalization. Algorithms may be able to predict previously unseen positions well if they are close to previously seen positions, but fail to generalize to areas that are further apart, i.e., they fail to interpolate between known fingerprints. This can also be the result of overfitting.

Environmental scaling. The positioning performance can differ over area scales, which are known to correlate with model size or prediction quality in many algorithms. The positioning schemes then may have a much larger positioning error for large environmental scales.

Scale transition. Having both small and large scale areas in the same dataset may affect performance, e.g. when features have to be learned scale-invariant.

Volatility. Another type of overfitting, i.e., to input data, happens if the algorithm learns a training set's volatile or mobile features and later fails to generalize to changed features or previously unseen test samples.

Ambiguity. The environment may include ambiguous features, i.e., that are repetitive or untextured, affecting the accuracy of the prediction. As a result an algorithm may estimate a mixture of those positions.

Motion artifacts. In real-world use cases, images may show challenging blur, unsteady angles, and new view points. The artifacts are manifold and include, e.g. the typical bobbing of human walking motion or swift turning of vehicles in curves. Such motion artifacts have an influence on the features of the image as their features are commonly not part of the training data. This may cause a poor positioning performance.

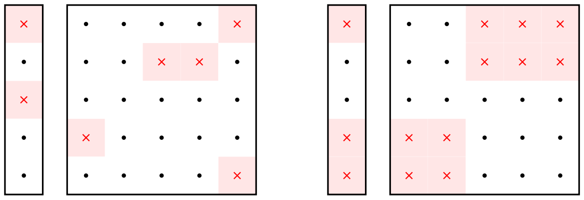
These properties represent a set of any positioning scheme's crucial performance indicators under hard real-world conditions. We will use them to construct benchmark datasets, i.e., property-specific training and testing data.

B. Data Recording and Sampling

The datasets can either be recorded in a uniformly distributed grid over the target area, or along trajectories. This choice depends on the technologies involved in recording and processing, and (to a large part) on cost and time constraints.

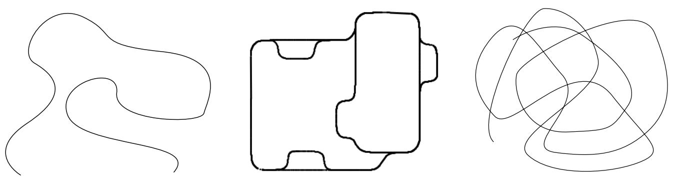
Grid-based approaches record data uniformly and with variable density in the covered area. We may sample training and test data sets from it either element-wise or block-wise, see Fig. 1. The scale-related properties, i.e., environmental scaling and scale transition, and feature ambiguity, i.e., ambiguous walls and floors, are testable using any of the two sampling strategies.

The element-wise random sampling in Fig. 1a is commonly used to evaluate the general functionality of a positioning scheme. However, we need to be careful as testing may easily degenerate to benchmarking the system against its own training samples. The sampled data from densely recorded grids is too similar to the training data. In contrast, the block-wise sampling in Fig. 1b allows for better insights regarding interpolation or generalization, depending on the scale of the block's size and its location, i.e., the larger the better.



(a) Random sampling. (b) Random block sampling.

Figure 1: Grid-based recording: training data (black dots on white background) and test data (red crosses).



(a) Vehicular motion (random, small tail size).

(b) Restricted vehicular motion.

(c) Human motion (random, large tail size).

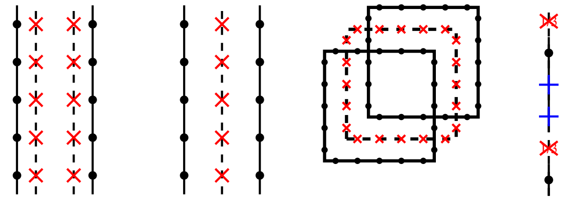
Figure 2: Motion of objects.

However, grid-based recording is a bad choice for the remaining two properties, i.e., volatility and motion artifacts. First, a highly volatile scenario requires many separate recordings. This can be very time-consuming and costly as it does not scale with the grid density and covered area. Second, typical motion artifacts practically cannot be tested (as there are none).

Recorded trajectories on the other hand not only capture volatility more easily but also allow to evaluate for motion artifacts. The motion artifacts depend on the tracked object, see Fig. 2. Vehicular motion (Fig. 2a) often includes blur due to angular shifts of the camera and stuttering due to drive controls. However, motion artifacts are often reduced for vehicles on restricted paths (Fig. 2b) such as automated conveyer systems. The unpredictable human random walk (Fig. 2c) includes most artifacts, e.g. steps cause vertical bobbing, shaking and blur, and the camera pose is much more random and difficult to stabilize.[10] If sequential positioning schemes, i.e., that use a series of images over time, are evaluated, it is important to capture trajectories that cover such motion models.

Recorded trajectories in general are less dense than uniformly recorded grids and less time consuming and costly to record. Therefore, they are the practical option for recording largely volatile scenarios. Most of the other properties, i.e., the scale-related ones and ambiguity, can also be tested by simply transfering sampling strategies from the grid-based recordings to the trajectory strategy.

To evaluate for generalization it is best to use additional recordings, see Fig. 3. Figs. 3a and 3b show tests for interpolation between learned data. While tests that sample from positions closer to training data in Fig. 3a are more fine-grained, the equidistant spacing of the test samples in Fig. 3b may additionally cause reduced precision if predictions are on either training trajectories (overfitting). The 2D trajectory in Fig. 3c is the equivalent for the grid-based block sampling of tests in Fig. 1b. The trajectory in Fig. 3d represents sampling of training and test data from a single trajectory. The gaps in both training and test, indicated by blue crosses, increase the difficulty further, in sparsely recorded trajectories, or may be necessary to prevent over-representation, i.e., overfitting to densely clustered training samples. This sampling strategy is especially useful for ML-based approaches that make use of sequential, i.e., trajectorial, training data.



(a) Close interpolation.

(b) Precision of generalization.

(c) 2D trajectory.

(d) Gaps in data.

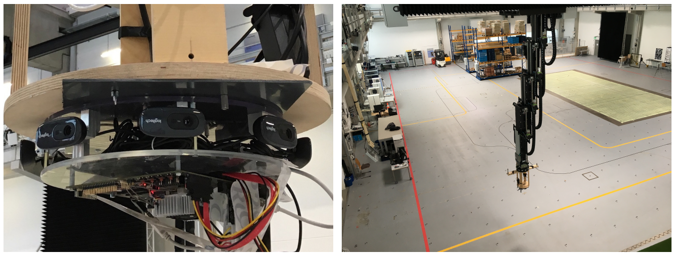
Figure 3: Sampling strategies for trajectorial data (training data with black dots, test data with red crosses). The strategies show: sampling in (3a) tests close range interpolation, (3b) tests precision using a centered trajectory, (3c) transfers these trajectories to 2D scenarios. (3d) shows how sampling from only one trajectory, including gaps (blue crosses) can be done.

IV. Warehouse Dataset

With our indoor logistics *Warehouse* dataset we aim at providing a solid basis for the development and evaluation of ML-based positioning schemes and criteria to tackle the current challenges [38]. *Warehouse* includes different scenarios that allow a detailed analysis of positioning schemes based on the properties that we lined out in Sec. III.

Our dataset covers an area of 1,320 m^2 and 464,804 images with a size of 640 × 480 pixels. Each image is labeled with a sub-millimeter position and sub-degree orientation that we acquired using an optical laser-based Nikon iGPS reference system. We recorded the images using a platform with 300mm diameter that carries eight cameras (calibrated Logitech C270) facing in different directions, see Fig. 4a. The distance between the cameras is a few centimeters, which we calibrated out. We mounted the cameras on a programmable 3D-positioning system (PosSys), see Fig. 4b, and moved it through the Fraunhofer IIS L.I.N.K. (localization, identification, navigation, communication) test center.

In the test center we model a complex warehouse scenario that poses realistic challenges to optical positioning schemes. It includes three high-level racks (see Fig. 5a) and open spaces (see Fig. 5b) with complex, volatile structures (boxes, mobile work benches, and mobile wall segments, see Fig. 5c) and ambiguous elements, e.g. repeated structures and homogeneous texturing (e.g. white walls, large black wall segments, unmarked floors, see Fig. 5d). The lighting conditions vary among combinations of artificial and natural light.

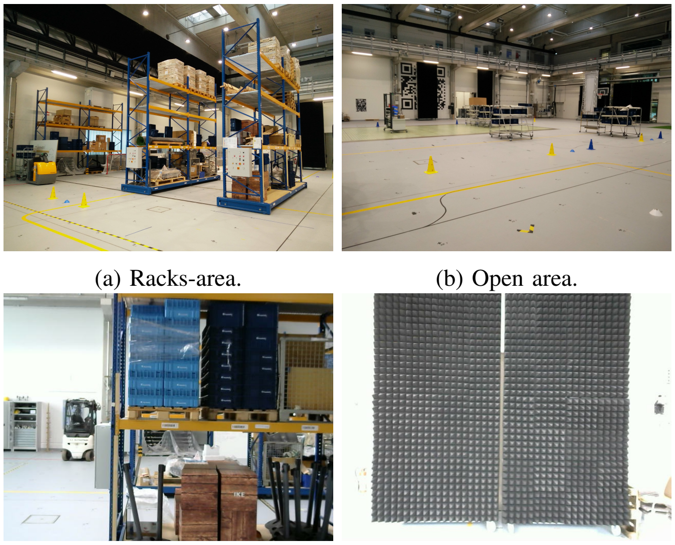


(a) Recording device.

(b) 3D-PosSys.

Figure 4: Recording platform with eight cameras, spaced at 45°, mounted on 3D-positioning system.

[10] Similarly, the big vertical freedom and extreme velocities of drones introduce very high levels of motion artifacts.



(a) Racks-area. (b) Open area.

(c) Volatile features. (d) Repetitive features.

Figure 5: The L.I.N.K. warehouse environment.

Our dataset includes separate recordings of various trajectories. We designed two trajectories for training and seven trajectories for testing, see Table I for an overview.

We recorded the training data in horizontal and vertical meandering trajectories through the whole hall, see Fig. 7a, to have a dense recording that is similar to a grid pattern. This gives the user the ability to sample freely from the training data, e.g. only from one of the camera angles facing a white wall, or from eight cameras between two high-level racks.

We generated testing data with the intent to evaluate the specific properties that we lined out in Sec. III-A, and that require their own test trajectories. The blue line in Figs. 7b-h show the testing trajectories in the warehouse.

We recorded three trajectories to test for generalization and follow the strategy from Fig. 3. Our dataset includes multiple parallel trajectories (10cm apart) that pass between horizontal or vertical training trajectories. These parallel trajectories are recorded once for the open area (Fig. 7b) and once for the racks-area (Fig. 7c). The trajectory of *cross* uses both areas and is at 45° to both the meandering paths, and thereby has varying distances to positions seen in the training data (Fig. 7d).

Second, there are two scale-related datasets. *Small scale* is a trajectory that is only at close proximity to the racks (Fig. 7e). The *large scale* recording is only in the large open area with larger distances to larger, more ambiguous global features (Fig. 7f).

The *volatility* dataset includes volatile features. Starting from the initial training dataset, we change the illumination and move small, medium and large objects, i.e., boxes, work benches and wall segments (Fig. 7g). This affects both the rack-area and the open area.

To allow the evaluation of the influence of motion artifacts, one test scenario was recorded mounting the recorder on a *forklift*. It includes blurry images at quick turns, higher camera speeds, and the camera angles are not steady.

Table I:
THE SPECIFICATION OF THE *WAREHOUSE* DATASET.

	name	area	images	tests for
Training	hor. meander	30 × 20	99,807	
	vert. meander	30 × 20	102,417	
Testing	generalize open area	20 × 17	74,046	General.
	generalize rack area	8.25 × 18.5	57,408	General.
	cross	24.5 × 16	17,979	General.
	large scale	19 × 19	44,590	Env.
	small scale	10 × 11	16,211	scaling
	volatility	29 × 13	29,239	Volatility
	forklift	37 × 13	23,097	Motion

V. Experiments

In our experiments we showcase the evaluation criteria under application of our *Warehouse* dataset on a state-of-the-art deep learning method.

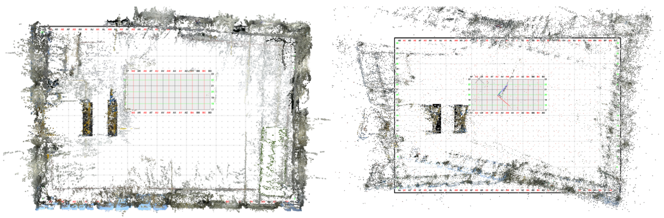
A. Experimental Setup

Fig. 6a shows a birds-eye view of a 3D-reconstruction of the warehouse, that was obtained using VisualSfM [39] with 800 images from all camera angles, exemplifying that SIFT-based methods struggle in this difficult environment. The mapping of the walls and high-level racks does not align with the floor plan. The newer SfM approach COLMAP [40] performs worse in reconstructing the layout of the warehouse, see Fig. 6b.

Hence, our experiments use positioning schemes that directly work on the images. As our baseline positioning scheme, we use the CNN-based PoseNet [8]. PoseNet is a modified version of GoogLeNet, with the classification/softmax layer replaced by a fully connected layer followed by pose regressors. The network takes an input image of 224x224 pixels to calculate the 6-DOF pose, which consists of the absolute 3D-position in meters and the orientation in degrees.

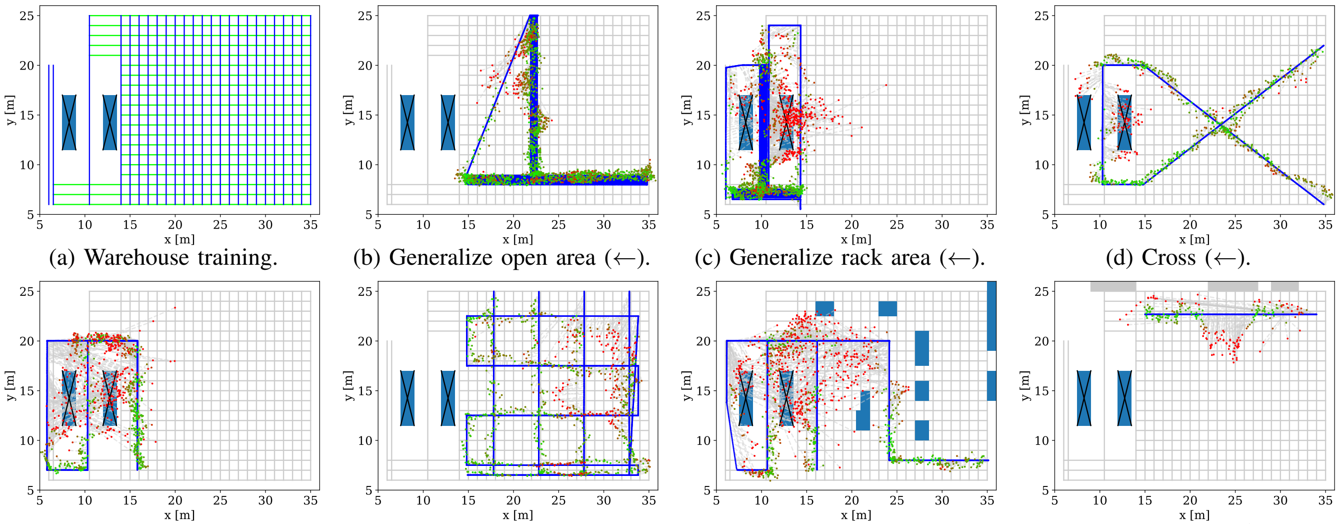
We use the pre-trained weights made available by the authors and fine-tune the network for *Warehouse*. For training, we set the loss function's β-parameter to 250 (pre-tests showed the best results for that). Furthermore, we randomly sample the training set as 95% of the combined meandering datasets, featuring all camera angles of both horizontal and vertical trajectories, and calculate the accuracy using the remaining 5% samples every 1,000 iterations and choose the intermediate network weights that provide the best accuracy.

To use the input data for the evaluation we need to normalize the images. Normalization is an important step with CNNs to reduce the correlations among the training and testing data. For the evaluation of a training dataset we calculate a mean



(a) Dense reconstruction of Warehouse with VisualSfM. (b) Sparse reconstruction of Warehouse with COLMAP SfM.

Figure 6: Structure from motion.



(a) Warehouse training. (b) Generalize open area (←). (c) Generalize rack area (←). (d) Cross (←).

(e) Small scale (←). (f) Large scale (←). (g) Volatility (→). (h) Ambiguity (↑).

Figure 7: **Warehouse dataset** training data and test results for PoseNet. The training dataset includes horizontal and vertical meandering, with two high-level racks indicated by rectangles. In test plots it is drawn in grey grid lines, testing data in blue. Predicted positions are colored from green to red (red $> 2m$ error). Blue boxes depict the high-level racks. The arrows show the direction of the camera that is used for testing.

from all images of this set and subtract it from each individual image. The same mean image is also subtracted from the test images. Since the size of the images is 640 x 480 we scale them down and center-crop them to 224 x 224 to fit the network's input layer.

Our training and test hardware setup is a Linux system with an Intel(R) Core(TM) i7-7700 CPU 3.60GHz and an NVidia GeForce GTX 1070 with 8GB VRAM. One forward pass through the network takes $16ms$ on average.

Since many industrial use-cases only require 2D positioning, e.g. forklift tracking, we focus on performance in 2D (*xy*-axis position and yaw angle only) in our experiments, while the data itself would allow for full 6-DOF evaluations. As a key performance indicator for the position errors we use the Mean Absolute Error (MAE) on to the *xy*-plane, and the Circular Error Probable (CEP) median, and its 95 percentile *CE95*. Additionally, we calculate the median error of the yaw angles.

To evaluate the properties of the ML-based positioning schemes, we conduct different tests. The results from each of the test datasets are given in Table II.

B. Generalization

To evaluate for the generalization property we consider the rack area, the open area, and their combination. First, we only test with the camera that faces the racks (Figs. 7b-d, ←).

Fig. 7b shows the results for the open area. Predictions are colored by their error, i.e., from green if the error is $< 2m$ to red if the error is $> 2m$. The positions of the recorded test trajectories are between the horizontal and the vertical training, spaced $10cm$ apart to test for unseen positions.

With a CEP of $1.06m$, an MAE of $1.72m$, and an angular error of $0.27°$, the results show that the system can locate itself relatively well between densely recorded training data. Fig. 7c shows the results for the racks-area. The horizontal trajectories, facing the wall next to and behind the racks, and the ones in the open area, are as good as in the previous test. However, errors are clustered in the vertical trajectories behind and between the racks, and are biased towards the open area. Fig. 7d shows the result of the cross trajectory test. The predictions of positions mostly perform well. Again, predictions between the racks have an error of up to about $5m$ on the x-axis, while there is a much lower error on the y-axis. There are also some discrepancies of the predictions in the end segment at the bottom, where the negative influence of natural light from the south wall windows has an effect.

The results in the open area are better than in the racks-area as the rack-area's training samples are underrepresented. These tests show the overfitting to the training samples of open area. But in summary the results show that the positioning scheme interpolates well between known trajectories. The precise clusters of predictions on the horizontal trajectories and the low CEP indicate a real-world applicability for many use-cases.

Table II:
POSITION RESULTS OF TEST SCENARIOS.

Scenario	**MAE 2D**	**CEP**	**CE95**	Rotation CEP
generalize open area	1.72m	1.06m	5.05m	0.27 °
generalize rack area	2.43m	1.76m	7.96m	0.457 °
cross	1.08m	0.86m	3.08m	0.18 °
small scale	2.31m	1.17m	8.99m	0.18 °
large scale	1.14m	0.90m	2.83m	0.18 °
volatile	2.5m	1.74m	7.35m	0.56 °
ambiguity	2.95m	1.26m	15.86m	0.23 °
forklift motion	6.76m	5.42m	16.61m	145.6 °

C. Environmental Scaling

Figs. 7e and 7f show the scale-specific tests with the camera that faces the racks (←). While most of the positions in the small-scale scenario in Fig. 7e have an error below $1.17m$ there are also outliers with large errors between the high-level racks (CE95 $8.99m$), similarly to the ones in Fig. 7c (CE95 $7.96m$). The results in Fig. 7f show that the performance decreases with a growing distance to the south and west walls. There the test samples contain many ambiguous features and difficult illumination, e.g. illuminated windows, and the positions get biased towards the center.

Similarly to generalization, the training dataset does not allow an accurate prediction in the small racks-area due to the unbalanced training data density. In the open area, while the predictions are mostly within expectations, the system suffers from the difficult environment, e.g. homogeneous areas (white, featureless walls or gates, black wall segments) and illumination. The training samples contain windows that are either illuminated by natural light or night dark, while the test samples only show illuminated windows. With many ambiguous features and difficult illumination, the performance deteriorates. However, with a CEP of $0.90m$ and a CE95 of $2.83m$ PoseNet performs very well in general which may be sufficient for many real-world use-cases.

D. Volatility

The evaluation with variations in the environment in the *volatility* scenario (Fig. 7g) uses added mobile work benches in the open area, moved large wall segments and shifted boxes in the high racks, see also Figs. 8a and 8b. The test data is sampled from the camera facing the east wall, so that both the volatile elements in the rack-area and in the open area have an impact.

Fig. 7g shows that the system tolerates small additions passably, if these do not occlude large parts of the image, such as crates. But larger, moved features pose challenges, see the predictions at $y{>}15m$. With an MAE of $2.5m$ and CE95 of $7.35m$, the negative effects are visible in Fig. 7g. Above the horizontal at $y{>}14m$, added features occlude already learned features, and moved, large wall segments at the east wall cause larger prediction errors. Besides recording training data with different lighting conditions, it is also important to have as many volatile features in a training dataset, as the environment can realistically contain. However, it is impressive that although there are significant changes in the environment, the positioning scheme's fingerprinting approach still maintains a CEP of $1.74m$.

E. Ambiguity

For testing the robustness to ambiguity we examine a relevant section of the large scale trajectory facing towards the upper wall (↑). It features two similar black wall segments (Fig. 5d) at $x = 15$ and $x = 30$. Fig. 7h shows that the system performs well in the areas of these wall segments with a small CEP of $1.26m$. However, with a fraction of the test images, the system confuses the two wall segments, due to their extremely similar features. Furthermore, between the wall segments, the system falsely predicts its distance to a large wall-filling QR code. From $21 < x < 26$, the code covers most of the images and the prediction error is the largest. This shows in the high CE95 of $15.86m$, which is heavily scewed by the clusters of erroneous outliers.

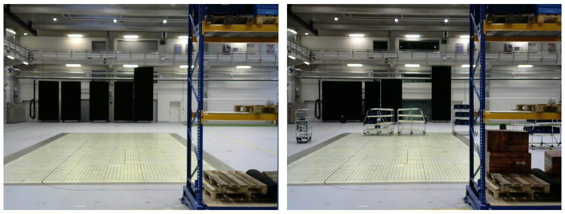

(a) Training. (b) Test.

Figure 8: Volatile features in Warehouse.

Frame filling ambiguity is problematic for optical positioning schemes. Temporal information, using for example LSTM-cells, may help cover these cases. Similarly, a Bayesian filter modeling constraint motion may be able to stabilize ML-based predictions. Alternatively, a multi-camera approach with orthogonal viewing directions may be viable.

F. Motion Artifacts

To test a typical warehouse vehicle's positioning, we mounted the cameras on a forklift's roof. We sampled the test data from the camera facing backwards, relatively to vehicular motion. The *forklift* results in Table II show that the predictions have large errors with a CEP of $5.42m$ and a CE95 of $16.61m$. The trained network is not easily transferable to the changed camera motion and height without network fine tuning with additional training samples. These results lead to the conclusion that a transfer to more dynamic use cases with higher velocity requires additional work, e.g., sensor fusion, filter algorithms or more training samples. A stand-alone ML-approach that uses single images to predict a pose (such as PoseNet) gets confused by the motion artifacts.

VI. Conclusion

With *Warehouse*, we presented the first dataset for self-positioning in a large industrial indoor scenario with high precision ground truth labels. We introduced criteria to properly evaluate ML-based positioning schemes and introduced six key algorithmic properties of such positioning schemes. We used these criteria and our dataset to evaluate a popular ML-based algorithm.

The results show strengths and weaknesses of ML-based positioning schemes. Based on this work ML-based approaches can be evaluated and developed under consistent criteria that enables better insights and comparability.

In future work we use our dataset to evaluate complementary positioning schemes, i.e., that incorporate temporal correlation of features using, e.g. Bayesian filters and LSTM cells, in the neural network architecture.

Dataset

The *Warehouse* dataset and the trained models are available for download under `https://www.iis.fraunhofer.de/warehouse`.

Acknowledgment

This work was supported by the Federal Ministry of Education and Research of Germany in the framework of Machine Learning Forum (grant number 01IS17071).

References

[1] A. Mulloni, D. Wagner, I. Barakonyi, and D. Schmalstieg, "Indoor positioning and navigation with camera phones," *IEEE Pervasive Computing*, vol. 8, no. 2, pp. 22–31, 2009.

[2] S. Lee and J.-B. Song, "Mobile robot localization using infrared light reflecting landmarks," in *Intl. Conf. Control, Automation and Systems*, (Seoul, Korea), pp. 674–677, 2007.

[3] G. Jang, S. Lee, and I. Kweon, "Color landmark based self-localization for indoor mobile robots," in *Proc. IEEE Intl. Conf. Robotics and Automation*, (Washington, D.C.), pp. 1037–1042, 2002.

[4] T. Krajník, M. Nitsche, J. Faigl, P. Vaněk, M. Saska, L. Přeučil, T. Duckett, and M. Mejail, "A practical multirobot localization system," *Intelligent & Robotic Systems*, vol. 76, no. 3, pp. 539–562, 2014.

[5] S. Saito, A. Hiyama, T. Tanikawa, and M. Hirose, "Indoor marker-based localization using coded seamless pattern for interior decoration," in *Proc. Virtual Reality Conf.*, (Charlotte, NC), pp. 67–74, 2007.

[6] J. Wang, H. Zha, and R. Cipolla, "Coarse-to-fine vision-based localization by indexing scale-invariant features," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 36, no. 2, pp. 413–422, 2006.

[7] Y. Li, N. Snavely, D. P. Huttenlocher, and P. Fua, "Worldwide pose estimation using 3d point clouds," in *Large-Scale Visual Geo-Localization*, pp. 147–163, Springer, Switzerland, 2016.

[8] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *Proc. IEEE Intl. Conf. Computer Vision*, (Santiago de Chile, Chile), pp. 2938–2946, 2015.

[9] R. Mautz and S. Tilch, "Survey of optical indoor positioning systems," in *Proc. IEEE Intl. Conf. Indoor Positioning and Indoor Navigation*, (Guimares, Portugal), pp. 1–7, 2011.

[10] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *Proc. Intl. Symp. Mixed and Augmented Reality*, (Nara, Japan), pp. 225–234, 2007.

[11] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, 2018.

[12] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *Proc. IEEE Intl. Conf. Computer Vision*, (Barcelona, Spain), pp. 2320–2327, 2011.

[13] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *Proc. Europ. Conf. Computer Vision*, (Zurich, Switzerland), pp. 834–849, 2014.

[14] M. Cummins and P. Newman, "Fab-map: Probabilistic localization and mapping in the space of appearance," *The Intl. J. Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Intl. J. Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[16] L. Meng, J. Chen, F. Tung, J. J. Little, J. Valentin, and C. W. de Silva, "Backtracking regression forests for accurate camera relocalization," in *Proc. IEEE/RSJ Intl. Conf. Intelligent Robots and Systems*, (Vancouver, Canada), pp. 6886–6893, 2017.

[17] J. Valentin, M. Nießner, J. Shotton, A. Fitzgibbon, S. Izadi, and P. H. Torr, "Exploiting uncertainty in regression forests for accurate camera relocalization," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, (Boston, MA), pp. 4400–4408, 2015.

[18] 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. IEEE Conf. Computer Vision and Pattern Recognition*, (Portland, OR), pp. 2930–2937, 2013.

[19] A. Kendall and R. Cipolla, "Modelling uncertainty in deep learning for camera relocalization," in *IEEE Intl. Conf. Robotics and Automation*, (Stockholm, Sweden), pp. 4762–4769, 2016.

[20] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *IEEE Conf. Computer Vision and Pattern Recognition*, (Honolulu, HI), pp. 6555–6564, 2017.

[21] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu, "Image-based localization using hourglass networks," in *Proc. IEEE Intl. Conf. Computer Vision Workshop*, (Santiago de Chile, Chile), pp. 870–877, 2017.

[22] A. Valada, N. Radwan, and W. Burgard, "Deep auxiliary learning for visual localization and odometry," in *IEEE Intl. Conf. Robotics and Automation*, (Brisbane, Australia), 2018.

[23] 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. IEEE Conf. Computer Vision and Pattern Recognition*, (Honolulu, HI), pp. 2652–2660, 2017.

[24] 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. IEEE Conf. Computer Vision and Pattern Recognition*, (Honolulu, HI), pp. 627–637, 2017.

[25] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, *et al.*, "Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera," in *Proc. 24th An. ACM Symp. User interface software and technology*, (Santa Barbara, CA), pp. 559–568, 2011.

[26] Z. Laskar, I. Melekhov, S. Kalia, and J. Kannala, "Camera relocalization by computing pairwise relative poses using convolutional neural network," in *Proc. IEEE Intl. Conf. Computer Vision Workshop*, (Santiago de Chile, Chile), pp. 920–929, 2017.

[27] R. Roberto, J. P. Lima, T. Arajo, and V. Teichrieb, "Evaluation of motion tracking and depth sensing accuracy of the tango tablet," in *Proc. Intl. Symp. Mixed and Augm. Reality*, (Merida, Mexico), pp. 231–234, 2016.

[28] X. Sun, P. Xie, Luo, and L. Wang, "A dataset for benchmarking image-based localization," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, (Honolulu, HI), pp. 5641–5649, 2017.

[29] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, S. Savva, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," in *Proc. Intl. Conf. 3D Vision*, (QingDao, China), 2017.

[30] E. Wijmans and Y. Furukawa, "Exploiting 2d floorplan for building-scale panorama rgbd alignment," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, (Honolulu, HI), pp. 308–316, 2017.

[31] 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 *IEEE Conf. Computer Vision and Pattern Recognition*, (Salt Lake City, UT), 2018.

[32] J. Valentin, A. Dai, M. Niessner, P. Kohli, P. Torr, S. Izadi, and C. Keskin, "Learning to navigate the energy landscape," in *Proc. Intl. Conf. 3D Vision*, (Stanford, CA), pp. 323–332, 2016.

[33] X. Wang, L. Gao, S. Mao, and S. Pandey, "Csi-based fingerprinting for indoor localization: A deep learning approach," *IEEE Vehicular Technology*, vol. 66, no. 1, 2017.

[34] J. Torres-Sospedra, R. Montoliu, A. Martinez-Uso, J. P. Avariento, T. J. Arnau, M. Benedito-Bordonau, and J. Huerta, "Ujiindoorloc: A new multi-building and multi-floor database for wlan fingerprint-based indoor localization problems," in *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation*, (Busan, Korea), pp. 261–270, 2014.

[35] W. Zhang, K. Liu, W. Zhang, Y. Zhang, and J. Gu, "Deep neural networks for wireless localization in indoor and outdoor environments," *Neurocomputing*, vol. 194, pp. 279 – 287, 2016.

[36] Z. Wu, J. K. Ng, and K. R. Leung, "Location estimation via support vector regression," *Trans. Mob. Comp.*, vol. 6, no. 3, pp. 311–321, 2007.

[37] A. Niitsoo, T. Edelhäußer, and C. Mutschler, "Convolutional neural networks for position estimation in tdoa-based locating systems," in *Proc. 9th Intl. Conf. Indoor Positioning and Indoor Navigation*, (Nantes, France), 2018.

[38] N. Piasco, D. Sidibé, C. Demonceaux, and V. Gouet-Brunet, "A survey on visual-based localization: On the benefit of heterogeneous data," *Pattern Recognition*, vol. 74, no. C, pp. 90–109, 2018.

[39] C. Wu, "Towards linear-time incremental structure from motion," in *Proc. Intl. Conf. 3D Vision*, (Seattle, WA), pp. 127–134, 2013.

[40] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, (Las Vegas, NV), pp. 4104–4113, 2016.