Camera Relocalization by Exploiting Multi-View Constraints for Scene Coordinates Regression

Ming Cai, Huangying Zhan, Chamara Saroj Weerasekera, Kejie Li, Ian Reid
The University of Adelaide
{ming.cai, huangying.zhan, saroj.weerasekera, kejie.li, ian.reid}@adelaide.edu.au

Abstract

We propose a method for learning a scene coordinate regression model to perform accurate camera relocalization in a known environment from a single RGB image. Our method incorporates self-supervision for scene coordinates via multi-view geometric constraints to improve training. More specifically, we use an image-based warp error between different views of a scene point to improve the ability of the network to regress to the correct absolute scene coordinates of the point. For the warp error we explore both RGB values, and deep learned features, as the basis for the error. We provide a thorough analysis of the effect of each component in our framework and evaluate our method on both indoor and outdoor datasets. We show that compared to the coordinate regression model trained with single-view information, this multi-view constraint benefits the learning process and the final performance. It not only helps the networks converge faster compared to the model trained with single-view reprojection loss, but also improves the accuracy of the absolute pose estimation using a single RGB image compared to the prior art.

1. Introduction

Relocalizing a camera in a known environment is an important task in computer vision and robotic vision, with applications in Simultaneous Localization and Mapping (SLAM) and Virtual Reality. Commonly formulated as a retrieval problem, the goal of camera relocalization is to estimate the six degree of freedom (6DoF) pose of the camera w.r.t. to a fixed world coordinate frame given an RGB image (or augmented with depth image if a range sensor is available). This is also known as single-shot absolute camera pose estimation, which is different from accumulating relative camera motions between consecutive frames in a Visual Odometry system.

The mapping from the image space to the 6DoF transformation space can be found via geometric constraints based on correspondence matches between the query image and the pre-built map, or directly through a parametric model, *e.g.*, Convolution Neural Networks (CNNs). While achieving highly accurate pose estimates, the methods in the first group usually need to define a handcrafted feature with limited generalization ability. When correspondence matching fails (insufficient or erroneous matches), these methods likewise fail. To overcome this problem, methods in the second group use deep learning models to directly regress the pose of the camera from the image, *e.g.*, the well-known PoseNet [15]. Although PoseNet and its derivatives (*e.g.*, [14, 26]) can handle a variety of dynamic effects in the images such as lighting changes and motion blur, there is still a wide gap between the accuracies, with deep regression methods being significantly less accurate than those that rely on explicit correspondence and geometric constraints.

As a combination of the advantages of both categories,

Figure 1. **The projections of scene coordinates predicted by models trained with (*reprojection loss only*) and (*reprojection loss + reconstruction loss*) on a pair of test images.** In the left image we show some sample points (colored circles) for which we predict the 3D coordinates using two models: one trained with single-view reprojection loss and the other trained with the multi-view geometry-based reconstruction loss as the additional supervision. In the right view, whose relative pose to the left is known, we show the projections of the regressed coordinates from left image as squares (reprojection loss) and as stars (geometry loss). Observe that the geometry loss (i.e. with feature consistency constraints), produces a model that produces better coordinates, as seen by the better match locations of the star points compared with the squares. Best viewed in color.

DSAC++ (*LessIsMore*) [4] and its predecessor DSAC [2] apply a deep neural network over the image to establish the dense correspondences between 2D and 3D space by regressing each image pixel directly to its scene coordinates (i.e., its 3D coordinates in the *global* reference frame). Pose is then obtained using standard RANSAC [6] plus PnP solvers. In that sense, this model can also be interpreted as a network that performs 3D scene reconstruction from an single image, and then estimates the pose as a post process. Because of the robustness of the CNN coordinate regressor and the strong geometric constraints of the PnP solver, this line of methods achieves great accuracy for pose estimation, and even outperforms the traditional approaches [23].

The key to these methods working well is the ability of the deep model to map to the fixed 3D location of any given scene point from an image of that point. Since the viewpoint can be anywhere, the appearance of the scene point may vary, but the network should still regress to the same global 3D coordinates. It is not clear if such a network is capturing the invariance of features to different viewpoints and therefore implicitly encoding multi-view geometric constraints, or if it is acting as a huge look-up table that simply memorizes all possible appearances and corresponding mappings. Regardless, in our work we aim to make the multi-view constraints more explicit during training.

To that end, the main innovation in this paper is to exploit constraints from multi-view geometry to supervise the learning of a model for scene coordinate regression. We aim to retain the advantages of the training for single view reconstruction, but to incorporate the additional information available from viewpoint invariant image features under motion parallax. Specifically, after predicting the scene coordinates from one image in the database during training, we project the predictions to another image that shares an intersection of the camera frustum with the query image, using the ground truth pose of the target image. We then compare local image feature descriptors – any difference that we assume arises from an error in the predicted scene coordinates – and use this error for back-propagation. In this work we explore two types of local image features: (i) simple RGB values (which are invariant to viewpoint under the common lambertian reflection assumption); (ii) high dimensional features that are learned to be good for matching [27, 28].

The advantage of our method is that it produces more accurate scene coordinates compared to the single-view training approaches. Therefore it yields better 2D-3D correspondence for single view pose estimation using RANSAC during test. On top of this accuracy improvement, our system also avoids the scale issue that the methods with single view training may suffer. The reason for the first stage – training with pseudo depth in the RGB-only case (See section 2 for details) – is needed in DSAC++ [4] is that it assigns an initial scale to the scene coordinates. A good guess of the scale helps the next learning stages and vice versa. This makes it heavily reliant on the heuristic. In contrast, experiments show that our method relaxes the requirement for this strong prior through the use of multi-view geometry. One should bear in mind that our training pipeline also requires the initialization stage, but only a rough guess for depth is needed to avoid the case when all the photometric/feature construction losses are meaningless.

A similar technique has been applied to the topic of self-supervised depth estimation such as [8, 28, 29]. The differences between these works and ours are twofold. First, the objectives are different. Depth estimation focuses on purely recovering the geometric structure of each frame. However in our task, the scene coordinates inferred from the network are intermediate values whose purpose is ultimately to enable camera pose estimation.

Second, the *label consistency* of these two representations is different, and this has a significant effect on learning. In the case of depth estimation, the labels are the depth values of each pixel. As the camera moves these depth values change even for the same part of the scene because they are camera position dependent. In contrast, for the scene coordinate estimation task, the scene coordinates are described in a fixed world coordinate frame – the label for a scene point is its 3D coordinates and this label is consistent across all viewing locations and appearances, *i.e.*, independent of camera location. We believe this makes the 3D scene coordinates easier to regress than depth values in the known environments.

Our main contribution can be summarized as:

- We explicitly introduce the multi-view geometric constraints of temporal image pairs to the learning of the scene coordinate regression model.
- We design a multi-component loss function as the main supervision in our system. It contains two types of image feature reconstruction errors and a structural smoothness penalty over the featureless regions of the scene. The effect of each component to the model is investigated via a detailed ablation study.
- We achieve state-of-the-art results for pose estimation on both indoor and outdoor datasets with only RGB images, while requiring less training steps compared to the prior art. In addition, we observe that our method obtains consistent performance regardless of varying scene-dependent factors, *e.g.*, depth range.

2. Related Work

The classic solutions for camera relocalization are mainly based on extraction of (handcrafted) sparse image features (*e.g.*, [19, 22]), establishing feature correspondences, followed by well-studied geometric methods to solve for pose using the correspondences. Having the cor-

respondences established between an image and the map database which is built beforehand, the pose of the camera can be estimated using the PnP algorithm and its variants [17, 7]. To further improve the robustness of the pose prediction, RANSAC has been widely applied to 1) generate a pool of pose hypotheses; 2) then score them according to the number of compatible inliers; and 3) iteratively refine the best hypothesis based on the consensus of inliers. This three-stage pose estimation pipeline is continuously acting as the cornerstone of the state-of-the-art methods in this area, which will be discussed shortly.

PoseNet was introduced in Kendall *et al*. [15] which pioneered the idea of applying a deep learning model to the problem of camera pose estimation. A CNN is used to extract high-dimensional features directly from the RGB image, followed by two fully connected layers to regress the translation vector and the rotation quaternion. Although the model is robust to dynamic changes of the scene due to its high-level generalization ability, the performance of PoseNet [15] and its variants [13, 14, 26] do not perform sufficiently well for accurate localization. Nevertheless, the most notable improvement comes from the subsequent geometric loss based PoseNet [14]. It leverages the physical model of the scene and supervises the learning of the pose regression model by minimizing the reprojection error of the 3D points, eliminating the dependence on the choice of hyperparameters between translational and rotational losses. Moreover, a homoscedastic task loss is also used to learn the model, which relies on RGB information only and achieve on-par performance to the RGB-D version. The need of the 3D model however means that this method is inapplicable when only RGB images are at hand. Recently, Balntas *et al*. [1] proposed RelocNet that relies on evaluating the similarity between the query image and images in the training database. The pose of the query image is then recovered based on the absolute pose of its nearest neighbour and the estimated relative transformation between them. Despite the progress, there is much room for improvement in accuracy for these methods.

The idea of using scene coordinates to obtain dense 2D-3D correspondences was initially proposed by Shotton *et al*. in [24]. A Random Forest was trained to infer the 3D scene (world) coordinate for the image pixels with RGB-D data. The RANSAC pipeline is then revisited to estimate the camera pose accurately. Valentin *et al*. [25] exploits the uncertainty in the estimate from the Random Forest to benefit the pose optimization. This work is then further extended to the object-centered scenario in [3], where they used only the RGB image as input for object pose estimation.

DSAC [2] and DSAC++ [4] deploy two versions of an end-to-end scene coordinate regressor based on CNNs, and are devoted to make all the steps in the traditional RANSAC differentiable to enable an end-to-end training pipeline. In DSAC [2], the CNN for scene coordinate regression takes a small patch of the image as the input, and its output is the 3D coordinate associated to the central pixel of the input patch. As an ameliorator, DSAC++ [4] was upgraded to a fully convolutional network (FCN) [18] to improve the efficiency of training and to preserve the image-patch-to-coordinate property. To perform the three-step RANSAC algorithm, they start by sampling a pool of pose hypotheses using the PnP solver over the dense 2D-3D correspondences given by the scene coordinate prediction. In the second stage of ranking the hypotheses, DSAC [2] scores them with another CNN whose input is the reprojection error map of the predicted scene coordinates given each pose hypothesis and the camera intrinsics. On the other hand, to overcome the overfitting issue of the scoring CNN in DSAC [2], DSAC++ [4] simply uses a soft inliers counting scheme to evaluate the merits of the hypotheses. The difference also exists in the last refinement step. To make this iterative procedure differentiable, DSAC [2] approximates the gradient via finite differences, and DSAC++ [4] uses the iterative Gauss-Newton algorithm to linearize the model. Combining these techniques, they achieve the state-of-the-art result for camera relocalization in both indoor and outdoor scenes, even without the 3D model of scene.

A multi-step scheme is also adopted in the training phase of DSAC++ [4]. The training of the scene coordinate regression CNN consists of three stages and the performance of the model progressively increases with additional training. Since they will be repetitively mentioned hereinafter, we give a brief introduction to them. The scene coordinate regression model is initially trained with either ground truth scene coordinates or a heuristic assuming a constant distance of the scene, depending on the availability of depth images or the 3D scene model. In the second stage, the model is enhanced by the supervision from the distance between the 2D projection of predicted scene coordinates (given the ground truth camera pose) and the ideal image pixel position, namely the reprojection error. In the third step, DSAC++ [4] refines the model with an end-to-end scheme that combines the inlier soft counting based hypotheses scoring and differentiable refinement that is mentioned above, resulting in superior performance.

Along with scene coordinates, Bui *et al*. [5] also estimate the confidence/uncertainty of the scene coordinates as an auxiliary prediction from the network, and then run RANSAC using those inferred coordinates that have high confidence, which improves the robustness of the system.

3. Our Method

The overall objective of our work is to efficiently train a FCN-based scene coordinate regression model using multi-view geometric constraints, and then apply this model to a single view RGB image to infer dense 2D-3D correspon-

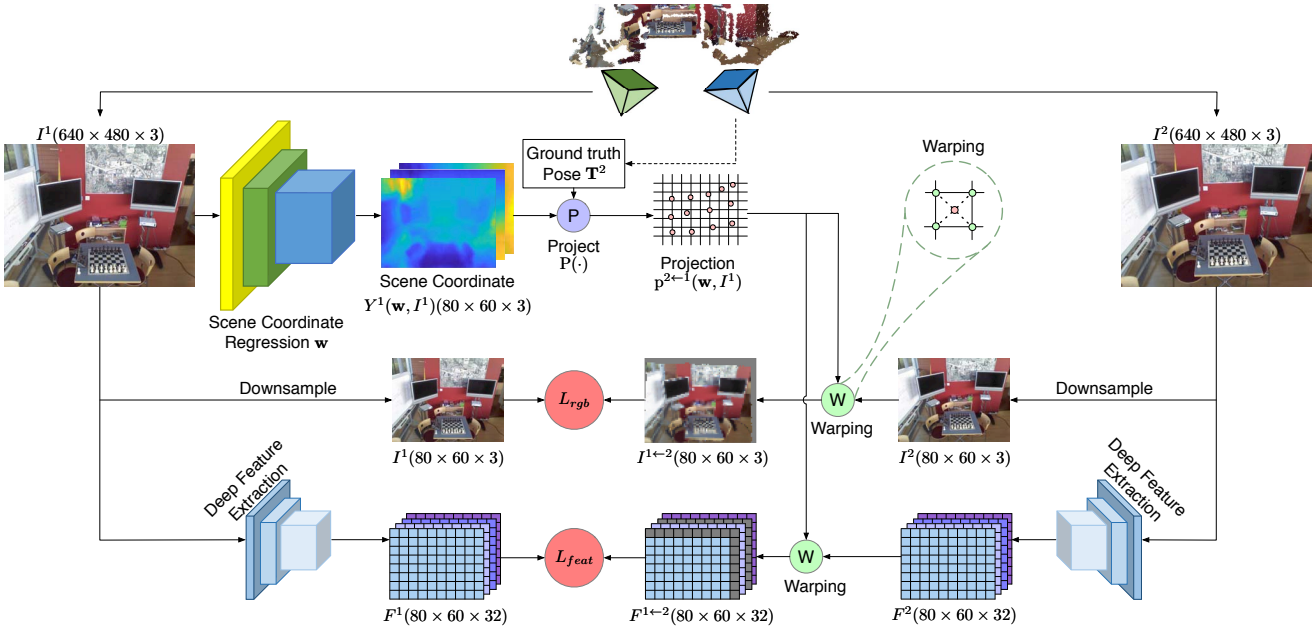

Figure 2. **The training pipeline of our framework with photometric loss and feature reconstruction loss.** The spatial size of all variables are specific for *7Scenes* dataset. The reprojection loss and smoothness prior loss are omitted for simplicity.

dences for pose estimation in a RANSAC pipeline. We start by describing the network architecture in Section 3.1. In addition to the single view reprojection loss, we introduce three more supervisions that come from the multi-view geometry induced by camera motion. The photometric warp error based image reconstruction loss is introduced in Section 3.2. An additional deep feature reconstruction loss which takes contextual information into consideration rather than per pixel color alone is introduced in Section 3.3. We propose a smoothness prior in 3D space to regularize training in Section 3.4, in order to mitigate the effect of featureless, ambiguous-to-match scene regions. Fig. 2 shows our framework in the training phase. The overall training loss and inference procedure are summarized in Section 3.5 and Section 3.6, respectively.

3.1. Scene Coordinate Regression

The FCN model of DSAC++ [4] is inherited into our system for scene coordinate regression. We denote this model as $\mathbf{w}$. The output of this network is the scene coordinate map $\mathbf{Y}(\mathbf{w}, I)$ of an input image I. Every element of this map is a 3D vector $\mathbf{y}_{i,j} \in \mathbb{R}^3$, which represents the coordinates in the world reference frame of the point that corresponds to an image pixel. This FCN comprises of 12 convolutional layers, 3 of which have stride size 2. Thus, $\mathbf{Y}(\mathbf{w}, I)$ is one-eighth the size of the input image I. This means that the 3D scene coordinates predicted by the model represent that of the center of 8×8 pixel tiles in I. Note that however each output corresponds to an overall receptive field of 41×41 pixels.

3.2. Photometric Reconstruction

The main supervision for learning the scene coordinate regression model in our framework comes from the reconstruction of two types of features: (i) RGB color; (ii) deep features trained to be good for matching [27] (used "off-the-shelf" in our work). We begin by introducing the photometric (RGB) constraint in this section.

Given an image pair $\{I^1, I^2\}$ from the pair selection stage, firstly I^1 is fed into the regressor $\mathbf{w}$ for predicting the map of the scene coordinates $\mathbf{Y}^1(\mathbf{w}, I^1)$. Then, they are projected onto the image plane of I^2 with the ground truth pose of second frame $\mathbf{T}^2$ and camera intrinsics $\mathbf{K}$, for computing the projected pixel positions $\mathbf{p}^{2\leftarrow 1}(\mathbf{w}, I^1)$. Using the RGB values of I^2 at $\mathbf{p}^{2\leftarrow 1}(\mathbf{w}, I^1)$, a warped image $I^{1\leftarrow 2}$ is formed to synthesize image I^1. The procedure can be formulated as Eq. (1),

$$I^{1\leftarrow 2} = f(\mathbf{Y}^1(\mathbf{w}, I^1), I^2, \mathbf{T}^2, \mathbf{K}), \tag{1}$$

where the function $f(\cdot)$ is the reconstruction function based on image warping. This operation is fully differentiable when using the bilinear interpolation reconstruction method proposed in Spatial Transformer Networks (STN) [12], which guarantees differentiability in the whole system.

The loss based on photometric difference between the real image I^1 and the synthetic image $I^{1\leftarrow 2}$ is defined as

$$L_{rgb} = \frac{1}{H \times W} \sum_{m,n}^{H,W} \left\| I^1_{m,n} - I^{1\leftarrow 2}_{m,n} \right\|_1, \tag{2}$$

where H and W are the spatial dimensions of the output scene coordinate map $\mathbf{Y}^1(\mathbf{w}, I^1)$.

3.3. Dense Deep Feature Reconstruction

Since the RGB values of an image are sensitive to change in the lighting condition, the consistency of the light/color intensity of a 3D point across two images cannot always be assured, especially in uncontrolled environments. There are also cases in both indoor and outdoor scenes where a large patch of the image is filled with same RGB value due to lack of texture on the objects and surfaces in the scene. Photometric reconstruction loss is only useful in regions where intensity gradient is large. Hence, a robust dense image feature, which contains more contextual information, can be used for dealing with these issues. In this work, we exploit the deep CNN features for dense matching proposed in [27].

While any dense visual descriptor such as dense SIFT may be suitable for the dense matching task, the learned deep visual descriptor in [27] is light-weight allowing for efficient training, and has been proven to be successful for dense monocular reconstruction. To extract the deep features for each pixel in the image, the whole image is passed into a fully convolutional neural network which is pre-trained using the method in [27] on the raw NYU-D v2 dataset [20]. A 32-dimensional feature map F with the same spatial dimensions as the input image is regressed by the network which can be subsequently used for dense image alignment. We then downsize it to one-eighth of the image size to match the scene coordinate map. Given the feature map F^2 regressed for I^2, we can warp it into I^1's frame of reference as follows,

$$F^{1\leftarrow 2} = f(\mathbf{Y}^1(\mathbf{w}, I^1), F^2, \mathbf{T}^2, \mathbf{K}). \tag{3}$$

Similar to Eq. (2), the deep dense feature reconstruction loss is defined as

$$L_{feat} = \frac{1}{H \times W} \sum_{m,n}^{H,W} \left\| F^1_{m,n} - F^{1\leftarrow 2}_{m,n} \right\|_1 . \tag{4}$$

3.4. 3D Smoothness Prior

The predicted scene coordinates from a single view image can be considered as the reconstruction of the scene. So far, the learning of our model for coordinate prediction only considers the input(image)-output(3D points) relationship. The correlation between the predicted 3D points is also important to recover the geometry of the scene. In particular, we utilize the intensity consistency within the image to constrain a smooth prediction in the coordinate map. A similar idea has been applied in [8, 9, 10, 28] in the depth estimation topic. We extend this mechanism to the 3D space.

The idea behind this smoothness prior is that a large 3D Euclidean distance between predicted neighbouring scene coordinates should be penalized if there is no image evidence to support this (*e.g.* if the image is uniform). Specifically, it is formulated as

$$L_s = \sum_{m,n}^{H,W} e^{-|\partial_x I_{m,n}|} \left\| \partial_x \mathbf{Y}_{m,n} \right\|_2 + e^{-|\partial_y I_{m,n}|} \left\| \partial_y \mathbf{Y}_{m,n} \right\|_2 , \tag{5}$$

where $\mathbf{Y}$ is the predicted coordinate map, $\partial_x(\cdot)$ and $\partial_y(\cdot)$ are the horizontal and vertical gradient operators.

3.5. Training Loss

Apart from the three losses previously mentioned, we also use the single view reprojection error of I_1 as the base loss to train our model, since the ground truth pose $\mathbf{T}_1$ is available. The reprojection error loss is defined as

$$L_{repro} = \frac{1}{H \times W} \sum_{m,n}^{H,W} \left\| P(\mathbf{Y}^1, \mathbf{T}^1, \mathbf{K}) - \mathbf{p}^1 \right\|_2 , \tag{6}$$

where $P(\cdot)$ is the projection function that projects a 3D point and computes its pixel position in the image plane. Note that this is the loss that DSAC++ used in the training of the second stage of their system.

Hence, the total loss that we use to train our model is

$$L = w_r L_{repro} + w_p L_{rgb} + w_f L_{feat} + w_s L_s, \tag{7}$$

where w_r, w_p, w_f and w_r are the loss weights hyper-parameters.

3.6. Single View Inference

Although our system is trained with image pairs, it only requires a single view image to perform inference. Once the model for scene coordinate prediction is trained, we can establish the dense correspondences between image pixel positions and the 3D points and then use RANSAC+PnP to estimate the pose of the camera.

Similar to DSAC++ [4], we first sample N sets of four 2D-3D correspondences using the predicted coordinate map (i.e. each sample contains four image points and corresponding 3D scene coordinates). After solving the PnP problems independently, a pool of N pose hypotheses is built for the best candidate selection. To rank the hypotheses, we compute the reprojection error map for each hypothesis using all predicted 3D coordinates. The best hypothesis is selected depending on the number of inliers, which is defined as the points whose reprojection error is less than a threshold τ. Finally, the best hypothesis is refined with updated inliers iteratively to produce the final pose estimate.

4. Experiments

4.1. Datasets

To verify the effectiveness of the multi-view photo or/and feature reconstruction loss and smoothness prior, we

	repro(baseline)	repro+rgb	repro+feat	repro+rgb+feat	w/ smooth
Chess	5.03cm, 1.36°	4.58cm, 1.56°	3.63cm, **1.21**°	**3.59cm**, 1.23°	**3.59cm**, 1.23°
	49.6%	54.7%	70.75%	**71.1%**	69.9%
Fire	9.41cm, 3.00°	7.94cm, 2.99°	5.28cm, 1.92°	**5.05cm**, **1.87**°	5.32cm, 2.04°
	26.8%	38.9%	47.9%	**49.75%**	48.25%
Heads	24.9cm, 10.0°	6.05cm, 3.67°	13.2cm, 7.45°	5.02cm, 3.22°	**4.99cm**, **2.98**°
	6.4%	46.5%	21.1%	48.9%	**50.1%**
Office	6.21cm, 1.43°	6.00cm, 1.48°	5.71cm, 1.35°	**5.62cm**, **1.34**°	5.71cm, 1.36°
	36.25%	39.28%	42.18%	**42.78%**	41.83%
Pumpkin	7.26cm, 1.77°	6.23cm, 1.60°	5.60cm, 1.48°	**5.56cm**, **1.47**°	5.58cm, 1.48°
	32%	36.8%	44.33%	**44.55%**	43.7%
Kitchen	11.0cm, 2.23°	**8.62cm**, 1.95°	9.07cm, 1.99°	8.67cm, **1.93**°	8.73cm, **1.93**°
	12.04%	22.54%	22.16%	23.32%	**23.48%**
Stairs	62.9cm, 11.6°	**35.6cm**, **6.94**°	35.6cm, 7.27°	35.9cm, 7.33°	36.0cm, 7.25°
	0.2%	0.3%	1.6%	1.5%	**1.9%**

Table 1. **The median pose errors and accuracy for *7Scene* of models using different losses.** The number ending with *cm* (resp. °) is the median translation (rotation) error for test set. The percentage is the proportion of test frames with both translation and rotation error is below (5*cm*, 5°). The overall performance of the model is significantly improved with the additional constraints provided by the multi-view consistency of the features.

apply the proposed method to two of the most widely used datasets for camera relocalization task: *7Scenes* [24] and *Cambridge Landmarks* [15]. *7Scenes* has 7 indoor scenes that are captured using a Kinect camera, provided with RGB-D images and ground truth poses. We only use the RGB images and ground truth poses to train our models. Note that the depth images can be very helpful to supervise the learning of the model (as shown in DSAC++ [4]), however our work focuses on the case when the 3D model of the scene is not available. Hence we omit the ground truth depth in our experiments.

Training images are selected following the official split of the datasets. Since the images are taken from a monocular camera, it is important to select proper target frames for each training image to enable multi-view geometry based supervision. To that end, we randomly select 3 images from its nearest [-100, +100] neighbours as the pair candidates (thanks to to the fact that the images are from a continuous sequence). Then we use an off-the-self optical flow estimation method (FlowNet2.0 [11] and its implementation [21]) to compute the overlap between the current frame and its pair candidates. We choose the candidate as the final pair image if the ratio of their overlap area to the image spatial area is within the range of [0.4, 0.9]. On average, a training image has $\sim$2 pairs to build multi-view constraints. We also use the overlap as the mask to zero out the meaningless reconstruction loss caused by the pixels that are projected out of frame on the target images.

4.2. Training and Test Regime

We use a two-stage scheme for our training pipeline. Firstly, we train the model with the heuristic suggested in DSAC++ [4] since only RGB images are used for training, which means the actual scales of the scenes are missing. This heuristic assumes a constant distance between the camera plane and the scene surface for every image. The distance is set to 3m and 10m for *7Scenes* and *Cambridge Landmarks* respectively, which are the approximate scales of the indoor and outdoor scenes.

We apply our proposed multi-view geometry based losses in the second stage of training, which is initialized by the model from the previous heuristic. To conduct a detailed ablation study, we train the model with five different combinations of losses for the *7Scenes* respectively:

1. `repro`: the model is trained with only single view reprojection loss Eq. (6). This is our baseline model.
2. `repro+rgb`: the model is trained with photometric reconstruction loss Eq. (1) along with `repro`.
3. `repro+feat`: the model is trained with deep feature reconstruction loss Eq. (3) along with `repro`.
4. `repro+rgb+feat`: the model is trained with photometric reconstruction loss Eq. (1) *and* deep feature reconstruction loss Eq. (3) along with `repro`.
5. `w/ smooth`: the smoothness prior Eq. (5) is added to `repro+rgb+feat`.

All five models are optimized in an end-to-end fashion with ADAM [16] for *30k* iterations in total. The initial learning rate is set to 1e-4 and decreased to half at 10k step and the next every 5k step. The training samples for `repro` model are also pairs of images to ensure an identical training environment. The hyper-parameters in (7) are *not* highly tuned and are kept identical between scenes.

We use the PnP solver plus RANSAC to estimate the 6D pose for the test images after predicting the scene coordinate from these 5 trained models. For RANSAC, $N = 256$ pose hypotheses are generated as the pool, and the reprojection

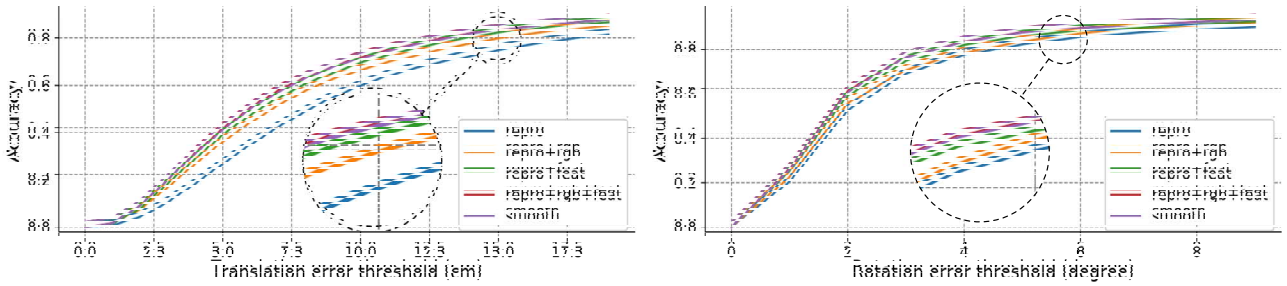

Figure 3. **Localization accuracy of position and orientation as a cumulated histogram of errors.** The horizontal axis is the threshold for transnational error (left, in cm) and rotational error (right, in degree). The vertical axis is the proportion of the test images of which transnational or rotational error is smaller than the thresholds on the horizontal axis.

error threshold τ is set to 10 pixels for inliers selection. The final pose refinement step runs up to 100 iterations.

4.3. Results Analysis

Table 1 and Fig. 3 shows the pose estimation performance of the models trained with different combinations of the losses introduced in previous sections. The pose for a test image is considered as *correct* if the pose error is below 5cm and 5°.

Multi-view vs. Single-view. One can see from Table 1 that the addition of photometric loss supervised by multi-view constraint in training *always* improves the accuracy of the estimated pose than purely training with single-view reprojection loss (Column `repro` and `repro+rgb`). The deep feature reconstruction loss also helps the reprojection loss and the effect is even more obvious generally (Column `repro` and `repro+feat`), due to the more informative (both fine and course) features that are extracted from a deep model, especially when the scene contains textrueless regions. The accuracy of the pose estimation is furthermore, slightly though, improved by using the photometric and feature reconstruction loss together as the additional supervision for the coordinate regression model.

The reason behind this gain of pose estimation performance is that the model predicts more accurate scene coordinates if it is supervised with multi-view constraints during training. We show one set of 4 points used by hypothesis generation in PnP algorithm for a test image in Fig. 1. The predicted 3D points for the left image are projected to the right image using the ground truth pose to show the quality of these points. The projections of points from the model with reconstruction loss on the right image are closer to the pixels that share the similar surrounding pattern on the left one, compared to the model trained with only single view reprojection loss. This behavior affirms the usage of photometric/feature reconstruction consistency in the training.

Smoothness prior. The best pose estimation performance of scene `heads` comes from the model trained with all components of the final loss, which suggests the best 3D reconstruction. See Fig. 4 for visualization. As can be seen,

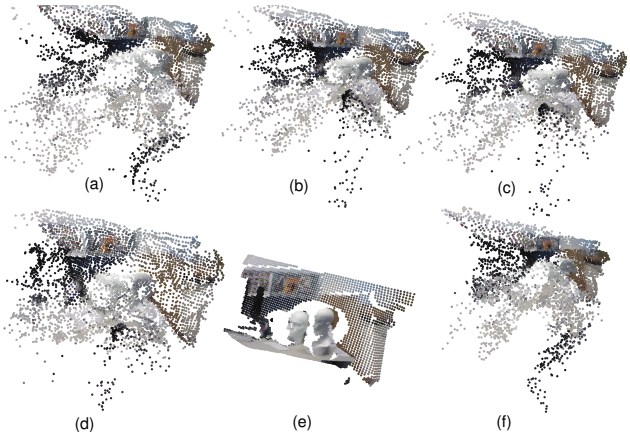

Figure 4. **Reconstructed point clouds of one sample image from the test set of scene `heads` using different models.** We visualize the point cloud reconstruction from our model trained with (**a**) `repro`, (**b**) `repro+rgb`, (**c**) `repro+rgb+feat`, (**d**) `w/ smooth`. The ground truth point cloud and the reconstruction from DSAC++ [4] is showed in (**e**) and (**f**) respectively. All point clouds are visualized from the same viewpoint.

the point cloud reconstructed from the models trained with the first three (a, b c) losses are not visually good enough to recover the actual geometry of the scene (e). In this case, the smoothness prior (d) helps to produce an improved model for the 3D reconstruction, especially by reducing the noise in the bottom part of the point cloud. As for other scenes, we found the usefulness of the smoothness prior is limited (Column `repro+rgb+feat` and `w/ smooth` in Table 1). When the pose estimate is accurate enough from the model trained without the smoothness prior, for instance in scenes `fire` and `office`, the use of the smoothness penalty does not help, with the effect even being negative. We presume the negative effect is caused by the smoothness loss pushing inlier 3D scene points towards the outliers that would otherwise have been ignored by RANSAC.

4.4. Comparison with Single-view Based Work

To establish a fair comparison between our model and other work, we increase the training iteration number of the

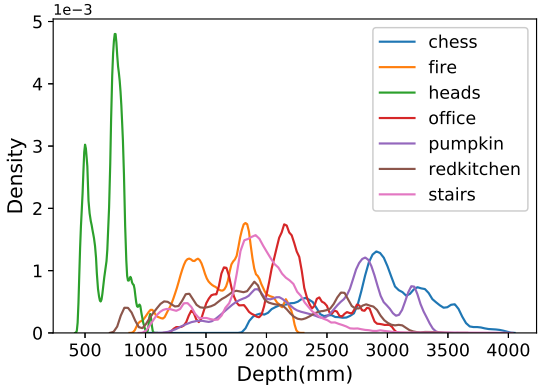

Figure 5. **The distribution of depth value of *7Scenes***. We randomly select 10 depth images from the training set of each scene, and show the distributions of all the valid depth values of them. One can see that the depth distribution of scene `heads` has the mean value around 0.7m, which does not follow the distributions of other scenes.

model `repro+rgb+feat` to 300k (which is used in [4]), likewise for the steps for learning rate decay. Table 2 shows the results of this model for *7Scenes* and *Cambridge Landmarks* [15]. Except for the relatively poor performance in the `stairs` scene due to the self-similarity of the RGB images, our method achieves a consistently good result for all of the indoor scenes. The percentage of the correct test frames of all scenes in *7Scenes* of our model is 70.1%, compared to 76.1% and 60.4% of DSAC++ [4]'s model that is trained with and without ground truth scene coordinate respectively. The gap of training without and with the 3D model of the scene is closed by our method.

The conclusions from previous ablation study Table 1 and this Table 2 together are: 1) our losses help the coordinate regression model converge faster than the single-view baseline (Table 1), and 2) when converged it performs better than the state-of-the-art single-view method [4] (Table 2).

A noticeable point is that the median error of scene `heads` is relatively large for DSAC++ [4] compared to other indoor scenes when trained without 3D model, which is 12cm and 6.7°. We observe that this is presumably due to the misused heuristic for scene `heads`, which assumes a constant distance between the image plane and the scene surface for every frame that is used to initialize the model in the first stage of training. To support our hypothesis, we plot the distributions of the ground truth depth samples from training images of each scene in Fig. 5. The heuristic constant distance we (as well as DSAC++ [4]) used for *7Scenes* is 3m, which properly simulate the substantial depth of most of the scenes, except for `heads`, whose true depth is around 0.7m. We therefore train another model for `heads` with the heuristic set to 0.7m. The result of the new model is increased to 0.02m and 1.3°. This backs our speculation. Nonetheless, our training scheme eliminates the negative effect of the inappropriate heuristic, and achieves better reconstruction when the poor prior is applied to both our method and [4] (we still use 3m for *7Scenes* as the approximate depth for the first stage in our experiment). From this standpoint, our method based on multi-view consistency reduces the dependence on the initialization of the model.

	DSAC++ [4]		ours
Scene	w/ 3D	w/o 3D	
Chess	0.02m, 0.5°	**0.02m**, **0.7°**	**0.02m**, 0.8°
Fire	0.02m, 0.9°	0.03m, 1.1°	**0.02m**, **1.0°**
Heads	0.01m, 0.8°	0.12m, 6.7°	**0.04m**, **2.7°**
Office	0.03m, 0.7°	**0.03m**, **0.8°**	**0.03m**, **0.8°**
Pumpkin	0.04m, 1.1°	0.05m, **1.1°**	**0.04m**, **1.1°**
Kitchen	0.04m, 1.1°	0.05m, 1.3°	**0.04m**, **1.1°**
Stairs	0.09m, 2.6°	0.29m, 5.1°	**0.18m**, **3.9°**
Acc.	76.1%	60.4%	**70.1%**
K. Col.	0.18m, 0.3°	0.23m, 0.4°	**0.20m**, **0.3°**
Old Hos.	0.20m, 0.3°	0.24m, 0.5°	**0.19m**, **0.4°**
Shop Fac.	0.06m, 0.3°	0.09m, 0.4°	**0.07m**, **0.3°**
St M. Ch.	0.13m, 0.4°	**0.20m**, 0.7°	**0.20m**, **0.6°**
G. Court	0.40m, 0.2°	0.66m, **0.4°**	**0.62m**, **0.4°**

Table 2. **Comparison between our method and DSAC++ [4].** The gap between model trained with and without is closed using our multi-view geometry-based training method. Numbers are boldened only among the w/o 3D methods.

Since the performance of pose estimation heavily relies on the quality of the scene coordinate prediction, we also show the quantitative comparison of the scene coordinate regressed by our model trained with multi-view constrains (`repro+rgb+feat`) and the single-view method [4] in Table 3. This shows that our model predicts more accurate scene coordinates for the geometrical task.

*For all test images in *7Scenes*	DSAC++[4]	Ours
Average No. of inliers per image	245	**319**

Table 3. We project the predicted scene coordinates from the models in DSAC++[4] and ours using ground truth poses. The reprojection error threshold for inlier is set to 2 pixel.

5. Conclusion

We have proposed an efficient learning method for scene coordinate regression to carry out accurate 6DoF camera relocalization in a known scene from a single RGB image. Our learning method explicitly enforces multi-view geometric constraints to learn the regression model in a self-supervised manner in the absence of the ground truth 3D model. The constraints imposed by our proposed loss improve the efficiency of training. Additionally, the regression model learned via our method allows for more reliable 2D-3D correspondences which in turn lead to consistent and accurate camera relocalization performance.

References

[1] Vassileios Balntas, Shuda Li, and Victor Prisacariu. RelocNet: Continuous Metric Learning Relocalisation using Neural Nets. In *ECCV*, 2018.

[2] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. DSAC - Differentiable RANSAC for Camera Localization. In *CVPR*, 2017.

[3] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. Uncertainty-Driven 6D Pose Estimation of Objects and Scenes from a Single RGB Image. In *CVPR*, 2016.

[4] Eric Brachmann and Carsten Rother. Learning Less Is More - 6D Camera Localization via 3D Surface Regression. In *CVPR*, 2018.

[5] Mai Bui, Shadi Albarqouni, Slobodan Ilic, and Nassir Navab. Scene Coordinate and Correspondence Learning for Image-Based Localization. In *BMVC*, 2018.

[6] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 1981.

[7] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete Solution Classification for the Perspective-three-point Problem. *TPAMI*, 2003.

[8] Ravi Garg, Vijay Kumar B.G., Gustavo Carneiro, and Ian" Reid. Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue. In *ECCV*, 2016.

[9] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.

[10] Philipp Heise, Sebastian Klose, Brian Jensen, and Alois Knoll. Pm-huber: Patchmatch with huber regularization for stereo matching. In *ICCV*, 2013.

[11] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, Jul 2017.

[12] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *NIPS*, 2015.

[13] Alex Kendall and Roberto Cipolla. Modelling Uncertainty in Deep Learning for Camera Relocalization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

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

[15] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. In *ICCV*, 2015.

[16] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *CoRR*, 2014.

[17] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o(n) solution to the pnp problem. *IJCV*.

[18] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. In *CVPR*, 2015.

[19] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov 2004.

[20] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor Segmentation and Support Inference from RGBD Images. In *ECCV*, 2012.

[21] Fitsum Reda, Robert Pottorff, Jon Barker, and Bryan Catanzaro. flownet2-pytorch: Pytorch implementation of flownet 2.0: Evolution of optical flow estimation with deep networks. `https://github.com/NVIDIA/flownet2-pytorch`, 2017.

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

[23] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *TPAMI*, 39(9):1744–1756, 2017.

[24] J Shotton, B Glocker, C Zach, S Izadi, A Criminisi, and A Fitzgibbon. Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images. In *CVPR*, 2013.

[25] Julien Valentin, Matthias Nießner, Jamie Shotton, Andrew Fitzgibbon, and Philip Torr. Exploiting Uncertainty in Regression Forests for Accurate Camera Relocalization. In *CVPR*, 2015.

[26] Florian Walch, Caner Hazirbas, Laura Leal-Taixé, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization using LSTMs for structured feature correlation. In *ICCV*, 2017.

[27] Chamara Saroj Weerasekera, Ravi Garg, and Ian D. Reid. Learning deeply supervised visual descriptors for dense monocular reconstruction. *CoRR*, 2017.

[28] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised Learning of Monocular Depth Estimation and Visual Odometry with Deep Feature Reconstruction. In *CVPR*, 2018.

[29] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017.