Supplementary Material: Tangent Images for Mitigating Spherical Distortion

Marc Eder, Mykhailo Shvets, John Lim, and Jan-Michael Frahm
University of North Carolina at Chapel Hill
Chapel Hill, NC
{meder, mshvets, jlim13, jmf}@cs.unc.edu

In this supplementary material we provide the following additional information:

- Expanded discussion of some of the current limitations of tangent images (Section 1)
- The details of the camera normalization process and class-level results of our transfer learning experiments (Section 2)
- Class-level and qualitative results for the semantic segmentation experiments at different input resolutions (Section 3)
- Details of our 2D3DS Keypoints dataset along with individual image pair results and a qualitative comparison of select image pairs (Section 4)
- Training and architecture details for all CNN experiments detailed in this paper (Section 5)
- An example of a spherical image represented as tangent images (Figure 1)

1. Limitations

We have demonstrated the usefulness of our proposed tangent images, but we have also exposed some limitations of the formulation and opportunities for further research.

Level	**0**	**1**	**2**	**3**	**4**
Patch FOV	$\sim 73°$	$\sim 51°$	$\sim 31°$	$\sim 16°$	$\sim 8°$

Table 1: Tangent images FOV at different base levels. There is a slight dependence on input resolution at low levels due to smaller faces near the 12 icosahedral singular points, but this effect becomes negligible at higher levels.

Resolution When using tangent images, low angular resolution spherical data is processed on potentially low pixel resolution images. This can severely limit the receptive field of the networks, to which we attribute our poor performance on the level 5 semantic segmentation task, for example. However, this limitation is only notable in the context of the existing literature, because prior work has been restricted to low resolution spherical data, as shown in Figure 2 in the main paper. One viable solution is to incorporate the rendering into the convolution operation. In this way, we could regenerate the tangent images at every operation and effectively resolve the receptive field issue. However, as this is an issue for low resolution images, and our work is focused on addressing high resolution spherical data, we leave this modification for future study.

FOV The base subdivision level provides a constraint on the FOV of the tangent images. Table 1 shows the FOV of the tangent images at different base subdivision levels. As the FOV decreases, algorithms that rely on some sense of context or neighborhood break down. We observe this effect for both the CNN and keypoint experiments. While this is certainly a trade-off with tangent images, we have demonstrated that base levels and resolutions can be modulated to fit the required application. Another important point to observe regarding tangent image FOV is that the relationship between FOV and subdivision level does not hold perfectly at lower subdivision levels due the outsize influence of faces near the 12 singular points on the icosahedron. This effect largely disappears after base level 2, but when normalizing camera matrices to match spherical angular resolution at base levels 0 and 1, it is necessary to choose the right base level for the data. We use a base level of 1 in our transfer learning experiments on the OmniSYNTHIA dataset for this reason.

2. Network Transfer

In this section, we detail the camera normalization process used when training the network for transferring to spherical data. We also provide class-level results for our experiments.

2.1. Camera normalization

In order to ensure angular resolution conformity between perspective training data and spherical test data, we normalize the intrinsic camera matrices of our training images to match the angular resolution of the spherical inputs, α_s. To do this, we resample all perspective image inputs to a common camera with the desired angular resolution. The angular resolution in radians-per-pixel of an image, α_x and α_y, is defined as:

$$\alpha_x = \frac{\Omega_x}{W} \qquad \alpha_y = \frac{\Omega_y}{H} \tag{1}$$

where Ω_x and Ω_y are the fields of view of the image as a function of the image dimensions, W and H, and the focal lengths, f_x and f_y:

$$\begin{aligned} \Omega_x &= 2\arctan\left(\frac{W}{2f_x}\right) \\ \Omega_y &= 2\arctan\left(\frac{H}{2f_y}\right) \end{aligned} \tag{2}$$

Because spherical inputs have uniform angular resolution in every direction, we resample our perspective inputs likewise: $\alpha_x = \alpha_y = \alpha_s$.

Choosing camera properties For our camera-normalized perspective images, we want to choose fields of view, Ω'_x and Ω'_y, and image dimensions, W' and H' that satisfy:

$$\frac{\Omega'_x}{W'} = \frac{\Omega'_y}{H'} = \alpha_s \tag{3}$$

While there are a variety of options that we could use, we choose to set $\Omega'_x = \Omega'_y = \frac{\pi}{4}$ because $\frac{\pi}{4}$ radians ($45°$) is a reasonable field of view for a perspective image. We select W' and H' accordingly. For a level 8 input, this results in $W' = H' = 128$.

Normalizing intrinsics Recall the definition of the intrinsic matrix, K, given focal lengths f_x and f_y and principal point (c_x, c_y):

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

Given our choices for fields of view and image dimensions explained above, we compute a new, common intrinsic matrix. The new focal lengths, f'_x and f'_y, are computed as:

$$f'_x = \frac{W'}{2\tan\left(\frac{\Omega'_x}{2}\right)} \qquad f'_y = \frac{H'}{2\tan\left(\frac{\Omega'_y}{2}\right)} \tag{5}$$

and, for simplicity, the new principal point is chosen to be:

$$c'_x = \frac{W'}{2} \qquad c'_y = \frac{H'}{2} \tag{6}$$

Defining:

$$K' = \begin{bmatrix} f'_x & 0 & c'_x \\ 0 & f'_y & c'_y \\ 0 & 0 & 1 \end{bmatrix} \tag{7}$$

the camera intrinsics can be normalized using the relation:

$$x' = K'K^{-1}x \tag{8}$$

where x and x' are homogeneous pixel coordinates in the original and resampled images, respectively, and K^{-1} is the inverse of the intrinsic matrix associated with the original image.

Random shifts If we were to simply resample the image using Equation (8), we would end up with center crops of the original perspective images. In order to ensure that we do not discard useful information, we randomly shift the principle point of the original camera by some (δ_x, δ_y) before normalizing. This produces variations in where we crop the original image. Including this shift, we arrive at the formula we use for resampling the perspective training data:

$$x' = K'(K + \Delta)^{-1}x \tag{9}$$

where:

$$\Delta = \begin{bmatrix} 0 & 0 & \delta_x \\ 0 & 0 & \delta_y \\ 0 & 0 & 0 \end{bmatrix} \tag{10}$$

To ensure our crops stay within the bounds of the original image, we want:

$$\begin{aligned} \delta_x + P(0,0)_x &\geq 0 \\ \delta_y + P(0,0)_y &\geq 0 \\ \delta_x + P(W',H')_x &\leq W \\ \delta_y + P(W',H')_y &\leq H \end{aligned} \tag{11}$$

where $P(x',y')_{\{x,y\}}$ denotes the x- and y-dimensions of the new camera's coordinates projected into the original camera's coordinate system:

$$P(x',y') = KK'^{-1}\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \tag{12}$$

Using this constraint, we sample crops over the entire image by randomly choosing δ_x and δ_y from the ranges:

$$\begin{aligned} \frac{f_x}{f'_x}c'_x - c_x \leq \delta_x \leq W - c_x - \frac{f_x}{f'_x}(W' - c'_x) \\ \frac{f_y}{f'_y}c'_y - c_y \leq \delta_y \leq H - c_y - \frac{f_y}{f'_y}(H' - c'_y) \end{aligned} \tag{13}$$

2.2. Per-class results

Table 2 gives the per-class results for our semantic segmentation transfer experiment. While perspective image performance should be considered an upper bound on spherical performance, note that in some classes, we appear to actually perform better on the spherical image. This is because the spherical evaluation is done on equirectangular images in order to be commensurate across representations. This means that certain labels are duplicated and others are reduced due to distortion, which can skew the per-class results.

3. Semantic Segmentation

We provide the per-class quantitative results for our semantic segmentation experiments from Section 4.2 in the main paper. Additionally, we qualitatively analyze the benefits of training higher resolution networks, made possible by the tangent image representation.

3.1. Class results

Per-class results are given for semantic segmentation in Table 3. Nearly every class benefits from the high-resolution inputs facilitated by tangent images. This is especially noticeable for classes with fine-grained detail and smaller objects, like *chair*, *window*, and *bookcase*. The *table* class is an interesting example of the benefit of our method. While prior work has higher accuracy, our high resolution classification has significantly better IOU. In other words, our high resolution inputs may not result in correct classifications of every *table* pixel, but the classifications that are correct are much more precise. This increased precision is reflected almost across the board by mean IOU performance.

3.2. Qualitative results

Figure 2 gives 3 examples of semantic segmentation results at each resolution. The most obvious benefits of the higher resolution results are visible in the granularity of the output segmentation. Notice the fine detail preserved in the chairs in the level 10 output in the bottom row and even the doorway and whiteboard in the middle row. However, recall that our level 10 network uses a base level of 1. The effects of the smaller FOV of the tangent images are visible in the misclassifications of wall on the right of the level 10 output in the middle row. The level 5 network has no such problems classifying that surface, having been trained at a lower input resolution and using base level 0. Nevertheless, it is worth noting that large, homogeneous regions are going to be problematic for high resolution images, regardless of method, due to receptive field limitations of the network. If the region in question is larger than the receptive field of the network, there is no way for the network to infer context. As such, we are less concerned by errors on these large regions.

4. Stanford 2D3DS Keypoints Dataset

4.1. Details

Tables 4 and 5 give the details of the image pairs in our keypoints dataset. Tables 6 and 7 provide the individual metrics computed for each image pair.

4.2. Qualitative examples

We provide a qualitative comparison of keypoint detections in Figure 3. These images illustrate two interesting effects, in particular. First, in highly distorted regions of the image that have repeatable texture, like the floor in both images, detecting on the equirectangular format produces a number of redundant keypoints distributed over the distorted region. With tangent images, we see fewer redundant points as the base level increases and the ones that are detected are more accurate and robust, as indicated by the higher MS score. Additionally, the equirectangular representation results in more keypoint detections at larger scales. These outsize scales are an effect of distortion. Rotating the camera so that the corresponding keypoints are detected at different pixel locations with different distortion characteristics will produce a different scale, and consequently a difference descriptor. This demonstrates the need for translational equivariance in keypoint detection, which requires the lower distortion provided by our tangent images. This is reflected quantitatively by the higher PMR scores.

Figure 4 shows an example of inlier correspondences computed on the equirectangular images and at different base levels for an image pair from the hard split. Even though we detect fewer keypoints using tangent planes, we still have the same quality or better inlier correspondences. Distortion in the equirectangular format results in keypoint over-detection, which can potentially strain the subsequent inlier model fitting. Using tangent images, we detect fewer, but higher quality, samples. This results in more efficient and reliable RANSAC [2] model fitting. This is why tangent images perform noticeably better on the hard set, where there are fewer correspondences to be found.

5. Network Training Details

We detail the training parameters and network architectures used in our experiments to encourage reproducible research. All deep learning experiments were performed using the PyTorch library.

5.1. Shape classification

For the shape classification experiment, we use the network architecture from [4], replacing their MeshConv layers with 3×3 2D convolutions with unit padding. For downsampling operations, we bilinearly interpolate the tangent images. We first render the ModelNet40 [7] shapes to equirectangular images to be compatible with our tangent image generation implementation. The equirectangular image dimensions are 64×128, which is equivalent to the level 5 icosahedron. We train the network with a batch size of 16 and learning rate of $5 * 10^{-3}$ using the Adam optimizer [5].

5.2. Semantic segmentation

We use the residual U-Net-style architecture from [4, 8] for our semantic segmentation results at levels 5 and 7. Instead of MeshConv [4] and HexConv [8], we swap in 3×3

2D convolutions with unit padding. For the level 10 results, we use the fully-convolutional ResNet101 [3] pre-trained on COCO [6] provided in the PyTorch model zoo. We train and test on each of the standard folds of the Stanford 2D3DS dataset [1]. For all spherical data, evaluation metrics are computed *on the re-rendered spherical image*, not on the tangent images.

Level 5, 7 parameters For the level 5 and 7 experiments, our tangent images were base level 0 RGB-D images, and we use all 20 tangent images from each image in a batch of 8 spherical images, resulting in an effective batch size of 160 tangent images. We use Adam optimization [5] with an initial learning rate of 10^{-2} and decay by 0.9 every 20 epochs, as in [4].

Level 10 parameters For the level 10 experiments, we use RGB-D images at base level 1 and randomly sample 4 tangent images from each image in a batch of 4 spherical inputs, resulting in an effective batch size of 16 tangent images. Because the pre-trained network does not have a depth channel, we initialize the depth channel filter with zero weights. This has the effect of slowly adding the depth information to the model. Similarly, the last layer is randomly initialized, as Stanford 2D3DS has a different number of classes than COCO. We use Adam optimization [5] with a learning rate of 10^{-4}.

5.3. Transfer learning

We again use the fully-convolutional ResNet101 [3] architecture pre-trained on COCO [6]. We fine tune for 10 epochs on the perspective images of the Standford2D3DS dataset [1]. We use a batch size of 16 and a learning rate of 10^{-4}.

References

[1] 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. 4, 5, 7

[2] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 3

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 4

[4] Chiyu Max Jiang, Jingwei Huang, Karthik Kashinath, Prabhat, Philip Marcus, and Matthias Niessner. Spherical CNNs on unstructured grids. In *International Conference on Learning Representations*, 2019. 3, 4, 5

[5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015. 3, 4

[6] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014. 4

[7] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015. 3

[8] Chao Zhang, Stephan Liwicki, William Smith, and Roberto Cipolla. Orientation-aware semantic segmentation on icosahedron spheres. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3533–3541, 2019. 3, 5

mAcc													
Format	*beam*	*board*	*bookcase*	*ceiling*	*chair*	*clutter*	*column*	*door*	*floor*	*sofa*	*table*	*wall*	*window*
Persp.	16.1	66.5	65.9	79.4	61.0	56.5	30.1	45.4	80.3	35.5	67.5	61.6	57.9
Spher.	17.8	57.3	47.7	81.1	59.0	34.3	20.2	57.3	91.3	32.4	38.7	76.3	57.9
Ratio	1.11	0.86	0.72	1.02	0.97	0.61	0.67	1.26	1.14	0.91	0.57	1.24	1.00

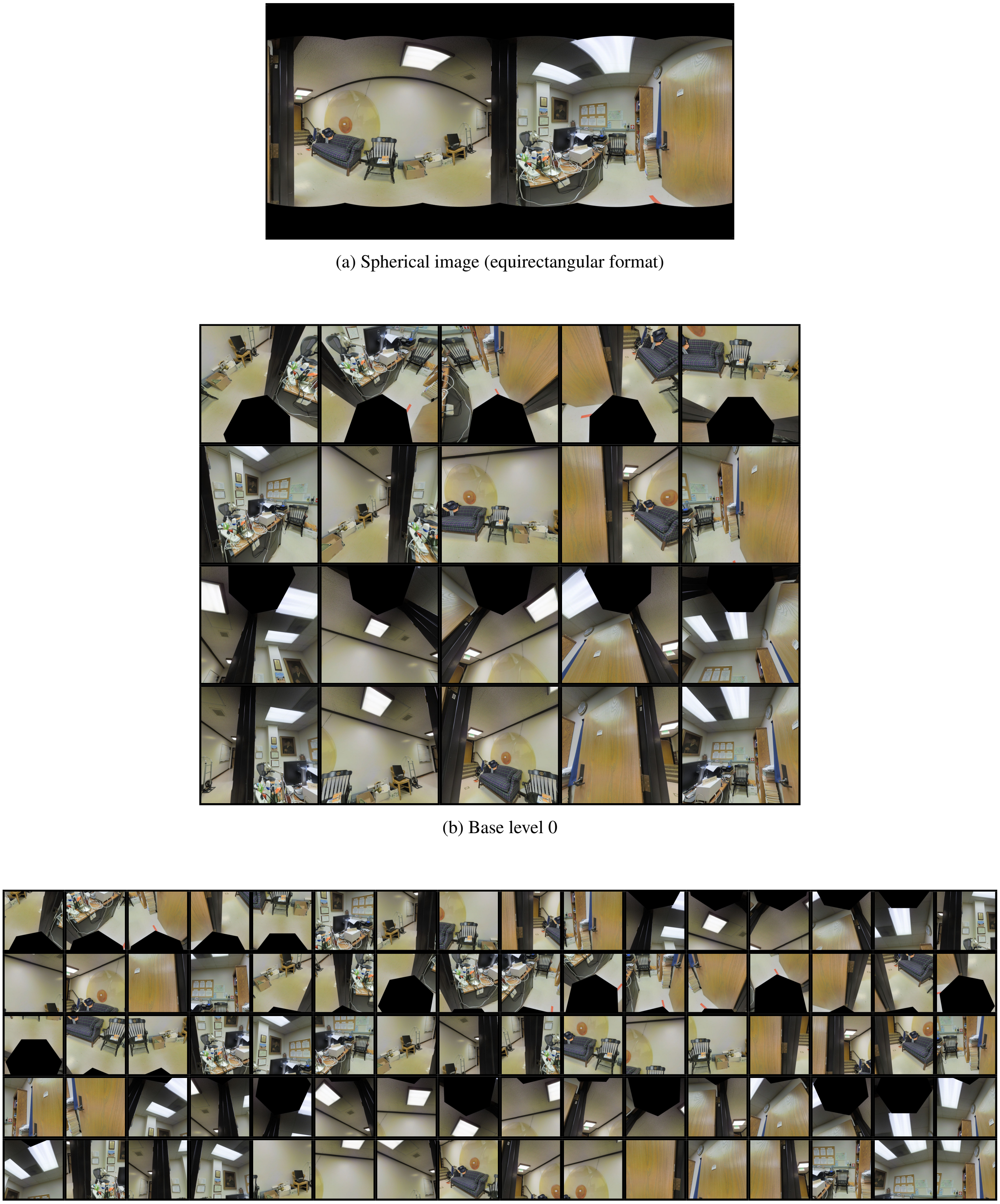
mIOU													
Format	*beam*	*board*	*bookcase*	*ceiling*	*chair*	*clutter*	*column*	*door*	*floor*	*sofa*	*table*	*wall*	*window*
Persp.	5.5	43.9	39.2	64.4	41.7	28.2	12.0	35.0	73.2	20.4	43.7	53.0	45.5
Spher.	8.6	47.4	37.4	64.0	43.6	24.4	10.1	34.3	51.1	20.9	33.4	54.6	40.8
Ratio	1.56	1.08	0.95	0.99	1.05	0.87	0.84	0.98	0.70	1.02	0.76	1.03	0.90

Table 2: Per-class results for the semantic segmentation transfer learning experiment on the Stanford 2D3DS dataset [1]. Ratio denotes the performance ratio between the network transferred to the spherical test set and the network evaluated on the perspective image test data.

mAcc													
Method	*beam*	*board*	*bookcase*	*ceiling*	*chair*	*clutter*	*column*	*door*	*floor*	*sofa*	*table*	*wall*	*window*
Jiang *et al*. [4]	19.6	48.6	49.6	93.6	63.8	43.1	28.0	63.2	96.4	21.0	70.0	74.6	39.0
Zhang *et al*. [8]	23.2	56.5	62.1	**94.6**	66.7	41.5	18.3	64.5	96.2	41.1	**79.7**	77.2	41.1
Ours L5	25.6	33.6	44.3	87.6	51.5	44.6	12.1	64.6	93.6	26.2	47.2	78.7	42.7
Ours L7	**29.7**	45.0	49.7	88.1	64.0	54.1	15.7	71.3	94.3	15.8	57.8	77.9	58.0
Ours L10	22.6	**62.0**	**70.0**	90.3	**84.7**	**55.5**	**41.4**	**76.7**	**96.9**	**70.3**	73.9	**80.1**	**74.3**

mIOU													
Method	*beam*	*board*	*bookcase*	*ceiling*	*chair*	*clutter*	*column*	*door*	*floor*	*sofa*	*table*	*wall*	*window*
Jiang *et al*. [4]	8.7	32.7	33.4	82.2	42.0	25.6	10.1	41.6	87.0	7.6	41.7	61.7	23.5
Zhang *et al*. [8]	10.9	39.7	37.2	84.8	50.5	29.2	11.5	45.3	92.9	19.1	49.1	63.8	29.4
Ours L5	10.9	26.6	31.9	82.0	38.5	29.3	5.9	36.2	89.4	12.6	40.4	56.5	26.7
Ours L7	**13.8**	37.4	37.0	84.0	42.9	35.1	7.8	41.8	90.6	11.6	48.0	62.7	36.5
Ours L10	4.5	**49.9**	**50.3**	**85.5**	**71.5**	**42.4**	**11.7**	**50.0**	**94.3**	**32.1**	**61.4**	**70.5**	**50.0**

Table 3: Per-class results for RGB-D inputs on the Stanford 2D3DS dataset [1]



(a) Spherical image (equirectangular format)

(b) Base level 0

(c) Base level 1

Figure 1: Example of tangent images at base levels 0 and 1.

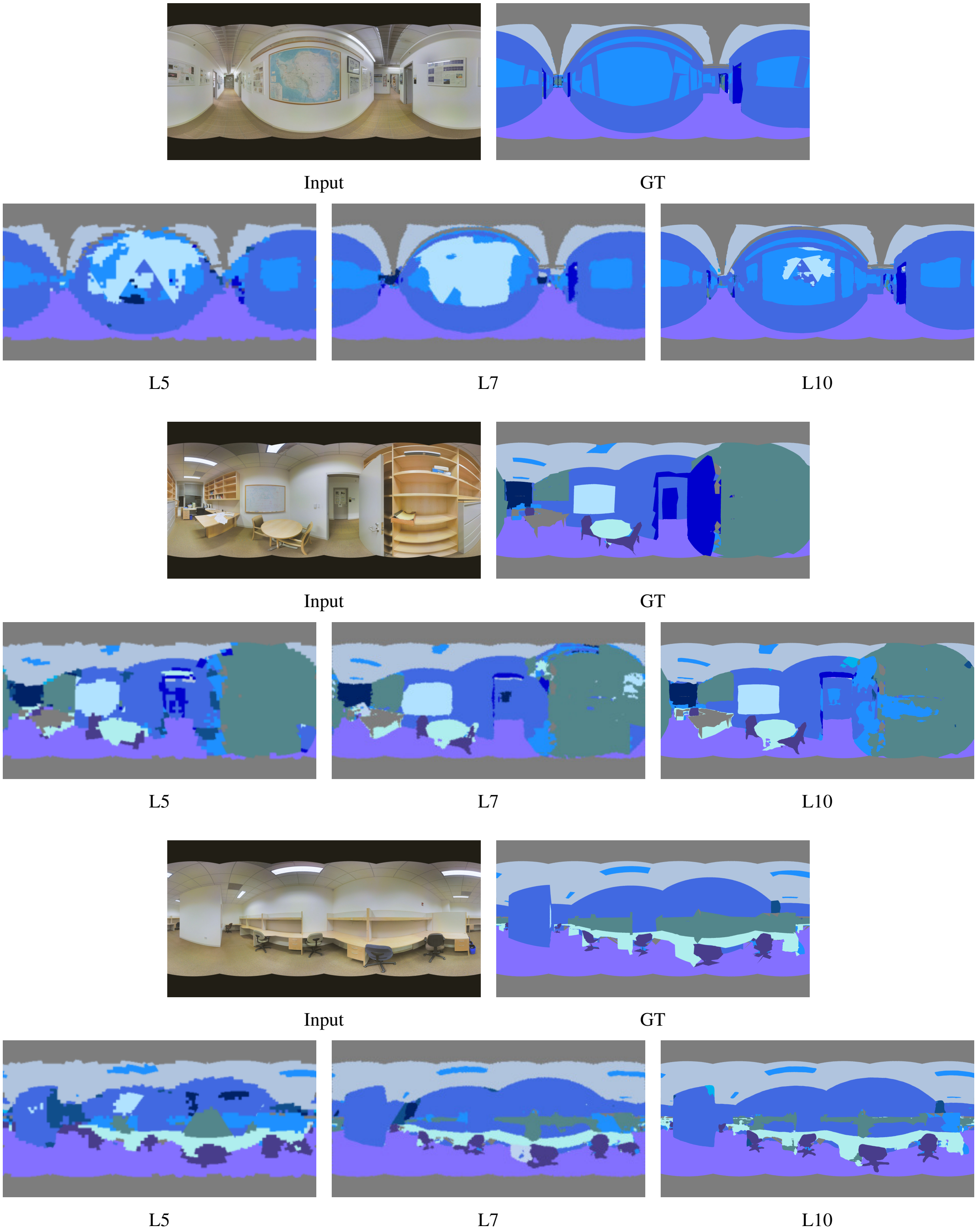


Figure 2: Qualitative results of semantic segmentation on the Stanford 2D3DS dataset [1] at different input resolutions. These results illustrate the performance gains we access by being able to scale to high resolution spherical inputs.

Easy			
Pair ID	*Left Image*	*Right Image*	*FOV Overlap*
1	0f65e09_hallway_7	1ebdfef_hallway_7	0.87
2	08a99a5_hallway_7	251a331_hallway_7	0.97
3	08a99a5_hallway_7	f7c6c2a_hallway_7	0.89
4	251a331_hallway_7	f7c6c2a_hallway_7	0.87
5	251a331_hallway_7	b261c3b_hallway_7	0.97
10	f7c6c2a_hallway_7	b261c3b_hallway_7	0.89
20	9178f6a_hallway_6	29abbc1_hallway_6	0.89
23	bc12865_hallway_6	7d58331_hallway_6	0.88
24	ee20957_hallway_6	bed890d_hallway_6	0.86
25	ee20957_hallway_6	eaba8c8_hallway_6	1.00
28	bed890d_hallway_6	eaba8c8_hallway_6	0.86
29	077f181_hallway_6	83baa70_hallway_6	0.86
30	97ab30c_hallway_6	eaba8c8_hallway_6	0.86
31	fc19236_office_18	e7d9e58_office_18	0.88
34	09ad38a_office_26	04a59ce_office_26	0.96
35	04a59ce_office_26	c16a90f_office_26	0.96
37	c40ca55_office_31	7b74e08_office_31	0.87
39	4a41b27_office_31	7b74e08_office_31	0.87
43	5512025_office_23	7f04c9b_office_23	0.92
44	5512025_office_23	5a18aa0_office_23	0.86
45	7f04c9b_office_23	5a18aa0_office_23	0.87
47	433548f_hallway_3	dcab252_hallway_3	0.91
49	d31d981_office_8	54e6de3_office_8	0.89
50	f85a909_office_3	c9feabc_office_3	0.88
51	f85a909_office_3	97be01e_office_3	0.89
52	c9feabc_office_3	97be01e_office_3	0.90
54	8fd8146_office_10	ab03f88_office_10	0.88
55	7c870c2_hallway_8	4de69cf_hallway_8	0.87
56	33e598f_office_15	8910cb1_office_15	0.88
58	46b4538_office_1	db2e53f_office_1	0.92

Table 4: Easy split of Stanford2D3DS keypoints dataset image pairs.

Hard			
Pair ID	*Left Image*	*Right Image*	*FOV Overlap*
0	c14611b_hallway_7	37a4f42_hallway_7	0.83
6	1b253d2_hallway_7	6e945c8_hallway_7	0.84
7	5d3a59a_hallway_7	ec0b9b3_hallway_7	0.81
8	ac01e35_hallway_7	649838b_hallway_7	0.85
9	f6c6ce3_hallway_7	5221e31_hallway_7	0.85
11	438c5fb_hallway_7	ec0b9b3_hallway_7	0.82
12	ec0b9b3_hallway_7	531efee_hallway_7	0.85
13	724bbea_hallway_7	c8c806b_hallway_7	0.85
14	724bbea_hallway_7	55db392_hallway_7	0.82
15	32d9e73_hallway_7	55db392_hallway_7	0.85
16	fcd2380_office_22	2d842ce_office_22	0.85
17	2d842ce_office_22	ffd2cca_office_22	0.86
18	89d9828_hallway_6	87e6e35_hallway_6	0.81
19	89d9828_hallway_6	7d58331_hallway_6	0.84
21	75acaa8_hallway_6	87e6e35_hallway_6	0.84
22	92b146f_hallway_6	8c78856_hallway_6	0.86
26	b640b47_hallway_6	87e6e35_hallway_6	0.80
27	bed890d_hallway_6	97ab30c_hallway_6	0.85
32	af50002_WC_1	36dd48f_WC_1	0.84
33	1edba7e_WC_1	e0c041d_WC_1	0.84
36	c40ca55_office_31	a77fba5_office_31	0.85
38	4a41b27_office_31	da4629d_office_31	0.82
40	da4629d_office_31	9084f21_office_31	0.84
41	75361af_office_31	ecf7fb4_office_31	0.82
42	2100dd9_office_4	26c24c7_office_4	0.83
46	84cdc9a_conferenceRoom_1	0d600f9_conferenceRoom_1	0.83
48	dcab252_hallway_3	a9cda4d_hallway_3	0.82
53	6549526_office_21	08aa476_office_21	0.83
57	dbcdb33_office_20	f02c98c_office_20	0.83
59	24f42d6_hallway_5	684b940_hallway_5	0.84

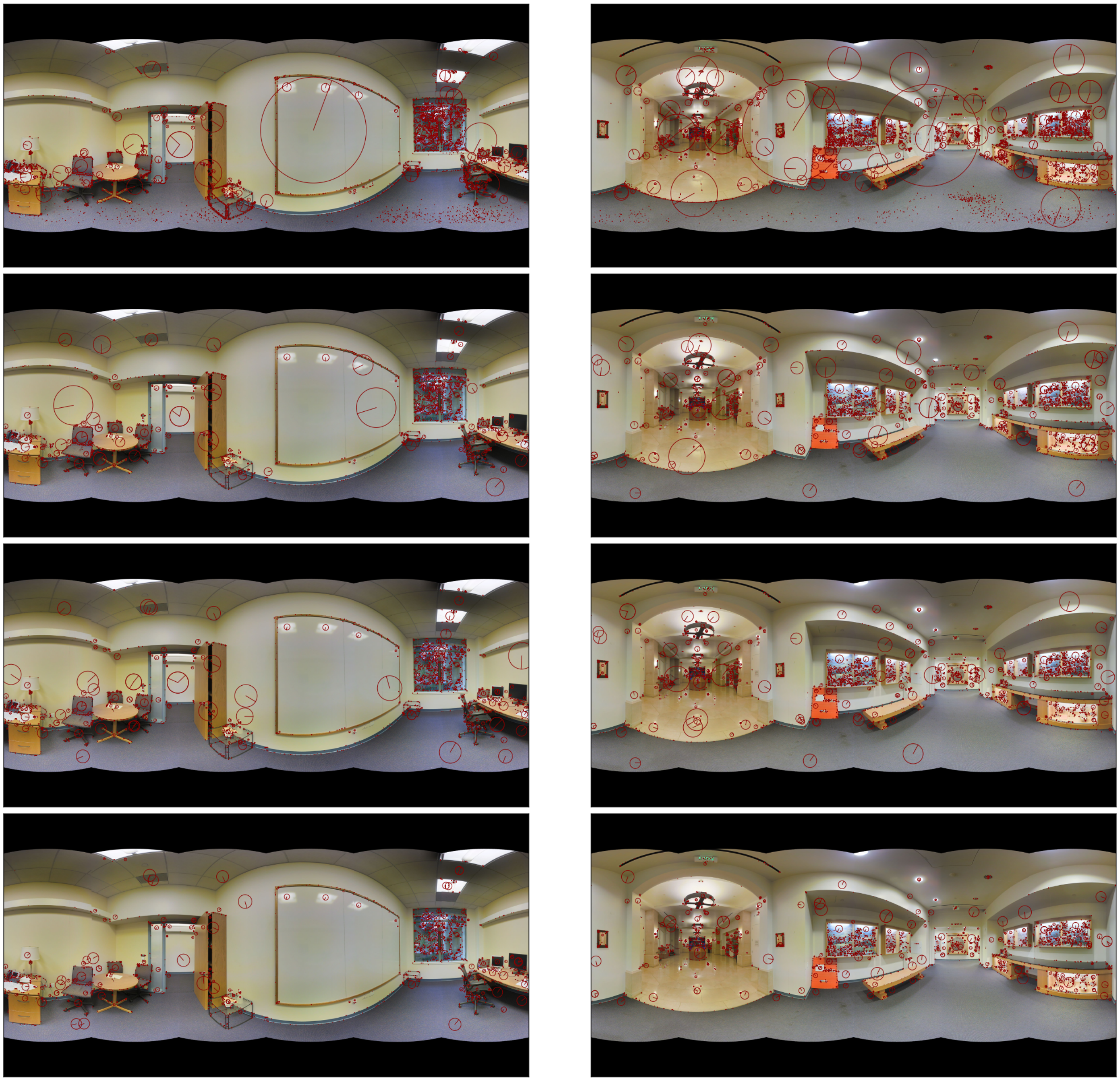
Table 5: Hard split of Stanford2D3DS keypoints dataset image pairs.

Easy												
Pair ID	**Equirect.**			**L0**			**L1**			**L2**		
	PMR	*MS*	*P*	*PMR*	*MS*	*P*	*PMR*	*MS*	*P*	*PMR*	*MS*	*P*
1	0.27	0.07	0.27	0.35	0.09	0.26	**0.40**	**0.13**	**0.33**	0.34	0.10	0.28
2	0.41	0.17	0.42	0.45	0.27	**0.60**	**0.50**	**0.28**	0.57	0.48	**0.28**	0.58
3	0.34	0.19	0.55	0.40	0.23	0.56	**0.46**	**0.27**	**0.60**	0.44	0.25	0.58
4	0.21	0.08	0.36	0.27	0.11	0.41	**0.30**	**0.14**	**0.48**	0.25	0.12	**0.48**
5	0.59	**0.48**	**0.80**	0.67	0.47	0.70	**0.70**	0.45	0.64	0.66	0.35	0.54
10	0.29	0.10	0.35	0.35	0.12	0.35	**0.36**	**0.14**	**0.38**	0.34	0.11	0.33
20	0.30	0.09	0.30	0.46	0.14	0.32	**0.50**	**0.19**	**0.38**	0.43	0.14	0.33
23	0.23	0.07	0.32	0.33	0.13	0.39	**0.34**	**0.14**	**0.42**	**0.34**	0.13	0.38
24	0.16	0.07	0.43	0.22	0.09	0.42	**0.24**	**0.10**	0.43	0.23	**0.10**	**0.44**
25	0.83	0.48	0.58	0.99	**0.71**	**0.72**	**1.01**	0.58	0.58	0.95	0.58	0.61
28	0.18	0.06	0.34	**0.25**	0.08	0.33	0.24	**0.10**	**0.40**	0.24	**0.10**	**0.40**
29	0.29	0.13	0.46	0.35	0.16	0.46	**0.38**	**0.21**	**0.55**	0.34	0.17	0.51
30	0.17	0.07	**0.42**	**0.25**	**0.08**	0.33	0.24	**0.08**	0.32	0.21	**0.08**	0.37
31	0.14	0.06	0.47	0.15	0.07	0.48	**0.16**	**0.08**	**0.51**	0.14	0.07	**0.51**
34	0.46	0.39	**0.86**	0.48	0.38	0.78	0.51	0.42	0.82	**0.52**	**0.43**	0.84
35	0.42	0.34	**0.82**	0.43	0.33	0.77	0.46	**0.38**	**0.82**	**0.47**	**0.38**	**0.82**
37	0.23	0.09	0.40	0.24	0.10	0.42	**0.25**	**0.12**	**0.47**	**0.25**	0.11	0.43
39	0.22	0.08	**0.39**	0.23	0.08	0.34	**0.24**	**0.09**	0.36	0.23	0.07	0.32
43	0.27	0.19	**0.70**	0.38	0.24	0.63	**0.39**	**0.25**	0.63	0.37	0.24	0.65
44	0.13	0.04	0.30	0.18	0.07	**0.39**	**0.23**	**0.08**	0.36	0.16	0.06	0.36
45	0.14	0.05	0.36	0.17	0.07	0.40	**0.21**	**0.09**	**0.44**	0.17	0.06	0.39
47	0.31	0.21	**0.67**	0.40	**0.25**	0.64	**0.42**	0.22	0.52	0.36	0.21	0.57
49	0.10	0.04	0.41	**0.15**	0.05	0.36	**0.15**	**0.06**	0.37	**0.15**	**0.06**	**0.42**
50	0.18	0.08	0.46	0.21	**0.11**	0.50	**0.23**	**0.11**	0.47	0.22	**0.11**	**0.52**
51	0.15	0.04	0.31	0.19	0.06	0.32	**0.22**	**0.07**	0.31	0.19	**0.07**	**0.38**
52	0.15	0.05	0.32	0.18	0.06	0.35	**0.19**	**0.08**	**0.39**	0.18	0.06	0.34
54	0.17	0.05	0.31	0.24	**0.09**	0.35	**0.25**	0.08	0.33	0.22	0.08	**0.38**
55	0.18	0.10	**0.53**	0.24	0.11	0.45	**0.26**	**0.13**	0.49	0.22	0.07	0.32
56	0.22	0.11	**0.50**	0.32	**0.16**	**0.50**	**0.33**	**0.16**	0.48	0.29	0.14	0.49
58	0.16	0.06	0.37	0.19	0.07	0.37	**0.22**	**0.09**	**0.40**	0.20	0.08	0.38

Table 6: Keypoint matching results on individual image pairs in the easy split.

Hard												
Pair ID	**Equirect.**			**L0**			**L1**			**L2**		
	PMR	*MS*	*P*	*PMR*	*MS*	*P*	*PMR*	*MS*	*P*	*PMR*	*MS*	*P*
0	0.25	0.12	**0.51**	0.28	0.14	0.50	**0.30**	**0.15**	0.49	**0.30**	**0.15**	0.49
6	0.26	0.09	0.33	0.35	0.15	**0.43**	**0.38**	**0.16**	**0.43**	0.34	0.14	0.40
7	0.22	0.07	0.32	0.28	0.09	**0.34**	**0.29**	0.09	0.32	**0.29**	**0.10**	**0.34**
8	0.23	0.08	0.38	**0.30**	**0.14**	**0.46**	**0.30**	0.11	0.37	0.29	0.11	0.39
9	0.31	0.08	0.26	**0.42**	0.12	0.28	**0.42**	**0.13**	**0.32**	0.39	0.11	0.27
11	0.23	0.07	0.32	0.28	**0.11**	**0.41**	**0.32**	0.08	0.25	0.27	0.08	0.29
12	0.20	0.05	0.24	**0.34**	**0.09**	0.26	0.33	0.07	0.21	0.29	**0.09**	**0.32**
13	0.24	0.07	0.30	0.35	0.10	0.29	**0.37**	**0.11**	0.30	0.31	0.10	**0.31**
14	0.26	0.08	0.32	0.43	0.10	0.24	**0.50**	**0.17**	**0.33**	0.40	0.12	0.30
15	0.30	0.12	0.40	0.40	0.19	0.47	**0.42**	**0.21**	**0.51**	0.36	0.17	0.49
16	0.16	0.05	0.34	**0.19**	0.06	0.35	**0.19**	**0.07**	**0.37**	0.17	0.06	0.36
17	0.19	0.09	0.47	0.21	0.11	**0.51**	**0.24**	**0.12**	0.49	0.21	0.11	**0.51**
18	0.24	0.06	0.26	**0.36**	**0.12**	0.33	**0.36**	0.10	0.28	0.31	**0.12**	**0.38**
19	0.20	0.06	0.28	0.31	0.09	0.29	**0.34**	**0.12**	0.34	0.28	**0.12**	**0.42**
21	0.22	0.08	0.35	0.30	0.11	0.37	**0.31**	**0.12**	**0.38**	0.29	0.10	0.34
22	0.25	0.07	0.29	0.35	0.12	0.35	**0.36**	**0.13**	0.37	0.33	**0.13**	**0.41**
26	0.21	0.06	**0.31**	0.31	**0.10**	**0.31**	**0.33**	**0.10**	0.30	0.29	0.08	0.29
27	0.16	0.06	0.37	0.24	0.11	0.46	**0.25**	**0.12**	**0.48**	0.22	0.10	0.46
32	0.25	0.09	0.37	0.30	0.12	0.39	**0.34**	**0.15**	**0.43**	0.30	0.12	0.39
33	0.19	0.09	0.49	0.24	0.12	0.50	0.25	0.13	0.51	**0.26**	**0.14**	**0.53**
36	0.23	0.10	0.42	0.25	**0.11**	**0.44**	**0.26**	**0.11**	**0.44**	0.25	0.10	0.42
38	0.22	**0.09**	0.39	**0.23**	0.08	0.37	**0.23**	**0.09**	0.37	**0.23**	**0.09**	**0.41**
40	0.20	0.10	0.50	0.22	**0.11**	0.51	**0.23**	**0.11**	0.50	0.22	**0.11**	**0.52**
41	0.23	0.12	0.52	0.25	0.14	0.54	0.26	0.14	0.54	**0.27**	**0.15**	**0.56**
42	0.17	0.05	0.30	**0.21**	0.08	0.37	**0.21**	**0.09**	0.41	0.20	0.08	**0.42**
46	0.21	**0.10**	**0.50**	**0.25**	**0.10**	0.40	**0.25**	**0.10**	0.39	0.24	0.08	0.35
48	0.27	0.08	0.29	**0.32**	0.12	0.38	**0.32**	**0.13**	0.41	0.29	**0.13**	**0.44**
53	0.15	0.05	**0.33**	0.15	0.05	0.32	**0.17**	**0.06**	0.32	0.15	0.04	0.26
57	0.17	0.07	0.43	**0.20**	**0.10**	0.47	**0.20**	**0.10**	**0.51**	**0.20**	**0.10**	0.49
59	0.26	0.13	0.50	0.28	0.15	0.53	0.27	0.15	0.53	**0.29**	**0.16**	**0.54**

Table 7: Keypoint matching results on individual image pairs in the hard split.



(a) From image pair 58

(b) From image pair 33

Figure 3: Comparison of SIFT keypoint detections. Each column, top to bottom: equirectangular, L0, L1, L2

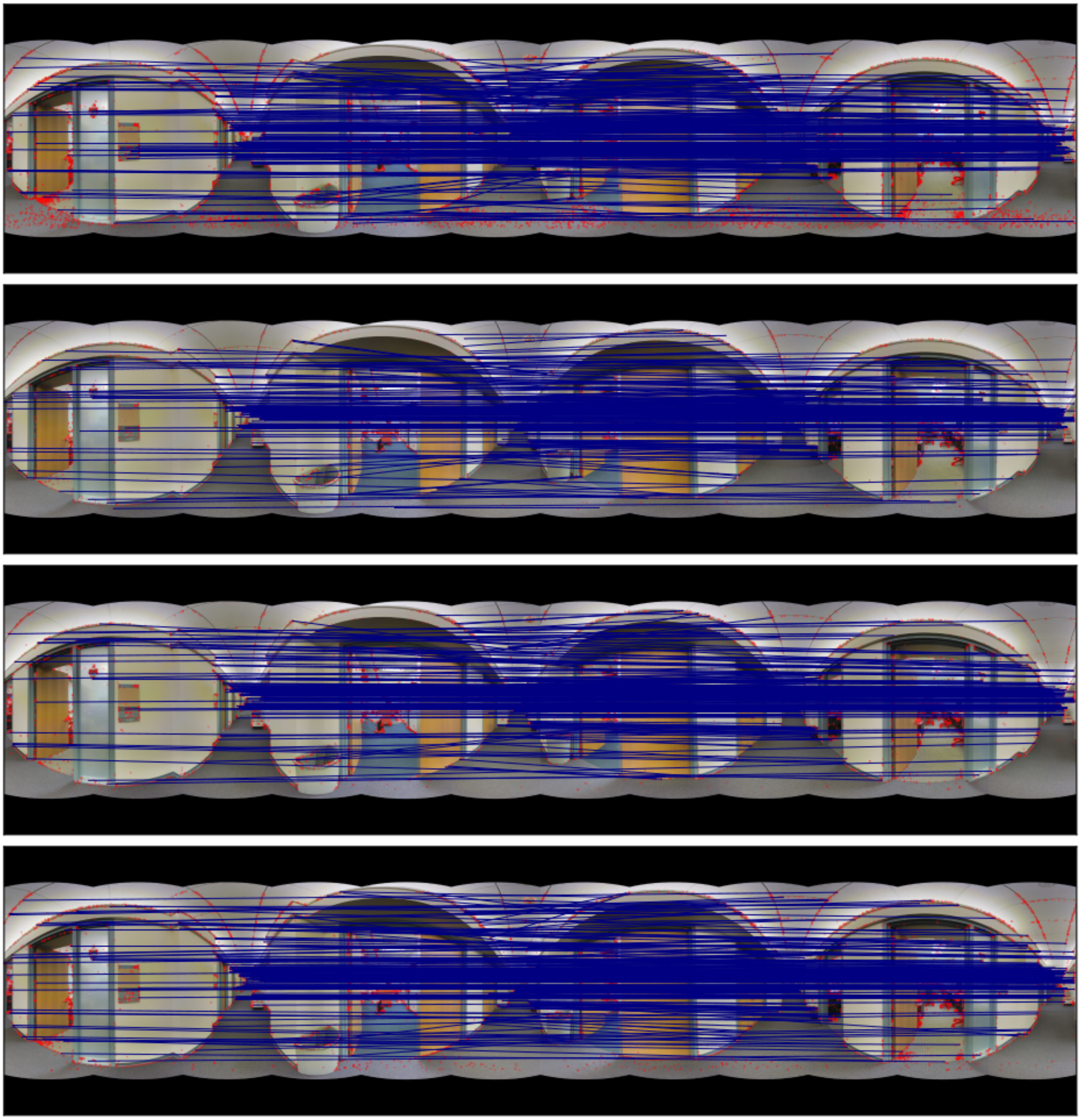


Figure 4: Comparison of SIFT matches on image pair 15. From top to bottom: equirectangular, L0, L1, L2.