AnyCalib:
On-Manifold Learning for Model-Agnostic Single-View Camera Calibration

Supplementary Material

A. Ablations

We report ablation results in Tab. 5. Experiments 1-4. are conducted by training AnyCalib on OP_p and averaging errors across the benchmarks of Sec. 4.1. The fifth, RANSAC, ablation, is performed on ScanNet++, following Sec. 4.2. MACs are computed for a 280×364 input image, which results from resizing an image with a 3:4 (H:W) aspect ratio to the training resolution of 320^2 pixels.

1-2. Intermediate representation. We test the performance of AnyCalib when learning rays instead of our proposed FoV fields (Sec. 3.1). As first baseline, we use the target representation (rays) and loss function of WildCam [100], which is a cosine similarity loss. As a stronger baseline, we evaluate also the training strategy of DSINE [8] for learning rays *i.e.* using an angular loss. Compared to these baselines, FoV fields lead to more accurate calibrations.

3. Decoder architecture. Our proposed light DPT decoder, when compared to the original [67], decreases $\sim$20% the computation and leads to slight accuracy improvements.

4. Dataset extension. Our extended version of OpenPano [86] leads to improved accuracy. This experiment shows that AnyCalib is scalable.

5. RANSAC [25] can also be applied to our derivations in Sec. 3.2 by using minimal samples from the set of 2D-3D correspondences between the regressed rays and image points. However, minimal samples lead to inaccurate intrinsics when fitting high-complexity camera models such as Kannala-Brandt (KB) [39]. This motivates our non-minimal estimation of the intrinsics.

In conclusion, FoV fields are an appropriate intermediate representation for calibration and key to the performance of AnyCalib: their supervision, when compared to alternatives, leads to learning patterns that are more useful. Moreover, since FoV fields are not tied to extrinsic cues, this is what has allowed us to extend the training dataset with panoramas not aligned with the gravity direction.

B. Datasets details

As mentioned in Sec. 3.3, we create four datasets. We separately train AnyCalib in each of them to study its accuracy according to the trained projection models. The intrinsics used to create the datasets are detailed in Tab. 6. For the camera rotations[7] we follow GeoCalib [86] and uniformly sample the roll and pitch angles within $\pm 45°$. All datasets are formed by sampling 16 square images in each of the 3651/202/202 training/val/test panoramas, which yields an approximate distribution of 54k/3k/3k training/val/test images per dataset.

[7] For panoramas not aligned with the gravity direction, these rotations are only approximate.

Experiment	RE	{v, h}FoV		MACs
AnyCalib	23.81	2.89	3.02	187.8G
1. Learning rays as [100]	26.05	3.13	3.24	187.8G
2. Learning rays as [8]	24.95	3.02	3.13	187.8G
3. Original DPT [67]	24.99	3.00	3.12	243.2G
4. Orig. OpenPano [86]	26.62	3.23	3.35	187.8G
AnyCalib	20.61	3.90	3.38	n/a
5. RANSAC in KB [39]	1019	16.1	16.2	n/a

Table 5. **Ablation study** over representation, decoder design, dataset and intrinsics fitting method. See Supp. A for details.

Obtaining the focal length. As shown in Tab. 6, we do not directly sample the focal length f. Instead, to ensure a uniform distribution of image FoVs, we indirectly sample it from the rest of the parameters. For pinhole images, we use the well-known conversion $f = (H/2)/\tan(\text{FoV}/2)$. For BC [14] and EUCM [41], we note that, from Eq. (8):

$$f = \frac{H/2}{R\,\phi(R,\,Z)}\,, \qquad \begin{matrix} R = \sin(\text{FoV}/2)\,, \\ Z = \cos(\text{FoV}/2)\,, \end{matrix} \tag{13}$$

since we form the datasets with square images ($H = W$), unit aspect ratio and centered principal point. During training, images are geometrically transformed on-the-fly to match the training resolution and sampled aspect-ratio.

Ensuring valid intrinsics. Independently sampling intrinsics of BC [14] and EUCM [41] can lead to projection models that project different, distant, rays to the same image coordinates [47], which is not physically valid. We guard for this by clamping f according to its limits [47, 85]:

$$\text{BC} \rightarrow \; f \geq \begin{cases} 0 & \text{if } k \geq 0\,, \\ \frac{r_\text{im}}{r_\text{max}(1+k r_\text{max}^2)} & \text{if } k < 0\,, \end{cases} \tag{14}$$

$$\text{EUCM} \rightarrow \; f \geq \begin{cases} 0 & \text{if } \alpha \leq 0.5\,, \\ r_\text{im}\sqrt{\beta(2\alpha - 1)} & \text{if } \alpha > 0.5\,, \end{cases} \tag{15}$$

where $r_\text{im} = 0.5(H^2 + W^2)^{0.5}$ and $r_\text{max} = 1/\sqrt{-3k}$.

Mapping LensFun coefficients. As explained in Sec. 3.3 and Fig. 4, we use the LensFun database [4] for defining the sampling bounds of EUCM's α and β. LensFun uses

Data	Models	FoV [°]	$\hat{k} = kH/f$	α	β
OP_p	100% pinhole	$\mathcal{U}(20, 105)$	-	-	-
OP_r	100% BC [14]	$\mathcal{U}(20, 105)$	$\mathcal{N}_t(0,\ 0.07,\ [-0.3,\ 0.3])$	-	-
OP_d	50% BC [14]	$\mathcal{U}(20, 105)$	$\mathcal{N}_t(0,\ 0.07,\ [-0.3,\ 0.3])$	-	-
	50% EUCM [41]	$\mathcal{U}(50, 180)$	-	$\mathcal{U}(0.5, 0.8)$	$\mathcal{U}(0.5, 2)$
OP_g	34% pinhole	$\mathcal{U}(20, 105)$	-	-	-
	33% BC [14]	$\mathcal{U}(20, 105)$	$\mathcal{N}_t(0,\ 0.07,\ [-0.3,\ 0.3])$	-	-
	33% EUCM [41]	$\mathcal{U}(50, 180)$	-	$\mathcal{U}(0.5, 0.8)$	$\mathcal{U}(0.5, 2)$

Table 6. **Sampling distributions within the datasets.** $\mathcal{U}(a,\ b)$ denotes a uniform distribution $\in [a,\ b]$. $\mathcal{N}_t(\mu,\ \sigma,\ [a,\ b])$ denotes a normal distribution $\mathcal{N}(\mu,\ \sigma)$ truncated at $[a,\ b]$. $\textbf{OP}_\text{p}$ and $\textbf{OP}_\text{r}$ follow the setup of GeoCalib [86]. Since the distortions allowed by the Brown-Conrady (radial) model [14] are limited [47, 85], for creating $\textbf{OP}_\text{d}$ and $\textbf{OP}_\text{g}$, we use EUCM [41] to generate strongly distorted images. The limits for sampling its parameters α and β are based on real-lens values from the public LensFun database [4] (Fig. 4).

Figure 5. **Sample** images and intrinsics from the dataset OP_g.

its own polynomial distortion models[8], so we need to map them. Conveniently, our formulation in Sec. 3.2 is applicable: given normalized image coordinates (obtained with the lens focal and image sensor size) and their unprojected rays, we can linearly recover α and β. For getting these unprojections, we first undistort a uniform grid of image/sensor coordinates using Newton's root finding algorithm and finally invert the ideal (equisolid, equidistant, orthographic or stereographic [4]) fisheye projection model of the lens.

Sample datapoints of OP_g are shown in Fig. 5.

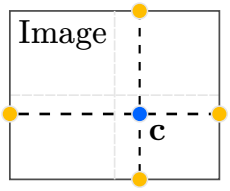
[8]Explained in https://lensfun.github.io/manual/v0.3.2/group__Lens.html#gaa505e04666a189274ba66316697e308e

C. Model-agnostic evaluation

Intrinsics in different models. Different camera models, can have an order of magnitude difference in their focal length f values [85, Tab. 3]. We visualize this behavior in Fig. 6 by mapping the ground-truth Kannala-Brandt (KB) [39] intrinsics from ScanNet++ [95] to UCM intrinsics using our formulation from Sec. 3.2. As shown, if we fix the ground-truth KB focal length and only map the distortion coefficients, the resulting UCM intrinsics fail to accurately model the camera lens projection, leading to an undistortion failure. The converse occurs when we also map the focal length.

Figure 6. **The focal length (f) in different camera models** can take significantly different values. We show this for UCM [55]. We map the KB [39] intrinsics corresponding to an image from ScanNet++ [95] (left) to UCM following Sec. 3.2. We do this without fixing f, *i.e.*, also mapping it to UCM (middle) and fixing it (right). The resulting intrinsics are used to undistort the image. The same KB focal for UCM leads to a model that does not truthfully model the lens, leading to a failed undistortion. The converse occurs when also mapping f.

Model-agnostic FoV. The horizontal (hFoV) and vertical (vFoV) angular extents of an image can be computed independently of the camera model. To compute the hFoV, we unproject the rays located at the left and right borders, based on the location of the principal point, $\mathbf{c}$ (yellow points on the schematic on the right), and sum the angles between them and the optical axis. The vFoV is computed similarly, but using the top and bottom borders instead.



D. Linear constraints

Lochman et al. [52] show that the distortion parameters of a wide range of camera models can be estimated linearly from 1D-1D correspondences between the radii on the retinal plane, $\|(\mathbf{x}-\mathbf{c})/f\|$, and the ray radii, $\sqrt{X^2+Y^2}$. Building on this, we show in this section that, together with Eq. (11), *all the intrinsics* of a wide range of standard camera models can be linearly recovered from 2D-3D correspondences between image coordinates $\mathbf{x} \in \Omega$ and ray directions in $\mathcal{S}^2$.

To obtain the remaining linear constraints, presented in Tab. 1, we first define auxiliary variables according to the notation in Sec. 3.2:

$$R_a \coloneqq \sqrt{X^2 + a^2 Y^2}\,, \quad r_c \coloneqq \|\mathbf{x} - \mathbf{c}\|\,, \tag{16}$$

$$\theta \coloneqq \operatorname{atan2}(R, Z)\,, \quad r \coloneqq \sqrt{m_x^2 + m_y^2}\,, \tag{17}$$

$$d \coloneqq \sqrt{R^2 + Z^2}\,, \quad r_{ca}^2 \coloneqq (u - c_x)^2 + (v - c_y)^2/a^2\,. \tag{18}$$

For forward camera models (Eq. (8)), the linear constraints derive from Eq. (21):

$$\pi(\mathbf{p}) = \mathbf{x} = f\ \phi(R, Z) \begin{bmatrix} X \\ aY \end{bmatrix} + \mathbf{c}\,, \tag{19}$$

$$\implies \|\mathbf{x} - \mathbf{c}\| = f\ \phi(R, Z) \left\| \begin{bmatrix} X \\ aY \end{bmatrix} \right\|\,, \tag{20}$$

$$\implies r_c = f\ \phi(R, Z)\ R_a\,. \tag{21}$$

by substituting the corresponding model-specific function $\phi(R, Z)$ from Tab. 1.

Pinhole: $r_c = \frac{f}{Z} R_a \implies R_a f = Z r_c\,.$

Brown-Conrady:

$$r_c = f \frac{R_a}{Z} \left(1 + \sum_{n=1}^{N} k_n (R/Z)^{2n}\right)\,, \tag{22}$$

$$\implies r_c Z/f - R_a \sum_{n=1}^{N} k_n (R/Z)^{2n} = R_a\,. \tag{23}$$

Kannala-Brandt:

$$r_c = f \frac{R_a}{R} (\theta + \sum_{n=1}^{N} k_n \theta^{2n+1})\,, \tag{24}$$

$$\implies R r_c/f - R_a \sum_{n=1}^{N} k_n \theta^{2n+1} = R_a \theta\,. \tag{25}$$

UCM:

$$r_c = f \frac{R_a}{\xi d + Z} \implies R_a f - r_c d \xi = r_c Z\,. \tag{26}$$

EUCM For this camera model, linearity in Eq. (21) is lost when f is unknown. Instead, to estimate f, we use a proxy

camera model [39] that leads to practically the same focal length value [52, 85]. Thus, instead of Eq. (10) we start from $r = \phi(R, Z)R$, which for the EUCM model, leads to:

$$r = \frac{R}{\alpha\sqrt{\beta R^2 + Z^2} + (1-\alpha)Z} , \tag{27}$$

$$\implies r\alpha\sqrt{\beta R^2 + Z^2} = R - (1-\alpha)Zr , \tag{28}$$

$$\implies r^2\alpha^2(\beta R^2 + Z^2) = (R - (1-\alpha)Zr)^2 , \tag{29}$$

$$\implies r^2R^2\alpha^2\beta + 2rZ(rZ - R)\alpha = (R - rZ)^2 . \tag{30}$$

Division For this backward model, from Eq. (9) we know that

$$f \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \lambda \begin{bmatrix} (u - c_x) \\ (v - c_y)/a \\ f\psi(r) \end{bmatrix} . \tag{31}$$

To remove the nonlinearity stemming from λ, we use an approach similar to DLT [33]. Since both sides must be parallel, their cross product is the null vector, which leads to the following constraints:

$$(f + \sum_{n=1}^{N} k'_n r_{ca}^{2n}) \begin{bmatrix} X \\ aY \end{bmatrix} = Z(\mathbf{x} - \mathbf{c}) , \tag{32}$$

with $k'_n := k_n / f^{2n-1}$. As inferred from Eq. (31), these two equations are linearly dependent. Thus, we consider only the norm of both sides, which results in:

$$R_a(f + \sum_{n=1}^{N} k'_n r_{ca}^{2n}) = Z r_c . \tag{33}$$

E. Additional qualitative results

We show qualitative results, using AnyCalib$_{\text{gen}}$ (trained on OP$_{\text{g}}$) on perspective images in Figs. 7 to 10 and on distorted images in Figs. 11 and 12. We also show undistortion results using the same model in Fig. 14. Additional qualitative results on edited (stretched and cropped) images are shown in Fig. 13, with AnyCalib being trained following [34, 100] (Sec. 3.3).

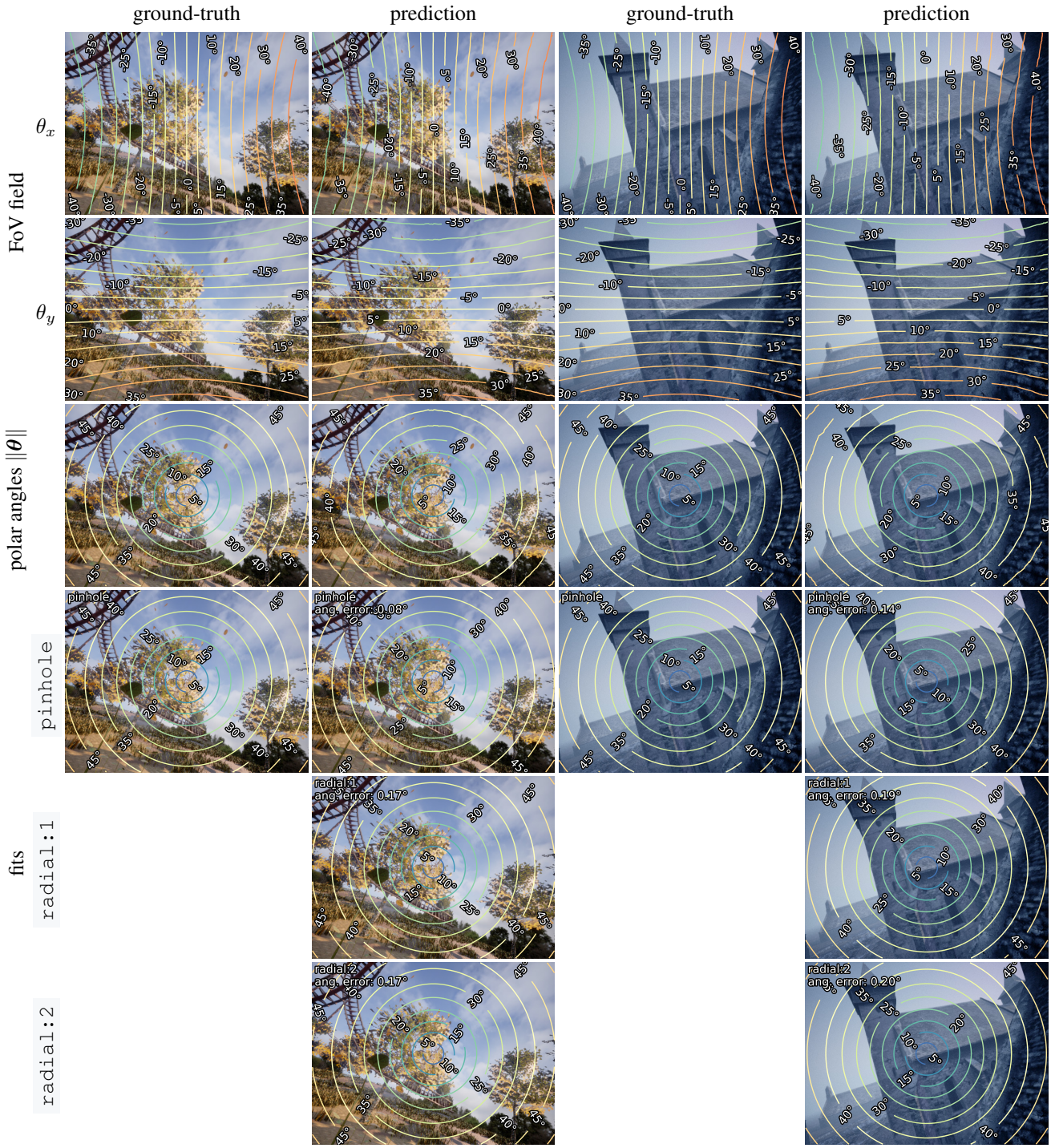


Figure 7. **Qualitative results on perspective images** in TartanAir [91] with AnyCalib$_{\text{gen}}$—trained on OP$_{\text{g}}$. The FoV field (θ_x and θ_y) is regressed by the network and $\|\boldsymbol{\theta}\|$ represents both its norm and the polar angle of the ray corresponding to each pixel. The predicted FoV field is used to fit the camera model of choice. `radial:i` corresponds to the Brown-Conrady model with `i` distortion coefficients.

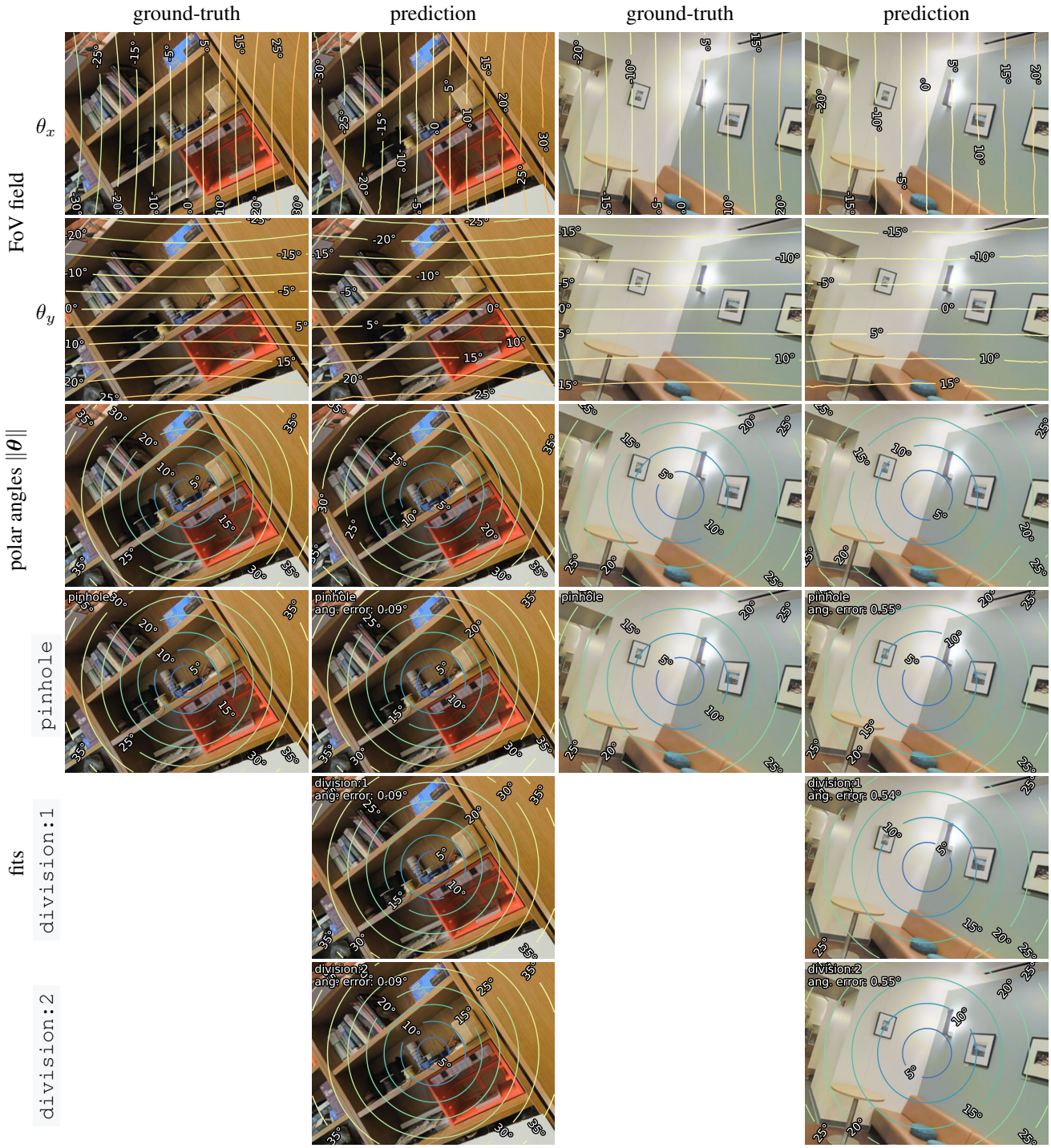


Figure 8. **Qualitative results on perspective images** in Stanford2D3D [7] with AnyCalib$_{\text{gen}}$—trained on OP$_{\text{g}}$. The FoV field (θ_x and θ_y) is regressed by the network and $\|\boldsymbol{\theta}\|$ represents both its norm and the polar angle of the ray corresponding to each pixel. The predicted FoV field is used to fit the camera model of choice. `division:i` corresponds to the division model with `i` distortion coefficients.

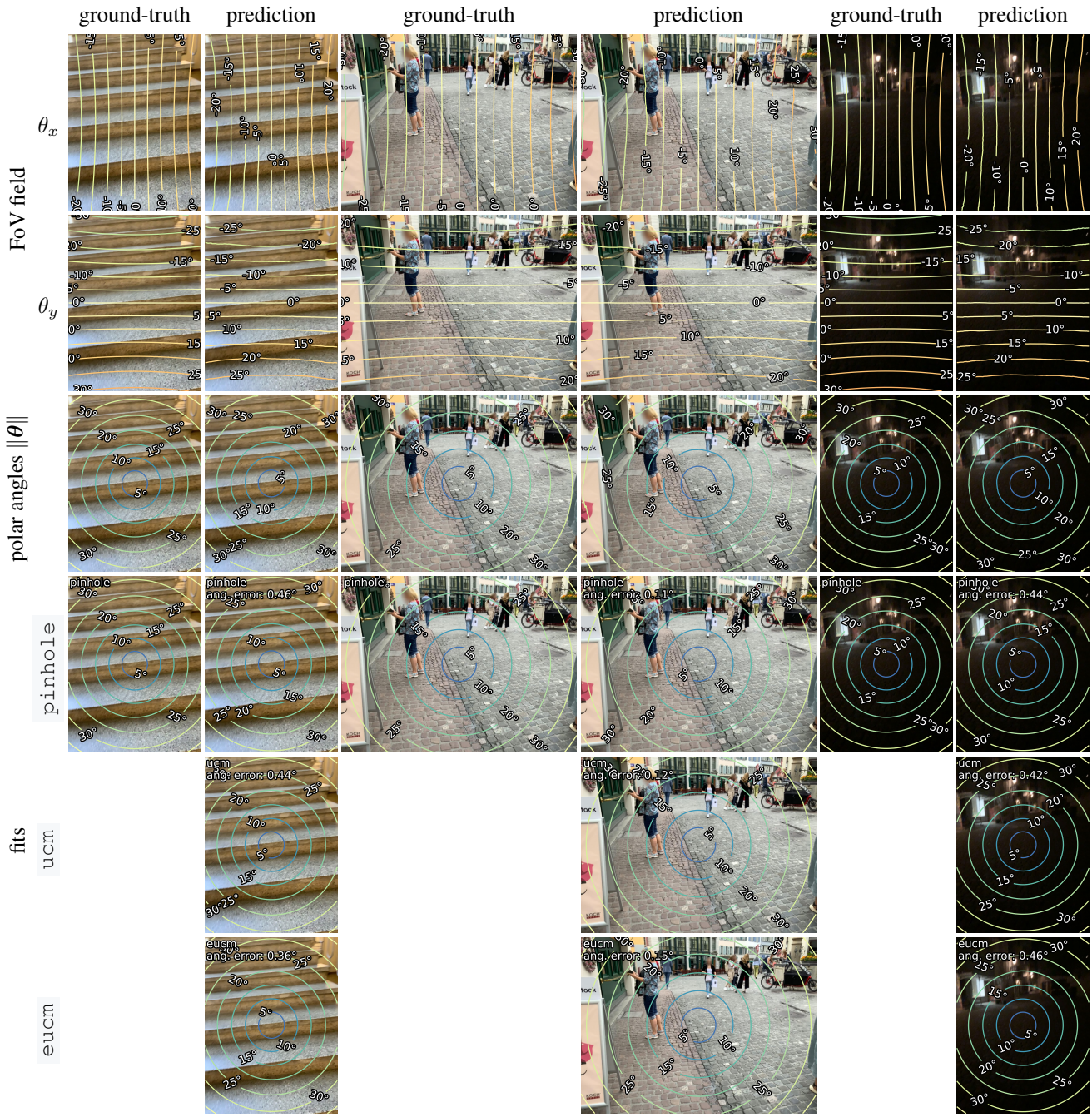


Figure 9. **Qualitative results on perspective images** in LaMAR [69] with AnyCalib$_{\text{gen}}$—trained on OP$_{\text{g}}$. The FoV field (θ_x and θ_y) is regressed by the network and $\|\boldsymbol{\theta}\|$ represents both its norm and the polar angle of the ray corresponding to each pixel. The predicted FoV field is used to fit the camera model of choice.

Figure 10. **Qualitative results on perspective images** in MegaDepth [49] with AnyCalib$_{\text{gen}}$—trained on OP$_{\text{g}}$. The FoV field (θ_x and θ_y) is regressed by the network and $\|\boldsymbol{\theta}\|$ represents both its norm and the polar angle of the ray corresponding to each pixel. The predicted FoV field is used to fit the camera model of choice. `kb:i` corresponds to the Kannala-Brandt model with `i` distortion coefficients.

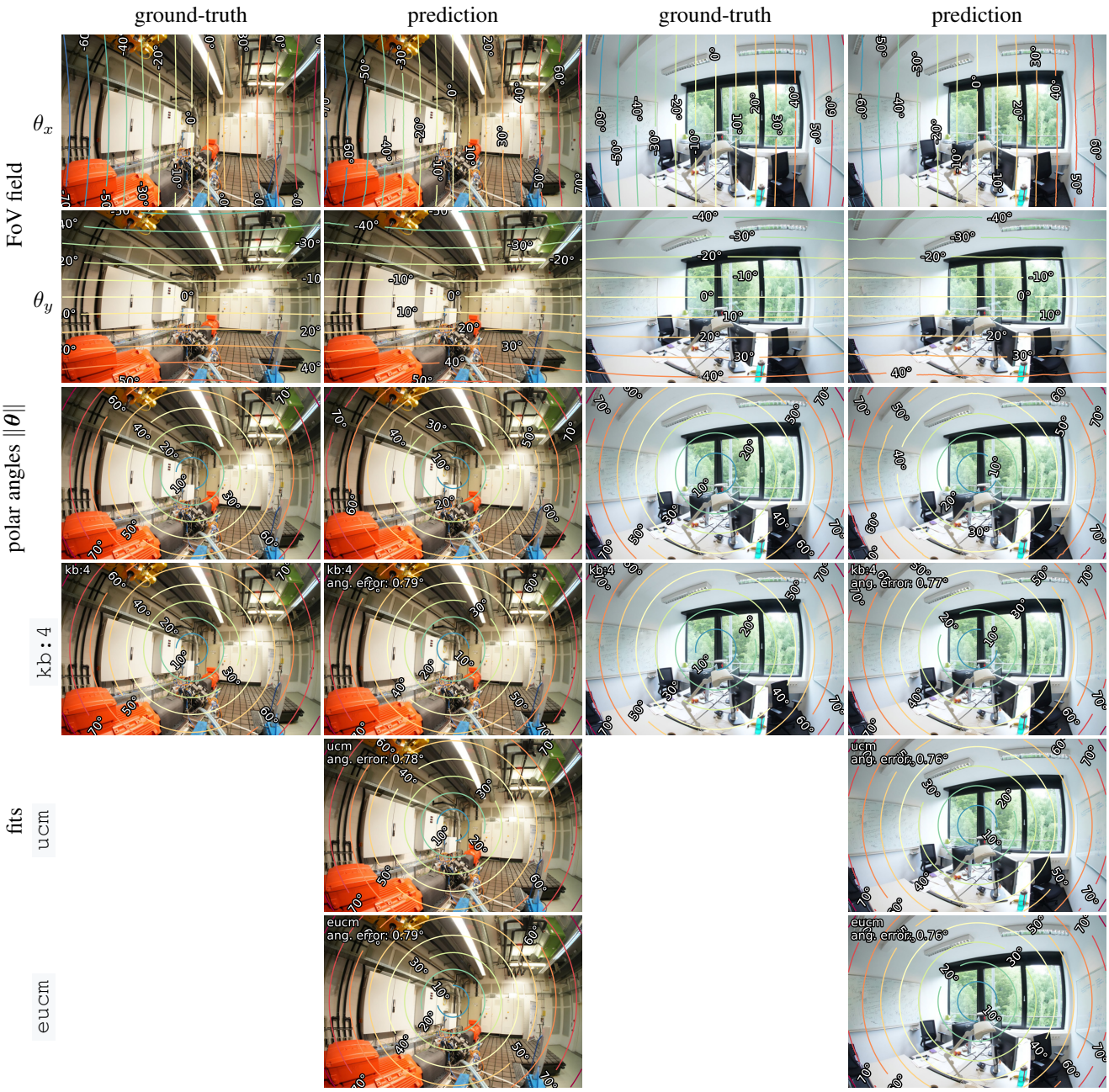


Figure 11. **Qualitative results on distorted images** in ScanNet++ [95] with AnyCalib$_{\text{gen}}$—trained on OP$_{\text{g}}$. The FoV field (θ_x and θ_y) is regressed by the network and $\|\boldsymbol{\theta}\|$ represents both its norm and the polar angle of the ray corresponding to each pixel. The predicted FoV field is used to fit the camera model of choice. `kb:i` corresponds to the Kannala-Brandt model with `i` distortion coefficients.

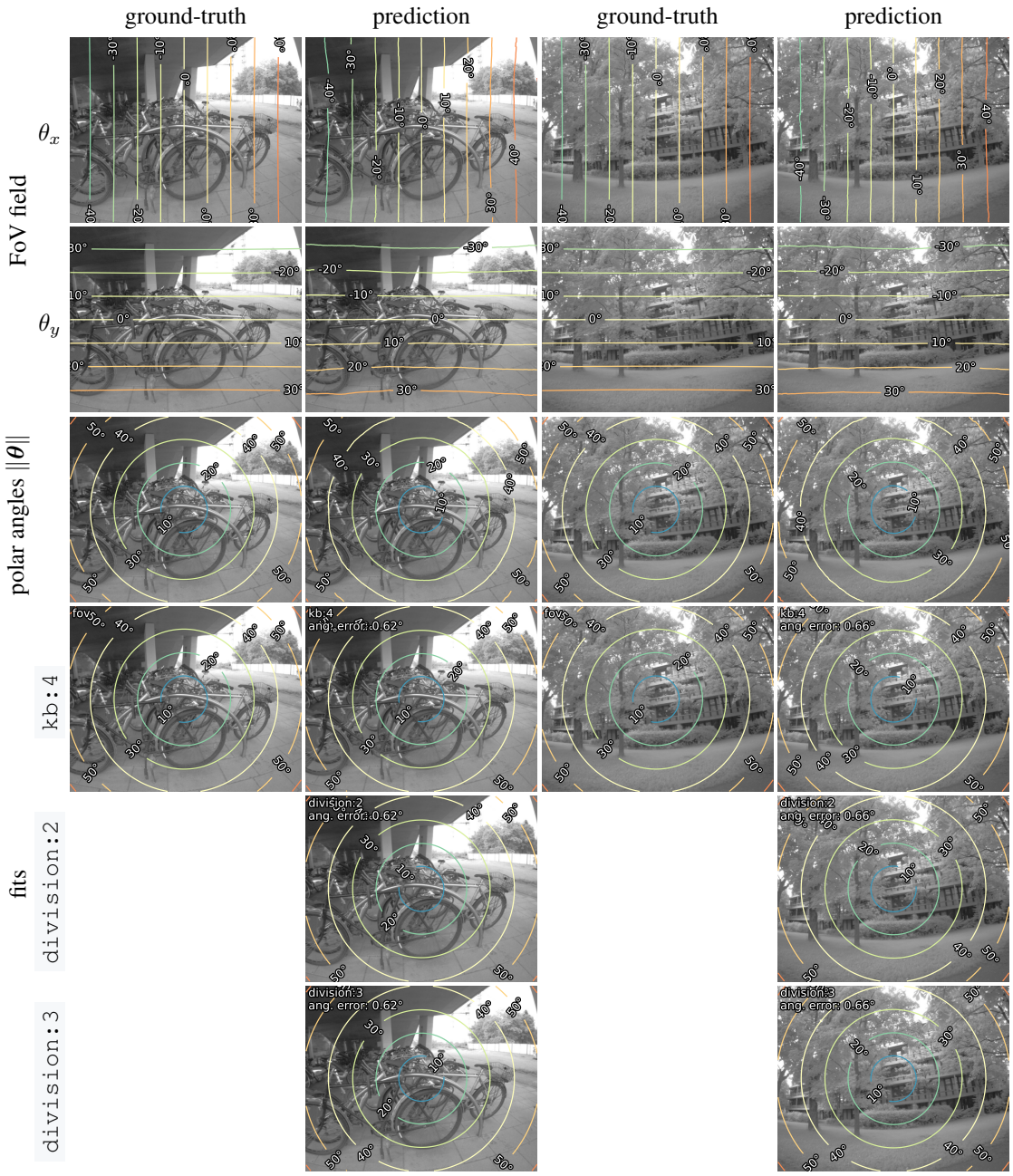


Figure 12. **Qualitative results on distorted images** in the Mono Dataset [23] with AnyCalib$_{\text{gen}}$—trained on OP$_{\text{g}}$. The FoV field (θ_x and θ_y) is regressed by the network and $\|\boldsymbol{\theta}\|$ represents both its norm and the polar angle of the ray corresponding to each pixel. The predicted FoV field is used to fit the camera model of choice. `division:i` corresponds to the division model with `i` distortion coefficients.

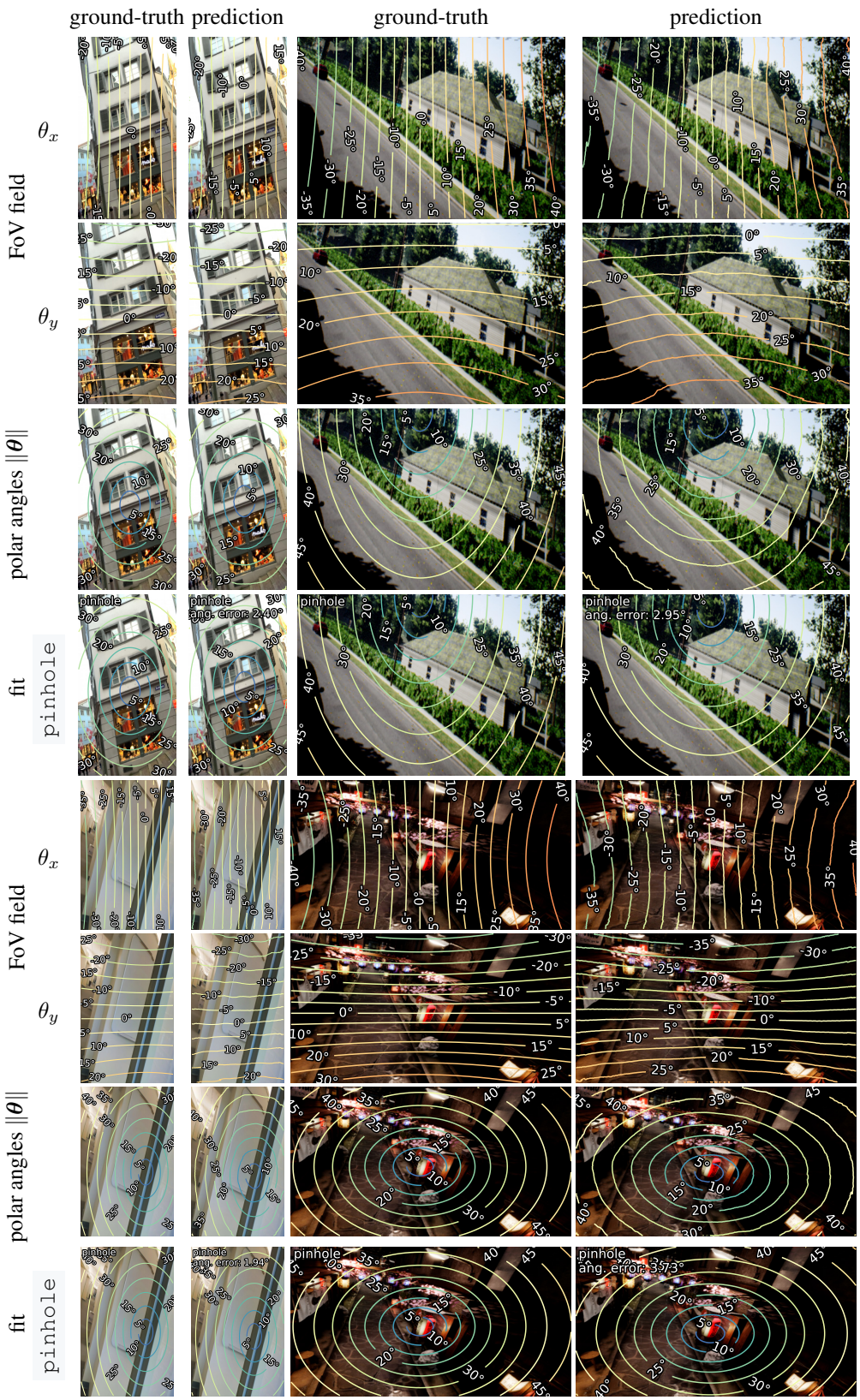


Figure 13. **Qualitative results on edited images** with AnyCalib being trained following [34, 100] (Sec. 3.3).

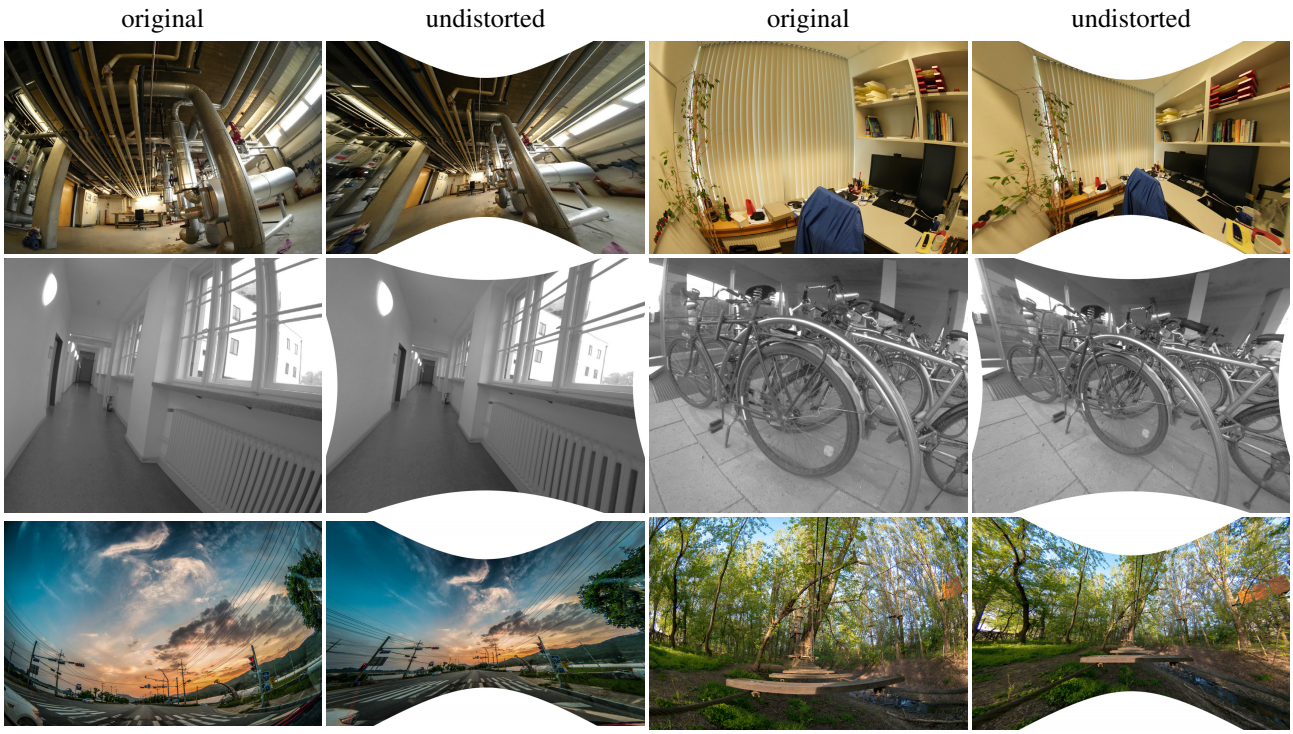


Figure 14. **Qualitative undistortion results** with AnyCalib$_{\text{gen}}$ (trained on OP$_{\text{g}}$), on images from ScanNet++ [95] (top), Mono [23] (middle) and captured with a Samsung NX 10mm F3.5 Fisheye lens (bottom), provided by ExploreCams—authors: crystal Yang (left) and Imre Farago (right).