

# SparsePose: Sparse-View Camera Pose Regression and Refinement

Samarth Sinha<sup>1</sup> Jason Y. Zhang<sup>2</sup>  
 Andrea Tagliasacchi<sup>1,3,4</sup> Igor Gilitschenski<sup>1</sup> David B. Lindell<sup>1,5</sup>

<sup>1</sup>University of Toronto <sup>2</sup>Carnegie Mellon University <sup>3</sup>Simon Fraser University <sup>4</sup>Google <sup>5</sup>Vector Institute

## Abstract

*Camera pose estimation is a key step in standard 3D reconstruction pipelines that operate on a dense set of images of a single object or scene. However, methods for pose estimation often fail when only a few images are available because they rely on the ability to robustly identify and match visual features between image pairs. While these methods can work robustly with dense camera views, capturing a large set of images can be time-consuming or impractical. We propose SparsePose for recovering accurate camera poses given a sparse set of wide-baseline images (fewer than 10). The method learns to regress initial camera poses and then iteratively refine them after training on a large-scale dataset of objects (Co3D: Common Objects in 3D). SparsePose significantly outperforms conventional and learning-based baselines in recovering accurate camera rotations and translations. We also demonstrate our pipeline for high-fidelity 3D reconstruction using only 5-9 images of an object.*

## 1. Introduction

Computer vision has recently seen significant advances in photorealistic new-view synthesis of individual objects [24, 41, 52, 59] or entire scenes [5, 61, 80]. Some of these multiview methods take tens to hundreds of images as input [5, 35, 36, 41], while others estimate geometry and appearance from a few sparse camera views [43, 52, 75]. To produce high-quality reconstructions, these methods currently require accurate estimates of the camera position and orientation for each captured image.

Recovering accurate camera poses, especially from a limited number of images is an important problem for practically deploying 3D reconstruction algorithms, since it can be challenging and expensive to capture a dense set of images of a given object or scene. While some recent methods for appearance and geometry reconstruction jointly tackle the problem of camera pose estimation, they typically require dense input imagery and approximate initialization [34, 66] or specialized capture setups such as imag-

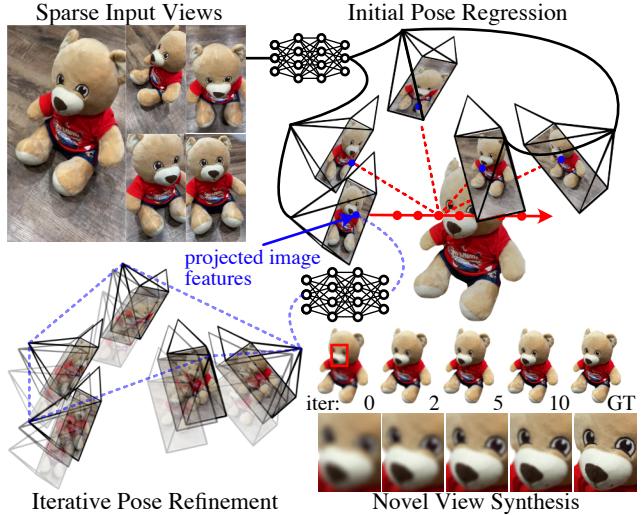


Figure 1. **SparsePose** – Given sparse input views, our method predicts initial camera poses and then refines the poses based on learned image features aggregated using projective geometry. SparsePose outperforms conventional methods for camera pose estimation based on feature matching within a single scene and enables high-fidelity novel view synthesis (shown for each iteration of pose refinement) from as few as five input views.

ing rigs [27, 77]. Most conventional pose estimation algorithms learn the 3D structure of the scene by matching image features between pairs of images [46, 58], but they typically fail when only a few wide-baseline images are available. The main reason for this is that features cannot be matched robustly resulting in failure of the entire reconstruction process.

In such settings, it may be outright impossible to find correspondences between image features. Thus, reliable camera pose estimation requires learning a *prior* over the geometry of objects. Based on this insight, the recent RelPose method [78] proposed a probabilistic energy-based model that learns a prior over a large-scale object-centric dataset [52]. RelPose is limited to predicting camera rotations (i.e., translations are not predicted). Moreover, it op-

erates directly on image features without leveraging explicit 3D reasoning.

To alleviate these limitations, we propose *SparsePose*, a method that predicts camera rotation and translation parameters from a few sparse input views based on 3D consistency between projected image features (see Figure 1). We train the model to learn a prior over the geometry of common objects [52], such that after training, we can estimate the camera poses for sparse images and generalize to unseen object categories. More specifically, our method performs a two-step coarse-to-fine image registration: (1) we predict *coarse* approximate camera locations for each view of the scene, and (2) these initial camera poses are used in a pose refinement procedure that is simultaneously iterative and autoregressive, which allows learning *fine-grained* camera poses. We evaluate the utility of the proposed method by demonstrating its impact on sparse-view 3D reconstruction.

Our method outperforms other methods for camera pose estimation in sparse view settings. This includes conventional image registration pipelines, such as COLMAP [58], as well as recent learning-based methods, such as RelPose [78]. Overall, SparsePose enables real-life, sparse-view reconstruction with as few as five images of common household objects, and is able to predict accurate camera poses, with only 3 source images of previously unseen objects.

In summary, we make the following contributions.

- We propose SparsePose, a method that predicts camera poses from a sparse set of input images.
- We demonstrate that the method outperforms other techniques for camera pose estimation in sparse settings;
- We evaluate our approach on 3D reconstruction from sparse input images via an off-the-shelf method, where our camera estimation enables much higher-fidelity reconstructions than competing methods.

## 2. Related Work

Camera pose estimation from RGB images is a classical task in computer vision [42, 47]. It finds applications in structure-from-motion (SfM [20]), visual odometry [44], simultaneous localization and mapping (SLAM [17]), rigid pose estimation [9], and novel view synthesis with neural rendering (NVS [41]). In our discussion of related work, we focus on several related types of pose inference as well as few-shot reconstruction.

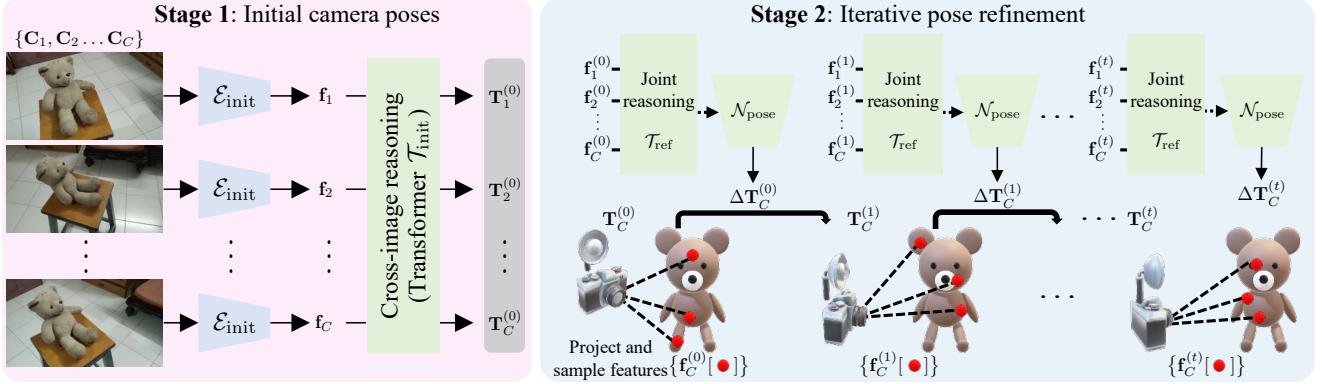
**Local pose estimation.** A variety of techniques estimate camera poses by extracting keypoints and *matching* their local features across input images [3, 26, 37, 38, 54, 57, 60]. Features can be computed in a hand-crafted fashion [6, 39] or can be learnt end-to-end [14, 74]. Cameras are then refined via bundle adjustment, where camera poses and 3D keypoint locations are co-optimized so to minimize repro-

jection errors [1, 62]. Generally speaking, feature matching methods fail in *few-shot* settings due to occlusion and limited overlap between images, where a sufficient number of common keypoints cannot be found. These methods fail because they are *local* – and therefore do not learn *priors* about the solution of the problem.

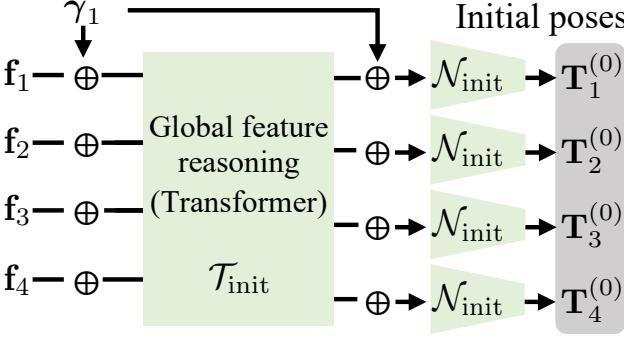
**Global pose optimization.** Global pose optimization methods rely on differentiable rendering techniques to recover camera poses by minimizing a photometric reconstruction error [68, 71]. Within the realm of neural radiance fields [41], we find techniques for estimating pose between images [73] and between a pair of NeRFs [19], as well as models that *co-optimize* the scene’s structure altogether with camera pose [12, 34, 66], even including camera distortion [25]. SAMURAI [7] is able to give very accurate camera poses by performing joint material decomposition and pose refinement, but relies on coarse initial pose estimates, many images for training ( $\approx 80$ ), and trains from scratch for each new image sequence. In contrast, our method is trained once and can then perform forward inference on unseen scenes in seconds. Overall, such global pose optimization methods often fail in few-shot settings since they require sufficiently accurate pose initializations to converge. While local methods are often used for initialization, they too fail with sparse-view inputs, and so cannot reliably provide pose initializations in this regime.

**Global pose regression.** Given a set of input images camera poses can also be directly regressed. In visual odometry, we find techniques that use neural networks to autoregressively estimate pose [65, 72], but these methods assume a small baseline between subsequent image pairs, rendering them unsuitable to the problem at hand. *Category-specific* priors can be learnt by robustly regressing pose w.r.t. a “canonical” 3D shape [28, 70, 79], or by relying on strong semantic priors such as human shape [40, 63] or (indoor) scene appearance [10]. Closest to our method, recent *category-agnostic* techniques, such as RelPose [78], are limited to estimating *only rotations* by learning an energy-based probabilistic model over  $SO(3)$ . However, RelPose only considers the global image features from the sparse views and does not perform local 3D consistency. Even for predicting rotations, our method performs significantly better than RelPose since it takes into account both global and local features from images.

**Direct few-shot reconstruction.** Rather than regressing pose and then performing reconstruction, it is also possible to reconstruct objects *directly* from (one or more) images using data-driven category priors [52], or directly training on the scene. Category-specific single-image 3D reconstruction estimate geometry and pose by matching pixels or 2D keypoints to a 3D template [31–33, 45], learning to synthesize class-specific 3D meshes [18, 33], or exploiting

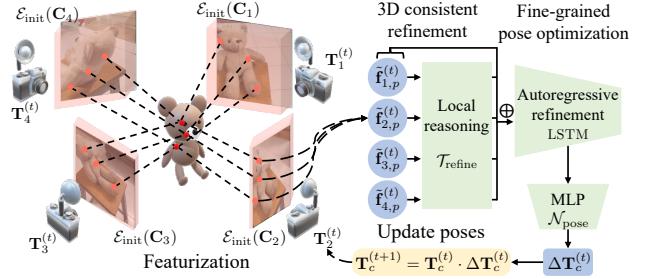


**Figure 2. Method Overview.** We propose Sparse-View Camera Pose Regression and Refinement (SparsePose), which takes as input few-views of an object from wide baselines, and predicts the camera poses. SparsePose is trained on a large scale dataset of “common objects” to learn a prior over the 3D geometry of the scene and the object. Our method works by first predicting coarse initial camera poses by performing cross-image reasoning. The initial camera pose estimates are then iteratively refined in an auto-regressive manner, which learns to implicitly encode the 3D geometry of the scene based on sampled image features. For notational convenience and simplicity, we use  $\mathbf{T}$  to represent the rotations  $\mathbf{R}$  and translations  $\mathbf{t}$  in homogeneous coordinates (as used in the text).



**Figure 3. Stage 1 architecture:** We initialize the camera poses by directly estimating the models using *global reasoning*, and directly regressing the poses using pretrained features and joint-reasoning over the source images. We note that  $\oplus$  denotes a skip connection (or addition) between the input and the output of the transformer  $\mathcal{T}_{init}$ , and for learnable positional encoding  $\gamma$ . For simplicity we use  $\mathbf{T}$  to represent the rotations  $\mathbf{R}$  and translations  $\mathbf{t}$  in homogeneous coordinates (as used in the text). A detailed approach of stage 1 is in Section 3.1.

image symmetry [68, 69]. Recent progress has also enabled few-shot novel view synthesis, where images of the scene from a novel viewpoint are generated *conditioned* on only a small set of images [13, 16, 21, 23, 43, 52, 55, 67, 76]. Such methods are either trained to learn category-centric features [21, 52, 68], or are trained on a large-scale dataset to encode the 3D geometry of the scenes [55, 67], or propose regularization schemes for neural radiance based methods [13, 23, 43]. However, 3D consistency in these models is learnt by *augmentation* rather than by construction, resulting in lower visual quality compared to our approach.



**Figure 4. Stage 2 architecture:** After obtaining the initial camera poses from Stage 1, we iteratively and autoregressively refine the camera poses using a local feature reasoning module, which learns the optimization dynamics of the camera poses. Since the optimization is non-linear, the model iteratively updates the camera poses by resampling points and predicting pose offsets. We note that  $\oplus$  denotes a skip connection between the input and the output of the transformer  $\mathcal{T}_{refine}$ , and for simplicity we use  $\mathbf{T}$  to represent the rotations  $\mathbf{R}$  and translations  $\mathbf{t}$  in homogeneous coordinates (as used in the text). A detailed approach of stage 2 is in Section 3.2.

### 3. Method

Estimating camera parameters typically involves predicting the intrinsics (i.e. focal length, principal point, skew) and extrinsics (i.e. rotation, translation) from a set of images. In this paper, we only consider the task of estimating *extrinsics*; we assume the intrinsics are known as they can be calibrated once for a camera a priori and are often provided by the camera manufacturer. More formally, our goal is to jointly predict the rotation  $\mathbf{R}_c \in \text{SO}(3)$  and translation  $\mathbf{t}_c \in \mathbb{R}^3$  for all input images  $\mathbf{C}_c$ . Our proposed method

for this task consists of two phases, which are illustrated in Fig. 2:

- **Section 3.1** – we first **initialize** the camera poses in a coarse prediction step which considers the *global* image features in the scene.
- **Section 3.2** – we **refine** the poses using an iterative procedure in which an autoregressive network predicts updates to align *local* image features to match the 3D geometry of the scene.

The goal of the coarse pose estimation is to use global image features to provide an initial 3D coordinate frame and estimates of the relative camera rotations and translations which can then be iteratively refined.

### 3.1. Initializing camera poses

Given a sparse set of  $C$  images  $\{\mathbf{C}_1, \dots, \mathbf{C}_C\}$  of a single object, the first task is to predict initial camera poses and establish a coordinate frame. The initialization:

- extracts low-resolution image features  $\mathbf{f} \in \mathbb{R}^F$ ;
- combines image features into a global representation;
- regresses rotation and translation for each camera.

We use a pre-trained, self-supervised encoder  $\mathcal{E}_{\text{init}}$  [8] to extract features from each image. Following ViT [15], we also add a learnable positional embedding  $\gamma_c \in \mathbb{R}^F$  to each feature:

$$\mathbf{f}_c = \overbrace{\mathcal{E}_{\text{init}}(\mathbf{C}_c)}^{\text{pre-trained features}} + \overbrace{\gamma_c}^{\text{learnable encoding}}. \quad (1)$$

These features are passed to a transformer  $\mathcal{T}_{\text{init}}$  [15, 64] and a skip connection to aggregate global context and predict a new set of features:

$$\mathbf{g}_c = \mathcal{T}_{\text{init}}(\{\mathbf{f}_1, \dots, \mathbf{f}_C\}; \boldsymbol{\theta}) + \mathbf{f}_c. \quad (2)$$

Finally, a shallow fully-connected network  $\mathcal{N}_{\text{init}}$  (we use two hidden layers) predicts quaternions representing the initial camera rotations  $\mathbf{R} \in \text{SO}(3)$ , and translations  $\mathbf{t} \in \mathbb{R}^3$ :

$$\mathbf{R}_c^{(0)}, \mathbf{t}_c^{(0)} = \mathcal{N}_{\text{init}}(\mathbf{g}_c; \boldsymbol{\theta}). \quad (3)$$

The initial rotation and translation estimates are then refined using the iterative procedure described in Section 3.2.

### 3.2. Refining camera poses

After obtaining the initial poses,  $\mathbf{R}^{(0)}, \mathbf{t}^{(0)}$ , we can leverage geometric reasoning to iteratively refine the pose estimates  $\mathbf{R}^{(t)}, \mathbf{t}^{(t)}$ , where  $1 \leq t \leq T$ . We achieve this by probing a collection of 3D points within the scene given the *current* camera estimate (i.e. the points are re-sampled at each step). After projecting the points back into the images, features are fetched and aggregated into global feature vectors from which camera pose updates are computed.

**Sampling.** We aim to uniformly sample within the volume where we expect the imaged object to be located. Given the initially estimated camera poses, the center of the capture volume  $\mathbf{c}$  is predicted considering principal rays (i.e. rays passing through the principal point of each image). We compute the point closest to the principal rays (in the least-squares sense) and calculate the average camera radius as  $r^{(t)} = \mathbb{E}_c \|\mathbf{c} - \mathbf{c}_c^{(t)}\|_2$ , where  $\mathbf{c}_c^{(t)}$  is the camera center of image  $c$  at iteration  $t$ . We then uniformly sample  $P$  points within an Euclidean ball:

$$\{\mathbf{p}_p^{(t)}, \dots, \mathbf{p}_P^{(t)}\} \sim \mathcal{B}(\mathbf{c}^{(t)}, r^{(t)}/2). \quad (4)$$

To increase robustness of the optimization and to ensure the camera does not get stuck in a *local minima*, we re-sample the 3D points after each camera pose update to jitter the 3D points and image features, analogous to how PointNet jitters input pointcloud data [49].

**Featurization.** Let  $\mathbf{R}_c^{(t)}$  and  $\mathbf{T}_c^{(t)}$ , denote the estimated rotation and translation at the  $(t)$ -th iteration of the refinement procedure. Given the set of 3D points and a known camera intrinsic matrix  $\mathbf{K}$ , we project them into the coordinate frame of each camera using 3D geometry:

$$\mathbf{p}_{c,p}^{(t)} = \Pi_c^{(t)}(\mathbf{p}_p^{(t)}) = \mathbf{K}(\mathbf{R}_c^{(t)} \mathbf{p}_p^{(t)} + \mathbf{t}_c^{(t)}). \quad (5)$$

We interpolate samples of the previously extracted image features  $\mathbf{f}$  at the projected 2D pixel coordinates for each source image [21], resulting in a set of feature embeddings for *each* camera image and *each* point at the *current* refinement iteration. We concatenate the positional encoding for the current predicted rotations and translations and the original 3D points to the embedding to generate a joint local feature vector:

$$\mathbf{f}_{c,p}^{(t)} = \mathcal{E}_{\text{init}}(\mathbf{C}_c)[\mathbf{p}_{c,p}] \quad [\cdot] \equiv \text{bilinear} \quad (6)$$

$$\tilde{\mathbf{f}}_{c,p}^{(t)} = [\mathbf{f}_{c,p}^{(t)}, \gamma(\mathbf{p}_p^{(t)}), \gamma(\mathbf{R}_c^{(t)}), \gamma(\mathbf{t}_c^{(t)})] \in \mathbb{R}^{134}, \quad (7)$$

where  $\gamma(\cdot)$  is a Fourier positional encoding [41]. We then reduce the dimensionality of this large vector with a single linear layer  $\mathcal{E}_{\text{refine}}$ , and then concatenate along the *samples* dimension:

$$\tilde{\mathbf{f}}_c^{(t)} = \left[ \mathcal{E}_{\text{refine}}(\tilde{\mathbf{f}}_{c,1}^{(t)}; \boldsymbol{\theta}), \dots, \mathcal{E}_{\text{refine}}(\tilde{\mathbf{f}}_{c,P}^{(t)}; \boldsymbol{\theta}) \right] \in \mathbb{R}^{P \cdot 32}, \quad (8)$$

where  $P$  is the number of sampled points, which was chosen to be 1,000 resulting in a 32,000 dimensional local feature vector for each pose iteration step. With this *local* feature vector  $\tilde{\mathbf{f}}_c^{(t)}$  we summarize the appearance and geometry of the scene as sampled by the  $c$ -th camera, allowing learned refinement of the camera poses in a 3D consistent manner to predict the camera pose updates.

**Optimization.** Similar to (2), we use a multi-headed self-attention module along with a skip connection to perform

*joint* reasoning over the source views:

$$\tilde{\mathbf{g}}_c = \mathcal{T}_{\text{refine}}(\tilde{\mathbf{f}}_0^{(t)}, \dots, \tilde{\mathbf{f}}_c^{(t)}) + \tilde{\mathbf{f}}_c^{(t)}, \quad (9)$$

and regress pose updates using a long short-term memory network LSTM [22] and a 2-layer MLP  $\mathcal{N}_{\text{pose}}$ :

$$\begin{aligned} \bar{\mathbf{g}}_c^{(t)} &= \text{LSTM}(\tilde{\mathbf{g}}_c^{(t)}; \{\tilde{\mathbf{g}}_c^{(t-1)}, \dots, \tilde{\mathbf{g}}_c^{(0)}\}) \quad (10) \\ \Delta \mathbf{R}_c^{(t)}, \Delta \mathbf{t}_c^{(t)} &= \mathcal{N}_{\text{pose}}(\bar{\mathbf{g}}_c^{(t)}) \\ \mathbf{R}_c^{(t+1)} &= \mathbf{R}_c^{(t)} \cdot \Delta \mathbf{R}_c^{(t)} \quad (11) \\ \mathbf{t}_c^{(t+1)} &= \mathbf{t}_c^{(t)} + \Delta \mathbf{t}_c^{(t)}. \end{aligned}$$

Such auto-regressive models have been shown effective in implementing meta-optimization routines [2], as they can learn priors on the dynamics of the optimization in few-shot settings [51]. In practice, we perform 10 steps of the LSTM pose refinement to allow for the camera poses to converge. An additional ablation study for the number of steps is provided in the supplementary.

To train the model we minimize the loss

$$\mathcal{L}_{\text{pose}} = \mathbb{E}_C \sum_{k \in \{0, K\}} \mathcal{L}_{\text{pose}}^{\mathbf{R}} + \mathcal{L}_{\text{pose}}^{\mathbf{t}}, \quad (12)$$

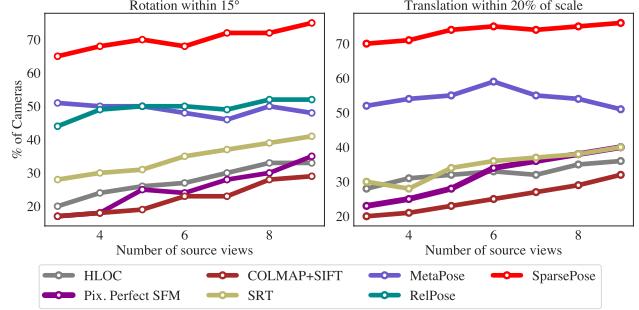
$$\mathcal{L}_{\text{pose}}^{\mathbf{R}} = d(\|\mathbf{R}_c^{(t)} \mathbf{R}_c^{\text{GT}} - \mathbb{I}\|_{\mathcal{F}}, \quad (13)$$

$$\mathcal{L}_{\text{pose}}^{\mathbf{t}} = d(\|\mathbf{t}_c^{(t)} - \mathbf{t}_c^{\text{GT}}\|_2^2), \quad (14)$$

where GT denotes ground truth data obtained by applying COLMAP [58] to *densely* sampled video footage—note that we require dense frames at training time only, while our inference procedure uses sparse views. To make the regression invariant to choices of coordinate frames, we align the ground truth rotation and translation such that the first source image is always at the unit camera location  $\mathbf{R}=\mathbb{I}$ ,  $\mathbf{t}=0$ . The model then predicts *relative* rotations and translations in this canonical coordinate space. To stabilize training, we follow [11, 29] and take the loss over normalized quaternions. For the penalty function  $d(\cdot)$  we use an adaptive and robust loss function [4]. The losses are applied only to the initial pose estimator and the output of the pose refinement module (i.e., the last iteration of the LSTM update). Both prediction stages are trained jointly end-to-end.

### 3.3. Implementation Details

**Training data.** We train the model on the CO3D dataset [52], which contains 19,000 videos, with 1.5 M individual frames and camera poses across 50 different categories of common objects. Training the model on this large dataset with diverse objects facilitates learning object-appearance priors, which ultimately enables pose prediction from sparse views. We split the dataset into 30 train and 20 test categories, to verify the method’s ability to adapt to novel classes.



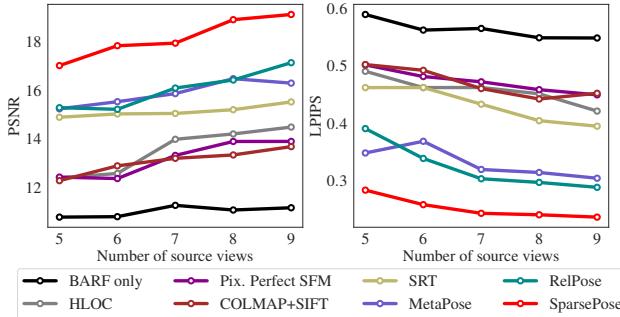
**Figure 5. Quantitative evaluation of sparse-view camera pose estimation.** We evaluate the quality of rotations and translations for varying numbers of source views. We show the percentage of cameras that were predicted to be within  $15^\circ$  of the ground truth (left) and translations that were predicted within 20% of the scale of the scene compared to ground truth (right). SparsePose outperforms both classical and learning-based methods.

**Testing data.** To construct the test set, we use the 20 test categories, and sample 100 sequences for each number of source images  $C \in \{3, 9\}$ . To sample a test-set, we follow the uniform-variant of the evaluation protocol from RelPose [78], and perform stratified sampling along the frame indices in the CO3D dataset to select for wide-baseline views [52]. Furthermore, we use batch sampling from PyTorch3D [50] to shuffle batches with a random number of source images  $C \in \{3, \dots, 9\}$ , so that the model learns how to aggregate features and jointly predicts camera poses with a different number of source images. We note that the model architecture is designed to work with an arbitrary number of unposed source images. As mentioned previously, the camera intrinsic matrix  $\mathbf{K}$  is assumed to be known for all training and testing sequences, which is a reasonable assumption, given that the CO3D dataset was collected from smartphones and such parameters can be easily obtained from smartphone manufacturers.

**Training details.** The model is trained on two A6000 48 GB GPUs for 3 days until convergence. The model is jointly trained using an Adam optimizer [30] with initial learning rate of  $10^{-4}$ , which is decayed once by a factor of 10 after 250 epochs of training. The other Adam optimizer parameters are left to the default values from the PyTorch implementation [48]. A pre-trained DINO Vision Transformer is used to compute the pre-trained embeddings  $\mathcal{E}_{\text{init}}$  [8, 15]. We use the frozen weights from the official release of the ViT-B/8 variant of the DINO ViT model.

## 4. Experiments

For the task of sparse-view camera pose estimation, we consider both classical Structure-from-Motion baselines and modern deep learning based techniques. More specif-



**Figure 6. Quantitative evaluation of sparse-view, novel view synthesis.** We use the camera poses predicted by each method to perform novel view synthesis; SparsePose significantly outperforms other baseline methods for this task in terms of PSNR (higher is better) and LPIPS (lower is better).

ically, we compare SparsePose against three classical SfM baselines:

- COLMAP with SIFT features [39, 58];
- Hierarchical Localization (HLOC) [56] which uses COLMAP with SuperPoint for feature extraction [14] and SuperGlue for image matching [57];
- Pixel-Perfect SfM [37] which is a state-of-the-art SfM method that refines COLMAP camera poses using “featuremetric bundle adjustment”.

We further compare SparsePose against:

- Scene Representation Transformer (SRT) [55] by adding an additional layer to the transformer output which jointly learns 3D reconstruction and pose estimation over the large dataset;
- MetaPose [63] where we initialize the camera estimates using our initial pose estimation model, and perform pose refinement using their architecture;
- RelPose [78] which only predicts rotations by learning a energy-based probabilistic model over  $\text{SO}(3)$ , given a set of images by only considering the *global* features across the images in the scene.

For camera pose estimation, since the *ground-truth* cameras from CO3D [52] are in arbitrary coordinate frames, we measure the relative rotations and translations. That is, we measure the absolute angle difference between the ground truth and the predicted camera poses by using the Rodriguez’s formula between the rotations [53], and measure the  $\ell_2$  norm between the translations. Following RelPose [78], we report the percentage of cameras that were predicted within  $15^\circ$  of the ground truth rotation. For translations, since the scale of the scene changes between sequences, we report the percentage of cameras that were within 20% of the scale of the scene.

We then evaluate the predicted cameras for a downstream task of few-view 3D reconstruction on 20 unseen

test categories. For the novel-view synthesis task, we report the Peak-Signal-to-Noise-Ratio (PSNR) which measures the difference in the RGB space, and Learned Perceptual Image Patch Similarity (LPIPS) which measures the difference as a “perceptual” score.

#### 4.1. Wide-baseline camera pose estimation

We report quantitative results with different numbers of source views in Figure 5 which shows the percentage of predicted rotations within  $15^\circ$  of ground truth and predicted translations within 20% of the scale of the scene. The ground truth is obtained using SfM on dense videos with more than 300 images [58]. SparsePose is able to significantly outperform both classical SfM and learning-based baselines by a significant margin. Moreover, SparsePose consistently improves in performance as the number of source views increases, which is not the case across all baselines (e.g., MetaPose [63]). Using our method, 65 – 80% of predicted camera locations and orientations fall within the thresholds described above. In contrast, correspondence based approaches such as COLMAP [39, 58], HLOC [56], and Pixel-Perfect SfM [37], are only able to recover 20 – 40% of the cameras within the thresholds for rotation and translation with  $C < 10$ . This significant difference in performance motivates the effectiveness of learning appearance priors and learning to perform geometry-based pose refinement. We note that RelPose [78] only predicts rotations, and therefore cannot be evaluated on the translation prediction task.

We also show a visualization of the predicted and ground truth cameras in Figure 7, where we project the camera centers onto the  $x$ - $y$  plane to help visualize 3D offsets in the camera centers. SparsePose predicts accurate camera poses given a sparse set of images with very wide baselines, significantly outperforming other methods. Even on very challenging sequences with uneven lighting and low textural information (e.g.,  $C=7$ ), our method predicts accurate cameras, which is important in practical cases. We note that on many sequences HLOC fails to register all the source images and so does not converge to a usable output; we cannot include a result in this case.

#### 4.2. Sparse-view 3D reconstruction

We test the predicted cameras on the downstream task of sparse-view 3D reconstruction using a NeRFormer that is trained on the test categories [52] (but not any of the test *sequences* we evaluate on). Training is performed using the default hyperparameters from PyTorch3D [50]. During evaluation, we further finetune the cameras using BARF [34], an off-the-shelf 3D reconstruction and pose refinement technique which updates the camera poses by minimizing the photometric loss. In addition to the previous baselines, we add a “BARF-only” baseline, which performs

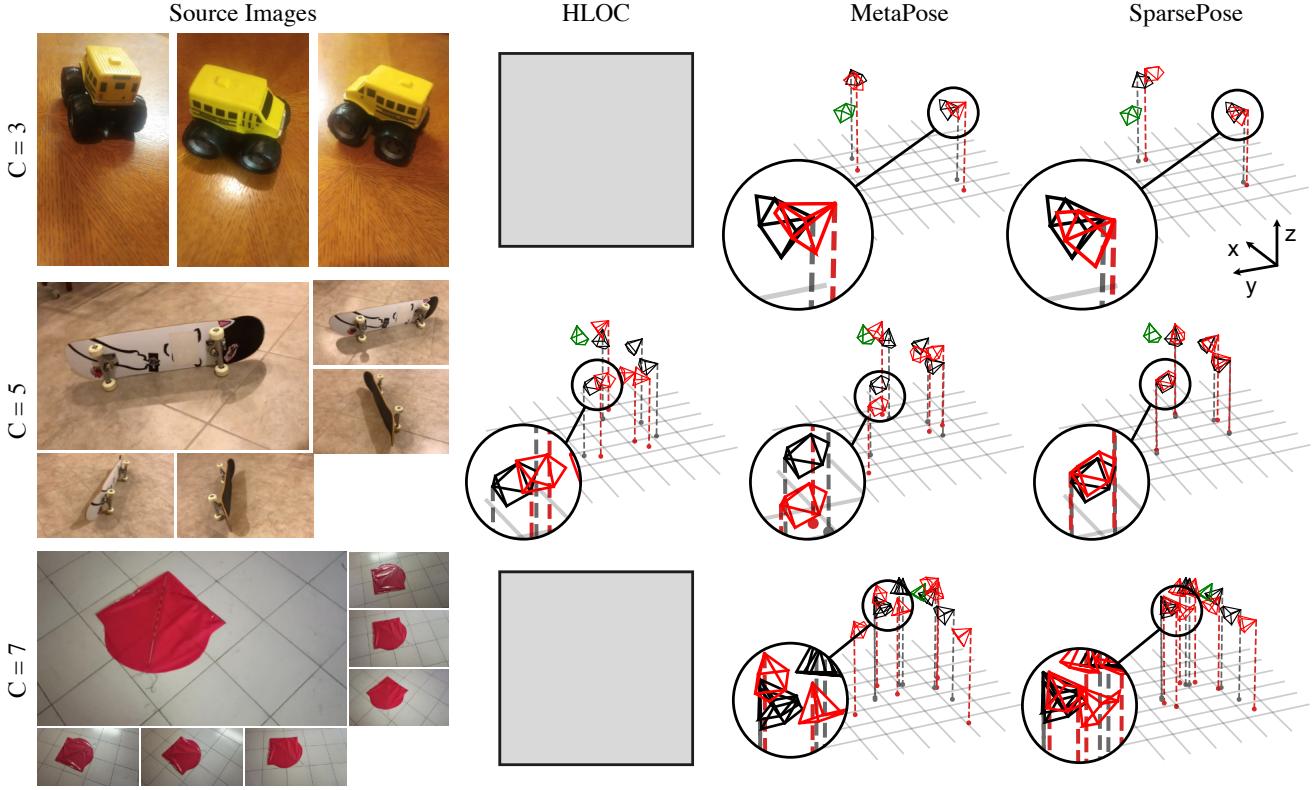


Figure 7. **Visualization of camera poses.** We compare the predicted camera poses for various methods and different number of images, camera centers are projected to the  $x - y$  plane for comparison. The ground truth poses are shown in black, predicted poses in red, and the first camera for each sequence (used to align predictions) in green. Gray boxes indicate failure to converge.

refinement after a unit camera initialization for finetuning with BARF (i.e.,  $\mathbf{R} = \mathbb{I}$ ,  $\mathbf{T} = 0$ ).

Quantitative results are shown in Figure 6. Here, SparsePose significantly outperforms all baselines. Most importantly, we show that, while predicted rotations and translations are not perfect, we can yet recover high-fidelity 3D reconstructions in the downstream task. Additionally, we find that BARF without good initial pose estimates does not converge to accurate camera poses. This further highlights the importance of accurate initial pose estimates. For the comparison to RelPose, we use the ground truth translations as predicted by CO3D (since the method does not predict translations); yet, SparsePose significantly outperforms RelPose across all numbers of the source views in both visual metrics. Finally, we also show qualitative novel-view synthesis results in Figure 8. SparsePose results in significantly better novel-view synthesis compared to baselines such as RelPose (with ground truth translations) and HLOC. Note that when using significantly more source images, the performance of conventional methods such as HLOC [56] or COLMAP [58] typically improves. For example, see the analysis in Zhang et al. [78] which shows performance of such methods with up to 20 images.

### 4.3. Ablation study

We provide an ablation study of SparsePose using different variants of the model to justify the design choices (see Figure 9): (1) only initial pose (i.e., no pose refinement), (2) no resampling of the 3D points between iterations, (3) using an MLP instead of LSTM, (4) no positional encoding on the inputs to  $\mathcal{T}_{\text{refine}}$ , (5) using RGB values instead of features from  $\mathcal{E}_{\text{init}}$  for the refinement step, (6) no robust kernel [4]. We show that SparsePose with the proposed design outperforms the other variants. Interestingly, using the initial poses  $\mathbf{R}^{(0)}, \mathbf{t}^{(0)}$ , we achieve performance similar to classical SfM methods, which highlights the importance of learned appearance priors from large datasets. The best performance is achieved when refining camera poses using the proposed method, including positional encoding, autoregressive prediction, etc.

## 5. Conclusion

In this paper, we presented SparsePose, a learning-based solution to perform sparse-view camera pose estimation from wide-baseline input images. The strong performance of the method highlights the utility of leveraging large

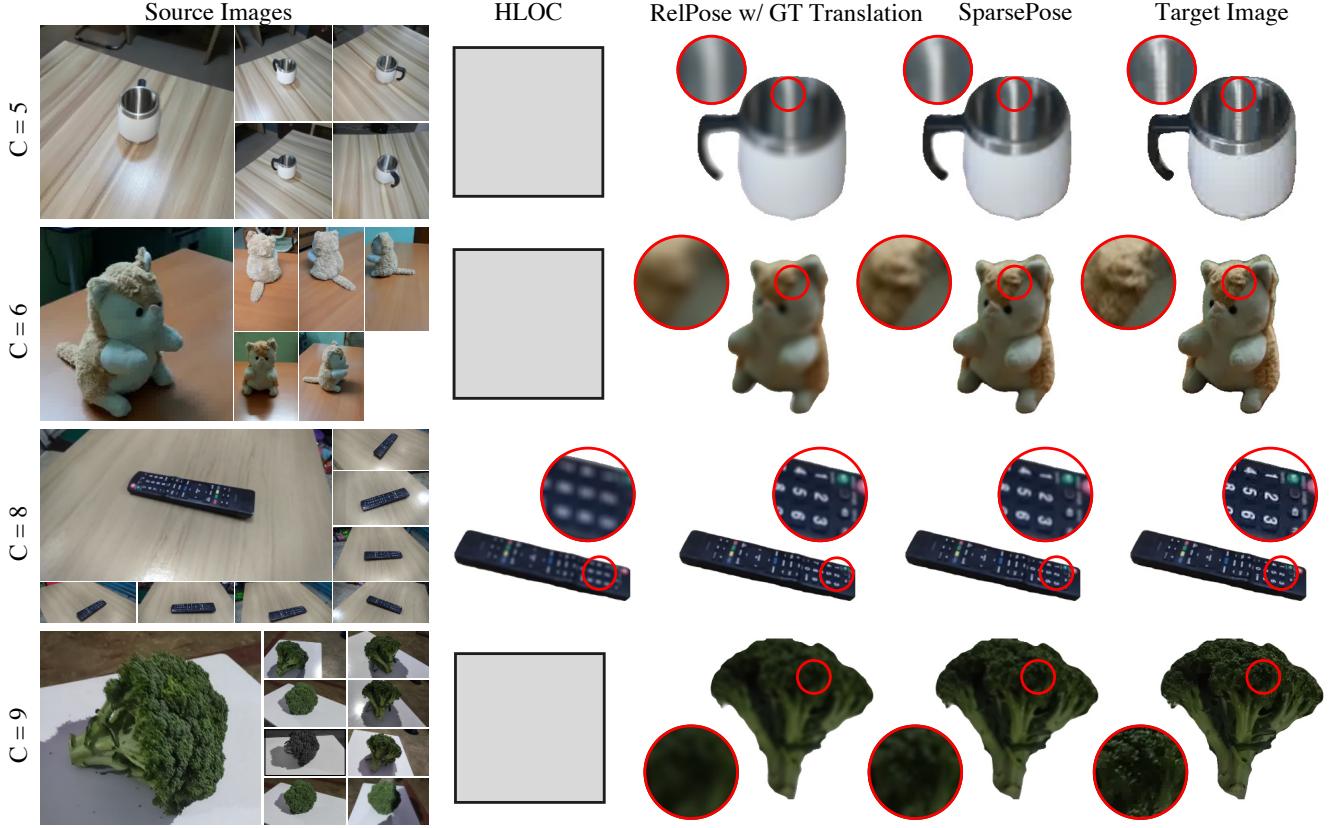


Figure 8. **Visualizing the rendered from few-sparse unposed images.** We use the initial predicted poses by each of the methods, and refine them using BARF [34], and a category-centric pretrained NeRFormer model, trained on the category. The importance of predicting accurate initial poses can be seen since SparsePose is able to generate photorealistic renders. Gray boxes indicate failure to converge.

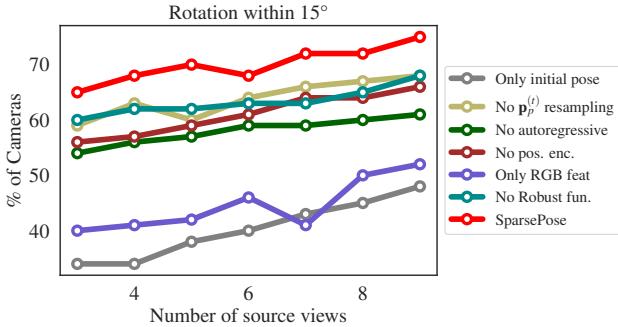


Figure 9. **Ablation results over different design choices for SparsePose.** We perform an exhaustive ablation to justify the design choices made in the paper, and report the ability of the model to correctly predict the rotations within  $15^\circ$  of the ground truth rotations, over different number of source views.

object-centric datasets for learning pose regression and refinement. Moreover, we show that accurate few-view pose estimation can enable few-view novel-view synthesis even

from challenging “in-the-wild” datasets, where our method outperforms other baselines.

There remain many potential directions for future work, among which, we believe that joint methods for pose regression and 3D scene geometry prediction may enable further improved capabilities for novel view synthesis from sparse views. Additional work on learning where and *how* to sample the 3D points used in our pose refinement step may help extend the approach to other camera motions beyond the “tracked-to-object” camera poses common in the CO3D dataset. Furthermore, it may be interesting to apply variants of our approach to the challenge of non-rigid structure from motion, where correspondence-based methods tend to fail. Learning a prior over *motion*, may help with camera pose estimation for highly non-rigid scenes, such as for scenes with smoke, loose clothing, or humans performing complex movements. Finally, our work may be broadly relevant to improving the robustness of robotic vision, autonomous navigation systems, and for efficient digital asset creation. We envision that creators will be able to take a few sparse images of common objects and generate photorealistic 3D assets for applications in augmented or virtual reality.

**Acknowledgements.** This project was supported in part by NSERC under the RGPIN program.

## References

- [1] Sameer Agarwal, Noah Snavely, Steven M Seitz, and Richard Szeliski. Bundle adjustment in the large. In *ECCV*, 2010. 2
- [2] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. *NeurIPS*, 2016. 5
- [3] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012. 2
- [4] Jonathan T Barron. A general and adaptive robust loss function. In *CVPR*, 2019. 5, 7
- [5] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022. 1
- [6] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 2008. 2
- [7] Mark Boss, Andreas Engelhardt, Abhishek Kar, Yuanzhen Li, Deqing Sun, Jonathan T Barron, Hendrik Lensch, and Varun Jampani. Samurai: Shape and material from unconstrained real-world arbitrary image collections. *NeurIPS*, 2022. 2
- [8] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *CVPR*, 2021. 4, 5, 13
- [9] Jiale Chen, Lijun Zhang, Yi Liu, and Chi Xu. Survey on 6d pose estimation of rigid object. In *Chinese Control Conference (CCC)*, 2020. 2
- [10] Kefan Chen, Noah Snavely, and Ameesh Makadia. Wide-baseline relative camera pose estimation with directional learning. In *CVPR*, 2021. 2
- [11] Yun-Chun Chen, Haoda Li, Dylan Turpin, Alec Jacobson, and Animesh Garg. Neural shape mating: Self-supervised object assembly with adversarial shape priors. In *CVPR*, 2022. 5
- [12] Shin-Fang Chng, Sameera Ramasinghe, Jamie Sherrah, and Simon Lucey. Garf: Gaussian activated radiance fields for high fidelity reconstruction and pose estimation. *arXiv e-prints*, 2022. 2
- [13] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *CVPR*, 2022. 3
- [14] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabenovich. Superpoint: Self-supervised interest point detection and description. In *CVPR (workshops)*, 2018. 2, 6, 13
- [15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 4, 5, 13
- [16] Emilien Dupont, Miguel Bautista Martin, Alex Colburn, Aditya Sankar, Josh Susskind, and Qi Shan. Equivariant neural rendering. In *International Conference on Machine Learning*. PMLR, 2020. 3
- [17] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 2006. 2
- [18] Shubham Goel, Angjoo Kanazawa, and Jitendra Malik. Shape and viewpoint without keypoints. In *ECCV*, 2020. 2
- [19] Lily Goli, Daniel Rebain, Sara Sabour, Animesh Garg, and Andrea Tagliasacchi. nerf2nerf: pairwise registration of neural radiance fields. *arXiv preprint arXiv:2211.01600*, 2022. 2
- [20] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 2
- [21] Philipp Henzler, Jeremy Reizenstein, Patrick Labatut, Roman Shapovalov, Tobias Ritschel, Andrea Vedaldi, and David Novotny. Unsupervised learning of 3d object categories from videos in the wild. In *CVPR*, 2021. 3, 4
- [22] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8), 1997. 5
- [23] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *ICCV*, 2021. 3
- [24] Wonbong Jang and Lourdes Agapito. Codenerf: Disentangled neural radiance fields for object categories. In *ICCV*, 2021. 1
- [25] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Anima Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields. In *ICCV*, 2021. 2
- [26] Yuhe Jin, Dmytro Mishkin, Anastasia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image matching across wide baselines: From paper to practice. *IJCV*, 129(2), 2021. 2
- [27] Hanbyul Joo and Hao Liu. Panoptic studio: A massively multiview system for social motion capture. In *ICCV*, 2015. 1
- [28] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *ICCV*, 2017. 2
- [29] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *ICCV*, 2015. 5
- [30] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5, 13
- [31] Filippos Kokkinos and Iasonas Kokkinos. To the point: Correspondence-driven monocular 3d category reconstruction. *NeurIPS*, 2021. 2
- [32] Nilesh Kulkarni, Abhinav Gupta, and Shubham Tulsiani. Canonical surface mapping via geometric cycle consistency. In *ICCV*, 2019. 2
- [33] Xueteng Li, Sifei Liu, Kihwan Kim, Shalini De Mello, Varun Jampani, Ming-Hsuan Yang, and Jan Kautz. Self-supervised single-view 3d reconstruction via semantic consistency. In *ECCV*, 2020. 2

- [34] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *ICCV*, 2021. 1, 2, 6, 8
- [35] David B Lindell, Julien NP Martel, and Gordon Wetzstein. AutoInt: Automatic integration for fast neural volume rendering. In *CVPR*, 2021. 1
- [36] David B Lindell, Dave Van Veen, Jeong Joon Park, and Gordon Wetzstein. BACON: Band-limited coordinate networks for multiscale scene representation. In *CVPR*, 2022. 1
- [37] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-perfect structure-from-motion with featuremetric refinement. In *ICCV*, 2021. 2, 6, 12, 13
- [38] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE TPAMI*, 2010. 2
- [39] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 2, 6, 12
- [40] Wei-Chiu Ma, Anqi Joyce Yang, Shenlong Wang, Raquel Urtasun, and Antonio Torralba. Virtual correspondence: Humans as a cue for extreme-view geometry. In *CVPR*, 2022. 2
- [41] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*, 2020. 1, 2, 4
- [42] Hans Peter Moravec. *Obstacle avoidance and navigation in the real world by a seeing robot rover*. Stanford University, 1980. 2
- [43] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *CVPR*, 2022. 1, 3
- [44] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry. In *CVPR*, 2004. 2
- [45] David Novotny, Nikhila Ravi, Benjamin Graham, Natalia Neverova, and Andrea Vedaldi. C3dpo: Canonical 3d pose networks for non-rigid structure from motion. In *ICCV*, 2019. 2
- [46] Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion\*. *Acta Numerica*, 2017. 1
- [47] Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion\*. *Acta Numerica*, 2017. 2
- [48] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 5
- [49] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*, 2017. 4
- [50] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv preprint arXiv:2007.08501*, 2020. 5, 6
- [51] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017. 5
- [52] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *ICCV*, 2021. 1, 2, 3, 5, 6, 12
- [53] Olinde Rodrigues. Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire. *J. Math. Pures Appl.*, 1840. 6
- [54] Barbara Roessle and Matthias Nießner. End2end multi-view feature matching using differentiable pose optimization. *arXiv preprint arXiv:2205.01694*, 2022. 2
- [55] Mehdi SM Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lučić, Daniel Duckworth, Alexey Dosovitskiy, et al. Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. In *CVPR*, 2022. 3, 6, 12
- [56] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 6, 7, 12, 13
- [57] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 2, 6, 13
- [58] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1, 2, 5, 6, 7, 12
- [59] Shih-Yang Su, Frank Yu, Michael Zollhöfer, and Helge Rhodin. A-nerf: Articulated neural radiance fields for learning human shape, appearance, and pose. *NeurIPS*, 2021. 1
- [60] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *CVPR*, 2021. 2
- [61] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *CVPR*, 2022. 1
- [62] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, 1999. 2
- [63] Ben Usman, Andrea Tagliasacchi, Kate Saenko, and Avneesh Sud. Metapose: Fast 3d pose from multiple views without 3d supervision. In *CVPR*, 2022. 2, 6, 12
- [64] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017. 4
- [65] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *Proc. ICRA*. IEEE, 2017. 2
- [66] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 1, 2

- [67] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. *arXiv preprint arXiv:2210.04628*, 2022. 3
- [68] Shangzhe Wu, Tomas Jakab, Christian Rupprecht, and Andrea Vedaldi. Dove: Learning deformable 3d objects by watching videos. *arXiv preprint arXiv:2107.10844*, 2021. 2, 3
- [69] Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In *CVPR*, 2020. 3
- [70] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017. 2
- [71] Gengshan Yang, Deqing Sun, Varun Jampani, Daniel Vlasic, Forrester Cole, Huiwen Chang, Deva Ramanan, William T Freeman, and Ce Liu. Lasr: Learning articulated shape reconstruction from a monocular video. In *CVPR*, 2021. 2
- [72] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *CVPR*, 2020. 2
- [73] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Inverting neural radiance fields for pose estimation. In *IROS*, 2021. 2
- [74] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *ECCV*. Springer, 2016. 2
- [75] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *CVPR*, 2021. 1
- [76] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021. 3
- [77] Zhixuan Yu, Jae Shin Yoon, In Kyu Lee, Prashanth Venkatesh, Jaesik Park, Jihun Yu, and Hyun Soo Park. Humbi: A large multiview dataset of human body expressions. In *CVPR*, 2020. 1
- [78] Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. Relpose: Predicting probabilistic relative rotation for single objects in the wild. *arXiv preprint arXiv:2208.05963*, 2022. 1, 2, 5, 6, 7, 12
- [79] Kaifeng Zhang, Yang Fu, Shubhankar Borse, Hong Cai, Fatih Porikli, and Xiaolong Wang. Self-supervised geometric correspondence for category-level 6d object pose estimation in the wild. *arXiv preprint arXiv:2210.07199*, 2022. 2
- [80] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 1

# SparsePose: Sparse-View Camera Pose Regression and Refinement

## Supplemental Material

### A. Further Ablation study

#### A.1. LSTM iterations

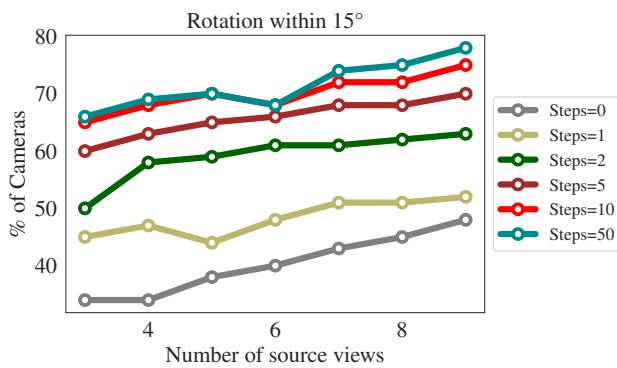


Figure 10. Ablation results varying the number of LSTM steps.

We add an additional ablation experiment over the number of steps required for the LSTM. We vary the number of LSTM iterations between 0 and 50, and report the percentage of cameras that were predicted between  $15^\circ$  of ground truth. We report the results in Figure 10. As previously noted, all the experiments were performed with 10 LSTM iterations, which balances out speed and accuracy of predictions. While we observe slight improvements with 50 LSTM iterations, overall, using 10 LSTM iterations performs similarly.

#### A.2. Timing Analysis

	Time (seconds)
HLOC [56]	38s
COLMAP + SIFT [39, 58]	18s
Pix. Perfect SFM [37]	55s
RelPose [78]	48s
SRT [55]	2.7s
MetaPose [63]	2.6s
SparsePose	3.6s

Table 1. Time (in seconds) to perform registration on a single sequence with 9 source images. To enable fair comparison between all methods, only sequences where *all* baseline methods were able to register all the source images were included in the analysis. Each of the methods are run on the same NVIDIA A6000 for fair comparison.

### A.3. Different rotation threshold

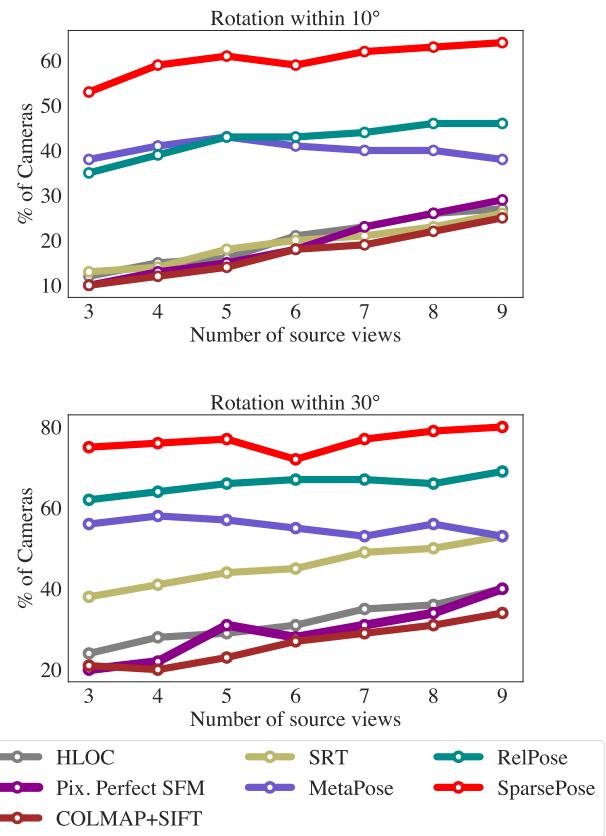


Figure 11. Evaluating on the percentage of cameras accurately predicted with  $10^\circ$  and  $30^\circ$  thresholds.

### B. Baseline details

**MetaPose.** For the MetaPose baseline [63], we used the initialization from SparsePose, and adapt the MetaPose architecture in the officially released code to perform pose updates on the current updates. We do not utilize the human-specific information proposed, since our data is more general than the data used to evaluate MetaPose. We train MetaPose on the same subset of CO3D [52] that was used in training SparsePose.

**Scene Representation Transformer (SRT).** SRT [55] proposes to learn a prior over the 3D geometry from data implicitly by learning a “set-latent scene representation” from sparse or dense images of the scene using transformer encoder and decoder layers. Although SRT does not learn a

direct 3D geometry of the scene, it does learn a prior over the 3D geometry, as it can perform novel-view synthesis. To adapt SRT to our evaluation protocol, we add an additional 3-layer MLP that is trained to predict the relative rotations and translations for the input image sequence. We train the “unposed” version of SRT, and add an additional MLP (with 3-hidden layers) to predict rotations and translations, and we train the method with the same training dataset and loss as SparsePose. For training, we use the same hyperparameters as suggested in the original paper.

**Heirarchical Localization (HLOC).** For HLOC [56], we use the officially released code from <https://github.com/cvg/Hierarchical-Localization>, which uses SuperPoint [14] for generating correspondances and SuperGlue [57] for image matching.

**COLMAP + SIFT.** For the COLMAP baseline with SIFT features, we used the officially released code from HLOC <https://github.com/cvg/Hierarchical-Localization>, which supports SIFT image features.

**RelPose.** For the RelPose baseline, we used the officially released code from <https://github.com/jasonyzhang/relpose>, and trained on the same dataset used to train SparsePose. We use the default RelPose hyperparameters.

**Pixel Perfect SFM.** For the Pixel Perfect SFM [37], we used the officially released codebase from <https://github.com/cvg/pixel-perfect-sfm>.

## C. More implementation details

Hyperparameter	Value
Number of training steps	500,000
Number of source views during step	$\mathcal{U}[3, 9]$
Number of sequences sampled per step	1
Choice of $\mathcal{E}_{\text{init}}$	DINO [8]
Architecture of $\mathcal{E}_{\text{init}}$	ViT-B/8 [15]
Number of heads $\mathcal{T}_{\text{init}}$	8
Number of heads $\mathcal{T}_{\text{refine}}$	2
Number of hidden dim. $\mathcal{T}_{\text{init}}, \mathcal{T}_{\text{refine}}$	2048
Number of hidden layers $\mathcal{N}_{\text{init}}, \mathcal{N}_{\text{pose}}$	3
Number of hidden dim. $\mathcal{N}_{\text{init}}, \mathcal{N}_{\text{pose}}$	512
Activation for $\mathcal{N}_{\text{init}}, \mathcal{T}_{\text{init}}, \mathcal{T}_{\text{refine}}, \mathcal{N}_{\text{pose}}$	GELU
Number of LSTM steps	10
Optimizer	Adam [30]
Learning rate	$10^{-4}$
Learning rate decay iterations	250,000
Learning rate decay factor	10

Table 2. **Hyperparameters and implementation details.** These hyperparameters are shared through all experiments for SparsePose, unless stated otherwise.

## D. Qualitative results

In Figure 12 we provide additional qualitative results with different numbers of source views and visualize the predicted camera poses by our method compared to baselines. We also include additional qualitative novel-view synthesis results for different categories over different numbers of source views in Figure 13 and Figure 14. In both cases, we see that SparsePose predicts more accurate camera poses, resulting in higher quality novel-view synthesis compared to other baselines.

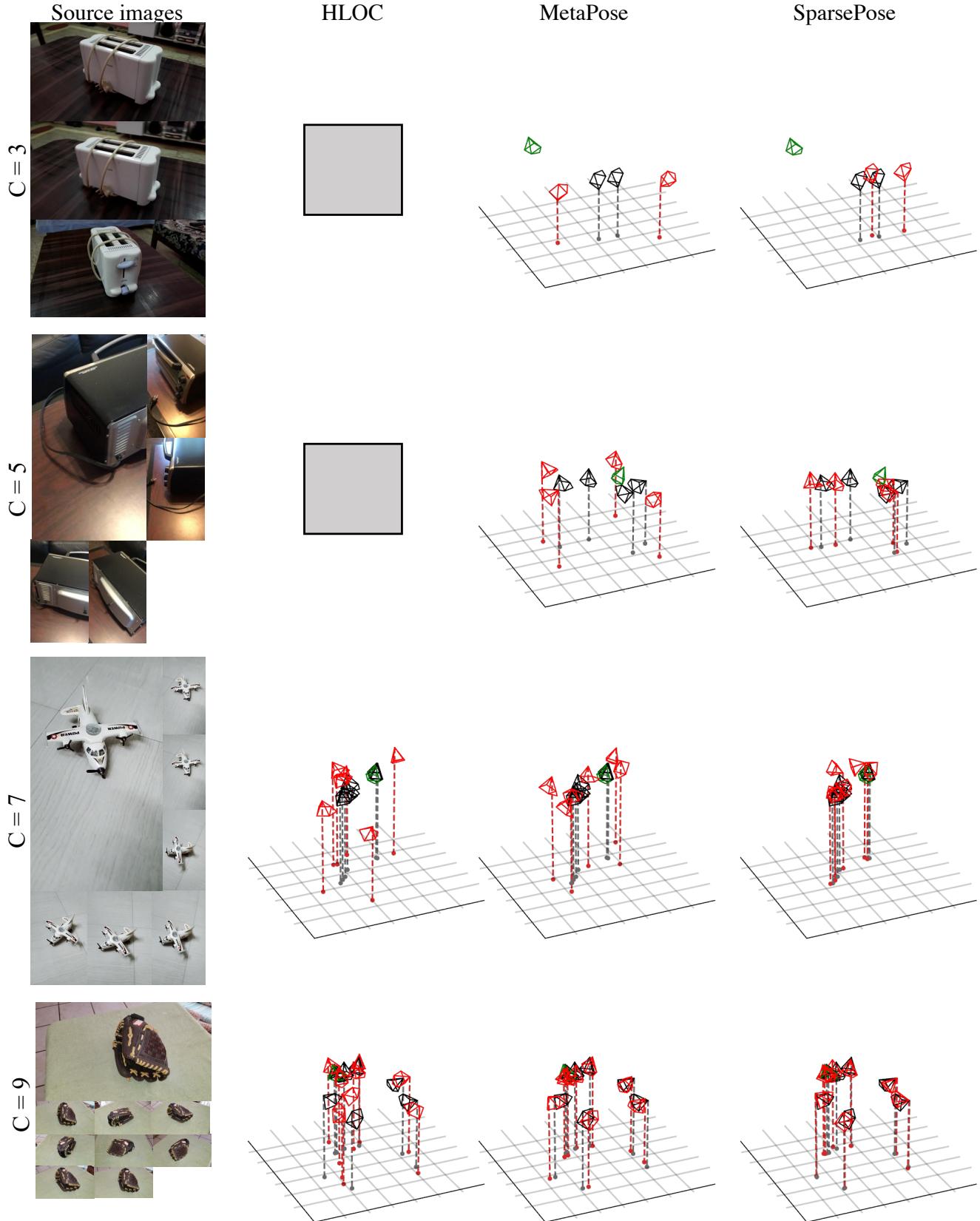


Figure 12. **More qualitative results for the predicted camera poses.** The camera centers are projected to the  $x - y$  plane for easy visual comparison. The ground truth poses are shown in black, predicted poses in red, and the first camera for each sequence (used to align predictions) in green. Gray boxes indicate failure to converge.



Figure 13. More qualitative renders from a sparse set of unposed images.



Figure 14. More qualitative renders from a sparse set of unposed images.