2020 International Conference on Unmanned Aircraft Systems (ICUAS)
Athens, Greece. September 1-4, 2020

# RCPNet: Deep-Learning based Relative Camera Pose Estimation for UAVs

Chenhao Yang[1], Yuyi Liu[2] and Andreas Zell[1]

*Abstract*— **In this paper, we propose a deep neural-network based regression approach, combined with a 3D structure based computer vision method, to solve the relative camera pose estimation problem for autonomous navigation of UAVs. Different from existing learning-based methods that train and test camera pose estimation in the same scene, our method succeeds in estimating relative camera poses across various urban scenes via a single trained model. We also built a *Tuebingen Buildings* database of RGB images collected by a drone in eight urban scenes. Over 10,000 images with corresponding 6DoF poses as well as 300,000 image pairs with their relative translational and rotational information are included in the dataset. We evaluate the accuracy of our method in the same scene and across scenes, using the *Cambridge Landmarks* dataset and the *Tuebingen Buildings* dataset. We compare the performance with existing learning-based pose regression methods PoseNet and RPNet on these two benchmark datasets.**

## I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) has rapidly developed in the past decade and is widely applied in robot navigation and augmented reality. Visual SLAM, as the mainstream of SLAM, estimates camera motion by extracting sparse keypoints (e.g., SIFT [1], SURF [2]) or directly from pixels as in dense SLAM systems [3], [4]. In many situations, the localization problem has been tackled using 3D geometry.

Visual SLAM plays a significant role in robot navigation and autonomous driving. However, for unmanned ground and aerial vehicles in urban scenes, classical visual SLAM system faces several challenges. First, a large viewpoint change caused by the high-speed motion of UAVs leads to a massive shift in appearance between keyframes [5], [6]. Second, the number of correspondences between keypoints decreases with texture-less objects such as smooth walls. Third, the feature matching becomes noisy with repetitive textures such as self-similar windows. Last but not least, these methods can estimate the translation vector only up to scale and therefore require a good initialization [7].

Recently, convolutional neural networks (CNNs) have been successfully applied in areas of object identification, image classification [8] and visual place recognition [9]. Structure from motion (SfM) [10] has shown tremendous progress in 3D reconstruction, obtaining centimeter-scale accuracy in localizing 3D points and camera poses. SfM methods automatically generate camera poses of images captured around a 3D model, and these 6-DOF poses can be used as training labels of a CNN regressor. The combination of CNN with SfM reduces human labor in building databases, makes deep-learning based camera pose regression approaches possible, and opens up a new direction for addressing the challenges faced by classical visual SLAM.

CNN-based absolute camera pose regression usually trains models to regress the camera pose of an input image to a fixed scene [11], [12], [13], which implicitly represents the 3D information of a scene via the weights of the network. This makes the system less general. In contrast, relative camera pose regression between two input images is a more general problem. An ideal relative camera pose regressor can not only be trained and tested within a limited distance, but also be spanned to different seen scenes in parallel, or even to unseen scenes.

In general, we may develop a robust and universal CNN-based camera pose estimation system following three steps: the first step is to train and test the model on images of the same scene while the second step is able to train a model on images of multiple scenes and to test the models on each of them, which includes our proposed work of across training in this paper. The third step is to train one model on images of many scenes and test it on unseen scenes. The reader needs to note that the second and third steps of such a system can be developed easier via using relative camera pose estimation than absolute camera pose regression in an end-to-end way. Absolute camera pose estimation across scenes is much more challenging since the 6DoF of camera pose in different scenes could be extremely unbalanced. In the meantime, relative camera pose estimation could be used in wide applications, such as in visual odometry systems for neighbor frame pose estimation, in multi-robot cooperation systems for robots relative pose prediction, or even combining with a global localization method like NetVLAD[9], to get more precise absolute pose estimation.

In this paper, we propose an end-to-end CNN-based approach to realize across-scene relative camera pose estimation (RCPNet) in urban outdoor environments. We also build a dataset (thereafter called *Tuebingen Buildings* database) via collecting more than 10,000 images with a drone in eight urban scenes and obtain the absolute pose of each image by the SfM method. For relative camera pose estimation, we generate over 300,000 pairs of images with SIFT feature matching, traversing every subset. The performance of our proposed method is compared with existing learning-based pose regression approaches PoseNet and RPNet, using the

[1]C. Yang and A. Zell are with the Chair of Cognitive Systems, Department of Computer Science, University of Tübingen, Tübingen, Germany
chenhao.yang@uni-tuebingen.de
[2]Y. Liu is with HRI Laboratory, Dept. of Social Informatics, Graduate School of Informatics, Kyoto University, Kyoto, Japan
yuyi.liu@robot.soc.i.kyoto-u.ac.jp

978-1-7281-4277-7/20/$31.00 ©2020 IEEE


Authorized licensed use limited to: Northrop Grumman Mission Systems. Downloaded on January 08,2021 at 23:10:56 UTC from IEEE Xplore. Restrictions apply.

same two databases, namely, *Cambridge Landmarks* and *Tuebingen Buildings* datasets.

The rest of the paper is organized as follows: Sec. II presents the related work. Sec. III discusses the model for relative camera pose and network architecture. Sec. IV introduces the data collection and preparation. Sec. V compares the experimental results of the proposed method with two baselines. In the end, Sec. VI concludes the paper.

## II. Related work

The localization of visual SLAM mainly includes three tasks [12], [14]: i) Image matching to recognize viewed places in loop closure, ii) Relative pose estimation between recognized images in back-end optimization to eliminate the drift of localization, iii) Relative pose estimation between continuous keyframes for visual odometry. We categorize the first one as topological localization and the last two as metric localization.

**Topological Localization:** Given a query image and a set of images with known locations, the closest or most similar image will be retrieved by leveraging different feature matching methods. These methods have succeeded in coarsely relocalizing the camera to a known place from Google Street View [15], satellite view [16], or aerial images [17].

Suenderhauf et al. [18] proposed a system that utilizes convolutional network features as robust landmark descriptors to recognize places despite severe viewpoint and condition changes. Workman et al. [17] and Vo et al. [19] collected two large datasets (CVUSA, Vo and Hays, respectively) of aerial-ground images for cross-view geo-localization. Based on these two datasets, Hu et al. [20] proposed CVM-Net, combining NetVLAD [9] layers with a Siamese network to learn robust representations for cross-view image matching.

Majdik et al. [14] achieved localization for Unmanned Aerial Vehicles (UAVs) in GPS-denied urban environments from Google Street View data. They collected a dataset in downtown Zurich, Switzerland, with a drone flying along a 2 km trajectory. The dataset includes 405 air images matching 113 discrete street view locations. Recall rate at precision 1 of this work is more than 45%. Topological localization methods focus on providing a query image with a label of limited and discretized position information, rather than providing continuous and accurate pose estimation.

**Metric Localization:** Continuous and accurate localization of new images in a map or a known environment, which is the final goal for autonomous robot application. Classical visual SLAM methods [21], [22] mainly focus on creating a sparse or dense map of the environment using point-based features. Pascoe et al. [23] made use of a map built by combining data from LIDAR and cameras to localize live camera images.

PoseNet [11] pioneered in 6-DOF camera relocalization by using a SfM method to label images' poses and an end-to-end CNN to train and predict absolute camera poses. They replaced all three softmax classifiers of GoogLeNet [24] with affine regressors to output translations and rotations and proposed the *Cambridge Landmarks* dataset. RelocNet [25] proposes a method of learning suitable convolutional representations for camera pose retrieval based on nearest neighbor matching and continuous metric learning-based feature descriptors. Naseer and Burgard [12] generate synthetic depth maps and viewpoints as data augmentation to learn a more discriminative regression function. Walch et al. [13] proposed a CNN+LSTM (Long Short Term Memory) architecture for camera pose regression by modelling the context of images. They also proposed the *TMU-LSI* dataset and showed that classical approaches completely fail under texture-less conditions.

Instead of regressing the absolute camera pose, Melekhov et al. [26] proposed the first end-to-end system aiming at solving the relative camera pose estimation based on a pre-trained hybrid neural network with fully connected layers as pose regressor. However, they estimate the translations only up to scale rather than as full vectors, making comparisons difficult. To the best of our knowledge, all the existing camera pose regression approaches (both absolute and relative) are trained and tested in a similar scene, which weakens the generalization ability. Sattler et al. showed that absolute pose regression in a fixed scene is more closely related to pose-approximation via image retrieval than to accurate pose estimation via 3D structure in [27] and regarded absolute camera pose regression as topological localization rather than metric localization. The authors of [28] chose to solve the localization problem by combining 2D model based method with local SfM reconstruction since they considered it still a significant challenge to construct large-scale 3D models for metric localization. However, our perspective is that a single trained model can be used to estimate the relative camera pose in different scenes, while the 3D reconstruction is implemented only in the training stage.

Besides visual SLAM, our proposed end-to-end relative camera pose estimation system could also be used by visual odometry systems like VINS [29], [30], VINet [31] and DeepVO [32] to estimate pose changing between continuous keyframes. It could also work in other situations, for example, directly obtaining the relative pose between camera-mounted robots in a cooperating system, in a centralized or decentralized way, for a group of ground or/and aerial robots.

## III. Model for deep regression of relative pose

In this section, we discuss the model of Relative Camera Pose and the proposed regression convolutional neural network (convnet) architecture (RCPNet). With two images input together, RCPNet outputs a relative pose vector $p$, given by a 3D relative camera translation $t$ and rotation represented by quaternion $q$:

$$p = [t, q]. \tag{1}$$

Quaternions are selected as our rotation representation since arbitrary 4-D values are easily mapped to a legitimate rotation by normalizing them to unit length.



Authorized licensed use limited to: Northrop Grumman Mission Systems. Downloaded on January 08,2021 at 23:10:56 UTC from IEEE Xplore. Restrictions apply.

### A. Simultaneously learning relative translation and rotation

We calculate the relative pose between two images before training. Set ($R_1$, $t_1$), ($R_2$, $t_2$) as the rotation matrices and translation vectors projecting a point from world coordinate separately to camera 1 and 2's systems. $P_{12}$ is the transformation matrix, $R_{12}$ is the rotation matrix, and $t_{12}$ is the translation vector from camera system 1 to 2:

$$P_{12} = \begin{bmatrix} R_{12} & t_{12} \\ 0 & 1 \end{bmatrix}; \quad \begin{cases} R_{12} = R_2 R_1^T, \\ t_{12} = R_1(t_2 - t_1). \end{cases} \tag{2}$$

Take ($q_1$, $q_2$, $q_{12}$) as quaternion representations of ($R_1$, $R_2$, $R_{12}$). We make a numerical inversion for all the $q_{12}$ with negative $w$ value since the unit quaternions $q$ and $-q$ denote the same rotation.

Decoupling translation and rotation regressors each denies the necessity to factor out rotation from translation, and vice versa [11]. Therefore, the ConvNet was initially trained by optimizing the following objective function which minimizes the Euclidean loss between translation and rotation estimate predictions ($\hat{t}$ and $\hat{q}$) and the ground truth ($t$ and $q$), using stochastic gradient descent:

$$\mathcal{L}(I) = ||\hat{t} - t||_2 + \beta||\hat{q} - q||_2. \tag{3}$$

To learn translation and rotation simultaneously, they fine-tuned the weighting factor $\beta$ for the importance of balance between translation and rotation error using grid search. They found that the scale factor $\beta$ has a significant change interval between 250 to 2000 for outdoor scenes. RPNet [7] uses cross-validation to find the most suitable values of hyper-parameter $\beta$ in different scenes, which takes a lot of time to group the original dataset and test the trained models as a performance indicator for evaluating the regressor. For our proposed approach, we implement homoscedastic uncertainty based automatic weight that scales on loss function (as in [33]) across all the scenes, with a more numerically stable effect on $\beta$:

$$\mathcal{L}_\sigma(I) = \mathcal{L}_t(I)exp(-\hat{s_t}) + \hat{s_t} + \mathcal{L}_q(I)exp(-\hat{s_q}) + \hat{s_q}, \tag{4}$$

where $\mathcal{L}_t$ denotes the translation loss and $\mathcal{L}_q$ denotes the rotation loss. $\hat{s_t}$ and $\hat{s_q}$ are the factors to balance the penalty value between translation loss and rotation loss, ensuring that the regression error of the network for rotation and translation is not skewed. Since $exp(s_i)$ is resolved to the positive domain giving valid values for variance, the exponential mapping allows us to regress unconstrained scalar values. Initial values of $\hat{s_t} = 0.0$, $\hat{s_q} = -3.5$ (equivalent to $\beta$ starting from 30 but fine-tuned during training process) are used for all scenes, as this loss is very robust. This novel loss makes the result more general to suit different scenes, even with one single model.

### B. Architecture

Different from PoseNet [11] and RPNet [7] based on GoogLeNet, we build a weights-shared Siamese Network [34] with two branches of pre-trained ResNet34 networks [35], regressing the 6-DOF relative camera pose from a pair of monocular RGB images in an end-to-end manner.

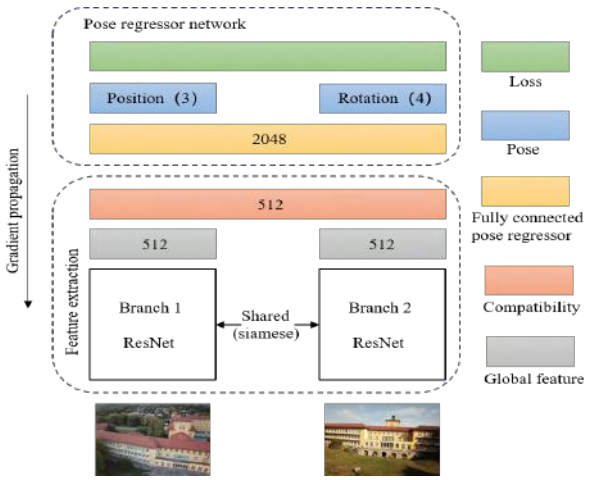



Fig. 1. RCPNet: our proposed deep regression architecture based on ResNet34 with final regressors that estimate relative translation and rotation between two input images.

As illustrated in Fig. 1, the relative pose regressor of RCPNet is based on fully connected layers (FCs) with *ReLU* [36] activations. For each ResNet34 branch, we extract the 512-Dimension (512-D) output of the second to last layer as a global feature of every input image. Following [34], we measure the compatibility between images $I_1$ and $I_2$ with:

$$E_w(I_2, I_1) = ||G_w(I_2) - G_w(I_1)||, \tag{5}$$

where $G_w(I_i)$ denotes the global feature of each image. Two 2048-D FCs are inserted afterward as regressors to output a 7-D vector of relative translation (3-D) and rotation (4-D), respectively. However, they are simultaneously learned from the loss function of equation (4). The quaternion rotation vector is normalized to unit length.

## IV. Data Collection and Preparation

In this section, we describe how we set up a new dataset for usage of outdoor urban localization. The dataset of over 10,000 images and corresponding poses was collected from SfM. We further extended the dataset to a real 3D world with vertical viewpoint changes by capturing images with a UAV. The dataset makes the training of a pose regressor more efficient by offering more image pairs with a more extensive range of translation and rotation.

### A. Data collection

We built an outdoor urban localization dataset, *Tuebingen Buildings*, with eight scenes. This dataset provides data to train and test both absolute and relative pose regression algorithms within or across diverse urban environments. Our dataset was collected in several places near Tuebingen, Germany, with a DJI Mavic Pro drone.

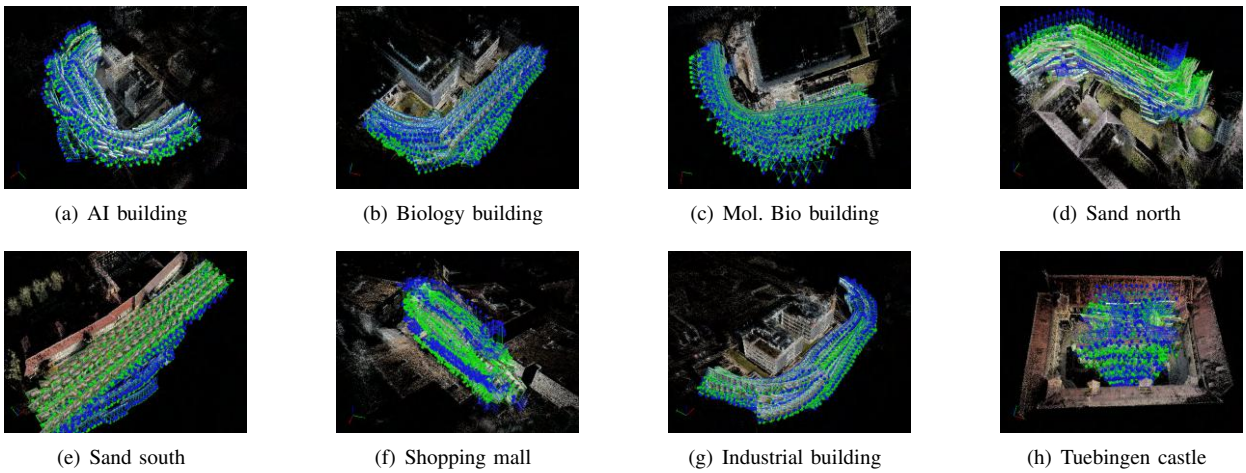
The drone was manually piloted around every scene, collecting images throughout the environment at different flying altitudes from 2 to 35 m by keeping the camera always facing the buildings. For each scene, at least four flights have been made at different times, to collect images under variant



Authorized licensed use limited to: Northrop Grumman Mission Systems. Downloaded on January 08,2021 at 23:10:56 UTC from IEEE Xplore. Restrictions apply.

TABLE I. Mean uncertainties of absolute camera position and orientation

| Mean Uncertainties | X | Y | Z | Yaw | Pitch | Roll | Images |
|---|---|---|---|---|---|---|---|
| AI Building | 0.136m | 0.142m | 0.223m | 0.549° | 0.089° | 0.444° | 1438 |
| Biology Building | 0.248m | 0.246m | 0.402m | 0.377° | 0.160° | 0.266° | 1209 |
| Mol. Bio Building | 0.120m | 0.125m | 0.201m | 0.236° | 0.070° | 0.146° | 1112 |
| Sand North | 0.147m | 0.140m | 0.227m | 0.381° | 0.163° | 0.326° | 1504 |
| Sand South | 0.152m | 0.138m | 0.228m | 0.073° | 0.084° | 0.026° | 1035 |
| Shopping Mall | 0.124m | 0.126m | 0.205m | 0.256° | 0.183° | 0.191° | 1537 |
| Industrial Building | 0.403m | 0.356m | 0.603m | 0.302° | 0.236° | 0.157° | 1302 |
| Tuebingen Castle | 0.088m | 0.088m | 0.152m | 0.172° | 0.092° | 0.083° | 1216 |



(a) AI building (b) Biology building (c) Mol. Bio building (d) Sand north

(e) Sand south (f) Shopping mall (g) Industrial building (h) Tuebingen castle

Fig. 2. Camera poses uniformly cover 3D models from different viewpoints and depth. Trajectories have different types of convex and concave shape. Original camera poses offered by UAV are blue, camera poses calibrated by SfM are in green color.

lighting and weather conditions. Although urban clutter such as pedestrians and vehicles were present, they had a minor influence for most images captured from viewpoints higher than 5 m.

The poses, as training labels and ground truth measurements, were generated using the SfM software Pix4D Mapper [37]. The output uncertainties of absolute camera position and orientation can be seen in Table I. Vertical viewpoint variance of images brought new 3D constraints, leading to good performance on localization: position error around 10 to 40 cm and orientation error below 1°. Instead of recording videos and sub-sampling them into frames, the drone was programmed to directly capture images every 2 m movement in any direction (distance determined by GPS). Images were collected with high resolution (4000×3000) for better 3D reconstruction and more uniform nets of image-poses to cover 3D models from different depth (see Fig. 2).

The eight scenes (see example images in Fig. 3) in the dataset are diverse for three reasons: i) consisting of both, modern buildings and classical buildings, ii) some buildings have repetitive concrete structure while others are nestled in the natural environment, e.g., surrounded by big trees, iii) trajectories of scenes are different. For instance, there are convex trajectories around some center buildings or concave trajectories in a yard surrounded by several buildings.

We also use the *Cambridge Landmarks* dataset with four scenes (*King's College*, *Old Hospital*, *Shop Facade*, *St Mary's Church*). This dataset contains images from ground viewpoints by a hand-held Google LG Nexus 5 smartphone. As a comparison, our newly built dataset contains large viewpoint changes, especially in the vertical direction, which is very common in UAV applications.

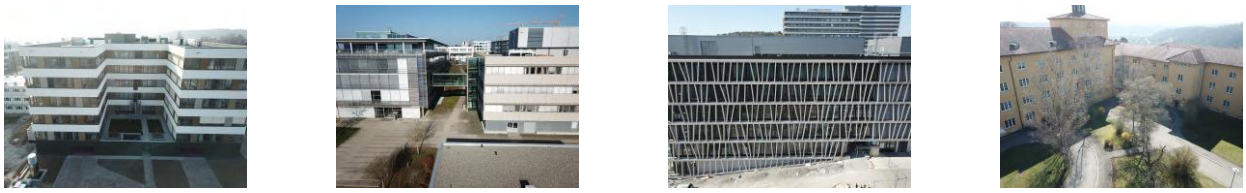
### B. Data preparation

The position data we obtained from 3D reconstruction is based on the World Geodetic System 1984 (WGS84), the longitude and latitude reference points in this system are much farther than sea level. As a result, the camera position is quite skewed (X:Y:Z ≈ 1000:10000:1). Our solution is to subtract the average values of X, Y, and Z of each camera position in every scene separately.

For relative camera pose regression, we need an efficient way to generate matching pairs. In [7], En et al. randomly paired each image with eight different images in the same sequence of *Cambridge Landmarks* dataset. We followed their setting when using this dataset for relative pose regression. For scenes in the *Tuebingen Buildings* dataset, we generate pairs by SIFT feature matching, traversing the whole test or train set for every image inside. Image pairs that had large differences of translation (more than 30 meters) or rotation (more than 75 degrees) were checked manually. At last, we obtained around 300,000 valid pairs in all eight



Authorized licensed use limited to: Northrop Grumman Mission Systems. Downloaded on January 08,2021 at 23:10:56 UTC from IEEE Xplore. Restrictions apply.



(a) AI building (b) Biology building (c) Mol. Bio building (d) Sand north


(e) Sand south (f) Shopping mall (g) Industrial building (h) Tuebingen castle

Fig. 3. Tuebingen buildings dataset, consists of 8 different scenes include modern and classical, repetitive concrete structure and relative more natural appearance of the urban environment. Collected by manually piloted UAV with flying altitudes from 2 to 35 m.

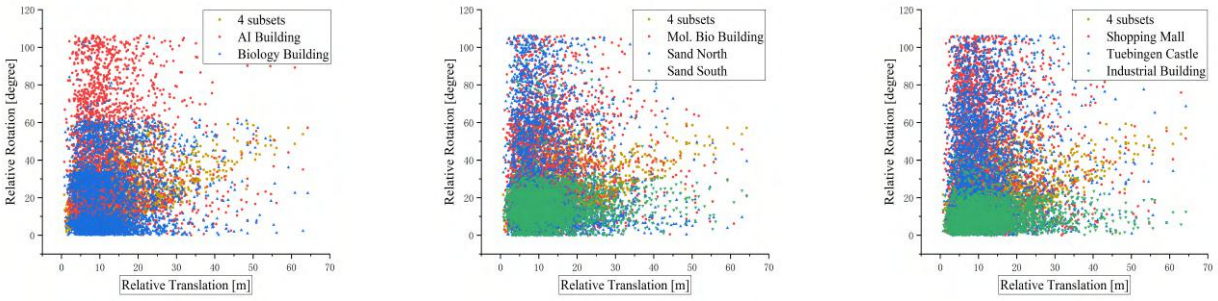



Fig. 4. Plots of relative camera pose distribution on translation and rotation, eight scenes in *Tuebingen buildings*, separately compare with four subsets together (*King's College*, *Old Hospital*, *Shop Facade*, and *St Mary's Church*) in *Cambridge Landmarks*. The proposed *Tuebingen buildings* dataset has advantages on both data density and diversity.

scenes, 30 pairs per image on average. This extension is not surprising since our dataset covers real 3D spaces, and any image inside has many overlapping neighbors in different directions. Fig. 4 shows that the relative camera pose samples in our proposed dataset spread broader in both ranges of rotation and translation.

In previous work [11], [7], the images were rescaled so that the smallest dimension was 256 pixels before being cropped to $224 \times 224$ patches as input to the convnet. Convnets was trained on random crops and afterward tested by center crops, as is common for data augmentation in visual tasks such as image classification. However, the crop operation (just like flip, shift, zoom, and rotate) likely affects the spatial information inside an image. For our data, we made the random cropping consistent in two input images, and then tested several subsets with rescaling size from 256 to 236. The results show that for subsets with shorter object distance like *Shop Facade*, smaller rescaling size makes a larger field of view (also a more extensive overlapping) and brought a slight improvement in localization. While for most subsets with a longer distance to objects, the effect of data augmenting with larger rescaling size prevailed. This indicates that we may need other data augmentation methods in the future (brightness, contrast, saturation, and hue tuning are already adopted).

## V. Experiments

### A. Experimental setup

We first evaluated our approach on four subsets of *Cambridge Landmarks* individually. In addition, three of them (*King's College*, *Old Hospital*, *St Mary's Church*) were selected to train a single model of RCPNet to be tested across scenes, keeping *Shop Facade* unseen. We used the same test and train split as RPNet [7]. Similar to the *Tuebingen Buildings* dataset, we first trained and tested a model in every subset, as individual training, and then trained a single model with six subsets (*AI Building*, *Biology Building*, *Mol. Bio Building*, *Sand North*, *Shopping Mall*, and *Tuebingen Castle*) fed in together and finally tested the trained single model separately in every subset, as across training.

For individual training, we randomly split data in each scene to test and train set according to the ratio of 1 to 4. For across training, we kept the test set of each scene identical, extracted 20,000 pair samples from every scene's training set, obtaining a set of 120k training data. In this setting, some spatially close images may show up in the test and the train subsets at the same time, but the images are distinct, and the

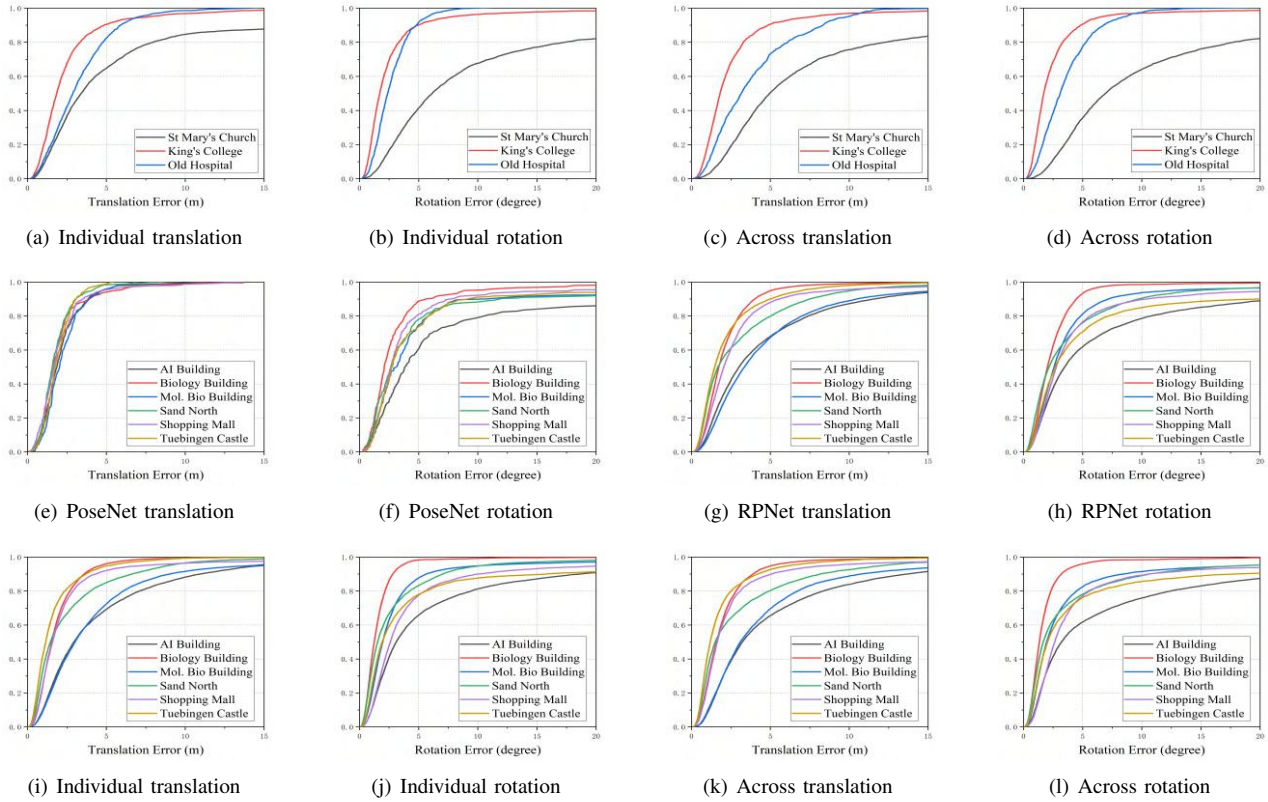


Authorized licensed use limited to: Northrop Grumman Mission Systems. Downloaded on January 08,2021 at 23:10:56 UTC from IEEE Xplore. Restrictions apply.

TABLE II. Dataset details and results: median absolute/relative localization errors for the *Cambridge Landmarks* and *Tuebingen Buildings* datasets.

| Scene | Frames Test | Frames Train | Pairs Test | Pairs Train | Spatial Extent(m) | (Absolute) PoseNet [11] | (Relative) RPNet [7] | (Individual) RCPNet | (Across) RCPNet |
|---|---|---|---|---|---|---|---|---|---|
| King's C. | 343 | 1220 | 2424 | 9227 | 140×40 | 1.92m,5.40° | 1.93m,3.12° | 1.85m,**1.72**° | **1.80m,1.72**° |
| Old Hospital | 182 | 895 | 1228 | 6417 | 50×40 | **2.31m**,5.38° | 2.41m,4.81° | 2.87m,**2.41**° | 3.15m,3.09° |
| St Mary's C. | 530 | 1487 | 3944 | 10736 | 80×60 | 2.65m,8.48° | **2.29m,5.90**° | 3.43m,6.14° | 4.84m,6.93° |
| Shop Facade* | 103 | 231 | 607 | 1643 | 35×25 | **1.46m**,8.08° | 1.68m,**7.07**° | 1.63m,7.36° | 13.8m,28.6° |
| AI Building | 288 | 1150 | 9326 | 38549 | 145×90×28 | **1.87m**,3.84° | 3.01m,3.47° | 2.94m,**3.10**° | 3.22m,3.21° |
| Biology B. | 242 | 967 | 8589 | 34421 | 120×95×26 | 1.58m,2.12° | 1.73m,2.02° | **1.53m,1.24**° | 1.58m,1.32° |
| Mol. Bio B. | 223 | 889 | 10492 | 41528 | 190×95×25 | **2.03m**,3.15° | 3.36m,2.59° | 3.02m,**1.95**° | 3.09m,**1.95**° |
| Sand North | 301 | 1203 | 6680 | 27315 | 100×45×23 | 1.57m,2.65° | 1.67m,2.15° | **1.45m,1.52**° | 1.50m,1.66° |
| Shopping M. | 308 | 1229 | 4991 | 20412 | 50×55×13 | 1.66m,2.75° | 2.05m,2.77° | **1.58m**,2.66° | 1.63m,**2.64**° |
| Tue. Castle | 244 | 975 | 5787 | 23182 | 40×35×21 | 1.64m,2.80° | 1.47m,2.69° | **1.16m,1.92**° | 1.19m,2.01° |
| Sand South* | 207 | 828 | 6276 | 25729 | 190×50×37 | 2.06m,2.27° | 1.52m,6.64° | **1.17m,2.64**° | 17.6m,21.3° |
| Industrial B.* | 261 | 1041 | 9910 | 40329 | 170×60×33 | 1.75m,2.69° | 1.34m,1.68° | **1.12m,1.37**° | 16.7m,18.4° |

* these scenes are the unseen subsets in across training.




(a) Individual translation (b) Individual rotation (c) Across translation (d) Across rotation

(e) PoseNet translation (f) PoseNet rotation (g) RPNet translation (h) RPNet rotation

(i) Individual translation (j) Individual rotation (k) Across translation (l) Across rotation

Fig. 5. Localization performance. These cumulative histograms of errors for scenes in *Cambridge Landmarks* (1st row) and *Tuebingen Buildings* (2nd and 3rd row) show localization accuracy of four approaches: PoseNet, RPNet, the individual trained RCPNet, and RCPNet with across training. PoseNet has better performance in *Mol. Bio building* and *AI Building* subsets, while our proposed individual training leads in other subsets. The across trained RCPNet keeps up with the individual training in performance in most scenes, proving that the same features exist between image pairs from different scenes.

similarities between image pairs are very weak since they have been matched from multiple directions.

We now discuss the quantitative evaluation of different approaches with different scenes. For comparison, we also tested absolute camera pose regression with PosNet and relative camera pose estimation with RPNet on all eight scenes in *Tuebingen Buildings*. We directly cite the results in their papers on the *Cambridge Landmarks* dataset. We measured translation errors in meters and rotation errors in degrees.

We normalized the range of input image's pixel intensities from $-1$ to 1. We trained RCPNet using an implementation in PyTorch [38]. All models are optimized with ADAM [39] using the default parameters and a learning rate of $1 \times 10^{-4}$. We trained each model until convergence of loss. We used a batch size of 32 for individual training on an NVIDIA 1080 Ti GPU; training took 20k - 100k iterations, i.e. 10 hours - 2 days. For across training, we used a batch size of 128 on



Authorized licensed use limited to: Northrop Grumman Mission Systems. Downloaded on January 08,2021 at 23:10:56 UTC from IEEE Xplore. Restrictions apply.

two NVIDIA 1080 Ti GPUs, and training took 2 days.

### B. Experimental results

We compare the performances between different network architectures on each scene (including dataset details), individually or across scenes, seen or unseen, in Table II. The baselines are PoseNet trained by frames and RPNet trained by pairs of images, both within one scene. In general, the absolute pose regression accuracy is not directly comparable with the relative pose estimation. However, since PoseNet [11] is the basis for both RPNet [7] and our proposed RCPNet, we think it is a good reference to be compared with. The result shows that RCPNet outperformed PoseNet and RPNet in most subsets by individual training. While across scenes training brought a little drop in performance of seen scenes in both two datasets, it is still competitive and consistent. This result indicates that different scenes have many common features and one fine-tuned single model can fit them at the same time. For unseen subsets (*Shop Facade*, *Sand South* and *Industrial building*), the single model encountered a great challenge, and we leave this as an open question for better architectures or data augmentation algorithms.

As we mentioned in Sec. I, we upgraded the CNN-based camera pose estimation system from the first step to the second step, which is meaningful. When a robot works in a large factory with many workshops, or a UAV delivers goods between several GPS-denied locations in an urban environment, they need to train and store many models for different locations, if they only have localization ability of the first step. Every time when the robots come to one location, they need to first find the right model to use. With our method, the robot can work well with only one model. Therefore, a system of the second step is more practical and general in real robot applications.

The cumulative histograms of Fig. 5 exhibit the absolute/relative camera translation and rotation errors in different subsets of *Cambridge Landmarks* and *Tuebingen Buildings* by PoseNet, RPNet, and RCPNet with individual or across scenes training. RCPNet with individual training does perform better in rotations, while for *Mol. Bio building*, *AI Building*, and *St Marys Church*, PoseNet or RPNet leads in translation. Comparing to individually trained models within each scene, the performance of an across-trained model only drops a little.

## VI. Conclusion and Discussion

We present RCPNet, a CNN-based application of relative camera pose estimation across different scenes. RCPNet could be used in multi-robot cooperation applications, visual odometry systems, or combining with global localization methods such as NetVLAD, to get more precise absolute pose estimation. A novel camera pose regression dataset for both absolute and relative camera pose estimation was collected with the SfM method. Our convnet is based on a Siamese architecture with 2 ResNet34 branches. We have demonstrated that such a network outperforms two baseline methods in two datasets. The fact that across training has very similar performance to individual training within one scene indicates that common features of image pairs from different scenes exist. In future work, we aim to improve our proposed system with respect to network architecture and data augmentation such as using synthetic images from 3D models. In this way, it will lead to a more reliable camera relocalization outcome even in unseen environments.

## ACKNOWLEDGMENT

We would like to thank Maximus Mutschler, Benjamin Kiefer, Yanjun Cao, Jonas Tebbe, Yapeng Gao and Sujit Rajappa for valuable discussions and feedback.

## References

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov 2004.

[2] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds., Berlin, Heidelberg, 2006, pp. 404–417.

[3] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *2011 International Conference on Computer Vision*, Nov 2011, pp. 2320–2327.

[4] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 834–849.

[5] Y. Tian, C. Chen, and M. Shah, "Cross-view image matching for geo-localization in urban environments," *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, vol. 2017-Janua, pp. 1998–2006, 2017.

[6] T. Y. Lin, Y. Cui, S. Belongie, and J. Hays, "Learning deep representations for ground-to-aerial geolocalization," *2015 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, vol. 07-12-June, pp. 5007–5015, 2015.

[7] S. En, A. Lechervy, and F. Jurie, "Rpnet: An end-to-end network for relative camera pose estimation," in *Computer Vision – ECCV 2018 Workshops*, L. Leal-Taixé and S. Roth, Eds., Cham, 2019, pp. 738–745.

[8] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, June 2009, pp. 248–255.

[9] R. Arandjelovi, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1437–1451, June 2018.

[10] S. Agarwal, N. Snavely, I. Simon, S. M. Sietz, and R. Szeliski, "Building rome in a day," in *Twelfth IEEE International Conference on Computer Vision (ICCV 2009)*, September 2009.

[11] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Re," *2015 IEEE International Conference on Computer Visio, ICCV*, vol. 284, no. 15, pp. 1980–1983, 2015.

[12] T. Naseer and W. Burgard, "Deep regression for monocular camera-based 6-DoF global localization in outdoor environments," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2017-Septe, pp. 1525–1530, 2017.

[13] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers, "Image-Based Localization Using LSTMs for Structured Feature Correlation," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-Octob, pp. 627–637, 2017.

[14] A. Majdik, D. Verda, Y. Albers-Schoenberg, and D. Scaramuzza, "Air-ground matching: Appearance-based gps-denied urban localization of micro aerial vehicles," *J. Field Robotics*, vol. 32, pp. 1015–1039, 2015.

[15] D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. Ogale, L. Vincent, and J. Weaver, "Google street view: Capturing the world at street level," *Computer*, vol. 43, no. 6, pp. 32–38, June 2010.



Authorized licensed use limited to: Northrop Grumman Mission Systems. Downloaded on January 08,2021 at 23:10:56 UTC from IEEE Xplore. Restrictions apply.

[16] A. Viswanathan, B. R. Pires, and D. Huber, "Vision based robot localization by ground to satellite matching in gps-denied situations," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2014, pp. 192–198.

[17] S. Workman, R. Souvenir, and N. Jacobs, "Wide-area image geolocalization with aerial reference imagery," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 Inter, no. 2, pp. 3961–3969, 2015.

[18] N. Suenderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford, "Place Recognition with ConvNet Landmarks: Viewpoint-Robust, Condition-Robust, Training-Free," *Robotics: Science and Systems XI*, 2015.

[19] N. N. Vo and J. Hays, "Localizing and orienting street views using overhead imagery," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9905 LNCS, pp. 494–509, 2016.

[20] S. Hu, M. Feng, R. M. H. Nguyen, and G. H. Lee, "CVM-Net: Cross-View Matching Network for Image-Based Ground-to-Aerial Geo-Localization," *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 7258–7267, 2018.

[21] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, Nov 2007, pp. 225–234.

[22] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "isam2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 3281–3288.

[23] G. Pascoe, W. Maddern, and P. Newman, "Direct visual localisation and calibration for road vehicles in changing city environments," in *Proceedings of the 2015 IEEE International Conference on Computer Vision Workshop*, ser. ICCVW '15, 2015, pp. 98–105.

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CoRR*, vol. abs/1409.4842, 2014.

[25] V. Balntas, S. Li, and V. Prisacariu, "Relocnet: Continuous metric learning relocalisation using neural nets," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11218 LNCS, pp. 782–799, 2018.

[26] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu, "Relative camera pose estimation using convolutional neural networks," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10617 LNCS, pp. 675–687, 2017.

[27] T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixe, "Understanding the Limitations of CNN-based Absolute Camera Pose Regression," *2019 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2019.

[28] 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 ?" *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 1637–1646, 2017.

[29] T. Qin, S. Cao, J. Pan, and S. Shen, "A general optimization-based framework for global pose estimation with multiple sensors," 2019.

[30] T. Qin, J. Pan, S. Cao, and S. Shen, "A general optimization-based framework for local odometry estimation with multiple sensors," 2019.

[31] R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni, "Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem," 2017.

[32] V. Mohanty, S. Agrawal, S. Datta, A. Ghosh, V. D. Sharma, and D. Chakravarty, "Deepvo: A deep learning approach for monocular visual odometry," *ArXiv*, vol. abs/1611.06069, 2016.

[33] 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*, vol. 2017-Janua, pp. 6555–6564, 2017.

[34] S. Chopra, R. Hadsell, and Y. Lecun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. of Computer Vision and Pattern Recognition Conference, CVPR*. IEEE Press, 2005, pp. 539–546.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[36] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *2010 International Conference on Machine Learning, ICML*, 2010.

[37] C. Strecha, "Pix4dmapper: The leading photogrammetry software for professional drone mapping," Website, accessed June 2019. https://www.pix4d.com/product/pix4dmapper-photogrammetry-software.

[38] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.



Authorized licensed use limited to: Northrop Grumman Mission Systems. Downloaded on January 08,2021 at 23:10:56 UTC from IEEE Xplore. Restrictions apply.