Wide-Baseline Relative Camera Pose Estimation with Directional Learning

Kefan Chen*
Google Research
chenkefan950518@gmail.com

Noah Snavely
Google Research
snavely@google.com

Ameesh Makadia
Google Research
makadia@google.com

Abstract

Modern deep learning techniques that regress the relative camera pose between two images have difficulty dealing with challenging scenarios, such as large camera motions resulting in occlusions and significant changes in perspective that leave little overlap between images. These models continue to struggle even with the benefit of large supervised training datasets. To address the limitations of these models, we take inspiration from techniques that show regressing keypoint locations in 2D and 3D can be improved by estimating a discrete distribution over keypoint locations. Analogously, in this paper we explore improving camera pose regression by instead predicting a discrete distribution over camera poses. To realize this idea, we introduce DirectionNet, which estimates discrete distributions over the 5D relative pose space using a novel parameterization to make the estimation problem tractable. Specifically, DirectionNet factorizes relative camera pose, specified by a 3D rotation and a translation direction, into a set of 3D direction vectors. Since 3D directions can be identified with points on the sphere, DirectionNet estimates discrete distributions on the sphere as its output. We evaluate our model on challenging synthetic and real pose estimation datasets constructed from Matterport3D and InteriorNet. Promising results show a near 50% reduction in error over direct regression methods.

1. Introduction

Estimating the relative pose between two images is fundamental to many applications in computer vision such as 3D reconstruction, stereo rectification, and camera localization [19]. For calibrated cameras, relative pose is synonymous with the essential matrix, which encapsulates the projective geometry relating two views. Prevailing approaches recover the global model from corresponding points [30, 20, 42] within an iterative robust model fitting process [15, 64]. Recent progress has introduced deep-learned modules that can replace components of this classic pipeline [62, 50, 10, 9, 52, 43, 47, 2]. While this class of techniques has been extensively analyzed [49], well-known failure cases include where feature detection or matching is difficult, such as low image overlap, large changes in scale or perspective, or scenes with insufficient or repeated textures.

In these cases, it is natural to consider if supervised deep learning can address this basic task of essential matrix estimation, considering its success in tackling a variety of challenging computer vision problems. Specifically, can we train a deep neural network to represent the complex function that directly maps image pairs to their relative camera pose? Such a model would provide an appealing alternative to formulations that are sensitive to correspondence estimation performance.

Unfortunately, evidence suggests designing regression models for pose estimation is challenging, and in fact finding a parameterization of the motion groups effective in deep learning models is still an active research topic [33, 66, 44]. Not surprisingly, the initial works exploring relative pose regression (e.g. [38, 46]) are not conclusively successful in the difficult scenarios described above.

In this work we introduce a novel deep learning model for relative pose estimation, focused on the challenging wide-baseline case. Conceptually, our model generates a discrete probability distribution over relative poses, and the final pose estimate is taken as the expectation of this distribution. Our method is inspired by works that show estimating a discrete distribution, or a heatmap over a quantized output space, consistently outperforms direct regression to the continuous output space. This conclusion has been observed in different applications, such as estimating 2D and 3D keypoint locations [55, 32, 57], and estimating periodic angles [25].

However, it is currently unclear if this idea translates directly to complex higher dimensional output spaces. Relative camera pose lives in a five dimensional space, so predicting a discrete distribution would require $O(N^5)$ storage in the output space alone [35]! Given that this is currently intractable for neural networks at any reasonable resolution, the question is how can we apply this concept to the relative pose problem effectively?

To this end, we introduce a novel formulation that builds

*Work done while Kefan was a member of the Google AI Residency program (g.co/airesidency).

upon the idea of estimating discrete probability distributions on the 5D relative pose space. We propose two key components to execute this idea effectively:

1. A parameterization of the motion space that factorizes poses as a set of 3D direction vectors. A non-parametric differentiable projection step can map these directions to their closest pose.
2. *DirectionNet*, a convolutional encoder-decoder model for predicting sets of 3D direction vectors. The network outputs discrete distributions on the sphere S^2, the expected values of which produce direction vectors.

Our core contributions are in recognizing that incorporating a dense structured output in the form of a discrete probability distribution can improve wide-baseline relative pose estimation, and in introducing a technique to execute this idea efficiently. The attributes of this approach that help make it effective include (1) DirectionNet is fully-convolutional and as such does not utilize any fully-connected regression layers, and (2) it allows for additional supervision as both the dense distribution and final estimated pose can be supervised.

DirectionNet is deployed in two stages, where the relative rotation is estimated first, followed by the relative translation direction. This allows us to derotate the input images after the first stage, which reduces the complexity of the translation estimation task.

DirectionNet is evaluated on two difficult wide-baseline pose estimation datasets created from the synthetic images in Interior-Net [27], and the real images in Matterport3D [4]. DirectionNet consistently outperforms direct regression approaches (for the same pose representation as well as numerous alternatives), as well as classic feature-based approaches. This illustrates the effectiveness of estimating discrete probability distributions as an alternative to direct regression even for a complex problem such as relative pose estimation. Furthermore, these results validate that a supervised data-driven approach for wide-baseline pose estimation can succeed in cases that are extreme for traditional methods.

2. Related Work

Feature matching–based methods are still prevalent for the relative pose problem, yet suffer under large motions that yield unreliable correspondence (see [49] for a survey). Recent works deploy deep learning in subproblems such as feature detection [52, 10], filtering or reweighting outliers [62, 50]. Differentiable versions of consensus methods like RANSAC have also been proposed [47, 2]. These still rely on sufficiently accurate matches, an uncertain prospect in wide-baseline settings.

Many deep regression methods have addressed 3D object pose recovery from a single image [39, 54, 58, 28, 56]. For our task of relative pose estimation, [38] adopts a Siamese convolutional regression model to directly estimate relative camera pose from two images. [11] proposes a relative pose layer atop Siamese camera localization towers. [46] introduces a model for uncalibrated cameras that regresses to a fundamental matrix via an intermediate representation of camera intrinsics, rotation, and translation. Related to these efforts, [8] and [41] proposed deep convolutional networks for homography estimation. Despite targeting various tasks, most of the above methods share a common architecture design—a convolutional network culminating with fully connected layers for regression. A related problem to ours is camera re-localization which estimates pose from a single image in a *known* scene [22, 16]. Our setting is quite different as we try to recover the *relative* camera pose from two images in a previously unvisited scene.

Deep learning for ego-motion estimation or visual odometry is an active area which includes supervised methods such as [59] (depth supervision), and many self-supervised methods (see [5] for a survey). In general, these systems make design choices specific for small-baselines and video sequences, such as using frame-to-frame image reconstruction losses [65, 63, 61], or training with more than two frames [60, 12, 65]. In contrast, our focus is on learning to estimate relative pose for wide-baseline image pairs.

Probabilistic deep models have been used to capture uncertainty in pose predictions. In [7, 36] they estimate the distribution of 6D object poses to tackle shape symmetries and ambiguities, while [48, 18] uses directional statistics to model object rotations and regress the parameters of the probability distributions. [3] adopts mixtures of von Mises-Fisher and quasi-Projected Normal distributions to represent point sets. The multi-headed approach in [45] combines multiple predictions into a mean pose and associated covariance matrix. In contrast to most of these techniques, ours is a discrete representation and is not tied to any choice of parametric probability distribution model.

3. Method

We now outline our method for estimating relative pose from image pairs. The relative pose between two views is specified by a 3D rotation $R \in \mathrm{SO}(3)$ ($R \in \mathbb{R}^{3\times3}$, $R^T R = I$, $\det(R) = 1$), and a translation $t \in \mathbb{R}^3$. Without additional assumptions, the relative translation can only be recovered up to a scale factor, so we adopt the normalization $\|t\| = 1$; equivalently, t is restricted to the 2-sphere, $t \in S^2$. Our task is to estimate the relative pose in $\mathrm{SO}(3) \times S^2$.

In principle, we desire a model that estimates a discrete distribution over the pose space $\mathrm{SO}(3) \times S^2$. This requires discretizing this five dimensional space at a reasonable resolution, which is computationally infeasible in deep networks. Instead, our approach presents a novel parameterization for relative poses along with simplifying assumptions to make the task tractable.

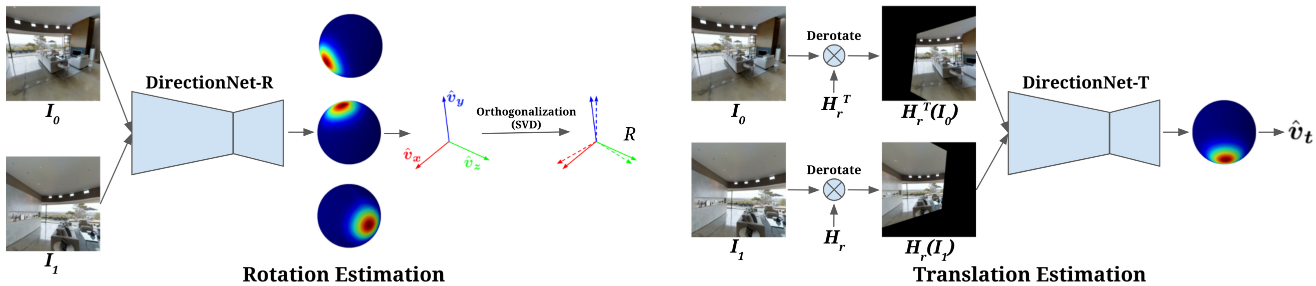


Figure 1. **Relative pose estimation.** At the core of our method is the DirectionNet, which maps a source image I_0 and a target image I_1 to a number of directional probability distributions over the 2-sphere, shown here as color-coded spheres. We convert the distributions to vectors by finding their expected values. The rotation matrix R is approximated by orthogonal Procrustes from three estimated unit vectors $(\hat{v}_x, \hat{v}_y, \hat{v}_z)$. As an alternative, DirectionNet-R could generate two directional vectors and R could be determined by Gram-Schmidt orthogonalization. To facilitate estimating the translation $\hat{v}_t$, we derotate the input images by applying the homography introduced in Sec 3.4, yielding the transformed input images $H_r^T(I_0)$ and $H_r(I_1)$ where r is half-rotation of the estimated camera rotation R.

3.1. Directional parameterization of relative pose

Parameterizing $\mathrm{SO}(3) \times S^2$ requires a choice for $\mathrm{SO}(3)$. We choose a simple over-parameterization of $\mathrm{SO}(3)$, splitting the rotation matrix $R \in \mathrm{SO}(3)$ into its component vectors $R = [x\ y\ z]$, where each component is itself a unit vector in $\mathbb{R}^3$. The relative pose between images, $(R, t) \in \mathrm{SO}(3) \times S^2$, can then be specified by the four direction vectors $x, y, z, t \in S^2$. See Section 4.3 for a discussion and justification of our parameterization.

3.2. Estimating 3D directions

The relative pose task is now one of estimating a set of direction vectors $x, y, z, t \in S^2$. Making the simplifying assumption of independence of these vectors, our approach is to (1) predict a probability distribution over the space of possible directions (S^2) for each vector, and (2) extract the direction prediction from each distribution. It is important to note that representing functions on the sphere and integrating them requires careful consideration in the context of neural networks where the data representation is restricted to regular grids. The details of our approach are below.

Spherical distributions. We represent functions on S^2 with a 2D equirectangular projection indexed by spherical coordinates (θ_i, ϕ_j). Our discretization follows [24]: θ_i is the angle of colatitude ($0 \leq i < H$, $\theta_i = \frac{(2i+1)\pi}{2H}$), ϕ_j is the azimuth ($0 \leq j < W$, $\phi_j = \frac{2\pi j}{W}$), and $H \times W$ is the grid resolution. Let $u(\theta_i, \phi_j)$ denote the unnormalized output of a network, and $f(x) = \ln(1 + e^x)$ be the softplus function. We can map u to a probability distribution with a spherical normalization:

$$P(\theta_i, \phi_j) = \frac{f(u(\theta_i, \phi_j))}{\sum_{i=0}^{H-1} \sum_{j=0}^{W-1} f(u(\theta_i, \phi_j)) \sin(\theta_i)}, \tag{1}$$

where the $\sin(\theta_i)$ in the normalizing term comes from the area element on S^2.

Spherical expectation. The $\arg\max$ operator identifies the most probable direction from a distribution, but is not differentiable and its precision is limited by the grid resolution. An alternative is the Fréchet mean [1], which is appealing since it defines a spherical centroid using the natural metric (geodesics). However, it requires a non-convex optimization and the solution is not necessarily unique. Instead, we choose the alternative of taking the expected value of the distribution. In the continuous case, we define the expected value of a random variable X on S^2 with PDF p_X as $\mathrm{E}[X] = \int_{\rho \in S^2} \rho \cdot p_X(\rho) d\rho$. In the discretization of the sphere introduced above, this becomes

$$\mathrm{E}[X] = \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} \rho(\theta_i, \phi_j) P(\theta_i, \phi_j) \sin(\theta_i), \tag{2}$$

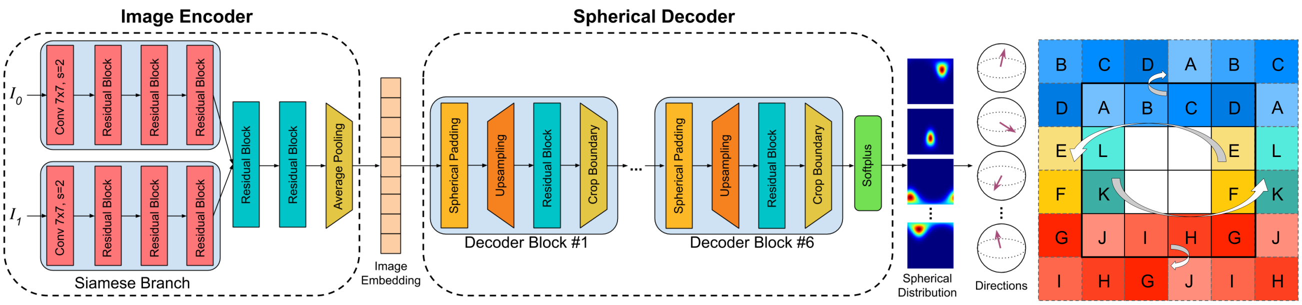
where $\rho(\theta_i, \phi_j)$ is the 3D unit vector corresponding to spherical point (θ_i, ϕ_j). The expected value $v = \mathrm{E}[X], v \in \mathbb{R}^3$ can be projected to the sphere in a straightforward manner: $\hat{v} = \frac{v}{\|v\|}$.[1]

3.3. DirectionNet for relative pose estimation

DirectionNet maps image pairs to sets of unit direction vectors following the steps outlined above. We adopt an encoder-decoder style architecture that learns a cross-domain mapping from two images to a spherical representation (see Figure 2(a) and the supplemental material). We describe two ways that DirectionNet can be instantiated for relative pose estimation.

The SVD variation. Here DirectionNet will produce four vectors $\{\hat{v}_x, \hat{v}_y, \hat{v}_z, \hat{v}_t\}$ which are the predictions of the directional pose components $\{x, y, z, t\}$. Due to the simplify-

[1] An alternative formulation to Equations 1 and 2 would include the traditional soft-argmax operator adapted to the sphere. This would reinterpret the network output u as log-probabilities and require $f(x) = e^x$ in Eq. 1.



(a) DirectionNet architecture (b) Spherical padding

Figure 2. (a) The image encoder generates embeddings from a pair of input images, and the spherical decoder transforms and upsamples these embeddings to produce probability distributions over S^2, which are represented with equirectangular maps. (b) Spherical padding ensures the boundary pixels reflect the correct neighbors on the sphere. In this example, a 4×4 grid is padded to size 6×6. The corresponding labeled squares illustrate the padding process. See the supplemental for further clarification on the padding process.

ing assumptions earlier, in general $[\hat{v}_x,\ \hat{v}_y,\ \hat{v}_z] \notin \mathrm{SO}(3)$ as orthogonality is not guaranteed. However, we can project $M = [\hat{v}_x,\ \hat{v}_y,\ \hat{v}_z]$ to $\mathrm{SO}(3)$ with orthogonal Procrustes [53], which is a differentiable procedure [57] and provides the optimal projection to $\mathrm{SO}(3)$ by Frobenius norm:

$$R = U \mathrm{diag}(1, 1, \det(UV^T))V^T, \tag{3}$$

where $U\Sigma V^T = M$ is the SVD of M [53].

The Gram-Schmidt variation. For $[\hat{v}_x,\ \hat{v}_y,\ \hat{v}_z] \in \mathrm{SO}(3)$, the last column $\hat{v}_z$ is fully constrained: $\hat{v}_z = \hat{v}_x \times \hat{v}_y$. Thus, we only need DirectionNet to produce three vectors $\{\hat{v}_x, \hat{v}_y, \hat{v}_t\}$ and obtain a rotation by a partial Gram-Schmidt projection of $\hat{v}_x$ and $\hat{v}_y$ as in [66].

Both the SVD and Gram-Schmidt projections have been shown recently to reach state-of-the-art performance for different 3D rotation estimation tasks, especially when predicting arbitrary (large) rotations. See [26] and [66] for the analysis. Although overall performance between the two is similar, SVD is shown to be slightly more effective, with a possible explanation being that SVD finds a least-squares projection while Gram-Schmidt is greedy. Our experimental findings confirm the analysis.

3.4. Two-stage model with derotation

Intuitively, we expect that learning camera translation should be easiest when the data never exhibit rotational motion. Hence, we propose estimating camera pose sequentially: (1) A DirectionNet (denoted DirectionNet-R) estimates the relative rotation between input images, (2) this rotation is used to *derotate* the input images, and (3) a second DirectionNet (denoted DirectionNet-T) predicts translations from the rotation-free image pair. Figure 1 illustrates the stages of this process.

However, when the relative rotation between the cameras is large, derotating one image relative to the other could result in projecting most of the scene outside the camera's field of view. To limit this effect we find that it is helpful to project both input images to an intermediate frame using half of the estimated rotation with a larger FoV and proportionally increased resolution. The derotation implementation involves a homography transformation $H_r = K' r^\top K^{-1}$, where K is the matrix of camera intrinsics for the input image, K' is the intrinsics for the desired derotated image, and r is the half-rotation of the estimated camera rotation R. Setting $K' = K$ maintains the field of view (FoV) and resolution of the input image after derotation. We use ${H_r}^T$ and H_r to project input images I_0 and I_1, respectively, onto the "middle" frame. The output FoV and resolution are controllable parameters of the model. To mitigate the effect of rotation estimation errors on translation prediction, we developed a rotation augmentation scheme with random perturbation (see the supplemental material).

3.5. Network architecture

The DirectionNet architecture is illustrated in Figure 2(a). Notably, we eschew the skip connections common to similar convolutional architectures ([40, 51]) since they would not reflect the correct spatial associations between the planar and spherical topologies (observed in [14]).

The **image encoder** embeds image pairs into $\mathbb{R}^{512}$ using a Siamese architecture. A Siamese branch consists of a 7×7, stride-2 convolution followed by a series of residual blocks[2], each of which downsamples by 2. The outputs are concatenated in the channel dimension, and after two more residual blocks, a global average pooling produces the embedding.

The **spherical decoder** maps embeddings to spherical distributions by repeatedly applying bilinear upsampling and residual blocks. We use spherical padding before upsampling to ensure adjacent pixels at the boundaries reflect the correct neighbors on the sphere (see Figure 2(b)). The final

[2]All residual blocks consist of two bottleneck blocks [21] with 3×3 convolutions, batch normalization, and leaky ReLU pre-activations.

outputs, of size $64{\times}64{\times}k$, are interpreted as k spherical distributions, and are subsequently mapped to k direction vectors following equations 1 and 2. In total, a single DirectionNet contains ~9M parameters. Note that we could also consider a Spherical CNN decoder [6, 13]. While such models are equivariant to 3D rotations, the benefit in our setting would be limited since the image encoder is not rotation equivariant. Hence, we opt for the computational efficiency of 2D convolutions.

3.6. Loss terms and model training

In this section, we describe the loss terms and training strategy for our model. We let $P(\theta_i, \phi_j)$ denote a single distribution generated by DirectionNet, and let its corresponding ground truth distribution be $P^*(\theta_i, \phi_j)$. The ground truth distributions are constructed using the von Mises-Fisher distribution as described in Sec. 4.1. With a slight abuse of notation we will denote with $E[P]$ the expected value of a distribution and is computed according to equation 2. One appealing property of DirectionNet is that both the output direction vectors as well as their corresponding dense probability distributions can be supervised.

The **direction loss** is the negative cosine similarity between two 3D vectors:

$$L_{\nearrow}(p_1, p_2) = -\frac{p_1^T p_2}{\|p_1\|\|p_2\|}. \tag{4}$$

We introduce two loss terms to supervise distributions. First, the **distribution loss** provides dense supervision on the equirectangular distribution grid:

$$L_D(P_1, P_2) = \frac{1}{HW}\sum_{i=0}^{H-1}\sum_{j=0}^{W-1}(P_1(\theta_i,\phi_j) - P_2(\theta_i,\phi_j))^2 \sin(\theta_i), \tag{5}$$

Second, the **spread loss** penalizes the spherical "variance" [37] to encourage unimodal and concentrated distributions:

$$L_\sigma(p) = 1 - \|p\|. \tag{6}$$

The full loss for a single predicted direction vector combines the three individual losses:

$$\begin{aligned} L(P, P^*) \quad = \quad & L_{\nearrow}(E[P], E[P^*]) + \lambda_D L_D(P, P^*) + \\ & \lambda_\sigma L_\sigma(E[P])). \end{aligned} \tag{7}$$

See the supplemental for an analysis of the individual loss terms. By far the most impactful loss on performance is the distribution loss L_D.

We train our two-stage pipeline sequentially. We first train DirectionNet-R, which takes the source and target image pair (I_0, I_1) as input and produces three direction vectors $[\hat{v}_x\ \hat{v}_y\ \hat{v}_z]$, which are mapped onto SO(3). The complete training loss for DirectionNet-R is the sum of individual losses for the three estimated directions $L_R = L(P_x, P_x^*) + L(P_y, P_y^*) + L(P_z, P_z^*)$. After training, DirectionNet-R is frozen and its predictions are used for derotating the inputs of DirectionNet-T. DirectionNet-T takes the input pair $(H_r^T(I_0), H_r(I_1))$ and outputs a translation direction $\hat{v}_t$. Since the translation is represented with a single unit vector, DirectionNet-T is trained with the loss $L_T = L(P_t, P_t^*)$.

4. Experiments

To evaluate our method on challenging data exhibiting a wide range of relative motion, we generate image pairs from existing panoramic image collections. Image pairs are generated by sampling pairs of panoramas from a common scene, then projecting them to overlapping planar perspective views. By varying the camera viewing angles we create a dataset with varied relative poses and overlap. Specific details of each dataset are provided below. Unless specified otherwise, our training image pairs have a resolution of 256×256 and a 90° FoV. In all cases, we generate 1M training pairs and ~1K test pairs. *Note that there is no overlap between the train and test scenes* (see the supplemental material for details).

InteriorNet [27] is a synthetic dataset with 560 scenes with panoramas rendered along smooth camera trajectories which we sample at random strides. InteriorNet-A is constructed to have rotations up to 30°, while InteriorNet-B has varied FoV (60° to 90°) and rotations up to 40°.

Matterport3D [4] contains 10K *real* panoramas captured from locations ~2.25m apart covering 90 scenes. Matterport-A is constructed to have rotations up to 45°, and Matterport-B up to 90°. These are more challenging than InterionNet due to a wider baseline and smaller overlap.

4.1. Training details

We train with loss weights $\lambda_\sigma = 0.1$, $\lambda_D = 8{\times}10^7$, and Adam [23] with a learning rate of 1e-3 and a batch size of 20. The ground truth distributions are generated by the von Mises-Fisher distribution with concentration $\kappa = 10.0$ (κ is analogous to $\frac{1}{\sigma^2}$ in a Gaussian distribution). The derotated images for InteriorNet have a 90° FoV, while the derotated Matterport3D images have a 105° FoV and an increased resolution of 344×344 to compensate for the larger rotations.

4.2. Baselines

We now introduce the baselines. For space considerations we leave the full implementation details, as well as results for additional methods, to the supplement.

DirectionNet variations. We consider multiple variants of our full two-stage model with intermediate derotation. **DirectionNet-9D** projects three direction vectors onto SO(3) using SVD for the rotation estimation, while **DirectionNet-6D** uses a partial Gram-Schmidt projection [66] (refer to Sec. 3.3 for details). To understand the

		Matterport-A						Matterport-B					
		R			t			R			t		
		mean (°)	med (°)	rank	mean (°)	med (°)	rank	mean (°)	med (°)	rank	mean (°)	med (°)	rank
DirectionNet	9D	**3.96**	2.28	**2.76**	**14.17**	**6.46**	**3.29**	13.60	**3.54**	**2.89**	**21.26**	**8.90**	**3.44**
	6D	4.30	**2.22**	2.79	16.37	7.07	3.29	14.85	3.69	3.45	23.60	9.42	3.79
	9D-Single	4.55	3.11	3.83	21.65	10.53	4.71	**13.37**	4.00	2.85	28.41	13.27	4.26
	Quat.	23.32	23.00	8.25	39.85	24.85	6.22	37.09	25.25	7.13	49.39	31.59	6.94
Regression	Bin&Delta	6.93	4.71	5.28	22.84	10.16	3.73	31.54	22.98	6.45	29.45	14.30	5.14
	Spherical	10.68	7.98	6.79	40.09	22.85	6.36	32.94	20.56	6.42	51.00	33.18	8.40
	6D	5.73	3.66	3.79	35.75	21.89	6.32	18.23	7.69	4.29	39.06	25.07	5.69
	Quat.	15.40	12.66	6.86	41.57	21.47	7.18	28.38	19.23	6.19	48.99	34.94	7.63
SIFT	LMedS	25.55	5.63	7.71	35.53	14.84	6.20	36.58	10.54	8.13	42.67	26.64	6.06
	RANSAC	19.33	6.66	7.31	45.04	29.78	8.08	31.30	9.55	7.74	47.74	26.19	6.19

Table 1. **Quantitative results on the Matterport A and B datasets.** We report the mean and median angular error in degrees, as well as average rank of each method over all test pairs. Rotation (R) and translation (t) shown separately.

importance of derotation, we also consider a single-stage version without derotation (**DirectionNet-9D-Single**) which estimates four directions from a single DirectionNet module.

Discrete pose representation alternatives. To evaluate our choice of representation for the discretized pose space, we consider multiple alternatives: **Bin&Delta** [34] is a hybrid model combining a coarse rotation classification (over clustered quaternions) with a refinement regression network. To understand if decoupling a 3D rotation into a set of direction vectors is necessary, we introduce **DirectionNet-Quat** which estimates a discrete distribution over the space of unit quaternions at resolution 32^3. The spherical decoder is replaced by a 3D volumetric CNN decoder. **3D-RCNN** [25] estimates individual (Euler) angles with a discrete distribution over the quantized circle. 3D-RCNN consistently under-performed Bin&Delta so we leave its discussion to the supplemental material.

Regression baselines. We evaluated multiple direct pose regression baselines: **Spherical regression** [29] uses a novel spherical exponential activation for regression to n-spheres; **6D** [66] regresses 6D outputs followed by a Gram-Schmidt projection for rotations; **Quaternion** regresses a unit quaternion for the rotation. This is what is used in the camera pose regression modules from PoseNet [22] and [38]. The multiple regression baselines share the same image encoder architecture as our DirectionNet.

Parametric probabilistic pose. vM [48] estimates individual (Euler) angles by directly regressing the continuous parameters of a von-Mises distribution (location and concentration) on the circle. We were unable to train vM successfully on all datasets so we delegate these results and discussion to the supplement.

Feature-based baselines. We consider two versions of the classic correspondence-based pipeline, SIFT features [31] with robust LMedS [64] (**SIFT+LMedS**) or with RANSAC [15] (**SIFT+RANSAC**). We also consider pipelines with learned components such as **SuperGlue** [52] and **D2-Net** [10].

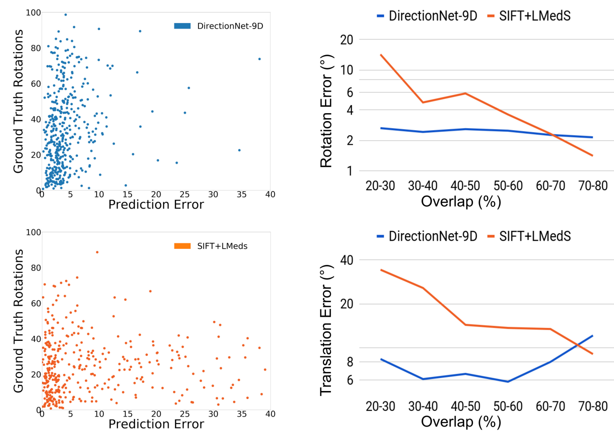


Figure 3. **(a)** True rotation magnitude (°) vs error (°). The scatter plot shows that our model is robust in the presence of large relative rotations. **(b)** Median error (°) vs. overlap (%). As image overlap decreases from 90% to 20%, the median test errors of our method increases much slower than the SIFT+LMedS. When overlap is very high, local feature based techniques are still superior.

Note, the differences in most baselines are in the rotation representation. Unless specified otherwise, the baselines predict a unit-normalized 3D vector for the camera translation direction.

Evaluation metrics. We report geodesic errors for both rotations and translation directions, separately. Additionally, we rank each method on every test pair, reporting the mean rank across examples (1 is the best possible rank, 10 is the worst possible rank).

4.3. Analysis

Table 1 reports the quantitative results of the different methods on the Matterport datasets (see supplemental for results on the InteriorNet datasets). We begin by considering the two main questions for analyzing our approach.

Can relative camera pose be better learned using a dis-

Matterport-A	R		t	
	mean (°)	med (°)	mean (°)	med (°)
SuperGlue (indoor)[52]	8.34	5.22	21.08	11.86
D2-Net [10]	18.79	7.12	41.58	25.56
DirectionNet-9D	**3.96**	**2.28**	**14.17**	**6.46**
Matterport-B	R		t	
	mean (°)	med (°)	mean (°)	med (°)
SuperGlue (indoor)[52]	**13.23**	7.19	29.90	13.28
D2-Net [10]	21.03	7.74	43.82	27.30
DirectionNet-9D	13.60	**3.54**	**21.26**	**8.90**

Table 2. Performance of learned feature-based methods on the Matterport A and B datasets.

crete pose distribution versus direct regression? We observe that our DirectionNet outperforms all baselines on each metric for each dataset. A particularly illustrative comparison is DirectionNet-6D vs 6D regression, where for the same pose representation, prediction via a discrete distribution is consistently better. We do not see the same improvement for DirectionNet-Quat over Quaternion regression, however, which indicates our choice of the lower-dimensional spherical/directional representation is important (see discussion below). Finally, we observe that predicting a parametric probabilistic representation of pose does not help (vM [48] performed 5x worse than DirectionNet – see supplemental).

Is our directional representation better than alternatives for estimating discrete distributions over relative camera pose? Both DirectionNet-6D/9D and DirectionNet-9D-Single consistently outperform alternatives which also use a quantized output space in some way: Bin&Delta [34], DirectionNet-Quat, and 3D-RCNN [25] (discussed in supplemental). Our DirectionNet-Quat baseline predicts a distribution over the half hypersphere in S^3. Its poor performance supports our hypothesis that the smaller resolution allowed by its $O(N^3)$ space requirements limits performance, whereas our directional models require just $O(N^2)$ space.

Feature-based approaches In this work our aim is to understand if relative pose regression can be improved using a discrete pose distribution representation, especially in the difficult wide-baseline setting. Thus, our primary experimental analysis is a comparison of different regression and probabilistic techniques. For completeness, we also evaluate feature-based approaches. Unsurprisingly, the feature-based methods have different performance characteristics compared to the learned direct methods. For example, for those image pairs where feature extraction and matching are likely successful (e.g. high overlap), the estimated motion based on SIFT features is consistently better than any learned technique (Fig. 3-b). We note that in the best cases the SIFT methods both regularly reach sub-1° errors in relative rotation estimation, while all of the learning methods rarely reach errors that low. However, our dataset construction intentionally includes a large fraction of large-motion pairs likely to have low overlap, and this drives down the overall performance of these methods (Fig. 3-a). In addition to classic feature-based methods, we compare with learned descriptors and matching pipelines. Table 2 shows results using pretrained models for SuperGlue [52] and D2-Net [10]. SuperGlue has slightly better performance than DirectionNet on mean rotation error for Matterport-B, but in general DirectionNet outperforms the learned feature-based methods. These results indicate that our datasets include many image pairs where keypoint detection and matching is difficult (weakly textured regions and repeated patterns make up a large subset of Matterport data).

Qualitative Results. Figure 4 shows results on the challenging real Matterport data which includes large baselines and occlusions. We qualitatively assess each method by visualizing epipolar lines after prediction. We see that DirectionNet can still recover the correct relative pose in never-seen test scenes even when presented with extreme motions.

Generalization. To demonstrate the generalization ability of our model, we train DirectionNet-9D on InteriorNet-A (synthetic) and test it on Matterport-A (real). The mean and the median errors of the rotation are 8.42° and 5.13°, and 20.71° and 8.60° for translation. *Even without any fine-tuning on real data our approach still outperforms most baselines which had the benefit of training on Matterport-A.* To test the model's performance on outdoor scenes, we trained on a subset of KITTI [17]. DirectionNet gives 9.19° mean rotation error and 19.36° translation error while the best baseline gives 13.44° and 22.53° for rotation and translation respectively. See the supplemental for details.

Supplemental preview. The supplemental material contains many additional details, results, and discussions that were left out of this main paper due to limited space. These additional sections include all dataset generation and training details, as well as all results (quantitative and qualitative) on InteriorNet. We also visualize the predicted distributions from DirectionNet and observe their properties in ambiguous or failure cases.

5. Conclusion

The results presented above tell a consistent story. Models that regress relative pose directly from wide-baseline image pairs can be improved by estimating a discrete probability distribution in the pose space. Our approach effectively executes this idea by operating on a factorized pose space that is lower dimensional than the 5D pose space and suitable for discretized outputs. Evaluated on challenging synthetic and real wide-baseline datasets, DirectionNet generally outperforms regression models, parametric probabilistic models, alternative discretization schemes, and feature-based baselines.

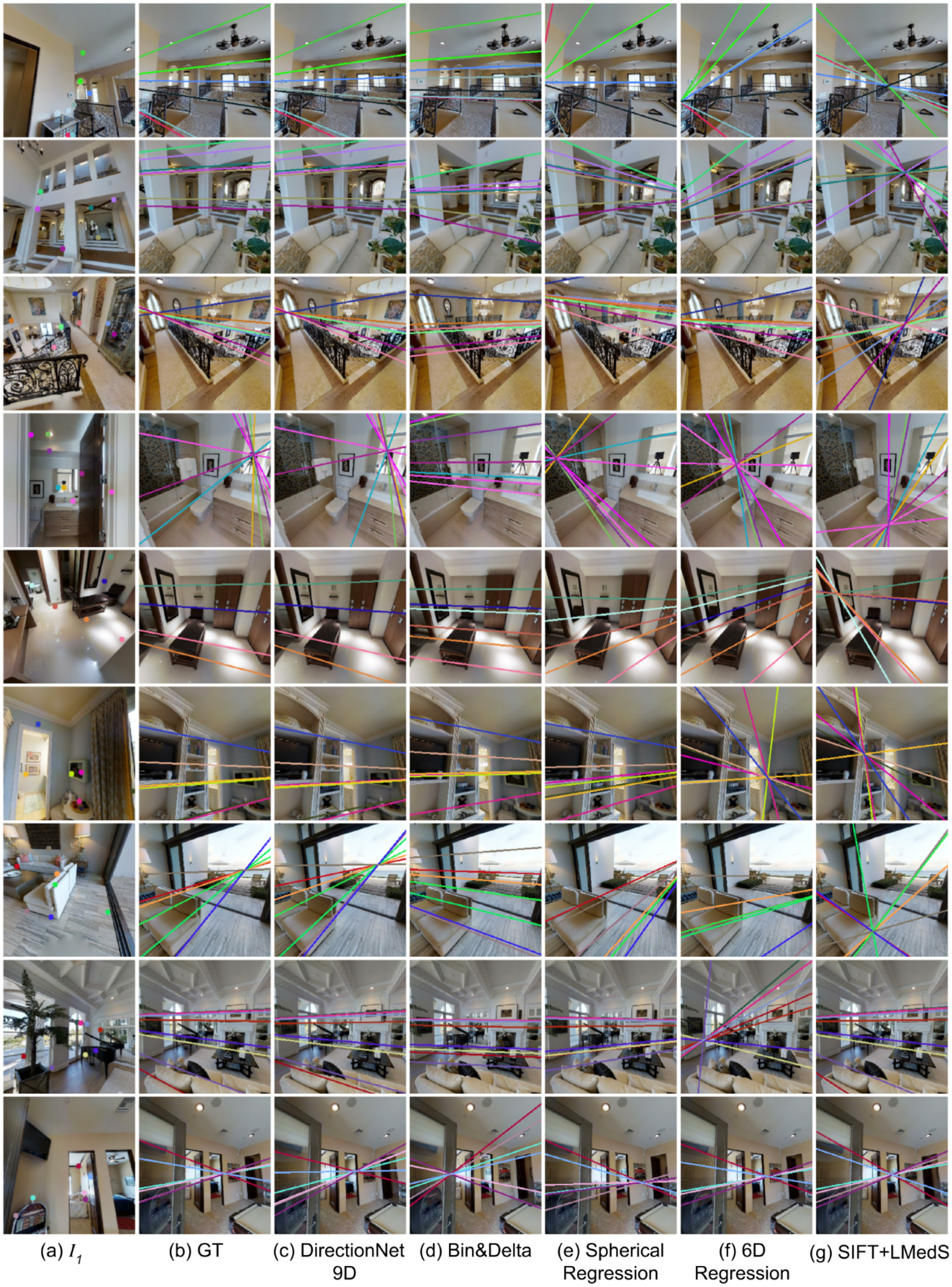


Figure 4. **Qualitative evaluation on Matterport-B.** Any point in one image plane corresponds to a ray shooting from the optical center, which could be projected to the other image plane as the *epipolar line*. (a) We draw a number of points detected by SIFT in different colors on each target image I_1, and (b) show their corresponding epipolar lines on the source image using the ground truth pose, (c) visualizations from our DirectionNet-9D, (d) Bin & Delta, (e) spherical regression, (f) 6D regression, (g) SIFT+LMedS. Most examples demonstrates some of the most difficult scenarios, such as drastic change in viewpoint and significant occlusion. The last two rows show that SIFT+LMedS can outperform the others in the case of smaller motions for which the the feature-based approach can find reliable feature correspondences.

References

[1] Bijan Afsari. Riemannian L^p center of mass: Existence, uniqueness, and convexity. *Proceedings of the American Mathematical Society*, 139, 02 2011.

[2] 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)*, page 9, 2018.

[3] Dylan Campbell, Lars Petersson, Laurent Kneip, Hongdong Li, and Stephen Gould. The alignment of the spheres: Globally-optimal spherical mixture alignment for camera pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[4] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. In *International Conference on 3D Vision (3DV)*, 2017.

[5] Changhao Chen, Bing Wang, Chris Xiaoxuan Lu, Niki Trigoni, and Andrew Markham. A survey on deep learning for localization and mapping: Towards the age of spatial machine intelligence. *arXiv preprint arXiv:2006.12567*, 2020.

[6] Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical CNNs. In *International Conference on Learning Representations (ICLR)*, 2018.

[7] Xinke Deng, Arsalan Mousavian, Yu Xiang, Fei Xia, Timothy Bretl, and Dieter Fox. PoseRBPF: A Rao-Blackwellized Particle Filter for 6D Object Pose Estimation. In *Proceedings of Robotics: Science and Systems*, FreiburgimBreisgau, Germany, June 2019.

[8] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. In *RSS Workshop on Limits and Potentials of Deep Learning in Robotics*, 2016.

[9] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[10] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[11] Sovann En, Alexis Lechervy, and Frédéric Jurie. Rpnet: An end-to-end network for relative camera pose estimation. In *European Conference on Computer Vision Workshops (ECCVW)*, 2018.

[12] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):611–625, 2017.

[13] Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. Learning SO(3) equivariant representations with spherical CNNs. In *European Conference on Computer Vision (ECCV)*, September 2018.

[14] Carlos Esteves, Avneesh Sud, Zhengyi Luo, Kostas Daniilidis, and Ameesh Makadia. Cross-domain 3D equivariant image embeddings. In *International Conference on Machine Learning, ICML*, 2019.

[15] 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*, 24(6):381–395, June 1981.

[16] Victor Fragoso, Chunhui Liu, Aayush Bansal, and Deva Ramanan. Patch correspondences for interpreting pixel-level cnns. *arXiv: Computer Vision and Pattern Recognition*, 2017.

[17] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[18] Igor Gilitschenski, Roshni Sahoo, Wilko Schwarting, Alexander Amini, Sertac Karaman, and Daniela Rus. Deep orientation uncertainty learning based on a bingham loss. In *International Conference on Learning Representations*, 2020.

[19] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003.

[20] Richard I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, June 1997.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision (ECCV)*, 2016.

[22] 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)*, page 2938–2946, 2015.

[23] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference for Learning Representations*, 2015.

[24] Peter J. Kostelec and Daniel N. Rockmore. Ffts on the rotation group. *Journal of Fourier Analysis and Applications*, 14(2):145–179, Apr 2008.

[25] A. Kundu, Y. Li, and J. M. Rehg. 3D-RCNN: Instance-level 3D object reconstruction via render-and-compare. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3559–3568, 2018.

[26] Jake Levinson, Carlos Esteves, Kefan Chen, Noah Snavely, Angjoo Kanazawa, Afshin Rostamizadeh, and Ameesh Makadia. An analysis of SVD for deep rotation estimation. In *Advances in Neural Information Processing Systems 34*, 2020.

[27] Wenbin Li, Sajad Saeedi, John McCormac, Ronald Clark, Dimos Tzoumanikas, Qing Ye, Yuzhong Huang, Rui Tang, and Stefan Leutenegger. Interiornet: Mega-scale multi-sensor photo-realistic indoor scenes dataset. In *British Machine Vision Conference (BMVC)*, 2018.

[28] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. In *European Conference on Computer Vision (ECCV)*, September 2018.

[29] Shuai Liao, Efstratios Gavves, and Cees G. M. Snoek. Spherical regression: Learning viewpoints, surface normals and 3D rotations on n-spheres. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[30] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293(5828):133–135, 1981.

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

[32] Diogo C. Luvizon, Hedi Tabia, and David Picard. Human pose regression by combining indirect part detection and contextual information. *Computers & Graphics*, 85:15–22, Dec 2019.

[33] S. Mahendran, H. Ali, and R. Vidal. 3d pose regression using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017.

[34] Siddharth Mahendran, Haider Ali, and René Vidal. A mixed classification-regression framework for 3D pose estimation from 2d images. *The British Machine Vision Conference (BMVC)*, 2018.

[35] Ameesh Makadia, Christopher Geyer, and Kostas Daniilidis. Correspondence-free structure from motion. *International Journal of Computer Vision*, 75(3):311–327, 2007.

[36] Fabian Manhardt, Diego Martin Arroyo, Christian Rupprecht, Benjamin Busam, Tolga Birdal, Nassir Navab, and Federico Tombari. Explaining the Ambiguity of Object Detection and 6D Pose From Visual Data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[37] K. V. Mardia. Statistics of directional data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 37(3):349–393, 1975.

[38] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. Relative camera pose estimation using convolutional neural networks. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, 2017.

[39] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d bounding box estimation using deep learning and geometry. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[40] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision (ECCV)*, pages 483–499. Springer, 2016.

[41] Ty Nguyen, Steven W. Chen, Shreyas S. Shivakumar, Camillo J. Taylor, and Vijay Kumar. Unsupervised deep homography: A fast and robust homography estimation model. In *IEEE Robotics and Automation Letters*, volume 3, pages 2346–2353, 2018.

[42] David Nister. An efficient solution to the five-point relative pose problem. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 195–202, 2003.

[43] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. Lf-net: Learning local features from images. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[44] Valentin Peretroukhin, Matthew Giamou, David M. Rosen, W. Nicholas Greene, Nicholas Roy, and Jonathan Kelly. A Smooth Representation of SO(3) for Deep Rotation Learning with Uncertainty. In *Proceedings of Robotics: Science and Systems (RSS'20)*, 2020.

[45] Valentin Peretroukhin, Brandon Wagstaff, and and Jonathan Kelly. Deep Probabilistic Regression of Elements of SO(3) using Quaternion Averaging and Uncertainty Injection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[46] Omid Poursaeed, Guandao Yang, Aditya Prakash, Qiuren Fang, Hanqing Jiang, Bharath Hariharan, and Serge Belongie. Deep fundamental matrix estimation without correspondences. In *European Conference on Computer Vision (ECCV)*, pages 485–497, 2018.

[47] Thomas Probst, Danda Pani Paudel, Ajad Chhatkuli, and Luc Van Gool. Unsupervised learning of consensus maximization for 3D vision problems. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[48] Sergey Prokudin, Peter Gehler, and Sebastian Nowozin. Deep directional statistics: Pose estimation with uncertainty quantification. In *European Conference on Computer Vision (ECCV)*, Sept. 2018.

[49] Rahul Raguram, Jan-Michael Frahm, and Marc Pollefeys. A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus. In *European Conference on Computer Vision (ECCV)*, pages 500–513, 2008.

[50] René Ranftl and Vladlen Koltun. Deep fundamental matrix estimation. In *European Conference on Computer Vision (ECCV)*, 2018.

[51] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, 2015.

[52] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[53] P.H. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31:1–10, 1966.

[54] Hao Su, Charles R. Qi, Yangyan Li, and Leonidas J. Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3D model views. In *IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[55] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *European Conference on Computer Vision (ECCV)*, 2018.

[56] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3D orientation learning for 6d object detection from rgb images. In *European Conference on Computer Vision (ECCV)*, September 2018.

[57] Supasorn Suwajanakorn, Noah Snavely, Jonathan J Tompson, and Mohammad Norouzi. Discovery of latent 3D keypoints via end-to-end geometric reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2063–2074, 2018.

[58] Shubham Tulsiani and Jitendra Malik. Viewpoints and keypoints. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[59] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox.

Demon: Depth and motion network for learning monocular stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[60] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2043–2050. IEEE, 2017.

[61] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1281–1292, 2020.

[62] Kwang Moo Yi, Eduard Trulls Fortuny, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 9, 2018.

[63] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1983–1992, 2018.

[64] Zhengyou Zhang, Rachid Deriche, Olivier Faugeras, and Quang-Tuan Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, pages 87–119, 1995.

[65] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[66] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.