Learning to Detect Scene Landmarks for Camera Localization

Tien Do[1] Ondrej Miksik[2] Joseph DeGol[2] Hyun Soo Park[1] Sudipta N. Sinha[2]

[1] University of Minnesota [2] Microsoft

Figure 1. We present a new method to recognize scene-specific *scene landmarks* to localize a camera, which preserves privacy and achieves high accuracy. [Left] Scene landmark detections in a query image obtained from a heatmap-based CNN architecture. [Middle] A visualization of the predicted heatmap scores. [Right] The 3D scene landmarks (in red) and the estimated camera pose (in blue) are shown over the 3D point cloud (in gray). The 3D point cloud is shown only for the purpose of visualization.

Abstract

Modern camera localization methods that use image retrieval, feature matching, and 3D structure-based pose estimation require long-term storage of numerous scene images or a vast amount of image features. This can make them unsuitable for resource constrained VR/AR devices and also raises serious privacy concerns. We present a new learned camera localization technique that eliminates the need to store features or a detailed 3D point cloud. Our key idea is to implicitly encode the appearance of a sparse yet salient set of 3D scene points into a convolutional neural network (CNN) that can detect these scene points in query images whenever they are visible. We refer to these points as scene landmarks. We also show that a CNN can be trained to regress bearing vectors for such landmarks even when they are not within the camera's field-of-view. We demonstrate that the predicted landmarks yield accurate pose estimates and that our method outperforms DSAC, the state-of-the-art in learned localization. Furthermore, extending HLoc (an accurate method) by combining its correspondences with our predictions boosts its accuracy even further.*

1. Introduction

Camera localization is the task of estimating the 3D position and 3D orientation of a camera from a query image with respect to a pre-built scene map. This task is a fundamental building block to enable VR/AR systems that allow users to persistently interact with the surrounding 3D scene. These scenes are often private spaces; e.g., homes, where existing localization methods that use retrieval and feature matching [1,4,16,38,57,62] are not suitable because stored images or features can be inverted to reveal sensitive scene content (raising serious privacy concerns [54,72,73]). Furthermore, existing localization methods usually require long term storage of many images or a vast amount of features and 3D points. In lifelong localization settings, new images and features will be continuously added, causing the database to grow over time. The ensuing memory footprint may also exceed the limits of on-device localization for VR/AR systems. Map pruning can help [12,39,71], but it's efficacy for lifelong localization is unproven.

Learned localization approaches such as absolute pose regression [29,30,84] and scene coordinate regression [7,8,10,69] address both aforementioned issues. These methods implicitly encode scene information in the learned parameters of a convolutional neural network (CNN), rather than explicitly storing images or features. Thus, they preserve privacy by design. However, their performance is not yet on par with the top performing methods that use retrieval, feature matching, and structure-based pose estimation [65,67].

In this paper, we present a new learned method for camera localization that (1) preserves privacy, (2) requires low storage, and (3) outperforms the state-of-the-art storage-free pose regression methods. Our idea is inspired by the

recent success of landmark detection in human pose estimation [46,86] and keypoint recognition for objects [34,51,77] and faces [18]. Instead of human body joints or object keypoints, we recognize salient, scene-specific 3D points called *scene landmarks* from a query image as shown in Figure 1. This landmark recognition approach is privacy preserving and requires low data storage as no visual features need to be retained. The landmark recognition establishes 2D-3D correspondences that can be used to robustly estimate the camera pose. We implement the proposed idea by training a scene-specific CNN architecture that detects the landmarks, i.e., regresses the 2D coordinates of the landmarks in the input image. We show that running structure from motion (SfM) on the mapping images is sufficient to find a set of salient landmarks and automatically produce the data needed to train the architecture.

Unlike human pose estimation where most landmarks are typically visible (up to occlusion), most of the scene landmarks are not expected to be simultaneously visible, due to limited camera field-of-view and because landmarks in different parts of the scene cannot be observed simultaneously. We address this challenge by proposing a new Neural Bearing Estimator (NBE) that can directly regress the 3D bearing vectors for the scene landmarks in the camera coordinate frame. NBE learns a global scene representation, similar to PoseNet [30], while learning to predict the direction vectors of scene landmarks even when they are invisible. We show that NBE is highly effective and outperforms PoseNet by a significant margin. Our full approach combines scene landmark detection and the NBE method.

Although our method learns to predict 2D-3D correspondences similar to existing scene coordinate regression (SCR) approaches [10, 69], there are several crucial differences. First, SCR methods predict *dense* 3D world coordinates for scene points observed at every pixel. In contrast, we assume the 3D coordinates for a few salient scene landmarks are given, and we only infer their 2D positions in the image. Thus, our matches are extremely sparse (between 10–40) compared to SCR approaches that use thousands.

Our method has comparable performance to HLoc [59, 60] and DSAC* [10] on 7-scenes [69] but outperforms DSAC* on our new INDOOR-6 dataset that contains changing scenes, day/night images, and strong illumination variations. Although we have motivated our method based on the need to avoid storage, we show that it is also useful when storage is not a concern. The 2D–3D correspondences recovered from our method appear to complement those recovered by other methods; *e.g.*, by combining our method with HLoc, we boost its accuracy even further.

Contributions. This paper presents the following technical contributions: **(1)** a new formulation for heatmap-based landmark localization and bearing angle estimation that is privacy preserving and can be used to localize a camera in a pre-built map; **(2)** a new dataset that can be used to effectively evaluate camera localization performance in challenging scenarios; and **(3)** superior results with low storage compared to existing storage-free localization methods.

2. Related Work

We briefly review the literature for camera localization, a topic that has been studied for decades. We shed some new light on the topic in the context of modern AR/VR applications while focusing on privacy and pose accuracy.

Classical Approaches. Many classical approaches detect local image features [4, 38, 43, 57], match them against a 3D point cloud of the scene, and use robust absolute pose estimation algorithms [14, 21, 37, 56] to obtain the camera pose. Further, image retrieval-based methods use scalable techniques [25,27,45,79] to estimate the query camera pose by interpolating poses of the retrieved database images [11, 53, 79, 80]. Other alternatives include large-scale location classification and regression [5, 22, 89]. Many classical approaches have been extended by replacing handcrafted features with learned features [1, 16, 19, 24, 44, 59–61, 78, 94].

Retrieval or classification-based approaches only provide an approximate pose estimate. On the other hand, approaches that use stored maps (database of images, features, and 3D points) are usually more accurate. However, they can compromise user privacy [54]. Furthermore, maintaining up-to-date maps of changing scenes can be challenging and memory expensive. Unlike map-based methods that may store several images and hundreds of features per image, our approach needs to store only 200–300 3D points for the whole scene and does not require any matching.

Absolute and Relative Pose Regression. Learned absolute pose regression (APR) approaches such as PoseNet [29,30] regress the camera pose from an input image and do not maintain a 3D scene map. New variants of APR are scene-agnostic, are faster during training and inference [68,92], or use attention for improved accuracy [84]. It has been shown that APR models are closely related to pose approximation via image retrieval rather than 3D structure-based pose estimation [65], and require uniformly sampled training images for high accuracy [47]. Therefore, APR methods are unable to surpass the accuracy of the current state-of-the-art in camera localization, such as DSAC* [10].

Relative pose regression (RPR) approaches regress the relative pose of the query image with respect to one or more database images. While these often generalize better than APR methods, they require long-term storage of images or features [3, 32]. While storage can be avoided by using implicit scene anchors and predicting relative pose to anchors, it typically leads to lower pose accuracy [58].

Unlike APR and RPR methods, our proposed neural bearing estimator (NBE) predicts landmark bearings as an

intermediate step towards recovering absolute pose. That lets us leverage robust, geometric pose solvers. NBE is simpler to train as its training loss is a pure angular quantity obtained from bearing predictions. No hyperparameters are needed to balance rotational and translational losses compared to APR methods. NBE clearly outperforms PoseNet on our new dataset. However, it does not surpass top performing methods such as HLoc [59, 60] and DSAC* [10].

3D Scene Coordinates Regression (SCR). SCR methods regress dense 3D scene coordinates to establish 2D-3D correspondences that are fed into pose solvers [35, 36, 41, 42, 69, 82], allowing them to exploit geometric reasoning. They have been extended to (1) improve scalability by using ensembles [9], (2) make them scene agnostic [93], and (3) allow continual updates [85]. DSAC [7] was the first example of an end-to-end trainable SCR architecture. Subsequently, DSAC++ [8] was proposed to alleviate the need for RGBD ground truth. In DSAC* [10], further improvements were proposed that yield faster training and higher accuracy. In contrast to SCR approaches that densely predict 3D points for every pixel in the query image, we store a few salient 3D points with known coordinates and produce highly sparse 2D detections (or regress 3D bearings) of those 3D points in the query image. SCR approaches generate several thousands of redundant 2D–3D correspondences, whereas our predictions are highly sparse but also accurate.

Keypoint and Landmark Detection. Beyond the camera localization literature, there is a long tradition of formulating keypoint or landmark recognition as a classification problem. These were initially proposed to find the 6-DoF pose of small objects using random forests [34], random ferns [49], and nowadays, CNNs [48, 51, 52, 55]. Such techniques also work for deformable object categories (humans [46, 81, 86], faces [18, 91, 96], and hands [70]) to find semantic keypoints such as joint positions. Modern keypoint detectors [13, 74, 90] leverage appearance and spatial contextual cues, and CNNs (by virtue of their large receptive field) can effectively reason jointly about them [46, 86]. Surprisingly, we are not aware of such approaches being used for 6-DoF camera localization. While some methods were studied for improved place recognition using scene keypoints [5] or leveraging 2D scene object detection for camera pose estimation [87], they do not address the camera localization task in its full generality.

Datasets. There are many camera localization datasets [2, 15, 20, 26, 33, 40, 63, 64, 67, 75, 88]. However, they typically capture sparse imagery and are focused on large-scale, outdoor scenes, the self-driving setting, or provide limited training data. Indoor datasets, such as 7-Scenes [69] and 12-scenes [82], are popular for learned methods and relevant for us. However, these datasets are for small-scale, stationary scenes captured under fixed illumination. Changing environments are still a major hurdle for localization methods, as shown on the RIO10 dataset [83]. Unfortunately, for RIO10, limited training data was publicly released (only two sequences per scene), making it difficult to develop learned methods. Therefore, to evaluate our method and compare it to existing baselines, we collect our own multi-session, INDOOR-6 dataset in non-stationary environments where images captured at different times of day and night also exhibit strong illumination variations.

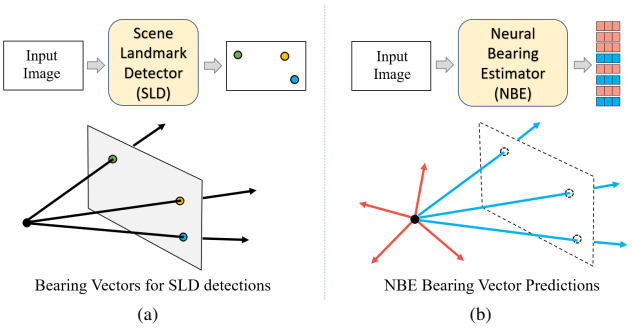

Figure 2. We present two methods for scene landmark prediction. (a) Our Scene Landmark Detection (SLD) formulation involves training a network to detect 2D landmarks in the image from which bearing vectors can be easily computed. (b) In our second formulation, Neural Bearing Estimation (NBE), a network directly regresses 3D bearings in camera coordinates (blue vectors are inside the camera's FoV whereas red vectors are outside).

3. Proposed Methodology

In this section, we present two formulations for predicting scene landmarks from which we recover the camera pose. In our first approach, we train a model to identify 2D scene landmarks in the image, which we call our Scene Landmark Detector (SLD). Since we assume known camera intrinsics, these 2D detections can be converted into 3D bearing vectors or rays. In the second approach, we train a different model to directly predict the 3D bearing vectors in camera coordinates for the landmarks, which we call our Neural Bearing Estimator (NBE). These two proposed ideas are outlined in Figure 2. With SLD, only landmarks visible in the camera's field of view (FoV) can be detected, whereas NBE predicts bearings for all the landmarks, including invisible ones outside the camera's field of view.

Notation and Preliminaries. To train SLD and NBE, we use a 3D point cloud $\mathcal{P}$ reconstructed by structure from motion (SfM) using a set of RGB images, $\mathcal{I} = \{I_i\}_{i=1}^{N}$, captured by a pinhole camera, where N is the number of images. Each image I is associated with an operation of camera projection $\pi_{\mathbf{K},\mathbf{R},\mathbf{t}}(\mathbf{x})$ that maps the 3D point $\mathbf{x}$ to the image coordinate where $\mathbf{R} \in SO(3)$, $\mathbf{t} \in \mathbb{R}^3$, and $\mathbf{K} \in \mathbb{R}^{3\times3}$ are the rotation, translation, and intrinsic parameters of the

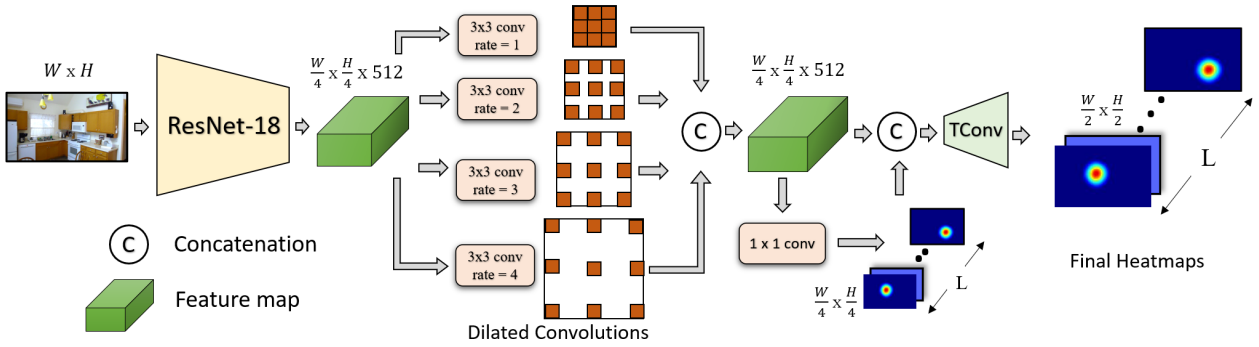

Figure 3. Fully convolutional architecture for the Scene Landmark Detector (SLD) that outputs a stack of heatmaps, one per landmark.

image, respectively. A subset of the 3D point cloud, $\mathcal{S}$, is selected to form *scene landmarks* to learn SLD and NBE, $\mathbf{s}_\ell \in \mathcal{S} \subset \mathcal{P}$, where $\mathbf{s}_\ell \in \mathbb{R}^3$ is the ℓ-th landmark. We use the visibility computed by SfM for each image I and denote it as $\mathcal{S}_{\text{vis}}(I)$. A landmark $\mathbf{s}_\ell$ is projected onto the camera to form the image projection $\mathbf{u}_\ell = \pi_{\mathbf{K},\mathbf{R},\mathbf{t}}(\mathbf{s}_\ell)$. For a 2D image point $\mathbf{u}$, we denote the associated unitized bearing vector in the camera coordinate as $\mathbf{b} = \frac{\mathbf{K}^{-1}\mathbf{u}}{\|\mathbf{K}^{-1}\mathbf{u}\|_2} \in \mathbb{S}^2$.

3.1. Scene Landmark Detector (SLD)

We implement the scene landmark detector (SLD) using a CNN-based architecture. Inspired by prior work [18, 46], SLD is designed to take an RGB image I as input and output a set of pixel likelihood maps (heatmaps) $\{H_\ell(I,\Phi) \in [0,1]^{W' \times H'}\}_{l=1}^{L}$ that indicate the location of each visible landmark $\ell = 1, \ldots, L$ respectively, where W' and H' are the width and height of the heatmaps. Φ denotes the learnable CNN parameters, which are specific to each scene.

The neural network architecture consists of four main components. First, a ResNet-18 [23] backbone is used with the last three max-pool layers removed to retain high resolution feature maps (the output resolution is a quarter of the input image resolution). Second, a dilated convolution block [95] is used after the ResNet-18 backbone with dilation rates set to 1, 2, 3, and 4. Next, a transposed convolution layer performs upsampling and is responsible for generating heatmaps that are half the resolution of the input image. The final layer consists of 1×1 convolutions which predict L heatmap channels, one for each landmark $\mathbf{s}_\ell \in \mathcal{S}$. The architecture is illustrated in Figure 3.

To train the architecture, we use the ground truth heatmap $\widehat{H}_\ell(I)$ and employ the mean squared loss:

$$\mathcal{L}_{\text{SLD}}(\Phi) = \sum_{I} \sum_{\ell=1}^{L} ||H_\ell(I,\Phi) - \widehat{H}_\ell(I)||_F^2 \qquad (1)$$

where $||.||_F$ denotes the Frobenius norm, $\widehat{H}_\ell$ is obtained by convolving a Dirac delta function at the projected landmark location $\mathbf{u}_\ell$ with a 2D Gaussian filter with a standard deviation of σ, and $\mathbf{u}_\ell$ is the 2D position where landmark $\mathbf{s}_\ell \in \mathcal{S}_{\text{vis}}(I)$ is observed in image I. If the landmark ℓ is not observed in the image I, we set $\widehat{H}_\ell(I)$ as a $\mathbf{0}$ matrix with proper dimension. We assign σ = 5 (pixels) and generate two sets of ground truth heatmaps at quarter $(W' = \frac{W}{4}, H' = \frac{H}{4})$ and half $(W' = \frac{W}{2}, H' = \frac{H}{2})$ resolutions where $W \times H$ is the input image dimension.

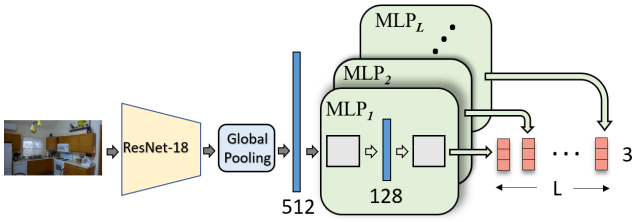

Figure 4. Our NBE architecture directly regresses 3D bearing vectors for all the landmarks. A ResNet-18 backbone feeds into L multi-layered perceptron (MLP) heads that predict the bearings.

During inference, we assume a landmark has been detected when the maximum heatmap value in its channel exceeds a threshold $\tau = 0.2$. To get the 2D location $\widehat{\mathbf{u}}_\ell$ with *subpixel accuracy*, we compute the expectation over the cropped 17×17 patch at the heatmap peak location:

$$\widehat{\mathbf{u}}_\ell = \mathbb{E}_{H_\ell(I,\Phi)}[\mathbf{u}]. \qquad (2)$$

3.2. Neural Bearing Estimator (NBE)

We design a simple model to regress bearing vectors for the full set of scene landmarks (*even if it is invisible*) given an image I. Specifically, our CNN (ResNet-18 [23] backbone) takes as input an image I to produce a deep feature map. It is then followed by multiple MLP (multi-layer perceptron) blocks. Each block outputs a bearing vector towards a landmark. Our MLP blocks contain two fully connected layers with 128 ReLU activation nodes.

We denote the network parameters as Θ and each landmark bearing vector prediction as $\mathrm{B}_\ell(I,\Theta) \in \mathbb{R}^3$. We train

our neural network using the ground truth bearing $\mathbf{b}_\ell(I)$ in camera coordinates with the robust angular loss $\mathcal{L}_{\text{ang}}$ [17]:

$$\mathcal{L}_{\text{NBE}}(\Theta) = \sum_{I} \sum_{\ell=1}^{L} \mathcal{L}_{\text{ang}}\Big(\frac{\mathrm{B}_\ell(I,\Theta)}{||\mathrm{B}_\ell(I,\Theta)||_2}, \mathbf{b}_\ell(I)\Big). \quad (3)$$

3.3. Camera Pose Estimation

We first feed each query image into the SLD network to obtain 2D detections, which we convert to a set of landmark bearings, $\mathbf{B}_1$. If more than eight scene landmarks are detected, we compute the camera pose using a robust minimal solver (P3P [28] + RANSAC [21]) followed by a Levenberg-Marquardt based nonlinear refinement. Otherwise, we feed the same image into the NBE network and obtain predicted bearings $\mathbf{B}_2$ after which we merge the sets of bearing estimates $\mathbf{B}_1$ and $\mathbf{B}_2$ to form a new set $\mathbf{B}_3$. When a bearing in both sets $\mathbf{B}_1$ and $\mathbf{B}_2$ refer to the same landmark, we keep the estimate from $\mathbf{B}_1$ since SLD is usually more accurate than NBE. Finally, we compute camera pose using the same procedure described above but with $\mathbf{B}_3$.

Unlike DSAC* [10] which estimates 3D scene coordinates from an input image I, our SLD approach focuses on recognizing salient parts of the scene. Compared to APR [29, 30] approaches that directly predict camera pose by memorization, NBE predicts 3D direction vectors that are directly associated with pixels; i.e., bearings are implicitly associated with pixel locations. We conjecture that this geometric association in NBE and the ability to utilize geometric constraints imposed by absolute pose solvers improve the overall generalization over pure APR methods. This is confirmed by empirical evidence presented later (Section 4.1). Finally, combining NBE predictions and SLD detections improves overall recall, especially when a small number of landmarks are visible in the image.

3.4. Scene Landmark Selection

Finding the optimal subset of the L scene landmarks from the SfM point cloud $\mathcal{P}$ is a combinatorial problem where evaluating every subset is intractable. Instead, inspired by prior work [5] that also proposed finding discriminative keypoints or scene elements in a greedy fashion, we select the scene landmarks that are (a) robust (longer track), (b) repeatable (seen in multiple episodes[1]), and (c) generalizable (observed from many different viewing directions and depths). We measure a saliency score $A(\mathbf{x})$ for a 3D point $\mathbf{x}$ with a track longer than a threshold t, as follows:

$$A(\mathbf{x}) = \lambda \log_2(l) + \frac{e}{E} + \min(a, 2) + \min(d, 1) \quad (4)$$

where l is the observation track length, e is the number of unique episodes (videos) the point was observed in, and E is the number of episodes in the training set. a denotes the largest angle in radians formed by any two rays among all visible views where a ray is the line between the 3D point and the position of a camera. $d = \sigma_d/\mu_d$ indicates a normalized depth variation where μ_d and σ_d are the mean and the standard deviation of the depths for track observations respectively. We set $t = 25$ and $\lambda = 0.25$. We compute the set of saliency scores for all points $\mathbf{x} \in \mathcal{P}$ as $\mathcal{A} = \{A(\mathbf{x})\}_{\mathbf{x}\in\mathcal{P}}$.

[1] We refer to each unique video in the training set as an episode.

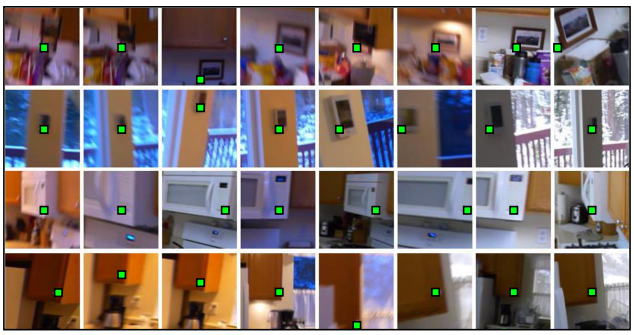

Figure 5. Each row shows image patches from training images for four selected landmarks (green dots). Notice the appearance variation caused by changes in viewpoint, scale, lighting, and blur.

In addition to maximizing the overall saliency score, we seek scene landmarks that spatially cover the 3D scene such that some landmarks are visible from anywhere within the scene; i.e., no matter where the camera is in the scene, we want some landmarks to be visible. To that end, we use the constrained greedy approach described in Algorithm 1 which iteratively invokes the routine $\text{SelectBestPoint}(\mathcal{P}, \mathcal{A}, \mathcal{S}, r)$:

$$\underset{\mathbf{x}\in\Omega}{\text{argmax}}\ A(\mathbf{x}), \qquad \Omega = \{\mathbf{x} |\ \mathbf{x} \in \mathcal{P}, \|\mathbf{x} - \mathbf{s}\| > r\ \ \forall \mathbf{s} \in \mathcal{S}\},$$

which finds the point with the highest saliency score whose distance to any scene landmarks is greater than r. This point selection approach achieves high saliency scores while ensuring full scene coverage. Figure 5 shows cropped images for a few selected landmarks, and the 3D landmark positions for one scene is shown in Figure 1.

Algorithm 1: Landmark Selection

Input : $\mathcal{P}$, $\mathcal{A}$, L, r_0
Output: $\mathcal{S}$
$\mathcal{S} \leftarrow \{\}$ and $\mathrm{r} \leftarrow \mathrm{r}_0$; // Initialize with large coverage radius
do
  $\mathbf{x} \leftarrow \text{SelectBestPoint}(\mathcal{P}, \mathcal{A}, \mathcal{S}, r)$;
  if $\mathbf{x} \neq \emptyset$ **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup \mathbf{x}$;
  else
    $r \leftarrow \frac{r}{2}$; // Halve coverage radius to find more points
  end
while $|\mathcal{S}| < L$;

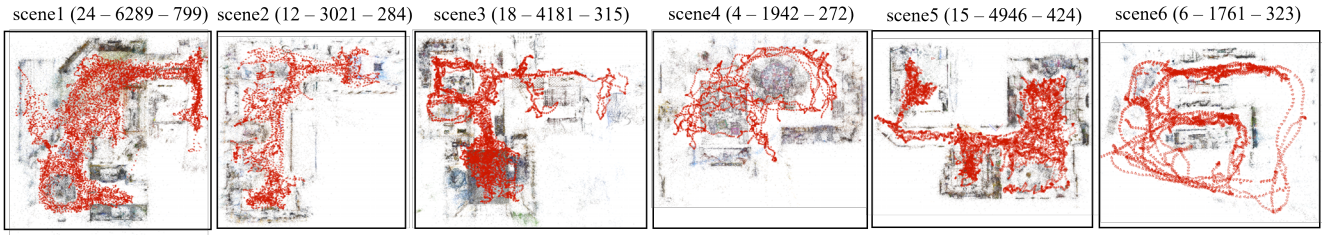

Figure 6. **INDOOR-6 Dataset.** Top-view SfM reconstructions of the multi-room indoor scenes in our INDOOR-6 dataset with the point cloud (in gray) and camera locations (in red). The number of (episodes – training images – test images) is shown.

3.5. Implementation Details

Training SLD. The high variation in saliency and visibility of the selected landmarks can lead to unbalanced training data causing the network to focus on landmarks with more training samples. To address that, we adopt a batch-balanced training strategy which is designed to treat all landmarks in the training data equally. At each iteration, we construct a mini-batch by randomly selecting 128 cropped image patches of size 96×96. The mini-batch contains 64 cropped patches where each patch contains at least one visible landmark in $\mathcal{S}$ and 64 random crops from a random set of training images. For data augmentation, we apply transformations to each training patch: (1) random cropping and scaling (0.75–1.25×); (2) homography warping generated with 0-20 degree roll, pitch, and yaw camera rotations; and (3) intensity gain change up to $\pm 10\%$. Finally, as the point visibilities estimated from COLMAP can suffer from false negatives, we build an extended visibility set $\mathcal{S}_{\text{vis}}(I)$ for each image I by adding landmarks that are visible in nearby images (with poses up to 10cm/10° from that of I).

Training NBE. For each image I, we split the set of landmarks $\mathcal{S}$ into the visible set $\mathcal{S}_{\text{vis}}(I)$ and the invisible set $\mathcal{S} \setminus \mathcal{S}_{\text{vis}}(I)$. We then weigh the angular loss for bearings in $\mathcal{S}_{\text{vis}}$ ten times higher than bearings in the invisible set.

We implement both SLD and NBE architectures in PyTorch [50] and trained them with batch sizes of 128 patches and 32 images respectively on a single NVIDIA Tesla V100 GPU using the Adam [31] optimizer. We train SLD for 200 epochs. The learning rate is initially set to 10^{-3} and halved every 20 epochs. For NBE, we use 100 epochs with an initial learning rate of 10^{-3} and halve it every 10 epochs. The code and data can be found at https://github.com/microsoft/SceneLandmarkLocalization.

4. Experimental Results

Datasets. We use the public 7-SCENES [69] dataset and our new INDOOR-6 dataset for evaluation. The 7-SCENES dataset contains multiple Kinect RGBD videos captured in seven indoor scenes. The videos are split into train/test splits for chess (4/2), fire (2/2), heads (1/1), office (6/4), pumpkin (4/2), redkitchen (7/5), and stairs (4/2), where (train/test) denotes the number of videos for train and test respectively, and each video consists of 500-1000 images at 640×480 resolution. As shown in Table 4, the state-of-the-art methods achieve nearly 100% recall on 7-SCENES because of small scale scene sizes and fixed illumination.

Our INDOOR-6 dataset was created from multiple RGB 30 fps videos captured in six indoor scenes over multiple days. We extracted frames at 3 fps and split them into training and test sets.[2] The pseudo ground truth (pGT) 3D point clouds and camera poses for each scene are computed using COLMAP [66]. Figure 6 shows the camera poses (in red) and point clouds (in gray) and for each scene, the number of video and images in the training and test split respectively. Compared to 7-SCENES, the scenes in INDOOR-6 are larger, have multiple rooms, and contain illumination variations as the images span multiple days and different times of day.

Evaluation Metrics. We evaluate the estimated pose using the standard metrics [69] as rotational and positional errors:

$$\Delta R = \arccos \frac{\text{Tr}(\mathbf{R}^\top \hat{\mathbf{R}}) - 1}{2}, \quad \Delta T = \|\mathbf{R}^\top \mathbf{t} - \hat{\mathbf{R}}^\top \hat{\mathbf{t}}\|_2.$$

where $(\mathbf{R}, \mathbf{t})$ and $(\hat{\mathbf{R}}, \hat{\mathbf{t}})$ are the estimated and ground truth camera poses respectively. We report the median of ΔR and ΔT per scene and the percentage of test images where $\Delta R \leq 5°$ and $\Delta T \leq 5$cm respectively.

Quantitative Comparison. We have evaluated five variants of our method – **NBE**, **SLD**, **NBE+SLD**, **NBE+SLD(E)**, and **HLoc+SLD**. While NBE+SLD and HLoc+SLD use ResNet-18 backbones, the variant NBE+SLD(E) uses a 4× more compact EfficientNet-Lite0 [76] backbone. In HLoc+SLD, we combine the 2D–3D correspondences recovered by HLoc with the 2D–3D landmark correspondences from SLD and then use them all to estimate pose.

We also evaluated four baselines. **(1) PoseNet.** We reimplemented PoseNet [30] using the ResNet-18 backbone to ensure a fair comparison to NBE. Both architectures use the same backbone and are almost identical except that PoseNet uses a different MLP layer to regress the full camera pose. **(2) DSAC*** [10]**.** We used the DSAC*+3D model (SfM) configuration and trained a model for each scene.

[2] Frames from each video are only added to one of the two splits.

Method	INDOOR-6																	
	scene1			scene2			scene3			scene4			scene5			scene6		
	(cm.)↓	(deg.)↓	(%)↑	(cm.)↓	(deg.)↓	(%)↑	(cm.)↓	(deg.)↓	(%)↑	(cm.)↓	(deg.)↓	(%)↑	(cm.)↓	(deg.)↓	(%)↑	(cm.)↓	(deg.)↓	(%)↑
PoseNet	159.0	7.46	0.0	193.0	8.42	0.0	141.0	9.26	0.0	109.4	7.84	0.0	179.3	9.37	0.0	118.2	9.26	0.0
NBE	22.3	4.03	2.0	29.9	4.88	2.1	24.7	4.85	2.9	39.9	5.35	1.5	37.8	5.28	0.0	30.8	6.60	0.3
DSAC*	12.3	2.06	18.7	17.5	3.4	12.3	13.1	2.34	19.7	5.5	0.84	44.9	40.7	6.72	10.6	6.0	1.40	44.3
NBE+SLD(E)	7.5	1.15	28.4	11.8	2.30	26.1	6.2	1.28	43.5	5.1	0.75	48.9	6.3	0.96	37.5	5.8	1.30	44.6
NBE+SLD	**6.5**	**0.9**	**38.4**	**7.4**	**1.6**	**37.0**	**4.4**	**0.91**	**53.0**	**4.0**	**0.63**	**62.5**	**6.0**	**0.91**	**40.0**	**5.0**	**0.99**	**50.5**
HLoc-L$_{300}$	-	-	12.9	-	-	7.0	-	-	27.3	-	-	44.5	-	-	9.7	-	-	28.4
HLoc-L$_{1000}$	8.7	1.20	33.3	-	-	25.4	5.5	1.02	48.3	4.3	0.64	56.6	-	-	21.9	5.6	1.10	47.4
HLoc-L$_{3000}$	5.3	0.73	48.1	-	-	31.3	3.4	0.65	61.9	3.6	0.54	69.5	-	-	31.1	3.7	0.71	59.1
HLoc	3.2	0.47	64.8	3.9	0.76	60.6	2.1	0.37	**81.0**	3.3	0.47	70.6	6.1	0.86	42.7	**2.1**	**0.42**	79.9
HLoc+SLD	**2.9**	**0.43**	**68.7**	**3.4**	**0.63**	**62.7**	**1.9**	**0.32**	**81.0**	**2.8**	**0.45**	**73.9**	**5.4**	**0.78**	**45.3**	**2.1**	**0.42**	**82.0**

Table 1. **INDOOR-6 quantitative evaluation.** We report the median position error (cm), median rotation error (°), and recall at (5cm, 5°). Low recall methods have invalid medians (marked "-"). The top five rows are for storage-free methods, the best amongst them is marked **green**. The bottom five are for methods with high storage. The best overall method is marked **blue**. HLoc+SLD performs the best overall.

Method	Storage (GB)
DSAC*	0.027
NBE+SLD(E)	0.029
NBE+SLD	0.132
HLoc-L$_{300}$	0.14–0.19
HLoc-L$_{1000}$	0.17–0.21
HLoc-L$_{3000}$	0.21–0.48
HLoc	0.73–2.36

Method					INDOOR-6 (recall (5cm, 5°))						
					scene1	scene2	scene3	scene4	scene5	scene6	
	Patches	Res.	Aug.	L	# of visible points ≥ 8 ↑						Average ↑
DSAC*	-	-	-	-	-	-	-	-	-	-	25.1
SLD	×	1/4	×	200	24.9	20.4	42.2	77.6	40.1	39.6	18.2
SLD	✓	1/4	×	200	77.2	38.0	53.0	94.1	72.2	66.3	28.2
SLD	✓	1/2	×	200	61.1	38.4	44.4	91.5	58.3	59.4	36.9
SLD	✓	1/2	✓	200	66.0	34.9	52.4	90.4	62.7	57.6	38.4
SLD	✓	**1/2**	✓	**300**	74.6	48.0	68.6	94.9	88.9	66.3	**42.7**
SLD	✓	1/2	✓	400	73.8	45.1	80.3	96.3	93.2	74.3	42.4

Table 2. [Left] **Storage.** Storage used by various methods on the INDOOR-6 scenes. The range shows that HLoc variants use notable storage for larger scenes. [Right] **SLD training ablation.** We show the effect of training SLD with patches ("Patches"), various output heatmap resolutions ("Res."), data augmentation ("Aug."), and various landmark counts (L). Best results are in **blue**.

The training had two phases – (i) scene coordinate initialization (1M iterations) and (ii) end-to-end training with differentiable RANSAC and pose estimation (100K iterations). **(3) HLoc [59, 60].** In HLoc's offline phase, we used our pseudo ground truth poses to triangulate 3D points from SuperPoint features and stored VLAD [25] features. In the online phase, we first retrieved the top-10 images similar to the query using VLAD-based nearest neighbor search and then ran feature matching between the query and retrieved images to find 2D–3D correspondences. Finally, we computed camera pose using our pose estimation pipeline. **(4) HLoc-L$_X$.** In this HLoc variant, we only store X scene landmarks and their descriptors. The landmarks were selected using our method (Section 3.4), We evaluated three settings; X=$\{300, 1000, 3000\}$. These baselines help us study the effect of reduced storage on HLoc's performance.

4.1. Results on INDOOR-6

Table 1 shows results on the INDOOR-6 dataset. The top half is for storage-free methods. The lower half is for methods that use high storage. Overall, our method, NBE+SLD, performs the best (highlighted in **green**) amongst storage-free methods for almost all metrics on all scenes. Our NBE+SLD and its efficient version NBE+SLD(E), outperforms the state-of-the-art DSAC* by a wide margin on all scenes. However, this could be partially because our dataset has SfM pseudo ground truth (pGT), whereas DSAC* is known to perform better on datasets with RGBD SLAM pGT [6] (see Section 4.2). It is worth noting that NBE is consistently more accurate than PoseNet even though both use identical CNN backbones. However, NBE+SLD is always more accurate than NBE, which confirms that heatmap representations is crucial for higher accuracy.

Visibility-weighted loss	×	×	✓	✓
Landmarks	50 (select)	50 (random)	50 (select)	100 (select)
Average Recall (10cm, 10°) ↑	7.11	7.32	7.67	8.36

Table 3. **NBE training ablation.** Effect of visibility-weighted loss and three different landmark selection methods on recall accuracy.

While NBE+SLD outperforms other storage-free methods, it is not yet competitive with HLoc, a method with high storage usage. However, NBE+SLD does outperform the compact HLoc variants, HLoc-L$_{300}$ and HLoc-L$_{1000}$ designed for reduced storage. Among methods that require high storage (bottom half of Table 1), HLoc+SLD, the method that computes pose from both HLoc's matches and SLD's predictions, has the best overall results, showing the complementary nature of scene landmark predictions. Figure 7 shows qualitative results on the INDOOR-6 dataset.

Storage. Table 2 reports storage used by the baselines and our method. NBE+SLD(E) with its EfficientNet-Lite0 backbone and DSAC* use similar storage, but NBE+SLD(E) outperforms DSAC* significantly. NBE+SLD uses 0.132 GB of storage, which is 4× higher than that of NBE+SLD(E) but is also consistently more

Method	7-SCENES																					
	chess			fire			heads			office			pumpkin			redkitchen			stairs			recall
	(cm.)↓	(deg.)↓	(%)↑	(cm.)↓	(deg.)↓	(%)↑	(cm.)↓	(deg.)↓	(%)↑	(cm.)↓	(deg.)↓	(%)↑	(cm.)↓	(deg.)↓	(%)↑	(cm.)↓	(deg.)↓	(%)↑	(cm.)↓	(deg.)↓	(%)↑	(%)↑
MS-Transformer^{+}	11	4.66	–	24	9.6	–	14	12.19	–	17	5.66	–	18	4.44	–	17	5.94	–	26	8.45	–	–
HLoc^{+}	2.4	0.77	94.2	1.8	0.75	93.7	0.9	0.59	99.7	2.6	0.77	83.2	4.4	1.15	55.1	4.0	1.38	61.9	5.1	1.46	49.4	76.7
DSAC*$^{+}$	1.8	0.59	97.8	1.7	0.77	94.5	1.0	0.66	98.8	2.7	0.79	83.9	3.9	1.05	62.0	3.9	1.24	65.5	3.5	0.93	78.0	82.9
NBE+SLD^{+}	2.2	0.75	93.7	1.8	0.73	94.1	0.9	0.68	96.6	3.2	0.91	74.8	5.6	1.55	44.6	5.3	1.52	45.7	5.5	1.41	44.6	70.6
HLoc	0.8	**0.11**	**100**	0.9	**0.24**	99.4	0.6	**0.25**	**100**	**1.2**	**0.20**	**100**	**1.4**	**0.15**	**100**	1.1	**0.14**	**98.6**	2.9	0.80	72.0	95.7
DSAC*	**0.5**	0.17	99.9	0.8	0.28	98.9	**0.5**	0.34	99.8	1.2	0.34	98.1	1.2	0.28	99.0	**0.7**	0.21	97.0	2.7	0.78	**92**	**97.8**
NBE+SLD	0.6	0.18	**100**	**0.7**	0.26	**99.6**	0.6	0.35	98.4	1.3	0.33	95.8	1.5	0.33	94.4	0.8	0.19	96.6	**2.6**	**0.72**	85.2	95.7

Table 4. **7-SCENES evaluation.** Median position error (cm), median rotation error (°), and recall at (5cm, 5°) metrics for methods evaluated using the original RGBD pGT indicated with the suffix $^{(+)}$ (see top four rows). The same metrics for methods evaluated using SfM pGT (see bottom three rows). HLoc and DSAC* results are from [6]. DSAC* is more accurate than both HLoc and NBLE+SLD, which have similar performance. The $^{(+)}$ variants for DSAC*, HLoc and NBE+SLD show a similar trend. Best result per column shown in **blue**.

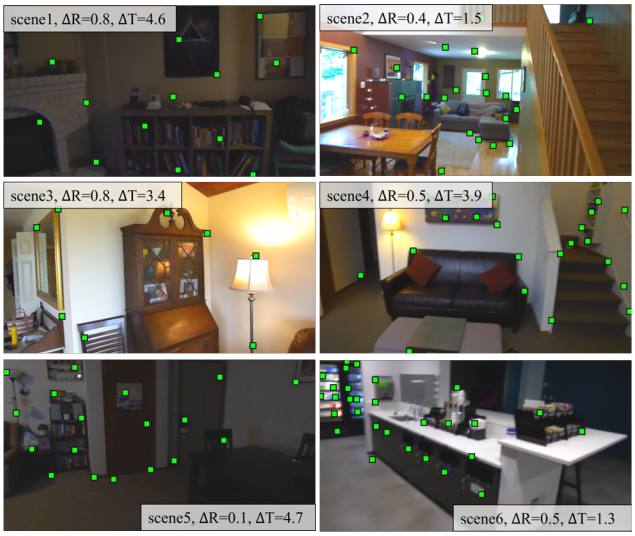

Figure 7. **Qualitative Results on INDOOR-6.** Detected landmarks (green) and pose errors ΔR (in deg.) and ΔT (in cm.) shown on test images from the six scenes. The images for scene1, scene5, and scene6 were taken in dark rooms in low light.

accurate. In contrast, HLoc, which has the highest overall accuracy, has a significantly higher storage requirement (up to 2.36 GB for the largest scene in the INDOOR-6 dataset). HLoc-L_{300} and HLoc-L_{1000} use less storage in comparison, but also have lower accuracy.

Ablation study for SLD. In Table 2 [Right], we report how various factors affect SLD training; namely, (1) training with patches (Patches) vs. whole image, (2) output resolutions (Res.), (3) data augmentation (Aug.), and (4) the number of scene landmarks (L). We also compare the performance to DSAC*. The main takeaway is that all three components of the training pipeline (training with patches, data augmentation, and high resolution) are crucial for SLD, and we achieve the highest recall when using 300 landmarks. All SLD results were obtained using these parameters.

Ablation study for NBE. The study reported in Table 3 shows that (1) the choice or count of landmarks does not matter much (i.e., randomly choosing landmarks from the 3D point cloud also works well); and (2) using known visibility slightly improves performance. Thus, we used 50 landmarks for NBE and trained it with visibility masks.

4.2. Results on 7-SCENES

Table 4 compares MS-Transformer [68], HLoc [59, 60], DSAC* [10], and our method on the 7-SCENES dataset. The top four rows show methods (marked +) using RGBD SLAM-based pGT for training and evaluation. The methods in the bottom three rows show results for methods using SfM pGT. The bottom row shows that performance of NBE+SLD is quite competitive. It is on par with HLoc while slightly less accurate than DSAC* (average recall 95.7% *vs*. 97.8%). In contrast, NBE+SLD^{+} (70.6%) is less accurate than HLoc^{+} (76.7%) and DSAC*$^{+}$ (82.9%). This indicates that SfM pGT, which is already known to produce a more reliable 3D point cloud than RGBD SLAM on 7-SCENES [6], is more suitable than RGBD pGT for training and evaluating our method on 7-SCENES.

5. Conclusion and Discussion

In this paper, we revisited the task of learned camera localization while aiming for a method which has low storage requirements but has high accuracy. Our main insight is that modern CNN architectures widely used for keypoint detection in human and object pose estimation are also suitable for detecting salient, scene-specific 3D landmarks and a small number of such landmarks may suffice. Our evaluation on a new dataset demonstrates that our method outperforms previous storage-free methods but is not as accurate as HLoc, one of the top retrieval-and-matching methods, although HLoc does require high storage. However, our landmark-based 2D–3D correspondences complement those of HLoc and combining the correspondences before computing pose boosts the accuracy of HLoc further.

Our method has some limitations. First, our CNNs are scene-specific and like other learned methods [10, 30] need many training images per scene. Thus, networks with shared, scene-agnostic backbones that can be fine-tuned to different scenes, should be explored. Second, to handle larger scenes effectively, it is worth leveraging scene partitioning strategies similar to ESAC [9]. Finally, jointly selecting landmarks and training the network to maximize overall pose accuracy could address the potential weakness of using a fixed set of landmarks selected prior to training.

References

[1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *CVPR*, 2016. 1, 2

[2] Hernan Badino, Daniel Huber, and Takeo Kanade. The CMU Visual Localization Data Set. http://3dvis.ri.cmu.edu/data-sets/localization, 2011. 3

[3] Vassileios Balntas, Shuda Li, and Victor Prisacariu. Relocnet: Continuous metric learning relocalisation using neural nets. In *ECCV*, 2018. 2

[4] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *CVIU*, 2008. 1, 2

[5] Alessandro Bergamo, Sudipta N Sinha, and Lorenzo Torresani. Leveraging structure from motion to learn discriminative codebooks for scalable landmark classification. In *CVPR*, 2013. 2, 3, 5

[6] Eric Brachmann, Martin Humenberger, Carsten Rother, and Torsten Sattler. On the limits of pseudo ground truth in visual camera re-localisation. In *ICCV*, 2021. 7, 8

[7] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *CVPR*, 2017. 1, 3

[8] Eric Brachmann and Carsten Rother. Learning less is more - 6D camera localization via 3D surface regression. In *CVPR*, 2018. 1, 3

[9] Eric Brachmann and Carsten Rother. Expert sample consensus applied to camera re-localization. In *ICCV*, 2019. 3, 8

[10] Eric Brachmann and Carsten Rother. Visual camera relocalization from rgb and rgb-d images using dsac. *T-PAMI*, 2021. 1, 2, 3, 5, 6, 8

[11] Bingyi Cao, Andre Araujo, and Jack Sim. Unifying deep local and global features for image search. In *ECCV*, 2020. 2

[12] Song Cao and Noah Snavely. Minimal scene descriptions from structure from motion models. In *CVPR*, 2014. 1

[13] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S. Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *CVPR*, 2020. 3

[14] Ondrej Chum and Jiri Matas. Matching with prosac - progressive sample consensus. In *CVPR*, 2005. 2

[15] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 3

[16] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPR workshops*, 2018. 1, 2

[17] Tien Do, Khiem Vuong, Stergios I. Roumeliotis, and Hyun Soo Park. Surface normal estimation of tilted images via spatial rectifier. In *ECCV*, 2020. 5

[18] Xuanyi Dong, Yi Yang, Shih-En Wei, Xinshuo Weng, Yaser Sheikh, and Shoou-I Yu. Supervision by registration and triangulation for landmark detection. *T-PAMI*, 2020. 2, 3, 4

[19] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A Trainable CNN for Joint Detection and Description of Local Features. In *CVPR*, 2019. 2

[20] ETH Zurich Computer Vision Group and Microsoft Mixed Reality & AI Lab Zurich. The ETH-Microsoft Localization Dataset. https://github.com/cvg/visloc-iccv2021, 2021. 3

[21] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981. 2, 5

[22] James Hays and Alexei A. Efros. Im2gps: estimating geographic information from a single image. In *CVPR*, 2008. 2

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

[24] Janghun Hyeon, Joohyung Kim, and Nakju Doh. Pose correction for highly accurate visual localization in large-scale indoor spaces. In *ICCV*, 2021. 2

[25] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, 2010. 2, 7

[26] Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image Matching across Wide Baselines: From Paper to Practice. *IJCV*, 2020. 3

[27] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 2021. 2

[28] Tong Ke and Stergios I. Roumeliotis. An efficient algebraic solution to the perspective-three-point problem. In *CVPR*, 2017. 5

[29] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *CVPR*, 2017. 1, 2, 5

[30] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *ICCV*, 2015. 1, 2, 5, 6, 8

[31] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014. 6

[32] Zakaria Laskar, Iaroslav Melekhov, Surya Kalia, and Juho Kannala. Camera relocalization by computing pairwise relative poses using convolutional neural network. In *ICCV*, 2017. 2

[33] Donghwan Lee, Soohyun Ryu, Suyong Yeon, Yonghan Lee, Deokhwa Kim, Cheolho Han, Yohann Cabon, Philippe Weinzaepfel, Nicolas Guérin, Gabriela Csurka, and Martin Humenberger. Large-scale localization datasets in crowded indoor spaces. In *CVPR*, 2021. 3

[34] Vincent Lepetit and Pascal Fua. Keypoint recognition using randomized trees. *T-PAMI*, 2006. 2, 3

[35] Xiaotian Li, Juha Ylioinas, and Juho Kannala. Full-frame scene coordinate regression for image-based localization. In *RSS*, 2018. 3

[36] Xiaotian Li, Juha Ylioinas, Jakob Verbeek, and Juho Kannala. Scene coordinate regression with angle-based reprojection loss for camera relocalization. In *ECCV workshop*, 2018. 3

[37] Yunpeng Li, Noah Snavely, Dan Huttenlocher, and Pascal Fua. Worldwide pose estimation using 3d point clouds. In *ECCV*, 2012. 2

[38] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 1, 2

[39] Simon Lynen, Torsten Sattler, Michael Bosse, Joel A Hesch, Marc Pollefeys, and Roland Siegwart. Get out of my lab: Large-scale, real-time visual-inertial localization. In *RSS*, 2015. 1

[40] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *IJRR*, 2017. 3

[41] Daniela Massiceti, Alexander Krull, Eric Brachmann, Carsten Rother, and Philip H. S. Torr. Random forests versus neural networks - what's best for camera localization? In *ICRA*, 2017. 3

[42] Lili Meng, Jianhui Chen, Frederick Tung, James J Little, Julien Valentin, and Clarence W de Silva. Backtracking regression forests for accurate camera relocalization. In *IROS*, 2017. 3

[43] Krystian Mikolajczyk and Cordelia Schmid. Scale and affine invariant interest point detectors. *IJCV*, 2004. 2

[44] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *NeurIPS*, 2017. 2

[45] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP*, 2009. 2

[46] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016. 2, 3, 4

[47] Tony Ng, Adrian Lopez-Rodriguez, Vassileios Balntas, and Krystian Mikolajczyk. Reassessing the limitations of cnn methods for camera pose regression. *arXiv*, 2021. 2

[48] Markus Oberweger, Mahdi Rad, and Vincent Lepetit. Making deep heatmaps robust to partial occlusions for 3d object pose estimation. In *ECCV*, 2018. 3

[49] Mustafa Ozuysal, Michael Calonder, Vincent Lepetit, and Pascal Fua. Fast keypoint recognition using random ferns. *T-PAMI*, 2009. 3

[50] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 6

[51] Georgios Pavlakos, Xiaowei Zhou, Aaron Chan, Konstantinos G Derpanis, and Kostas Daniilidis. 6-dof object pose from semantic keypoints. In *ICRA*, 2017. 2, 3

[52] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *CVPR*, 2019. 3

[53] Noé Pion, Martin Humenberger, Gabriela Csurka, Yohann Cabon, and Torsten Sattler. Benchmarking image retrieval for visual localization. In *3DV*, 2020. 2

[54] Francesco Pittaluga, Sanjeev J Koppal, Sing Bing Kang, and Sudipta N Sinha. Revealing scenes by inverting structure from motion reconstructions. In *CVPR*, 2019. 1, 2

[55] Mahdi Rad and Vincent Lepetit. Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In *ICCV*, 2017. 3

[56] Rahul Raguram, Ondrej Chum, Marc Pollefeys, Jiri Matas, and Jan-Michael Frahm. Usac: A universal framework for random sample consensus. *T-PAMI*, 2013. 2

[57] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: an efficient alternative to SIFT or SURF. In *ICCV*, 2011. 1, 2

[58] Soham Saha, Girish Varma, and C. V. Jawahar. Improved visual relocalization by discovering anchor points. In *BMVC*, 2018. 2

[59] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 2, 3, 7, 8

[60] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 2, 3, 7, 8

[61] Paul-Edouard Sarlin, Ajaykumar Unagar, Måns Larsson, Hugo Germain, Carl Toft, Victor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, and Torsten Sattler. Back to the Feature: Learning Robust Camera Localization from Pixels to Pose. In *CVPR*, 2021. 2

[62] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Improving image-based localization by active correspondence search. In *ECCV*, 2012. 1

[63] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *CVPR*, 2018. 3

[64] Torsten Sattler, Tobias Weyand, B. Leibe, and Leif P. Kobbelt. Image retrieval for image-based localization revisited. In *BMVC*, 2012. 3

[65] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression. In *CVPR*, 2019. 1, 2

[66] Johannes Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *CVPR*, 2016. 6

[67] Johannes Schönberger, Hans Hardmeier, Torsten Sattler, and Marc Pollefeys. Comparative Evaluation of Hand-Crafted and Learned Local Features. In *CVPR*, 2017. 1, 3

[68] Yoli Shavit, Ron Ferens, and Yosi Keller. Learning multi-scene absolute pose regression with transformers. In *ICCV*, 2021. 2, 8

[69] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *CVPR*, 2013. 1, 2, 3, 6

[70] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017. 3

[71] Hyun Soo Park, Yu Wang, Eriko Nurvitadhi, James C Hoe, Yaser Sheikh, and Mei Chen. 3d point cloud reduction using mixed-integer quadratic programming. In *CVPR Workshops*, 2013. 1

[72] Pablo Speciale, Johannes Schonberger, Sing Bing Kang, Sudipta N Sinha, and Marc Pollefeys. Privacy preserving image-based localization. In *CVPR*, 2019. 1

[73] Pablo Speciale, Johannes Schonberger, Sudipta N Sinha, and Marc Pollefeys. Privacy preserving image queries for camera localization. In *ICCV*, 2019. 1

[74] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019. 3

[75] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. Inloc: Indoor visual localization with dense matching and view synthesis. In *CVPR*, 2018. 3

[76] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv*, 2019. 6

[77] Bugra Tekin, Sudipta N Sinha, and Pascal Fua. Real-time seamless single shot 6d object pose prediction. In *CVPR*, 2018. 2

[78] Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub Heijnen, and Vassileios Balntas. Sosnet: Second order similarity regularization for local descriptor learning. In *CVPR*, 2019. 2

[79] Giorgos Tolias, Yannis Avrithis, and Hervé Jégou. To aggregate or not to aggregate: Selective match kernels for image search. In *ICCV*, 2013. 2

[80] Akihiko Torii, Relja Arandjelović, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *CVPR*, 2015. 2

[81] A. Toshev and C. Szegedy. DeepPose: Human pose estimation via deep neural networks. In *CVPR*, 2014. 3

[82] Julien Valentin, Matthias Niessner, Jamie Shotton, Andrew Fitzgibbon, Shahram Izadi, and Philip H. S. Torr. Exploiting uncertainty in regression forests for accurate camera relocalization. In *CVPR*, 2015. 3

[83] Johanna Wald, Torsten Sattler, Stuart Gododetz, Tommaso Cavallari, and Federico Tombari. Beyond controlled environments: 3d camera re-localization in changing indoor scenes. In *ECCV*, 2020. 3

[84] Bing Wang, Changhao Chen, Chris Xiaoxuan Lu, Peijun Zhao, Niki Trigoni, and Andrew Markham. Atloc: Attention guided camera localization. In *AAAI*, 2020. 1, 2

[85] Shuzhe Wang, Zakaria Laskar, Iaroslav Melekhov, Xiaotian Li, and Juho Kannala. Continual learning for image-based camera localization. In *ICCV*, 2021. 3

[86] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *CVPR*, 2016. 2, 3

[87] Philippe Weinzaepfel, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. Visual localization by learning objects-of-interest dense match regression. In *CVPR*, 2019. 3

[88] P. Wenzel, R. Wang, N. Yang, Q. Cheng, Q. Khan, L. von Stumberg, N. Zeller, and D. Cremers. 4Seasons: A cross-season dataset for multi-weather SLAM in autonomous driving. In *GCPR*, 2020. 3

[89] Tobias Weyand, Ilya Kostrikov, and James Philbin. Planet-photo geolocation with convolutional neural networks. In *ECCV*, 2016. 2

[90] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, 2018. 3

[91] Zixuan Xu, Banghuai Li, Ye Yuan, and Miao Geng. Anchorface: An anchor-based facial landmark detector across large poses. In *AAAI*, 2021. 3

[92] Fei Xue, Xin Wu, Shaojun Cai, and Junqiu Wang. Learning multi-view camera relocalization with graph neural networks. In *CVPR*, 2020. 2

[93] Luwei Yang, Ziqian Bai, Chengzhou Tang, Honghua Li, Yasutaka Furukawa, and Ping Tan. Sanet: Scene agnostic network for camera localization. In *ICCV*, 2019. 3

[94] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *ECCV*, 2016. 2

[95] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2015. 4

[96] Amir Zadeh, Yao Chong Lim, Tadas Baltrusaitis, and Louis-Philippe Morency. Convolutional experts constrained local model for 3d facial landmark detection. In *CVPR Workshops*, 2017. 3