UprightNet: Geometry-Aware Camera Orientation Estimation from Single Images
Supplementary Material

Wenqi Xian*,1 Zhengqi Li*,1 Matthew Fisher[2] Jonathan Eisenmann[2] Eli Shechtman[2] Noah Snavely[1]

[1] Cornell Tech, Cornell University [2] Adobe Research

1. Experimental settings

For the experiments InteriorNet and ScanNet experiments, we train our model and all other learning methods for 20 and 15 epochs respectively, on single GPU.

In terms of data augmentation, for training on both InteriorNet and ScanNet, we vary the vertical field of view of the input images to lie between 35 degrees and 45 degrees by cropping the images while keeping their aspect ratios unchanged before resizing to 384×284. We randomly crop training images, but during both validation and test phases, we precompute crop sizes in order so that evaluation results are deterministic.

For our method, we use the following loss hyperparameters in the main manuscript: $\alpha_F = 0.5$, $\alpha_\nabla = 0.125$ for InteriorNet, and $\alpha_F = 2$, $\alpha_\nabla = 0.5$ for ScanNet, found using their respective validation sets.

2. Mathematical details

2.1. Differentiable constrained least squares

In this section, we provide additional mathematical details to supplement Section 3.1 in the main paper. Recall that the Lagrangian in Equation 5 in the main paper is:

$$L = (\mathbf{A}\mathbf{r} - \mathbf{b})^T(\mathbf{A}\mathbf{r} - \mathbf{b}) - \lambda(\mathbf{r}^T\mathbf{r} - 1) \tag{1}$$

The first order optimality condition of Equation 1 leads to the following equations:

$$\frac{\partial L}{\partial \mathbf{x}} = 0 \Rightarrow (\mathbf{A}^T\mathbf{A} - \lambda\mathbf{I})\mathbf{r} = \mathbf{A}^T\mathbf{b} \tag{2}$$

$$\frac{\partial L}{\partial \lambda} = 0 \Rightarrow \mathbf{r}^T\mathbf{r} = 1 \tag{3}$$

To solve fo λ and $\mathbf{r}$ analytically, we use the techniques proposed by [1]. Specifically, we have following proposition from [1]:

Proposition 1. *Equation 2 and Equation 3 can be reduced to a quadratic eigenvalue problem (QEP)*

$$\mathbf{I}\lambda^2 - 2\mathbf{H}\lambda + \mathbf{H}^2 - \mathbf{g}\mathbf{g}^T = 0 \tag{4}$$

where $\mathbf{H} = \mathbf{A}^T\mathbf{A}$ *and* $\mathbf{g} = \mathbf{A}^T\mathbf{b}$.

Proof. According to Equation 2 and Equation 3, we have

$$(\mathbf{H} - \lambda\mathbf{I})\mathbf{r} = \mathbf{g} \tag{5}$$

$$\mathbf{r} = (\mathbf{H} - \lambda\mathbf{I})^{-1}\mathbf{g} \tag{6}$$

$$\mathbf{r}^T\mathbf{r} = 1 \Rightarrow \mathbf{g}^T(\mathbf{H} - \lambda\mathbf{I})^{-2}\mathbf{g} = 1 \tag{7}$$

Now let $\gamma = (\mathbf{H} - \lambda\mathbf{I})^{-2}g$. We then have

$$\begin{cases} \mathbf{g}^T\gamma = 1 \\ (\mathbf{H} - \lambda\mathbf{I})^2\gamma = \mathbf{g} \end{cases} \Rightarrow (\mathbf{H} - \lambda\mathbf{I})^2\gamma = \mathbf{g}\mathbf{g}^T\gamma \tag{8}$$

From Equation 8, we have

$$(\mathbf{I}\lambda^2 - 2\mathbf{H}\lambda + \mathbf{H}^2 - \mathbf{g}\mathbf{g}^T)\gamma = \mathbf{0} \tag{9}$$

and since γ is not necessarily zero, we finally have

$$\mathbf{I}\lambda^2 - 2\mathbf{H}\lambda + \mathbf{H}^2 - \mathbf{g}\mathbf{g}^T = \mathbf{0} \tag{10}$$

□

In terms of the solvability of this QEP described in the Theorem 1 of the main paper, we refer readers to Theorem 5.1 and Theorem 5.2 of [1] for a complete proof.

In order to solve this QEP, we can reduce it to an ordinary eigenvalue problem [1]. Specifically, let $\mu = (\mathbf{H} - \lambda\mathbf{I})\gamma$, then the following two equations can be obtained:

$$\mathbf{H}\gamma - \mu = \lambda\gamma \tag{11}$$

$$\mathbf{H}\mu - \mathbf{g}\mathbf{g}^T\gamma = \lambda\mu \tag{12}$$

Written in matrix form, we finally have

$$\begin{bmatrix} \mathbf{H} & -\mathbf{I} \\ -\mathbf{g}\mathbf{g}^\mathbf{T} & \mathbf{H} \end{bmatrix} \begin{bmatrix} \gamma \\ \mu \end{bmatrix} = \lambda \begin{bmatrix} \gamma \\ \mu \end{bmatrix} \tag{13}$$

2.2. Derivative of eigenvalues for general matrix

Since the block matrix on the left hand side of Equation 13 is not necessarily a real symmetric matrix, the optimal λ corresponds to the minimum real eigenvalue of Equation 13. In order to backpropagte through a neural networks, we need to compute the derivative of this eigenvalue.

The definition of a general eigenvalue problem can be written as:

$$\mathbf{A}\mathbf{X} - \mathbf{X}\mathbf{\Lambda} = \mathbf{0} \tag{14}$$

$$\mathbf{Y}^*\mathbf{A} - \mathbf{A}\mathbf{Y}^* = \mathbf{0} \tag{15}$$

where * is the conjugate transpose of a matrix, and $\mathbf{X}$ is a matrix whose columns contain right eigenvectors, $\mathbf{Y}$ is a matrix whose columns contain left eigenvectors, and $\mathbf{\Lambda}$ is diagonal matrix consisting of eigenvalues. In our case, the left-eigenvectors are chosen such that $\mathbf{Y}^*\mathbf{X} = \mathbf{I}$.

Therefore, the derivative of the eigensystem is given by

$$\partial\mathbf{A}\mathbf{X} - X\partial\mathbf{\Lambda} = -\mathbf{A}\partial\mathbf{X} + \partial\mathbf{X}\mathbf{\Lambda} \tag{16}$$

Premultiplication with the conjugate transpose of left-eigenvectors in Equation 16 leads to

$$\mathbf{Y}^*\partial\mathbf{A}\mathbf{X} - \mathbf{Y}^*\mathbf{X}\partial\mathbf{\Lambda} = \mathbf{Y}^*\mathbf{A}\partial\mathbf{X} + \mathbf{Y}^*\partial\mathbf{X}\mathbf{\Lambda} \tag{17}$$

$$\mathbf{Y}^*\partial\mathbf{A}\mathbf{X} - \partial\mathbf{\Lambda} = \mathbf{Y}^*\mathbf{A}\partial\mathbf{X} + \mathbf{Y}^*\partial\mathbf{X}\mathbf{\Lambda} \tag{18}$$

Numerically speaking, the minimum real eigenvalue of $\mathbf{A}$ is unique. In that case, we have that $\partial\mathbf{X} = \mathbf{X}\mathbf{C}$ since the eigenvectors form a basis of $\mathbb{C}^n$. Thus, we have

$$\mathbf{Y}^*\partial\mathbf{A}\mathbf{X} - \partial\mathbf{\Lambda} = -\mathbf{Y}^*\mathbf{A}\mathbf{X}\mathbf{C} + \mathbf{Y}^*\mathbf{X}\mathbf{C}\mathbf{\Lambda} \tag{19}$$

$$\mathbf{Y}^*\partial\mathbf{A}\mathbf{X} - \partial\mathbf{\Lambda} = -\mathbf{\Lambda}\mathbf{C} + \mathbf{C}\mathbf{\Lambda} \tag{20}$$

$$\mathbf{Y}^*\partial\mathbf{A}\mathbf{X} = \partial\mathbf{\Lambda} \tag{21}$$

Writing Equation 21 in terms of each component, the derivative of each eigenvalue is given as

$$\partial\lambda_k = \mathbf{y}_k^*\partial\mathbf{A}\mathbf{x}_k \tag{22}$$

where λ_k is the k^{th} eigenvalue of $\mathbf{A}$, $\mathbf{y}_k$ and $\mathbf{x}_k$ are the k^{th} left-eigenvector and right-eigenvector of $\mathbf{A}$, respectively. We refer readers to [3, 2] for the original derivation.

3. Ground truth camera orientation distributions

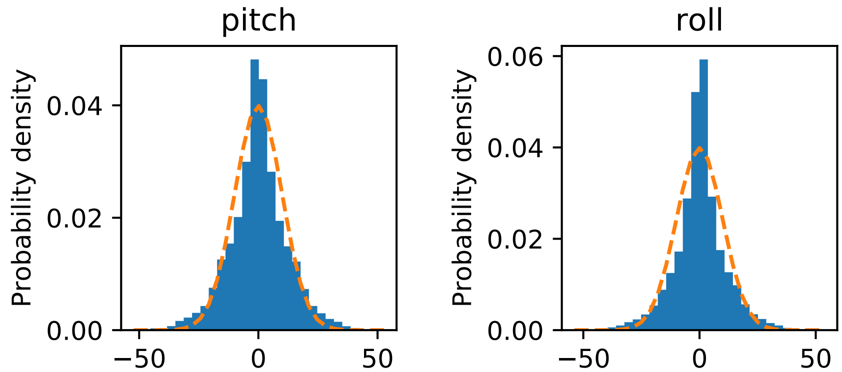

Figure 1. **Distribution of ground-truth roll and pitch angles in InteriorNet.** x-axes indicate pitch (left) and roll (right) value in degrees. y-axes indicate the frequency of samples.

Figure 1 shows the distribution of ground-truth roll and pitch in InteriorNet, and Figure 2 shows the distribution of ground-truth roll and pitch in ScanNet.

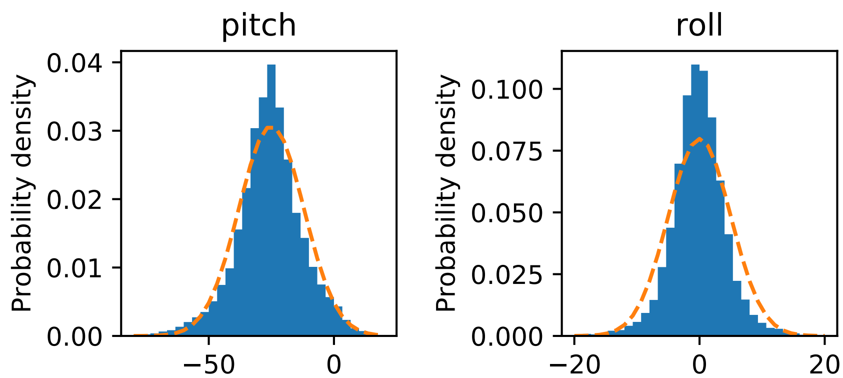

Figure 2. **Distribution of ground-truth roll and pitch in ScanNet.** x-axes indicate pitch (left) and roll (right) value in degrees. y-axes indicate the frequency of samples. Note that ScanNet has a bias towards cameras pitched downwards.

4. Additional results

Figure 3 shows predicted surface frames and weight images from the ScanNet test set, and Figure 4 shows predicted surface frames and weight images from the SUN360 test set.

References

[1] W. Gander, G. H. Golub, and U. von Matt. A constrained eigenvalue problem. *Linear Algebra and its applications*, 114:815–839, 1989. 1, 2

[2] R. B. Nelson. Simplified calculation of eigenvector derivatives. *AIAA journal*, 14(9):1201–1205, 1976. 2

[3] N. van der Aa. Perturbation theory for eigenvalue problems. 2005. 2

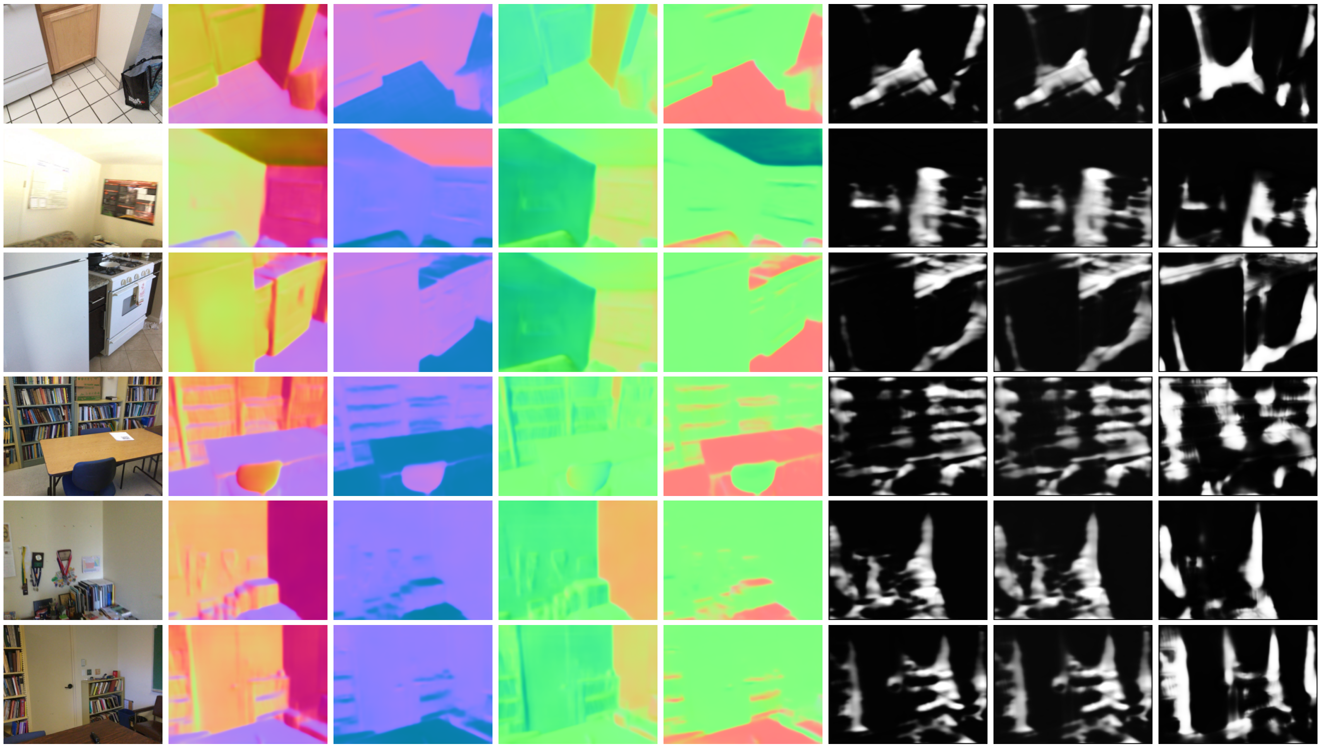

Figure 3. **Visualizations of predicted surface frames and associated weights in the ScanNet testset.**

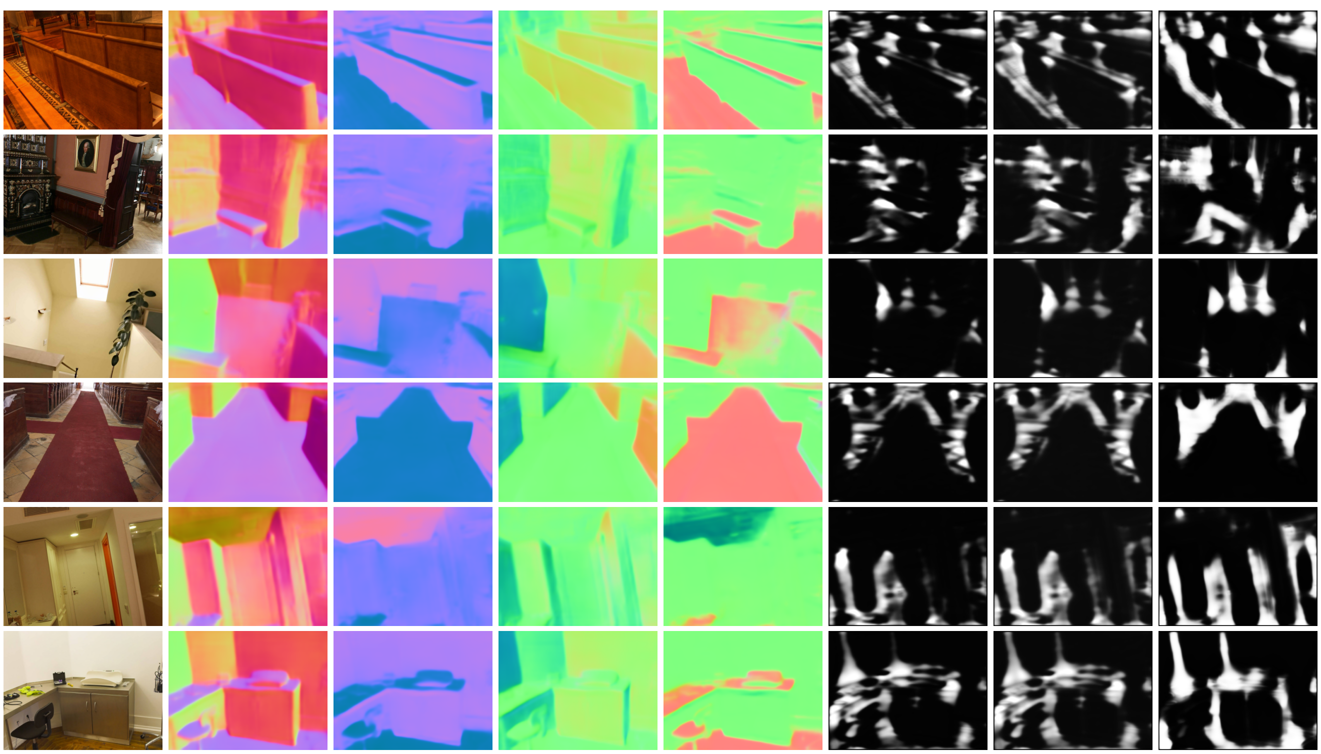

Figure 4. **Visualizations of predicted surface frames and associated weights in the SUN360 testset.**