2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

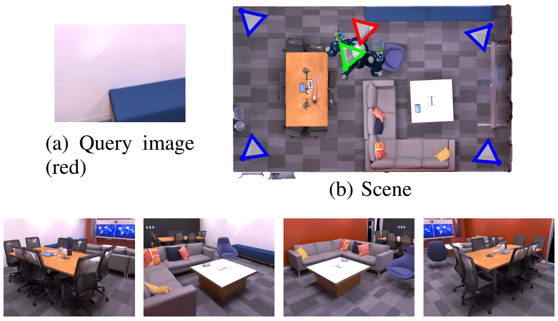
Learning to Localize in New Environments from Synthetic Training Data

Dominik Winkelbauer Maximilian Denninger Rudolph Triebel

Abstract— **Most existing approaches for visual localization either need a detailed 3D model of the environment or, in the case of learning-based methods, must be retrained for each new scene. This can either be very expensive or simply impossible for large, unknown environments, for example in search-and-rescue scenarios. Although there are learning-based approaches that operate scene-agnostically, the generalization capability of these methods is still outperformed by classical approaches. In this paper, we present an approach that can generalize to new scenes by applying specific changes to the model architecture, including an extended regression part, the use of hierarchical correlation layers, and the exploitation of scale and uncertainty information. Our approach outperforms the 5-point algorithm using SIFT features on equally big images and additionally surpasses all previous learning-based approaches that were trained on different data. It is also superior to most of the approaches that were specifically trained on the respective scenes. We also evaluate our approach in a scenario with only very few reference images, showing that under such more realistic conditions our learning-based approach considerably exceeds both existing learning-based and classical methods.**

I. INTRODUCTION

Robot localization is one of the major, fundamental tasks in robotics, and it has been investigated already for several decades. Concretely, the robot localization task consists of determining the coordinates of the robot's position and orientation with respect to a global map frame, based on its observed sensor input. To achieve this, some kind of map representation must be available beforehand, which distinguishes the localization task from the more general problem of Simultaneous Localization and Mapping (SLAM). This has the major advantage that the localization process itself is simpler, because it only requires finding a "match" between the current observation and the map representation. The downside however is that building an accurate and useful map of the environment is itself an nontrivial process, which is most often performed by again employing SLAM techniques, albeit this is done offline, i.e. before actually deploying the robot. An additional problem of handling localization separately from mapping is that, in terms of sensing modality the map representation must be comparable to the observations made during localization, because otherwise a good matching is difficult to achieve. For most current robot systems, this means that mapping is done with the same sensor setup and the same parameter set as is used during the actual localization process. This implies that a robot that is to be deployed in a new environment needs to build a map of that environment first with its own sensor stack, although some map representation might already be available. In this paper, we are going to address these issues with a novel approach. The only sensing modality we consider are standard RGB cameras, which makes our framework very flexible and versatile. Concretely, our map representation simply consists of a set of reference images where each has an assigned 6D transform with respect to the global map frame. Thus, to determine the robot's pose we compare the currently observed image frame with all reference images and estimate the transforms between the current pose and the reference images using a deep learning architecture. Although there already exist similar approaches [1]–[3] , they still are inaccurate on scenes not contained in their training data. We show that training our specific network architecture only with synthetic data is sufficient to localize a robot in an unknown, real environment, and the accuracy of this learning-based localization outperforms classical model-based approaches. Furthermore, we show that in the case where only a few reference images are available (see fig. 1), our method considerably outperforms classical methods.



(a) Query image (red)

(b) Scene

(c) Reference images (blue)

Fig. 1. Localization with reference images from only four given poses (blue triangles in (b), reference views in (c)). Note that finding an accurate pose from the query view in (a) is very difficult, but our approach makes a very good prediction (green triangle in (b)) to the ground truth pose (red).

II. RELATED WORK

A. Image retrieval

Image retrieval can be used to get a rough but fast estimate of the pose of a given image [4]. By using techniques like DenseVLAD [5], NetVLAD [6] or Disloc [7] a compact representation is calculated for each image, then the pose of the image with the closest descriptor is used. As the accuracy of image retrieval is low, more sophisticated approaches [3],

German Aerospace Center (DLR), Institute of Robotics and Mechatronics, Dep. of Perception and Cognition, Münchner Str. 20, 82234 Wessling, Germany {firstname.lastname} at dlr.de

978-1-7281-9077-8/21/$31.00 ©2021 IEEE

[8], including ours, further refine the pose estimate and use image retrieval only as a first step to reduce the search space.

B. 3D structure-based localization

3D structure-based methods do pose estimation based on matches between points in the image and points in a 3D model of the scene. Although this leads to accurate predictions, building such a 3D model requires that enough reference images are available. Our approach can also handle situations with a low number of reference images. There also exist learning-based approaches using CNNs [9]–[12] or random forests [9], [10], [13]–[15] which predict for every pixel of a given image the corresponding matching 3D coordinate. Altough these approaches are more memory efficient during inference, they need a 3D model for training or in the case of DSAC++ [11] need at least expensive retraining for each new scene.

C. Absolute pose regression

Absolute pose regression methods are learning-based approaches that try to regress directly the absolute pose of a given query image. [16] initially proposed this idea in the form of PoseNet, a CNN based on a pretrained GoogLeNet [17]. Follow-up approaches have interchanged the GoogLeNet with various other backbones [18]–[23]. Independent of the exact architecture, [24] discovered that absolute pose estimation models poorly generalize to query images from the same scene that are further away from the training images. By training on a big synthetic dataset our approach does not overfit on specific camera constellations. Bayesian PoseNet [25] is trained via monte carlo dropout which we also use in our network to improve the triangulation step. Other approaches [20], [26]–[28] suggest to additionally train the network on auxiliary losses like semantic segmentation or relative pose estimation. As there are always losses which focus on scene specific properties, a retraining for each new scene is still necessary. We do not suffer from this as we train only once.

D. Relative pose regression

Some prior methods [1]–[3], [29] estimate the relative pose between the query image and close-by reference images whose absolute poses are known. Then, the absolute pose of the query image is triangulated. Although these approaches are in principle built to generalize, they do not perform better than models that were specifically trained on these scenes[3], [24]. In contrast, we show for the first time that it is possible and even beneficial to generalize from unrelated data without retraining. Our approach is most similar to EssNet [3], however with some major differences, as we use an extended regression part and hierarchical correlation layers. Also, we represent the relative pose as a translation direction and a quaternion, which removes the need of scene-specific weighting, while no projection layer is necessary and the predicted pose does not contain ambiguities, which is the case when using essential matrices. Additionally, we require a smaller image size which makes our approach require less computation time, which is limited on mobile robots.

III. Problem description

Visual relocalization can be defined as estimating the absolute pose in the form of a homogeneous transformation matrix $T_q \in \mathbb{R}^{4\times4}$ of a given query image $\mathcal{I}_q$. Thereby, only a set of reference images $\mathcal{I}_i$ together with their absolute poses T_i is given. We call the rotational part of such a transformation matrix then $R_i \in \mathbb{R}^{3\times3}$ and the translational part $t_i \in \mathbb{R}^3$. Instead of estimating the absolute pose directly, relative pose estimation methods like our approach estimate the relative pose $T_{i\to q}$ between the query image $\mathcal{I}_q$ and one or multiple reference images $\mathcal{I}_i$ and combine these with the known absolute poses of the reference images to get an estimate for the absolute pose of the query image.

IV. Pose estimation pipeline

For a given query image we first retrieve five reference images, then estimate the relative pose between each query-reference pair and finally triangulate the absolute pose of the given query image.

A. Image retrieval

Similar to Zhou et al. [3], we collect the five closest reference images according to the Euclidean distance of the respective DenseVLAD descriptors, while skipping images whose distance to the camera centers of the already selected images is not in range $[0.05\text{m}, 10\text{m}]$.

B. Relative pose estimation

To estimate the relative pose between two given images we train a siamese neural network, which we call ExReNet (see Fig. 2). This consists of three stages: individual feature extraction, feature matching, and relative pose regression.

Feature extraction: A ResNet50 pretrained on ImageNet is used up to its third stack to extract a feature map of $8 \times 8 \times 1024$ per image. We further apply two upsampling steps in a UNet-like [30] fashion to obtain a second feature map of $32 \times 32 \times 256$ per image.

Feature matching: As in Zhou et al. [3] we use a correlation layer to combine both feature maps. An extensive correlation layer l computes for two given feature maps F_1 and F_2 with spatial resolution s_l a three-dimensional correlation map C_l by performing the dot product between all possible feature vector combinations, i.e.

$$C_l(y_1, x_1, w) = F_1(y_1, x_1)^T F_2(y_2, x_2), \tag{1}$$

where $w = x_2 s_l + y_2$ and $y_1, x_1, y_2, x_2 \in \{0, ..., s_l - 1\}$.

We found that, especially if the baseline between the two images is small, having higher resolution feature matches improves the relative pose estimates drastically. However, for a given image side length s_l the extensive correlation layer performs s_l^4 dot products, which makes it computationally infeasible for higher resolutions. To counteract this issue we propose to apply multiple correlation layers in a hierarchical fashion: Thereby, only the first low resolution correlation layer performs extensive matching, while the following correlation layers are guided by the matchings produced at the previous layer. For a given feature vector

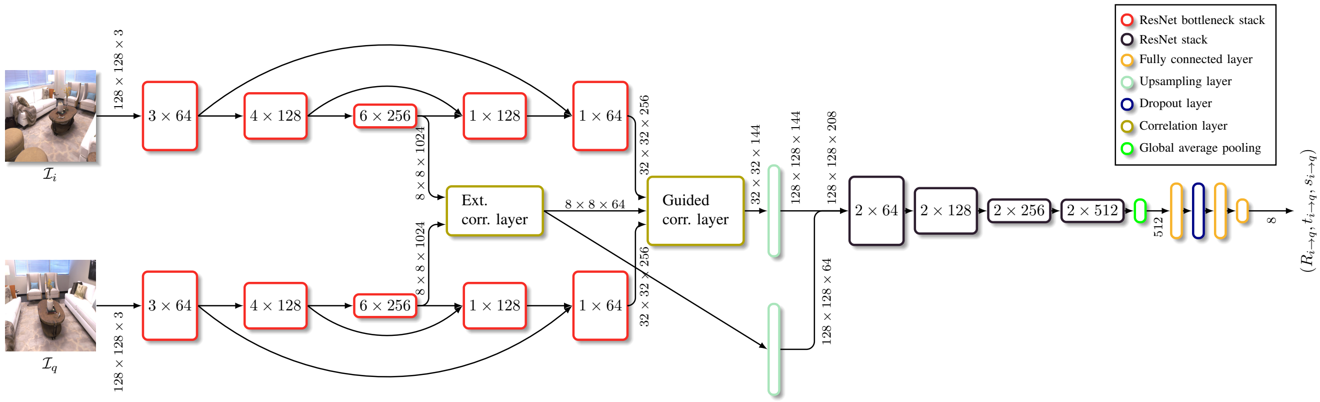


Fig. 2. Visualization of our relative pose estimation network ExReNet. Inside the ResNet stacks the number of blocks and the feature sizes are noted.

a guided correlation layer only looks for potential matches in the region where the match of the corresponding coarse feature vector in the previous correlation layer has been found. Fig. 3 visualizes this concept using an example image pair from "7-Scenes". In detail, a correlation layer l guided by the previous correlation layer p is defined as follows:

$$C_l(y_1, x_1, w) = F_1(y_1, x_1)^T F_2((M_y(y_1, x_1) - b_l)u_l + y_2, \\ (M_x(y_1, x_1) - b_l)u_l + x_2) \quad (2)$$

where the size of the search area is $d_l = u_l(1 + 2b_l)$ and the channel coordinate $w = x_2 d_l + y_2$. Feature coordinates are defined as $y_1, x_1 \in \{0, ..., s_l - 1\}$ and $y_2, x_2 \in \{0, ..., d_l - 1\}$. The border b_l is added around the search area to increase the number of possible correct matches. The upscale factor $u_l = s_l/s_p$ describes the change in spatial resolution between the current correlation layer l and the previous one p. The guidance from the previous layer p is represented as the corresponding match M:

$$M_y(y, x) \equiv w_{max}(y, x) \mod s_p \quad \text{or} \quad (3)$$

$$M_x(y, x) = \lfloor \frac{w_{max}(y, x)}{s_p} \rfloor \quad (4)$$

where $w_{max}(y, x) = \arg\max_w C_p(\lfloor y/u_l \rfloor, \lfloor x/u_l \rfloor, w)$.

We found that training a network using hierarchical correlation layers with only the pose regression loss does not fully make use of the network's potential. Even if the non-differentiable argmax is replaced by an equivalent softmax, the training gets stuck in a local minimum, such that the network only makes use of the high-level extensive correlation layer and neglects any further guided correlation layers. To solve this issue, we train our network using an additional auxiliary loss, that makes sure the feature combinations with maximum dot products correspond to actual correct feature matches. Using depth information we compute for every pixel in the first image the corresponding matching pixel in the second image. Afterwards, we can compute for every feature vector combination $F_1(y_1, x_1), F_2(y_2, x_2)$ the number of pixels $n(y_1, x_1, w)$ that are included in the spatial region covered by $F_1(y_1, x_1)$ and whose matching pixel from the other image is covered by $F_2(y_2, x_2)$.

For an extensive correlation layer l the auxiliary loss $\mathcal{L}^l_{aux}(y, x, w)$ for a given feature combination is defined as

$$\mathcal{L}^l_{aux}(y, x, w) = \frac{1}{\sum_q \delta_{w,q}} \sum_q \delta_{w,q} \cdot e(y, x, w, q) \quad (5)$$

$$e(y, x, w, q) = \max(0, C_l(y, x, q) - C_l(y, x, w) + 1) \quad (6)$$

$$\delta_{w,q} = \begin{cases} 1 & \text{if } n_l(y, x, w) > 0 \text{ and } n_l(y, x, q) = 0 \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

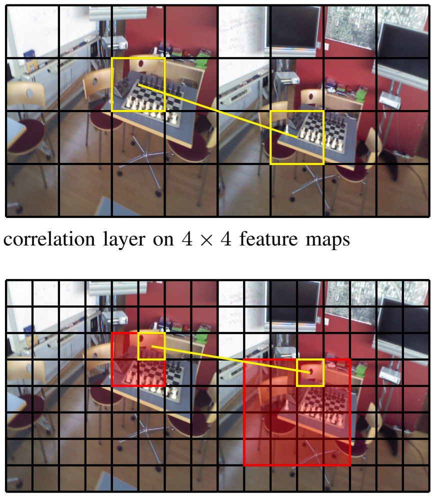
We formulate the auxiliary loss in the form of a triplet loss, where positive pairs are feature matches that correspond to correct pixel matches and negative pairs are matches that correspond to no actual matches. For the guided correlation layer the auxiliary loss can be formulated accordingly. The total auxiliary loss is defined as the mean over all correlation layers and across all their feature combinations.

In ExReNet we use one extensive corr. layer applied to 8×8 feature maps and one guided corr. layer applied to 32×32 using a border $b = 1$. Compared to using only one corr. layer applied to 32×32, this reduces the total training memory consumption by 32% and training time by 19% while being more accurate as higher dimensional feature vectors are taken into account.

Pose regression: In ExReNet we first upsample the feature matches of both corr. layers to 128×128, concatenate them along the channel dimension and then apply a ResNet18. The spatial dimension is reduced afterwards via a global average pooling layer, followed by three fully-connected layers: Two with size 512 and one to map the features to the output vector which has a length of seven. This output consists of a three dimensional vector representing the translation direction $\hat{t}$ and a four dimensional quaternion q.

Our regressor has substantially more layers compared to existing architectures[2], [3], [31], as we found that especially layers that operate before collapsing the spatial dimension increase the generalization capability to new scenes.

Loss function: The used loss consists of one L1-loss for translation and one for rotation: $\mathcal{L}_{pose} = |\hat{t} - \hat{t^*}| + \mu|q - q^*|$ In contrast to previous approaches, we train our network to predict the inverse relative pose, as the inverse is also used in the triangulation step: So, for a given image pair



(a) Extensive correlation layer on 4×4 feature maps

(b) Correlation layer on 8×8 feature maps guided by the results from the correlation layer above using a border b of 0.5

Fig. 3. Visualizes two correlation layers whereby one is guided by the other: In each layer one correct matching is visualized in yellow. The guidance of the upper corr. layer is visualized in red in the lower one.

$(\mathcal{I}_i, \mathcal{I}_q)$ the ground truth relative translation direction $\hat{t}^*$ equals $-R^T_{i\to q}\hat{t}_{i\to q}$ and the ground truth relative rotation q^* corresponds to $R^T_{i\to q}$. As we do not learn the fully-scaled translation, but only the normalized direction vector, no scene-specific weighting between translation and rotation loss is required, and as both parts have the same scale, we found that $\mu = 1$ performs well.

Scale estimation: Previous methods [1]–[3] only make use of the direction of the relative translation between the two given images. However, by using the translation scale we could further improve the pose estimation accuracy in the triangulation step. So, we predict the scale s using an extra output dimension. The loss function is adjusted accordingly: $\mathcal{L}_{pose} = |\hat{t} - \hat{t^*}| + \mu|q - q^*| + \beta|s - \|t^*\|_2|$. As we only train on indoor scenes, where the scale has a maximum value of a few meters, we use $\beta = 1$.

Uncertainty estimation: By using Monte Carlo dropout [32] we obtain uncertainty estimates which are then used to make the triangulation step described in the next section more precise. One dropout layer is therefore placed between the two hidden fully connected layers in the last part of the network using a dropout rate of 0.1. At test time, we sample 100 pose estimations, use their mean as final pose estimate and the variance of the predicted camera center as uncertainty estimate $\sigma_{i\to q}$.

C. Triangulation

By combining two relative pose estimates $(R_{i\to q}, t_{i\to q})$ and $(R_{j\to q}, t_{j\to q})$ the absolute pose $(R_{\mathcal{I}_q}, t_{\mathcal{I}_q})$ of the query image $\mathcal{I}_q$ is estimated. Here, we extend the absolute pose estimation step proposed by Zhou et al. [3], which is summarized in the following paragraph.

1) Triangulation baseline: The camera center c_q of the query image is estimated by triangulation [33] of the two rays defined by the two relative pose estimates. For a given relative pose estimate $(R_{i\to q}, t_{i\to q})$ the ray is defined as $c_{\mathcal{I}_i} - \lambda_{i\to q}v_i$ with the ray direction $v_i = R^T_{\mathcal{I}_i}R^T_{i\to q}\hat{t}_{i\to q}$ and the position $\lambda_{i\to q}$ along the ray. The rotation $R_{\mathcal{I}_q}$ of the query image is estimated by the average rotation between $R_{i\to q}R_{\mathcal{I}_i}$ and $R_{j\to q}R_{\mathcal{I}_j}$. In this way, a hypothesis $T_{\mathcal{H}(i,j)}$ of the query pose is formed.

As there might be outliers in the relative pose estimations, RANSAC is used to find a hypothesis that is consistent based on the five retrieved reference images. Given a hypothesis $T_{\mathcal{H}(i,j)}$ a reference image $\mathcal{I}_k$ counts as an inlier, if the angle α_k between the direction of its ray v_k and the vector from its camera center $c_{\mathcal{I}_k}$ to the camera center of the hypothesis $c_{\mathcal{H}(i,j)}$ is smaller than a threshold α_{max}. Finally, the hypothesis with the most inliers is used.

2) Our adaptation: We found that using only the angle α_k to determine inliers can often lead to false positives. To prevent such cases, we propose to additionally use the translation scale estimation $s_{k\to q}$ for determining inliers. Thus, in addition to the constraint of $\alpha_k < \alpha_{max}$, a reference image $\mathcal{I}_k$ has to also fulfill the criteria $s_{min} < s_{\mathcal{H}}/s_{k\to q} < s_{max}$ to count as an inlier for a hypothesis $T_{\mathcal{H}(i,j)}$. Here, $s_{k\to q}$ is the predicted scale between the query image $\mathcal{I}_q$ and the reference image $\mathcal{I}_k$ and $s_{\mathcal{H}}$ is the actual scale of the relative translation between its camera center $c_{\mathcal{I}_k}$ to the camera center of the hypothesis $c_{\mathcal{H}(i,j)}$. For all our experiments we used $\alpha_{max} = 15°$, $s_{min} = 0.5$ and $s_{max} = 2.0$.

To further improve the absolute pose estimation, we additionally take the uncertainty estimate $\sigma_{k\to q}$ of the network into account: We use uncertainty information to disambiguate cases where multiple hypotheses have the same number of inliers. If that is the case, we use the hypothesis $T_{\mathcal{H}(i,j)}$ with the smallest mean uncertainty $0.5\,(\sigma_{i\to q} + \sigma_{j\to q})$ of both corresponding relative pose estimations.

V. Experiments

A. General procedure

If not stated otherwise, all our networks are trained only on synthetic data generated using BlenderProc[35]. As 3D models, we select the 1500 most furnished houses from the SUNCG dataset [36], in which we randomly sample valid camera poses. We use two different synthetic training sets, one for densely and one for sparsely covered scenes. While we sample for the dense training set camera poses along all six DoF, for the sparse training set we mostly only vary along three degrees of freedom (x, y and yaw angle), which is done to simulate a moving robot.

The network itself is trained on randomly sampled image pairs from the respective training set, whereby we require that the overlap $o_{i,j}$ is bigger than a configured threshold o_{max}, the distance between the camera centers $\left\|c_{\mathcal{I}_i} - c_{\mathcal{I}_j}\right\|_2$ is smaller than d_{max}, and that the rotation difference γ is smaller than a threshold γ_{max}. Here, the overlap factor $o_{i,j}$ of two given images $\mathcal{I}_i$ and $\mathcal{I}_j$ is defined as the ratio of scene coordinates that are located at most $p_{max} = 0.2m$ away from a scene coordinate from the other image. For the dense training set we use $o_{max} = 30\%$, $d_{max} = 0.6m$ and $\gamma_{max} = 30°$. For the sparse training set we use $o_{max} = 10\%$ as

TABLE I

Comparison of a big variety of approaches on the 7-Scenes dataset: The models are compared based on their median translation and rotation errors averaged across all scenes.

(a) Methods trained on 7-Scenes or requiring 3D models for each scene

	Scene-specific	Errors
3D structure-based		
Active Search [34]	Yes	0.05m / 2.46°
DSAC++ [11]	Yes	0.04m / 1.10°
Absolute pose estimation		
PoseNet [16]	Yes	0.44m / 10.44°
Geo. PN [31]	Yes	0.23m / 8.12°
MapNet [26]	Yes	0.21m / 7.78°
Relative pose estimation		
Relative PN [1]	No	0.21m / 9.28°
RelocNet [2]	No	0.21m / 6.73°
EssNet [3]	No	0.22m / 8.03°
NC-EssNet [3]	No	0.21m / 7.50°
AnchorNet [21]	Yes	0.09m / 6.74°
CamNet [29]	No	0.04m / 1.69°

(b) Methods trained on unrelated data or with no training at all

			Input size	Traindata	Errors
Image retrieval					
DenseVLAD [5]			640 × 480	-	0.26m / 13.11°
Relative pose estimation					
Sift+5pt [3]			640 × 480	-	0.08m / 1.99°
Sift+5pt			149 × 110	-	0.13m / 2.86°
Relative PN [1]			224 × 224	University	0.36m / 18.37°
RelocNet [2]			224 × 224	ScanNet	0.29m / 11.29°
Ours					
EssNet [3]			128 × 128	SUNCG	0.25m / 9.76°
ExReNet	Scale: ✗	Unc: ✗	128 × 128	ScanNet	0.12m / 3.30°
ExReNet	Scale: ✗	Unc: ✗	128 × 128	SUNCG	0.11m / 2.97°
ExReNet	Scale: ✓	Unc: ✗	128 × 128	SUNCG	0.10m / 2.98°
ExReNet	Scale: ✓	Unc: ✓	128 × 128	SUNCG	0.09m / 2.72°

the only condition. During training, all images are randomly augmented by applying random brightness, contrast and saturation. All models are trained using the Adam optimizer using a learning rate of 5e−5 for 500k iterations.

B. Dense reference image coverage

In this section, we evaluate ExReNet on scenes that are densely covered by reference images in the form of the commonly used 7-Scenes dataset. In the case of scene-agnostic approaches, like ours, the training images of 7-Scenes are used as reference images during inference.

1) Generalization to unseen scenes: Table Ia and Ib list the pose estimation accuracy of a variety of approaches on 7-Scenes. It is apparent that 3D structure-based methods still outperform all other approaches. However, as stated earlier, they need either a detailed 3D model or require extensive scene-specific training. Apart from that, our scene-agnostic approach outperforms all methods trained on unrelated data and also most of the approaches that were specifically trained on 7-Scenes. Especially, the rotational error of our network is much lower compared to networks trained on 7-Scenes, and it is more in the range of classical methods. Using scale and uncertainty information further improves especially the translation error, which is reasonable as both extensions are used for the estimation of the camera center of the query image. In this way, we outperform handcrafted SIFT features together with the 5-point algorithm on an equivalent resolution. This also stays true, if the number of reference images is artificially reduced (see Fig. 5a). To better understand the influence of synthetic data on these results, we also trained our model on ScanNet which surprisingly results in a slightly worse performance compared to using synthetic data. We relate this phenomenon to the fact that the images in ScanNet[37] are not sampled uniformly as in our synthetic dataset, but are recorded on trajectories.

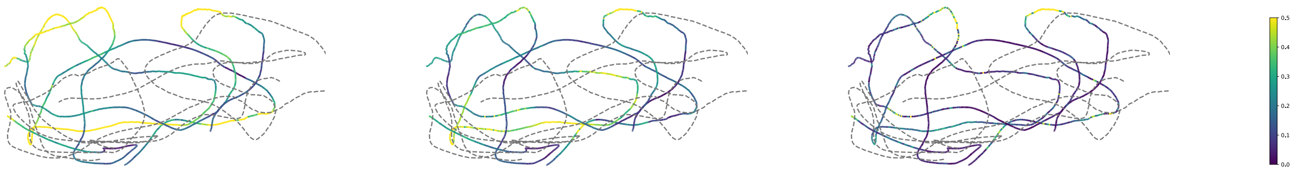
2) Overfitting effect: By further investigating why our network trained on synthetic data outperforms many methods trained on 7-Scenes itself, we discovered similar results as had been shown for absolute pose estimation methods [24]: Relative pose estimation methods also tend to overfit on training trajectories. We visualize this phenomenon in Fig. 4. Here, we compare the errors of the predicted camera centers between an EssNet trained on 7-Scenes, an ExReNet also trained on 7-Scenes and one only trained on synthetic data. It is clearly visible that the error of the predictions made by EssNet increases if the test images are further away from the training trajectories. When training ExReNet on 7-Scenes this phenomenon is weaker but still present. However, it cannot be observed in the figure based on ExReNet trained on synthetic data. From that we conclude that also relative pose estimation methods can overfit on the training trajectories, and the training on unrelated synthetic data that incorporates a big variety of camera poses helps the generalization.

TABLE II

Results of training ExReNetClassic with different feature matching methods and regression parts.

	Avg. of median errors
ExReNet	0.11m / 2.97°
ExReNetClassic (one corr. layer on 8 × 8)	0.15m / 4.19°
ExReNetClassic with no ResNet blocks in regressor	0.39m / 9.01°
ExReNetClassic with EssNet-like regression part	0.30m / 7.22°
ExReNetClassic with concatenation channel-wise	0.25m / 5.54°
ExReNetClassic with concatenation after pooling	0.36m / 7.89°

3) Ablation studies: The fact that EssNet, trained on our synthetic training data, still results in bad generalization (see table Ib) hints that our proposed architectural changes also have beneficial effects: As can be seen in table II, when using only one correlation layer on 8 × 8 feature maps,



(a) EssNet trained on 7-Scenes (b) ExReNet trained on 7-Scenes (c) ExReNet trained only on synthetic data

Fig. 4. Visualization of the overfitting effect of relative pose estimation networks. Here, we show the deviations between predicted and ground truth camera center on the fire scene from 7-Scenes. The gray dashed line represents the 7-Scenes reference images (training data for a and b), while the query images are visualized with a color depending on the prediction error. The mapping between deviation in meters and color is shown in the bar on the right.

TABLE III

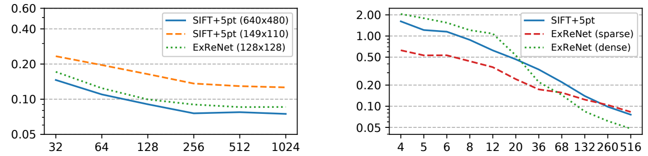
COMPARISON OF MULTIPLE APPROACHES ON THE SPARSE COVERED SCENES: THIS ERROR CONSIDERS QUERY IMAGES WHERE A PREDICTION WAS POSSIBLE. THE RATIO OF IMAGES FOR WHICH A PREDICTION WAS POSSIBLE IS GIVEN IN BRACKETS BEHIND EACH ERROR.

	replica-based dataset	custom real-world data
DenseVLAD	2.85m / 88.49° (100 %)	1.80m / 19.26° (100 %)
Sift+5pt + IR fallback	2.16m / 58.11° (100 %)	1.62m / 20.09 ° (100 %)
EssNet	1.50m / 160.98°(100 %)	1.10m / 125.09°(100 %)
Sift+5pt (Lowe's ratio: 0.8)	0.50m / 18.68° (36 %)	0.65m / 10.66° (26%)
Sift+5pt (Lowe's ratio: 0.9)	1.66m / 104.91°(75 %)	1.05m / 108.3° (79%)
ExReNet Scale: ✗, Unc: ✗	0.92m / 29.74° (100 %)	0.64m / 10.00° (100 %)
ExReNet Scale: ✓, Unc: ✗	0.77m / 17.22° (100 %)	0.61m / 8.37° (100 %)
ExReNet Scale: ✓, Unc: ✓	0.64m / 9.39° (100 %)	0.61m / 7.74° (100 %)

the accuracy decreases, however, it is still more accurate than most methods listed in table I. If the regressor is scaled down further, the model performs considerably worse. So, it seems that using a larger regressor, that can fully learn the complex mapping from the feature matches to the resulting relative pose, has a stronger ability to generalize. Furthermore, the ablation study confirms that the performance decreases considerably when replacing the correlation layer with concatenation.

C. Sparse map

In contrast to 7-scenes, where each scene is densely covered with reference images, we evaluate here the case where only four reference images per indoor scene are given. Each of them is placed in one corner of the room, oriented towards the center. We evaluate this scenario on two datasets: One is based on replica [38], where we manually placed the four reference cameras in every room and sample 500 query poses the same way as for the sparse training set. Using a second dataset, we show that this also works on images recorded from a sensor of our robotic system. Therefore, an Intel Realsense d435 sensor is used to record multiple trajectories in our kitchen setup. The ground truth poses are determined by applying ORBSLAM2 [39] to the recorded trajectories. For comparison, ORBSLAM2, when evaluated on 7-Scenes, achieves an error of 0.06m / 2.55°, which makes it comparable to other 3D structure-based methods. Again, we extract four reference images this time from the recorded trajectories, one per corner facing towards the center. The results in table III show that image retrieval performs worst, which is reasonable due to the low number of reference images. Using Sift+5pt in the same way as with the 7-Scenes dataset leads to a big number of cases where no prediction is possible at all. Especially if the baseline between a given query reference pair is very large, the number of successful feature matches is too low to estimate the relative pose via the 5-point algorithm. By increasing Lowe's ratio, we can increase the number of feature matches we mark as correct, but at the same time also the accuracy of the predicted poses decreases due to the lower quality of feature matches. Compared to all tested classical and learning-based methods, our approach performs best. This shows that in the case of only a few reference images and thus large baselines between query and reference images, learning-based approaches can outperform classical non-learning based approaches in scene-agnostic pose estimation. As can be seen in Fig. 5b, this stays true if further additional reference images are given. Starting from 64 additional reference images the ExReNet trained on dense training data used in the last section can be used to achieve the best accuracy.



(a) 7-Scenes (b) Our recorded dataset

Fig. 5. Visualizes how the translational error changes depending on the number of reference images per scene. On 7-Scenes the reference images per scene are randomly reduced, while on our recorded dataset random reference images are added additional to the four ones used in table III. ExReNets are evaluated using scale and uncertainty information.

VI. CONCLUSIONS

In this paper we showed that relative pose estimation can be used to localize in new scenes without retraining or detailed 3D models. By extending the regression part and adding a guided correlation layer, we create a novel relative pose estimation architecture called ExReNet, which is able to generalize to new scenes. Additionally, we demonstrated how translation-scale and uncertainty information can be further used to improve the results. On sparse and dense covered datasets such as 7-Scenes our approach beats SIFT+5pt and outperforms all methods trained on unrelated data and most approaches directly trained on the evaluation scene, while our network is only trained on unrelated synthetic data.

References

[1] Z. Laskar, I. Melekhov, S. Kalia, and J. Kannala, "Camera relocalization by computing pairwise relative poses using convolutional neural network," *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 920–929, 2017.

[2] V. Balntas, S. Li, and V. Prisacariu, "Relocnet: Continuous metric learning relocalisation using neural nets," in *ECCV*, 2018.

[3] Q. Zhou, T. Sattler, M. Pollefeys, and L. Leal-Taixé, "To learn or not to learn: Visual localization from essential matrices," *ArXiv*, vol. abs/1908.01293, 2019.

[4] T. Sattler, A. Torii, J. Sivic, M. Pollefeys, H. Taira, M. Okutomi, and T. Pajdla, "Are large-scale 3d models really necessary for accurate visual localization?" in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1637–1646.

[5] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 place recognition by view synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1808–1817.

[6] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5297–5307.

[7] R. Arandjelović and A. Zisserman, "Dislocation: Scalable descriptor distinctiveness for location recognition," in *Asian Conference on Computer Vision*. Springer, 2014, pp. 188–204.

[8] W. Zhang and J. Kosecka, "Image based localization in urban environments," in *Third international symposium on 3D data processing, visualization, and transmission (3DPVT'06)*. IEEE, 2006, pp. 33–40.

[9] T. Cavallari, S. Golodetz, N. A. Lord, J. Valentin, L. Di Stefano, and P. H. Torr, "On-the-fly adaptation of regression forests for online camera relocalisation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4457–4466.

[10] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother, "Dsac-differentiable ransac for camera localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6684–6692.

[11] E. Brachmann and C. Rother, "Learning less is more-6d camera localization via 3d surface regression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4654–4662.

[12] X. Li, J. Ylioinas, J. Verbeek, and J. Kannala, "Scene coordinate regression with angle-based reprojection loss for camera relocalization," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.

[13] L. Meng, F. Tung, J. J. Little, J. Valentin, and C. W. de Silva, "Exploiting points and lines in regression forests for rgb-d camera relocalization," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 6827–6834.

[14] D. Massiceti, A. Krull, E. Brachmann, C. Rother, and P. H. Torr, "Random forests versus neural networkswhat's best for camera localization?" in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5118–5125.

[15] J. Valentin, M. Nießner, J. Shotton, A. Fitzgibbon, S. Izadi, and P. H. Torr, "Exploiting uncertainty in regression forests for accurate camera relocalization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4400–4408.

[16] A. Kendall, M. Grimes, and R. Cipolla, "Convolutional networks for real-time 6-dof camera relocalization," *CoRR*, vol. abs/1505.07427, 2015. [Online]. Available: http://arxiv.org/abs/1505.07427

[17] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[19] T. Naseer and W. Burgard, "Deep regression for monocular camera-based 6-dof global localization in outdoor environments," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1525–1530.

[20] A. Valada, N. Radwan, and W. Burgard, "Deep auxiliary learning for visual localization and odometry," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6939–6946, 2018.

[21] S. Saha, G. Varma, and C. V. Jawahar, "Improved visual relocalization by discovering anchor points," in *BMVC*, 2018.

[22] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu, "Image-based localization using hourglass networks," *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 870–877, 2017.

[23] F. Walch, C. Hazirbas, L. Leal-Taixé, T. Sattler, S. Hilsenbeck, and D. Cremers, "Image-based localization using lstms for structured feature correlation," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 627–637, 2016.

[24] T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixe, "Understanding the limitations of cnn-based absolute camera pose regression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3302–3312.

[25] A. Kendall and R. Cipolla, "Modelling uncertainty in deep learning for camera relocalization," *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4762–4769, 2015.

[26] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz, "Geometry-aware learning of maps for camera localization," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2616–2625, 2017.

[27] Q. Li, J. Zhu, R. Cao, K. Sun, J. M. Garibaldi, Q. Li, B. Liu, and G. Qiu, "Relative geometry-aware siamese neural network for 6dof camera relocalization," *ArXiv*, vol. abs/1901.01049, 2019.

[28] N. Radwan, A. Valada, and W. Burgard, "Vlocnet++: Deep multitask learning for semantic visual localization and odometry," *CoRR*, vol. abs/1804.08366, 2018. [Online]. Available: http://arxiv.org/abs/1804.08366

[29] M. Ding, Z. Wang, J. Sun, J. Shi, and P. Luo, "Camnet: Coarse-to-fine retrieval for camera re-localization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2871–2880.

[30] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[31] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6555–6564, 2017.

[32] Y. Gal, "Uncertainty in deep learning," 2016.

[33] R. I. Hartley and P. Sturm, "Triangulation," *Computer vision and image understanding*, vol. 68, no. 2, pp. 146–157, 1997.

[34] T. Sattler, B. Leibe, and L. Kobbelt, "Efficient effective prioritized matching for large-scale image-based localization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 9, pp. 1744–1756, 2017.

[35] M. Denninger, M. Sundermeyer, D. Winkelbauer, Y. Zidan, D. Olefir, M. Elbadrawy, A. Lodhi, and H. Katam, "Blenderproc," 2019.

[36] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1746–1754.

[37] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5828–5839.

[38] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, A. Clarkson, M. Yan, B. Budge, Y. Yan, X. Pan, J. Yon, Y. Zou, K. Leon, N. Carter, J. Briales, T. Gillingham, E. Mueggler, L. Pesqueira, M. Savva, D. Batra, H. M. Strasdat, R. D. Nardi, M. Goesele, S. Lovegrove, and R. Newcombe, "The Replica dataset: A digital replica of indoor spaces," *arXiv preprint arXiv:1906.05797*, 2019.

[39] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.