Deep Learning for Camera Pose Estimation: A Comparative Study on 7-Scenes 3D Reconstruction Final Project 2025 Group 15

Binheng Zheng
4746998
Scientific Computing

Yuefeiyang Li
4753723
Scientific Computing

Abstract

We present a comparative study of a deep regression model (PoseNet) and a classical geometric baseline (PnP+RANSAC) for camera pose estimation on the 7-Scenes dataset. PoseNet, trained on the Chess *scene, demonstrates reasonable accuracy in-domain but shows poor generalization to unseen scenes such as* Fire *and* Heads*. In contrast, PnP+RANSAC achieves near-perfect accuracy within individual sequences, underscoring the strength of explicit geometric reasoning. An ablation study on the loss weight β further highlights its role in balancing translation and rotation errors. Together, these findings indicate the limited generalization ability of learning-based regression and the robustness of geometric methods for indoor camera relocalization.*

Keywords: Camera pose estimation, Deep learning, PoseNet, PnP+RANSAC, 7-Scenes dataset

1. Introduction

Camera pose estimation, the task of predicting a camera's six degrees-of-freedom (6-DoF) position and orientation relative to a scene, is a fundamental problem in computer vision. It underpins robotics, autonomous navigation, augmented reality (AR), and simultaneous localization and mapping (SLAM) [7]. Accurate and efficient pose estimation enables robots to localize in new environments, AR systems to anchor virtual content in the real world, and mapping pipelines to build consistent 3D reconstructions. A persistent challenge in this field is achieving both high accuracy and robust generalization across visually diverse environments, yet systematic comparisons between learning-based regression and geometric baselines under such conditions remain limited.

Traditional approaches are geometry-based. Given 2D–3D correspondences between image pixels and scene points, the camera pose can be recovered via the Perspective-n-Point (PnP) algorithm, often embedded in a RANSAC loop for robustness [12, 6]. These methods achieve high accuracy when sufficient textured features are present and yield interpretable solutions grounded in projective geometry. However, their reliance on reliable correspondences makes them brittle in indoor relocalization scenarios, where textureless regions, repetitive patterns, and motion blur can make feature detection and matching unreliable.

Deep learning methods attempt to bypass explicit correspondence search. PoseNet [10] introduced the idea of directly regressing camera translation and rotation from a single RGB image using a convolutional neural network. This demonstrated that global scene appearance cues can serve as a surrogate for local features. Extensions such as Bayesian PoseNet [8] and uncertainty-weighted losses [9] improved stability and accuracy. Nevertheless, while these methods demonstrate the feasibility of end-to-end pose regression, they often fail to match the accuracy of geometry-based pipelines and struggle to generalize across unseen environments, which is precisely the setting we study in this work. Hybrid methods, such as DSAC [1], attempt to bridge the gap by embedding RANSAC into a differentiable pipeline, but they demand dense supervision and higher computational resources.

In this work, we provide a focused comparison between learning-based regression and classical geometry-based baselines in the context of indoor relocalization. We evaluate on the 7-Scenes dataset [20], a widely used benchmark characterized by perceptual aliasing, textureless surfaces, and viewpoint variability. PoseNet is trained on the *chess* scene and tested on unseen scenes to assess generalization, while PnP+RANSAC is applied within sequence-consistent settings to assess the upper bound of explicit geometry. By combining quantitative evaluation (translation and rotation errors) with qualitative analysis (error distributions and trajectory visualizations), we clarify where deep regression is competitive, where geometry remains dominant, and how future methods may combine their complementary strengths.

Overall, this study contributes a systematic compara-

tive evaluation of PoseNet and PnP+RANSAC on 7-Scenes, highlighting the trade-off between accuracy and generalization in camera pose estimation. This provides a reference point for future hybrid methods that aim to combine the scalability of learning-based models with the robustness of geometric reasoning.

2. Related Work

Camera pose estimation is a long-standing problem in computer vision, with applications in robotics, augmented reality, and simultaneous localization and mapping (SLAM). Existing approaches can be broadly categorized into geometry-based pipelines, learning-based regression models, and hybrid methods.

2.1. Geometry-based Approaches

The classical pipeline relies on local feature detection and matching. Descriptors such as SIFT [13] or ORB [16] are used to establish 2D–3D correspondences, and the camera pose is recovered via Perspective-n-Point (PnP) [12], typically embedded in RANSAC [6] for robustness. These methods achieve high accuracy when reliable features are available, but performance degrades in low-texture or repetitive indoor scenes. Beyond single-frame relocalization, structure-from-motion (SfM) and visual SLAM pipelines such as ORB-SLAM [14] and COLMAP [18] exploit multi-view consistency to achieve high accuracy, but they remain brittle under perceptual aliasing and motion blur. For the 7-Scenes dataset specifically, structure-based methods such as Scene Coordinate Regression Forests [20] predict dense scene coordinates and then solve PnP, achieving strong results. More recently, learned local features and matchers (e.g., SuperPoint [4], SuperGlue [17]) have been adopted to improve robustness.

2.2. Learning-based Regression Models

Kendall *et al*. introduced PoseNet [10], a convolutional neural network that directly regresses the 6-DoF camera pose (translation and quaternion rotation) from a single RGB image. This demonstrated the feasibility of end-to-end pose regression, but accuracy was generally lower than geometric methods. Subsequent extensions improved PoseNet by modeling predictive uncertainty (Bayesian PoseNet [8]) and introducing uncertainty-weighted loss functions [9]. Other works explored temporal modeling, such as VidLoc [3] and MapNet [11], which incorporated sequence-level consistency. More recently, attention-based and Transformer architectures (e.g., CameraPoseTransformer [19]) have been explored to enhance feature aggregation. Despite these advances, regression models still struggle to generalize across unseen scenes, particularly in complex indoor datasets like 7-Scenes.

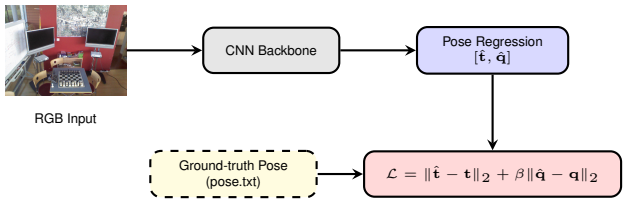

Figure 1. PoseNet pipeline: an RGB image is mapped to pose parameters via CNN regression. The predicted pose is compared against ground-truth poses (from dataset `pose.txt`) to compute the joint loss in Eq. 4.

2.3. Hybrid Methods

To combine the strengths of learning and geometry, Brachmann *et al*. proposed **DSAC** (Differentiable RANSAC) [1], where a neural network predicts dense scene coordinates and PnP with RANSAC is embedded into a differentiable pipeline. This enables end-to-end training while retaining geometric robustness. Follow-up work, often referred to as **DSAC++**, improved accuracy and efficiency on 7-Scenes and other benchmarks. Importantly, DSAC++ was not published as an independent CVPR 2019 paper; rather, it corresponds to the method described in *Learning Less is More – 6D Camera Localization via 3D Surface Regression* (CVPR 2018) [2], where the official code release explicitly names the method DSAC++. Beyond dense coordinate regression, hybrid approaches also exploit learned local features (e.g., D2-Net [5], R2D2 [15]) to improve the robustness of geometric pipelines.

3. Method

In this section, we present the two approaches compared in our study. We first introduce the PoseNet learning formulation (Sec. 3.1), followed by the evaluation metrics (Sec. 3.2). Finally, we describe the classical geometric baseline, PnP with RANSAC (Sec. 3.3).

3.1. PoseNet Learning Formulation

Given an input image I, PoseNet directly regresses the camera pose, consisting of a translation vector $\mathbf{t} \in \mathbb{R}^3$ and a rotation quaternion $\mathbf{q} \in \mathbb{R}^4$ (see Fig. 1):

$$f_\theta(I) = [\hat{\mathbf{t}}, \hat{\mathbf{q}}], \tag{1}$$

where f_θ denotes a convolutional neural network with parameters θ. To ensure a valid rotation, the predicted quaternion is normalized:

$$\hat{\mathbf{q}} \leftarrow \frac{\hat{\mathbf{q}}}{\|\hat{\mathbf{q}}\|}. \tag{2}$$

The predicted pose can equivalently be represented as a 4×4 homogeneous transformation matrix:

$$T = \begin{bmatrix} R(\hat{\mathbf{q}}) & \hat{\mathbf{t}} \\ 0 & 1 \end{bmatrix}, \tag{3}$$

where $R(\hat{\mathbf{q}})$ is the rotation matrix corresponding to the normalized quaternion.

The loss function jointly optimizes translation and rotation:

$$\mathcal{L} = \|\hat{\mathbf{t}} - \mathbf{t}\|_2 + \beta \cdot \|\hat{\mathbf{q}} - \mathbf{q}\|_2, \tag{4}$$

where β balances the different units of translation (meters) and rotation (radians). Following [10], we fix $\beta = 120$ in our experiments, though later works explore uncertainty-based weighting [9].

3.2. Pose Error Metrics

For evaluation, we follow standard pose regression metrics used in camera relocalization [10, 20]. The translation error is defined as the Euclidean distance between predicted and ground-truth positions:

$$e_t = \|\hat{\mathbf{t}} - \mathbf{t}\|_2, \tag{5}$$

measured in meters. The rotation error is computed as the angular distance between predicted and ground-truth quaternions:

$$e_r = 2 \cdot \arccos(|\langle \hat{\mathbf{q}}, \mathbf{q} \rangle|), \tag{6}$$

where $\langle \cdot, \cdot \rangle$ denotes the quaternion inner product. Equivalently, given the predicted rotation $\hat{R}$ and ground truth R, the error can also be expressed as

$$e_r = \arccos\left(\frac{\mathrm{trace}(R^\top \hat{R}) - 1}{2}\right). \tag{7}$$

This quantity corresponds to the geodesic distance on the rotation manifold, measured in degrees. We report both mean and median errors across the test set.

3.3. Geometric Baseline: PnP with RANSAC

Given a set of 2D–3D correspondences $(\mathbf{X}_i, \mathbf{p}_i)$, the perspective projection model is expressed as

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} R & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \tag{8}$$

where (u, v) are pixel coordinates, (X, Y, Z) are 3D world points, K is the intrinsic calibration matrix, and $(R, \mathbf{t})$ denotes the camera pose. The overall baseline pipeline is illustrated in Fig. 2.

PnP can be formulated as a nonlinear least-squares problem:

$$\min_{R,\mathbf{t}} \sum_{i=1}^{N} \|\mathbf{p}_i - \pi(K(R\mathbf{X}_i + \mathbf{t}))\|_2^2, \tag{9}$$

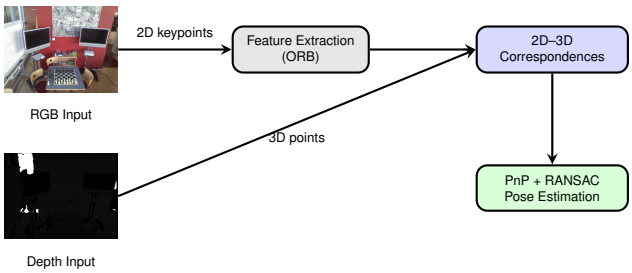

Figure 2. PnP+RANSAC baseline: RGB provides 2D keypoints, depth provides 3D points, and correspondences are used for robust pose estimation.

RANSAC repeatedly samples minimal sets of four correspondences, solves PnP, and evaluates the reprojection error

$$e_{\text{reproj},i} = \|\mathbf{p}_{\text{obs},i} - \mathbf{p}_{\text{proj},i}\|_2, \tag{10}$$

selecting the pose hypothesis with the maximum number of inliers:

$$\hat{R}, \hat{\mathbf{t}} = \arg\max_{R,\mathbf{t}} |\{i \mid e_{\text{reproj},i} < \tau\}|, \tag{11}$$

Since OpenCV's PnP solver estimates the world-to-camera transformation, we invert it to obtain the camera-to-world pose, ensuring consistency with the ground-truth format provided in the 7-Scenes dataset.

4. Experiments

In this section, we present our experimental setup, training details, evaluation metrics, and results. We compare PoseNet and a PnP+RANSAC baseline on the 7-Scenes dataset [20].

4.1. Dataset and Experimental Setup

We evaluate our methods on the 7-Scenes dataset, which contains RGB-D video sequences of seven indoor environments, each with ground-truth 6-DoF camera poses obtained from KinectFusion. The dataset presents challenges such as perceptual aliasing, textureless surfaces, and motion blur, making it a standard benchmark for camera relocalization.

For simplicity, we select three representative scenes. PoseNet is trained on the *chess* scene and evaluated for generalization on two unseen scenes, *fire* and *heads*. As a geometric baseline, we implement a PnP+RANSAC pipeline using ORB features. In particular, we extract 4000 keypoints per image and employ a ResNet-18 global descriptor for image retrieval, selecting the top-10 nearest neighbors before establishing 2D–3D correspondences. Camera poses are then estimated by solving PnP inside a RANSAC loop.

Since the sequences of the 7-Scenes dataset are reconstructed independently and are not aligned to a common

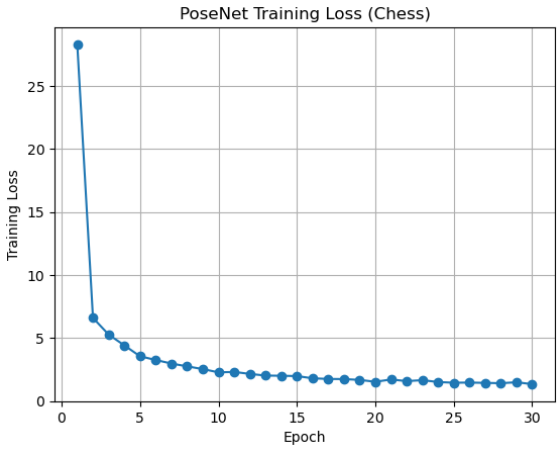

Figure 3. Training loss curve of PoseNet on the *Chess* scene. The model converges after approximately 20 epochs.

global coordinate system, we restrict the PnP+RANSAC evaluation to within-sequence retrieval. For each scene, we first compute per-sequence results and then report scene-level performance by averaging across all test sequences, so that each sequence contributes equally regardless of its length:

$$E_{\text{scene}} = \frac{1}{N} \sum_{i=1}^{N} E_{\text{seq}_i}, \tag{12}$$

where E_{seq_i} denotes the error statistic (mean or median) computed on sequence i, and N is the total number of test sequences in the scene.

4.2. PoseNet Training Details

PoseNet is implemented with a ResNet-34 backbone pre-trained on ImageNet. The final fully connected layer is modified to regress a 7-dimensional vector, consisting of translation $\hat{\mathbf{t}} \in \mathbb{R}^3$ and rotation quaternion $\hat{\mathbf{q}} \in \mathbb{R}^4$. During training, the quaternion is normalized to unit length. The model is trained for 30 epochs with the AdamW optimizer, a learning rate of 1×10^{-4}, weight decay of 1×10^{-4}, and batch size of 32. The loss function used for training is defined in Eq. 4. Figure 3 shows the training loss curve on the *Chess* scene, which decreases smoothly and converges after about 20 epochs, demonstrating stable optimization.

4.3. Quantitative Results

We evaluate performance using the translation and rotation errors defined in Sec. 3.2 (Eq. 5, Eq. 6). For each scene, we report both mean and median errors over the test split. For PoseNet, errors are computed directly across all test frames, while for PnP+RANSAC, results are averaged per-sequence as described in Sec. 4.1.

Table 1 summarizes the results of PoseNet and the PnP+RANSAC baseline on three scenes. PoseNet achieves

	Chess	Fire	Heads
PoseNet			
e_t mean (m)	0.198	0.933	0.807
e_t median (m)	0.151	0.920	0.769
e_r mean (°)	5.40	39.99	47.88
e_r median (°)	4.55	35.66	46.98
PnP+RANSAC			
e_t mean (m)	0.005	0.005	0.004
e_t median (m)	0.004	0.004	0.003
e_r mean (°)	0.211	0.206	0.243
e_r median (°)	0.143	0.153	0.168

Table 1. Comparison of PoseNet and PnP+RANSAC on three scenes of the 7-Scenes dataset. We report mean and median translation error (m) and rotation error (degrees). PnP+RANSAC results are averaged over per-sequence statistics within each scene.

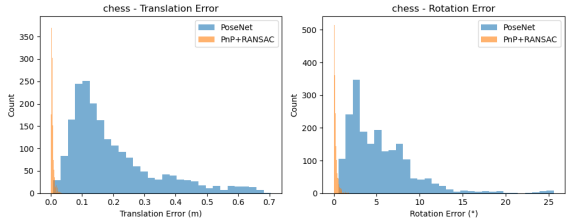

Figure 4. Distribution of translation and rotation errors for the chess scene. PoseNet exhibits long-tailed error distributions, while PnP+RANSAC is tightly clustered near small values.

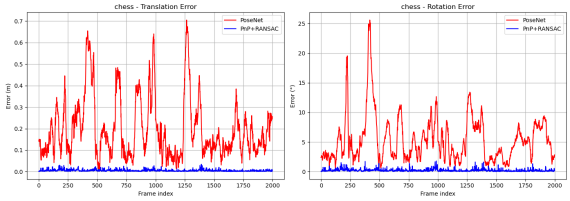

Figure 5. Frame-wise translation errors on the chess scene. PoseNet errors gradually drift as the sequence progresses, while PnP+RANSAC remains stable across frames.

reasonable accuracy on the training scene (*chess*), with median translation error $\approx$ 0.15 m and rotation error $\approx$ 5°. On unseen scenes (*fire*, *heads*), however, the errors increase substantially, reflecting limited generalization. In contrast, the geometric baseline consistently outperforms PoseNet in both translation and rotation accuracy, owing to explicit 2D–3D reasoning.

Based on these results, the following qualitative analysis focuses on the *chess* scene, where PoseNet achieves meaningful accuracy for visual comparison with PnP+RANSAC.

4.4. Qualitative Analysis

To complement the quantitative results, we further analyze performance on the *chess* scene through error distributions (Fig. 4), frame-wise error trends (Fig. 5), and trajec-

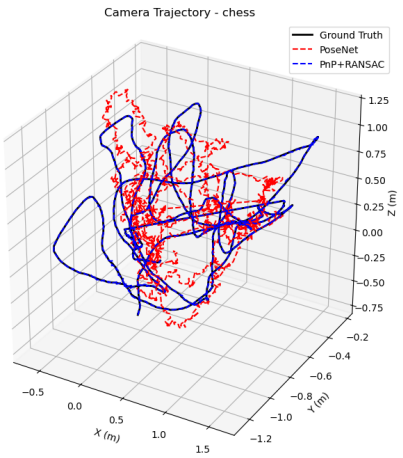

Figure 6. Camera trajectories on the chess scene. PoseNet predictions drift away from the ground truth, while PnP+RANSAC follows the true path more closely.

tory visualization (Fig. 6). We restrict the qualitative study to *chess* since PoseNet achieves reasonable accuracy only on this scene, while performance on unseen scenes is too poor for meaningful visual comparison.

Histograms (Fig. 4) show that PoseNet errors are broadly distributed, with long tails in both translation and rotation. In contrast, PnP+RANSAC errors are tightly clustered near small values, indicating higher robustness.

Frame-wise error curves (Fig. 5) provide a temporal perspective. Translation errors for PoseNet fluctuate between 0.1–0.7 m and occasionally spike above 0.6 m, while rotation errors reach up to 25°. In comparison, PnP+RANSAC maintains consistently low errors across the entire sequence, typically below 0.05 m for translation and 2° for rotation. This illustrates that PoseNet accumulates error as the sequence progresses, whereas the geometric baseline remains stable.

Trajectory visualization (Fig. 6) confirms these findings: PoseNet predictions drift and misalign with the ground-truth path, leading to significant deviations in 3D space. PnP+RANSAC, on the other hand, closely follows the true trajectory, demonstrating the advantage of explicit 2D–3D reasoning.

Overall, these qualitative results highlight the trade-off between learned regression, which provides fast inference but suffers from weaker generalization and stability, and geometric reasoning, which is computationally heavier but highly accurate.

5. Ablation Study

We perform an ablation to analyze the effect of the loss balancing weight β in Eq. 4, which controls the trade-off between translation and rotation errors in PoseNet. Since PoseNet shows limited generalization to unseen scenes (Sec. 4.2), we evaluate β on two transfer settings, Chess→Fire and Chess→Heads.

Fire (β)	Translation (m)		Rotation (°)	
	mean	median	mean	median
50.0	0.936	0.948	38.76	33.76
120.0	**0.933**	**0.920**	**39.99**	**35.67**
250.0	0.981	0.957	40.74	36.28
Heads (β)	Translation (m)		Rotation (°)	
	mean	median	mean	median
50.0	0.879	0.845	47.72	46.52
120.0	**0.807**	**0.769**	**47.88**	**46.98**
250.0	0.816	0.775	47.73	46.40

Table 2. Ablation on the loss weight β in Eq. 4, evaluated on Chess→Fire and Chess→Heads. Bold indicates the chosen operating point ($\beta = 120$) as the most balanced trade-off across transfer scenes.

β	Best training loss	Epochs to convergence
50	0.586	∼25
120	1.344	∼28
250	2.696	∼30

Table 3. Convergence statistics for different β, measured on the training scene (Chess). Reported are the best training loss and approximate epochs to convergence.

Table 2 reports both mean and median errors. For *Fire*, $\beta = 50$ yields slightly lower translation error, but rotation accuracy degrades substantially. For *Heads*, $\beta = 120$ outperforms $\beta = 50$ and $\beta = 250$ consistently across translation and rotation metrics. Taken together, $\beta = 120$ provides the most balanced trade-off across transfer scenes. Importantly, varying β has little influence on the in-domain *Chess* performance, indicating that its primary role is in adjusting error balance rather than improving generalization.

To examine optimization dynamics, Table 3 summarizes the best training loss and approximate number of epochs to convergence on the *Chess* training scene. Larger β values (e.g., $\beta = 250$) converge slower and to higher final loss, while smaller values converge faster. Nevertheless, all settings converge stably and achieve similar in-domain accuracy, confirming that β mainly re-weights translation versus rotation objectives without altering overall fitting capacity. Notably, a smaller training loss does not necessarily imply improved cross-scene performance.

Finally, Fig. 7 visualizes the effect of β on median errors. Both Fire and Heads exhibit consistent behavior: smaller β reduces translation error but increases rotation error, while larger β favors rotation at the cost of translation. This confirms $\beta = 120$ as a stable compromise. However, the persistent gap across unseen scenes indicates that tuning β alone cannot resolve the generalization challenge of PoseNet.

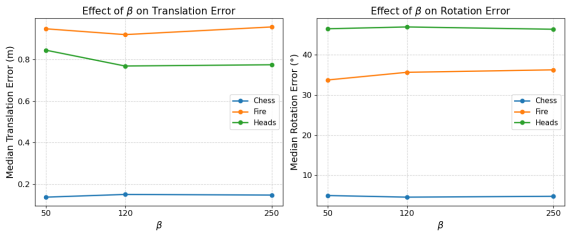

Figure 7. Effect of the loss weight β on translation and rotation errors for Chess→Fire and Chess→Heads. Curves show median errors for each setting.

6. Conclusion

We conducted a comparative study of learning-based and geometry-based approaches for camera pose estimation on the 7-Scenes dataset. PoseNet, trained on the *chess* scene, reached reasonable accuracy in-domain (median error $\approx$ 0.15 m / 5°), but its performance degraded severely on unseen scenes (*fire*, *heads*), highlighting the difficulty of cross-scene generalization. In contrast, the PnP+RANSAC baseline consistently achieved near-perfect accuracy within individual sequences, confirming the strength of explicit 2D–3D reasoning, although its reliance on per-sequence reconstructions limits global applicability.

Beyond the main comparison, we analyzed the effect of the loss balancing weight β. The ablation study showed that β controls a clear trade-off: smaller values reduce translation error at the cost of rotation accuracy, while larger values favor rotation but degrade translation. Across both Chess→Fire and Chess→Heads, $\beta = 120$ provided the most balanced compromise. Training loss curves further revealed that larger β slowed convergence, yet scene-internal accuracy remained largely unaffected. This suggests that β modulates the balance of error components rather than overall fitting capacity. Ultimately, PoseNet's fundamental limitation lies in its weak cross-scene generalization, which no adjustment of β could fully resolve.

Our findings underline the complementary strengths and weaknesses of deep regression and geometric pipelines. PoseNet enables fast inference and flexibility but lacks robustness across environments, whereas PnP+RANSAC is highly accurate under consistent geometry but unsuitable for global relocalization. Future research could combine the advantages of both, for example through differentiable RANSAC or uncertainty-aware loss functions, to achieve models that are both accurate and generalizable in indoor camera pose estimation.

References

[1] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac - differentiable ransac for camera localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6684–6692, 2017. 1, 2

[2] Eric Brachmann and Carsten Rother. Learning less is more – 6d camera localization via 3d surface regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4654–4662, 2018. 2

[3] Ronald Clark, Sen Wang, Andrew Markham, Niki Trigoni, and Hongkai Wen. Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2652–2660, 2017. 2

[4] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 224–236, 2018. 2

[5] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint detection and description of local features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8092–8101, 2019. 2

[6] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In *Communications of the ACM*, pages 381–395, 1981. 1, 2

[7] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. 1

[8] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 241–249, 2016. 1, 2

[9] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5974–5983, 2017. 1, 2, 3

[10] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2938–2946, 2015. 1, 2, 3

[11] Zubair Laskar, Iaroslav Melekhov, Shubham Kalia, and Juho Kannala. Mapnet: Geometry-aware learning of maps for camera localization. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 261–269, 2018. 2

[12] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o(n) solution to the pnp problem. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2009. 1, 2

[13] David G. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision (IJCV)*, volume 60, pages 91–110. Springer, 2004. 2

[14] Raul Mur-Artal, JMM Montiel, and Juan D. Tardos. Orb-slam: a versatile and accurate monocular slam system. In *IEEE Transactions on Robotics (T-RO)*, volume 31, pages 1147–1163. IEEE, 2015. 2

[15] Jerome Revaud, Philippe Weinzaepfel, Claudio R. de Souza, and Martin Humenberger. R2d2: Reliable and repeatable detector and descriptor. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019. 2

[16] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2564–2571, 2011. 2

[17] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4938–4947, 2020. 2

[18] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016. 2

[19] Yonathan Shavit, Roy Ferens, and Michael Lindenbaum. Learning multi-scene absolute pose regression with transformers. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2733–2742, 2021. 2

[20] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2930–2937, 2013. 1, 2, 3