Synthetic View Generation for Absolute Pose Regression and Image Synthesis

Pulak Purkait[1]
pulak.cv@gmail.com
Cheng Zhao[2]
irobotcheng@gmail.com
Christopher Zach[1]
christopher.m.zach@gmail.com

[1] Toshiba Research Europe Ltd.
Cambridge, UK

[2] University of Birmingham
Birmingham, UK

Abstract

Image based localization is one of the important problems in computer vision due to its wide applicability in robotics, augmented reality, and autonomous systems. There is a rich set of methods described in the literature on how to geometrically register a 2D image w.r.t. a 3D model. In particular, data augmentation methods such as synthetic image generation have been shown to be useful for this task. In this work, we propose a synthetic data augmentation technique and design a deep neural network, that can be trained to estimate the absolute pose of an image from synthesized sparse feature descriptors. Our choice of using sparse feature descriptors has two major advantages: first, our network is significantly smaller than the CNNs proposed in the literature for this task—thereby making our approach more efficient and scalable. Second—and more importantly—, usage of sparse features allows to augment the training data with synthetic viewpoints, which leads to substantial improvements in the generalization performance to unseen poses. The synthetic views are further employed to augment realistic RGB images which again surpasses recent deep learning based synthetic image generation technique. A detailed analysis of the proposed networks and a rigorous evaluation on the existing datasets are provided to support our method.

1 Introduction

In recent years deep learning has become the method of choice to address many computer vision tasks. Despite many successful applications deep learning methods still require huge amounts of training data, and they can have very limited generalization ability when faced with small training sets (e.g. [29]). Leveraging synthetic datasets for training can improve the accuracy of deep learning methods, when limited real training data is available [5, 6]. In this work we utilize structure-from-motion (SfM) to generate synthetic data that improves the performance of the deep networks employed for pose regression and RGB image synthesis.

Traditionally, absolute pose estimation has been tackled either by direct 2D-3D matching (e.g. [14, 18]) or by inserting an image retrieval stage to narrow down the search space (e.g. [8, 19, 30]). Synthetic views have been useful in the latter cases [8, 25]. These synthetic poses may cover regions in pose space not available in the training data and boost the

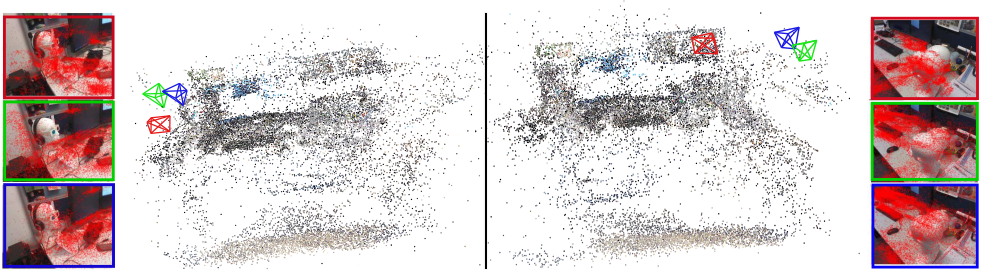


Figure 1: Two examples of 6DOF pose estimation results on heads sequence of the 7-scenes dataset [21] where PoseNet [11] fails to predict accurate pose (marked by red, positional error $= 0.31$m and angular error $= 27.4°$) whereas the proposed SPP-Net predicts a pose (marked by green, positional error $= 0.06$m and angular error $= 2.18°$) closer to the ground truth (marked by blue).

localization performance. PoseNet [11] and subsequent approaches [10, 26, 27] demonstrate that deep learning methods—which have shown excellent performance in numerous classification and regression problems—can estimate camera poses directly from input images. Despite the good performance of these methods and related architectures for image-based pose regression [5, 10, 11, 26], we believe that PoseNet-like methods are fundamentally limited in the following ways:

1. Forward regression architectures such as CNNs have no built-in reasoning about geometry and most likely do not extract an "understanding" of the underlying geometric concepts (such as the pinhole camera model) during the training phase. Consequently, we postulate (and empirically validate) that PoseNet-like approaches suffer from poor extrapolation ability to unseen poses significantly different from the ones in the training set. In many application settings the distributions of training poses and test poses can differ substantially: training images might be chosen such that structure-from-motion computation to obtain a 3D model is made easier, whereas test poses may be arbitrarily distributed within the maneuverable space. Hence, direct pose regression typically faces a domain adaptation problem in general. For instance, Li *et al.* [13] address this problem by augmenting the training set with synthetically warped RGB images using depth maps (but depth images may not be always available, and image synthesis is limited to nearby poses).
2. A lot of computation (and trainable parameters) in PoseNet-like architectures goes into the feature extraction stage, which is based on rather heavy-weight CNNs such as VGG-Net [22] or GoogleNet [23]. In light of empirical evidence supporting gold-standard feature descriptors (such as SIFT [15]), we conjecture that heavy-weight dense feature extraction via CNNs is not necessary for this task. The networks used in our approach are significantly smaller and faster to train than existing CNN-based solutions for pose regression. Using sparse features will also be beneficial for the domain adaptation problem.

The goal in this work is to generate realistic synthetic "images" in order to improve the performance of pose regression networks for unseen poses. In Fig. 1 we depict two examples where our proposed method predicts significantly superior poses than PoseNet [11]. We take a different route than e.g. DSAC [2, 3], which mimics a RANSAC approach within a differentiable architecture. It produces very competitive results, but requires about 0.2s per image on a high-end GPU (whereas our approach runs at up to 100 Hz).

The general advantages of using deep learning for pose regression over the direct 2D-3D matching (e.g. [14, 18]) are the benefits of end-to-end training, the reduced memory requirements (e.g., ≈36 MB for the proposed network instead of several GB for a typical 3D point cloud database), and real-time performance (e.g., our proposed method needs ≈10 ms to estimate the pose per image).

Our contributions can be summarized as follows:

- We propose a probabilistic selection method to generate synthetic "images" leveraging the 3D map and feature correspondences. We experimentally show that proposed synthetic image generation technique is very accurate and can replace ad-hoc techniques [8].
- We address the domain adaptation problem in pose estimation by augmenting the training set with synthetically generated training images and poses.
- We propose a DNN architecture based on an ensemble of spatial pyramid max-pooling units [7] for pose regression. This network can be trained (from scratch) on those real + synthetic datasets without pretraining and is significantly smaller than PoseNet-like networks reported in the literature.
- To demonstrate the quality of the proposed synthetic image generation method, we also include results for color image synthesis and compare to several existing baseline methods.

Overall, we demonstrate in this work that a relatively light-weight pose regression network trained on synthetic data substantially improves its generalization ability to novel poses.

2 Mining synthetic views

In this section, we discuss our proposed method to mine synthetic poses and feature descriptors "images". We leverage an SfM framework to obtain a set of geometrically consistent inlier 3D points $\overline{\mathcal{X}}$. The outlier points $\overline{\mathcal{O}}$, i.e. features detected on the training images but not participating in the reconstruction, are also stored. Inlier and outlier points follow different probabilistic models as explained below.

In the first step we ensure that no information about the test images remains in the training data: we remove all points $X_i \in \overline{\mathcal{X}}$ in the original point-cloud that are only seen in the test images. Further, feature points observed in fewer than two training images are placed into $\overline{\mathcal{O}}$. The observed image indices and the respective feature descriptors for the remaining points corresponding to the test images are also removed.

Each point X_i in the reconstructed 3D point cloud $\mathcal{X}$ contains the 3D location, the indices of the images where the point X_i was observed and the indices of the keypoints in the observed image. Moreover, the positions and the orientations $\{\mathcal{P}_{ij}\}$ of the observed images $\mathcal{I}_j$ at the point X_i are also available, which enable us to model the detectability of a 3D point at a particular pose. The detectability of an outlier is also modeled as follows: each descriptor from $\overline{\mathcal{O}}$ is assigned to the NN of a fixed vocabulary set $\mathcal{O}$. Thus, the poses $\{\mathcal{P}_{ij}\}$ of the observed images $\mathcal{I}_j$ at the points of $\mathcal{O}$ are available, which are utilized to model the detectability of an outlier in $\mathcal{O}$.

Inspired by the idea of view synthesis in the context of absolute pose estimation [8, 17, 25] we employ a probabilistic sampling strategy as described below. Note that aforementioned methods utilize ad-hoc techniques which do not necessarily produce accurate samples that can serve our purpose (validated in Section 5). Let us denote the joint probability distribution of pose-point space by $p_{\mathcal{P}\times\mathcal{X}}(P,X)$, where $\mathcal{P}$ is the 6D pose space and $\mathcal{X}$ is the 3D point space of the scene. The pose space $\mathcal{P} \subset \mathbb{R}^{4\times4}$ can be realized as a special Euclidean

group $SE(3)$ of intrinsic dimension 6 where $\exp_{\mathcal{P}} : \mathbb{R}^6 \to \mathcal{P}$ and $\log_{\mathcal{P}} : \mathcal{P} \to \mathbb{R}^6$ are the exponential and logarithm maps of $\mathcal{P}$ respectively. We use a conditional model for the joint probability,

$$p_{\mathcal{P}\times\mathcal{X}}(P, X \text{ is visible}) = p_{\mathcal{P}}(P)\, p_{\mathcal{X}|\mathcal{P}}(X \text{ is visible}|P). \tag{1}$$

$p_{\mathcal{P}}(P)$ determines the probability of a particular pose $P \in \mathcal{P}$, and $p_{\mathcal{X}|\mathcal{P}}(X \text{ is visible}|P)$ essentially determines the visibility and detectability of a 3D point $X \in \mathcal{X}$ w.r.t. the pose $P \in \mathcal{P}$. We estimate the parameters of the distributions $p_{\mathcal{P}}$ and $p_{\mathcal{X}|\mathcal{P}}$ from the training poses (described in the following Sections 2.1 and 2.2) and then generate synthetic "images" by sampling from the learned distributions.

2.1 Modeling $p_{\mathcal{P}}(P)$

$p_{\mathcal{P}}(P)$ is modeled as a mixture distribution, i.e. $p_{\mathcal{P}}(P) = \phi_0 \mathcal{N}^0_{\mathcal{P}} + \sum_{j=1}^{K} \phi_j \mathcal{N}_{\mathcal{P}}(P; Q_j, \mathcal{C}_j)$. Each mixture component is a Gaussian in the 6D Lie algebra representation, $\mathcal{N}_{\mathcal{P}}(P; Q_j, \mathcal{C}_j) \propto \exp(-\|\log_{\mathcal{P}}(PQ_j^{-1})\|^2_{\mathcal{E}_j}/2)$ (with $\|\cdot\|_{\mathcal{C}}$ denoting the Mahalanobis distance w.r.t. the covariance $\mathcal{C}$). Let $\mathbf{p} := \log_{\mathcal{P}}(P)$ and $\mathbf{q}_j := \log_{\mathcal{P}}(Q_j)$, then $\mathbf{p}$ follows a multivariate Gaussian distribution with mean $\mathbf{q}_j$ and covariance matrix $\mathcal{C}_j$. The number of mixtures K is chosen as the number of training images, hence our mixture model is essentially a kernel density estimator. For each training pose P_j the 10 nearest neighbor poses $\{P_{ij}\}$ within 20° of the P_j's viewing direction are selected to estimate the mean $\mathbf{q}_j$ and covariance $\mathcal{C}_j$ of $\log_{\mathcal{P}}(P_{ij})$.

We use the 0-th mixture component $\mathcal{N}^0_{\mathcal{P}}$ to add domain knowledge (in order to also sample poses far from the given training poses): For outdoor datasets, $\mathcal{N}^0_{\mathcal{P}}$ is induced by all training cameras with camera centers near a robustly fitted plane, and for indoor datasets $\mathcal{N}^0_{\mathcal{P}}$ is estimated using all training poses. Hence, $\mathcal{N}^0_{\mathcal{P}}$ allows to sample poses that are very different from the training data. The intrinsic camera parameters i.e, focal length, radial distortions of the synthetic views were chosen to be the same as the training image corresponding to the chosen Gaussian. Overall, no prior knowledge of the test poses is leveraged.

2.2 Modeling $p_{\mathcal{X}|\mathcal{P}}(X \text{ is visible}|P)$

$p_{\mathcal{X}|\mathcal{P}}(X \text{ is visible}|P)$ models the visibility and detectability of a 3D point X in an image defined by the camera pose P. As a simplifying assumption we ignore the co-occurrence of 3D points in images and use a fully factorized model for sets of 3D points, $p_{\mathcal{X}|\mathcal{P}}(X_i \text{ is visible} \forall i \in \mathcal{I}|P) = \prod_{i\in\mathcal{I}} p_{\mathcal{X}|\mathcal{P}}(X_i \text{ is visible}|P)$. This independence assumption also allows to easily sample 3D points that are predicted to be visible for any query pose. In our formulation $p_{\mathcal{X}|\mathcal{P}}(X \text{ is visible}|P)$ is the product of two probabilities,

$$\begin{aligned} p_{\mathcal{X}|\mathcal{P}}(X_i \text{ is visible}|P) &= \mathcal{N}_{\mathcal{X}}(P; \hat{Q}_i, \hat{\mathcal{C}}_i) \cdot p(X_i \text{ is inside frustum of } P) \\ &\propto \exp(-\|\log_{\mathcal{P}}(P\hat{Q}_i^{-1})\|^2_{\hat{\mathcal{C}}_i}/2) \cdot p(X_i \text{ is inside frustum of } P). \end{aligned} \tag{2}$$

$p(X_i \text{ is inside frustum of } P) \in \{0, 1\}$ captures the viewing frustum test, which modulates the underlying Gaussian model. In analogy to the model for $p_{\mathcal{P}}$ the 6D mean $\hat{Q}_i$ and covariance $\hat{\mathcal{C}}_i$ is estimated using the training poses the 3D point X_i is visible in. Since 3D points might be visible in only a few images, and therefore $\hat{\mathcal{C}}_i$ could be rank-deficient, we augment $\hat{\mathcal{C}}_i$ with a diagonal matrix $\frac{1}{5} \cdot I_{6\times 6}$ to ensure full rank of $\hat{\mathcal{C}}_i$. Rejection sampling is used to draw visible points for a pose P: random variables $u_i \sim \mathcal{U}_{[0,1]}$ are sampled for all i and compared against

the probability in Eq. (2). Once a 3D point is chosen, the feature descriptor $\mathcal{F}_j$ is copied from the nearest training image I_j. The pixel coordinate (p_i, q_i) of the feature point is computed under the perspective projection of the 3D point X_i on the image plane. The rotation of the feature descriptor is copied from the selected image. In order to make the synthesized view robust to noise and outliers, we also apply the following:

- Additive Gaussian noise with diagonal co-variance Σ_x is added to the feature descriptors $\mathcal{F}_i$. The matrix Σ_x is determined based on the descriptors in the training data. Further, Gaussian noise with 1 pixel variance is added to the projected pixel locations (p_i, q_i).
- Outlier keypoints $\mathcal{O}$ are also tested the detectability in similar manner to the synthetic poses. In this case, we pick 5 training images containing most number of detected outliers—fit an homography through all the inlier points of the synthesized pose and the training pose—then use the same homography to project the outliers (20%) to the target synthetic image. This step is omitted if the number of common inliers is less than 50.

2.3 Feature processing

Let $\mathcal{S} = \{x_1, x_2, \ldots x_N\}$ be the feature descriptors detected or synthesized in the image plane. The i-th keypoint x_i is described by its pixel coordinates (p_i, q_i), scale s_i, orientation θ_i, and a feature descriptor $\mathcal{F}_i$ of dimensionality D. The number of of sparse features extracted from input images varies, and there are two complementary approaches to facilitate the use of a CNN on sparse features: the method used in [4] embeds the sparse features into a dense grid (and has to handle feature collisions and empty locations), and PointNet [16] (and similarly [24]) processes each input feature independently and uses max-pooling to symmetrize the network output. Our approach combined elements of both by using spatial binning, independent feature processing and global max-pooling: we arrange the set of keypoints on a 2D regular grid based on the pixel locations (p_i, q_i) and split the input image of size $W \times H$ into $d_1 \times d_2$ cells (where each cell is of size $W/d_1 \times H/d_2$ pixels). If a cell occupies multiple features, we select a feature randomly and obtain a $(D+5)$-dimensional vector $\big(\mathcal{F}_i, p_i \bmod d_1, q_i \bmod d_2, \sin\theta_i, \cos\theta_i, \log(1+s_i)\big)$ corresponding to a keypoint x_i. This results in a spatially organized array of at most $d_1 \times d_2$ feature descriptors (with $D+5$ dimensions), which is the input to the network. Empty cells are represented by a zero feature vector. In all of our experiments, we used $d_1 = d_2 = 16$. The rationale behind our spatial binning approach is to reduce the amount of processing and to balance the spatial distribution of features across the image.

3 Spatial pyramid pose net for pose regression

Given a set of sparse feature descriptors $\mathcal{S}$ detected on a test image, the task is to estimate the pose of the camera (R, T) with respect to the global coordinate frame. In the following we describe our proposed DNN architecture for pose regression, which we term "spatial pyramid pose net" or SPP-Net. SPP-Net takes a set of sparse feature descriptors as input and estimates 6 d.o.f. camera pose. The input descriptors undergo a number of 1×1 convolutions/ReLU layers, followed by an ensemble of multiple parallel max-pooling layers and three subsequent fully connected pose regression layers. The proposed SPP-Net is lightweight and fast, and it performs competitively with the original PoseNet [11]. Moreover, the proposed architecture has an additional advantage that it can be further trained on augmented images generated from the reconstructed 3D map. Being trained on such synthetic poses, it

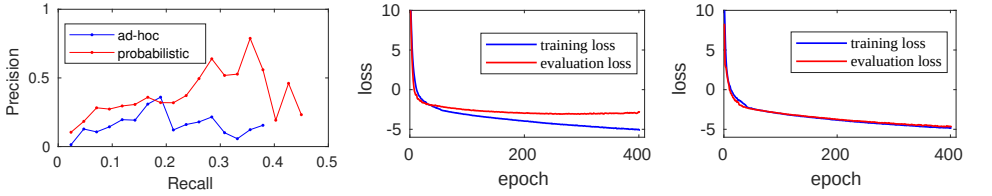


Figure 2: Left: Precision vs. recall curve of proposed probabilistic detection method and ad-hoc method [8]. Middle: training and testing losses when using only real training images. The testing loss quickly stalls. Right: training and testing losses when utilizing real training images and synthetic data corresponding to *test* poses.

improves results on benchmark datasets.

Network Architecture The proposed network consists of an array of deep feed-forward subnets, an ensemble layer of max-pooling units at different scales and two fully connected layers followed by the output pose regression layer. The detailed descriptions and a comparison of different architectures can be found in the supplementary material.

Loss function We follow [10] in the choice of the loss,

$$\mathcal{L}_\sigma(q,T) \propto \sigma_q^{-2}\left\|q^\dagger - q/\|q\|\right\| + \sigma_T^{-2}\|T^\dagger - T\| + \log\sigma_q^2 + \log\sigma_T^2 \tag{3}$$

where $q^\dagger$ and $T^\dagger$ are the ground truth orientation and position of the image, respectively. Note the reprojection error could be a geometrically more meaninful loss function, especially since SPP-Net takes sparse features as input. As also pointed out in [10], we found it difficult to train a network directly using the reprojection loss, hence we rely on (3) instead.

4 Realistic RGB image generation

The underlying task is to synthesize realistic RGB images for novel viewpoints given multiple input images. It has a number of applications in computer vision and virtual reality, *e.g.* it can create realistic video footage given a set of images. In a small baseline setting view synthesis can be solved by explicit dense correspondence search and subsequent interpolation. It also has been addressed by using CNNs recently [31]: instead of predicting a new image at the target pose (which often produces blurry outputs), the appearance flow (relative pixel shifts) is predicted, which is then leveraged (interpolated, differentiable) to synthesize the target RGB image. However, the method suffers from poor generalization ability in a large baseline setting.

In this work, we utilize synthetic sparse feature descriptors [sec. (2.2)] to further synthesize realistic RGB images. A conditional Generative adversarial network (GAN) [9] is proposed for this task. We use an architecture similar to [9]—consisting of a generator and a discriminator. The generator network takes a sparse feature descriptors of size $d_1 \times d_2 \times (D+5)$ as input and generates an RGB image of size $256 \times 256 \times 3$ as an output. The discriminator takes (descriptors, RGB) image pair as input and predicts if the input pair is real or synthetic. We train both the networks from scratch and discard the discriminator once the network is trained. A detailed description of the network architecture can be found in the supplementary material.

5 Experiments

The proposed pose regression network SPP-Net is trained on a number of widely used datasets for absolute pose estimation. The loss (3) is minimized using ADAM [12] with a batch size of 100. The weight decay is set to 10^{-5}. The network is trained for 400 epochs with an initial learning rate 0.01 which is gradually decreased by a factor of 10 after every 100 epochs. All the experiments are evaluated with Tensorflow [1] on a desktop equipped with a NVIDIA Titan X GPU, where evaluation of the SPP-Net requires about $2.5-5$ms of run-time. Note that the computation of descriptors requires another 2.5ms[1], and further 2ms are needed for spatial descriptor binning. Thus, the total frame time is approximately $7-10$ms. The network takes $2-4$ hours to train on a typical dataset, and the full set of weights consumes about 37 MB. In the following we describe the datasets utilized for evaluation of the proposed method.

Cambridge Landmarks Datasets [11] provides a labeled set of image sequences of different outdoor scenes where the ground-truth poses were obtained by utilizing VisualSFM [28]. The datasets also provide the SfM "reconstruction"(*.nvm*) files containing the 3D point-cloud and the 2D-3D assignments required by our pose augmentation. Creation of synthetic images takes approximately two hours for a typical dataset (per sequence). The SPP-Net is trained on the augmented (training and synthesized) dataset. The training and test image sequences were taken from distinct paths to make the pose estimation more challenging.

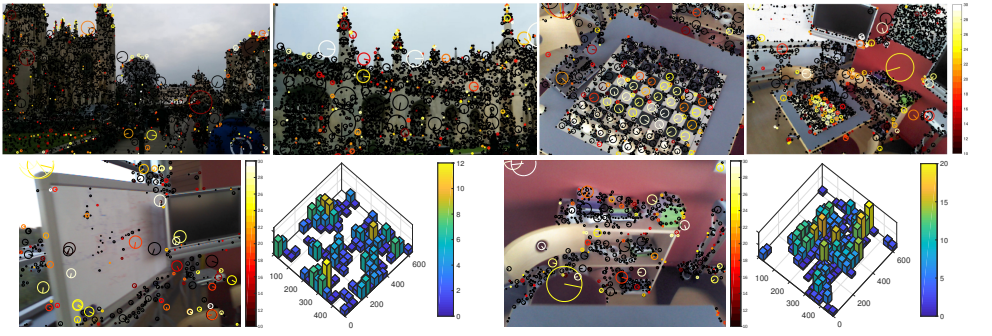
The Microsoft 7-Scenes Dataset [21] consists of texture-less RGB-D images of seven different indoor scenes. The 3D map and the feature descriptors are not provided with the datasets required by the proposed augmented pose generation technique. Thus, we reconstructed the 3D point cloud from scratch using toolboxes such as VisualSFM [28] and COLMAP [20]. We register the SfM camera poses to the KincetFusion reference poses by a similarity transformation, and the same transformation is used to register the 3D points w.r.t. the reference poses. A constrained bundle adjustment method is applied, that holds the camera poses fixed and thus only optimizes the 3D structure.

5.1 Validation of the proposed pose augmentation

To validate the efficiency of the augmented poses, we conduct an experiment with a difficult sequence (heads) of Microsoft's 7-Scenes Dataset [21]. From the 3D map of training images, we generate synthetic feature sets corresponding to the *test poses*. We evaluate the synthetic feature descriptors by comparing with the original descriptors. The matching is a hit if *cosine* of the angles between the descriptors is grater than 0.85 and spatial difference is less than 5 pixels. A precision-recall of all the test images of our probabilistic method including the ad-hoc technique [8] are plotted in Fig. 2(a). The proposed SPP-Net is then trained on the training images + the synthetically generated descriptors and evaluated on the descriptors extracted from the original test images. The generated synthetic test images do not exploit any test image content but the 6 d.o.f. poses.[2] If the network is provided only with the training images, it does not generalize well to the test images. However, after adding synthetic test poses to the training data, the evaluation loss decreases in conjunction with the training loss. In Figure 2(b-c) we depict the training and validation loss with and without additional synthetic training data. These results illustrate the benefits of our synthetic pose augmentation method.

[1] https://github.com/Celebrandil/CudaSift/blob/Maxwell/README.md

[2] Note that except the current experiment, no information of the 6 d.o.f. test poses were incorporated.



(a) Angular Error = 12.71°, Positional Error = 0.36m (b) Angular Error = 6.52°, Positional Error = 0.21m

Figure 3: Top row: the keypoints for images from the "Kings College" and "Chess" sequences are displayed. Highly relevant feature points (with average contributions > 10) are colored (using the "hot" colormap) according to their contributions to the ensemble layer (see text). Bottom-row: a pair of typical test images of the "chess" sequences are displayed along with the histograms where 56.2% and 51.8% cells of the 16×16 grids are empty. In general, feature descriptors at larger scales seem to be more relevant. Further, in the King's college sequence, keypoints near the building outlines are relatively consistently important for pose prediction. The indoor "Chess" sequence exhibits a mix of features on the unique chess pieces and on the background.

5.2 Visualizing leveraged image features

It is instructive to visualize which keypoints extracted in the image are eventually most relevant to predict the pose parameters. We define the contribution of a feature to pose prediction as the number of max-pooling units where the given feature is the winning branch in the max-pooling step. The higher the contribution, the more prominent is this feature represented in the following pose regression layers. In Fig. 3 we display the most contributing feature points for two complementary scenes. For outdoor environments many features relevant for pose prediction cluster near the skyline induced by building, and for indoor scenarios one generally observes a mix between distinctive small-scale features and background features at a larger keypoint scale. Further, we display a pair of images where more than 50% of bins (cells) are empty yet SPP-Net successfully estimates the pose. This indicates that SPP-Net shows robustness to unevenly distributed image features.

5.3 Benchmarking localization accuracy

Baseline Methods We compare the proposed SPP-Net against the following baselines:

- Active Search [18]: This is a direct feature-based approach where the feature descriptors are matched across the 3D point-cloud and the pose is estimated using the P3P algorithm.
- Original PoseNet [11]: The first convnet-based method where the last soft-max classification layer of GoogleNet is replaced by the fully connected regression layers.
- PoseNet LSTM [27] : Similar as above, but multiple LSTM units were utilized to the convnet features followed by a regression layers.
- PoseNet Geometric Cost [10] (PoseNet2): The network is trained with the same loss function as ours and fine-tuned with the re-projection cost.

Scene	Area or Volume	Active Search (SIFT) [28]	Original PoseNet [11]	PoseNet LSTM [23]	PoseNet Geo. Cost [10]	SPP-Net	SPP-Net (with Synthetic data)
Great Court	8000m^2	–	–	–	6.83m, 3.47°	13.2m, 8.02°	5.42m, 2.84°
King's College	5600m^2	0.42m, 0.55°	1.66m, 4.86°	0.99m, 3.65°	0.88m, 1.04°	1.91m, 2.36°	0.74m, 0.96°
Old Hospital	2000m^2	0.44m, 1.01°	2.62m, 4.90°	1.51m, 4.29°	3.20m, 3.29°	2.51m, 3.74°	2.18m, 3.92°
Shop Facade	875m^2	0.12m, 0.40°	1.41m, 7.18°	1.18m, 7.44°	0.88m, 3.78°	1.31m, 7.82°	0.59m, 2.53°
StMarrys Church	4800m^2	0.19m, 0.54°	2.45m, 7.96°	1.52m, 6.68°	1.57m, 3.32°	3.21m, 6.97°	1.44m, 3.31°
Street	50000m^2	0.85m, 0.83°	–	–	20.3m, 25.5°	–	24.5m, 23.8°
Chess	6m^3	0.04m, 1.96°	0.32m, 6.60°	0.24m, 5.77°	0.13m, 4.48°	0.22m, 7.61°	0.12m, 4.42°
Fire	2.5m^3	0.03m, 1.53°	0.47m, 14.0°	0.34m, 11.9°	0.27m, 11.3°	0.37m, 14.1°	0.22m, 8.84°
Heads	1m^3	0.02m, 1.45°	0.30m, 12.2°	0.21m, 13.7°	0.17m, 13.0°	0.22m, 14.6°	0.11m, 8.33°
Office	7.5m^3	0.09m, 3.61°	0.48m, 7.24°	0.30m, 8.08°	0.19m, 5.55°	0.32m, 10.0°	0.16m, 4.99°
Pumpkin	5m^3	0.08m, 3.10°	0.49m, 8.12°	0.33m, 7.00°	0.26m, 4.75°	0.47m, 10.2°	0.21m, 4.89°
Red Kitchen	18m^3	0.07m, 3.37°	0.58m, 7.54°	0.24m, 5.52°	0.23m, 5.35°	0.34m, 11.3°	0.21m, 4.76°
Stairs	7.5m^3	0.03m, 2.22°	0.48m, 13.1°	0.40m, 13.7°	0.35m, 12.4°	0.40m, 13.2°	0.22m, 7.17°

Table 1: Median localization results for the Cambridge [11] and 7-scenes datasets [21].

- Proposed SPP-Net trained without augmented posses is also included as baseline. Note that 5% randomly chosen "images" from the training data are employed as validation set.
- The computationally expensive DSAC approach [3] is not added to the baseline methods.

Results on Cambridge Landmarks Datasets The results are displayed in Table 1. SPP-Net without pose augmentation yields results similar to the original PoseNet, and is comparable to PoseNet2 [10] once trained with the augmented dataset. However, SPP-Net is more lightweight, much faster and does not require to be pre-trained on a larger datasets (*e.g.* ImagNet). Note that the proposed network is of limited size, increasing the size of the network (and the number of augmented poses) shall further improve the performance.

Results on Seven Scene Datasets are shown in Table 1. Note that as the reference poses are rather noisy, hence the similarity transformation does not relate SfM poses and reference poses well in all scenes. In particular, we observed good results on "Stairs", "Heads" and "Fire" sequences as the similarity transformation is a good fit for these scenes. Overall we obtained very competitive results in this dataset.

5.4 Evaluation of RGB image synthesis

We compare the proposed SPP-Net against the following baselines:

- Appearance Flow (AF) [31]: in order to generate a training dataset for the method proposed in [31], the "pose distances" (defined as the positional distance+0.1×angular distance) between all images in a dataset computed. The training set comprises 50,000 image pairs with a minimal pose distance of 1 (to ensure a significant baseline). At test time the nearest neighbor to the target pose (in terms of pose distance) from the training dataset is chosen to generate an RGB image. The method from [31] utilizes multiple images, however we did not observe any improvement with multiple inputs.
- We also include [4] as baseline to invert synthetic feature descriptors to RGB images. Here the network is trained to regress RGB pixels from sparse features. The network is trained on descriptor-image training pairs and to minimize ℓ_2 loss. Note that during the evaluation it takes our synthetic feature descriptors as input and produces RGB images.

In this experiment, "chess" sequence of the 7-Scenes Dataset [21] is employed. Synthetic descriptors generated at the training poses and original RGB images are used as the training

Figure 4: RGB images synthesized by different methods at the test poses of one of the image sequences of 7-Scenes [21] and Cambridge Dataset [11]. The indices of the images of the test sequence are mentioned in the top of the figure. More results can be found in suppl. mat.

pairs. A random crop strategy is employed for each descriptors-image pair during training. Synthetic descriptors generated at test poses are then used to synthesize RGB test images. In Fig. 4, we compare the synthesized RGB images generated by different methods. It can be observed that AF [31] can still predict well for nearby poses, however, the method in general degrades when encountering larger baselines. The direct descriptor inversion [4] produces accurate yet blurry outputs. Our proposed GAN based method using pose augmentation yields convincingly realistic RGB images—although abstract geometric constraints are not always satisfied. Further, we trained Posenet [11] for the pose estimation task with our synthesized RGB images as additional training data. Nevertheless, SPP-Net with augmented sparse descriptors still has better pose regression performance. Additional results can be found in the supplementary material.

6 Conclusion

In this work we presented a synthetic view augmentation technique, which is used to train a deep learning architecture for pose prediction, and which is further utilized for realistic RGB image generation. The proposed augmentation method aims to be sufficiently realistic by using an underlying 3D point cloud and probabilistic models for 3D point visibility and outlier processes. Thus, pose regression can be trained for any region in the pose space using a virtually unlimited amount of (synthetic) training data. We performed several numerical experiments to validate our architecture and the proposed augmentation procedure.

Despite the dominance of dense convolutional deep learning methods in computer vision we believe that there are many opportunities for combining deep learning with traditional sparse features. In particular, new view synthesis is one of these applications where such combination is promising. Enhancing the quality of view synthesis *e.g.* by leveraging multiple training images is subject of future work.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. In *OSDI*, volume 16, pages 265–283, 2015.

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

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

[4] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *Proc. CVPR*, pages 4829–4837, 2016.

[5] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *Proc. CVPR*, pages 2315–2324, 2016.

[6] Ankur Handa, Viorica Patraucean, Vijay Badrinarayanan, Simon Stent, and Roberto Cipolla. Understanding real world indoor scenes with synthetic data. In *Proc. CVPR*, pages 4077–4085, 2016.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *Proc. ECCV*, pages 346–361, 2014.

[8] Arnold Irschara, Christopher Zach, Jan-Michael Frahm, and Horst Bischof. From structure-from-motion point clouds to fast location recognition. In *Proc. CVPR*, pages 2599–2606, 2009.

[9] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proc. CVPR*, pages 1125–1134, 2017.

[10] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proc. CVPR*, 2017.

[11] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proc. ICCV*, pages 2938–2946, 2015.

[12] D.P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[13] Xiaotian Li, Juha Ylioinas, and Juho Kannala. Full-frame scene coordinate regression for image-based localization. *arXiv preprint arXiv:1802.03237*, 2018.

[14] Yunpeng Li, Noah Snavely, and Daniel P Huttenlocher. Location recognition using prioritized feature matching. In *Proc. ECCV*, pages 791–804, 2010.

[15] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[16] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proc. CVPR*, 2017.

[17] Torsten Sattler, Michal Havlena, Konrad Schindler, and Marc Pollefeys. Large-scale location recognition and the geometric burstiness problem. In *Proc. CVPR*, pages 1582–1590, 2016.

[18] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE Trans. on Pattern Anal. and Mach. Intell.*, 39(9):1744–1756, 2017.

[19] G. Schindler, M. Brown, and R. Szelisk. City-scale location recognition. In *Proc. CVPR*, 2007.

[20] Johannes L Schonberger, Filip Radenovic, Ondrej Chum, and Jan-Michael Frahm. From single image query to detailed 3d reconstruction. In *Proc. CVPR*, pages 5126–5134, 2015.

[21] 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 *Proc. CVPR*, pages 2930–2937, 2013.

[22] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. ICLR*, 2015.

[23] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proc. CVPR*, pages 1–9, 2015.

[24] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of CNN activations. *Proc. ICLR*, 2016.

[25] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *Proc. CVPR*, pages 1808–1817, 2015.

[26] Florian Walch, Caner Hazirbas, Laura Leal-Taixé, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization with spatial LSTMs. In *Proc. ICCV*, 2017.

[27] Tobias Weyand, Ilya Kostrikov, and James Philbin. Planet-photo geolocation with convolutional neural networks. In *Proc. ECCV*, pages 37–55, 2016.

[28] Changchang Wu. VisualSFM: a visual structure from motion system, 2011.

[29] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *Proc. ICLR*, 2017.

[30] Wei Zhang and Jana Kosecka. Image based localization in urban environments. In *3D DPVT, Third International Symposium on*, pages 33–40. IEEE, 2006.

[31] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *Proc. ECCV*, pages 286–301, 2016.