The 8-Point Algorithm as an Inductive Bias for Relative Pose Prediction by ViTs

Chris Rockwell, Justin Johnson, David F. Fouhey
University of Michigan

Abstract

We present a simple baseline for directly estimating the relative pose (rotation and translation, including scale) between two images. Deep methods have recently shown strong progress but often require complex or multi-stage architectures. We show that a handful of modifications can be applied to a Vision Transformer (ViT) to bring its computations close to the Eight-Point Algorithm. This inductive bias enables a simple method to be competitive in multiple settings, often substantially improving over the state of the art with strong performance gains in limited data regimes.

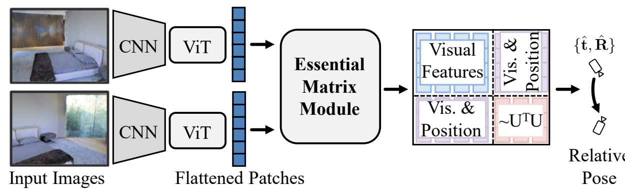

Figure 1. We propose three small modifications to a ViT via the Essential Matrix Module, enabling computations similar to the Eight-Point algorithm. The resulting mix of visual and positional features is a good inductive bias for pose estimation.

1. Introduction

Estimating the relative pose between two images is a fundamental vision problem [17], with applications including 3D understanding [22, 34, 57] and extended reality [30, 35, 38, 72]. Early work focused on robust [14] fitting of models [17, 19, 31, 42] on detected correspondences [2, 32, 54] between the images. This strategy can fail catastrophically with poor correspondence, which is especially frequent in the *wide baseline* setting, when the images have a substantial pose difference. Moreover, even when it is successful, it cannot recover the *scale* of the translation [17]. The situation is often improved in practice by obtaining more images (e.g., SfM [57] and SLAM [41]), or sensors like IMUs [15, 16, 26] and RGBD [11, 69]. Nonetheless, people routinely infer relative pose from two ordinary images with a wide baseline, and whole industries like real estate depend on this ability. Rather then use extra sensors or images, humans integrate cues like correspondence, familiar object size, and priors on scenes. This paper investigates such an ability, to estimate relative pose, including rotation and translation with scale, from two ordinary images.

Based on these observations, there has been much work applying learning to the problem. One line of attack [8, 10, 55, 59, 62] has been to follow the classic pipeline and replace classic correspondence methods [2, 32, 54] with learned ones. This approach is appealing since the learning method finds correspondence, an especially thorny challenge in the wide-baseline setting, and the conversion of correspondences to pose is done by a provably correct method [31, 42]. However, it comes at a cost of inheriting the Essential Matrix's intrinsic scale ambiguity, leading to translation-up-to-scale. Thus, another line of work treats relative camera pose estimation as a learning problem [5, 12, 22, 48]. These approaches have shown promise in the wide-baseline setting, but often involve multiple stages [7, 22], are not as performant as correspondence-based techniques in the settings we try [12, 48], or do not recover a translation scale [5, 7]. Moreover, since these methods learn an end-to-end mapping from images to camera pose with few inductive biases, they are often data hungry.

We propose a Vision Transformer (ViT) [9, 63] approach that estimates rotation and *translation with scale* in one forward pass by integrating the problem's structure implicitly as an inductive bias. We reconcile the Eight-Point Algorithm [19, 31] with ViTs by showing that a ViT forward pass can be made close to [19, 31] by three minor modifications: (1) bilinear attention [24] instead of attention [63]; (2) quadratic position encodings; and (3) dual-softmax [53, 59, 62] instead of softmax. These modifications are put in one module, the *Essential Matrix Module (EMM)*, that we place atop an otherwise ordinary ViT, as shown in Fig. 1. The EMM gives an inductive bias by providing positional features that approximate a key step of [31], visual features, and features that mix the two.

We attach the Essential Matrix Module to the end of an ordinary ViT [9] described in §3. We train and evaluate this ViT on multiple relative camera pose estimation tasks and datasets as described in §4 and compare with the state of the art for each on challenging datasets like Matterport3D [6], InteriorNet [29], and StreetLearn [39]. Our experiments on rotation+translation (§4.2) and rotation (§4.3) demonstrate: (1) that our simple approach outperforms (or occasionally matches) multiple alternate networks including concatenation [12, 48] and correlation volume methods [5, 22], techniques based on feature correspondence [8, 32], and techniques trained to optimize a full 3D reconstruction [22]; (2) that each component of the modification is important, as shown by extensive ablations; and (3) that the EMM improves data efficiency by substantially boosting performance in moderate data regimes (§4.4), suggesting that epipolar geometry is a good inductive bias.

2. Related Work

Our work introduces a learning-based approach to relative pose estimation by modifying vision transformers to perform computations similar to the Eight Point Algorithm.

Classic Work. Relative pose estimation from an image pair is a sufficiently broad problem to preclude a full account. We refer readers to [17], and focus on the closest works, which all follow a strategy of solving for pose given correspondences from local descriptors [2, 32, 54]. We revisit the 8-point algorithm [19, 31] that maps correspondences to an Essential Matrix, which was invented by Longuet-Higgins and extended to Fundamental Matrices by [13, 18]. While it often replaced by other approaches that use fewer correspondences (e.g., [27, 42]), much of the 8-point algorithm's structure are calculations that we show can be done by Transformers. In our wide baseline setting (i.e., large pose difference), historically there are alternate descriptors and specialized techniques [36, 40, 46].

Learned Pose Estimation. Given the difficulties associated with optimization on correspondence, multiple lines of work aim to improve the pipeline with learning. For instance, many methods improve detectors and descriptors [8, 10, 62] or correspondence estimation [3, 47, 50, 55, 59, 71]. These works typically turn correspondences to pose with the Essential Matrix [31, 42], which makes it impossible to recover translation scale without additional signals [17]. In contrast, our proposed work learns a direct mapping without explicitly constructing an Essential Matrix, and therefore recovers scale using image-based cues. We note that our components are also often used in correspondence work [59, 62]; here, we use them directly for pose and show a close relationship between ViTs and [31].

Our method is closer to work that learns a mapping from images to pose. This area of research is relatively newer, and has become more complex over time (e.g., early networks concatenated data from two images [12, 28, 37], which has been supplanted by correlation volumes [5, 22]). These approaches are often data and compute hungry [7], use multiple stages (e.g., discrete/continuous optimization in [22], two-stage networks in [7, 64]), and use little of the structure of pose estimation. In contrast, our approach brings ViT computations close to this structure, which we hypothesize helps use the data more effectively.

SLAM, SfM, and RGBD. Given the difficulties of pose estimation from two images, a wealth of other approaches have been tried that modify the problem. The most common solution is to use more images with typically high overlap, e.g. via Structure-from-Motion [57], SLAM [41, 60, 65] or localization [4, 49, 67]. In contrast, we aim to solve the two-view, wider-baseline case. Other solutions include adding sensors like an IMU [15, 16, 26] or depth data [11, 68, 69]; our approach relies only on RGB data.

Vision Transformers and Inductive Biases. Large parts of our proposed approach follow a basic recipe for Vision Transformers [9, 63]. These have emerged as a competitor to convolutional neural networks in the past few years, and we refer interested readers to [23, 43] for a more thorough summary. Our work shows that small modifications of the pipeline brings the computations close those of [31]. This is part of a broader trend of injecting geometric inductive biases to networks via layers [45] or token engineering [70].

3. Approach

Our goal is to map two overlapping images to a relative camera pose *including translation scale*, or a rotation $\mathbf{R} \in \mathrm{SO}(3)$ and translation $\mathbf{t} \in \mathbb{R}^3$. This task requires both robustness to large view changes with limited correspondence, and handling scale ambiguity. We propose a simple approach that fuses ideas from classical multi-view geometry with large-scale learning.

At the heart of our approach is a transformer with small critical changes that mimic a computation used in the Eight Point Algorithm [19, 31]. These changes include bilinear attention [24], dual-softmaxes [53, 59, 62], and an explicit positional encoding. We first analyze the relationship between the Eight Point Algorithm and and an alternate setup that is more amenable to computation by a transformer (§3.1). We then describe how we operationalize this by introducing our base transformer and our Essential Matrix Module (§3.2). We conclude by analyzing the learnability of this function with synthetic experiments (§3.3).

3.1. Transformers and the Eight Point Algorithm

The Fundamental and Essential matrices can be obtained from correspondences via the Eight-point algorithm [19, 31, 18, 13]. As input, one assumes N correspondences $[u_i, v_i] \leftrightarrow [u'_i, v'_i]$. With known intrinsics $\mathbf{K}$, one represents the locations of the correspondences with normalized points $\mathbf{x}_i \equiv \mathbf{K}^{-1}[u_i, v_i, 1]^\top$ and $\mathbf{x}'_i \equiv \mathbf{K}^{-1}[u'_i, v'_i, 1]^\top$ and recovers an Essential matrix ($\mathbf{E}$); if $\mathbf{K}$ is unknown, one uses standard homogeneous coordinates (i.e., $\mathbf{x}_i = [u_i, v_i, 1]^\top$) and recovers a Fundamental matrix ($\mathbf{F}$). Since we have the intrinsics, we will refer to the Essential matrix.

The Eight-Point Algorithm constructs a matrix $\mathbf{U} \in \mathbb{R}^{N\times 9}$ whose ith row $\mathbf{U}_{i,:}$ is the Kronecker product of the correspondences, or $\mathbf{x}_i \otimes \mathbf{x}'_i$. The matrix $\mathbf{U}^\top\mathbf{U} \in \mathbb{R}^{9\times 9}$ captures the information needed to estimate the Essential matrix: one computes the eigenvector corresponding to the smallest eigenvalue of $\mathbf{U}^\top\mathbf{U}$, reshapes the vector, and makes the reshaped matrix rank deficient. The resulting matrix $\mathbf{E}$ does not uniquely define the relative pose, but rather a family of solutions comprising two rotations $\mathbf{R}$ and $\mathbf{R}'$ and a translation direction (that can be scaled by any $\lambda \neq 0$).

Careful minor modifications of Transformer can enable the computation of the entries of $\mathbf{U}^\top\mathbf{U}$. We assume the transformer is given a set of P patches at locations $\{\mathbf{p}_j\}_{j=1}^P$ where every correspondence is at one of the patches. In addition to using these locations directly, we further define a 6D basis expansion $\phi([u, v, 1]) = [1, u, v, uv, u^2, v^2]$ that we apply to each patch to yield a matrix $\mathbf{\Phi} \in \mathbb{R}^{P\times 6}$ such that $\mathbf{\Phi}_{j,:} = \phi(\mathbf{p}_j)$. Finally, to represent correspondences implicitly, we define an indicator matrix $\mathbf{A} \in \{0, 1\}^{P\times P}$ such that $\mathbf{A}_{j,k} = 1$ if and only if points $\mathbf{p}_k$ and $\mathbf{p}_j$ are in correspondence and 0 otherwise.

Our key observation is that each unique entry of $\mathbf{U}^\top\mathbf{U} \in \mathbb{R}^{9\times 9}$ is in the matrix $\mathbf{\Phi}^\top\mathbf{A}\mathbf{\Phi} \in \mathbb{R}^{6\times 6}$. While this more compact form is not amenable to eigenvector analysis, it is all the information needed for a learned estimator. A derivation appears in the supplement, but the two critical steps are: first, to decompose the matrix as an explicit sum over correspondences $\mathbf{U}^\top\mathbf{U} = \sum_{i=1}^N \mathbf{U}_{i,:}^\top\mathbf{U}_{i,:}$ and rewrite it implicitly with $\mathbf{A}$; and second, that the 36 *unique* entries in $\mathbf{U}_{i,:}^\top\mathbf{U}_{i,:}$ can be generated from $\phi(\mathbf{x}_i)\phi(\mathbf{x}'_i)^\top$.

The remaining step is estimation of $\mathbf{R}$ and $\mathbf{t}$ from $\mathbf{U}^\top\mathbf{U}$. MLPs are universal approximators [21], but a number of things make this easier in practice. First, often one aims to solve a subset of problems from a distribution, rather than *all* instances. Additionally, one is also using a wealth of alternate image-based cues. In addition to facilitating learning, the network can use these cues to resolve the ambiguities intrinsic to $\mathbf{E}$: for instance, the scale ambiguity can be resolved implicitly via recognizing familiar objects. We explore the learnability of this function in §3.3.

Together, this suggests that transformers estimating $\mathbf{R}, \mathbf{t}$ may benefit from a few small modifications. The crux is that the computation of $\mathbf{\Phi}^\top\mathbf{A}\mathbf{\Phi}$, using quadratic position encodings per patch in $\mathbf{\Phi}$ and a correspondence indicator in $\mathbf{A}$ mimics the computation of the entries of $\mathbf{U}^\top\mathbf{U}$. Thus, a network may benefit from having $\mathbf{\Phi}^\top\mathbf{A}\mathbf{\Phi}$ during prediction. Moreover, $\mathbf{A}$ also should be able to represent unmatched correspondences (i.e., $\sum_{k=1}^P \mathbf{A}_{j,k}\approx 0$). Finally, we stress that the model should also contain features beyond $\mathbf{\Phi}$ to help learning and resolve ambiguities such as scale.

3.2. Putting things In Practice

Our approach consists of two components. The main component is an *Essential Matrix Module*, which maps from P, D-dimensional transformer tokens, one for each of the P patches in the image, to a feature that is used to predict $\mathbf{R}$ and $\mathbf{t}$. This module is added to a standard ViT [9] backbone that maps images to a set of tokens. Our backbone deliberately follows a standard vision transformer recipe [9, 63]: we see backbone innovations as orthogonal to innovations in the mapping from tokens to outputs. On the other hand, our Essential Matrix Module must contain critical modifications.

Backbone and Setup. Our backbone consists of two main components that function as a learned mapping from an image to a $\mathbb{R}^{P\times D}$ matrix of features, one per patch. The first component is an encoder that uses the first blocks from a standard ResNet-18 [20], which helps the network extract good features per-patch. On top of this, we use blocks from a standard ViT [9] (ViT-Tiny) to map the the patch features to our final set of P D-dimensional tokens. Since the architectures have different feature sizes, we bridge them with a ResNet block that maps the feature dimensions. A full network description appears in the supplemental.

Standard Transformer Model. The canonical ViT maps a set of patches from one image to an output embedding used for classification. Given a patch embedding, this entails computing query, key and value matrices $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{P\times D}$ followed by $\mathrm{softmax}(\mathbf{Q}\mathbf{K}^\top)\mathbf{V}$. To avoid notational clutter, we drop the usual [63] scaling factor of $1/\sqrt{D}$ inside the softmax here, and in all other softmax references.

For our case of two images, there are two sets of matrices, namely $\mathbf{Q}_1, \mathbf{K}_1, \mathbf{V}_1 \in \mathbb{R}^{P\times D}$ for image 1 and $\mathbf{Q}_2, \mathbf{K}_2, \mathbf{V}_2 \in \mathbb{R}^{P\times D}$ for image 2. The simplest cross attention is to concatenate cross-attention per-image, or

$$[\mathrm{softmax}(\mathbf{Q}_1\mathbf{K}_2^\top)\mathbf{V}_2, \mathrm{softmax}(\mathbf{Q}_2\mathbf{K}_1^\top)\mathbf{V}_1]. \tag{1}$$

This approach produces good results, but a few minor modifications can substantially improve its performance.

Essential Matrix Module. We propose three changes to Eqn. 1 that help approximate the entries of $\mathbf{U}^\top\mathbf{U}$. These are shown in Fig. 2.

Bilinear Attention and Quadratic Position Encodings. We apply bilinear attention [24] to the values and quadratic positional encodings, or

$$[\mathbf{V}_2, \mathbf{\Phi}]^\top \mathrm{norm}(\mathbf{Q}_1\mathbf{K}_2^\top)[\mathbf{V}_2, \mathbf{\Phi}] \in \mathbb{R}^{(D+6)\times(D+6)} \tag{2}$$

where $\mathbf{\Phi} \in \mathbb{R}^{P\times 6}$ contain the positional encodings $[1, u, v, uv, u^2, v^2]$ from §3.1 and norm is a normalization for the raw attention scores. Thus $\mathbf{A} = \mathrm{norm}(\mathbf{Q}_1\mathbf{K}_2^\top)$. To use both images, we also compute Eqn. 2 substituting in $\mathbf{Q}_2$, $\mathbf{K}_1$, and $\mathbf{V}_1$ and concatenate the results, leading to a $(2D^2 + 24D + 72)$-dimensional feature per attention head.

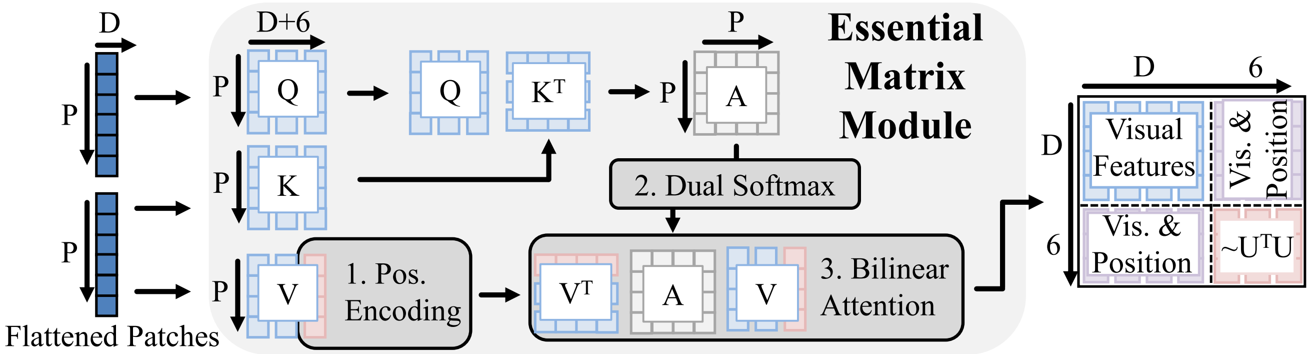

Figure 2. **Essential Matrix Module.** We make three small changes to standard ViT Cross-Attention: (1) appending positional encodings to Values, (2) applying a dual softmax on Affinities, and (3) applying bilinear attention.

Rationale. If $\mathbf{A} = \text{norm}(\mathbf{Q}_1\mathbf{K}_2^\top)$ correctly indicates correspondence, then this computation makes the bottom-right 6×6 submatrix of Eqn. 2 contain the entries of $\mathbf{U}^\top\mathbf{U}$. The top-left $D \times D$ submatrix are visual features; the rest mix position and visual features. These image features are important for scale estimation, since $\mathbf{U}^\top\mathbf{U}$ does not provide information about scale. They may also *implicitly* contain position encodings (e.g., due to convolutions using zero-padding as a proxy for image location). In practice, $\mathbf{U}^\top\mathbf{U}$ is followed by neural layers and thus does not need to match true $\mathbf{U}^\top\mathbf{U}$; though in the supplement we find non-zero rank correlation with ground truth.

Dual-softmax. The above is exact when $\mathbf{A}$ is a correspondence indicator matrix. While attention makes this impossible to ensure exactly, we help more closely approximate it with a dual-softmax [53, 59, 62], or set $\text{norm}(\mathbf{Q}_1\mathbf{K}_2^\top)$ to

$$\text{softmax}(\mathbf{Q}_1\mathbf{K}_2^\top, 1) \odot \text{softmax}(\mathbf{Q}_1\mathbf{K}_2^\top, 2), \qquad (3)$$

where $\text{softmax}(\cdot, k)$ applies softmax across the k-th axis.

Rationale. Traditional attention normalizes the matrix product $\mathbf{Q}_1\mathbf{K}_2^\top \in \mathbb{R}^{P\times P}$ by a single softmax, $\text{softmax}(\mathbf{Q}_1\mathbf{K}_2^\top, 1)$, forcing $\sum_{k=1}^{P} \mathbf{A}_{j,k} = 1$. This constraint means that $\mathbf{A}$ cannot indicate correspondence for patches without matches where $\sum_{k=1}^{P} \mathbf{A}_{j,k} = 0$. At best, attention can be a uniform distribution; at worst, attention can latch onto a random correspondence. In all cases, all patches contribute equally to the final product in Eqn. 2. The network can mitigate this by making non-matching attention uniformly distributed and $\frac{1}{P}\sum_{k=1}^{P} \mathbf{V}_{k,:} = 0$, but this strategy does not work for parts of $\mathbf{\Phi}$, e.g., the u^2 term is non-negative and usually positive.

A dual-softmax suppresses non-matching patches while not altering bidirectional matches. If the attention to-and-from patch j is uniformly distributed, then the total attention $\sum_{k=1}^{P} \mathbf{A}_{j,k}$ is $\frac{1}{P}$ instead of 1 in the normal softmax case. On the other hand, if patch j and patch k both match well, then the attention approaches 1. Then, even if all but one patches have no match, their total contribution is smaller ($\frac{P-1}{P}$) than even a single bidirectionally matching patch (1). Thus, the varying weighting helps suppress the contributions of patches without matches. While the form of $\mathbf{A}$ intrinsically makes the computation an approximation, we stress that the consumer of $\mathbf{\Phi}\mathbf{A}\mathbf{\Phi}^\top$ is a learned module and may be able to learn around approximation errors, especially with vision features.

Pose Regressor. Given essential matrix encodings, we regress pose using a 2 hidden layer MLP. We predict translation in real units, and predict rotation in quaternions, normalizing so scale is one. We train only using a l1 geodesic loss on pose where the geodesic loss is the magnitude of the vector between predicted and ground truth pose.

3.3. Synthetic Validation

While our approximation of $\mathbf{U}^\top\mathbf{U}$ can be understood analytically, one critical component is the learned mapping from $\mathbf{U}^\top\mathbf{U}$ to $\mathbf{R}$, $\mathbf{t}$ that is done by the *Pose Regressor*. To better understand the learnability of the function, we show the method on synthetic examples with the entries of $\mathbf{U}^\top\mathbf{U}$ but no visual features. Our scenes consist of points uniformly sampled inside a sphere with center $\sim\text{Unif}(-\frac{1}{2}, \frac{1}{2})$ and radius $\sim\text{Unif}(\frac{1}{2}, \frac{3}{2})$. We sample camera rotations and translations from distributions that we vary to analyze the learnability of the problem. For each pair of views with sufficient overlap (100 of 10K sampled 3D points projecting to the images), we compute $\mathbf{U}^\top\mathbf{U}$, which is used as a feature for pose estimation by a MLP (details in supplement).

We analyze two tasks. The first is *Translation*, or estimating the generating $\mathbf{t}$; due to scale-ambiguity, we assume $||\mathbf{t}||_2 = 1$ and $\mathbf{t}_z > 0$. We quantify errors by the angle between the estimated and true translation. The second is *Rotation*, or estimating the rotation that generated the data, which forces the network to resolve the usual rotation ambiguity of $\mathbf{E}$. We quantify errors by the rotation geodesic.

We try four distributions. In *3D* $\mathbf{R}$ is sampled via uniformly distributed Euler angles, and $\mathbf{t} \sim \text{Unif}(-1, 1)$. The

Table 1. **Synthetic Validation.** We report the median angular error across two tasks (translation & rotation) and four datasets (in decreasing difficulty: 3D, 2D Large/Medium/Small (L/M/S).

	Translation (°)				Rotation (°)			
	3D	2DL	2DM	2DS	3D	2DL	2DM	2DS
MLP	18.4	5.6	3.0	1.8	33.5	3.6	1.8	0.7
Chance	64.0	49.1	47.9	47.9	125.3	22.2	4.8	1.0

next three are *2D Small/Medium/Large*, consisting of 2D motion primarily in the xz plane with varying amounts of rotation variance: $\mathbf{R}$ is sampled from Normally distributed Euler angles with rotation mainly in y ($y \sim N(0, r)$) and $x, z \sim N(0, \frac{r}{20})$) where $r = 1, 5, 25°$ for small, medium, and large. Translation is mainly in the xz plane $\mathbf{t} \sim N(0, [\frac{1}{3}, \frac{1}{60}, \frac{1}{3}])$. To avoid epipolar degeneracies with no translation, we require $||\mathbf{t}|| \geq \frac{1}{2}$.

We report results in Table 1 for models trained on 100K samples using only $\mathbf{U}^\top\mathbf{U}$ as features, comparing to chance for context. We compare models trained on 100K samples. Even when trained on 100K samples and estimating a general problem case, the networks learn the function. Once the data is even moderately constrained (2D Large), relative errors drop considerably. This suggests that the function is especially learnable under more constrained rotations.

3.4. Implementation Details

Full implementation details appear in the supplemental, and we will release code for reproducibility. Our encoder is a pretrained ResNet-18, which we truncate to only use the first two of four modules, producing a $24 \times 24 \times 128$ feature map; we use an additional Residual block to map to feature size of 192 for the ViT. We use the Timm [66] ViT implementation, and use ViT-Tiny with a truncated depth of 5 plus our Essential Matrix Module. Outside of the proposed changes, our Essential Matrix Module follows a standard Cross-Attention Transformer Block architecture and normalization [33]. Positional encoding locations utilize known intrinsics, $\mathbf{x}'_i \equiv \mathbf{K}^{-1}[u'_i, v'_i, 1]^\top$, a manner similar to [64]. Each head of the Essential Matrix Module produces a 64-D feature, which is 70-D after concatenating the position encodings. With 3 heads, and the bilinear attention done once per image, this results in $3 \times 70^2 \times 2 = 29$K features. We map this large feature to a hidden size of 512 for two hidden layers in our MLP before regressing 7D pose. We implement using PyTorch [44] and use the LieTorch [61] extension for backpropogation of geodesic losses on quaternions. We use learning rate of 5e-4 and train using Adam [25] optimizer and 1cycle scheduler [58] for 120k iterations with a batch size of 60 split over 10 GTX 1080 Tis, which takes about 1 day.

4. Experiments

We now evaluate the proposed method's ability to estimate relative pose in comparison to the state of the art in two settings that share common metrics and evaluation settings (§4.1). Our first task (§4.2) is wide baseline rotation *and translation* estimation, or estimating a rotation in $\mathrm{SO}(3)$ and translation in $\mathbb{R}^3$ (i.e., including a scale). The second task (§4.3) is wide baseline rotation, or estimating a rotation in $\mathrm{SO}(3)$ but no translation. Finally, a crucial argument for our approach is that the modifications of the transformer architecture serve as an inductive bias for the network. We examine this empirically with experiments on substantially reduced data that test data efficiency (§4.4).

4.1. Metrics and Evaluation

For each method, we compute the rotation error (defined as the rotation geodesic to the ground-truth) and translation error (defined as the usual Euclidean distance to the ground truth), and aggregate three summary statistics: the *mean*, the *median*, and the *percent of errors within a threshold* that is task-specific (e.g, 30°) and will be described with each dataset. These capture different aspects of the problem. Specifically, due to symmetries in the data, pose estimation errors are often not unimodally distributed. Instead, often many results are highly accurate and a few are wrong by 90° or 180°. The median error captures what a typical prediction error is like and is outlier robust; the mean is the straight average and is therefore sensitive to outliers; the percent within a threshold captures a sense of how many predictions are "reasonable" for some threshold.

4.2. Wide Baseline Rotation and Translation

We begin by evaluating on our full problem, namely estimating a rotation in $\mathrm{SO}(3)$ and translation, *including scale*, in $\mathbb{R}^3$. We follow the setup of [22] to enable comparison with a variety of existing work and published baselines.

Dataset. We use data from Matterport3D [6] consisting of pairs of images with limited overlap (mean 2.3m translation, 53° rotation). This dataset is a re-rendering of a real capture, using the Habitat [56] system. The train/val/test set of the dataset consist of 32K/5K/8K image pairs, respectively. Following [22], we set the threshold for percent within a threshold to 30° for rotation and 1m for translation.

Baselines and Ablations. Our primary comparison is the Sparse Planes method of [22], a strong baseline estimating both rotation and translation (including scale). Sparse Planes does joint reconstruction and pose estimation and consists of: initial reconstruction and camera estimation, discrete optimization, and a bundle-adjustment on SIFT features [32] extracted from texture that has been made fronto-parallel. The final step adds substantial complexity, so we compare to (*SparsePlanes [22] No Bundle*) as well, which

Table 2. **Translation and Rotation Performance on Matterport.** Ours is best among methods producing translation scale. All baselines supervise depth except [22] (Camera Br) and Ours.

Method	Translation (m)			Rotation (degrees)		
	Med.↓	Avg.↓	≤1m↑	Med.↓	Avg.↓	≤ 30↑
[52] + [51]	3.34	4.00	8.3	50.98	57.92	29.9
Assoc.3D [48]	2.17	2.50	14.8	42.09	52.97	38.1
[22] (Camera Br)	0.90	1.40	55.5	7.65	24.57	81.9
[22] (No Bundle)	0.88	1.36	56.5	7.58	22.84	83.7
[22] (Full)	**0.63**	1.25	66.6	7.33	22.78	83.4
PlaneFormers [1]	0.66	1.19	66.8	5.96	22.20	83.8
Ours	0.64	**1.01**	**67.4**	8.01	19.13	85.4
SuperGlue [55]	-	-	-	3.88	24.17	77.8
LoFTR [59]	-	-	-	**0.71**	**11.11**	**90.5**

Table 3. **Essential Matrix Module Ablations on Matterport.** All three components of the Essential Matrix Module yield meaningful improvement across metrics.

Method	Translation (m)			Rotation (degrees)		
	Med.↓	Avg.↓	≤1m↑	Med.↓	Avg.↓	≤30↑
CNN Pose Regressor	1.53	1.83	28.6	31.31	45.05	48.8
+ViT	1.47	1.79	30.1	29.9	43.33	50.1
+Bilinear Attention	1.13	1.49	44.5	9.76	28.36	73.1
+Dual Softmax	0.70	1.06	64.8	8.62	21.23	83.3
Full	**0.64**	**1.01**	**67.4**	**8.01**	**19.13**	**85.4**

omits the final bundle adjustment, but still requires optimization. We also compare to the standalone pose estimation branch as *(Sparse Planes [22] Camera Branch)*. In addition, we compare to concurrent work [1] which closely builds off of Sparse Planes.

We next report three baselines used by [22]. The first is (*Associative 3D [48] camera branch*), which is an improved version of RPNet [12]. The second is the reconstruction-based RGBD odometry method of Raposo et al. [52] applied to [51]. Third, we compare with (*SuperGlue [55]*), using the settings from [22]. In addition, we compare to LoFTR [59]. Like SuperGlue, LoFTR supervises correspondences, and therefore requires depth supervision in addition to pose, and cannot recover translation scale.

Finally, we compare with four ablations that test the contributions of our method. All methods use the same MLP Regressor, and full descriptions of these appear in the supplement. The first is (*CNN Pose Regressor*), which predicts pose from concatenated base CNN extracted features. This gives a sense of how a simple method does. The second is (*+ViT*), which adds a ViT that is capped with standard attention (Eqn. 1) on top of the backbone. This tests the contribution of a ViT *without* the Essential Matrix Module. The third is (*+Bilinear*) which replaces standard attention (Eqn. 1) with bilinear attention, but without dual-softmax and quadratic positional encodings. Finally, we report (*+Dual Softmax*), which adds dual softmax.

Quantitative Results. We report results in Table 2. Joint prediction of rotation and translation (including scale) on wide-baseline pairs is a challenging problem. Non-trivial

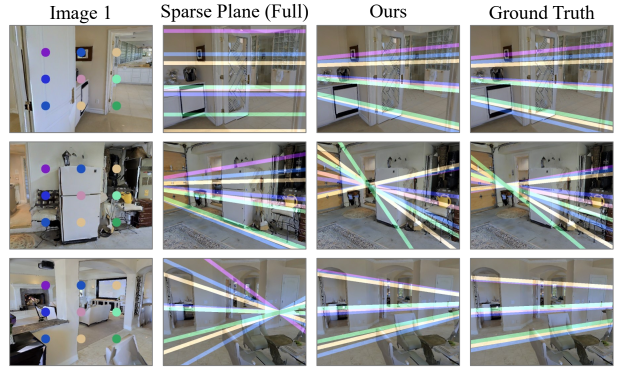

Figure 3. **Epipolar Lines on Matterport.** Our predictions better match true pose, particularly on large view changes.

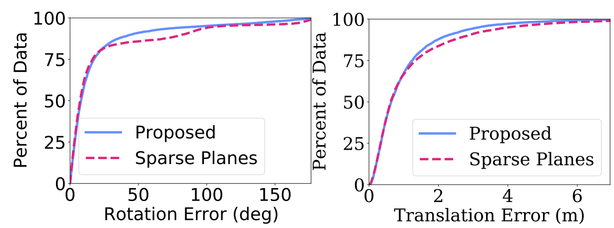

Figure 4. **Error CDFs on Matterport**. The proposed approach shows increased robustness to large view changes.

methods [48, 51, 52] have less than 40% of predictions within 30° of true rotation, and less than 20% of errors within 1m of true translation. Methods that are most competitive (LoFTR [59], SuperGlue [55], [22]) require depth supervision in addition to pose, while the best rotation results ([59], [55]) are produced by correspondence-based methods not predicting translation scale. Of methods producing translation scale, ours typically performs best, and it outperforms SuperGlue in both average rotation and percentage within 30°.

Ablations, shown in Table 3, show the reasons for success. As with [48, 51, 52], CNN and ViT models struggle at the task. Adding bilinear attention reduces errors tremendously – reducing median rotation error by two thirds – while the dual softmax reduces errors further significantly. Adding positional encodings further improves performance across all measurements.

Analysis. Qualitative results, in Figure 3, are consistent with quantitative findings. Namely, predicted pose more closely matches ground truth on difficult examples, resulting in much better mean performance than baselines. Error vs. view change is analyzed further in the Supplemental. Figure 4 displays error CDFs on Matterpor Compared to the most competitive baseline [22] (Full), the proposed method has fewer very large errors.

Table 4. **Rotation Performance on InteriorNet and StreetLearn.** We train and evaluate on only overlapping images. "*" indicates the method sometimes failed to produce pose estimation; errors were calculated only on successful image pairs. Gray text indicates failure over 50% of test pairs. The proposed method outperforms alternatives almost universally and often significantly.

		InteriorNet			InteriorNet-T			StreetLearn			StreetLearn-T		
Overlap	Method	Avg (° ↓)	Med. (° ↓)	10 (% ↑)	Avg (° ↓)	Med. (° ↓)	10 (% ↑)	Avg (° ↓)	Med. (° ↓)	10 (% ↑)	Avg (° ↓)	Med. (° ↓)	10 (% ↑)
Large	SIFT* [32]	6.09	4.00	84.86	7.78	2.95	55.52	5.84	3.16	91.18	18.86	3.13	22.37
	SuperPoint* [8]	5.40	3.53	87.10	5.46	2.79	65.97	6.23	3.61	91.18	6.38	1.79	16.45
	Reg6D [73]	5.43	3.87	87.10	10.45	6.91	67.76	3.36	2.71	97.65	12.31	6.02	69.08
	Cai *et al.* [5]	1.53	1.10	99.26	**2.89**	**1.10**	97.61	1.19	1.02	99.41	9.12	2.91	87.50
	Ours	**0.48**	**0.40**	**100.00**	2.90	1.83	**97.91**	**0.62**	**0.52**	**100.00**	**4.08**	**2.43**	**90.13**
Small	SIFT* [32]	24.18	8.57	39.73	18.16	10.01	18.52	16.22	7.35	55.81	38.78	13.81	5.68
	SuperPoint* [8]	16.72	8.43	21.58	11.61	5.82	11.73	19.29	7.60	24.58	6.80	6.85	0.95
	Reg6D [73]	17.83	9.61	51.37	21.87	11.43	44.14	7.95	4.34	87.71	15.07	7.59	63.41
	Cai *et al.* [5]	6.45	1.61	95.89	10.24	**1.38**	89.81	2.32	1.41	98.67	13.04	3.49	84.23
	Ours	**1.81**	**0.94**	**99.32**	**4.48**	2.38	**96.30**	**1.46**	**1.09**	**100.00**	**9.19**	**3.25**	**87.70**

Table 5. **Rotation Ablations InteriorNet and StreetLearn.** (Second best underlined). The ViT significantly improves over CNN only. Components of the proposed model perform in different settings, but the full model is often best and typically competitive with the best ablation, while ablations sometimes do poorly (Bilinear Att. on InteriorNet Small, ViT on StreetLearn Small).

		InteriorNet-T			StreetLearn-T		
Overlap	Method	Avg (° ↓)	Med. (° ↓)	10 (% ↑)	Avg (° ↓)	Med. (° ↓)	10 (% ↑)
Large	CNN Pose Regressor	5.29	2.6	89.85	15.25	10.00	50.00
	+ViT	2.99	1.64	96.72	**3.52**	2.56	**94.74**
	+Bilinear Attention	3.25	**1.49**	97.31	4.73	2.68	92.76
	+Dual Softmax	6.03	1.63	93.43	4.39	2.64	91.45
	Full	**2.90**	1.83	**97.91**	4.08	**2.43**	90.13
Small	CNN Pose Regressor	19.79	4.05	69.44	29.95	15.22	34.07
	+ViT	5.43	2.00	94.75	12.93	**3.16**	84.86
	+Bilinear Attention	8.54	**1.79**	90.43	**8.70**	3.41	**89.59**
	+Dual Softmax	10.44	1.96	89.51	10.74	3.24	87.07
	Full	**4.48**	2.38	**96.30**	9.19	3.25	87.70

4.3. Wide Baseline Rotation

We next study wide baseline rotation, where we compare with [5] and their baselines.

Datasets. We use the two datasets from [5], which were derived from panoramic photos and follow the setup of [5]. The first dataset is InteriorNet [29], which consists of 10,050 panoramic views across 112 synthetic houses. Of these, 82 houses are allocated for training and the remaining 30 houses are used for testing. The dataset has 610k image pairs (350K overlapping), with a test set of 1K pairs. StreetLearn [39], consists of panoramic outdoor images in New York City that have been scrubbed to ensure privacy (full details in Supplemental). This dataset has 1.1M train pairs (460K with overlap), and 1K test pairs from a set of 143K panoramas. We additionally evaluate on the "InteriorNet-T" and "StreetLearn-T" datasets, which select from *different* panoramas for each image in a pair, resulting in translation in addition to rotation. This translation is not, however, estimated in this setting. To facilitate comparisons we use 10° as a threshold for rotation error following [5]. We use the setup of Cai *et al.* [5] using only overlapping images, and breaking down overlap into *large* overlap (less than 45° rotation) and *small* overlap (more than 45°). Cai *et al.* also conduct experiments on non-overlapping images; we consider this beyond our scope, which is focused on the case where correspondences may exist.

Baselines and Ablations. We compare to the state of the art (*Extreme Rotation [5]*), which computes a cross-correlation volume on paired image features, and uses a CNN to classify pose. We also report this method's baselines: Reg6D [73], which predicts a 6D representation from concatenated image features, similar to the *Associative3D Camera Branch* from §4.2 as well as correspondence baselines SIFT [32] and SuperPoint [8]. These baselines occasionally fail. Following [5], we indicate failure on more than 50% of the test set by marking the number in gray. We report the same ablations as in §4.2.

Quantitative Results. The proposed method is typically better than all baselines across both InteriorNet and StreetLearn, for both versions and overlap settings of the dataset (Table 4). Often, the proposed method reduces error compared to competing methods by more than half (e.g., InteriorNet Mean, Median with Large Overlap; StreetLearn-T Mean with Large Overlap). Small overlap is an especially difficult setting. For instance, on InteriorNet-T, all baselines have mean error above 10°. Yet, the proposed method is within 10° more than *96%* of the time. Interestingly, median error on InteriorNet-T is worse than Cai *et al.* [5]. We believe the large scale of InteriorNet is not the method's strongest setting, and the method provides strong inductive bias for small data settings (see §4.4). Nevertheless, we consider Cai *et al.* to be a strong baseline as it is specialized to large angle changes.

Performance breakdown of the model is displayed in Table 5. Adding a ViT is quite important, likely attributable to the large scale of data available. Beyond the ViT, improvements by each step are more mixed compared to the clear improvement of each step on Matterport. For instance, adding the dual softmax without coordinate embeddings is typically not helpful compared to using only Bilinear Attention. Yet, the full model performs best (best 5 times, second best 3 times; Bilinear Attention is best 4 times, second best twice). Moreover, the full model is rarely significantly worse than any intermediate ablation. This suggests,

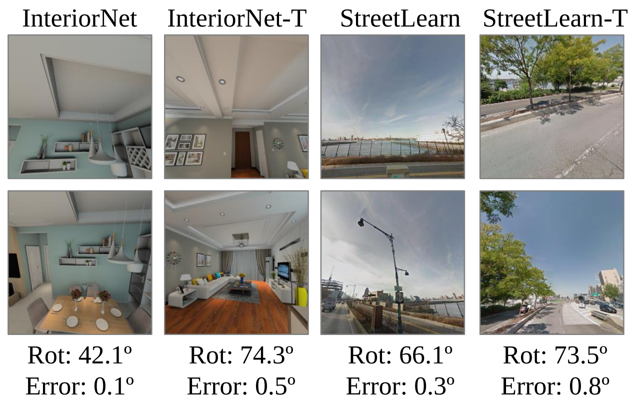

Figure 5. **Error vs. Rotation.** The proposed method produces high precision when faced with large view change.

as argued in §3, that all of the proposed components work together. We emphasize these settings have extraordinary numbers of views. §4.4 will show the substantially higher data efficiency of the essential module.

Analysis. Qualitative results validate quantitative findings in Fig. 5. While the evaluation datasets have huge rotations across indoor and outdoor settings, the proposed model is typically accurate, often even within 1% of true rotations.

4.4. Effectiveness on Smaller Datasets

One of the primary arguments for the use of the proposed network structure is that it provides a useful inductive bias by helping the network compute information that is known to constrain the set of feasible rotations and translations. In principle, since feedforward networks are universal approximators [21], networks ought to be able to learn to estimate relative pose with enough data. However, the right inductive biases ought to let them learn *faster*.

We now examine performance as a function of number of images. First, this helps empirically assess whether various networks structures provide useful inductive biases. Second, this is of practical concern since it tests data efficiency.

Datasets. We use InteriorNet-T and StreetLearn-T from §4.3, with significantly reduced 32K train image pairs. Collecting large-scale datasets such as these is challenging without a simulator or specialized company resources, so this smaller scale may be more realistic for e.g. user-collected posed images.

Ablations and Results. Our primary comparison is with the ViT baseline. Because it is a near alternative to our proposed Essential Matrix Module, we can measure the impact of our main contributions. Results are presented in Table 6, which is a reduced version of Table 5, with results also on the 32K image train set. Across datasets, the proposed method scales significantly better to a small train set. Even in cases the ViT slightly outperformed our proposed

Table 6. **Performance with limited data**. The proposed method scales better to small data than a typical learned model (e.g. ViT), indicating better inductive biases.

		InteriorNet-T					
		Full			32K		
		Avg	Med	$\% < 10°$	Avg	Med	$\% < 10°$
Large	ViT	0.61	0.49	**100.00**	5.78	3.23	92.84
	Full	**0.48**	**0.40**	**100.00**	**4.44**	**2.58**	**95.82**
Small	ViT	**1.44**	1.09	**100.00**	11.89	4.38	78.70
	Full	1.81	**0.94**	**100.00**	**8.22**	**4.27**	**89.20**

		StreetLearn-T					
		Full			32K		
		Avg	Med	$\% < 10°$	Avg	Med	$\% < 10°$
Large	ViT	**3.52**	2.56	**94.74**	11.51	7.69	56.58
	Full	4.08	**2.43**	90.13	**7.22**	**4.44**	**81.58**
Small	ViT	12.93	**3.16**	84.46	29.28	14.94	36.59
	Full	**9.19**	3.25	**87.70**	**13.29**	**5.55**	**71.72**

full model with full set, the inductive biases of the proposed method give it substantial improvement in the small setting.

5. Discussion

In this paper we presented a simple and interpretable end-to-end approach for pose estimation. Our key technical contribution is to implicitly represent correspondences from a ViT using an essential matrix module, from which an MLP can estimate pose. Theoretical results show this formulation can approximate the matrix $\mathbf{U}^\top\mathbf{U}$ that is analyzed in the Eight Point algorithm; empirical results show given this, the MLP can suitably estimate pose. While alternatives make additional assumptions about input or require optimization, this method requires only paired RGB images as input, and is competitive in a variety of settings and viewpoint changes while being computationally efficient.

Limitations and Social Impact. The model is generally robust across view change. However, other methods are better suited for the two extremes in view change. In the case of small view change, the transformer is limited in terms of precision by the number of patches. Alternative CNN-based methods such as [64] may more easily operate upon high resolution. The model is also not prepared to predict pose on images with no overlap or correspondences; classification-based work e.g. [5] is better suited for this. Using datasets such as Matterport collected in nice homes leads to models which will likely perform better in these homes and possibly not as well in less expensive homes. Using synthetic data such as InteriorNet may help combat this bias. Training and evaluating on StreetView images should be handled with special care, as these images can contain personal information. The original authors blurred faces in the dataset, and a random manual search of 500 images also revealed no personal identifying information.

Acknowledgments. Thanks to Linyi Jin, Ruojin Cai and Zach Teed for help replicating and building upon their works. Thanks to Mohamed El Banani, Karan Desai and Nilesh Kulkarni for their helpful suggestions. Thanks to Laura Fink and UM DCO for their tireless computing support. Toyota Research Institute ("TRI") provided funds to assist the authors with their research but this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity.

References

[1] Samir Agarwala, Linyi Jin, Chris Rockwell, and David F. Fouhey. Planeformers: From sparse view planes to 3d reconstruction. In *ECCV*, 2022. 6

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *ECCV*, 2006. 1, 2

[3] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *CVPR*, 2017. 2

[4] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *CVPR*, 2018. 2

[5] Ruojin Cai, Bharath Hariharan, Noah Snavely, and Hadar Averbuch-Elor. Extreme rotation estimation using dense correlation volumes. In *CVPR*, 2021. 1, 2, 7, 8

[6] 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 *3DV*, 2017. 2, 5

[7] Kefan Chen, Noah Snavely, and Ameesh Makadia. Wide-baseline relative camera pose estimation with directional learning. In *CVPR*, 2021. 1, 2

[8] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, 2018. 1, 2, 7

[9] 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. *ICLR*, 2021. 1, 2, 3

[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 *CVPR*, 2019. 1, 2

[11] Mohamed El Banani, Luya Gao, and Justin Johnson. Unsupervisedr&r: Unsupervised point cloud registration via differentiable rendering. In *CVPR*, 2021. 1, 2

[12] Sovann En, Alexis Lechervy, and Frédéric Jurie. Rpnet: An end-to-end network for relative camera pose estimation. In *ECCVW*, 2018. 1, 2, 6

[13] Olivier D Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *ECCV*, pages 563–578. Springer, 1992. 2

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

[15] Banglei Guan, Qifeng Yu, and Friedrich Fraundorfer. Minimal solutions for the rotational alignment of imu-camera systems using homography constraints. *Computer vision and image understanding*, 170:79–91, 2018. 1, 2

[16] Christopher Ham, Simon Lucey, and Surya Singh. Hand waving away scale. In *ECCV*, 2014. 1, 2

[17] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 1, 2

[18] Richard I Hartley. Estimation of relative camera positions for uncalibrated cameras. In *ECCV*. Springer, 1992. 2

[19] Richard I Hartley. In defense of the eight-point algorithm. *TPAMPI*, 19(6):580–593, 1997. 1, 2

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[21] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991. 3, 8

[22] Linyi Jin, Shengyi Qian, Andrew Owens, and David F Fouhey. Planar surface reconstruction from sparse views. In *ICCV*, 2021. 1, 2, 5, 6

[23] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM Computing Surveys (CSUR)*, 2021. 2

[24] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear attention networks. *NeurIPS*, 31, 2018. 1, 2, 3

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015. 5

[26] Laurent Kneip, Margarita Chli, and Roland Siegwart. Robust real-time visual odometry with a single camera and an imu. In *BMVC*, 2011. 1, 2

[27] Viktor Larsson, Magnus Oskarsson, Kalle Astrom, Alge Wallis, Zuzana Kukelova, and Tomas Pajdla. Beyond grobner bases: Basis selection for minimal solvers. In *CVPR*, 2018. 2

[28] Zakaria Laskar, Iaroslav Melekhov, Surya Kalia, and Juho Kannala. Camera relocalization by computing pairwise relative poses using convolutional neural network. In *ICCVW*, 2017. 2

[29] 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 *BVMC*, 2018. 2, 7

[30] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *ICCV*, 2021. 1

[31] H Christopher Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293(5828):133–135, 1981. 1, 2

[32] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*. 1, 2, 5, 7

[33] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *NeurIPS*, 32, 2019. 5

[34] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. In *SIGGRAPH*, 2020. 1

[35] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *CVPR*, 2021. 1

[36] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004. 2

[37] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. Relative camera pose estimation using convolutional neural networks. In *ACIVS*, 2017. 2

[38] 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*. Springer, 2020. 1

[39] Piotr Mirowski, Andras Banki-Horvath, Keith Anderson, Denis Teplyashin, Karl Moritz Hermann, Mateusz Malinowski, Matthew Koichi Grimes, Karen Simonyan, Koray Kavukcuoglu, Andrew Zisserman, et al. The streetlearn environment and dataset. *arXiv preprint arXiv:1903.01292*, 2019. 2, 7

[40] Dmytro Mishkin, Jiri Matas, Michal Perdoch, and Karel Lenc. Wxbs: Wide baseline stereo generalizations. *BMVC*, 2015. 2

[41] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *T-RO*. 1, 2

[42] David Nistér. An efficient solution to the five-point relative pose problem. *TPAMI*, 26(6):756–770, 2004. 1, 2

[43] Namuk Park and Songkuk Kim. How do vision transformers work? In *ICLR*, 2021. 2

[44] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32, 2019. 5

[45] Omid Poursaeed, Guandao Yang, Aditya Prakash, Qiuren Fang, Hanqing Jiang, Bharath Hariharan, and Serge Belongie. Deep fundamental matrix estimation without correspondences. In *ECCVW*, 2018. 2

[46] Phil Pritchett and Andrew Zisserman. Wide baseline stereo matching. In *ICCV*, 1998. 2

[47] Thomas Probst, Danda Pani Paudel, Ajad Chhatkuli, and Luc Van Gool. Unsupervised learning of consensus maximization for 3d vision problems. In *CVPR*, 2019. 2

[48] Shengyi Qian, Linyi Jin, and David F Fouhey. Associative3d: Volumetric reconstruction from sparse views. In *ECCV*, 2020. 1, 2, 6

[49] Noha Radwan, Abhinav Valada, and Wolfram Burgard. Vlocnet++: Deep multitask learning for semantic visual localization and odometry. *RA-L*, 2018. 2

[50] René Ranftl and Vladlen Koltun. Deep fundamental matrix estimation. In *ECCV*, 2018. 2

[51] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *TPAMI*, 2020. 6

[52] Carolina Raposo, Miguel Lourenço, Michel Antunes, and Joao Pedro Barreto. Plane-based odometry using an rgb-d camera. In *BMVC*. 6

[53] I. Rocco, R. Arandjelović, and J. Sivic. End-to-end weakly-supervised semantic alignment. In *CVPR*, 2018. 1, 2, 4

[54] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, 2011. 1, 2

[55] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 1, 2, 6

[56] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *ICCV*, 2019. 5

[57] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1, 2

[58] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, 2019. 5

[59] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *CVPR*, 2021. 1, 2, 4, 6

[60] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *NeurIPS*, 2021. 2

[61] Zachary Teed and Jia Deng. Tangent space backpropagation for 3d transformation groups. In *CVPR*, 2021. 5

[62] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. *NeurIPS*, 33, 2020. 1, 2, 4

[63] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017. 1, 2, 3

[64] Wenshan Wang, Yaoyu Hu, and Sebastian Scherer. Tartanvo: A generalizable learning-based vo. In *CoRL*, 2021. 2, 5, 8

[65] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *IROS*, 2020. 2

[66] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 5

[67] Fei Xue, Xin Wang, Zike Yan, Qiuyuan Wang, Junqiu Wang, and Hongbin Zha. Local supports global: Deep camera relocalization with sequence enhancement. In *ICCV*, 2019. 2

[68] Zhenpei Yang, Jeffrey Z Pan, Linjie Luo, Xiaowei Zhou, Kristen Grauman, and Qixing Huang. Extreme relative pose estimation for rgb-d scans via scene completion. In *CVPR*, 2019. 2

[69] Zhenpei Yang, Siming Yan, and Qixing Huang. Extreme relative pose network under hybrid representations. In *CVPR*, 2020. 1, 2

[70] Wang Yifan, Carl Doersch, Relja Arandjelović, João Carreira, and Andrew Zisserman. Input-level inductive biases for 3d reconstruction. *arXiv preprint arXiv:2112.03243*, 2021. 2

[71] Qunjie Zhou, Torsten Sattler, Marc Pollefeys, and Laura Leal-Taixe. To learn or not to learn: Visual localization from essential matrices. In *ICRA*, 2020. 2

[72] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *SIGGRAPH*, 2018. 1

[73] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *CVPR*, 2019. 7