

Perspective Fields for Single Image Camera Calibration

Supplementary Material

Linyi Jin¹, Jianming Zhang², Yannick Hold-Geoffroy², Oliver Wang²,
Kevin Blackburn-Matzen², Matthew Sticha¹, David F. Fouhey¹
University of Michigan¹, Adobe Research²

¹{jinlinyi, msticha, fouhey}@umich.edu

²{jianmzha, holdgeof, owang, matzen}@adobe.com

A. Video

Please check out the [video](#) for a demo.

B. Additional experiments

B.1. Perspective Fields on warped images

Table 1 shows the additional test results on warped images, extending Table 1 of the main paper. On warped images, which is another common operation of image post-processing, our method continues to outperform other baselines and keeps the error on Up and Latitude low. Previous methods which assume a global set of parameters poorly describe the perspective of the image and have a large performance drop.

B.2. Ablations: training on centered principal point images.

Our method is trained on non-centered principal points images. In Table 3, we re-train Ours without `RandomResizedCrop` during data augmentation (*Ours-centered*) so that all the methods are trained on centered principal point images. We show results on the test set and compare to Ours and the most competitive baseline Percep. [4]. When tested on centered principal point images (*Perturb=None*), *Ours-centered* is better than *Ours* in Table 1. When tested on image crops (*Perturb=Crop*), *Ours-centered* is slightly worse than *Ours*, but better than all other baselines. We obtain similar results on the TartanAir [45] dataset (not shown due to limited space). Even when trained on centered principal points, the dense per-pixel nature of the representation makes *Ours-centered* to be robust to image crops.

In Table 2, distilling the *Ours-centered* version on crops for COCO also improves over the baselines and is comparable to *Ours-distill*, see Table 4. For example, when *Perturb=crop*, it has 3.93 vs 3.76 median error for Up and 6.66 vs 7.57 median error for Latitude; when *Perturb=isolated*,

it has 4.57 vs 4.12 median error for Up and 10.08 vs 9.56 median error for Latitude (*Ours-centered-distill* vs *Ours-distill*).

B.3. Camera parameter estimation using optimization

In Sec. 4.2 we have shown that camera parameters can be accurately recovered from Perspective Fields using the ParamNet. In this section, we will show that optimization can also be used to recover camera parameters and, in some cases, to improve upon predictions from ParamNet.

Setup. The optimization problem is five dimensional as the five optimizable parameters are roll, pitch, relative focal length and the principal point (cx, cy). The relative focal length is defined as the focal length divided by image height. Relative focal length is then converted to FoV for evaluation. Adam was chosen as the optimizer with a learning rate of 10^{-4} . The optimization runs for 1000 iterations and stops if $\text{loss} < 10^{-7}$ or if $\text{loss} - \text{previous_loss} < 10^{-9}$. To perform the optimization, the Up-vector and Latitude fields are generated from the optimizable camera parameters. The loss is then calculated between these predicted fields and the ground truth Up-vector and Latitude fields. The objective function we minimize is the APFD metric

$$\text{Loss} = \lambda \arccos(\mathbf{u}_1 \cdot \mathbf{u}_2) + (1 - \lambda) \|l_1 - l_2\|_1, \quad (1)$$

where \mathbf{u}_i is the Up-vector and l_i is the Latitude value. The weight $\lambda = 0.5$ is used in our experiments.

Parameter Initialization. We experiment with two different methods of initializing the camera parameters for the optimization. *Opt:* Let \mathbf{u}_x be defined as the center of the Up-vector field and l_x be defined as the center of the Latitude field. The camera roll is initialized to $-\arctan(\mathbf{u}_{x_0} / -\mathbf{u}_{x_1})$. The pitch is initialized to l_x . Let l_1 be the value of the Latitude map at the top center of the image and l_2 be the value of the Latitude map at the bottom center of the image. The

Table 1. Quantitative evaluation for scene-level Perspective Field prediction on warped images, extending Table 1. We re-implement Percep. [4] using the same backbone and training data as ours. None of the methods have been trained on Stanford2D3D [2] or TartanAir [7].

Dataset	Perturb	Stanford2D3D [2]						TartanAir [7]					
		Up (°)			Latitude (°)			Up (°)			Latitude (°)		
Method		Mean ↓	Median ↓	% < 5° ↑	Mean ↓	Median ↓	% < 5° ↑	Mean ↓	Median ↓	% < 5° ↑	Mean ↓	Median ↓	% < 5° ↑
Upright [5]	Warp	11.16	10.47	38.46	20.50	20.38	13.65	13.77	13.11	34.82	18.20	18.44	15.89
Percep. [4]	Warp	10.01	9.25	34.29	14.23	13.77	20.93	9.55	8.76	33.84	9.85	9.59	27.14
CTRL-C [6]	Warp	15.92	14.79	19.86	13.09	12.38	22.96	14.61	13.34	20.72	10.86	10.66	24.44
Ours	Warp	3.39	2.72	66.82	5.95	5.48	46.79	4.11	3.45	61.08	5.47	5.12	48.62

Table 2. GSV *uncentered* principal-point optimization results. ParamNet is our method described in the main paper that regresses the camera parameters from predicted Perspective Fields. We show that camera parameters can be further improved by using optimization to adjust the predicted camera parameters to better match the Perspective Fields.

Method	Roll (°) ↓		Pitch (°) ↓		FoV* (°) ↓		cx ↓		cy ↓		Up(°)		Latitude(°)	
	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean. ↓	% < 5° ↑	Mean. ↓	% < 5° ↑
ParamNet	1.37	0.97	2.60	2.14	3.75	3.19	0.09	0.07	0.08	0.06	1.05	98.95	2.17	89.47
Opt	1.90	1.15	3.68	2.90	3.80	3.16	0.12	0.10	0.09	0.07	1.00	99.40	1.93	93.15
ParamNet + Opt	1.41	0.95	2.60	2.14	3.72	3.17	0.10	0.08	0.08	0.06	0.80	99.40	1.91	93.30

FoV is initialized to $l_1 - l_2$. The second method of initializing the camera parameters (*ParamNet + Opt*) initializes them to the output of ParamNet. The FoV is converted to relative focal length for the optimization.

Results. Results for both of these initialization methods on cropped images are shown in table 2. *ParamNet* is our method described in the main paper that regresses the camera parameters from predicted Perspective Fields. We show that our camera parameters can be further improved by using optimization to adjust the predicted camera parameters to minimize the APFD error with the predicted Perspective Fields. The method that combines ParamNet and optimization has 3.8% higher accuracy in the Latitude value.

C. Evaluation Details

C.1. Dataset

Scene level training set. Our training dataset contains 360° panoramas in equirectangular format which covers 180° vertically and 360° horizontally. The dataset contains diverse scenes including 30,534 indoor, 51,157 natural and 110,879 street views. We sample crops from the panoramas with camera roll in $[-45^\circ, 45^\circ]$, pitch in $[-90^\circ, 90^\circ]$ and FoV in $[30^\circ, 120^\circ]$. Our training and validation set consist of 190830/1740 panorama images respectively. We crop one perspective image per panorama and filter out ones without too much context (if all pixels values are white or black). Fig. 1 shows the camera parameter distribution of our training dataset.

Object centric training set. We choose images from COCO training set and inference our perspective field predictor to generate pseudo ground truth. We select categories in “bicycle”, “book”, “bottle”, “chair”, “laptop” and large objects whose area are greater than $96^2 = 9216$ pixels. We

also discard examples with low entropy value (< 3.5) from our network classification results. As a result, we generate a training set with 8192 images.

C.2. Training details

We use a transformer-based backbone from SegFormer [8] to extract features from the input RGB image. Specifically, we use the Mix Transformer encoders (MiT-B3) designed in SegFormer to extract hierarchical features. It extracts course and fine features from the hierarchical Transformer encoder using embedding dimensions of 64, 128, 320, 512. We find that the transformer based encoder is effective for our task since it can enforce global consistency in the perspective fields well.

The features are then fed into the All-MLP decoder in SegFormer. The decoder produces a distribution over a set of up directions or latitude values with the same resolution as the input image. The up-vector head predicts $k_{up} = 72$ classes representing evenly spaced unit vectors in 2D space. The latitude head predicts $k_{lati} = 180$ classes representing a discrete set of latitude value for each pixel evenly spaced from $-\pi/2$ to $\pi/2$.

The input resolution is 320×320 . We apply random flipping, rotation, color jittering and blurring to the training data. Since our perspective fields are translation invariant and defined on images with different geometric operations such as cropping, we also have random cropping and resizing on both the input image and ground truth perspective fields as part of the data augmentation. We use the SGD optimizer with momentum of 0.9. The learning rate is 0.01. The batch size is 32.

C.3. Test set details

Stanford2d3d / TartanAir test set generation. Assuming

Table 3. We re-train Ours without RandomResizedCrop during data augmentation (*Ours-centered*) so that all the methods are trained on centered principal point images, extending Table 1. We compare to Ours and the most competitive baseline Percep. [4].

Dataset		Stanford2D3D [2]						TartanAir [7]					
Method	Perturb	Up (°)			Latitude (°)			Up (°)			Latitude (°)		
		Mean ↓	Median ↓	% < 5° ↑	Mean ↓	Median ↓	% < 5° ↑	Mean ↓	Median ↓	% < 5° ↑	Mean ↓	Median ↓	% < 5° ↑
Percep. [4]	None	3.58	3.32	64.19	6.27	6.07	42.36	7.30	6.86	47.04	11.35	11.22	27.69
Ours	None	2.18	1.88	82.83	3.40	3.06	68.27	3.47	2.86	67.45	4.01	3.60	61.73
Ours-centered	None	1.83	1.66	89.09	2.06	1.88	82.02	2.11	1.86	83.06	2.23	2.04	81.03
Percep. [4]	Crop	5.78	5.55	45.52	9.76	9.65	29.13	5.54	5.18	51.72	9.22	8.66	30.10
Ours	Crop	2.21	1.87	78.80	5.57	5.15	50.36	2.81	2.35	71.89	5.73	5.28	50.16
Ours-centered	Crop	3.07	2.89	65.91	5.93	5.56	45.65	3.64	3.33	64.13	5.69	5.26	49.52

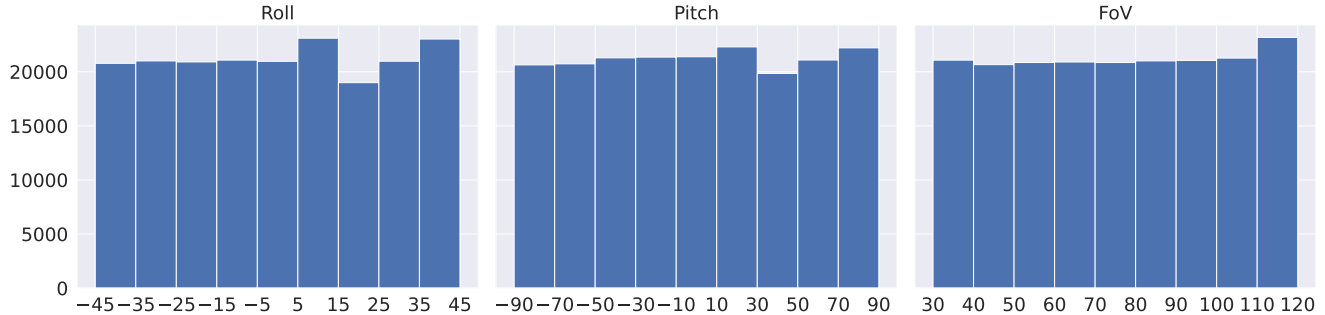


Figure 1. Training set camera distribution.

Table 4. Ablation study for training on centered principal point images only, extending Table 2.

Dataset		Objectron [1]					
Method	Perturb	Up (°)			Latitude (°)		
		Mean ↓	Median ↓	% < 5° ↑	Mean ↓	Median ↓	% < 5° ↑
CTRL-C [6]	crop	7.50	7.09	40.02	20.93	21.00	11.26
Ours-distill	crop	4.19	3.76	57.71	7.71	7.57	33.54
Ours-distill-centered	crop	4.19	3.93	57.31	7.02	6.66	36.76
CTRL-C [6]	isolated	7.49	7.13	39.38	9.87	9.85	27.32
Ours-distill	isolated	4.45	4.12	54.88	9.65	9.56	25.82
Ours-distill-centered	isolated	4.85	4.57	52.21	10.39	10.08	27.24

perspective projection, we uniformly sample 2,415 views from Stanford2D3D with camera roll in $[-45^\circ, 45^\circ]$, pitch in $[-50^\circ, 50^\circ]$ and FoV in $[30^\circ, 120^\circ]$. For TartanAir, we randomly sample 2,000 images from its test sequences with roll ranging in $[-20^\circ, 20^\circ]$, pitch in $[-45^\circ, 30^\circ]$, and fixed FoV (74°). To test the robustness of methods, we add image crop perturbation to the test image. We randomly crop a quarter of the original image of aspect ratio 1, which is implemented by RandomResizedCrop function from the Albumentation [3] package. The ground truth Perspective Fields can simply be cropped in the same way to match the RGB image. For warp perturbation, we perform a random four point perspective transform of the original image, the operation is also implemented in Albumentation [3], with hyperparameters set as scale=(0.1, 0.2), fit_output=False. The ground truth Latitude map is warped the same way as the RGB image. The corresponding Up-vectors are calculated by the Homography.

GSV uncentered principal-point test set generation. We randomly sample crops from the GSV views. Fig. 2 shows the camera parameter distribution for the GSV *uncentered* principal-point dataset.

C.4. Infer ground truth for in the wild images.

The qualitative examples in Figure 5 do not have a ground truth since they are from the internet. To help infer the ground truth, in Fig. 3 we show the location of the GT horizon location. Assuming the laptop is placed on a horizontal surface, we find the vanishing points of the two pairs of parallel lines (cyan dashed lines) at the base. The horizon line can be found by connecting the vanishing points (orange dashed lines), which is outside of the image. Our method has more accurate Latitude prediction compared to other baselines as shown in Figure 5 of the paper.

C.5. User study for perspective matching metrics.

Fig. 4 shows the statistics of the correlation scores for each metric. The box plot shows the minimum, maximum, median, 1st quartile, 3rd quartile and outliers of each metric following the standard box plot convention¹. For the camera parameter metrics, such as deviation in roll, pitch, FoV and the principal point (*Prin. Point*), the correlation score distributions vary wildly. Among them, deviation in FoV is a poor indication of human perception, which is consistent with [4]. The change in pitch is a dominant factor in

¹https://en.wikipedia.org/wiki/Box_plot

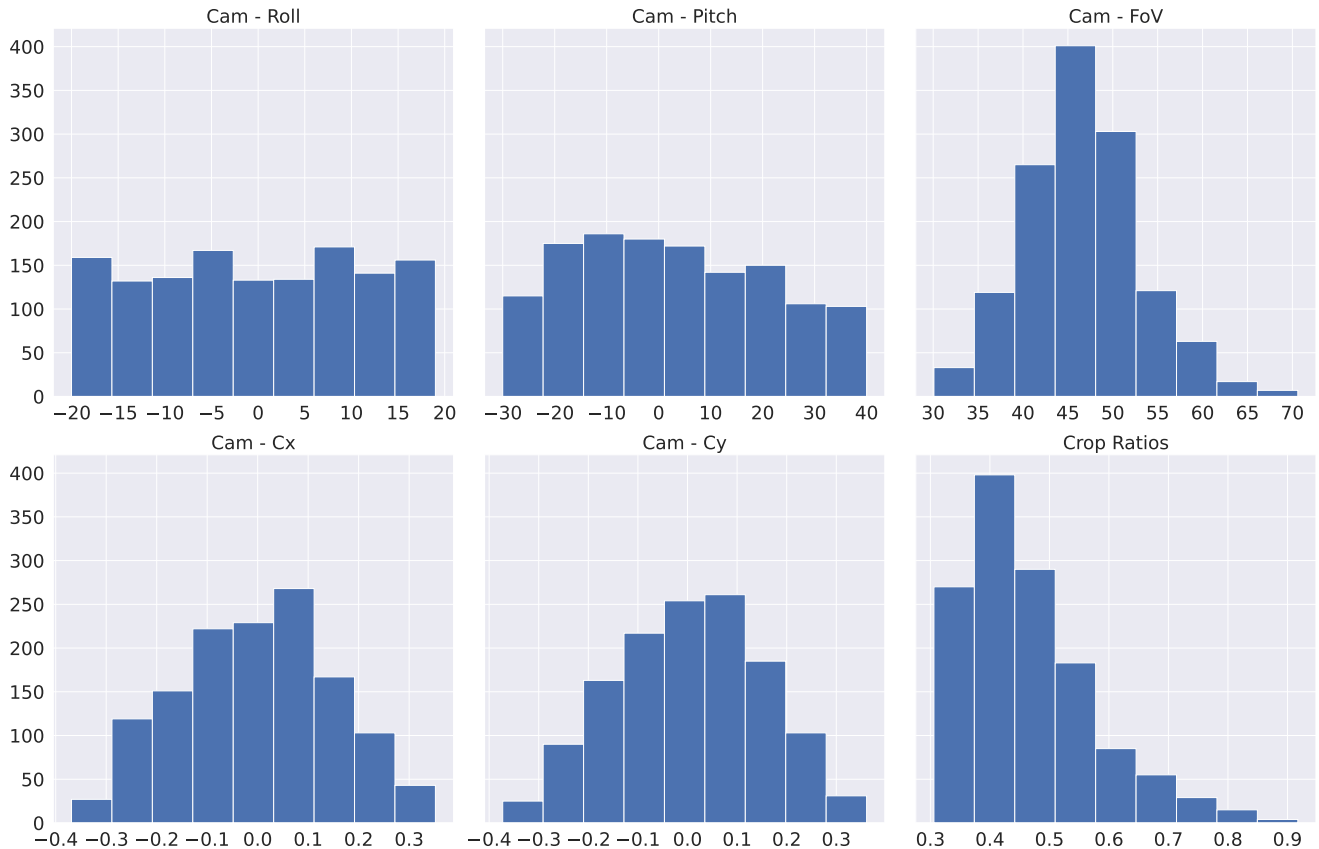


Figure 2. Camera parameter distribution for GSV *uncentered* principal-point dataset.

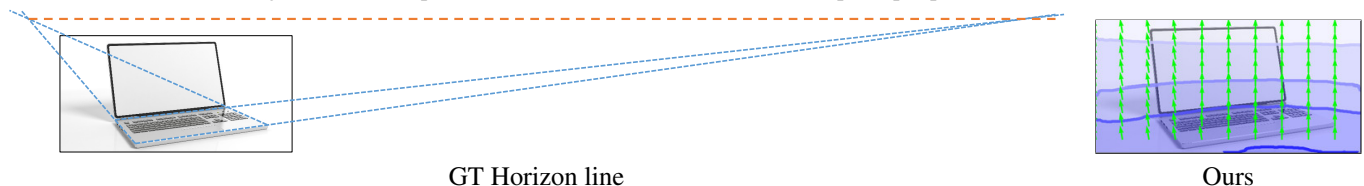


Figure 3. The GT horizon location (Orange dashed line) of the laptop example from the web image. Our method has more accurate Latitude prediction compared to other baselines as shown in Figure 5 of the paper.

perspective mismatch in our setting. Summing the parameter difference (*Camera All*) does not improve correlation scores, which shows the difficulty of using camera parameters to measure perceived perspective consistency. Fig. 5 shows the user rankings and APFD scores on different test images.

D. Additional Qualitative Results

Additional Qualitative Results on Test Set. We show qualitative results on Stanford2D3D and TartanAir test sets in Fig. 6 and Fig. 7.

Additional Qualitative Results on Web Images. We show additional qualitative results on web images in Fig. 8 and

Fig. 9.

Qualitative Results on Fisheye Images. We show qualitative results of predicting Perspective Fields for fisheye images in Fig. 10. *Sliding Win.:* We take advantage of the local representation and use a sliding window inference technique for images that are out of our training distribution. We inference on small crops and aggregate the prediction for each pixel from overlapping windows. The results in Fig. 10 use a window of size $(0.5\text{img_height}) \times (0.5\text{img_width})$. This window slides along a 12×18 grid uniformly on the image and at each point predicts the Up-vectors and Latitude Map within the window. The final output for these values at each pixel is the mean of that pixels values in each window that it was a part of. *Fine-tune,* we show results af-

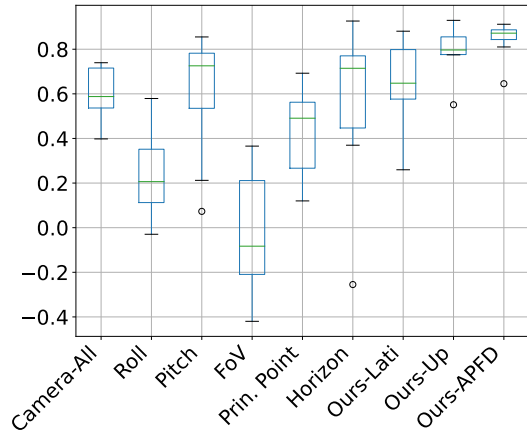


Figure 4. Pearson’s correlation for different metrics *w.r.t.* human perception. Our APFD metric has the highest correlation with human perception.

ter fine-tuning the PerspectiveNet on distorted images.

Additional Qualitative Results on Google Street View In Fig. 11 we show additional qualitative results from PerspectiveNet as well as Perspective Fields generated from the ParamNet predictions on GSV *uncentered principal-point* test set.

E. User Study Data Collection Interface

Fig. 12 shows the instruction users see and Fig. 13 is the interface users use when collecting human perceptual preferences.

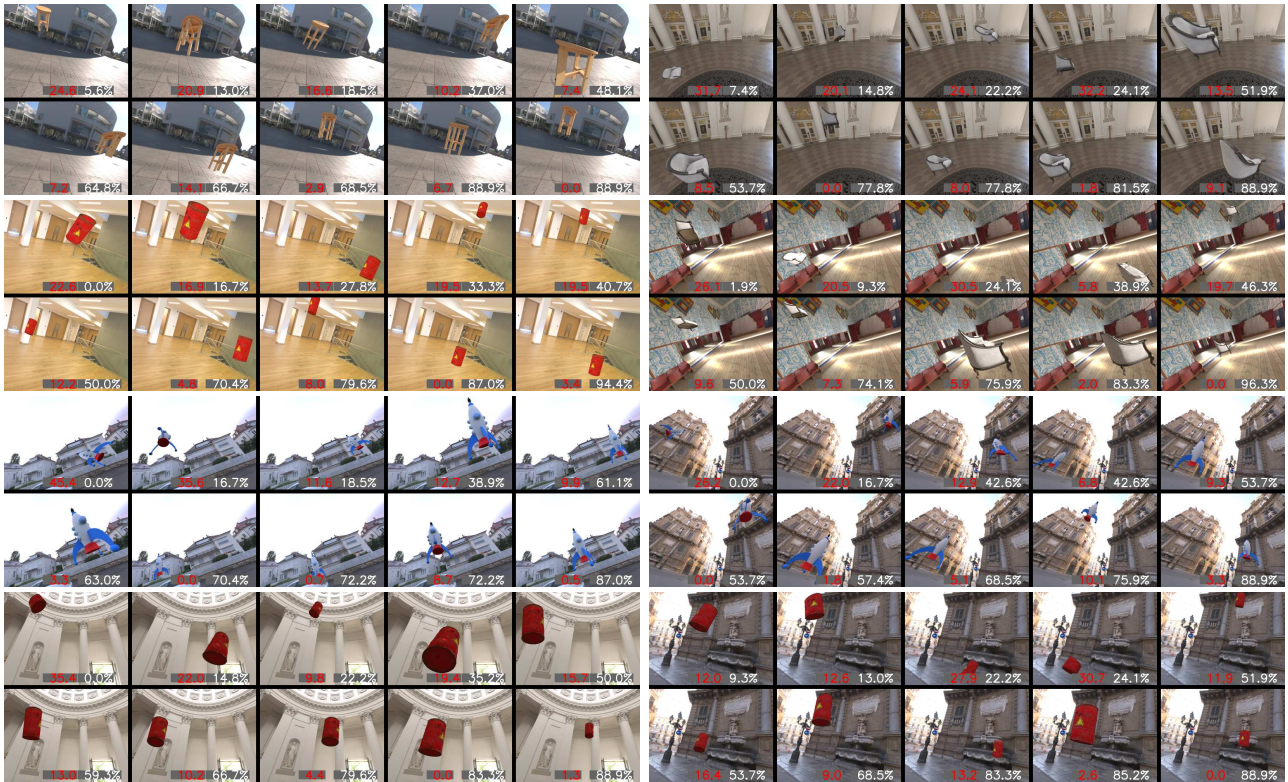


Figure 5. User study examples and results. Given a background image and an object, we randomly generate 10 compositing results with varied distortions. Pair-wise comparison is performed by a group of subjects. The white percentage number is the average winning rate based on human votes (the higher the better), and the red number is the APFD metric computed based on the Perspective Fields of the object and the background (the lower the better). There is a strong correlation between the perceptual quality and our metric.

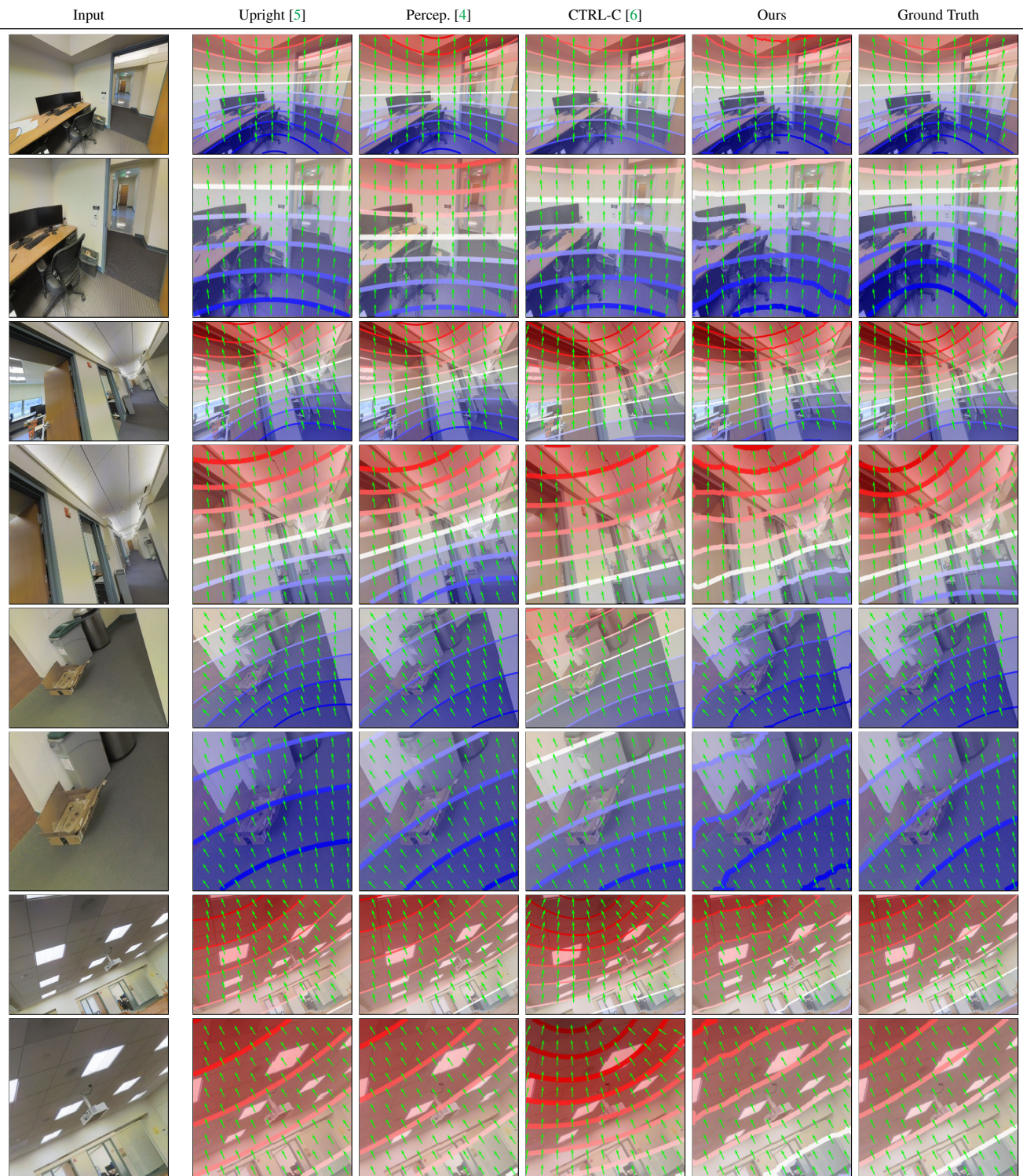


Figure 6. Comparison between baselines on Stanford2D3D dataset. Each test scene has two rows: the first row is the original image with a standard pin-hole camera perspective; the second row is a randomly cropped image. Up-vectors in the green vectors. Latitude colormap: $-\pi/2$ $\pi/2$.

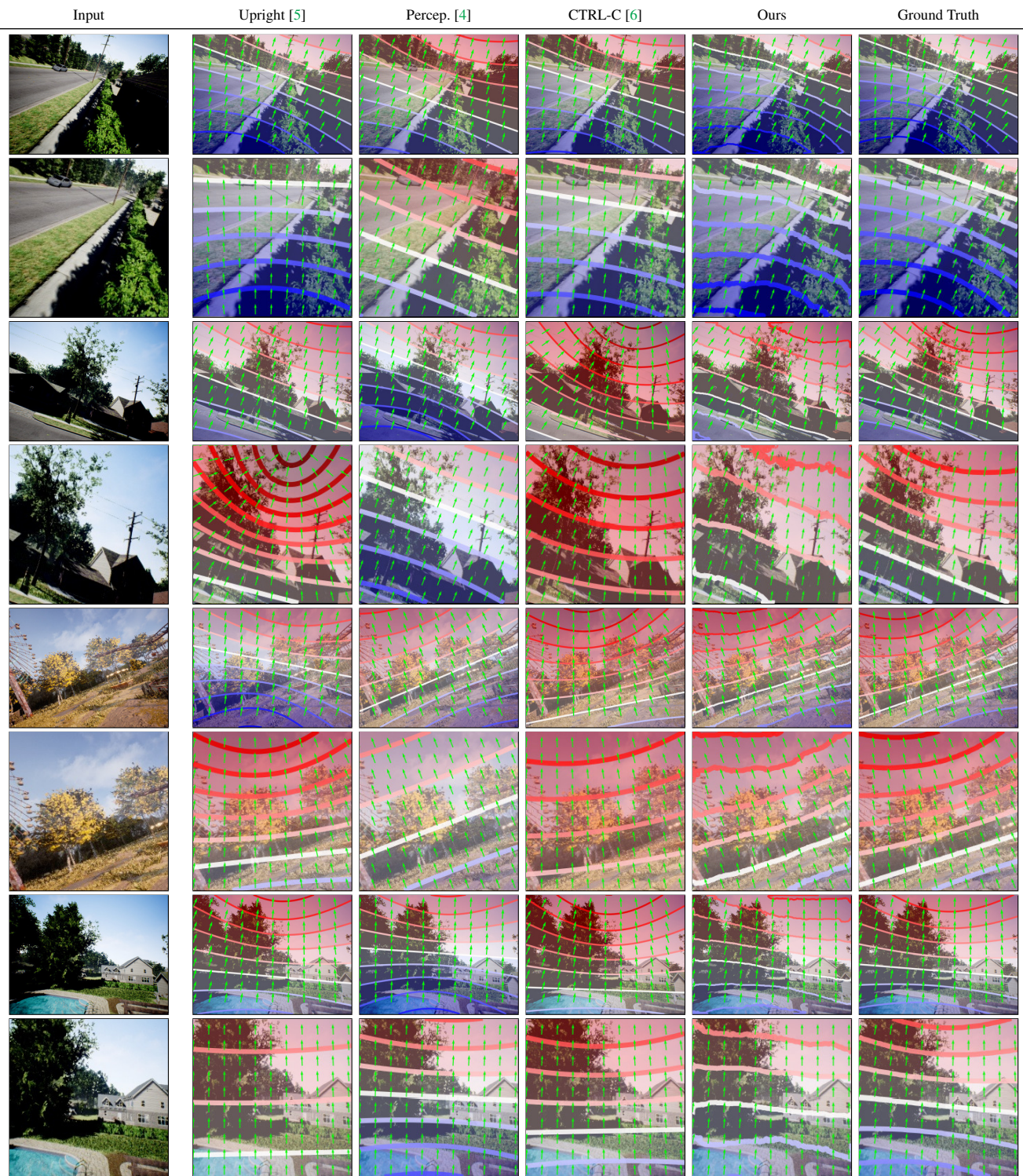


Figure 7. Comparison between baselines on TartanAir dataset. Each test scene has two rows: the first row is the original image with a standard pin-hole camera perspective; the second row is a randomly cropped image. Up-vectors in the green vectors. Latitude colormap: $-\pi/2$ $\pi/2$.

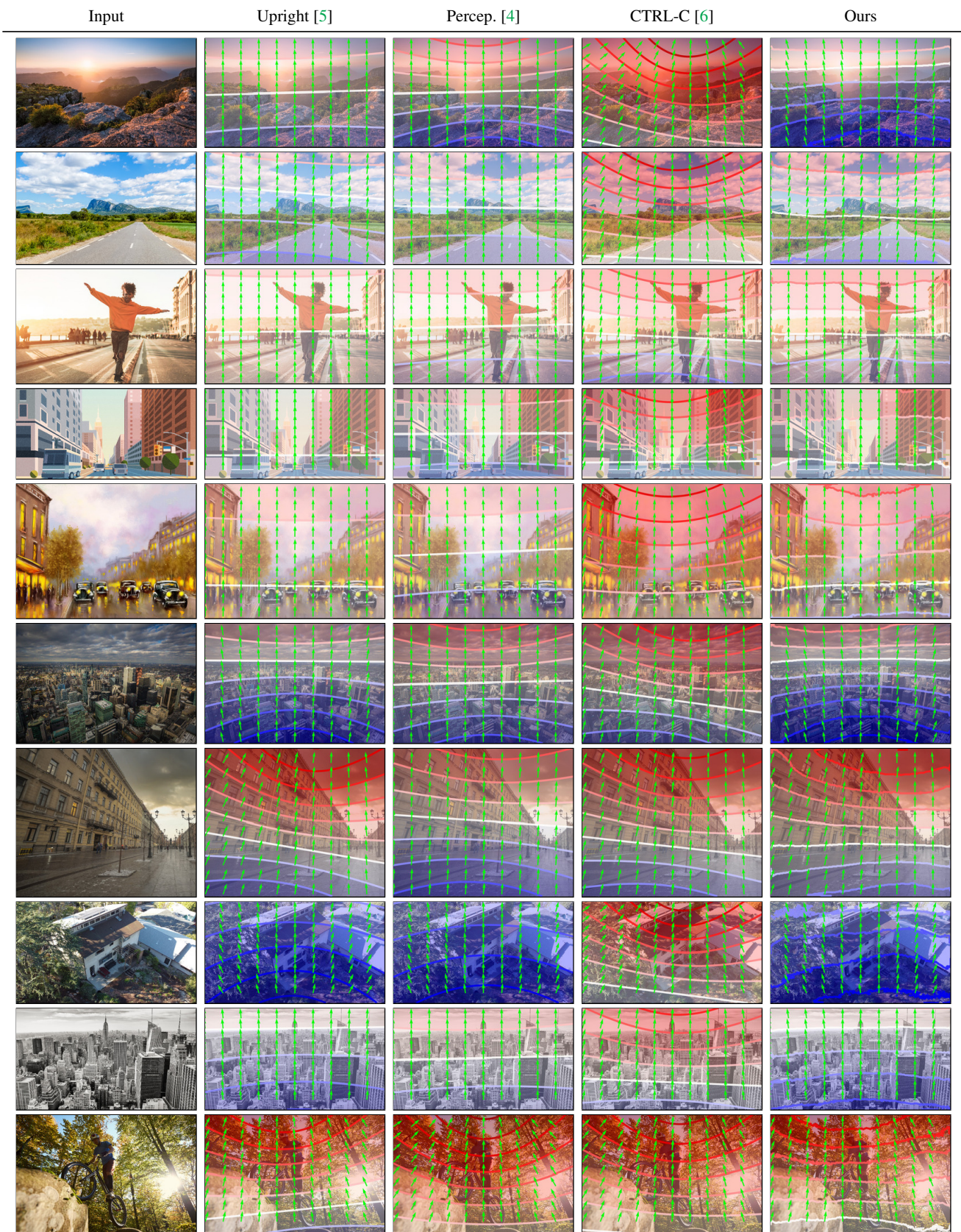


Figure 8. Additional qualitative results on web images, extending Fig. 5. Our approach produces better results compared to [5], [4], and [6]. There is no ground truth available.

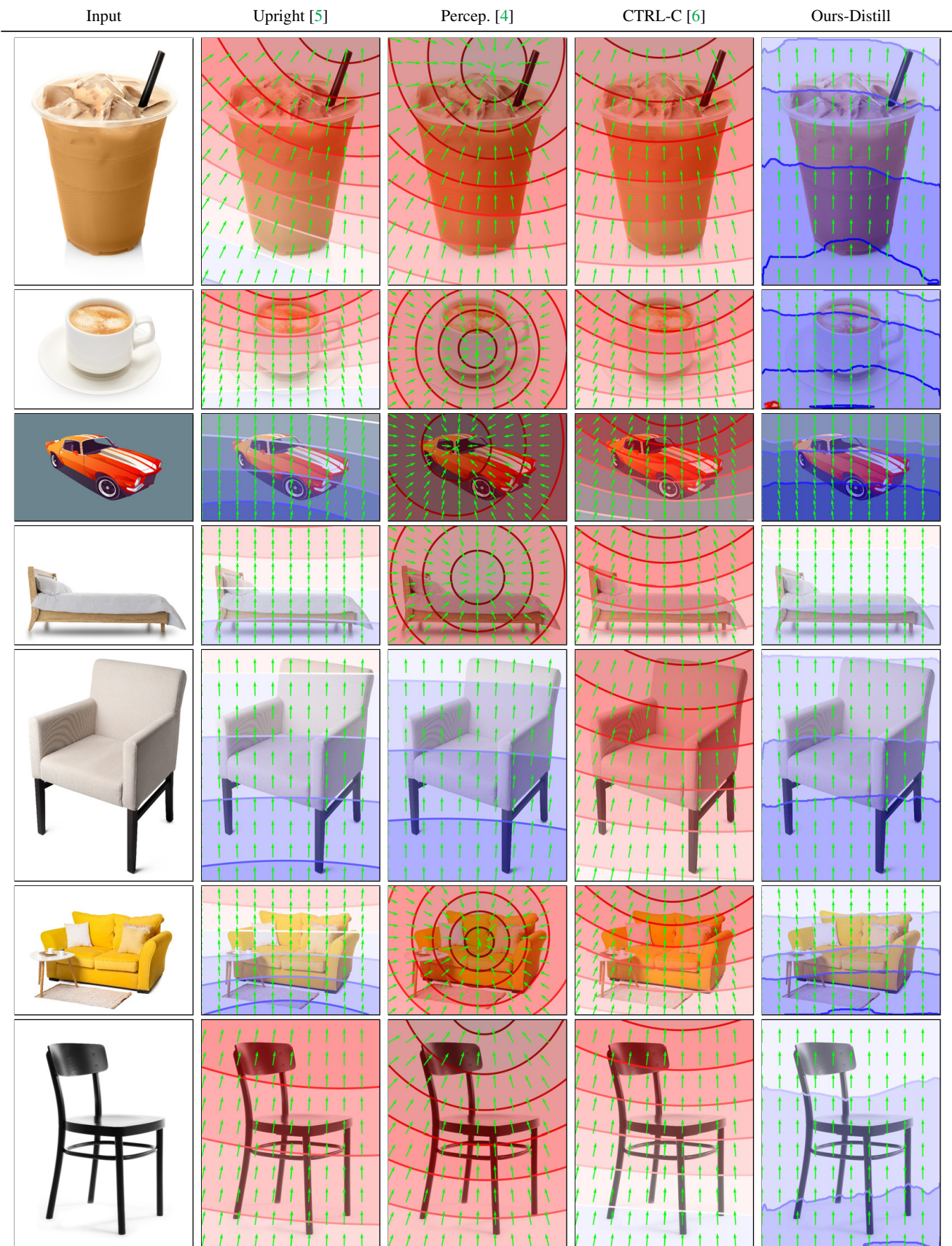


Figure 9. Additional qualitative results on web images, extending Fig. 5. Our approach produces better results compared to [5], [4], and [6].

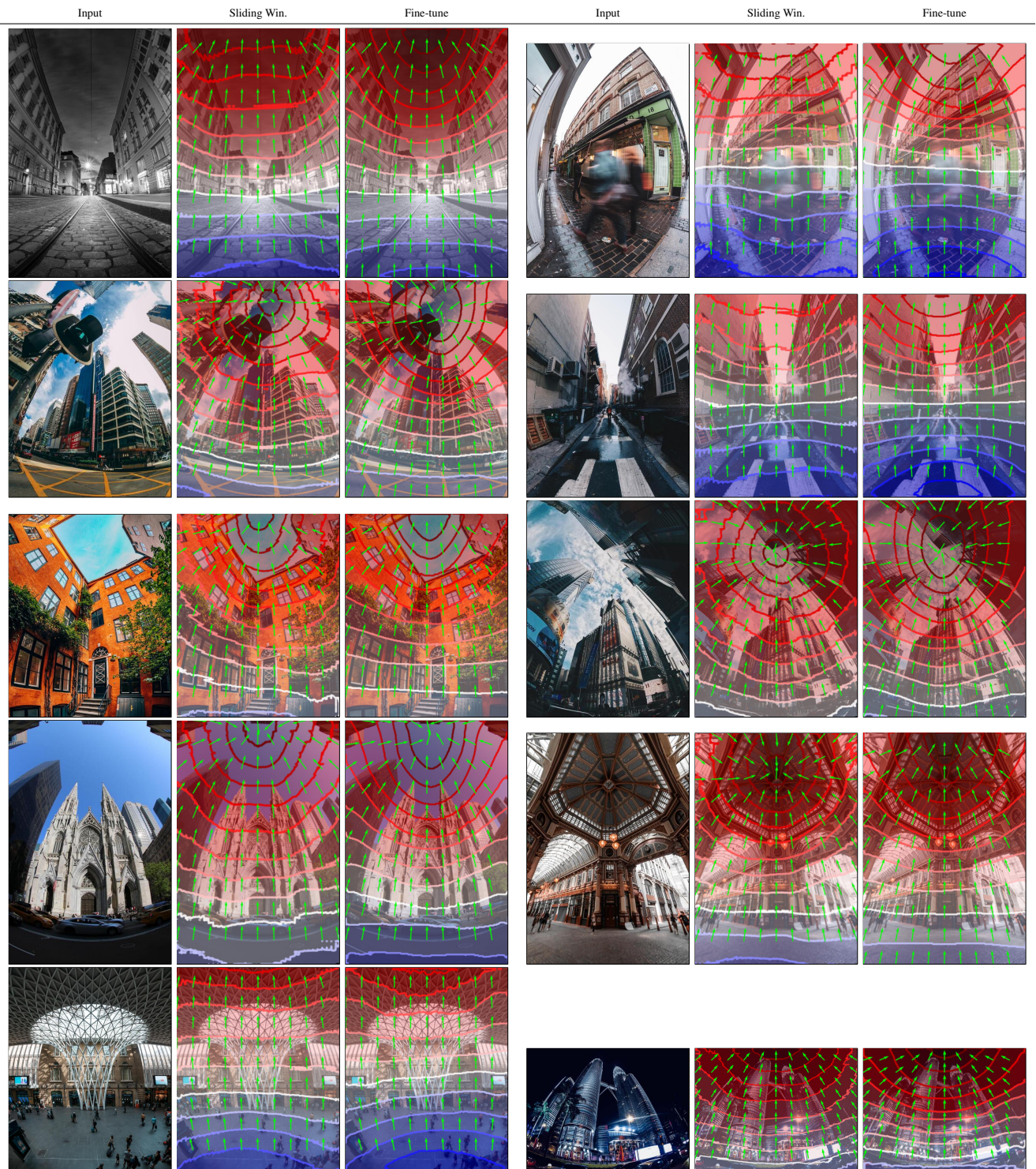




Figure 10. Qualitative results on fisheye images from the wild using both the sliding window and fine-tune techniques. Up-vectors in the green vectors. Latitude colormap: $-\pi/2$   $\pi/2$.

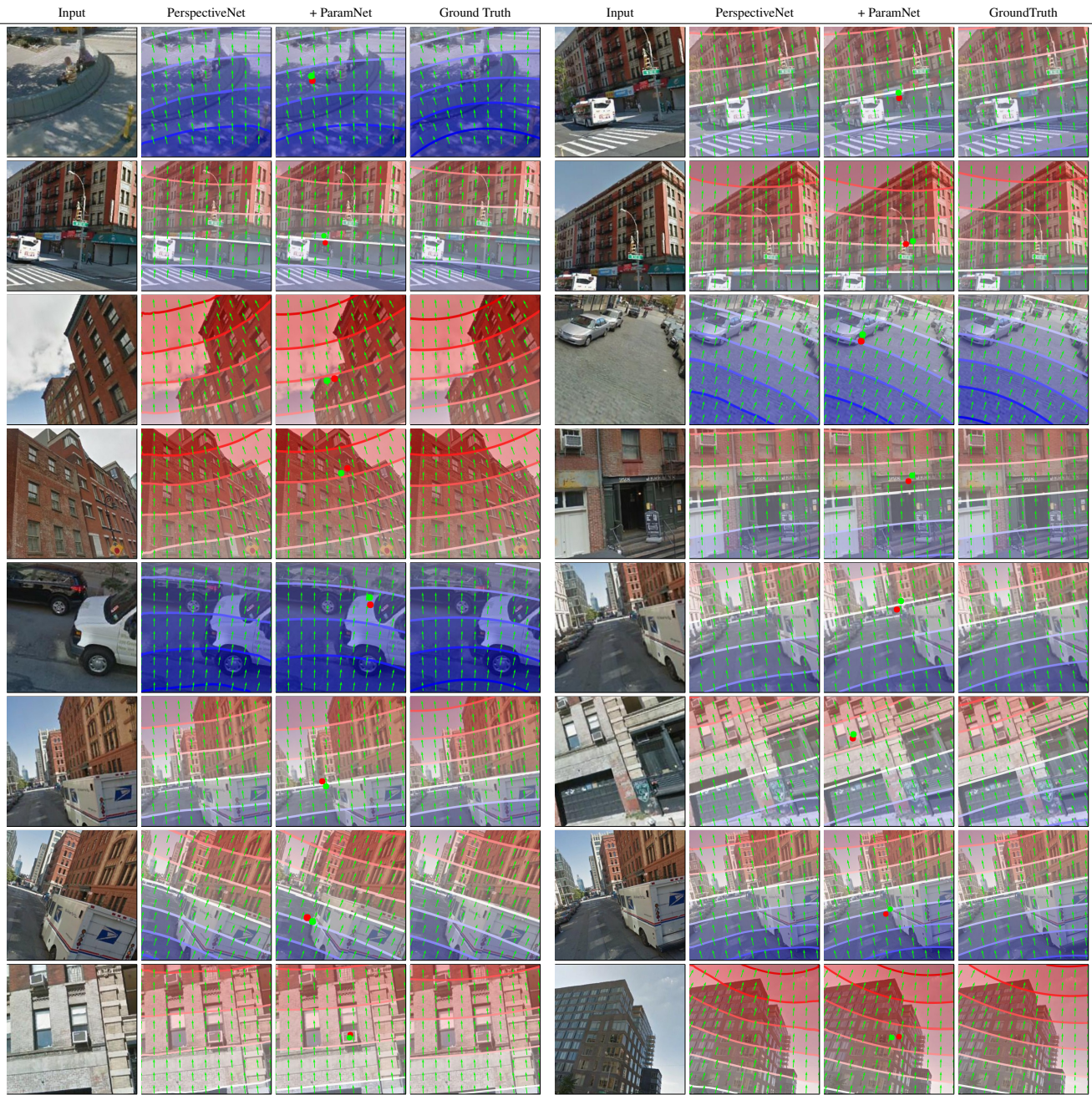



Figure 11. Additional qualitative results of PerspectiveNet and ParamNet on GSV *uncentered principal-point* images. In the ParamNet column, the ground truth principal point is indicated with a red dot and the predicted principle point is labeled with a green dot. Up-vectors in the green vectors. Latitude colormap: $-\pi/2$  $\pi/2$.

You will see a pair of images below. Each of them has a virtual object inserted in the scene. The virtual object is supposed to have an upright pose relative to the scene, but we purposefully add varying levels of distortion to the object. We would like you to decide which inserted object is **better** aligned with the rest of the scene based on the level of the object being tilted or distorted.

Below are some examples.

[Right] The background camera is looking from bottom to up. The rocket in the left image is captured as if the camera is looking horizontally. The rocket in the right image is captured as if the camera is looking up. So the right image is better.



[Left] We don't care about whether the chair should be on the ground. But we do want the chair to be aligned with the background trees since trees are upright.



[Right] Parallel lines should converge to one point. In this example, the side of the TV should be vertical in the world. The distortion of the left image does not match the background. The right image is more aligned.



Figure 12. Instructions users see before creating annotation.

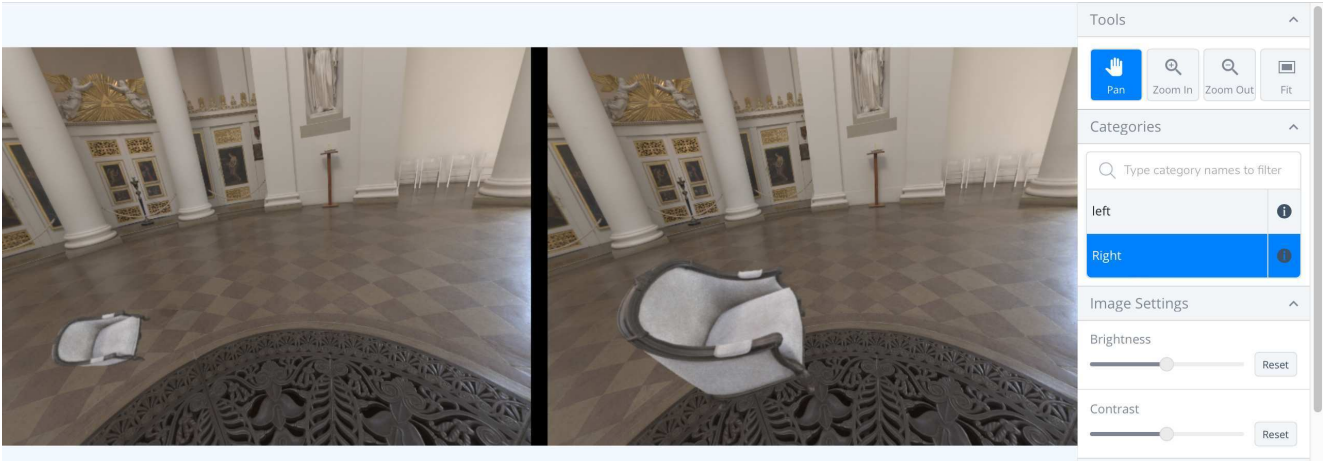


Figure 13. Example user study interface.

References

- [1] Adel Ahmadyan, Liangkai Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grundmann. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. *CVPR*, 2021. 3
- [2] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017. 2, 3
- [3] Alexander Buslaev, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. Alumentations: Fast and flexible image augmentations. 2020. 3
- [4] Yannick Hold-Geoffroy, Kalyan Sunkavalli, Jonathan Eisenmann, Matthew Fisher, Emiliano Gambaretto, Sunil Hadap, and Jean-François Lalonde. A perceptual measure for deep single image camera calibration. In *CVPR*, 2018. 1, 2, 3, 7, 8, 9, 10
- [5] Hyunjoon Lee, Eli Shechtman, Jue Wang, and Seungyong Lee. Automatic upright adjustment of photographs with robust camera calibration. *TPAMI*, 2014. 2, 7, 8, 9, 10
- [6] Jinwoo Lee, Hyunsung Go, Hyunjoon Lee, Sunghyun Cho, Minhyuk Sung, and Junho Kim. Ctrl-c: Camera calibration transformer with line-classification. In *ICCV*, 2021. 2, 3, 7, 8, 9, 10
- [7] Wenshan Wang, DeLong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. 2020. 2, 3
- [8] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, 2021. 2