A Deep Perceptual Measure for Lens and Camera Calibration Supplementary Material

Yannick Hold-Geoffroy[1], Dominique Piché-Meunier[3], Kalyan Sunkavalli[1],
Jean-Charles Bazin[2], François Rameau[2], and Jean-François Lalonde[3]
Adobe[1], KAIST[2], Université Laval[3]

1 Overview

These supplementary materials present the following additional results:

- Fig. 1 of the current document presents additional qualitative results on horizon estimation to complement fig. 2 from the main paper;
- Fig. 2 of the current document presents additional qualitative results for the network feature analysis to complement fig. 4 from the main paper;
- Fig. 3 of the current document presents a comparison of the network feature analysis before and after training, to complement fig. 4 from the main paper.
- Fig. 4 of the current document presents additional undistortion results to complement fig. 9 from the main paper;
- Fig. 5 and fig. 6 of the current document present examples of failure cases, for images with camera parameters outside of the ranges seen during training, and for images where the principal point is not in the middle.
- Sec. 7 provide the training details used to obtain out model.
- Sec. 8 of the current document presents derivations of equations for intrinsic parameters as well as relationships between them, extending sec. 3 of the main paper;
- Sec. 5 and table 1 presents the methodology used to compare state-of-the-art calibration methods as well as the estimated parameters by each method, extending sec. 5.3 and table 2 from the main paper;
- Fig. 7-10 of the current document presents additional user study results, comparing different parameters combinations and extending fig. 8 from the main paper;
- Sec. 9 shows an example of geometrically consistent object transfer as described in sec. 7 of the main paper;
- Fig. 12 of the current document illustrates our quantitative perceptual measure described in sec. 6.5 of the main paper.

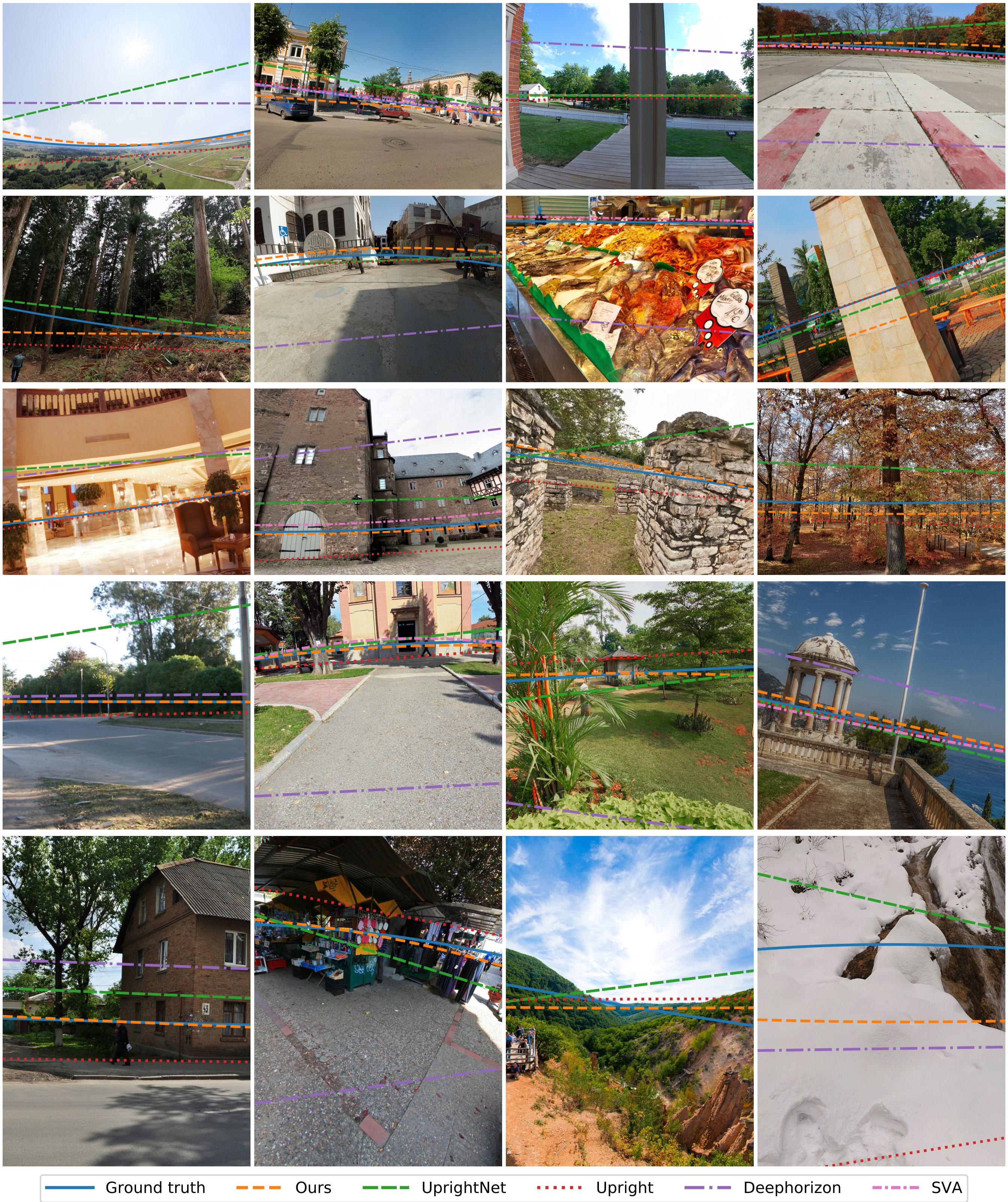


Fig. 1. Additional results of horizon line estimation randomly selected from our test set to complement fig. 2 from the main paper. We provide the ground truth field of view to UprightNet [1]. Only our method and SVA [2] allow for curved horizon lines (due to large distortions). Note how Upright and SVA perform well when sharp human-made objects are present in the scene, whereas deep learning methods offer a more robust performance across all scenes. Note also that SVA fails on 49% of our test images.

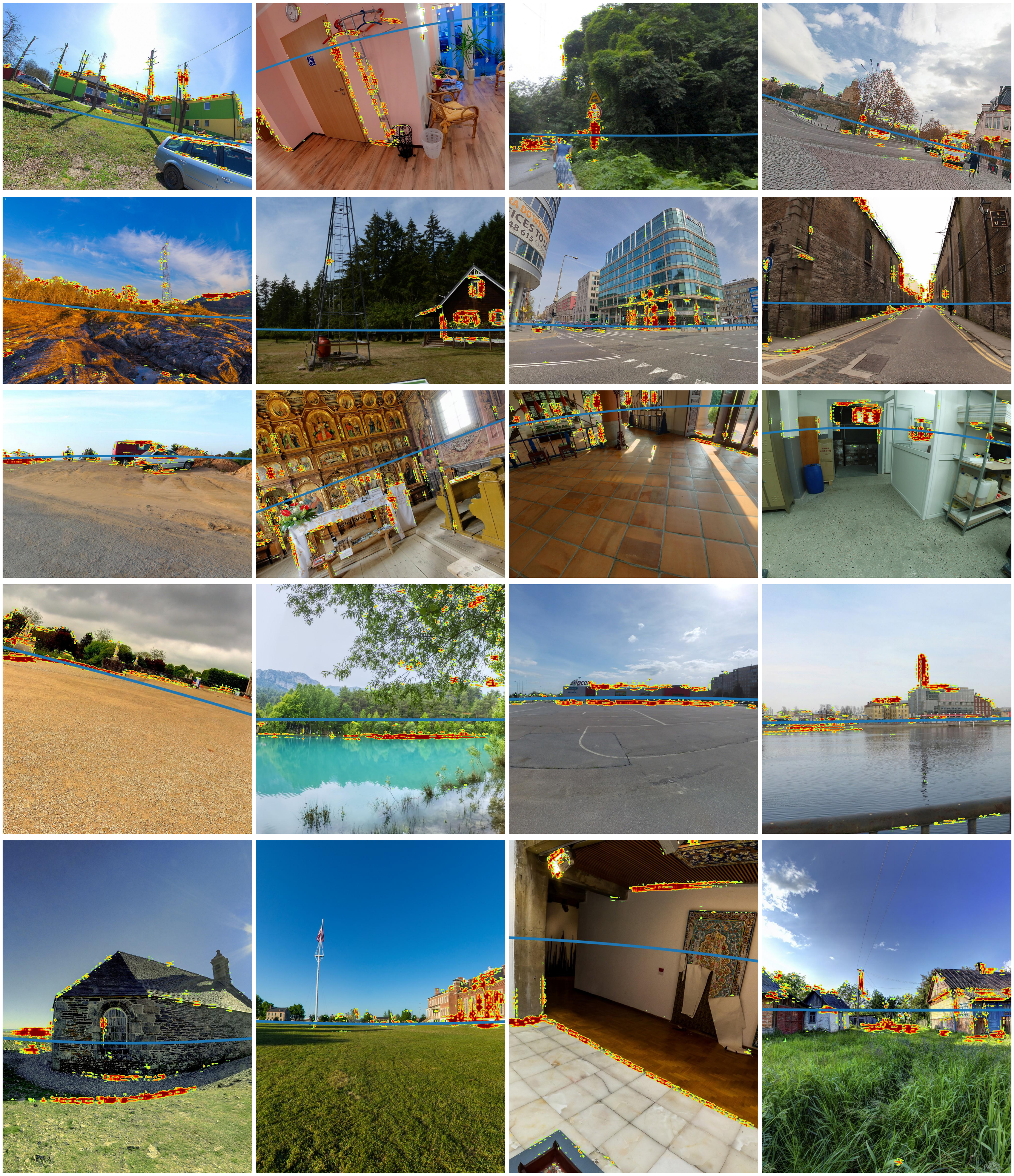


Fig. 2. Additional results for the analysis of the neural network focus to complement fig. 4 from the main paper. The result of smoothed guided backpropagation is displayed as a jet overlay, and the estimated horizon line is shown in blue.

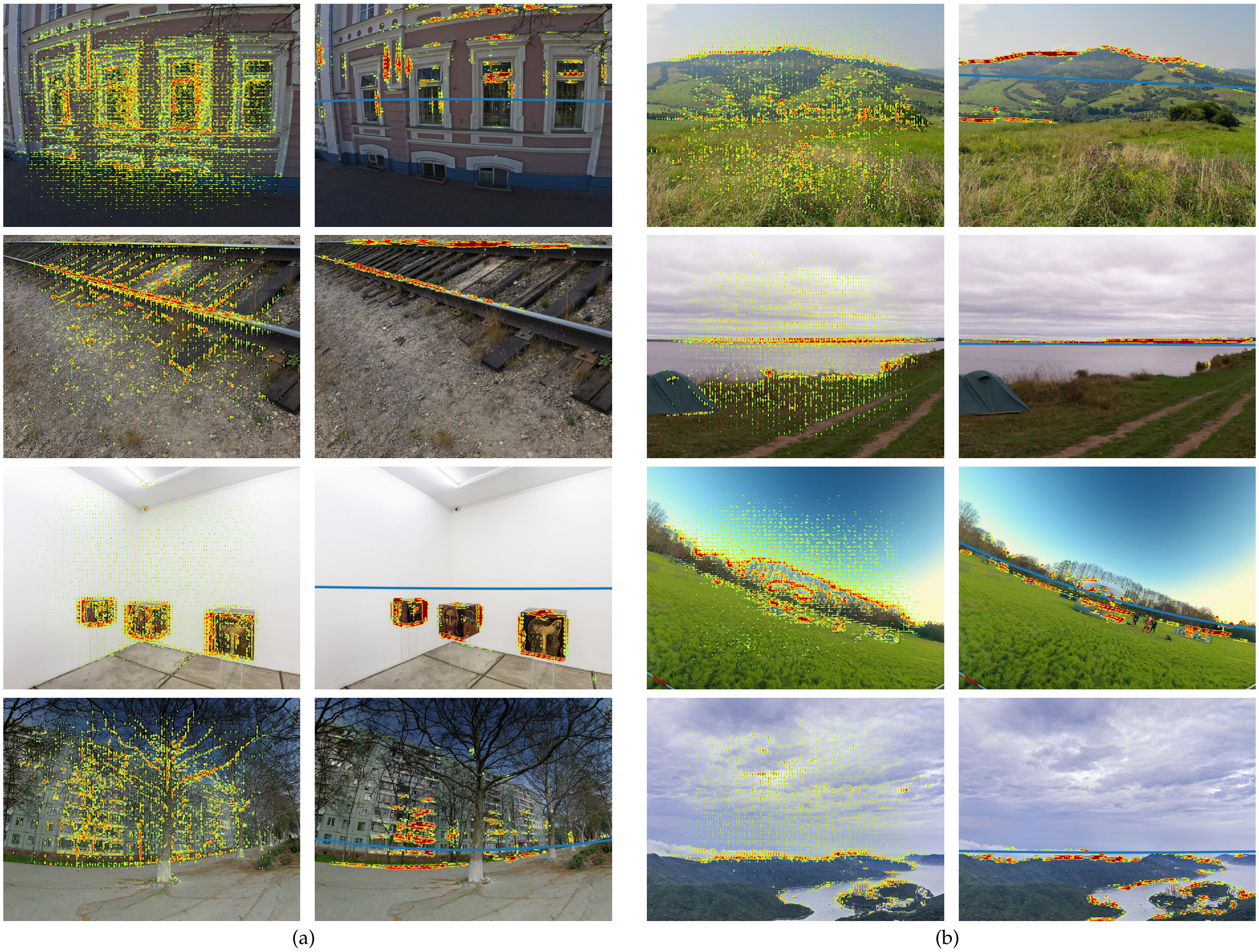
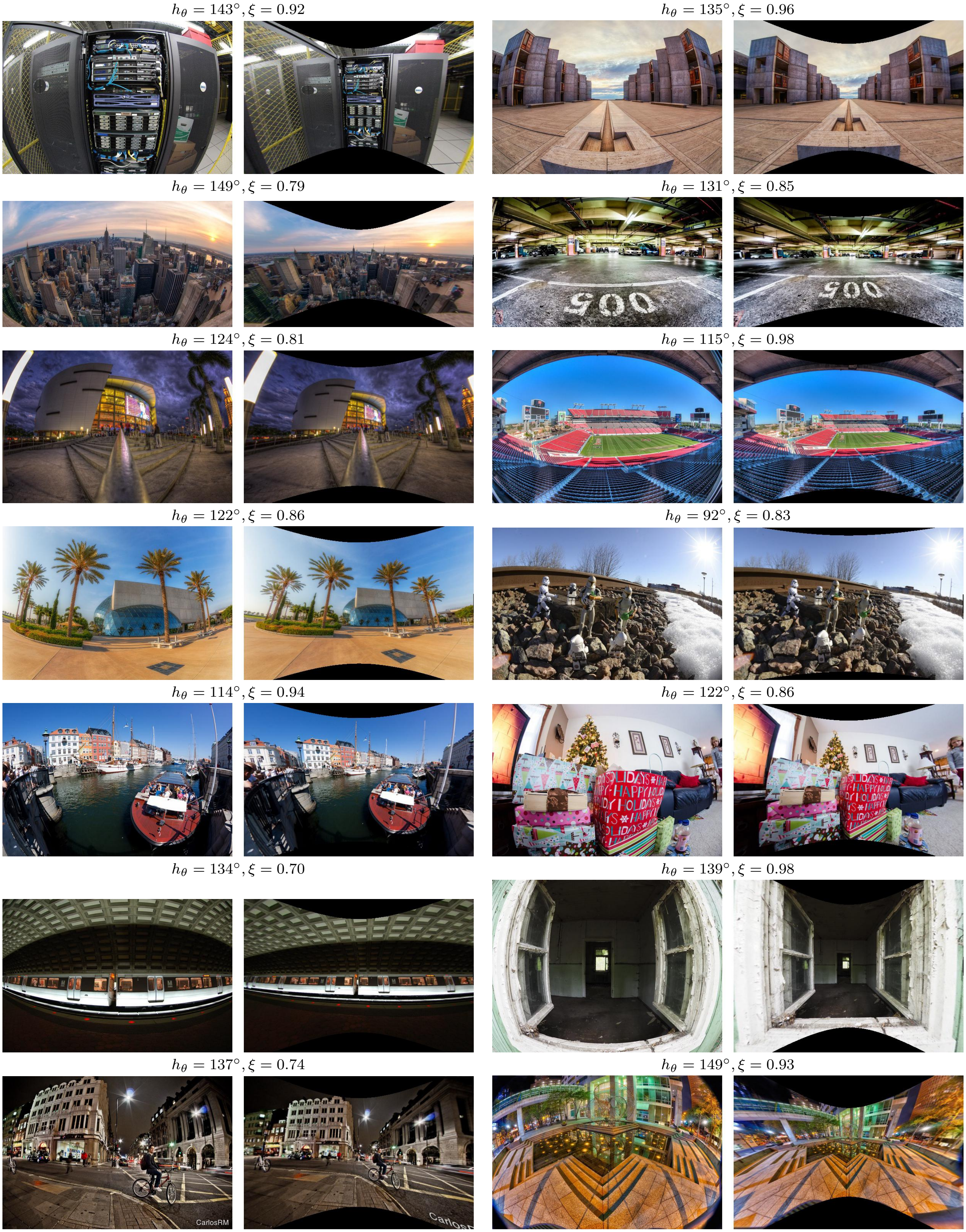


Fig. 3. Comparison of the neural network focus, when pretrained on ImageNet (left); and after being trained on our dataset (right). The result of smooth guided backpropagation is displayed as a jet overlay, on images with and without clear vanishing lines (resp. (a) and (b)). We can see that the network learns to ignore objects like trees or clouds, and focuses instead on vanishing lines or sky/land boundaries, which provide more clues about the location of the horizon line.



$h_\theta = 143°, \xi = 0.92$

$h_\theta = 135°, \xi = 0.96$

$h_\theta = 149°, \xi = 0.79$

$h_\theta = 131°, \xi = 0.85$

$h_\theta = 124°, \xi = 0.81$

$h_\theta = 115°, \xi = 0.98$

$h_\theta = 122°, \xi = 0.86$

$h_\theta = 92°, \xi = 0.83$

$h_\theta = 114°, \xi = 0.94$

$h_\theta = 122°, \xi = 0.86$

$h_\theta = 134°, \xi = 0.70$

$h_\theta = 139°, \xi = 0.98$

$h_\theta = 137°, \xi = 0.74$

$h_\theta = 149°, \xi = 0.93$

Fig. 4. Additional results for the automatic undistortion results on images in the wild to complement fig. 9 from the main paper, with the estimated field of view h_θ and distortion ξ. Left: Original image. Right: output of our algorithm.

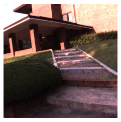
	Method	f	**distortion parameters** model-dependent (see text)	**avg err** (px)
Avenir 2.8mm 	Ours	1288	$\xi = 0.69$	0.64
	Mei [3]	1973	$\xi = 1.48$	0.12
	Division [4]	N/A	N/A	N/A
	Brown [5]	796	$K_n =$ -0.30, 0.17, -0.00, -0.00, -0.07	0.20
	Scaramuzza [6]	788	$a_n =$ -788.50, 0, 3.55^{-4}, 2.11^{-7}, -2.62^{-10}	1.01
Avenir 4mm 	Ours	1485	$\xi = 0.37$	0.57
	Mei	2270	$\xi = 0.97$	0.13
	Division	N/A	N/A	N/A
	Brown	1158	$K_n =$ -0.27, 0.28, 0.00, 0.00, -0.21	0.29
	Scaramuzza	1157	$a_n =$ -1157, 0.00, 2.89^{-4}, -2.45^{-7}, 1.79^{-10}	0.54
GoPro 	Ours	1182	$\xi = 0.82$	1.11
	Mei	1561	$\xi = 1.28$	0.12
	Division	787	$\lambda_n =$ 0.15, -0.77, 1.61, -0.98	0.70
	Brown	N/A	N/A	N/A
	Scaramuzza	791	$a_n =$ -791.4, 0.00, 0.00, -1.12^{-6}, 3.10^{-10}	0.82
Fisheye 	Ours	850	$\xi = 0.85$	1.25
	Mei	1351	$\xi = 1.79$	0.10
	Division	488	$\lambda_n =$ -0.09, 0.65, -1.79, 1.13	1.05
	Brown	N/A	N/A	N/A
	Scaramuzza	487	$a_n =$ -487.7, 0, 8.18^{-4}, -4.39^{-7}, 4.32^{-10}	1.48

TABLE 1
Camera calibration estimations for different cameras and calibration methods, including the estimated focal length f, the model-dependent distortion parameters, and the average pixel reprojection error. Please note that all the methods use multiple checkerboard pictures, except ours which require a single picture of a general scene. Both our method and Mei's [3] employ the spherical lens model parameterized by ξ (see sec. 3 of the main paper), while Fitzgibbon's division model [4], Brown's polynomial model [5] and Scaramuzza's toolbox [6] use sequences of real numbers to parameterize the radial distortion. We refer the reader to each publication for their respective definition of λ, K_n and a_n. Failures cases are noted N/A.

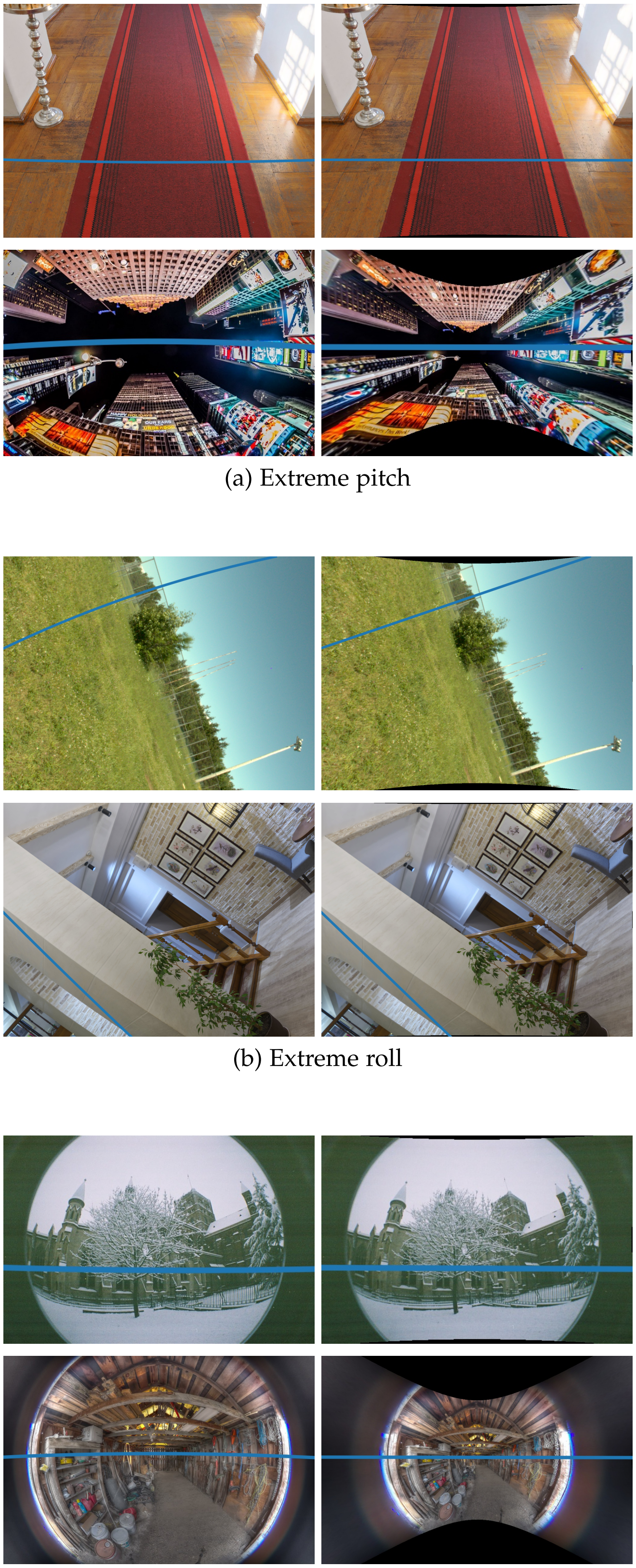
5 Comparison to state-of-the-art calibration methods

We provide the estimated parameters of each lens model as well as the average pixel reprojection error in table 1, extending the results from sec. 5.3 and table 2 of the main paper. In the following, we describe our methodology and capture setup for this experiment.

We experimented with four camera setups. The first three setups consist of a PointGrey Flea3 camera with a 1328×1048 resolution and equipped with three different lenses: 1) an Avenir 2.8mm providing a wide FoV, 2) an Avenir 4mm, 3) a GoPro HERO6 at a resolution of 1920×1080px, and 4) a Fujinon fisheye lens 1.8mm with a FOV over $180°$. To calibrate the cameras with existing methods based on checkerboards, we acquired about 30 pictures of checkerboard per camera, while we provided a single image of general scene per camera for our method (see images in the left column of Table 1). To compare the quality of the calibration results, we measure and report the reprojection error as reported by the calibration toolboxes. To compute the reprojection error of our approach, we input our calibration parameters (estimated from a single image) into Mei's toolbox and performed a pose-only bundle adjustment on the same checkerboard images (i.e., our parameters f and ξ, as well as the principal point (u_0, v_0), are not optimized).

As can be seen in the results, the conventional Brown model works well for most cameras but it is not suitable to map the distortion induced by a fisheye lens [7], which fails to converge on the GoPro and Fisheye cases. In practice, the automatic process in OpenCV Brown skipped around half of the calibration images for the GoPro camera and the Fujinon fisheye lens, which confirms that Brown's model is not adapted to wide FOV cameras. Alternatively, Fitzgibbon's division model [4]—also called the Fisheye calibration in OpenCV—fails to converge on the images taken with the Avenir lenses due to the low amount of distortion present in those images. However, it provides competitive accuracy on images with strongly visible distortion artifacts, such as the pictures taken with a GoPro and the Fisheye. Overall, most of the existing toolboxes obtains a sub-pixel accuracy. Despite providing slightly less accurate results than those existing toolboxes, our method still provides competitive results (between $^1/_2$px and 1px average reprojection error) across a large diversity of lenses using a single image *in the wild* as input, without calibration marker.

Finally, we would like to emphasize that our work does *not* aim to compete with checkerboard based approaches in terms of accuracy, but rather aims to provide a solution for camera calibration where traditional techniques cannot be applied, for example a single image in the wild, devoid of calibration reference.



(a) Extreme pitch

(b) Extreme roll

(c) Extreme distortion (catadioptric images)

Fig. 5. Examples of failure cases for images with camera parameters outside of the ranges seen during training. Left: estimated horizon line. Right: undistortion result.

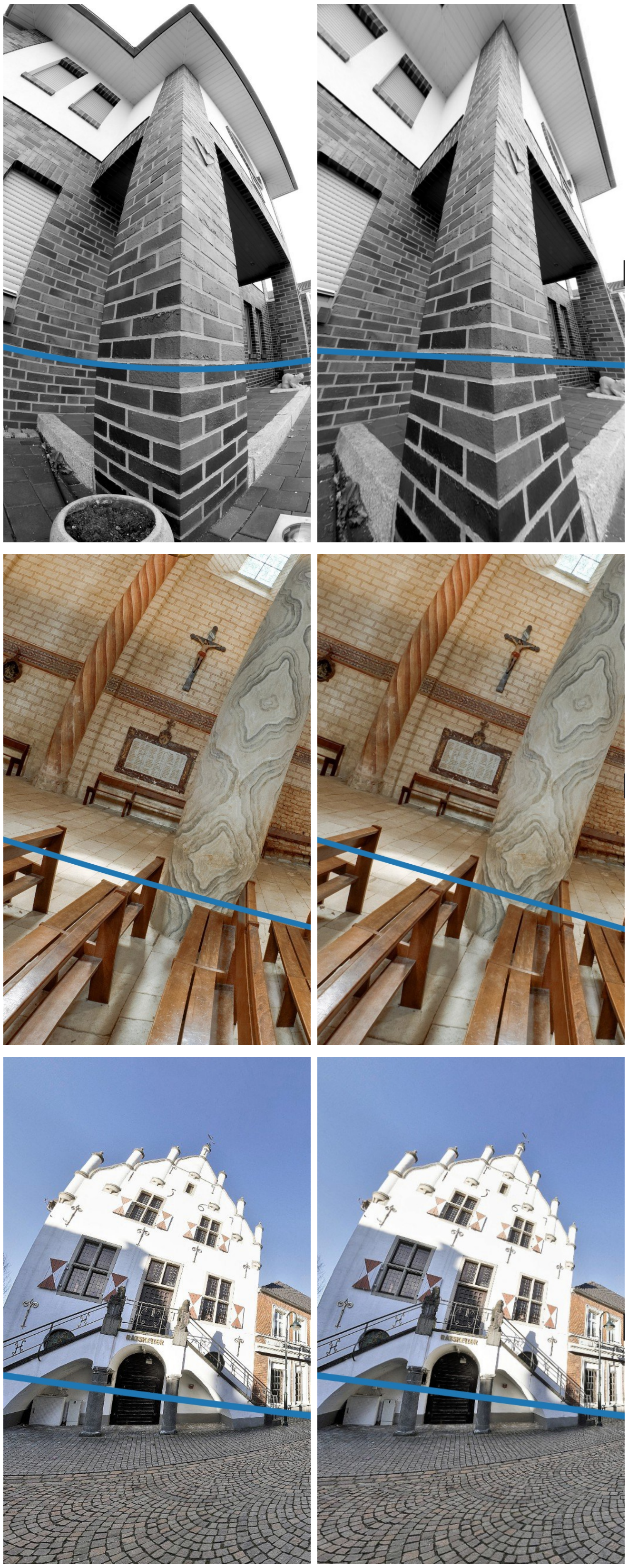


Fig. 6. Examples of failure cases when the principle point is not in the center of the image. Left: estimated horizon line. Right: undistortion result.

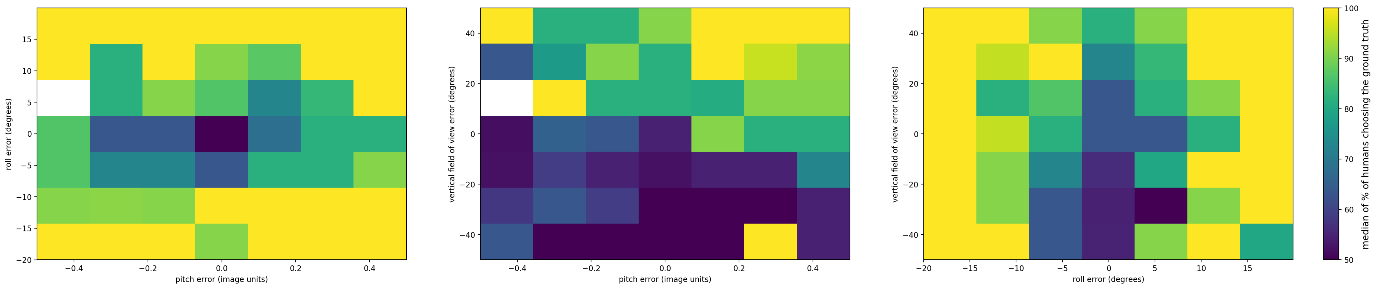


Fig. 7. Influence of parameter errors between themselves on human sensitivity to errors. We bin the percentage of people choosing the ground truth per image of the user study. The colors in the plot represents the median over all values in each bin. 100% (yellow) means humans always detect the distortion, 50% (blue) means total confusion, where humans statistically selects the distorted image as often as the ground truth.

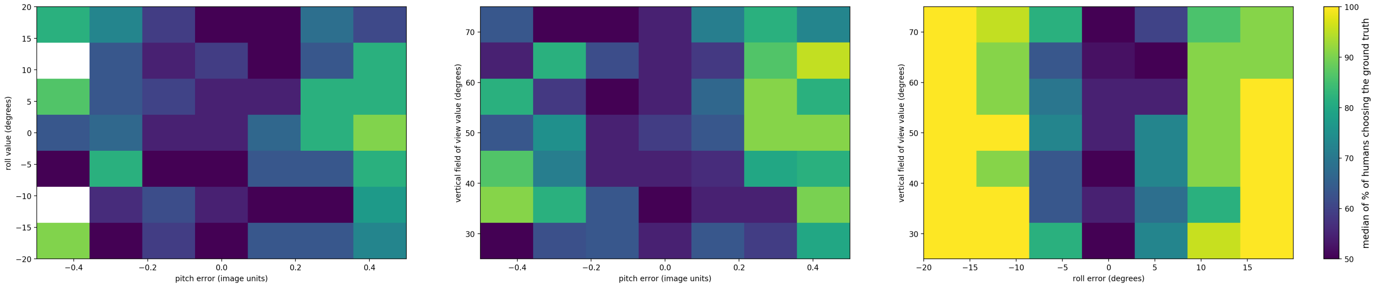


Fig. 8. Influence of parameter errors wrt. other parameter values on human sensitivity to errors.

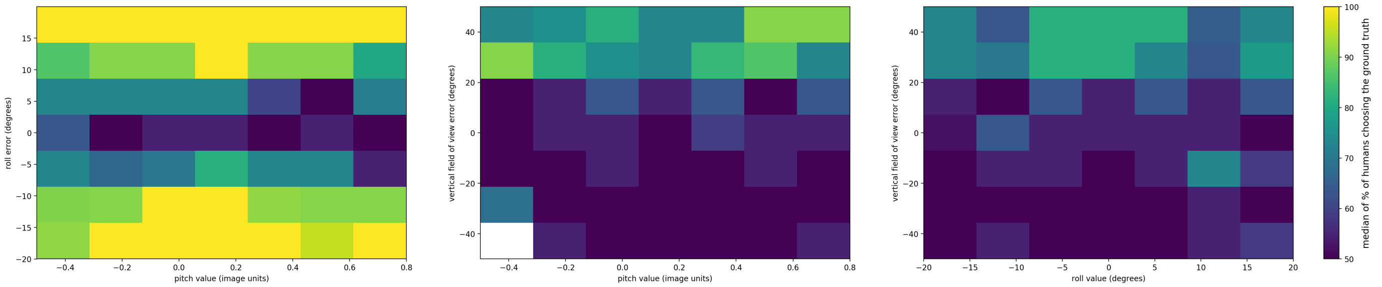


Fig. 9. Influence of parameter values wrt. other parameter errors on human sensitivity to errors.

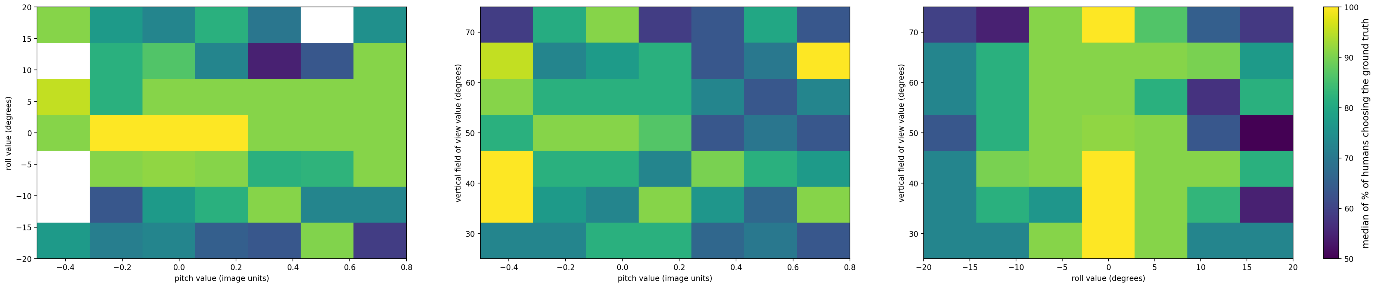


Fig. 10. Influence of parameter values between themselves on human sensitivity to errors. Note how when the roll parameter has a value very close to 0°, humans could detect distortions more easily, whereas larger roll value (image tilted around the viewing axis) yields lower detection rate by humans.

7 Training details

To train our model, we sum the Kullback-Leibler divergence computed on each output head and its corresponding ground truth (encoded as one-hot vectors) and use it as loss to train the model, which is minimized using stochastic gradient descent with the Adam optimizer [8] with an initial learning rate of $\eta = 0.001$ and a learning rate decay of $\alpha = 0.0002$. Training is performed on mini-batches of 42 images. Convergence is observed through early stopping typically after 9–10 epochs. We also trained our method with the field loss proposed in [9] and, despite an initial training instability alleviated with per-parameter pretraining, found no significant changes in test accuracy with our method.

8 Intrinsic parameters derivations and relationships

The spherical lens model used by our method has two degrees of freedom with respect to the camera intrinsics, despite having three explicit parameters. In the following, we first define the spherical lens model. We then describe how to compute either the focal length f, the distortion ξ or the effective horizontal field of view h_θ from the two other parameters.

8.1 Perspective model: focal length f and image resizing

First, we consider the pinhole model without lens distortion $\xi = 0$. The intrinsic calibration matrix is defined as

$$\mathbf{K} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{1}$$

In full resolution, a world point $[X, Y, Z]$ is projected to the image plane in pixels by

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \mathbf{K} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2}$$

i.e.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{fX}{Z} + u_0 \\ \frac{fY}{Z} + v_0 \end{bmatrix} \tag{3}$$

If we resize the image by a factor s, then this pixel is now at $[x', y'] = [{}^{x}/_{s}, {}^{y}/_{s})$, and the image center is not at $[{}^{u_0}/_{s}, {}^{v_0}/_{s})$. Let's denote f' the "new" focal length and

$$\mathbf{K}' = \begin{bmatrix} f' & 0 & u_0' \\ 0 & f' & v_0' \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} f' & 0 & u_0/s \\ 0 & f' & v_0/s \\ 0 & 0 & 1 \end{bmatrix}, \tag{4}$$

yielding

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \frac{f'X}{Z} + u_0' \\ \frac{f'Y}{Z} + v_0' \end{bmatrix} = \begin{bmatrix} \frac{f'X}{Z} + u_0/s \\ \frac{f'Y}{Z} + v_0/s \end{bmatrix}. \tag{5}$$

It follows from ${}^{x}/_{s} = \frac{f'X}{Z} + {}^{u_0}/_{s}$ and $x = \frac{fX}{Z} + u_0$ that $f' = {}^{f}/_{s}$.

In summary, resizing the image by a scaling factor s also scales its focal length f by ${}^{1}/_{s}$.

8.2 Sphere model: focal length f and distortion ξ

In the remainder of this section, we focus on defining the spherical lens model and establish relationships between its intrinsic parameters.

A point in 3D space $[X, Y, Z]$ is projected to the pixel $\mathbf{p} = [x, y]$ on the image plane using the relation

$$\begin{aligned} x &= \frac{Xf}{\xi\sqrt{X^2 + Y^2 + Z^2} + Z} + u_0 \\ y &= \frac{Yf}{\xi\sqrt{X^2 + Y^2 + Z^2} + Z} + v_0, \end{aligned} \tag{6}$$

The inverse transform, mapping an image point to a point on the unit sphere of the lens model, is defined as

$$\begin{aligned} \mathbf{P_s} &= (\omega\hat{x}, \omega\hat{y}, \omega - \xi) \\ \text{with} \quad \omega &= \frac{\xi + \sqrt{1 + (1 - \xi^2)(\hat{x}^2 + \hat{y}^2)}}{\hat{x}^2 + \hat{y}^2 + 1}, \end{aligned} \tag{7}$$

and

$$[\hat{x}, \hat{y}, 1]^T \simeq \mathbf{K}^{-1}\mathbf{p}, \tag{8}$$

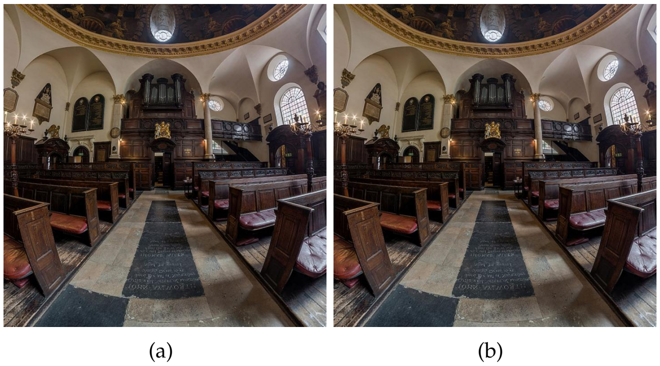
where $\mathbf{K}$ is defined in eq. (1).

When the image is resized by a scaling factor s, then a pixel is now at $[x', y'] = [\frac{x}{s}, \frac{y}{s}]$ and the image center is at $[\frac{u_0}{s}, \frac{v_0}{s}]$. Denoting the unknown resulting focal length f' and distortion parameter ξ', we have

$$\begin{aligned} \mathbf{p}' = [x', y'] &= [\frac{x}{s}, \frac{y}{s}] \\ &= \Bigg[\frac{Xf'}{\xi'\sqrt{X^2 + Y^2 + Z^2} + Z} + u_0', \\ &\qquad \frac{Yf'}{\xi'\sqrt{X^2 + Y^2 + Z^2} + Z} + v_0'\Bigg] \\ &= \left[\frac{Xf'}{\xi' + Z} + u_0', \frac{Yf'}{\xi' + Z} + v_0'\right] \\ &= \left[\frac{Xf'}{\xi' + Z} + \frac{u_0}{s}, \frac{Yf'}{\xi' + Z} + \frac{v_0}{s}\right]. \end{aligned} \tag{9}$$

To have $[x', y'] = [\frac{x}{s}, \frac{y}{s}]$ for all $[X, Y, Z]$, we need

$$\left(\frac{Xf'}{\xi' + Z}, \frac{Yf'}{\xi' + Z}\right) = \left[\frac{1}{s}\left(\frac{Xf}{\xi + Z}\right), \frac{1}{s}\left(\frac{Yf}{\xi + Z}\right)\right]. \tag{10}$$



(a)

(b)

Fig. 11. Testing the intrinsic parameters wrt image resizing. From an original image (a) with known focal length f and distortion ξ, we back-projected the image pixels to the sphere of our model and re-projected them (b) to a image plane resized by a scaling factor s, using the new effective focal length $f' = \frac{f}{s}$ and keeping the same ξ. This resulting image (b) is identical to the original (a) up to interpolation noise.

One possible solution is to have $f' = {}^{f}\!/_{s}$ and $\xi' = \xi$. It is important to note that this is not the only possible solution, as a family of solutions can be derived from those equations.

Fig. 11 shows an example of image resizing, where the original image (left) has been back-projected to the sphere and re-projected onto a new image plane with a different focal length $\frac{f}{s}$ while keeping ξ the same. As can be seen, both images are identical (up to interpolation noise), offering an empirical validation to our derivations.

8.3 Compute h_θ from f and ξ

We assume that the image size in pixels $H \times W$ is known and the image principal (center) point is at the center of the image $[u_0, v_0] = [{}^{W}\!/_{2}, {}^{H}\!/_{2}]$.

Projecting the center point $[u_0, v_0]$ onto the sphere by applying the camera intrinsics matrix K^{-1} and projecting it to the image plane, we obtain

$$P_0 = [0, 0, 1]. \tag{11}$$

Projecting the "left point" $[0, {}^{H}\!/_{2}] = [0, v_0]$ in the same way, we obtain a position on the sphere $[\hat{x}, \hat{y}, \hat{w}] = [{}^{-u_0}\!/_{f}, 0, 1]$, and a position on the unit sphere

$$P_s = \left[\omega\hat{x},\ 0,\ \frac{\xi + \sqrt{1 + (1-\xi^2)\hat{x}^2}}{\hat{x}^2 + 1} - \xi\right], \tag{12}$$

for a given factor ω. We can obtain the field of view h_θ by doubling the subtended angle between P_0 and P_s as

$$\begin{aligned} h_\theta &= 2 \arccos\left(P_0 \cdot P_s\right) \\ &= 2 \arccos\left(\frac{\xi + \sqrt{1 + (1-\xi^2)\hat{x}^2}}{\hat{x}^2 + 1} - \xi\right), \end{aligned} \tag{13}$$

where $\hat{x} = -u_0/f$.

Note that if u_0 (i.e. the image size) and f are scaled or resized by the same value s, then the field of view does not change for a given fixed ξ.

8.4 Using h_θ, compute ξ from f or vice-versa

The central point projected to the image plane is $P_0 = [0, 0, 1]$, (see sec. 8.3). We can define the x-axis component of the "left point" using

$$X = -\sin\left(\frac{h_\theta}{2}\right). \tag{14}$$

Since the spherical lens model uses a unit sphere $X^2 + Z^2 = 1$, we obtain

$$Z = \pm\sqrt{1 - X^2}. \tag{15}$$

The left point projected to the image plane can be expressed as

$$[0, v_0] = \left[\frac{Xf}{\xi + Z} + u_0, \frac{Yf}{\xi + Z} + v_0\right]. \tag{16}$$

The horizontal field of view h_θ is defined as the angle subtended by the spherical points on the horizon plane X-Z, which means $Y = 0$, hence $\frac{Yf}{\xi \cdot 1 + Z} + v_0 = 0 + v_0 = v_0$, validating eq. (16). For X, we have the relationship $\frac{Xf}{\xi \cdot 1 + Z} + u_0 = 0$. Given ξ, we obtain

$$f = -u_0 \frac{\xi + Z}{X}, \tag{17}$$

where X and Z are known given the effective horizontal field of view h_θ. Similarly, given f, we obtain

$$\xi = \frac{Xf + u_0 Z}{-u_0}. \tag{18}$$

8.5 Compute the image midpoint v_{m} using f and the camera pitch θ

As in the previous sections, we assume the image size in pixels $H \times W$ is known and the camera principal (center) point to be in the middle of the image $[u_0, v_0] = [{}^{W}\!/_{2}, {}^{H}\!/_{2}]$.

The central point projected to the image plane is $P_0 = [0, 0, 1]$, (see sec. 8.3). Rotating the central point on the sphere by the camera pitch (elevation) angle θ, we obtain

$$\begin{aligned} P_s &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ -\sin\theta \\ \cos\theta \end{bmatrix}. \end{aligned}$$

Projecting on the image plane, we obtain get

$$[x, y] = \left[u_0, \frac{-f\sin\theta}{\xi + \cos\theta} + v_0\right]. \tag{19}$$

The midpoint is thus

$$v_{\mathrm{m}} = -f\frac{\sin\theta}{\xi + \cos\theta} + v_0. \tag{20}$$

Note that if we set $\xi = 0$, we obtain $v_{\mathrm{m}} = -f\tan\theta + v_0$, the usual equation for perspective images without radial distortion.

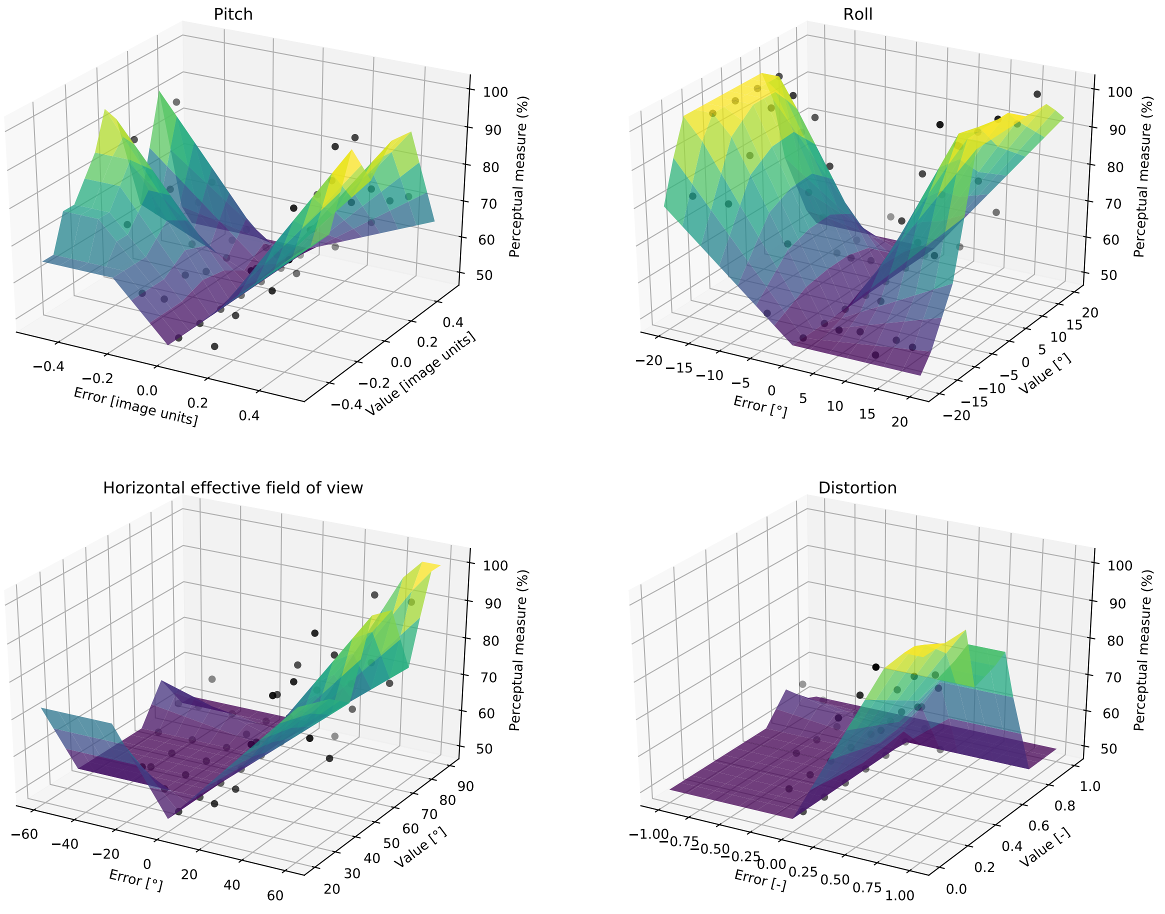


Fig. 12. Visualization of the quantitative perceptual measure described in sec. 6.5 of the main paper. Clockwise from the top-left: pitch, roll, distortion and field of view functions. All functions are computed by binning the perceptual data into a 7×7 histogram on the 2D space of errors and absolute parameter, and piece-wise linear functions are then fitted on the binned data. Lower is better, as a perceptual measure of 50% means the network can fool a human.

9 Geometrically-consistent object transfer

Transferring objects from one image to another requires matching the camera parameters. While previous techniques required the use of objects of known height in the image in order to infer camera parameters [10], our approach estimates them from the image itself, and can be used to realistically transfer objects from one image to another, as shown in fig. 13. In this example, we obtained the parameters of the background scene and sought an image with similar parameters, which gave us an image with a water tower. Despite the differences in lighting between the water tower and the background scene, note that simply copying pixels between the two matching images keeps the perspective correct. The foreshortening and orientation of the water tower matches well the scene, which provides a good starting point for an artist to work on this composite image.

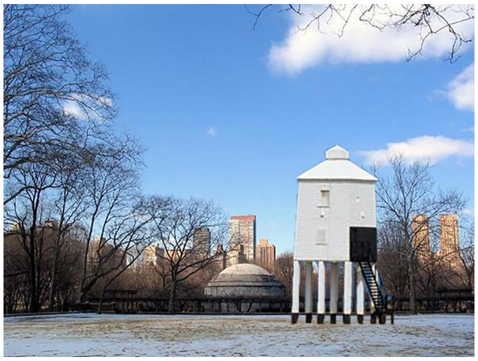


Fig. 13. The water tower from fig. 10 of the main paper (bottom row) pasted onto an image with an automatically detected similar horizon line. Note how the perspective looks right without modification.

References

[1] W. Xian, Z. Li, M. Fisher, J. Eisenmann, E. Shechtman, and N. Snavely, "Uprightnet: Geometry-aware camera orientation estimation from single images," in *ICCV*, 2019, pp. 9974–9983.

[2] Y. Lochman, O. Dobosevych, R. Hryniv, and J. Pritts, "Minimal solvers for single-view lens-distorted camera auto-calibration," in *WACV*, January 2021, pp. 2887–2896.

[3] C. Mei and P. Rives, "Single view point omnidirectional camera calibration from planar grids," in *ICRA*, 2007.

[4] A. Fitzgibbon, "Simultaneous linear estimation of multiple view geometry and lens distortion," in *CVPR*, 2001.

[5] Z. Zhang, "A flexible new technique for camera calibration," *TPAMI*, 2000.

[6] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A toolbox for easily calibrating omnidirectional cameras," in *IROS*, 2006.

[7] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional , wide-angle, and fish-eye lenses," *TPAMI*, 2006.

[8] D. Kingma and J. Ba, "ADAM: A Method for Stochastic Optimization," in *International Conference for Learning Representations*, 2015.

[9] M. López-Antequera, R. Marí, P. Gargallo, Y. Kuang, J. Gonzalez-Jimenez, and G. Haro, "Deep single image camera calibration with radial distortion," in *CVPR*, 2019.

[10] J.-F. Lalonde, D. Hoiem, A. A. Efros, C. Rother, J. Winn, and A. Criminisi, "Photo clip art," *SIGGRAPH*, vol. 26, no. 3, p. 3, 2007.