GeoCalib: Learning Single-image Calibration with Geometric Optimization

Alexander Veicht[1] Paul-Edouard Sarlin[1] Philipp Lindenberger[1]
Marc Pollefeys[1,2]

[1] ETH Zurich [2] Microsoft Mixed Reality & AI Lab

Abstract. From a single image, visual cues can help deduce intrinsic and extrinsic camera parameters like the focal length and the gravity direction. This single-image calibration can benefit various downstream applications like image editing and 3D mapping. Current approaches to this problem are based on either classical geometry with lines and vanishing points or on deep neural networks trained end-to-end. The learned approaches are more robust but struggle to generalize to new environments and are less accurate than their classical counterparts. We hypothesize that they lack the constraints that 3D geometry provides. In this work, we introduce GeoCalib, a deep neural network that leverages universal rules of 3D geometry through an optimization process. GeoCalib is trained end-to-end to estimate camera parameters and learns to find useful visual cues from the data. Experiments on various benchmarks show that GeoCalib is more robust and more accurate than existing classical and learned approaches. Its internal optimization estimates uncertainties, which help flag failure cases and benefit downstream applications like visual localization. The code and trained models are publicly available at https://github.com/cvg/GeoCalib.

Keywords: Camera calibration · Deep learning · Optimization

1 Introduction

Camera calibration consists of estimating the intrinsic and extrinsic parameters of a camera. This information is required for most image-based 3D applications, including metrology, 3D reconstruction, and novel view synthesis. This problem has been extensively studied, and many tools based on 3D geometry are available [51, 59, 73]. Since the process of image formation is well-understood, such tools can very accurately calibrate a camera from images taken in controlled lab conditions. The calibration can also be estimated in uncontrolled conditions, which generally requires additional sensors or multiple images observing the same scene, using structure-from-motion [5, 57, 60, 74] or SLAM [33, 40, 96].

In some applications, multiple images of the same scene are not available, such as in image editing, or multi-view constraints are not sufficient to accurately estimate the camera parameters, for example due to limited visual overlap across view. This occurs frequently when dealing with in-the-wild, crowd-sourced

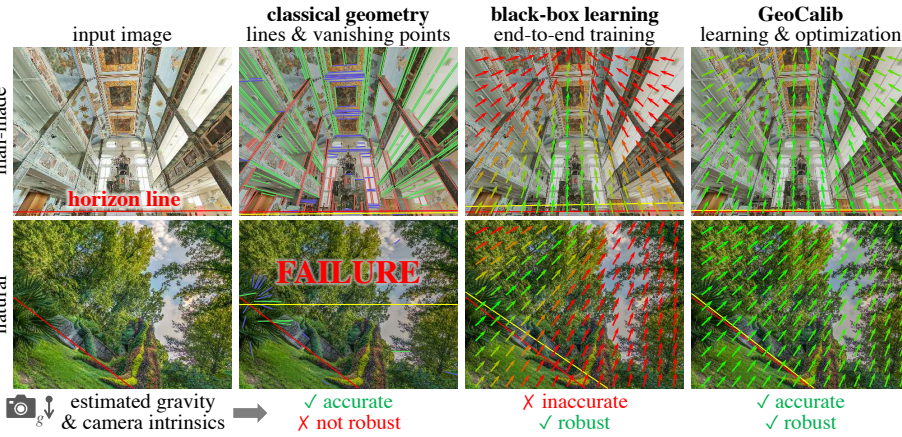


Fig. 1: Learning *vs.* geometry? To estimate the camera calibration from a single image, classical approaches struggle with environments devoid of lines while deep networks are so far not as accurate. GeoCalib combines the best of both: its learns to steer an optimization using diverse geometric and semantic cues learned end-to-end.

imagery, where each image is captured by a different camera [3, 75, 76, 89]. Visual cues visible in a single image can however help estimate some camera parameters, like the gravity direction, focal length, or distortion coefficients, without the need for multi-view cues.

Example of such geometric cues are straight lines, curves, and vanishing points. Estimating camera parameters from them has been extensively studied [6, 10, 50, 62, 88]. Because we have a good model of projective geometry, these approaches are extremely accurate. They are however limited to man-made environments in which straight lines are visible, and catastrophically fail when this condition is not met. Such low robustness significantly impairs their wide adoption.

Recent research has tackled the task of single-image calibration with deep networks trained in a supervised manner [14, 38, 45, 52, 77]. These approaches can leverage many more geometric and semantic cues and thus exhibit an impressively high robustness. To generalize well to different environment, they however require large amounts of training data that is costly to acquire. They are also far less accurate than their classical counterparts based on 3D geometry (Fig. 1). Intuitively, each deep network needs to relearn projective geometry from scratch when trained. Given finite model capacity, this can only be approximated within the domain of the training data, without any guarantee outside.

In this work, we introduce GeoCalib, a deep neural network (DNN) that leverages our knowledge of projective geometry through an optimization process. As this optimization is differentiable, GeoCalib learns end-to-end to estimate the vertical direction and the camera intrinsics given a single image. Our approach can thus learn the right visual cues without explicit supervision but does not need to learn the process of estimating camera parameters, which is better achieved with knowledge of 3D geometry. This improves the generalization to different environments with negligible overhead, which is important for practical

applications. Experiments on various benchmarks show that GeoCalib is both more robust and more accurate than existing classical and learned approaches.

Compared to black-box deep networks, GeoCalib has multiple practical benefits. When a subset of the parameters is known, *e.g.* the intrinsic parameters, GeoCalib can more accurately estimate the remaining parameters by leveraging this prior information. This makes it possible to handle different camera models, such as pinhole and fisheye, without any retraining. GeoCalib is also more interpretable: we can easily visualize the cues that it relies on, and the optimization uncertainties help flag failure cases and can benefit downstream applications. To support this, we show that GeoCalib can readily improve the accuracy of visual positioning. The code and trained models will be released publicly.

2 Related work

Classical geometric approaches for single-image calibration rely on parallel lines [31] that intersect at a vanishing point (VP) in the image. VP estimation in a single image can be categorized in two directions: i) unconstrained VP detection [12,42,94] which relies on coplanar lines and ii) approaches that assume a Manhattan world (3 orthogonal directions) [10,21,81] that is often found in man-made environments but not in natural ones. The vertical VP generally corresponds to the gravity direction. VP detection has been extensively studied with approaches based on RANSAC [6,10,50,88], optimization [11,82], exhaustive search [9,56,66], and even deep learning [7,48,81,95]. Some approaches also recover the focal length [50,62] or the distortion parameters, using curved lines, circular arcs, or covariant regions [50,65]. Our approach also solves an optimization based on low-level cues, including but not limited to lines, making it much more robust in natural and outdoor scenes.

Deep learning for single-image calibration has recently gained popularity. Existing research is divided into two categories: direct and indirect methods. Direct methods directly regress the camera parameters (focal length, distortion, horizon line) or classify them into bins [36, 52, 94]. This simple approach is typically less accurate than traditional methods because of missing geometric constraints. Indirect methods aim to alleviate this issue by predicting observable, lower-level cues. This includes classifying horizontal/vertical lines [45, 77] and regressing surface normals [90] or vector fields [38]. These cues are often used as auxiliary supervision [45, 77] or as inputs to a decoder network that regresses camera parameters [38]. Compared to classical algorithms, DNNs are usually more robust because they do not assume any scene configuration. They however fail to match their accuracy when sufficient geometric constraints are available. In this work, we combine deep priors with robust optimization, which enables the network to focus on well-constrained priors, boosting the estimation accuracy.

End-to-end learning with differentiable optimization has been recently popular in geometric computer vision, including for pose estimation [17,19,30,71], camera tracking [20,54,79,80,93], and image matching [18,69] or alignment [63,85].

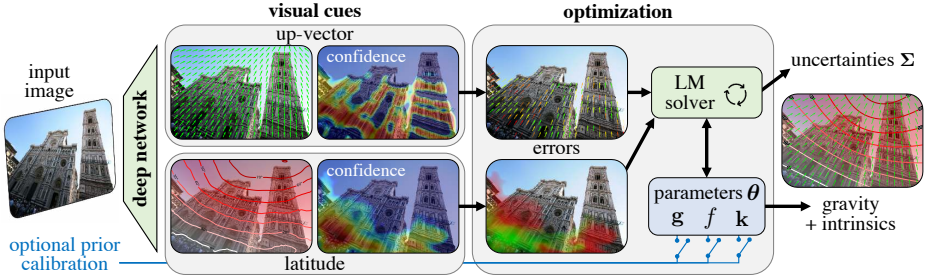


Fig. 2: Architecture of GeoCalib. A DNN predicts a Perspectivel Field with confidences, to which camera parameters are fitted with a Levenberg-Marquardt optimization. GeoCalib is trained end-to-end by supervising the optimized parameters. Priors over some of them or a different distortion model can be easily included without retraining.

In these, a DNN typically predicts observations on the image level, based on which an optimization problem is solved. It can be made differentiable by unrolling a fixed number of optimization steps or using implicit derivatives [67]. The DNN can be trained by only supervising the optimized quantity. The robustness to incorrect priors can be improved with carefully selected loss functions or by formulating the optimization through differentiable RANSAC [13, 16, 87].

For the task of single-image calibration, UprightNet [90] is closest to our work. It aligns normals in camera and world frames to estimate the gravity direction by solving a simpler weighted least squares problem with closed form solution. It cannot recover the camera intrinsics and its training requires ground-truth normals. In contrast, we use cues that are solely derived from the camera parameters and sufficient to constraint all of them. We show that the differentiable optimization improves both the generalization and the accuracy of GeoCalib.

3 Single-image calibration with GeoCalib

Camera calibration: A 3D point $\mathbf{P} = \begin{bmatrix} X & Y & Z \end{bmatrix}^\top$, expressed in the coordinate frame of the camera, maps to an image point $\mathbf{p} \in \mathbb{R}^2$ with the projection function $\Pi(\cdot)$. Assuming square pixels and no skew, we write for a pinhole camera $\Pi(\mathbf{P}) = f \begin{bmatrix} u & v \end{bmatrix}^\top + \mathbf{c}$, where $\begin{bmatrix} u & v \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} X & Y \end{bmatrix}$ are the normalized image coordinates, f is the focal length in pixels, and $\mathbf{c}$ is the principal point, which can often be assumed to be at the center of the image unless it has been cropped.

To handle lens distortion, various models are commonly used [27, 29, 39, 64, 72, 83, 84]. Many of them, including those based on polynomials, transform $\begin{bmatrix} u & v \end{bmatrix}$ into $\begin{bmatrix} u_d & v_d \end{bmatrix} := \mathcal{D}\left(\begin{bmatrix} u & v \end{bmatrix}, \mathbf{k}\right)$, where $\mathbf{k} \in \mathbb{R}^D$ are D distortion parameters. The undistortion function $\mathcal{D}^{-1}$ can be computed using an iterative process [22] or inverse parameters [26]. We generally refer to f, $\mathbf{c}$, and $\mathbf{k}$ as *intrinsic* parameters.

The gravity direction in the camera coordinate frame is the unit vector $\mathbf{g} = \begin{bmatrix} \mathbf{g}_x & \mathbf{g}_y & \mathbf{g}_z \end{bmatrix}$. It can be decomposed into two Euler angles, the roll, and the pitch. The third degree of freedom of the rotation, the yaw, is generally not observable from a single image unless one makes additional restrictive assumptions,

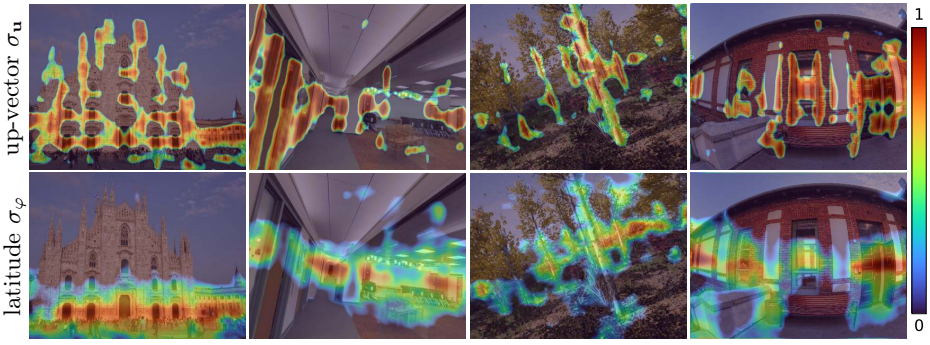


Fig. 3: Good features to calibrate. We show the confidences learned by GeoCalib for both components of the Perspective Field. The up-vector is most confident near vertical lines or upright objects like trees. The latitude is most confident near the horizon.

e.g. Manhattan-world. Our goal is to estimate $\mathbf{g}$, f, and $\mathbf{k}$, collectively referred to as $\boldsymbol{\theta}$, given a single image $\mathbf{I} \in \mathbb{R}^{H \times W}$ as input.

Overview: GeoCalib first infers visual cues – a Perspective Field, with associated confidences, from the input image with a DNN. An iterative optimization then fits the camera parameters $\boldsymbol{\theta}$ to be consistent with these cues (Fig. 2, Sec. 3.1). The DNN is then trained end-to-end by supervising the result of the optimization. We detail in Sec. 3.2 the multiple practical benefits of this simple approach.

3.1 Perspective alignment

Visual cues: We choose the *Perspective Field* [38] as visual cues driving the optimization. It is an over-parameterized, pixel-wise representation of the camera parameters and consists, for each pixel $\mathbf{p}$, of a unit up-vector $\mathbf{u}_{\mathbf{p}}$ and a latitude $\varphi_{\mathbf{p}}$. The up-vector is the projection of the up direction at $\mathbf{P}$, while the latitude $\varphi_{\mathbf{p}}$ is the angle between the ray $\mathbf{n} = \begin{bmatrix} u & v & 1 \end{bmatrix}^\top$ at pixel $\mathbf{p}$ and the horizontal plane:

$$\mathbf{u}_{\mathbf{p}} = \lim_{t \to 0} \frac{\Pi(\mathbf{P} - t\mathbf{g}) - \Pi(\mathbf{P})}{\|\Pi(\mathbf{P} - t\mathbf{g}) - \Pi(\mathbf{P})\|_2} \qquad \varphi_{\mathbf{p}} = \arcsin\left(\frac{\mathbf{n}^\top \mathbf{g}}{\|\mathbf{n}\|_2}\right) \tag{1}$$

This representation is visually observable: the up-vector corresponds to the direction of vertical lines, and the latitude is 0° on the horizon. It is not only geometric, but also semantic: the up-vector can be deduced from objects known to have a canonical orientation, like trees and humans. Unlike purely geometric cues like lines, it can be inferred in a wider range of environments. It is applicable to arbitrary camera models, though Jin *et al.* [38] mainly focused on perspective projection. Here we also apply it to distorted images.

We predict a pixel-wise Perspective Field $(\hat{\mathbf{u}}_{\mathbf{p}}, \hat{\varphi}_{\mathbf{p}})$ from the input image using a DNN. We employ a SegNeXt [32] encoder-decoder architecture, modified such that the outputs have the same spatial resolutions as the input. This provides more accurate constraints for the subsequent optimization.

Confidence: Predicting the Perspective Field is easy in some areas, such as near the horizon or vertical lines, but difficult in others, such as in texture-less areas. It needs to be accurate only for a subset of pixels to sufficiently constrain the camera parameters. The neural network thus also predicts a pixel-wise confidence $(\sigma_{\mathbf{u_p}}, \sigma_{\varphi_\mathbf{p}}) \in [0,1]$ for each component of the Perspective Field. The confidences are not directly supervised, but rather learned implicitly to help the optimization converge to the right solution. Figure 3 shows the predicted confidence after training. The up-vector is not limited to vertical lines; it is also confidently predicted on upright objects and curves in distorted images.

Objective function: We then find the camera parameters $\boldsymbol{\theta}$ that generate a Perspective Field $(\mathbf{u}(\boldsymbol{\theta}), \varphi(\boldsymbol{\theta}))$ similar to the one that is observed $(\hat{\mathbf{u}}, \hat{\varphi})$. This amounts to minimizing the objective function

$$E(\boldsymbol{\theta}) = \sum_{\mathbf{p} \in H \times W} \sigma_{\mathbf{u_p}} \| \underbrace{\mathbf{u_p}(\boldsymbol{\theta}) - \hat{\mathbf{u}}_\mathbf{p}}_{\mathbf{r}_{\mathbf{u_p}}} \|_2^2 + \sigma_{\varphi_\mathbf{p}} \| \underbrace{\sin \varphi_\mathbf{p}(\boldsymbol{\theta}) - \sin \hat{\varphi}_\mathbf{p}}_{\mathbf{r}_{\varphi_\mathbf{p}}} \|_2^2 \ . \tag{2}$$

This requires an explicit expression of the up-vector for the distortion camera model by solving for the limit of Eq. (1). We notice that the up-vector is collinear with the directional derivative of the projection function along the up direction, *i.e.* $\mathbf{u_p} \propto -\nabla \Pi(\mathbf{P}) \cdot \mathbf{g}$. For a radial distortion model such that $\mathcal{D}\left(\begin{bmatrix} u & v \end{bmatrix}, \mathbf{k}\right) = d(u,v,\mathbf{k}) \begin{bmatrix} u & v \end{bmatrix}$, the up-vector can be expressed as

$$\mathbf{u_p}(f, \mathbf{g}, \mathbf{k}) \propto \left(1 + \frac{1}{d(u,v,\mathbf{k})} \begin{bmatrix} u \\ v \end{bmatrix} \frac{\partial d(u,v,\mathbf{k})}{\partial (u,v)} \right) \begin{bmatrix} u\mathbf{g}_z - \mathbf{g}_x \\ v\mathbf{g}_z - \mathbf{g}_y \end{bmatrix} \ . \tag{3}$$

This can be derived similarly for other distortion models (see supplemental).

Optimization: Since Eq. (2) describes a non-linear least-squares problem, we find its minimum with the Levenberg–Marquardt (LM) algorithm [46, 58]. We parametrize the gravity $\mathbf{g}$ on the S^2 manifold [34, 35] since it has unit norm and the focal length by $\log f$ since it is strictly positive. The distortion parameters generally live in the Euclidean space, unless constrained by the camera model. At each iteration, we compute the update $\boldsymbol{\delta} \in \mathbb{R}^{3+D}$ by solving the linear system

$$\boldsymbol{\delta} = -\left(\mathbf{H} + \lambda \operatorname{diag}(\mathbf{H})\right)^{-1} \mathbf{J}^\top \mathbf{W} \mathbf{r} \quad \text{with} \quad \mathbf{J} = \frac{\partial \mathbf{r}}{\partial \boldsymbol{\theta}} \quad \text{and} \quad \mathbf{H} = \mathbf{J}^\top \mathbf{W} \mathbf{J} \ ,$$

where $\mathbf{r}$ stacks the residuals $\mathbf{r}_{\mathbf{u_p}}$ and $\mathbf{r}_{\varphi_\mathbf{p}}$ and $\mathbf{W}$ is a weight matrix whose diagonal stacks the corresponding confidences $\sigma_\mathbf{u}$ and σ_φ. The damping factor λ interpolates between the Gauss-Newton (λ=0) and gradient descent ($\lambda \to \infty$) strategies. It is usually adjusted at each iteration using heuristics [46, 55, 58]. We derive $\mathbf{J}$ analytically and provide it in the supplemental. The optimization stops when the update $\boldsymbol{\delta}$ is sufficiently small.

Initialization: The LM algorithm requires an initial guess. Since the objective function is not necessarily convex, the choice of initialization should matter. We consider several strategies. The trivial strategy initializes $\mathbf{g}$= $\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$, $\mathbf{k}$=$\mathbf{0}$, and f=$0.7 \max(W, H)$, assuming a common sensor size. Other, more complex

strategies are described in the supplemental. In practice, we have found that neither of them improves the performance over the trivial initialization.

Training: The LM optimization is differentiable, either by unrolling its steps and applying backpropagation to each of them [71, 80] or by leveraging implicit derivatives [67]. GeoCalib is thus end-to-end differentiable, from the estimated camera parameters to the input image. We train it by supervising the estimated parameters $\hat{\boldsymbol{\theta}}$ but also the intermediate perspective field, which accelerates the convergence of the training. This translates into the loss function

$$\mathcal{L} = \|\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}\|_\gamma + \beta \sum_{\mathbf{p} \in H \times W} \sigma_{\mathbf{u}_\mathbf{p}} \|\hat{\mathbf{u}}_\mathbf{p} - \mathbf{u}(\bar{\boldsymbol{\theta}})_\mathbf{p}\|_\gamma + \sigma_{\varphi_\mathbf{p}} \|\hat{\varphi}_\mathbf{p} - \varphi(\bar{\boldsymbol{\theta}})_\mathbf{p}\|_\gamma \ , \quad (4)$$

where $\bar{\boldsymbol{\theta}}$ are the ground-truth parameters, γ is the L1 loss, and β balances the two terms. The confidences are supervised only by the LM optimization and their gradient is detached in Eq. (4). They cancel the second term in areas that the network deems not useful for the optimization. No capacity is thus wasted to unnecessarily improve the prediction of the Perspective Field for all pixels.

3.2 Practical benefits

Because GeoCalib is based on well-understood optimization processes, it can be flexibly adapted to the constraints of real-world applications. None of the existing learned approaches provide such a degree of flexibility.

Arbitrary distortion model: Unlike most existing learned approaches, our formulation is not tied to a specific distortion model. One can train GeoCalib with a given model but optimize a different one at inference time without any retraining, as long as the Perspective Field is similarly observable. This is a significant gain given the large number of different distortion models used in practice. Our experiments show that a version of GeoCalib trained only on pinhole images can more accurately estimate a radial distortion than all existing approaches that operate on a single image. This makes it possible to handle heterogeneous datasets captured by many different cameras.

Partial calibration: A subset of the camera parameters might sometimes be already available. This includes camera intrinsics from a factory calibration, EXIF metadata, or structure-from-motion or a gravity direction estimated by an inertial measurement unit or assumed constant when the motion is restricted to be planar. Our optimization can leverage this information as a hard constraint, by fixing the associated parameters, or as a soft prior, by biasing the parameters with a simple regularization term, in which case the uncertainty of the priors can be leveraged. Our experiments show that such information directly improves the estimation accuracy of the remaining parameters.

Multi-image optimization: We can also couple the optimization across multiple images. For example, images captured by the same cameras share identical intrinsics, which should be optimized jointly, while each image has its own gravity direction. Unlike classical geometry tools, this does not require any covisibility

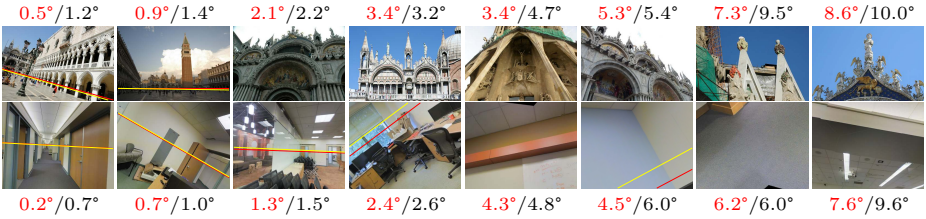


Fig. 4: Ranking images by uncertainty. We report the gravity *error* / *uncertainty* for 8 outdoor (top) and indoor (bottom) images from left-to-right, sorted by uncertainty. The estimated uncertainty correlates well with the ground truth error.

across the views. Differently, images taken by a multi-camera rig have distinct intrinsics but have a common gravity direction, given known relative poses.

Uncertainty estimation: The uncertainty of parameters estimated by the LM algorithm can be easily obtained via their covariance matrix, $\mathbf{\Sigma}_{\boldsymbol{\theta}} = \mathbf{H}^{-1}$ computed at convergence. It can be further decomposed into an uncertainty per parameter $\mathbf{g}$, f, and $\mathbf{k}$. This uncertainty can flag examples for which the estimate is likely wrong, *e.g.* due to poor geometric constraints in the optimization or excessively low-confident visual cues, for example due to occlusion, motion blur, or fully texture-less view (Fig. 4). This makes it easier to integrate the estimates of GeoCalib into a larger probabilistic framework, such as a factor graph [23, 24].

Comparison to Perspective Fields [38]: Our work is inspired by Jin *et al.* [38], who train a DNN to extract camera parameters from a pre-trained Perspective Field. This black-box approach is sensitive to the training distribution and lacks the flexibility described above. They also propose to refine its estimates with a first-order optimization based on Adam [41]. We found this to converge much slower than our LM optimization. Training it end-to-end and learning appropriate confidences is thus not practical. Unlike our approach, this optimization is sensitive to the initialization and yields marginal accuracy gains. We demonstrate the value of our improvements in an ablation study.

4 Implementation

OpenPano dataset: As prior works relied on proprietary datasets [36, 38, 52], reproducing them has been difficult. We have therefore assembled a new dataset, OpenPano, of panoramas that are publicly available from various sources [1, 2, 15]. OpenPano consists of 2.9k panoramas split into 2.6k/147/148 for training/validation/testing. Sampling 16 square images per panorama yields about 42k training images. Compared to the dataset used in [38], ours is much smaller but exhibits a better balance between indoor and outdoor scenes. It will be made publicly available to facilitate future research. See supplemental for more details.

Training: We trained variants of GeoCalib on the pinhole and distorted images from OpenPano, each for 49 epochs, with a learning rate of 10^{-4} and a batch size of 24. We linearly warm up the learning rate in the first 1000 steps and drop it after 25 and 42 epochs by a factor of 10. The LM optimization is stopped after

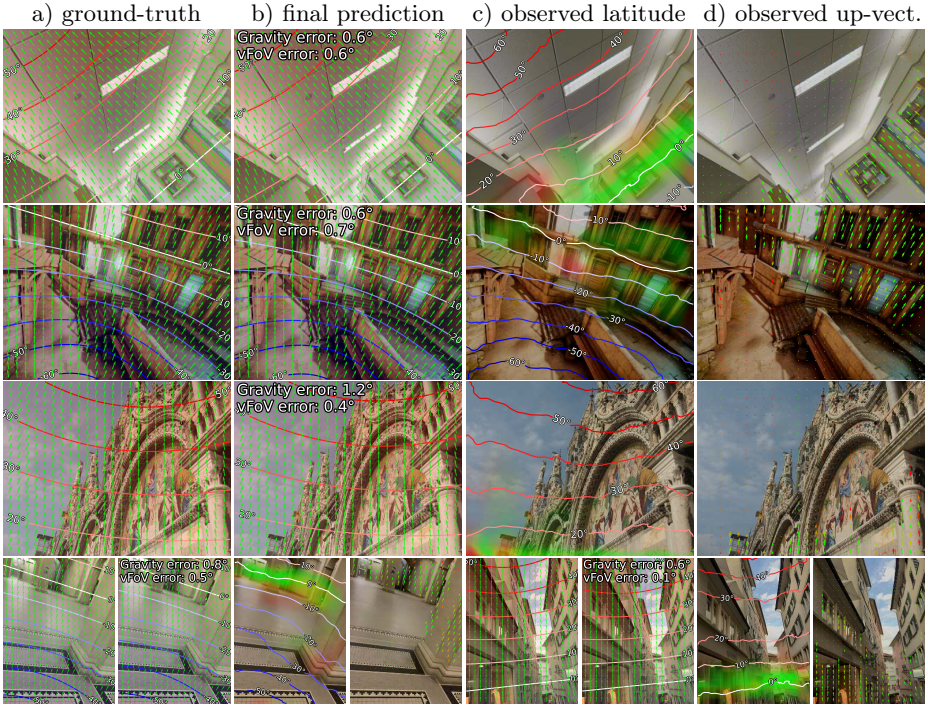


Fig. 5: Qualitative results. We show five examples of GeoCalib's prediction on Stanford2D3D [8], TartanAir [86], MegaDepth [47] and LaMAR [70] (x2). a-b) depict the generated up-vector and latitude field from ground-truth and estimated camera parameters. c-d) depict the latitude and up-vector fields observed by the DNN, with opacity from the learned confidences. Here, green denotes accurate and red inaccurate predicted fields w.r.t. ground-truth. GeoCalib learns to predict accurate fields and to discard observations in regions that are less informative, *e.g.* the floor.

10 iterations with an initial damping λ=0.1. The entire training takes one day on two RTX 4090 GPUs. Calibrating a single image takes about 100ms.

5 Experiments

We evaluate GeoCalib on single-image calibration and radial distortion estimation in a wide range of scenarios. Experiments are performed on a diverse range of real-world images (indoor, outdoor, and natural environments), and we analyze the impacts of our design decisions. We show some qualitative examples in Fig. 5. We also evaluate our gravity estimation as a prior for visual localization.

5.1 Gravity and Field-of-View estimation

We first compare GeoCalib to existing deep and classical approaches with pinhole images from four real-world datasets.

	Approach	Roll [degrees]				Pitch [degrees]				FoV [degrees]			
		error ↓	AUC ▷ 1/5/10° ↑			error ↓	AUC ▷ 1/5/10° ↑			error ↓	AUC ▷ 1/5/10° ↑		
Stanford2D3D [8]	DeepCalib* [52]	1.59	33.8	63.9	79.2	2.58	21.6	46.9	65.7	6.67	8.1	20.6	37.6
	Perceptual [36]	2.08	26.8	53.8	70.7	3.17	21.5	41.8	57.8	13.84	2.8	7.7	16.1
	CTRL-C [45]	3.04	23.2	43.0	56.9	3.43	18.3	38.6	53.8	8.50	7.7	18.2	31.5
	MSCC [77]	3.43	13.5	36.8	57.3	2.64	22.6	45.0	60.5	5.81	9.6	23.8	41.6
	ParamNet [38]	2.52	20.6	48.5	68.1	2.78	20.9	44.2	61.5	7.79	7.4	18.0	33.2
	ParamNet* [38]	1.14	44.6	73.9	84.8	1.94	29.2	56.7	73.1	9.01	5.8	14.3	27.8
	SVA [50]	-	21.7	24.6	25.8	-	15.4	19.9	22.4	-	6.2	11.5	15.2
	UVP [62]	0.52	65.3	74.6	79.1	0.95	51.2	63.0	69.2	3.65	**22.2**	39.5	51.3
	GeoCalib*	**0.40**	**83.1**	**91.8**	**94.8**	**0.93**	**52.3**	**74.8**	**84.6**	**3.21**	17.4	**40.0**	**59.4**
TartanAir [86]	DeepCalib* [52]	1.95	24.7	55.4	71.5	3.27	16.3	38.8	58.5	8.07	1.5	8.8	27.2
	Perceptual [36]	2.24	23.2	48.6	66.7	2.86	23.5	44.6	61.5	15.06	5.1	8.9	17.1
	CTRL-C [45]	1.68	32.8	59.1	74.1	2.39	24.6	48.6	65.2	5.64	10.7	25.4	43.5
	MSCC [77]	3.50	15.0	37.2	57.7	3.48	18.8	38.6	54.3	11.18	4.4	11.8	23.0
	ParamNet [38]	2.33	23.3	51.4	71.0	2.87	19.9	43.8	62.9	6.04	8.5	22.5	40.8
	ParamNet* [38]	1.63	34.5	59.2	73.9	3.05	19.4	42.0	60.3	8.21	6.0	16.8	31.6
	SVA [50]	9.48	32.4	39.6	44.1	18.46	21.2	28.8	34.5	43.01	8.8	16.1	21.6
	UVP [62]	0.89	52.1	64.8	71.9	2.48	36.2	48.8	58.6	9.15	**15.8**	25.8	35.7
	GeoCalib*	**0.43**	**71.3**	**83.8**	**89.8**	**1.49**	**38.2**	**62.9**	**76.6**	**4.90**	14.1	**30.4**	**47.6**
MegaDepth [47]	DeepCalib* [52]	1.41	34.6	65.4	79.4	5.19	11.9	27.8	44.8	11.14	5.6	12.1	22.9
	Perceptual [36]	1.07	47.9	72.4	83.2	3.49	19.8	39.1	54.2	13.40	2.9	8.2	16.8
	CTRL-C [45]	0.88	54.5	75.0	84.2	4.80	16.6	33.2	46.5	18.65	2.0	5.8	12.8
	MSCC [77]	0.90	53.1	72.8	82.1	5.73	19.0	33.2	44.3	10.80	6.0	14.6	26.2
	ParamNet [38]	1.46	37.0	66.4	80.8	3.53	15.8	37.3	57.1	10.98	5.3	12.8	24.0
	ParamNet* [38]	1.17	43.4	70.7	82.2	3.99	15.4	34.5	53.3	11.01	3.2	10.1	21.3
	SVA [50]	-	31.9	35.0	36.2	-	13.6	20.6	24.9	-	9.4	16.1	21.1
	UVP [62]	0.51	69.2	81.6	86.9	4.59	21.6	36.2	47.4	10.92	8.2	18.7	29.8
	GeoCalib*	**0.36**	**82.6**	**90.6**	**94.0**	**1.94**	**32.4**	**53.3**	**67.5**	**4.46**	**13.6**	**31.7**	**48.2**
LaMAR [70]	DeepCalib* [52]	1.15	44.1	73.9	84.8	4.68	10.8	28.3	49.8	10.93	0.7	13.0	24.0
	Perceptual [36]	1.29	40.0	68.9	81.6	2.83	21.2	44.7	62.6	17.78	3.0	5.3	10.7
	CTRL-C [45]	1.20	43.5	70.9	82.5	1.94	27.6	54.7	70.2	5.64	9.8	24.6	43.2
	MSCC [77]	1.44	39.6	60.7	72.8	3.02	20.9	41.8	55.7	14.78	3.2	8.3	16.8
	ParamNet [38]	1.30	38.7	69.4	82.8	2.77	19.0	44.7	65.7	15.15	1.8	6.2	13.2
	ParamNet* [38]	0.93	51.7	77.0	86.0	2.15	27.0	52.7	70.2	14.71	2.8	6.8	14.3
	SVA [50]	-	8.6	9.2	9.7	-	3.4	5.7	7.0	-	1.2	2.7	4.1
	UVP [62]	0.38	72.7	81.8	85.7	1.34	42.3	59.9	69.4	5.57	15.6	30.6	43.5
	GeoCalib*	**0.28**	**86.4**	**92.5**	**95.0**	**0.87**	**55.0**	**76.9**	**86.2**	**3.03**	**19.1**	**41.5**	**60.0**

Table 1: Evaluation on diverse datasets. Approaches marked as * were retrained on our dataset OpenPano. We color the best and second best results for each metric. GeoCalib is more accurate than approaches based on learning and more robust than those based on lines and vanishing points.

Setup: Following [38], we evaluate uncalibrated pinhole scenarios, *i.e.* no lens distortion. For each image, we evaluate the gravity estimation in terms of angular roll and pitch errors (in degrees), and the focal length in terms of the vertical field-of-view error (FoV, in degrees). For each metric, we report the median error and the Area Under the recall Curve (AUC) up to 1/5/10°.

Datasets: We conduct this experiment on four popular datasets not seen during training. i) Stanford2D3D [8] consists of images samples from 360° panoramas captured inside university buildings. ii) TartanAir [86] provides images from photo-realistic simulated environments and captures dynamic objects and various appearance changes. Its images exhibit large variations in gravity directions but share the same FoV. For both datasets, we use the same test set as [38], with about 2k images each. iii) The MegaDepth dataset [47] was built from crowd-sourced images covering popular phototourism landmarks using SfM. We

Fig. 6: Image undistortion. GeoCalib can robustly predict lens distortion from a single image (left), which can be used to rectify images (right) from any source.

Approach	FoV [degrees]	$\mathbf{k}_1$	Pixel Distortion Error			
	error ↓	error ↓	recall ▷ 0.5/1/3/5px ↑			
Perceptual [36]	13.50	(spherical)	0.97	3.42	19.67	30.01
SVA [50]	-	(divisional)	22.28	32.01	52.50	63.67
DeepCalib* [52]	10.80	0.09	23.89	34.32	55.82	67.06
ParamNet* [38]	11.17	0.10	22.56	32.37	53.07	64.34
GeoCalib - pinhole*	4.55	0.07	25.92	37.10	60.40	72.01
GeoCalib*	**4.38**	**0.06**	**28.36**	**40.28**	**64.14**	**75.57**

Table 2: Calibration of distorted images. On Internet images from the MegaDepth [47] dataset with radial distortion [27], GeoCalib estimates a more accurate distortion than all other approaches, even if it is trained only on pinhole images.

align the respective 3D models to gravity using COLMAP [74] and sample a total of 2k images with varying intrinsics from the scenes in the IMC 2021 test set [37]. iv) LaMAR [70] is an AR dataset for visual localization, captured over multiple years in university buildings and a city center. We use 2k images from the phone sequences.

Baselines: We benchmark our method against the deep methods DeepCalib [52], CTRL-C [45], Perceptual [36], MSCC [77] and ParamNet [38]. We follow the suggested evaluation setup of each method and refer to the supplementary material for more details. For fairness, we also retrain DeepCalib [52] and ParamNet [38] on our dataset. We also evaluate two recent classical approaches based on VPs: SVA [50] and UVP [62], which both assume a Manhattan world (3 orthogonal vanishing points) and rely on minimal solvers for VP estimation.

Results: Table 1 shows that GeoCalib largely improves on top of all deep single-image calibration networks, and outperforms classical methods in all metrics, except for the finest threshold on FoV on TartanAir [86] and Stanford2D3D [8]. UVP [62] assumes a Manhattan world, and this stronger assumption about scene configuration enables slightly more accurate predictions on easy samples, but completely fails in other scenarios. In contrast, GeoCalib is the first deep method that consistently matches or surpasses the accuracy of classical methods without any assumption on the scene, thus combining the accuracy of classical methods with the robustness of deep neural networks. We conclude that our method is more accurate in gravity estimation than FoV, likely due to the weaker constraint from latitude, which is generally harder to predict.

5.2 Lens distortion

We now evaluate how GeoCalib can handle radial distortion and estimate it.

Setup: We evaluate camera distortion estimation on the MegaDepth dataset [47] with crowd-sourced, distorted images. These images were taken from various

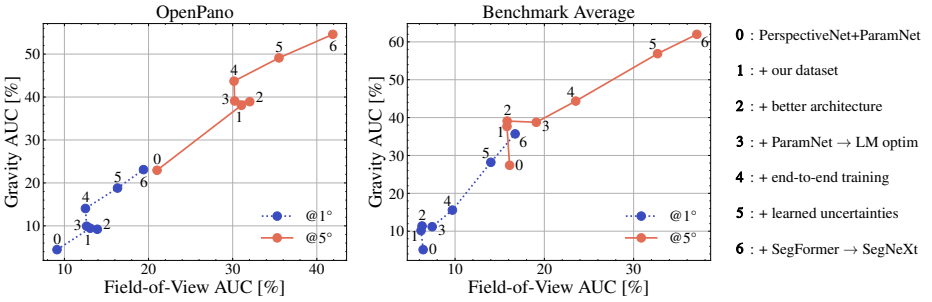


Fig. 7: Ablation study. Going from PerspectiveNet+ParamNet [38] to GeoCalib.

sensors and thus exhibit a diverse distribution of distortions. The ground-truth distortion models were obtained from SfM [74]. Similar to these GT camera models, we use a polynomial distortion model [27] with one parameter and report the median error. We also report the pixel distortion error, defined as the Euclidean distance between the pixel distorted by the ground truth camera model and the distortion from the predicted model but with GT focal length. In contrast to previous works [50], we use the GT focal length to disentangle the distortion estimation from FoV, which can bias the evaluation if the FoV estimate is incorrect. We average the pixel distortion errors over each image and compute the recall of the average distortion error within an image over the entire dataset. For completeness, we also report the median FoV error for each method.

Baselines: We compare our method against Perceptual [36], which uses a spherical distortion model [84], SVA [50] (divisional model [29]), as well as DeepCalib [52] and PerspectiveNet+ParamNet [38] trained on our dataset with a single parameter polynomial distortion model [27]. We evaluate both variants of GeoCalib trained with pinhole and distorted images.

Results: Table 2 reports the results. GeoCalib-pinhole already improves over all baselines, suggesting that the model can zero-shot generalize to radial distortion through optimization. Training our model on distorted images further boosts the accuracy. Figure 6 shows that GeoCalib can accurately handle strong lens distortion on a diverse range of images.

5.3 Insights

Ablation study: We perform an extensive ablation study to verify the design decisions of our method. We start from the closest baseline PerspectiveNet + ParamNet [38], trained on *360 cities*, and train 5 different models on our OpenPano dataset. We evaluate the trained models on the 2k test images sampled from OpenPano as well as on the previous benchmarks, averaging their results.

We plot the results in Fig. 7. **(1)** While OpenPano is about 5 times smaller than *360 Cities*, it is more balanced across different domains, resulting in improvements on the test set and on the benchmarks. **(2)** The improved architecture as described in Sec. 3 slightly increases the accuracy both in- and out-of-domain. **(3)** When switching from ParamNet [38] to our second-order optimization, the

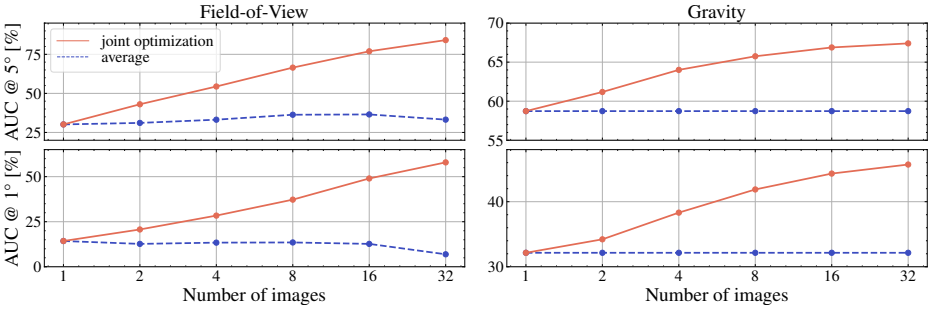


Fig. 8: Multi-image optimization. Simultaneously optimizing multiple images with shared intrinsic parameters improves the estimation accuracy of both field of view (left) and gravity direction (right). This is useful for calibrating an image sequence.

Dataset	known?		Gravity [degrees]				FoV [degrees]			
	gravity	FoV	error ↓	AUC ▷ 1/5/10° ↑			error ↓	AUC ▷ 1/5/10° ↑		
Stanford2D3D	✗	✗	1.12	45.6	71.5	82.7	3.21	17.4	40.0	59.4
	✗	✓	**0.86**	**57.0**	**79.0**	**87.0**	-	-	-	-
	✓	✗	-	-	-	-	**2.30**	**24.7**	**50.2**	**67.5**
TartanAir	✗	✗	1.69	31.6	58.6	73.4	4.90	14.1	30.4	47.6
	✗	✓	**1.18**	**44.7**	**66.8**	**78.6**	-	-	-	-
	✓	✗	-	-	-	-	**3.21**	**20.0**	**40.4**	**56.9**
MegaDepth	✗	✗	2.10	28.1	50.8	65.8	4.46	13.6	31.7	48.2
	✗	✓	**1.41**	**37.6**	**62.3**	**74.2**	-	-	-	-
	✓	✗	-	-	-	-	**2.42**	**27.0**	**48.8**	**64.3**
LaMAR	✗	✗	0.99	50.1	74.5	84.5	3.03	19.1	41.5	60.0
	✗	✓	**0.82**	**58.2**	**80.2**	**88.2**	-	-	-	-
	✓	✗	-	-	-	-	**2.28**	**23.5**	**50.0**	**67.4**

Table 3: Partial calibration. When a subset of the parameters is known, the optimization is better constrained and estimates more accurately the remaining parameters.

FoV estimation drops in domain but gets more accurate across the benchmarks. This shows that ParamNet is unable to generalize to out-of-domain images. **(4)** Supervising the result of the optimization and **(5)** learning uncertainties significantly boost the accuracy across the board as this i) allows GeoCalib to increase the weight for accurate predictions, improving both accuracy and robustness, and ii) enables it to focus more on well-constrained areas of the image, while saving capacity in less informative regions. **(6)** Finally, replacing the SegFormer [92] backbone with the more recent SegNeXt [32] also brings improvements. Despite relying on classical geometry, GeoCalib will thus benefit from further advances in deep learning architectures. See the supplemental for an extended analysis.

Partial Calibration: In many real-world scenarios, we either have access to GT gravity (from IMU) or known intrinsics. We conduct an additional experiment as in Sec. 5.1 to showcase how GeoCalib can leverage such priors. Here, we either fix the gravity direction or the intrinsics (FoV) in the optimization (Eq. (2)) and predict the other. We report the angular error of gravity and FoV.

Table 3 shows that partial calibration consistently improves the accuracy of our joint optimization. Fixing the FoV improves the gravity estimation by up to

Approach	DUC 1			DUC2		
	Recall at (0.25m,10°) / (0.5m,10°) / (1.0m,10°) ↑					
Baseline (SP+SG)	43.4	66.7	78.3	51.9	74.8	**78.6**
+ gravity in refinement	44.4	68.2	78.8	54.2	74.8	**78.6**
+ gravity in RANSAC	**44.9**	**69.2**	**79.3**	**55.0**	**76.3**	**78.6**
gravity from UVP [62]	43.9	67.7	78.8	52.7	**76.3**	**78.6**

Table 4: Visual Localization on InLoc [78]. The gravity prior estimated by GeoCalib improves pose estimation and is more effective than the estimate of UVP [62].

13.1% AUC@1° on TartanAir [86]. On the other hand, fixing the gravity direction improves the intrinsics estimation across all thresholds equally.

Multi-image optimization: We evaluate GeoCalib's ability to jointly estimate camera intrinsics for multiple images captured by the same camera. We randomly sample sets of N images from the TartanAir [86] dataset and report the AUC of the gravity and FoV at 1° and 5°.

Figure 8 shows that our joint optimization estimates progressively more accurate FoVs as more images are considered simultaneously. This also improves the estimate of the gravity, even though it is different for each image. In contrast, simply averaging the independently-estimated FoVs over all images is less effective and cannot benefit the gravity estimation. Thus, GeoCalib can effectively leverage geometric priors across images, even if they do not have any visual overlap.

5.4 Indoor Visual Localization

We evaluate the accuracy of gravity-aided visual localization on the InLoc [78] indoor dataset following hloc [25, 68, 69]. We report the pose recall under three different thresholds on the two locations DUC1 and DUC2. The absolute pose is estimated with RANSAC and LM-refinement from 2D-3D correspondences.

Because the gravity estimates are noisy, we avoid upright solvers [43, 44] and instead add soft constraints. In pose refinement, we add a regularization on the rotation towards the estimated gravity [4]. In RANSAC, we add a constant reward to the MSAC score if the estimated gravity is within 2σ of our estimated gravity uncertainty. As a baseline, we use the gravity estimated by UVP [62].

We report the results in Tab. 4. Adding the gravity constraint in both RANSAC and pose refinement improves localization accuracy. The baseline UVP [62] yields fewer improvements because of less accurate priors.

6 Conclusion

This paper introduces GeoCalib, a new approach to single-image calibration that combines the best of learning and geometry. Thanks to its differentiable optimization, it learns strong priors that make it both more accurate and more robust than existing approaches, with a strong generalization to different environments. GeoCalib offers much flexibility in terms of camera models, priors, and uncertainties and is thus easier to integrate in downstream applications.

Acknowledgments: Thanks to Xu Song for helping to evaluate MSCC. Thanks to Jean-François Lalonde for allowing us to release models trained on the Laval Indoor dataset. Thanks to Shaohui Liu and Linfei Pan for their valuable feedback. Philipp Lindenberger was supported by an ETH Zurich RobotX Fellowship.

A Extended ablation study

	Approach	Gravity [degrees]				FoV [degrees]			
		error ↓	AUC ▷ 1/5/10° ↑			error ↓	AUC ▷ 1/5/10° ↑		
OpenPano	PerspectiveNet+ParamNet [38]	4.99	4.5	23.0	43.7	6.63	9.1	21.0	37.7
	+ our dataset (OpenPano)	3.08	9.4	38.1	60.2	4.40	13.1	31.0	51.5
	+ better architecture	3.01	9.2	38.9	60.9	4.23	13.9	32.0	52.5
	+ ParamNet → SGD optim	3.03	10.3	39.1	61.0	4.75	12.4	29.3	49.2
	+ ParamNet → LM optim	3.03	9.9	39.1	61.0	4.58	12.6	30.2	50.7
	+ end-to-end training	2.65	14.0	43.7	64.0	4.54	12.5	30.2	49.7
	+ learned uncertainties	2.24	18.8	49.1	67.3	3.85	16.3	35.5	54.4
	+ SegFormer → SegNext = **ours**	**1.88**	**23.1**	**54.6**	**71.1**	**3.02**	**19.4**	**41.9**	**60.7**
MegaDepth	PerspectiveNet+ParamNet [38]	4.06	5.4	28.3	51.5	10.98	5.3	12.8	24.0
	+ our dataset (OpenPano)	4.51	6.8	28.0	48.6	11.01	3.2	10.1	21.3
	+ better architecture	4.35	8.8	29.9	50.4	11.23	3.9	10.0	20.8
	+ ParamNet → SGD optim	4.22	7.4	29.5	50.9	8.88	5.8	15.1	29.0
	+ ParamNet → LM optim	4.12	8.2	30.1	51.2	8.70	5.4	15.6	29.6
	+ end-to-end training	3.31	10.8	35.4	57.6	7.22	8.1	18.4	34.8
	+ learned uncertainties	2.36	20.7	48.1	65.1	4.80	11.7	28.9	46.3
	+ SegFormer → SegNext = **ours**	**2.10**	**28.1**	**50.8**	**65.8**	**4.46**	**13.6**	**31.7**	**48.2**
LaMAR	PerspectiveNet+ParamNet [38]	3.46	6.9	34.4	59.4	15.15	1.8	6.2	13.2
	+ our dataset (OpenPano)	2.57	13.6	44.7	65.3	14.71	2.8	6.8	14.3
	+ better architecture	2.53	14.8	46.0	66.2	13.93	2.3	6.4	14.4
	+ ParamNet → SGD optim	2.54	14.2	45.4	66.0	11.71	4.9	12.5	23.0
	+ ParamNet → LM optim	2.54	14.4	45.4	66.1	11.24	4.6	12.5	23.6
	+ end-to-end training	2.14	19.5	51.0	71.2	6.34	7.9	20.2	38.5
	+ learned uncertainties	1.31	37.3	67.2	80.3	3.66	15.9	36.3	54.8
	+ SegFormer → SegNeXt = **ours**	**0.99**	**50.1**	**74.5**	**84.5**	**3.03**	**19.1**	**41.5**	**60.0**

Table 5: Ablation study. We color the best and second best results for each metric.

We perform an extensive ablation study to verify the design decisions of our method. We start from the closest baseline PerspectiveNet + ParamNet [38], trained on *360 cities*, and train 5 different models on our OpenPano dataset. We evaluate the trained models on the MegaDepth [47] and LaMAR [70] test splits and on the 2k test images sampled from OpenPano . We report the median and AUC of angular gravity and FoV errors.

We report the results in Tab. 5. **1)** Re-training PerspectiveNet + ParamNet [38] on OpenPano, which is 5 times smaller than *360 cities* [38], yields large improvements on the test split of OpenPano and LaMAR, but no improvements or slightly weaker generalization to MegaDepth. We conclude that despite the smaller size of our dataset, it enables strong generalization with higher quality and more evenly sampled images across domains at a fraction of the size. **2)** The improved architecture, as described in Sec. 3 increases the accuracy both in- and

	Approach	Gravity [degrees]				FoV [degrees]			
		error ↓	AUC ▷ 1/5/10° ↑			error ↓	AUC ▷ 1/5/10° ↑		
OpenPano	Trivial	34.36	0.1	0.3	1.1	22.60	1.8	5.5	11.0
	Heuristic	4.74	6.0	26.5	46.2	6.94	9.1	21.3	37.1
	Solver	6.01	11.4	30.3	41.5	11.79	10.4	21.8	32.0
	GeoCalib								
	+Trivial	1.88	23.1	54.6	71.1	3.02	19.4	41.9	60.7
	+Heuristic	1.90	23.1	54.6	71.0	3.09	19.3	41.8	60.5
	+Solver	2.03	21.4	51.8	67.7	3.31	18.3	39.7	57.8
MegaDepth	Trivial	21.90	0.7	2.8	7.4	21.07	0.5	2.8	8.3
	Heuristic	6.93	4.6	16.7	35.7	8.45	5.2	14.8	29.4
	Solver	8.69	7.0	18.8	31.9	16.67	4.0	10.5	19.0
	GeoCalib								
	+Trivial	2.10	28.1	50.8	65.8	4.46	13.6	31.7	48.2
	+Heuristic	2.09	28.2	51.0	65.9	4.50	13.6	31.6	48.1
	+Solver	2.14	27.8	50.7	65.6	4.54	13.4	31.4	47.8

Table 6: Initialization strategies. While the heuristic and solver strategies are more accurate than the trivial strategy, their impact on the result of the optimization is minimal because its convergence is robust.

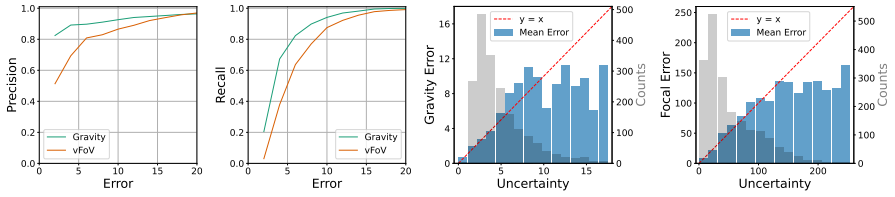
out-of-domain. The improved low-level features enable a more accurate prediction of up-vectors, which results in more precise gravity estimations. **3)** Replacing ParamNet with the optimization proposed in [38] drops the accuracy in-domain, but improves FoV estimation on MegaDepth and LaMAR. This suggests that ParamNet is not able to generalize well on FoV estimation. **4)** When switching from ParamNet [38] to our second-order optimization, we observe that it does not improve over the optimization proposed in [38] out-of-the-box. **5)** However, propagating the gradients through the LM-optimization enables the network to learn globally consistent up-vectors and latitude, which results in more robustness and generalization, concluded from large improvements for gravity and FoV estimation both in LaMAR and MegaDepth. **6)** Letting the network predict uncertainties on the perspective fields and propagating them through the LM optimization greatly improves the accuracy. These uncertainties i) allow the network to increase the weight for accurate predictions, improving both accuracy and robustness, and ii) enable the network to focus more on well-constrained areas of the image, while saving capacity in less informative regions. This results in a much stronger generalization of our model, supported by the 2x improvements at fine thresholds on MegaDepth and LaMAR. **7)** Finally, replacing the SegFormer [92] architecture with the more efficient SegNeXt [32] increases our model's accuracy, showing that GeoCalib will benefit from advances in deep learning architectures despite relying on classical geometry.

B Initialization strategies

We describe and evaluate additional initialization strategies mentioned in Sec. 3.1.

Trivial: The trivial strategy initializes $\mathbf{g}=\begin{bmatrix}0 & 1 & 0\end{bmatrix}$, $\mathbf{k}=\mathbf{0}$, and $f=0.7\max(W,H)$, assuming a common sensor size.

Heuristic: A heuristic strategy was proposed in [38]. We observe that, at the principal point $\mathbf{c}$, $\mathbf{u_c} \propto -\begin{bmatrix}\mathbf{g}_x & \mathbf{g}_y\end{bmatrix}$. Additionally, from Eq. (1), $\mathbf{g}_z = \sin\varphi_\mathbf{c}$.



a) Precision and recall curves b) Calibration plots

Fig. 9: Uncertainty Evaluation. a) Precision and recall curves show that the uncertainties can be used to confidently discard samples likely to be incorrect. b) The uncertainties follow the ideal $y{=}x$ curve and are thus fairly calibrated.

Therefore, $\mathbf{g}{=}\left[\alpha\mathbf{u}_\mathbf{c}^\top \sin\varphi_\mathbf{c}\right]$, where α is adjusted to ensure $\|\mathbf{g}\|{=}1$. Notably, the vertical field of view is linked to the difference in latitude, expressed as $\text{vFoV}{=}\varphi_{(\mathbf{c}_x,0)} - \varphi_{(\mathbf{c}_x,H)}$, which can be converted into an initial estimate for f.

Solver: Generalizing the heuristic strategy, we derive a simple minimal solver to infer $\mathbf{g}$ and f from any combination of two up-vectors and one latitude value. We embed this solver in a RANSAC [28] loop to maximize the confidence-weighted inlier count of up-vectors and latitudes. This can provide a more accurate estimate.

Results: We evaluate each initialization strategy and demonstrate its impact on the final estimation when used to initialize our optimization. We perform the evaluation on the OpenPano and MegaDepth [47] datasets and report the median and AUC of the angular gravity and vFoV errors. The results in Tab. 6 show that the increased accuracy of the heuristic strategy and the solver does not translate to any improvements in the final estimate of our model. We hypothesize that, once trained, the observed Perspective Field is sufficiently good to ensure convergence from any initial point.

C Uncertainty estimation

We evaluate the optimization uncertainties estimated by GeoCalib, as described in Sec. 3.2, on the OpenPano dataset.

We show precision and recall curves for gravity and vertical Field-of-View (vFoV) in Fig. 9-a). To assess the quality of our uncertainties, we discretize the errors by increasing thresholds and determine how accurately the predicted uncertainties classify samples into their respective categories. This means that for each threshold, we classify samples based on whether their predicted uncertainty surpasses the threshold or not and calculate the corresponding precision and recall values. The precision curve indicates that samples with low uncertainty generally exhibit low error rates. However, the low recall values for small errors suggest that the model is under-confident in its predictions.

We also show corresponding calibration plots in Fig. 9-b). To compute them, we bin the predictions based on the uncertainty and calculate the mean error per bin. We observe that the calibration tightly follows the optimal calibration of $y{=}x$ for samples with low uncertainty. For samples with high uncertainty, our

model tends to be under-confident in its prediction. The error is thus generally bounded by the predicted uncertainty.

D Distortion models

Following Eq. (1), the up-vector is collinear with the directional derivative of the projection function $\Pi(\cdot)$ along the up direction $-\mathbf{g}$, *i.e.* $\mathbf{u_p} \propto -\nabla\Pi(\mathbf{P}) \cdot \mathbf{g}$. We consider a radial distortion model such that $\mathcal{D}\left(\left[u\ v\right], \mathbf{k}\right) = d(u, v, \mathbf{k})\left[u\ v\right]$. Then

$$\nabla\Pi(\mathbf{P}) \propto \frac{\partial\mathcal{D}\left(\left[u\ v\right], \mathbf{k}\right)}{\partial\mathbf{P}} \tag{5}$$

$$\propto d(u, v, \mathbf{k})\frac{\partial\left[u\ v\right]^\top}{\partial\mathbf{P}} + \begin{bmatrix}u\\v\end{bmatrix}\frac{\partial d(u, v, \mathbf{k})}{\partial\left[u\ v\right]}\frac{\partial\left[u\ v\right]^\top}{\partial\mathbf{P}} \tag{6}$$

$$\propto \left(d(u, v, \mathbf{k}) + \begin{bmatrix}u\\v\end{bmatrix}\frac{\partial d(u, v, \mathbf{k})}{\partial(u, v)}\right)\frac{\partial\left[u\ v\right]^\top}{\partial\mathbf{P}} \tag{7}$$

$$\propto \left(1 + \frac{1}{d(u, v, \mathbf{k})}\begin{bmatrix}u\\v\end{bmatrix}\frac{\partial d(u, v, \mathbf{k})}{\partial(u, v)}\right)\begin{bmatrix}1 & 0 & -u\\0 & 1 & -v\end{bmatrix} . \tag{8}$$

The up-vector is thus such that

$$\mathbf{u_p}\left(f, \mathbf{g}, \mathbf{k}\right) \propto \left(1 + \frac{1}{d(u, v, \mathbf{k})}\begin{bmatrix}u\\v\end{bmatrix}\frac{\partial d(u, v, \mathbf{k})}{\partial(u, v)}\right)\begin{bmatrix}u\mathbf{g}_z - \mathbf{g}_x\\v\mathbf{g}_z - \mathbf{g}_y\end{bmatrix} . \tag{9}$$

For the pinhole camera model, this simplifies to

$$\mathbf{u_p}\left(f, \mathbf{g}, \mathbf{k}\right) \propto \begin{bmatrix}u\mathbf{g}_z - \mathbf{g}_x\\v\mathbf{g}_z - \mathbf{g}_y\end{bmatrix} . \tag{10}$$

For a polynomial radial distortion with $d(u, v, \mathbf{k}) = 1 + \mathbf{k}_1 r^2 + \mathbf{k}_2 r^4$, where $\mathbf{k} = (\mathbf{k}_1, \mathbf{k}_2)$ are the distortion parameters and $r^2 = u^2 + v^2$, we have

$$\frac{\partial d(u, v, \mathbf{k})}{\partial\left[u\ v\right]} = 2(\mathbf{k}_1 + 2\mathbf{k}_2 r^2)\left[u\ v\right] . \tag{11}$$

E Analytical Jacobians

We now report the analytical expression of the jacobians used in the optimization for the pinhole camera model. For each pixel, we write the up-vector and latitude

$$\mathbf{u} = \mathrm{norm}\left(\begin{bmatrix}u\mathbf{g}_z - \mathbf{g}_x\\v\mathbf{g}_z - \mathbf{g}_y\end{bmatrix}\right) = \mathrm{norm}\left(\bar{\mathbf{u}}\right) \qquad \sin\left(\varphi\right) = \mathrm{norm}\left(\mathbf{n}\right)^\top\mathbf{g} \tag{12}$$

where $\mathbf{n} = \left[u\ v\ 1\right]^\top$ is the ray and $\mathrm{norm}\left(\mathbf{x}\right) = \mathbf{x}/\|\mathbf{x}\|_2$ normalizes the input vector to have unit length. We write $\mathbf{J} = \left[\mathbf{J_u}\ \mathbf{J}_\varphi\right]$ and use the general chain rule:

$$\mathbf{J_u} = \frac{\partial\mathbf{r_u}}{\partial\boldsymbol{\delta}} = \frac{\partial\mathbf{r_u}}{\partial\mathbf{u}\left(\boldsymbol{\theta}\right)}\frac{\partial\mathbf{u}\left(\boldsymbol{\theta}\right)}{\partial\boldsymbol{\theta}}\frac{\partial\boldsymbol{\theta}}{\partial\boldsymbol{\delta}}, \tag{13}$$

$$\mathbf{J}_\varphi = \frac{\partial\mathbf{r}_\varphi}{\partial\boldsymbol{\delta}} = \frac{\partial\mathbf{r}_\varphi}{\partial\sin\left(\varphi\left(\boldsymbol{\theta}\right)\right)}\frac{\partial\sin\left(\varphi\left(\boldsymbol{\theta}\right)\right)}{\partial\boldsymbol{\theta}}\frac{\partial\boldsymbol{\theta}}{\partial\boldsymbol{\delta}}. \tag{14}$$

where we calculate the jacobians at $\boldsymbol{\delta}$=0. By simple derivation we can calculate the partial derivate for $\mathbf{u}$ as

$$\frac{\partial \mathbf{u}\left(f, \mathbf{g}, \mathbf{k}\right)}{\partial \mathbf{g}} = \left(\frac{1}{\left\|\bar{\mathbf{u}}\right\|_2}\mathbb{I} - \frac{\bar{\mathbf{u}}\bar{\mathbf{u}}^\top}{\left\|\bar{\mathbf{u}}\right\|_2^3}\right)\begin{bmatrix}-1 & 0 & u\\ 0 & -1 & v\end{bmatrix}, \tag{15}$$

$$\frac{\partial \mathbf{u}\left(f, \mathbf{g}, \mathbf{k}\right)}{\partial f} = \left(\frac{1}{\left\|\bar{\mathbf{u}}\right\|_2}\mathbb{I} - \frac{\bar{\mathbf{u}}\bar{\mathbf{u}}^\top}{\left\|\bar{\mathbf{u}}\right\|_2^3}\right)\begin{bmatrix}\mathbf{g}_z\\ \mathbf{g}_z\end{bmatrix}\begin{bmatrix}-(x-\mathbf{c}_x)/f^2\\ -(y-\mathbf{c}_y)/f^2\end{bmatrix}. \tag{16}$$

Similarly, we can calculate the partial derivatives for $\sin\left(\varphi\right)$ as

$$\frac{\partial \sin\left(\varphi_{\mathbf{p}}\left(f, \mathbf{g}, \mathbf{k}\right)\right)}{\partial \mathbf{g}} = \operatorname{norm}\left(\mathbf{n}\right)^\top \mathbb{I}, \tag{17}$$

$$\frac{\partial \sin\left(\varphi_{\mathbf{p}}\left(f, \mathbf{g}, \mathbf{k}\right)\right)}{\partial f} = \mathbf{g}^\top\left(\frac{1}{\left\|\mathbf{n}\right\|_2}\mathbb{I} - \frac{\mathbf{n}\mathbf{n}^\top}{\left\|\mathbf{n}\right\|_2^3}\right)\begin{bmatrix}-(x-\mathbf{c}_x)/f^2\\ -(y-\mathbf{c}_y)/f^2\\ 0\end{bmatrix}. \tag{18}$$

The Jacobians for radially distorted cameras can be derived in a similar way.

F Implementation details

In this section we provide more information and details on our dataset, how we train our models and evaluate our baselines.

F.1 Dataset

For our dataset we collect 360° panoramas in equirectangular format which covers 180° vertically and 360° horizontally. The dataset comprises 2912 panoramas sourced from hdrmaps [1], polyhaven [2], and parts of the Laval Photometric Indoor HDR dataset [15], where we manually select panoramas that are aligned with the gravity. The resulting dataset contains a good balance of about 800 outdoor and 2.1k indoor panoramas which are split into 2616 training, 147 validation and 148 testing panoramas. We create two iterations of the dataset: one featuring a simple pinhole camera and another assuming a radially distorted camera model. For both datasets we sample 16 crops per panorama with roll, pitch and vFoV sampled uniformly within $[\pm 45°]$, $[\pm 45°]$, and $[20, 105]$ respectively. To capture the relationship between the distortion parameter $\mathbf{k}$ and the vFoV, we sample $\hat{\mathbf{k}} = \mathbf{k}/\text{vFoV}$ from a truncated normal with μ=0 and σ=0.07 bounded to [-0.3, 0.3] for the distorted dataset. Since some panoramas are padded with black pixels to be equirectangular, we remove images with $\geq 1\%$ of black pixels. After this, we are left with 37015 images for training, 2100 for validation and 2128 for testing. Figure 10 shows the diversity of our dataset with difficult and easy samples, indoor and outdoor images as well as variation in gravity and camera intrinsics.

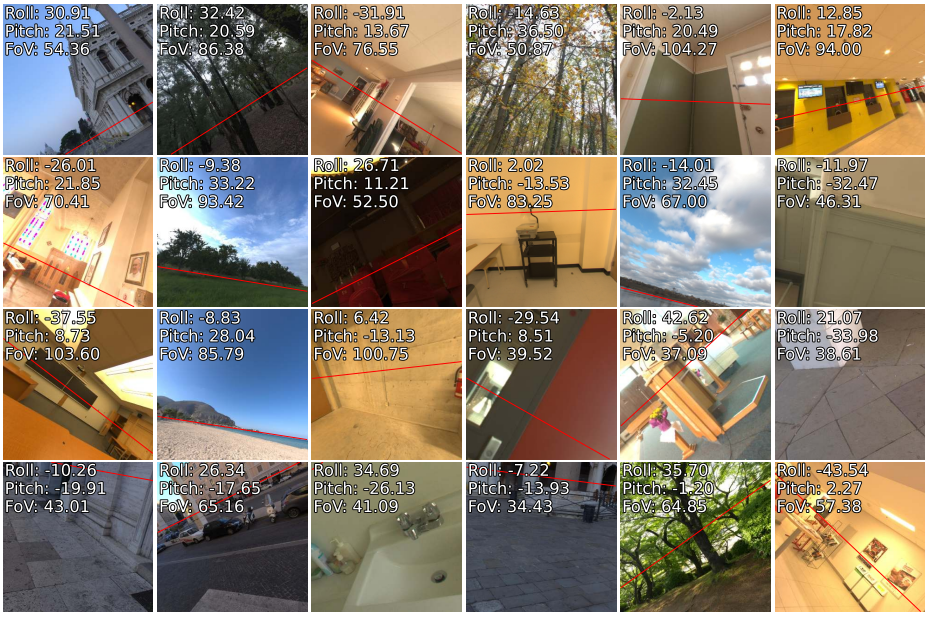


Fig. 10: Samples from our OpenPano-pinhole dataset. The dataset includes images sampled from panoramas captured in diverse outdoor and indoor scenes. We show the ground truth horizon as a red line.

F.2 Training

GeoCalib is trained for 75k steps with a batch size of 24 and a base learning rate of 10^{-4} using the AdamW [53] optimizer. We make use of a linear warm-up schedule in the first 1k steps and drop the learning rate after 40k and 65k steps by a factor of 10. The input resolution is 320×320 and we apply extensive color, blur and resize augmentations during training. During inference, non-square images are first resized to 320 along the shorter edge and afterwards cropped to fit the required multiple of 32 by our model. The resulting camera is re-scaled to fit the original image dimensions.

F.3 Indoor Visual Localization

RANSAC: We use the P3P RANSAC solver from PoseLib [44] as a baseline, and add the gravity constraint in their framework. We add a constant, multiplicative reward to the MSAC scoring if the angle between the estimated gravity from our network and the gravity obtained from the samples' pose estimate (the second column of the rotation matrix) is smaller than twice the uncertainty predicted by our network (in degrees).

In contrast to upright solvers [43], this weak constraint is less susceptible to noise and can directly utilize the uncertainty estimates of our model.

Absolute Pose Refinement: In the non-linear absolute pose refinement, we add an L2-regularization term to the optimization, which penalizes the deviation

of the rotation matrix from our estimated gravity. We use a simple L2-loss and weigh the regularization by the number of inlier correspondences. We use the Ceres-solver [4] to build the non-linear pose refinement, similar to COLMAP [74].

F.4 Evaluation

We provide further details on our evaluation setup.

Metrics: To account for inaccuracies in the ground truth, especially for datasets derived from structure-from-motion like MegaDepth and LaMAR, we compute the AUC for curves clipped to a minimum of 1°. When an approach fails to return a valid result [50, 62], we set its error to $+\infty$.

DeepCalib [52]: We re-implement the model and its training following the paper. We train the model for 20k steps using the Adam optimizer [41] with a learning rate of 10^{-4} and a batch size of 32 on our distorted dataset. The input images are resized to 320×320 and we use the same augmentations as for GeoCalib. We follow the parameterization proposed in [52] and train the model using 256 bins per parameter. The model is trained to minimize the negative log likelihood, and we apply early stopping on the validation set.

Perceptual [36]: We run inference using the online demo provided by the authors, which is based on a ConvNeXt CNN backbone [49].

CTRL-C [45]: We make use of the official code released by the authors to run inference on our benchmarks using the model weights trained on the SUN360 [91] dataset. The input resolution is 512×512.

MSCC [77]: We ask the authors to run inference on our benchmarks without sharing the ground-truth parameters.

ParamNet [38]: We re-implement ParamNet and validate our implementation by comparing its inference to the original code-base on various benchmarks. The model is re-trained on our pinhole dataset following the schedule proposed in the official repo: we first pre-train PerspectiveNet, then add ParamNet to train the full model. Each is trained for 45k steps with a base learning rate of 0.01 using the SGD optimizer and a batch size of 64. After a 1k step warm-up, the learning rate drops by a factor of 10 after 30k and 40k steps. The input resolution is 320×320, and we use the same augmentation and inference resizing patterns as in GeoCalib. To support radially distorted camera models (Tab. 2), we add another prediction head to ParamNet, using the parametrization of [52] for the distortion value $\mathbf{k}$. We then re-train PerspectiveNet and ParamNet on our distorted dataset.

SVA [50]: We use the official implementation released by the authors to run inference. The method fails when too few lines or coplanar repeats are visible in the image, which happens for about 58% of the samples in MegaDepth [47] and about 89% of images on Lamar [70]. Images are resized to a maximum of $3k \times 3k$.

UVP [62]: To run inference using UVP, we make the upright assumption and follow the suggested configuration using DeepLSD [61] to extract lines. We resize the short side of the image to 320 as recommended by the authors, which we also

found to work best. The approach generally does not return any value when too few lines are observable in the image.

G Qualitative Results

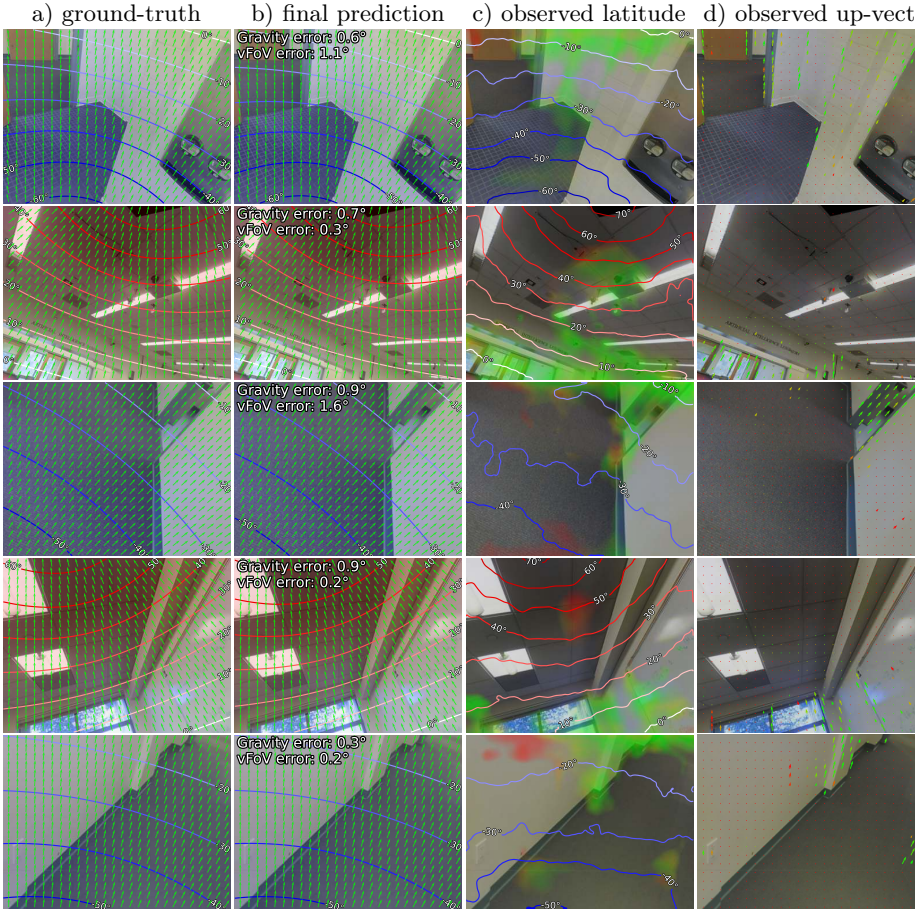


Fig. 11: Qualitative examples from the Stanford2D3D dataset [8].

a) ground-truth b) final prediction c) observed latitude d) observed up-vect.

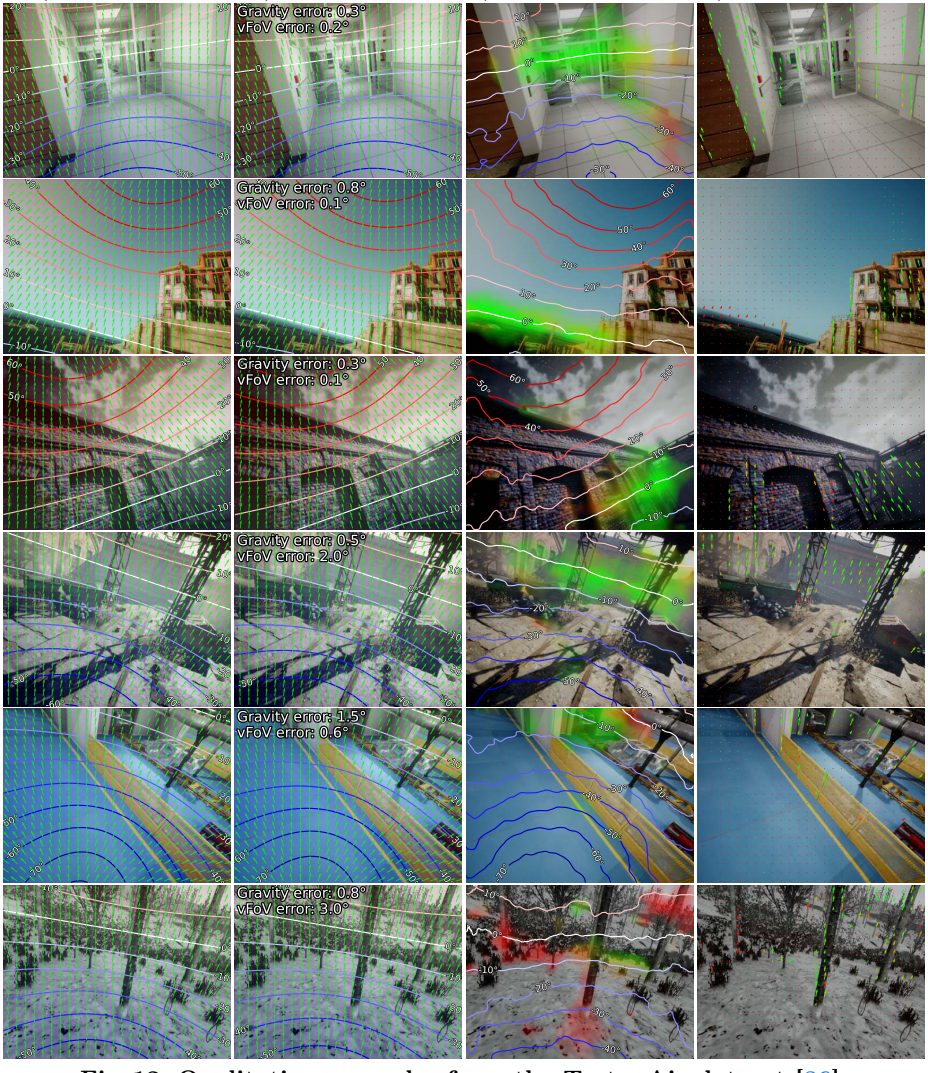


Fig. 12: Qualitative examples from the TartanAir dataset [86].

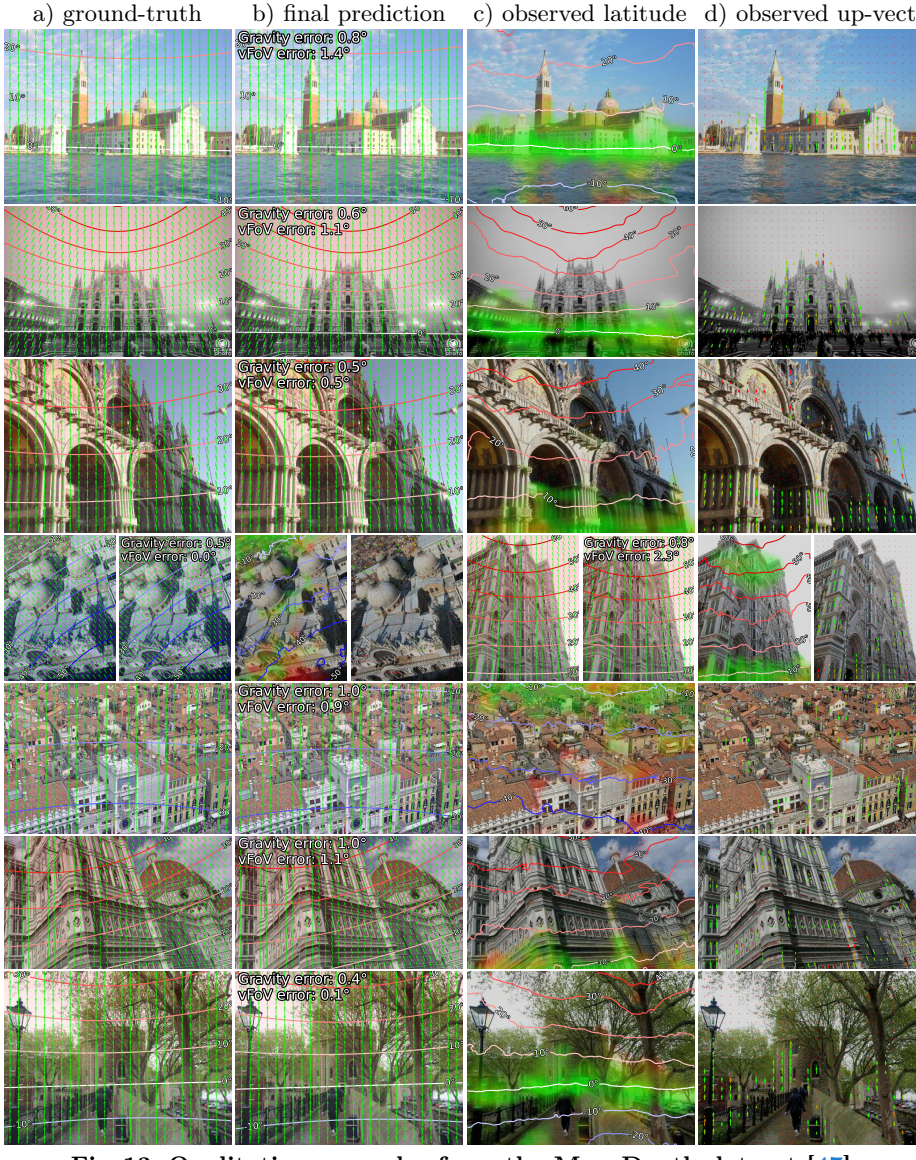


Fig. 13: Qualitative examples from the MegaDepth dataset [47].

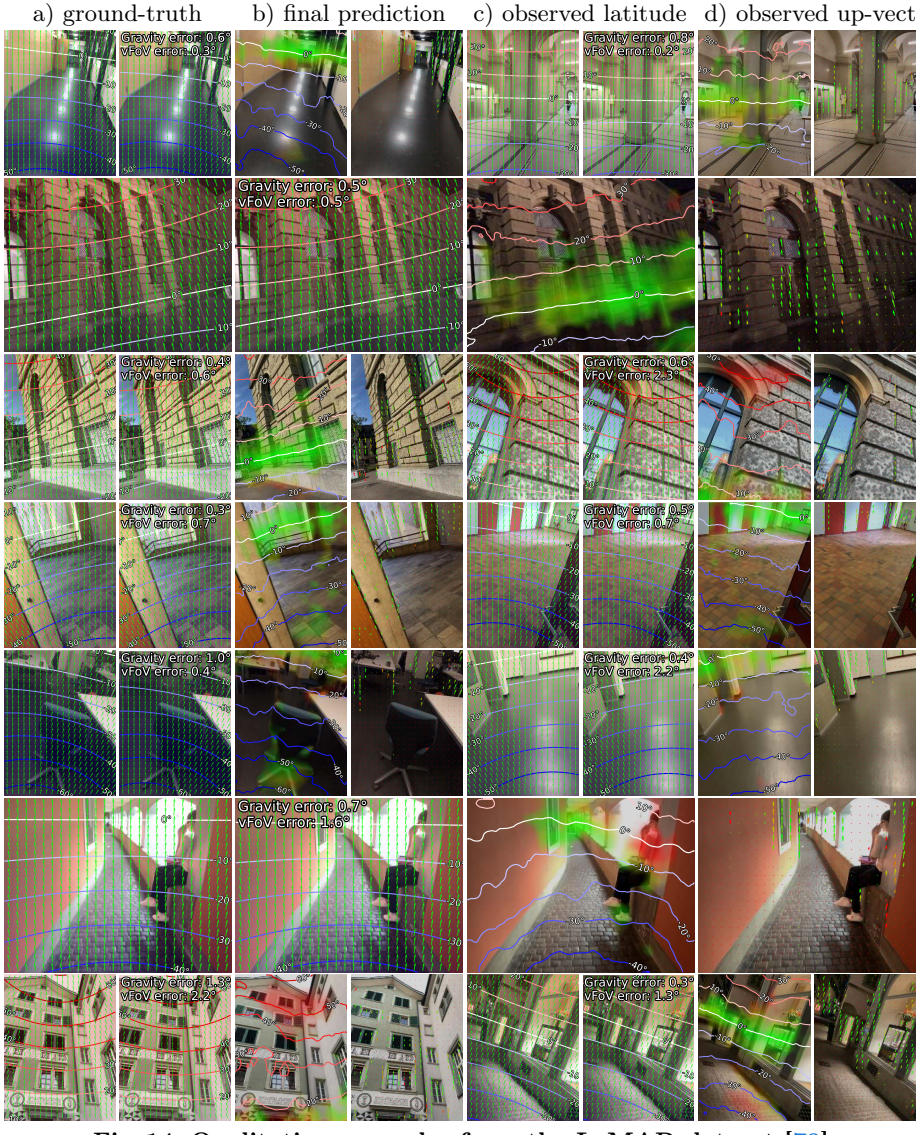


Fig. 14: Qualitative examples from the LaMAR dataset [70].

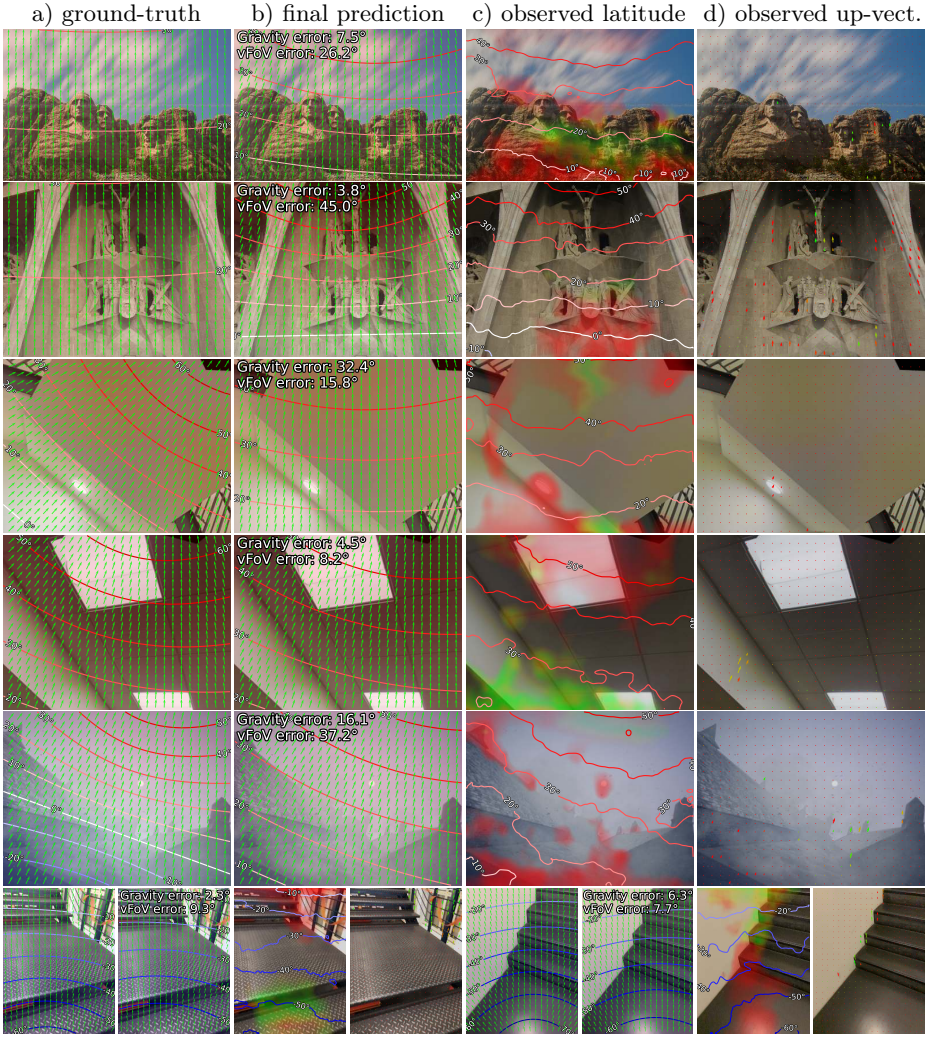


Fig. 15: Failure Cases. GeoCalib struggles on some challenging scenes, might miss important cues and is unable to leverage horizontal lines.

References

1. HDR Maps. https://hdrmaps.com/ 8, 19
2. Poly Haven. https://polyhaven.com/ 8, 19
3. Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S.M., Szeliski, R.: Building Rome in a day. Communications of the ACM **54**(10), 105–112 (2011) 2
4. Agarwal, S., Mierle, K., Others: Ceres Solver. http://ceres-solver.org 14, 21
5. Agarwal, S., Snavely, N., Seitz, S.M., Szeliski, R.: Bundle Adjustment in the Large. In: ECCV (2010) 1
6. Aguilera, D., Lahoz, J.G., Codes, J.F.: A new method for vanishing points detection in 3D reconstruction from a single view. Proceedings of ISPRS comission **2** (2005) 2, 3
7. Antunes, M., Barreto, J.P., Aouada, D., Ottersten, B.: Unsupervised Vanishing Point Detection and Camera Calibration from a Single Manhattan Image with Radial Distortion. In: CVPR (2017) 3
8. Armeni, I., Sax, S., Zamir, A.R., Savarese, S.: Joint 2D-3D-Semantic Data for Indoor Scene Understanding. arXiv:1702.01105 (2017) 9, 10, 11, 22
9. Bazin, J.C., Demonceaux, C., Vasseur, P., Kweon, I.: Rotation estimation and vanishing point extraction by omnidirectional vision in urban environment. IJRR **31**(1), 63–81 (2012) 3
10. Bazin, J.C., Pollefeys, M.: 3-line RANSAC for Orthogonal Vanishing Point Detection. In: IROS (2012) 2, 3
11. Bazin, J.C., Seo, Y., Demonceaux, C., Vasseur, P., Ikeuchi, K., Kweon, I., Pollefeys, M.: Globally Optimal Line Clustering and Vanishing Point Estimation in Manhattan World. In: CVPR (2012) 3
12. Bernard, S.T.: Interpreting Perspective Images. In: Artificial Intelligence (1983) 3
13. Bhowmik, A., Gumhold, S., Rother, C., Brachmann, E.: Reinforced Feature Points: Optimizing Feature Detection and Description for a High-Level Task. In: CVPR (2020) 4
14. Bogdan, O., Eckstein, V., Rameau, F., Bazin, J.C.: DeepCalib: A Deep Learning Approach for Automatic Intrinsic Calibration of Wide Field-of-View Cameras. In: CVMP (2018) 2
15. Bolduc, C., Giroux, J., Hébert, M., Demers, C., Lalonde, J.F.: Beyond the Pixel: a Photometrically Calibrated HDR Dataset for Luminance and Color Prediction. In: ICCV (2023) 8, 19
16. Brachmann, E., Rother, C.: Neural-Guided RANSAC: Learning Where to Sample Model Hypotheses. In: ICCV (2019) 4
17. Brachmann, E., Rother, C.: Visual Camera Re-Localization from RGB and RGB-D Images Using DSAC. IEEE TPAMI (2021) 3
18. Campbell, D., Liu, L., Gould, S.: Solving the Blind Perspective-n-Point Problem End-To-End With Robust Differentiable Geometric Optimization. In: ECCV (2020) 3
19. Chen, B., Parra, A., Cao, J., Li, N., Chin, T.J.: End-to-End Learnable Geometric Vision by Backpropagating PnP Optimization. In: CVPR (2020) 3
20. Clark, R., Bloesch, M., Czarnowski, J., Leutenegger, S., Davison, A.J.: LS-Net: Learning to Solve Nonlinear Least Squares for Monocular Stereo. In: ECCV (2018) 3
21. Coughlan, J.M., Yuille, A.L.: Manhattan World: Compass Direction from a Bayesian Inference. In: ICCV (1999) 3

22. Ddvernay, F., Faugeras, O.: Straight Lines Have to Be Straight: Automatic Calibration and Removal of Distortion from Scenes of Structured Environments. Machine Vision and Applications **13**(1), 14–24 (2008) 4
23. Dellaert, F.: Factor Graphs and GTSAM: A Hands-on Introduction. Georgia Institute of Technology, Tech. Rep **2**, 4 (2012) 8
24. Dellaert, F., Kaess, M., et al.: Factor Graphs for Robot Perception. Foundations and Trends® in Robotics **6**(1-2), 1–139 (2017) 8
25. DeTone, D., Malisiewicz, T., Rabinovich, A.: SuperPoint: Self-Supervised Interest Point Detection and Description. In: CVPR Workshop on Deep Learning for Visual SLAM (2018) 14
26. Drap, P., Lefèvre, J.: An Exact Formula for Calculating Inverse Radial Lens Distortions. Sensors **16**(6), 807 (2016) 4
27. Faugeras, O.D., Luong, Q.T., Maybank, S.J.: Camera Self-Calibration: Theory and Experiments. In: ECCV (1992) 4, 11, 12
28. Fischler, M.A., Bolles, R.C.: 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) 17
29. Fitzgibbon, A.W.: Simultaneous linear estimation of multiple view geometry and lens distortion. In: CVPR (2001) 4, 12
30. Fu, L.F.T., Fallon, M.: Batch Differentiable Pose Refinement for In-The-Wild Camera/LiDAR Extrinsic Calibration. In: CoRL (2023) 3
31. Grompone von Gioi, R., Jakubowicz, J., Morel, J.M., Randall, G.: LSD: A Fast Line Segment Detector with a False Detection Control. IEEE TPAMI **32**(4), 722–732 (2010) 3
32. Guo, M.H., Lu, C.Z., Hou, Q., Liu, Z., Cheng, M.M., Hu, S.M.: SegNeXt: Rethinking Convolutional Attention Design for Semantic Segmentation. NeurIPS (2022) 5, 13, 16
33. Hagemann, A., Knorr, M., Stiller, C.: Deep Geometry-Aware Camera Self-Calibration from Video. In: ICCV (2023) 1
34. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge university press (2003) 6
35. Hertzberg, C., Wagner, R., Frese, U., Schröder, L.: Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds. Information Fusion **14**(1), 57–77 (2013) 6
36. Hold-Geoffroy, Y., Piché-Meunier, D., Sunkavalli, K., Bazin, J.C., Rameau, F., Lalonde, J.F.: A Deep Perceptual Measure for Lens and Camera Calibration. IEEE TPAMI (2022) 3, 8, 10, 11, 12, 21
37. CVPR 2021 Image Matching Challenge. https://www.cs.ubc.ca/research/image-matching-challenge/ (2021) 11
38. Jin, L., Zhang, J., Hold-Geoffroy, Y., Wang, O., Matzen, K., Sticha, M., Fouhey, D.F.: Perspective Fields for Single Image Camera Calibration. In: CVPR (2023) 2, 3, 5, 8, 10, 11, 12, 15, 16, 21
39. Kannala, J., Brandt, S.S.: A Generic Camera Model and Calibration Method for Conventional, Wide-Angle, and Fish-Eye Lenses. IEEE TPAMI **28**(8), 1335–1340 (2006) 4
40. Keivan, N., Sibley, G.: Online SLAM with Any-time Self-calibration and Automatic Change Detection. In: ICRA (2015) 1
41. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. arXiv:1412.6980 (2014) 8, 21

42. Kluger, F., Brachmann, E., Ackermann, H., Rother, C., Yang, M.Y., Rosenhahn, B.: CONSAC: Robust Multi-Model Fitting by Conditional Sample Consensus. In: CVPR (2020) 3
43. Kukelova, Z., Bujnak, M., Pajdla, T.: Closed-Form Solutions to Minimal Absolute Pose Problems with Known Vertical Direction. In: ACCV (2010) 14, 20
44. Larsson, V.: PoseLib - Minimal Solvers for Camera Pose Estimation (2020), https://github.com/vlarsson/PoseLib 14, 20
45. Lee, J., Go, H., Lee, H., Cho, S., Sung, M., Kim, J.: CTRL-C: Camera calibration TRansformer with Line-Classification. In: ICCV (2021) 2, 3, 10, 11, 21
46. Levenberg, K.: A method for the solution of certain non-linear problems in least squares. Quarterly of applied mathematics **2**(2), 164–168 (1944) 6
47. Li, Z., Snavely, N.: MegaDepth: Learning Single-View Depth Prediction from Internet Photos. In: CVPR (2018) 9, 10, 11, 15, 17, 21, 24
48. Liu, S., Zhou, Y., Zhao, Y.: VaPiD: A Rapid Vanishing Point Detector via Learned Optimizers. In: ICCV (2021) 3
49. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A ConvNet for the 2020s. CVPR (2022) 21
50. Lochman, Y., Dobosevych, O., Hryniv, R., Pritts, J.: Minimal Solvers for Single-View Lens-Distorted Camera Auto-Calibration. In: WACV (2021) 2, 3, 10, 11, 12, 21
51. Lochman, Y., Liepieshov, K., Chen, J., Perdoch, M., Zach, C., Pritts, J.: BabelCalib: A Universal Approach to Calibrating Central Cameras. In: ICCV (2021) 1
52. Lopez, M., Mari, R., Gargallo, P., Kuang, Y., Gonzalez-Jimenez, J., Haro, G.: Deep Single Image Camera Calibration with Radial Distortion. In: CVPR (2019) 2, 3, 8, 10, 11, 12, 21
53. Loshchilov, I., Hutter, F.: Decoupled Weight Decay Regularization. arXiv preprint arXiv:1711.05101 (2017) 20
54. Lv, Z., Dellaert, F., Rehg, J.M., Geiger, A.: Taking a Deeper Look at the Inverse Compositional Algorithm. In: CVPR (2019) 3
55. Madsen, K., Nielsen, H.B., Tingleff, O.: Methods for Non-Linear Least Squares Problems (2004) 6
56. Magee, M.J., Aggarwal, J.K.: Determining Vanishing Points from Perspective Images. Computer Vision, Graphics, and Image Processing **26**(2), 256–267 (1984) 3
57. Mapillary: OpenSfM. https://opensfm.org 1
58. Marquardt, D.W.: An Algorithm for Least-Squares Estimation of Nonlinear Parameters. Journal of the society for Industrial and Applied Mathematics **11**(2), 431–441 (1963) 6
59. Maye, J., Furgale, P., Siegwart, R.: Self-Supervised Calibration for Robotic Systems. In: IEEE Intelligent Vehicles Symposium (IV) (2013) 1
60. Moulon, P., Monasse, P., Perrot, R., Marlet, R.: OpenMVG: Open Multiple View Geometry. In: International Workshop on Reproducible Research in Pattern Recognition. pp. 60–74. Springer (2016) 1
61. Pautrat, R., Barath, D., Larsson, V., Oswald, M.R., Pollefeys, M.: DeepLSD: Line Segment Detection and Refinement with Deep Image Gradients. In: CVPR (2023) 21
62. Pautrat, R., Liu, S., Hruby, P., Pollefeys, M., Barath, D.: Vanishing Point Estimation in Uncalibrated Images with Prior Gravity Direction. In: ICCV (2023) 2, 3, 10, 11, 14, 21
63. Pineda, L., Fan, T., Monge, M., Venkataraman, S., Sodhi, P., Chen, R.T., Ortiz, J., DeTone, D., Wang, A., Anderson, S., Dong, J., Amos, B., Mukadam, M.: Theseus: A Library for Differentiable Nonlinear Optimization. NeurIPS (2022) 3

64. Pollefeys, M., Koch, R., Gool, L.V.: Self-Calibration and Metric Reconstruction Inspite of Varying and Unknown Intrinsic Camera Parameters. IJCV **32**(1), 7–25 (1999) 4
65. Pritts, J., Kukelova, Z., Larsson, V., Lochman, Y., Chum, O.: Minimal Solvers for Rectifying from Radially-Distorted Conjugate Translations. IEEE TPAMI (2020) 3
66. Qian, Y., Elder, J.H.: A Reliable Online Method for Joint Estimation of Focal Length and Camera Rotation. In: ECCV (2022) 3
67. Russell, C., Toso, M., Campbell, N.: Fixing Implicit Derivatives: Trust-Region Based Learning of Continuous Energy Functions. In: NeurIPS (2019) 4, 7
68. Sarlin, P.E., Cadena, C., Siegwart, R., Dymczyk, M.: From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In: CVPR (2019) 14
69. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: SuperGlue: Learning Feature Matching with Graph Neural Networks. In: CVPR (2020) 3, 14
70. Sarlin, P.E., Dusmanu, M., Schönberger, J.L., Speciale, P., Gruber, L., Larsson, V., Miksik, O., Pollefeys, M.: LaMAR: Benchmarking Localization and Mapping for Augmented Reality. In: ECCV (2022) 9, 10, 11, 15, 21, 25
71. Sarlin, P.E., Unagar, A., Larsson, M., Germain, H., Toft, C., Larsson, V., Pollefeys, M., Lepetit, V., Hammarstrand, L., Kahl, F., Sattler, T.: Back to the Feature: Learning Robust Camera Localization from Pixels to Pose. In: CVPR (2021) 3, 7
72. Scaramuzza, D., Martinelli, A., Siegwart, R.: A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion. In: IEEE International Conference on Computer Vision Systems (2006) 4
73. Scaramuzza, D., Martinelli, A., Siegwart, R.: A Toolbox for Easily Calibrating Omnidirectional Cameras. In: IROS (2006) 1
74. Schönberger, J.L., Frahm, J.M.: Structure-from-Motion Revisited. In: CVPR (2016) 1, 11, 12, 21
75. Snavely, N., Seitz, S.M., Szeliski, R.: Photo Tourism: Exploring Photo Collections in 3D. ACM TOG **25**(3), 835–846 (2006) 2
76. Snavely, N., Seitz, S.M., Szeliski, R.: Modeling the World from Internet Photo Collections. IJCV (2008) 2
77. Song, X., Kang, H., Moteki, A., Suzuki, G., Kobayashi, Y., Tan, Z.: MSCC: Multi-Scale Transformers for Camera Calibration. In: WACV (2024) 2, 3, 10, 11, 21
78. Taira, H., Okutomi, M., Sattler, T., Cimpoi, M., Pollefeys, M., Sivic, J., Pajdla, T., Torii, A.: InLoc: Indoor Visual Localization with Dense Matching and View Synthesis. IEEE TPAMI (2019) 14
79. Tang, C., Tan, P.: BA-Net: Dense Bundle Adjustment Network. In: ICLR (2019) 3
80. Teed, Z., Deng, J.: DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. In: NeurIPS (2021) 3, 7
81. Tong, X., Ying, X., Shi, Y., Wang, R., Yang, J.: Transformer Based Line Segment Classifier with Image Context for Real-Time Vanishing Point Detection in Manhattan World. In: CVPR (2022) 3
82. Tretyak, E., Barinova, O., Kohli, P., Lempitsky, V.: Geometric Image Parsing in Man-Made Environments. IJCV (2012) 3
83. Urban, S., Leitloff, J., Hinz, S.: Improved Wide-Angle, Fisheye and Omnidirectional Camera Calibration. ISPRS Journal of Photogrammetry and Remote Sensing **108**, 72–79 (2015) 4
84. Usenko, V., Demmel, N., Cremers, D.: The Double Sphere Camera Model. In: 3DV (2018) 4, 12
85. Wang, C., Galoogahi, H.K., Lin, C.H., Lucey, S.: Deep-LK for Efficient Adaptive Object Tracking. In: ICRA (2018) 3

86. Wang, W., Zhu, D., Wang, X., Hu, Y., Qiu, Y., Wang, C., Hu, Y., Kapoor, A., Scherer, S.: TartanAir: A Dataset to Push the Limits of Visual SLAM. In: IROS (2020) 9, 10, 11, 14, 23
87. Wei, T., Patel, Y., Shekhovtsov, A., Matas, J., Barath, D.: Generalized Differentiable RANSAC. In: CVPR (2023) 4
88. Wildenauer, H., Hanbury, A.: Robust Camera Self-Calibration from Monocular Images of Manhattan Worlds. In: CVPR (2012) 2, 3
89. Wilson, K., Snavely, N.: Robust Global Translations with 1DSfM. In: ECCV (2014) 2
90. Xian, W., Li, Z., Fisher, M., Eisenmann, J., Shechtman, E., Snavely, N.: UprightNet: Geometry-Aware Camera Orientation Estimation from Single Images. In: ICCV (2019) 3, 4
91. Xiao, J., Ehinger, K.A., Oliva, A., Torralba, A.: Recognizing Scene Viewpoint using Panoramic Place Representation. In: CVPR. pp. 2695–2702. IEEE (2012) 21
92. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. In: NeurIPS (2021) 13, 16
93. Xu, B., Davison, A.J., Leutenegger, S.: Deep Probabilistic Feature-Metric Tracking. RA-L **6**(1), 223–230 (2021) 3
94. Zhai, M., Workman, S., Jacobs, N.: Detecting Vanishing Points using Global Image Context in a Non-Manhattan World. In: CVPR (2016) 3
95. Zhou, Y., Qi, H., Huang, J., Ma, Y.: NeurVPS: Neural Vanishing Point Scanning via Conic Convolution. NeurIPS (2019) 3
96. Zhuang, B., Tran, Q.H., Lee, G.H., Cheong, L.F., Chandraker, M.: Degeneracy in Self-Calibration Revisited and a Deep Learning Solution for Uncalibrated SLAM. In: IROS (2019) 1