Fast Relative Pose Estimation using Relative Depth

Jonathan Astermark[1], Yaqing Ding[1,2], Viktor Larsson[1], and Anders Heyden[1]
[1] Centre for Mathematical Sciences, Lund University
[2] Visual Recognition Group, Faculty of Electrical Engineering, Czech Technical University in Prague
{jonathan.astermark, yaqing.ding, viktor.larsson, anders.heyden}@math.lth.se

Abstract

In this paper, we revisit the problem of estimating the relative pose from a sparse set of point-correspondences. For each point-correspondence we also estimate the relative depth, i.e. the relative distance to the scene point in the two images. This yields an additional constraint, allowing us to use fewer matches in RANSAC to generate the pose candidates. In the paper we propose two novel minimal solvers: one for general motion and one for the case of known vertical direction. To obtain the relative depth estimates, we explore using scale estimates obtained from a keypoint detector as well as a neural network that directly predicts the relative depth for a pair of patches.

We show in experiments that while our estimates are more noisy compared to the purely point-based solvers, the smaller sample size leads to a significantly reduced runtime in settings with high outlier ratios.

1. Introduction

Relative pose estimation, i.e. estimating the relative translation and orientation of two images, is an important sub-problem in many applications in Structure-from-Motion (SfM) and Simultaneous Localization And Mapping (SLAM). The most common approach is to estimate the relative pose from a sparse set of point correspondences. To handle outlier matches, robust estimators such as Random Sample Consensus (RANSAC) [10] are used. These alternate between solving for the epipolar geometry for a minimal sample of point correspondences, and verifying the quality of solutions on the entire set of input correspondences. In the case of calibrated cameras, the minimal problem [19] has five degrees of freedom and requires five keypoint correspondences to solve. In RANSAC, the size of the minimal sample heavily impacts the required number of iterations. Smaller samples have less risk of being outlier-contaminated, and thus good models can be found quicker.

In this paper, we explore relative pose estimation using additional information about the *relative depth* of the matched keypoints, i.e. how much closer the observed point is in the first image compared to the second. This additional information can be well-approximated by the apparent relative size difference in the two images, see Figure 1. Closer objects appear larger and vice versa. When using scale-covariant features, such as SIFT [17], the scales from the detector can directly give us an estimate of the relative depth, assuming the focal lengths are known. The details are described in Section 3.

Figure 1. **Relative depth from scale.** The relative scale of regions around corresponding keypoints is inversely proportional to the relative depth. In this paper we leverage this fact for relative camera pose estimation.

Knowing the relative depth of the matched keypoints gives extra geometric constraints on the camera poses. This can be used to reduce the number of matches required in the minimal problem from five to three. In scenarios with high outlier ratios, this drastically reduces the number of iterations required, as this grows exponentially with sample size. For example, if 50 % of the correspondences are outliers, a solver using a sample size of five requires more than 4x the number of iterations compared to a solver using a sample size of three. With 90 % outliers, this ratio is instead 100x.

Since many of the state-of-the-art learned keypoint detectors do not provide point-wise scale estimates, we also propose a learning-based approach to estimate the relative depth from corresponding patches. This allows us to combine our approach with any modern keypoint detector.

In this paper we explore how to best integrate relative depth information into relative pose estimation pipelines.

We make the following contributions:

- We propose a novel minimal 3-point solver that is significantly faster compared to previous work.
- We additionally propose a novel minimal 2-point solver for the case of known vertical direction along with relative depth estimates.
- We experimentally show that these solvers can significantly speed up RANSAC estimators by requiring fewer iterations to find promising pose candidates.
- We propose a deep neural network for directly estimating the relative depths from pairs of image patches.

1.1. Related Work

The most common approach for robustly estimating relative pose from sparse point matches is using some variant of RANSAC [10] in combination with the minimal solver [19]. While there have been many RANSAC variants proposed, see e.g. [4, 6, 7, 14, 22], they all have the property that the number of required iterations increase with the size of the random sample used in the solver. Motivated by this, there have been several works that consider variants of the minimal problem (requiring five point correspondences [19]), where additional information is added to reduce the required number of point matches.

Fraundorfer et al. [11] proposed a minimal solver using only three points for the case of partially known rotation (e.g. known vertical direction). Later, Sweeney et al. [25] showed that this can be formulated and efficiently solved as a Quadratic Eigenvalue Problem (QEP).

Barath and Kukelova [3] derive an additional constraint on the relative pose by relating the SIFT scale and orientation to affine correspondences. They show that their solver, despite being less stable than the purely point-based five point solver [19], when used in GC-RANSAC [4] can give similar accuracy while being much faster. In this work we propose to only leverage (relative) scale information and derive geometric constraints from this. Our experiments show that using the relative scale information as a proxy for relative depth yields more accurate pose estimates.

Similarly, Barath and Sweeney [5] introduce a 4pt+D and 2pt+D solver, which leverages depths from a learned monocular depth network. In this case, the absolute depths are only given up to an unknown scale in each image. Parameterizing the two shared unknown scales, the authors derive minimal estimators that jointly estimate relative pose and the depth scales. Our work, in contrast, instead assumes that we directly have the *inter*-image relative depths given.

Most similar to our work, Liwicki and Zach [16] introduce a 3-point solver using relative depth for two correspondences. They parameterize the relative rotation matrix using Cayley parameterization and derive a solver using Gröbner basis techniques. In the paper, the authors experiment with using the relative SIFT scale to approximate the relative depth. To compensate for SIFT scale errors, the authors use a bisection algorithm, solving iteratively for multiple scales inside RANSAC. This leads to an overall slower algorithm that needs to evaluate point residuals multiple times for each minimal sample. Guan and Zhao [12] further extend this solver to the multi-camera setting solving the generalized relative pose problem.

We build on the work in [16] by introducing an alternative parameterization for a 3-point solver using relative depth, and show that this gives a significantly faster solver. We also show how our formulation can be easily adapted to integrate known vertical direction, further reducing the required number of matches, which allows us to introduce a 2-point solver for this case. While [16] and [12] only considers using SIFT scales, we propose to use a neural network to directly predict point-wise relative depths. This allows our approach to be combined with other keypoint detectors that do not provide scale estimates.

2. Relative Depth in Relative Pose Estimation

We first detail the geometry of relative pose based on relative depths. We then present a novel minimal estimator for the general case, from three correspondences of which two have relative depth information. Finally we extend our approach to include known vertical directions.

The geometric constraints induced by known relative depth in two-view relative pose estimation was first described in [16]. Given a calibrated camera pair with the relative pose $(\mathbf{R}, \boldsymbol{t}) \in SO(3) \times \mathbb{R}^3$, the projection of a 3D-point $\boldsymbol{X} \in \mathbb{R}^3$ into the two image points $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{P}^2$ is

$$\begin{cases} \lambda \boldsymbol{x} = \boldsymbol{X} \\ \lambda' \boldsymbol{x}' = \mathbf{R}\boldsymbol{X} + \boldsymbol{t} \end{cases} \quad \Rightarrow \quad \lambda' \boldsymbol{x}' = \lambda \mathbf{R}\boldsymbol{x} + \boldsymbol{t}, \tag{1}$$

where λ and λ' are unknowns which we will refer to as *depths*[1]. If the *relative depth* $\sigma = \lambda'/\lambda$ is known, we can rewrite (1) as

$$\lambda(\sigma \boldsymbol{x}' - \mathbf{R}\boldsymbol{x}) = \boldsymbol{t} \tag{2}$$

or equivalently, decomposed into magnitude and direction,

$$\begin{cases} \lambda \, ||\sigma \boldsymbol{x}' - \mathbf{R}\boldsymbol{x}|| = ||\boldsymbol{t}|| & (3) \\ \boldsymbol{t} \times (\sigma \boldsymbol{x}' - \mathbf{R}\boldsymbol{x}) = 0, & (4) \end{cases}$$

where $||\cdot||$ is the 2-norm. (3) gives no meaningful constraint when the absolute depths are unknown, since a change in translation magnitude $||\boldsymbol{t}||$ can always be counteracted by a corresponding change in depth λ. As described in [16], (4) enforces both the epipolar constraint and the law of sines for the correspondence $(\boldsymbol{x}, \boldsymbol{x}')$. Thus (4) gives us one extra

[1] Note that this definition of depth is *not* the distance from the camera to $\boldsymbol{X}$, but rather to the plane in which $\boldsymbol{X}$ lies, i.e. the last coordinates of $\boldsymbol{X}$. See the supplementary material for details.

constraint in addition to the epipolar constraint. To see the epipolar constraint, we multiply from the left with $(\boldsymbol{x}')^\mathsf{T}$ and notice that the first term vanishes, while from the second term we recover exactly the epipolar constraint

$$\underbrace{\sigma \boldsymbol{x}'^\mathsf{T}[\boldsymbol{t}]_\times \boldsymbol{x}'}_{\equiv 0} - \underbrace{\boldsymbol{x}'^\mathsf{T}[\boldsymbol{t}]_\times \mathbf{R}\boldsymbol{x}}_{\textit{epipolar constraint}} = 0, \tag{5}$$

where $[\boldsymbol{t}]_\times$ is the skew-symmetric matrix representation of the cross-product. To see the law of sines, we apply the 2-norm to (4), which gives

$$\sigma \left|\left|\boldsymbol{t} \times \boldsymbol{x}'\right|\right| = \left|\left|\boldsymbol{t} \times \mathbf{R}\boldsymbol{x}\right|\right|, \tag{6}$$

which is equivalent to the law of sines on the triangle in the epipolar plane formed by the translation vector $\boldsymbol{t}$ and the two projection rays $\boldsymbol{x}'$ and $\mathbf{R}\boldsymbol{x}$. We refer to the supplementary material for more details on this interpretation.

2.1. Solving for Relative Pose from Three Points

Solving for the relative pose $(\mathbf{R}, \boldsymbol{t})$ involves solving for five degrees of freedom: three for rotation and two for translation (since translation magnitude is arbitrary). When using pure point correspondences, i.e. (1), each correspondence $(\boldsymbol{x}_i, \boldsymbol{x}'_i)$, $i \in [1, n]$ yields three equations while introducing two new unknowns λ_i, λ'_i, resulting in a net gain of a single constraint. This allows the relative pose to be estimated minimally from five correspondences [19].

If the relative depths $\sigma_i = \lambda'_i/\lambda_i$ are known, each correspondence only introduces a single new unknown λ_i, see (2). Thus we have a net gain of two constraints per correspondence. This reduces the number of correspondences needed to solve a minimal problem to three, as the number of constraints for n correspondences is now $2n$. In fact, three correspondences with relative depth is overdetermined by one constraint. In [16] the authors proposed a solver for the minimal problem for 2+1 points, i.e. two correspondences that have relative depth and one without. The solver proposed in [16] parameterizes the rotation using Cayley parameterization and solves the resulting polynomial equation system using Gröbner basis techniques. In [16] the authors also consider a solver that use 1+3 points, i.e. four matches with one known relative depth. In their experiments this performs worse while requiring more matches.

In this section, we present an alternative formulation of a solver for 2+1 points, which instead parameterizes the problem via the depths. This simplifies the equations and directly allow us to decompose the problem into solving two quadratic equations that have closed-form solutions.

Given three correspondences, two of which have known relative depth σ, we get three equations from (2),

$$\sigma_1 \lambda_1 \boldsymbol{x}'_1 = \lambda_1 \mathbf{R}\boldsymbol{x}_1 + \boldsymbol{t}, \tag{7}$$
$$\sigma_2 \lambda_2 \boldsymbol{x}'_2 = \lambda_2 \mathbf{R}\boldsymbol{x}_2 + \boldsymbol{t}, \tag{8}$$
$$\lambda'_3 \boldsymbol{x}'_3 = \lambda_3 \mathbf{R}\boldsymbol{x}_3 + \boldsymbol{t}. \tag{9}$$

Since the global scale is unobservable, we can without loss of generality set $\lambda_1 = 1$. Then, forming the differences (7)-(8), we have

$$\sigma_1 \boldsymbol{x}'_1 - \sigma_2 \lambda_2 \boldsymbol{x}'_2 = \mathbf{R}(\boldsymbol{x}_1 - \lambda_2 \boldsymbol{x}_2). \tag{10}$$

Since the rotation matrix preserves lengths, $\|u\| = \|\mathbf{R}u\|$,

$$\|\sigma_1 \boldsymbol{x}'_1 - \sigma_2 \lambda_2 \boldsymbol{x}'_2\|^2 = \|\boldsymbol{x}_1 - \lambda_2 \boldsymbol{x}_2\|^2, \tag{11}$$

which gives us a quadratic equation in only λ_2 that can be solved in closed form. Next, similarly forming the differences (7)-(9) and (8)-(9), we have

$$\|\sigma_1 \boldsymbol{x}'_1 - \lambda'_3 \boldsymbol{x}'_3\|^2 = \|\boldsymbol{x}_1 - \lambda_3 \boldsymbol{x}_3\|^2, \tag{12}$$
$$\|\sigma_2 \lambda_2 \boldsymbol{x}'_2 - \lambda'_3 \boldsymbol{x}'_3\|^2 = \|\lambda_2 \boldsymbol{x}_2 - \lambda_3 \boldsymbol{x}_3\|^2. \tag{13}$$

For each solution of λ_2, these are quadratic equations in λ_3 and λ'_3. The key observation here is now that the quadratic terms in λ_3 and λ'_3 respectively, are the same in both equations. Thus forming their difference yields a linear equation

$$a\lambda_3 + b\lambda'_3 + c = 0, \tag{14}$$

where $a, b \in \mathbb{R}$ are constants that depend on λ_2. Inserting $\lambda_3 = -(b\lambda'_3 + c)/a$ into (12) we get a quadratic equation in λ'_3 that can be solved in closed form.

In summary, from (11) we get two solutions for λ_2, and from each of these solutions we get two solutions for (λ_3, λ'_3), for a total of four candidate relative poses. Once the depths are recovered, we can easily solve for the rotation using (10) and the corresponding equation from $\boldsymbol{x}_1$ and $\boldsymbol{x}_3$. Following [21], the rotation matrix can be computed as

$$\begin{aligned} &\mathbf{R} = \mathbf{Z}\mathbf{Y}^{-1}, \\ &\mathbf{Z} = [\mathbf{z}_1,\ \mathbf{z}_2,\ \mathbf{z}_1 \times \mathbf{z}_2], \\ &\mathbf{Y} = [\mathbf{y}_1,\ \mathbf{y}_2,\ \mathbf{y}_1 \times \mathbf{y}_2], \\ &\mathbf{z}_1 = \sigma_1 \boldsymbol{x}'_1 - \sigma_2 \lambda_2 \boldsymbol{x}'_2, \quad \mathbf{z}_2 = \sigma_1 \boldsymbol{x}'_1 - \lambda'_3 \boldsymbol{x}'_3, \\ &\mathbf{y}_1 = \boldsymbol{x}_1 - \lambda_2 \boldsymbol{x}_2, \quad \mathbf{y}_2 = \boldsymbol{x}_1 - \lambda_3 \boldsymbol{x}_3. \end{aligned} \tag{15}$$

Once the rotation is known we can insert it into any of the equations (7)-(9) to get the translation vector $\boldsymbol{t}$. Note that if the three points are co-linear, then $\boldsymbol{y}_1 \parallel \boldsymbol{y}_2$ and the matrix $\mathbf{Y}$ becomes singular. In this case, the rotation matrix $\mathbf{R}$ can not be recovered. In practice, this degeneracy seldom occurs and can be ignored when using a robust estimator.

Permutations of Input Correspondences. Given three points with relative scale information, we can select three

possible minimal 2+1 configurations (each using two of the relative scales). As the relative scale is significantly less accurate compared to the point coordinates, we can solve each of the three permutations. In Section 4.1 we will show that this significantly improves the accuracy.

2.2. Known Vertical Direction and Relative Depth

In many scenarios, the gravity (or vertical) direction is known, for example from an Inertial Measurement Unit (IMU) or vanishing point estimation. Knowing the vertical direction in both coordinate systems reduces the degrees of freedom in the rotation from three to one, as we have the additional constraint $\mathbf{R}\boldsymbol{g} = \boldsymbol{g}'$, where $\boldsymbol{g}, \boldsymbol{g}' \in S^2$ are the vertical directions in the two local coordinate systems. In this case it becomes possible to solve for the relative pose from two point correspondences, one of which has relative depth information.

For the two correspondences, we have from (1) that

$$\lambda'_1 \boldsymbol{x}'_1 - \lambda'_2 \boldsymbol{x}'_2 = \mathbf{R}(\lambda_1 \boldsymbol{x}_1 - \lambda_2 \boldsymbol{x}_2). \tag{16}$$

Taking the scalar product with $\boldsymbol{g}'$ we get

$$\begin{aligned}(\boldsymbol{g}')^\mathsf{T}(\lambda'_1 \boldsymbol{x}'_1 - \lambda'_2 \boldsymbol{x}'_2) &= (\boldsymbol{g}')^\mathsf{T}\mathbf{R}(\lambda_1 \boldsymbol{x}_1 - \lambda_2 \boldsymbol{x}_2) \\ &= \boldsymbol{g}^\mathsf{T}(\lambda_1 \boldsymbol{x}_1 - \lambda_2 \boldsymbol{x}_2),\end{aligned} \tag{17}$$

since $(\boldsymbol{g}')^\mathsf{T}\mathbf{R} = (\mathbf{R}^\mathsf{T}\boldsymbol{g}')^\mathsf{T} = \boldsymbol{g}^\mathsf{T}$. Thus, having a known vertical direction directly gives a linear constraint on the depths. As before, we can fix the scale with the first depth as $\lambda_1 = 1$, and use the relative depth to get $\lambda'_1 = \sigma_1$. Solving for λ'_2 in (17) gives

$$\lambda'_2(\lambda_2) = \lambda_2 \frac{\boldsymbol{g}^\mathsf{T}\boldsymbol{x}_2}{(\boldsymbol{g}')^\mathsf{T}\boldsymbol{x}'_2} + \frac{\sigma_1(\boldsymbol{g}')^\mathsf{T}\boldsymbol{x}'_1 - \boldsymbol{g}^\mathsf{T}\boldsymbol{x}_1}{(\boldsymbol{g}')^\mathsf{T}\boldsymbol{x}'_2}, \tag{18}$$

which is a linear function of λ_2. Taking the norm on both sides in (16) yields

$$\|\sigma_1 \boldsymbol{x}'_1 - \lambda'_2(\lambda_2)\boldsymbol{x}'_2\|^2 = \|\boldsymbol{x}_1 - \lambda_2 \boldsymbol{x}_2\|^2, \tag{19}$$

which is a quadratic equation in λ_2. This can be solved in closed form and gives us at most two real solutions for λ_2.

Once the depths are recovered we can solve for the rotation as in (15), but here we can use the gravity directions instead of $\boldsymbol{y}_2$ and $\boldsymbol{z}_2$ to get

$$\begin{aligned}\mathbf{R} &= \mathbf{Z}\mathbf{Y}^{-1}, \\ \mathbf{Z} &= [\mathbf{z},\ \boldsymbol{g}',\ \mathbf{z} \times \boldsymbol{g}'], \\ \mathbf{Y} &= [\mathbf{y},\ \boldsymbol{g},\ \mathbf{y} \times \boldsymbol{g}], \\ \mathbf{z} &= \sigma_1 \boldsymbol{x}'_1 - \lambda'_2 \boldsymbol{x}'_2, \\ \mathbf{y} &= \boldsymbol{x}_1 - \lambda_2 \boldsymbol{x}_2.\end{aligned} \tag{20}$$

We notice this solver has a degenerate case similar to the 3-point case; if $\boldsymbol{g} \parallel \boldsymbol{y}$, the matrix $\mathbf{Y}$ becomes singular and the rotation can not be recovered. This is equivalent to the two 3D-points being sampled from a vertical line parallel to $\boldsymbol{g}$ in 3D-space, in which case all rotations around $\boldsymbol{g}$ are possible solutions, see Figure 2.

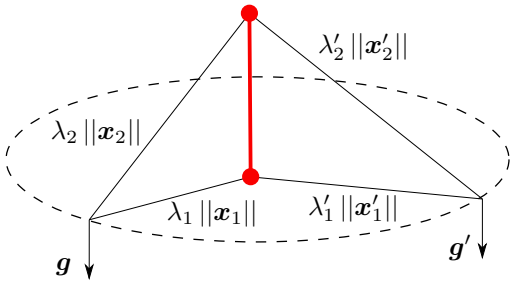

Figure 2. **Degenerate case for the 2-point solver.** When the two points are on a line parallel to the known vertical direction, all positions on the circle are valid poses.

3. Obtaining Relative Depth Estimates

3.1. Relative Depth from Keypoint Detection Scale

Scale-covariant detectors such as SIFT detect features at different scale levels. This information can be used to deduct an approximate relative depth. For a point correspondence with feature scales $s, s' \in \mathbb{R}^+$, we can define the *relative scale* γ as

$$\gamma = \frac{s}{s'}. \tag{21}$$

With known focal lengths f and f', we can relate relative scale to *relative depth* σ as

$$\sigma = \frac{s/f}{s'/f'} = \frac{f'}{f}\gamma. \tag{22}$$

In [16] the relative depth was directly approximated from inverse relative scale. In our case, we show that they are also related with the focal lengths. See the supplementary material for a more detailed discussion of why (22) holds.

3.2. Learning Improved Relative Depth

As noted in [16], the relative scale estimation from SIFT introduces errors that depend on the discretization in the algorithm. In [16], this was solved by introducing a bisection search algorithm and running the evaluation for multiple scales. However, this greatly increased the runtime of the algorithm, making the total runtime significantly worse than for the traditional 5pt-solver. Using the keypoint detection scale also restricts which keypoint detectors can be used. Many modern detectors (e.g. SuperPoint [9]) do not estimate an explicit keypoint scale.

To improve the stability of our pose estimates, and allow for using our solvers with any keypoint detector, we propose *RelScaleNet*, a neural network that predicts the relative depth directly from corresponding patches. While the

relative scale from SIFT is an approximation of the relative depth relying on the assumption of fronto-parallel patches, the proposed network has the opportunity to better handle view-point dependent effects as it is directly supervised with ground truth relative depth.

We base our model on HardNet [18] (which in turn is based on L2-net [26]), but double the number of channels and input size. We also remove the second convolutional layer and replace the final convolution with two fully connected layers with 256 and 1 output values, respectively, to directly estimate γ. We supervise with MSE loss on the ground truth relative scale γ^*. We train without batch normalization and dropout since we are directly regressing a value, and replace stride-2 convolutions with maxpool.

The ground-truth relative scale is calculated from ground-truth relative depth $\sigma^* = \lambda'/\lambda$ as

$$\gamma^* = \frac{f}{f'}\sigma^* = \frac{f}{f'}\frac{\lambda'}{\lambda} \tag{23}$$

using known focal lengths. The network is supervised by directly measuring the squared discrepancy between the prediction $\hat{\gamma}$ and ground-truth, i.e. minimizing

$$\mathcal{L}_2 = |\hat{\gamma} - \gamma^*|^2. \tag{24}$$

We also explored other losses such as $\mathcal{L}_1$ and measuring the 3D-consistency of the predicted scale (cf. (2)) but they performed similarly or worse in our experiments.

4. Experiments

4.1. Evaluation on Synthetic Data

We first evaluate the proposed minimal solvers on synthetic data. We generate random synthetic instances with two 90° field-of-view cameras viewing a scene with five points. For each scene we also generate a synthetic gravity direction used for the solvers in Section 2.2.

Numerical Stability and Runtime. First we compare the proposed solvers (Section 2.1 and Section 2.2) with the purely point-based solvers from Nistér et al. [19] (5-point) and Sweeney et al. [25] (3-point with gravity). We also compare with the relative-scale solver from Liwicki et al. [16] (3-point with relative scale) which uses Cayley-parameterization for the rotation matrix. Table 1 shows the average runtime over 10 000 synthetic instances. The proposed solvers have significantly lower runtime. Even running all permutations (Section 2.1) of the input is faster than solving a single permutation using the solver from [16]. In Figure 3 we show the distribution of pose errors (maximum of rotation and translation error) for noise-free data. All solvers exhibit good numerical stability.

Noise Sensitivity. Next we evaluate the impact of noise in the relative-depth estimates. For each synthetic instance we add zero-mean Gaussian noise to the keypoints with a standard deviation corresponding to one pixel in an image of size 2000×2000. We then vary the noise in the relative depth estimate and evaluate the pose accuracy. Figure 4 shows the success rate (pose error less than 5°) against the noise level in the relative scale. We can see that running all permutations greatly reduce the impact of noise in the relative depth.

4.2. Training of RelScaleNet

Dataset. We train RelScaleNet for relative scale estimation on the Image Matching Challenge PhotoTourism (IMC-PT) dataset [13], which consists of outdoor images of tourist attractions, captured from a wide variety of viewpoints and lighting conditions. We use the 2021 train/val split [2] (excluding the scenes removed for the 2022 challenge [1]), giving us ten training scenes and three validation scenes. Since the provided keypoints did not include SIFT scale information, new keypoints were detected using the SIFT algorithm implemented in COLMAP [24] and matched using exhaustive matching. The provided ground-truth poses and camera calibrations were used to triangulate 3D-points corresponding to the new keypoints. Points triangulated to behind the cameras were filtered away. For each scene in the training set, a maximum of 20 000 camera pairs were uniformly sampled from all camera pairs with at least 100 common inlier keypoints and used for training. Outliers were then further filtered by checking that Sampson error with ground-truth pose was less than one pixel. Sampled camera pairs with less than 100 common inliers remaining after this filtering were discarded. From the remaining matches, ten correspondences were uniformly sampled for each image pair, resulting in a total of 1.8 M training samples. Patches of size 64×64 pixels were extracted centered around each keypoint. When training the network, pairs of patches were concatenated into tensors with shape $64 \times 64 \times 6$. For hyperparameter tuning, the network was evaluated on the validation split of IMC-PT. Validation data was sampled the same way as training data, except all inlier matches were included.

Training Details. Training was done with stochastic gradient descent (SGD) with a starting learning rate of 10^{-4}, momentum of 0.9 and weight decay of 10^{-4}. The batch size was 1024 and training was done for 100 epochs with learning rate decimated every 20 epochs.

Since relative depth estimation should be equivariant to permutation, we augmented the training data with a 50 % chance of switching the order of patches, and replacing the label with its reciprocal. We further augment with 50 % chance of horizontally flipping each training sample, as well as augmenting brightness, contrast, saturation and hue through pytorch's "ColorJitter" [20] function with with the respective parameters set to 0.3.

	p	σ	Solver		Solver runtime Absolute	Solver runtime Relative	Solutions
General motion	1	2	Depth	*This paper*	**0.3** μs	**1.0x**	3.4
	0	3	Depth+Perm.	*This paper*	0.8 μs	2.7x	10.3
	1	2	Cayley param.	Liwicki and Zach [16]	2.2 μs	7.3x	4.0
	5	0	Essential mat.	Nistér [19]	3.2 μs	10.7x	4.4
Known vertical	1	1	Depth	*This paper*	**0.13** μs	**1.0x**	1.7
	0	2	Depth+Perm.	*This paper*	0.25 μs	1.9x	3.4
	3	0	QEP	Sweeney et al. [25]	0.46 μs	3.5x	2.1

Table 1. **Runtime statistics for the minimal solvers.** Table shows the median runtime over 10,000 synthetic instances which have an exact solution. For each solver the number of point correspondences with scale (σ) and without scale (p) is shown. The proposed 3-point solver for general motion, using depth-based parameterization, is significantly faster compared to the Cayley-based solver from [16]. The proposed 2-point solver for known vertical direction is significantly faster compared to the QEP-based solver from [25]. All experiments were run on a desktop computer with an Intel i7-12700KF CPU.

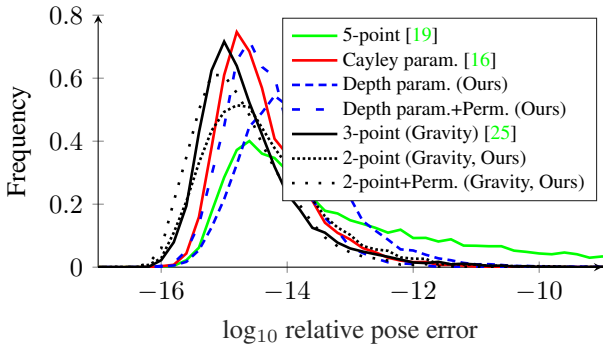

Figure 3. **Solver stability.** The graph shows the distribution of the $\log_{10}$ errors in the estimated pose for noise-free data.

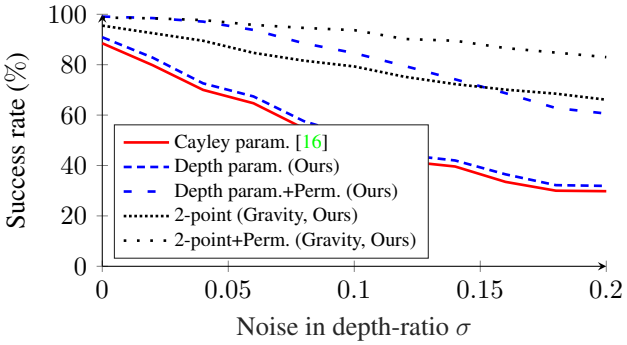

Figure 4. **Solver noise sensitivity.** The success rate (error less than 5°) for varying noise in the relative depth.

4.3. Evaluation of Relative Depth

Next we evaluate the accuracy of relative depth estimates, both those obtained via keypoint detection scales (from SIFT) and from the proposed RelScaleNet. We evaluate on held-out data from IMC-PT and ScanNet [8]. The latter contains indoor scenes and is quite different from the training data, and we use this to illustrate our model's ability to generalize. For IMC-PT we use the nine 100-image test sequences from the 2021 Image Matching Challenges [2]. We detect and match SIFT keypoints for all image pairs in the same way as for the training data. For ScanNet we follow the evaluation protocol from Sarlin et al. [23] to extract 1500 image pairs from the test set.

To evaluate the relative depth prediction, we collect all inlier correspondences with respect to the ground-truth relative pose for all test pairs in both datasets, again using the Sampson error. Using the ground-truth poses we compute the ground-truth relative depths via triangulation. Table 2 shows the errors in the estimated relative depth, i.e.

$$E_\sigma = \left| \hat{\gamma} \frac{f'}{f} - \sigma^* \right| . \tag{25}$$

We also include the errors obtained with scales from both SIFT [17] and the network from Lee et al. [15] for predicting keypoint scale. Firstly, we see that the predicted relative depth is more accurate compared to both the scale from SIFT and Self-Sca-Ori [15]. Secondly, despite being trained on only outdoor images from the IMC-PT training set, the network generalizes to the indoor images from ScanNet [8]. Finally, we evaluate the relative depth prediction for matches obtained with SuperPoint+SuperGlue [9, 23]. While the accuracy of the estimate degrades (as we only trained on SIFT keypoints), almost half of correspondences are within 10 % scale error. In Figure 5 we show a qualitative comparison of the estimated depths as heatmaps. In the supplementary material, we present additional qualitative results, including per-scene results.

4.4. Evaluation with RANSAC

Finally we evaluate the solvers in RANSAC for robust estimation of relative pose. We integrate them into an LO-RANSAC [14] framework and GC-RANSAC [4]. The rel-

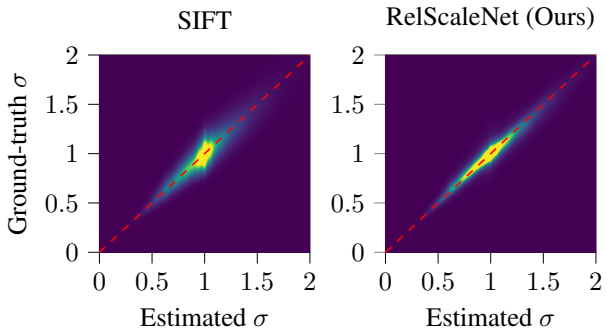

Figure 5. **Qualitative evaluation of relative depth estimates.** The heatmaps show qualitative differences in estimated relative depths (σ) compared to the ground-truth, using SIFT or RelScaleNet on 14 M correspondences from IMC-PT [13]. The dashed red line shows where estimation is equal to ground truth, i.e. perfect estimation. The bin values have been truncated to $\leq 10\,000$ for better visualization.

		Method	Med.↓	Accuracy @0.05↑	@0.1↑	@0.2↑
IMC-PT	SIFT	SIFT	0.063	0.41	0.69	0.92
		Self-Sca-Ori [15]	0.274	0.12	0.22	0.40
		RelScaleNet [**OURS**]	**0.033**	**0.65**	**0.86**	**0.97**
ScanNet	SIFT	SIFT	0.071	0.38	0.64	0.88
		Self-Sca-Ori [15]	0.120	0.27	0.45	0.66
		RelScaleNet [**OURS**]	**0.044**	**0.55**	**0.80**	**0.94**
	SP+SG	Self-Sca-Ori [15]	0.201	0.19	0.32	0.50
		RelScaleNet [**OURS**]	**0.114**	**0.27**	**0.46**	**0.68**

Table 2. **Evaluation of relative depth estimates.** The table shows the median error and accuracy at different thresholds in the estimated relative depth for the IMC-PT [13] and ScanNet [8] datasets. The proposed RelScaleNet provides the most accurate relative depth estimates. For comparison, we also include the predicted relative depth for SuperPoint+SuperGlue keypoints on ScanNet. Note that RelScaleNet was only trained on SIFT correspondences.

ative depth estimates are only used for generating candidate poses in RANSAC, and the model scoring/refinement is based purely on the point-correspondences. We explored using the relative depth estimates also for scoring and refinement, but were not able to get any improvement, likely due to the much higher noise levels. For the experiments we again consider the test set from IMC-PT and ScanNet as in Section 4.3.

PhotoTourism. Table 3 shows the aggregate statistics for the nine scenes in IMC-PT (individual results can be found in the supplementary material). We compute the Area-Under-Curve (AUC) of the pose error (max of rotation and translation error) up to some threshold as a percentage of the full square.[2] The proposed methods have slightly lower AUC compared to the 5-point solver, both with SIFT and predicted relative depth, with comparable runtimes. However, the dataset has quite high inlier ratio overall (median 78 %) so the benefit of the smaller minimal sample size is diminished. To highlight this, we also show the statistics for the top-5 % hardest image pairs (lowest inlier ratio w.r.t. ground truth pose), which have a median inlier ratio of 17 %. In this case, the runtime discrepancy between the 5-point solver and the 3-point becomes much more significant. We also compare with the SIFT-based 3-point solver from Barath and Kukelova [3]. The experiments show that the proposed solver using the relative depth constraints provide more accurate camera poses. Comparing LO-RANSAC to GC-RANSAC, we can see that since GC-RANSAC is more robust to poor candidate models, the gap between the two solvers is smaller.

ScanNet. Table 4 shows the results on ScanNet. This is a much more challenging dataset compared to IMC-PT, with a median inlier ratio of 16 % for SIFT, and 17 % for SP+SG, and we can again see that our methods achieve slightly worse accuracy compared to the 5-point solver, while having significantly lower runtimes.

To evaluate the solvers for known vertical direction from Section 2.2, we generate a synthetic gravity direction in each image using the ground-truth rotation. Table 4 also shows a comparison with the 3-point solver from Sweeney et al. [25]. As the difference in sample-size (three vs. two) is smaller than for the general motion solvers, the difference in runtime is also smaller.

Finally, we also evaluate on ScanNet with SuperPoint+SuperGlue [9, 23] keypoints and matches. In this case, we get improvements in both accuracy and runtime, compared to the 5-point solver. We visualize the cumulative distributions for the RANSAC experiments in Figure 6.

5. Conclusion

In this paper we have proposed new solvers for estimating the relative pose from point correspondences that have point-wise relative depth information. The relative depth can either be approximated as the relative scale from keypoint detectors, or as we show predicted from pair-wise patches with a neural network. Our experiments have shown that integrating relative depth information can reduce the runtime in RANSAC, by requiring fewer iterations to generate good candidate poses. While the relative poses we obtain are generally noisier, there are cases (SP+SG on ScanNet) where we see an improvement in both accuracy and runtime compared to the traditional 5-point solver.

Acknowledgements. The project received funding from the strategic research project ELLIIT.

[2] In the supplementary material we also report mAA@10°.

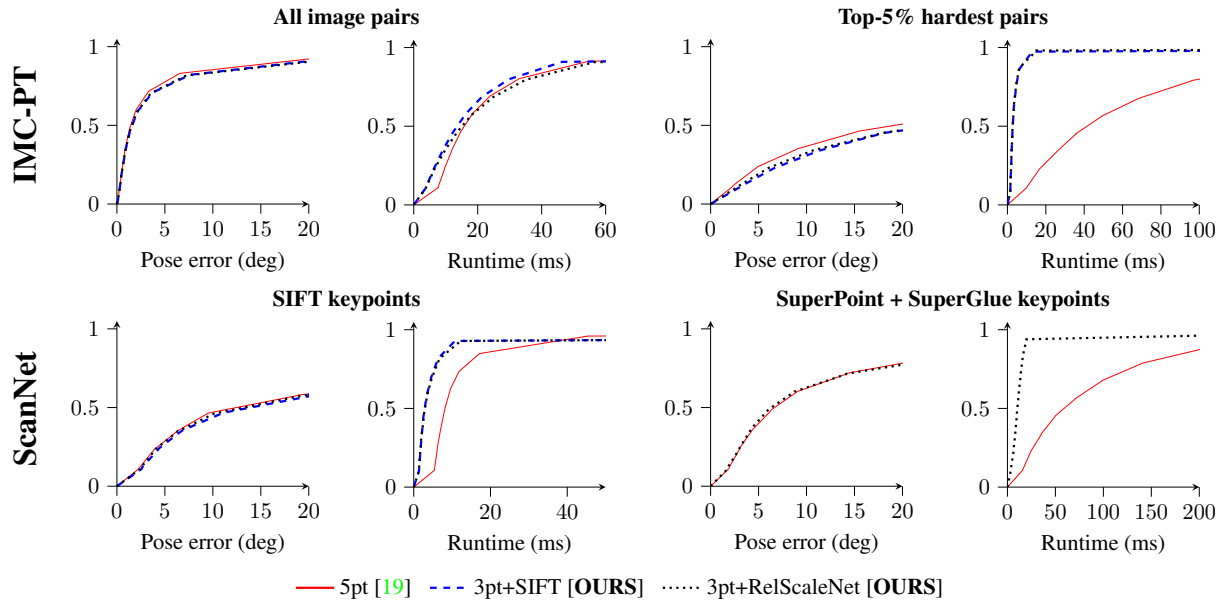

Figure 6. **Cumulative pose error and runtimes on IMC-PT [13] and ScanNet [8].** For IMC-PT we report results for all image pairs, and for only the top-5% hardest image pairs. For ScanNet, we report results using SIFT-keypoints and SP+SG [9, 23] keypoints.

			All image pairs				Top-5% hardest pairs			
RANSAC	Method		AUC@5°	AUC@10°	AUC@20°	RT(ms)	AUC@5°	AUC@10°	AUC@20°	RT(ms)
LO-RSC	5 pt.	(Nistér [19])	**56.89**	**70.43**	**80.39**	15.7	**12.13**	**21.79**	**33.55**	42.2
	3 pt.	+ SIFT (Barath & Kukelova [3])	30.77	39.02	46.98	**7.0**	1.23	3.03	6.25	21.3
	3 pt.	+ SIFT [**OURS**]	54.30	67.80	78.16	13.4	8.72	16.76	28.61	**2.8**
	3 pt.	+ RelScaleNet [**OURS**]	54.63	68.06	78.16	15.0	9.47	18.04	29.69	**2.8**
GC-RSC	5 pt.	(Nistér [19])	**56.22**	**69.87**	**79.99**	25.4	**9.76**	**18.82**	**31.12**	16.5
	3 pt.	+ SIFT (Barath & Kukelova [3])	50.55	63.74	74.10	**11.1**	2.16	5.59	11.37	6.0
	3 pt.	+ SIFT [**OURS**]	52.73	66.62	77.38	16.2	5.24	11.71	21.83	**4.7**
	3 pt.	+ RelScaleNet [**OURS**]	53.11	67.01	77.72	16.8	5.65	12.69	23.07	**4.7**

Table 3. **Relative pose estimation on IMC-PT** [13]. The **best** and second best method in each category is highlighted. We compare solvers using both LO-RANSAC [14] (LO-RSC) and GC-RANSAC [4] (GC-RSC) as the robust estimator. We also show the results restricted to the top-5% hardest image pairs (defined by outlier ratio w.r.t. the ground truth pose). For high outlier instances, the average runtime of the 5-point method is significantly higher compared to our 3-point method.

KP.	Method		AUC@5°	AUC@10°	AUC@20°	Runtime (ms)
SIFT [17]	5 pt.	(Nistér [19])	**11.06**	**21.99**	**33.32**	8.2
	3 pt.	+ SIFT (Barath & Kukelova [3])	4.94	10.33	17.16	3.7
	3 pt.	+ SIFT [**OURS**]	9.90	20.59	31.96	**2.9**
	3 pt.	+ RelScaleNet [**OURS**]	10.43	21.21	32.43	**2.9**
	3 pt.	+ Gravity (Sweeney et al. [25])	**13.31**	**26.50**	**40.50**	1.8
	2 pt.	+ Gravity + SIFT [**OURS**]	12.53	25.27	39.17	**1.4**
	2 pt.	+ Gravity + RelScaleNet [**OURS**]	12.65	25.73	39.37	**1.4**
SP+SG [9, 23]	5 pt.	(Nistér [19])	17.55	34.21	51.50	59.4
	3 pt.	+ RelScaleNet [**OURS**]	**18.39**	**35.46**	**52.24**	**10.4**
	3 pt.	+ Gravity (Sweeney et al. [25])	**20.86**	**39.00**	**57.07**	7.6
	2 pt.	+ Gravity + RelScaleNet [**OURS**]	20.05	38.12	56.24	**6.5**

Table 4. **Relative pose estimation on ScanNet** [8]. The **best** and second best method in each category is highlighted. For the experiments with known vertical we generate the gravity direction synthetically using the ground truth rotation. Note that the SIFT-based solvers can only be run when using SIFT keypoints.

References

[1] IMC-PT 2020 dataset | Kwang Moo Yi @ UBC. https://www.cs.ubc.ca/~kmyi/imw2020/data.html, 2023. [Online; accessed 6. Jul. 2023]. 5

[2] Data - 2021 IMW Challenge. https://www.cs.ubc.ca/research/image-matching-challenge/2021/data, 2023. [Online; accessed 3. Aug. 2023]. 5, 6

[3] Daniel Barath and Zuzana Kukelova. Relative pose from sift features. In *European Conference on Computer Vision (ECCV)*, pages 454–469, Cham, 2022. Springer Nature Switzerland. 2, 7, 8

[4] Daniel Barath and Jiri Matas. Graph-cut ransac. In *Computer Vision and Pattern Recognition (CVPR)*, pages 6733–6741, 2018. 2, 6, 8

[5] Daniel Barath and Chris Sweeney. Relative pose solvers using monocular depth. In *International Conference on Pattern Recognition (ICPR)*, pages 4037–4043, 2022. 2

[6] Daniel Barath, Jana Noskova, and Jiri Matas. Marginalizing sample consensus. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 44(11):8420–8432, 2021. 2

[7] Ondrej Chum, Tomas Werner, and Jiri Matas. Two-view geometry estimation unaffected by a dominant plane. In *Computer Vision and Pattern Recognition (CVPR)*, 2005. 2

[8] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 6, 7, 8

[9] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018. 4, 6, 7, 8

[10] 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*, 24(6):381–395, 1981. 1, 2

[11] Friedrich Fraundorfer, Petri Tanskanen, and Marc Pollefeys. A minimal case solution to the calibrated relative pose problem for the case of two known orientation angles. In *European Conference on Computer Vision (ECCV)*, 2010. 2

[12] Banglei Guan and Ji Zhao. Relative pose estimation for multi-camera systems from point correspondences with scale ratio. In *ACM International Conference on Multimedia*, 2022. 2

[13] Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image Matching Across Wide Baselines: From Paper to Practice. *International Journal of Computer Vision (IJCV)*, 129(2): 517–547, 2021. 5, 7, 8, 2, 3, 4, 6

[14] Karel Lebeda, Jirı Matas, and Ondrej Chum. Fixing the locally optimized ransac–full experimental evaluation. In *British Machine Vision Conference (BMVC)*, 2012. 2, 6, 8

[15] Jongmin Lee, Yoonwoo Jeong, and Minsu Cho. Self-supervised learning of image scale and orientation. In *British Machine Vision Conference (BMVC)*, 2021. 6, 7

[16] Stephan Liwicki and Christopher Zach. Scale exploiting minimal solvers for relative pose with calibrated cameras. In *British Machine Vision Conference (BMVC)*, 2017. 2, 3, 4, 5, 6

[17] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60:91–110, 2004. 1, 6, 8

[18] Anastasiya Mishchuk, Dmytro Mishkin, Filip Radenović, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *Neural Information Processing Systems (NeurIPS)*, 2017. 5

[19] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 26(6):756–770, 2004. 1, 2, 3, 5, 6, 8, 4

[20] 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. *Neural Information Processing Systems (NeurIPS)*, 2019. 5

[21] Mikael Persson and Klas Nordberg. Lambda twist: An accurate fast robust perspective three point (p3p) solver. In *European Conference on Computer Vision (ECCV)*, 2018. 3

[22] Rahul Raguram, Ondrej Chum, Marc Pollefeys, Jiri Matas, and Jan-Michael Frahm. Usac: A universal framework for random sample consensus. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 35(8):2022–2038, 2012. 2

[23] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 6, 7, 8

[24] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 5

[25] Chris Sweeney, John Flynn, and Matthew Turk. Solving for relative pose with a partially known rotation is a quadratic eigenvalue problem. In *International Conference on 3D Vision (3DV)*, 2014. 2, 5, 6, 7, 8

[26] Yurun Tian, Bin Fan, and Fuchao Wu. L2-net: Deep learning of discriminative patch descriptor in euclidean space. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 5

Fast Relative Pose Estimation using Relative Depth

Supplementary Material

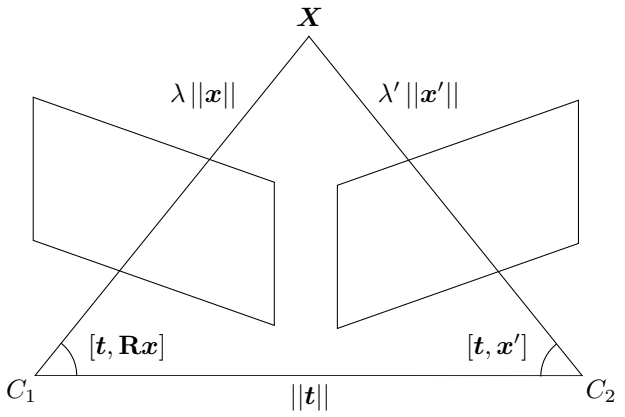

Figure 7. **Law of sines in the epipolar plane.** The epipolar geometry forms a triangle in the epipolar plane, with corners in $\boldsymbol{X}$, C_1, and C_2. The law of sines on the triangle gives (27).

6. Law of Sines Constraint

Here we show that (6) in Section 2 is equivalent to the law of sines. For convenience, we repeat (6) here:

$$\sigma\,||\boldsymbol{t} \times \boldsymbol{x}'|| = ||\boldsymbol{t} \times \mathbf{R}\boldsymbol{x}||\,.$$

Dividing both sides by $||\boldsymbol{t} \times \boldsymbol{x}'||$ and substituting $\sigma = \lambda'/\lambda$ gives

$$\frac{\lambda'}{\lambda} = \frac{||\boldsymbol{t} \times R\boldsymbol{x}||}{||\boldsymbol{t} \times \boldsymbol{x}'||} = \frac{||\boldsymbol{t}||\,||\boldsymbol{x}||\,|\sin[\boldsymbol{t}, \mathbf{R}\boldsymbol{x}]|}{||\boldsymbol{t}||\,||\boldsymbol{x}'||\,|\sin[\boldsymbol{t}, \boldsymbol{x}']|}, \tag{26}$$

where $[\cdot, \cdot]$ denotes the angle between two vectors. This angle can be assumed to be between 0° and 180°, so the absolute value can be removed. Then we can rewrite (26) as

$$\frac{\sin[\boldsymbol{t}, \boldsymbol{x}']}{\lambda\,||\boldsymbol{x}||} = \frac{\sin[\boldsymbol{t}, R\boldsymbol{x}]}{\lambda'\,||\boldsymbol{x}'||}, \tag{27}$$

which is exactly the law of sines on the triangle in Figure 7.

7. Recovering Relative Depth from Scales

In this section, we explain how to recover relative depth from image scales. Consider a feature at point $\boldsymbol{X}$ in 3D-space, with radius r, and let the point be at a distance d from the camera C, see Figure 8. Assume the feature is projected to an image at scale (radius) s. The corresponding scale in the normalized image plane $z_1 = 1$ will be s/f, where f is the focal length. The projection point in the normalized image plane according to the pinhole camera model is $\boldsymbol{x}$ such that $\lambda\boldsymbol{x} = \boldsymbol{X}$, i.e. $\boldsymbol{X}$ is in the plane

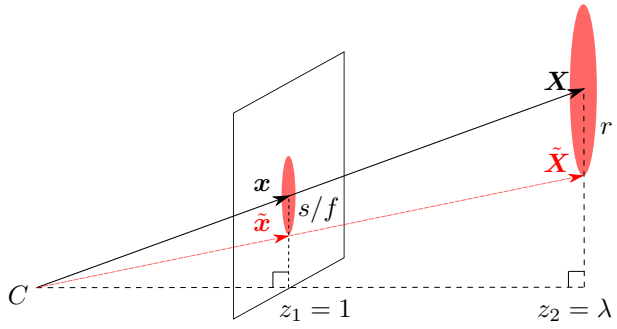

Figure 8. **Pinhole camera projection.** A feature of radius r in 3D space is projected into a feature of radius s in the image, equivalent to s/f in the normalized image plane. From similarity of triangles, we infer (28) and (29).

$z_2 = \lambda$. From Figure 8 we see that similarity of triangles $\triangle C\boldsymbol{x}z_1$ and $\triangle C\boldsymbol{X}z_2$ gives

$$\frac{1}{\lambda} = \frac{||\boldsymbol{x}||}{d}. \tag{28}$$

Now, let $\tilde{\boldsymbol{x}} = \boldsymbol{x} - (0, s/f, 0)$ and $\tilde{\boldsymbol{X}} = \boldsymbol{X} - (0, r, 0)$ be points at the edge of the feature at $\boldsymbol{x}$ and $\boldsymbol{X}$, respectively. Due to similarity of triangles $\triangle C\boldsymbol{x}\tilde{\boldsymbol{x}}$ and $\triangle C\boldsymbol{X}\tilde{\boldsymbol{X}}$ (see again Figure 8) we have

$$\frac{||\boldsymbol{x}||}{d} = \frac{s/f}{r}. \tag{29}$$

By combining (28) and (29), we get

$$\frac{1}{\lambda} = \frac{s/f}{r}. \tag{30}$$

For two corresponding projections of the same feature, we can write the relative depth as

$$\sigma \equiv \frac{\lambda'}{\lambda} = \frac{s/f}{s'/f'}\frac{r}{r} = \frac{f'}{f}\frac{s}{s'} = \frac{f'}{f}\gamma, \tag{31}$$

where we have used that r is independent of the projection.

8. Additional Experiment Details

When we evaluated the relative depth prediction in Section 4.3, we used a Sampson error inlier threshold of 1.0 for IMC-PT and 1.5 for ScanNet. When evaluating the solvers using RANSAC in Section 4.4, we used the parameters specified in Table 5.

The inference time of RelScaleNet was 55 µs per patch-pair when running with a batch size of 1024 on an NVIDIA RTX 3080 Ti.

Parameter	IMC-PT	ScanNet
Minimum iterations	1000	1000
Maximum iterations	100 000	100 000
Required confidence	0.9999	0.9999
IMC Inlier threshold	0.75	1.5

Table 5. **RANSAC parameters used in our experiments.** Different inlier thresholds were used for the two datasets IMC-PT and ScanNet.

9. Additional Results on IMC-PT

Here we present evaluation results on IMC-PT [13] divided by scene, for our solver and for the 5-point solver in LO-RANSAC. Additionally, we report mAA@10°. In Table 6, we present the results for the full set of image pairs; in Table 7, we present results for the top-5% hardest image pairs. In both tables, the average inlier ratio for each scene is also presented. We note that for scenes with low average inlier ratio (less than approximately 70 %), our 3-point solver is consistently faster than the 5-point solver.

Individual qualitative comparison of σ-estimation using RelScaleNet or SIFT is presented in Figures 9 and 10.

Scene	Inliers (%)	Method	AUC@5°	AUC@10°	AUC@20°	mAA@10°	RT(ms)
MR	87.26	5 pt. (Nistér [19])	**45.43**	**59.60**	**72.47**	**63.31**	19.6
		3 pt. + SIFT [**OURS**]	44.52	58.60	71.41	62.31	**16.7**
		3 pt. + RelScaleNet [**OURS**]	44.38	58.11	70.63	61.74	19.1
MC	83.33	5 pt. (Nistér [19])	**64.42**	**78.77**	**88.03**	**83.50**	**18.5**
		3 pt. + SIFT [**OURS**]	62.78	77.62	87.23	82.24	19.4
		3 pt. + RelScaleNet [**OURS**]	63.66	78.12	87.51	82.80	22.7
FCS	82.77	5 pt. (Nistér [19])	**71.03**	**81.68**	**88.78**	**85.90**	**19.9**
		3 pt. + SIFT [**OURS**]	70.11	81.09	88.42	85.31	21.0
		3 pt. + RelScaleNet [**OURS**]	70.34	81.30	88.59	85.54	22.6
LMS	78.16	5 pt. (Nistér [19])	**61.29**	**70.68**	**77.17**	**74.29**	11.4
		3 pt. + SIFT [**OURS**]	58.28	67.17	73.84	70.59	**11.0**
		3 pt. + RelScaleNet [**OURS**]	54.02	62.69	69.19	65.98	11.6
BM	76.88	5 pt. (Nistér [19])	**47.09**	**65.45**	**79.16**	**69.88**	14.7
		3 pt. + SIFT [**OURS**]	43.38	61.43	75.79	65.71	**13.0**
		3 pt. + RelScaleNet [**OURS**]	45.87	63.90	77.70	68.27	16.0
SF	75.43	5 pt. (Nistér [19])	**58.78**	**72.03**	**81.28**	**76.33**	18.2
		3 pt. + SIFT [**OURS**]	57.30	70.63	80.19	74.88	**13.5**
		3 pt. + RelScaleNet [**OURS**]	57.06	70.26	79.74	74.51	14.0
LB	75.00	5 pt. (Nistér [19])	**58.08**	**72.08**	**82.27**	**76.31**	14.5
		3 pt. + SIFT [**OURS**]	52.44	66.41	77.64	70.42	**12.0**
		3 pt. + RelScaleNet [**OURS**]	55.92	70.14	80.63	74.29	13.5
SPC	67.02	5 pt. (Nistér [19])	**60.00**	**74.15**	**84.03**	**78.54**	13.3
		3 pt. + SIFT [**OURS**]	57.96	72.18	82.46	76.46	**9.5**
		3 pt. + RelScaleNet [**OURS**]	58.60	72.65	82.72	76.94	10.1
PSM	61.87	5 pt. (Nistér [19])	**46.05**	**59.19**	**69.77**	**62.78**	20.0
		3 pt. + SIFT [**OURS**]	40.65	53.19	64.55	56.62	**6.8**
		3 pt. + RelScaleNet [**OURS**]	41.47	54.63	65.90	58.08	6.9
All	78.21	5 pt. (Nistér [19])	**56.89**	**70.43**	**80.39**	**74.59**	15.7
		3 pt. + SIFT [**OURS**]	54.30	67.80	78.16	71.85	**13.5**
		3 pt. + RelScaleNet [**OURS**]	54.63	68.06	78.16	72.12	15.0

Table 6. **Per-scene evaluation on IMC-PT [13].** The scenes are sorted in order of descending inlier ratio. The **best** and second best method in each category and metric is highlighted.

Scene	Inliers (%)	Method	AUC@5°	AUC@10°	AUC@20°	mAA@10°	RT(ms)
MR	55.39	5 pt. (Nistér [19])	**19.98**	**31.46**	**46.01**	**33.80**	7.5
		3 pt. + SIFT [**OURS**]	17.40	29.35	43.91	31.52	**4.0**
		3 pt. + RelScaleNet [**OURS**]	17.21	27.68	41.83	29.70	4.4
BM	45.00	5 pt. (Nistér [19])	**30.52**	**46.04**	**63.10**	**49.22**	8.1
		3 pt. + SIFT [**OURS**]	22.33	36.90	53.16	39.88	**3.7**
		3 pt. + RelScaleNet [**OURS**]	25.21	37.55	53.56	40.33	3.9
MC	42.64	5 pt. (Nistér [19])	**32.10**	**49.69**	**65.84**	**53.41**	13.0
		3 pt. + SIFT [**OURS**]	28.10	45.67	61.71	49.07	**5.1**
		3 pt. + RelScaleNet [**OURS**]	29.76	47.49	63.39	50.93	5.4
LMS	40.91	5 pt. (Nistér [19])	**10.03**	**15.95**	**24.28**	**17.00**	6.4
		3 pt. + SIFT [**OURS**]	8.32	14.00	23.11	14.75	2.5
		3 pt. + RelScaleNet [**OURS**]	7.49	13.22	21.94	14.04	**2.4**
FCS	32.31	5 pt. (Nistér [19])	**21.73**	**35.03**	**48.98**	**37.66**	15.1
		3 pt. + SIFT [**OURS**]	18.09	31.42	46.21	33.92	**8.7**
		3 pt. + RelScaleNet [**OURS**]	20.60	34.91	48.57	37.52	9.1
LB	31.33	5 pt. (Nistér [19])	**25.72**	**37.44**	**49.80**	**40.21**	25.7
		3 pt. + SIFT [**OURS**]	14.74	23.08	35.59	24.76	3.1
		3 pt. + RelScaleNet [**OURS**]	22.24	32.14	43.86	34.34	**3.0**
SPC	18.44	5 pt. (Nistér [19])	**8.36**	**17.64**	**31.89**	**19.04**	55.0
		3 pt. + SIFT [**OURS**]	7.54	15.45	29.23	16.61	3.9
		3 pt. + RelScaleNet [**OURS**]	5.25	13.19	26.75	14.27	**3.6**
SF	13.33	5 pt. (Nistér [19])	**6.84**	**10.96**	**15.53**	**11.66**	42.1
		3 pt. + SIFT [**OURS**]	5.52	8.99	13.78	9.59	**2.1**
		3 pt. + RelScaleNet [**OURS**]	3.18	6.99	10.94	7.51	2.2
PSM	1.88	5 pt. (Nistér [19])	1.84	4.24	9.43	4.49	64.5
		3 pt. + SIFT [**OURS**]	1.13	3.04	10.03	3.32	2.7
		3 pt. + RelScaleNet [**OURS**]	**2.43**	**5.35**	**11.13**	**5.76**	**2.6**
All	16.67	5 pt. (Nistér [19])	**12.13**	**21.79**	**33.55**	**23.57**	42.2
		3 pt. + SIFT [**OURS**]	8.72	16.76	28.61	18.30	**2.8**
		3 pt. + RelScaleNet [**OURS**]	9.47	18.04	29.69	19.72	**2.8**

Table 7. **Per-scene evaluation of top-5% hardest pairs from IMC-PT [13].** The scenes are sorted in order of descending inlier ratio. The **best** and second best method in each category and metric is highlighted.

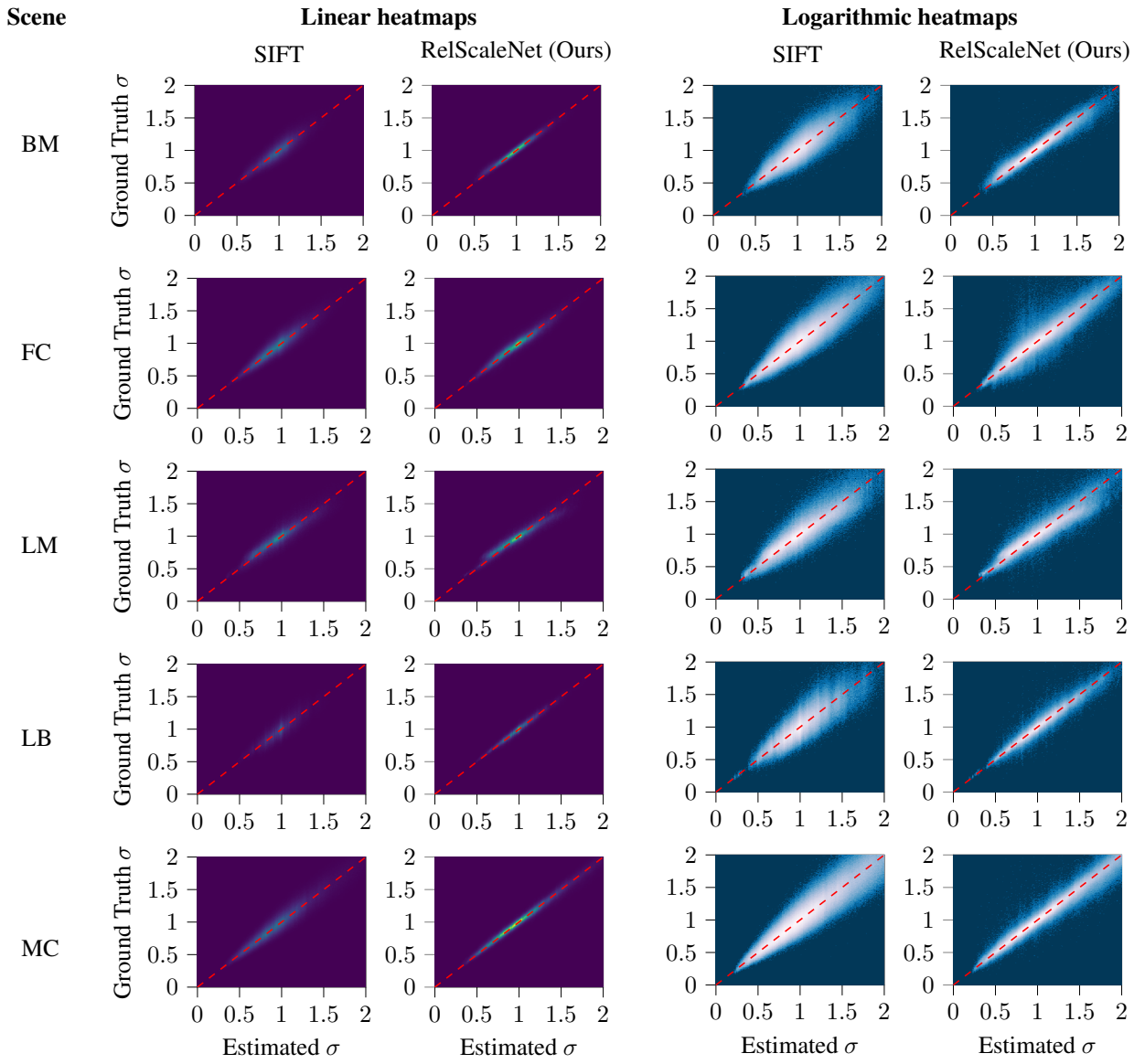

Figure 9. **Per-scene comparison of estimated relative depths.** The heatmaps show qualitative differences in estimated relative depths (σ) compared to the ground-truth, using SIFT or RelScaleNet on the test scenes in IMC-PT [13]. We present both linear and logarithmic heatmaps. The dashed red line shows where estimation is equal to ground truth, i.e. perfect estimation.

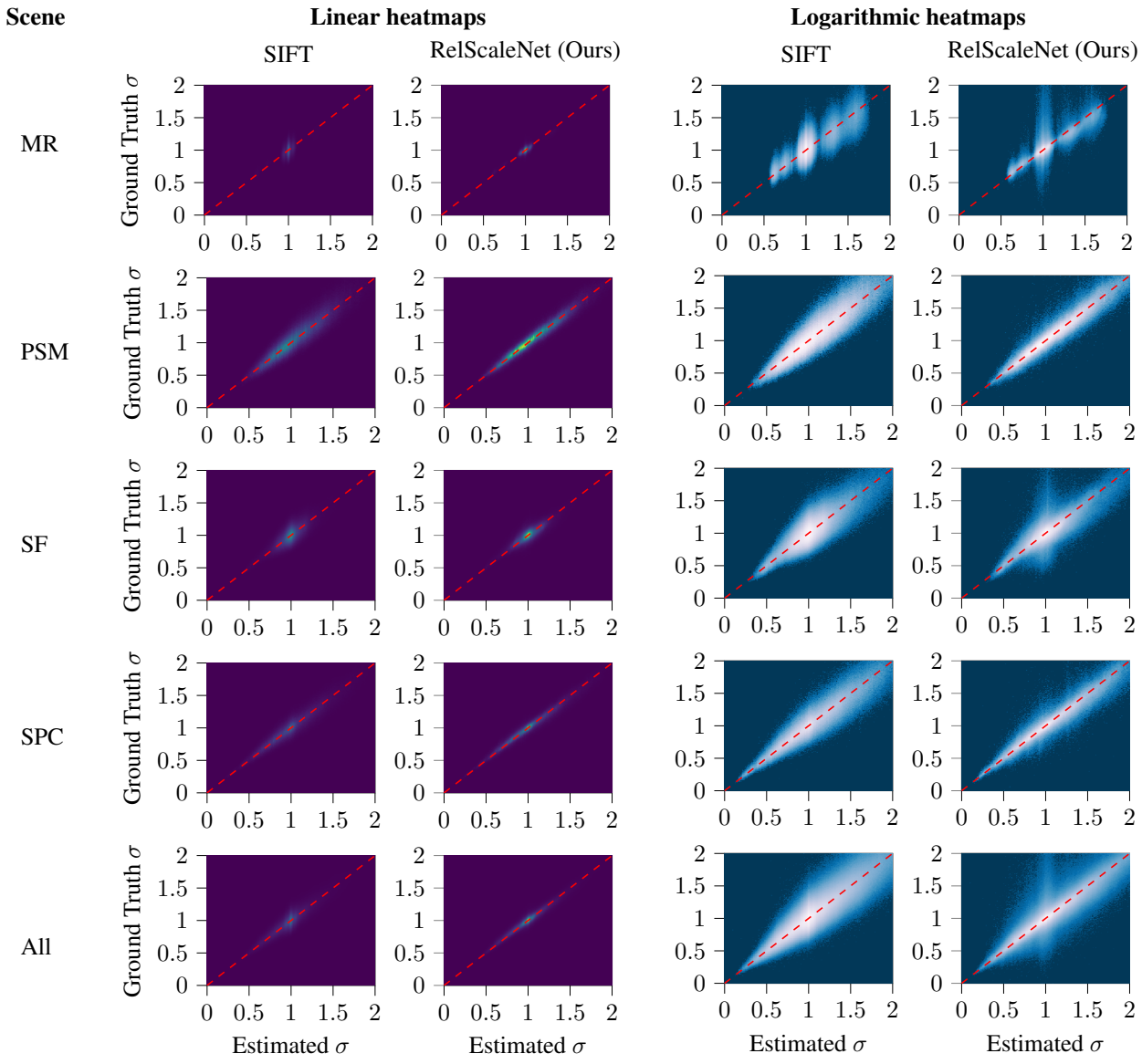

Figure 10. **Per-scene comparison of estimated relative depths.** The heatmaps show qualitative differences in estimated relative depths (σ) compared to the ground-truth, using SIFT or RelScaleNet on the test scenes in IMC-PT [13]. We present both linear and logarithmic heatmaps. The dashed red line shows where estimation is equal to ground truth, i.e. perfect estimation.