NeurVPS: Neural Vanishing Point Scanner via Conic Convolution

Yichao Zhou[*] Haozhi Qi[*] Jingwei Huang[ǂ] Yi Ma[*]

[*]University of California, Berkeley

[ǂ]Stanford University

NeurIPS 2019

Introduction

- Parallel lines in 3D intersect in one point after projection
- Vanishing points are important as it gives the **line direction** in 3D

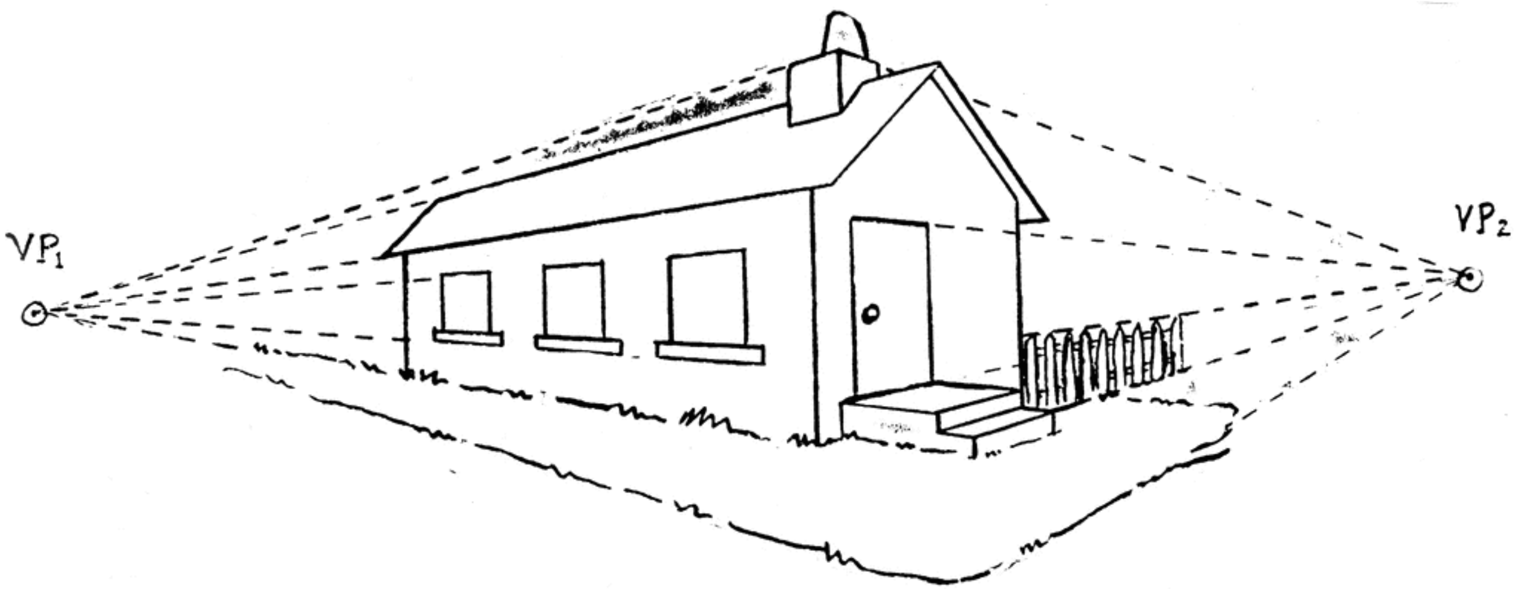
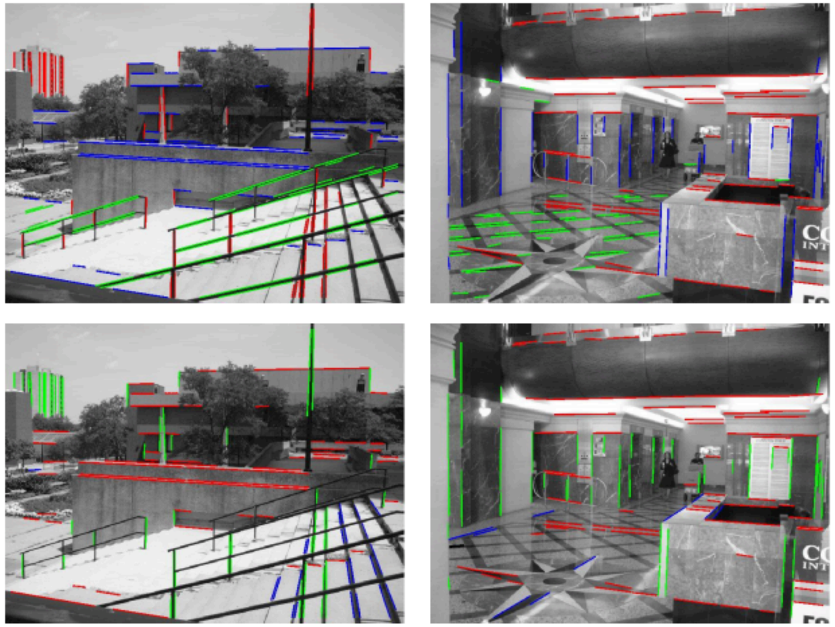


Image source: Military Science and Tactics.

Related Work (Traditional Approaches)

- Two-stage pipeline
- Heuristic Line Segment Detection
  - Canny Edge + Hough Transformation [1]
  - LSD [2]
  - Contour [3]
- Line Clustering
  - J-Linkage [4]
  - Line RANSAC [5]
  - Angle Histogram [6]
- Problems
  - Edges do not have semantic meaning
  - Edges can be noisy
  - Outliers can result in total failure



[1] Kiryati, Nahum, Yuval Eldar, and Alfred M. Bruckstein. "A probabilistic Hough transform." *Pattern recognition* 24.4 (1991): 303-316.
[2] Von Gioi, et al. “LSD: A fast line segment detector with a false detection control.” *PAMI* 32.4 (2008
[3] Zhou, Zihan, Farshid Farhat, and James Z. Wang. "Detecting dominant vanishing points in natural scenes with application to composition-sensitive image retrieval." *IEEE Transactions on Multimedia* 19.12 (2017
[4] Tardif, Jean-Philippe. "Non-iterative approach for fast and accurate vanishing point detection." *2009 ICCV*.
[5] Bazin, Jean-Charles, and Marc Pollefeys. "3-line ransac for orthogonal vanishing point detection." *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012.
[6] Li, Bo, et al. "Vanishing point detection using cascaded 1D Hough Transform from single images." *Pattern Recognition Letters* 33.1 (2012): 1-8.

Related Work (Neural Network Era)

- Recent data-driven approaches
  - [1], [2], [3]: divide image into patches and do classification
    - Hard to find vanishing point outside the image
  - [4] uses neural network to filter outliers
- Challenges:
  - Neural network does not have a geometric understanding of vanishing points
  - CNN only provides a coarse estimations of vanishing points

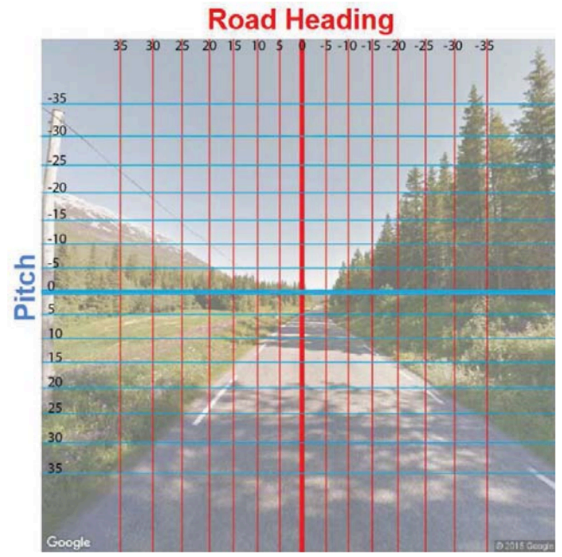


Fig. 3. Discretized VP labels in a total of 15x15=225 labels

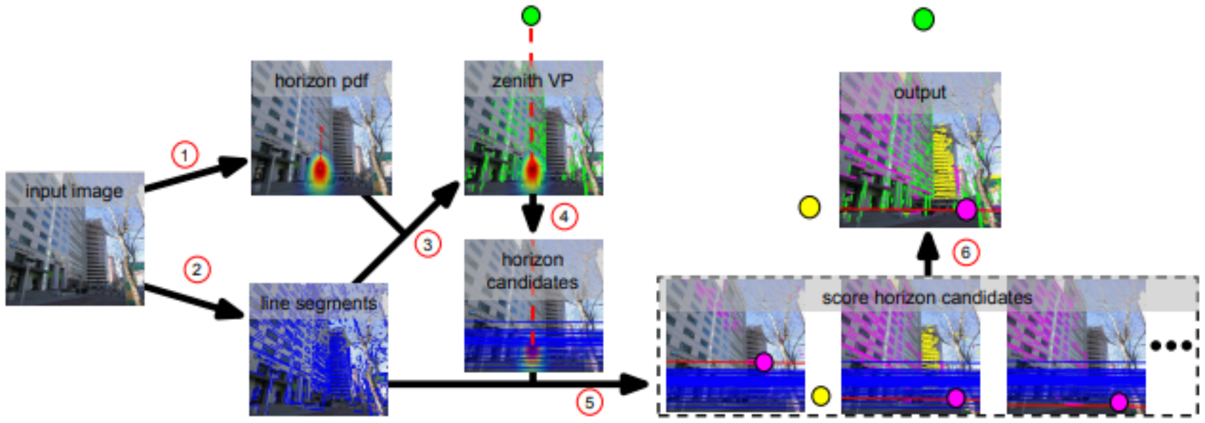
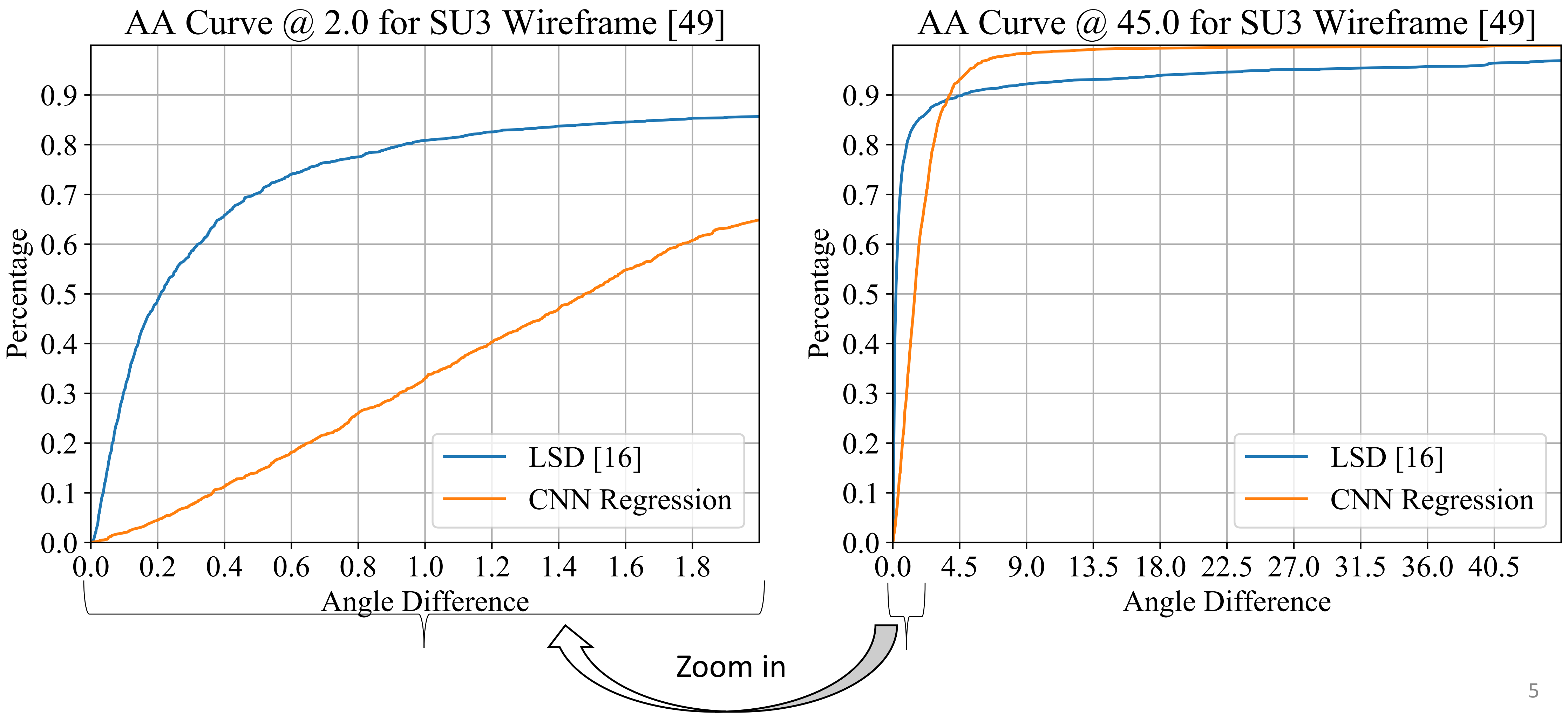


Figure 2: Algorithm overview: 1) use global image context to estimate a prior over horizon lines (Sec. 3); 2) extract line segments; 3) identify the zenith VP (Sec. 4.1); 4) sample horizon line candidates consistent with the zenith VP (Sec. 4.2); 5) find VPs on horizon line candidates (Sec. 4.2); and 6) select the best horizon line based on the VPs it contains (Sec. 4.3).

[1] "Vanishing point detection with convolutional neural networks", Ali Borji, arXiv 2016
[2] "DeepVP: Deep learning for vanishing point detection on 1 million street view images", Chin-Kai Chang, Jiaping Zhao, and Laurent Itti. ICRA 2018
[3] "Dominant vanishing point detection in the wild with application in composition analysis", Xiaodan Zhang, Xinbo Gao, Wen Lu, Lihuo He, and Qi Liu. NeuralComputing 2018
[4] "Detecting Vanishing Points using Global Image Context in a Non-Manhattan World" Menghua Zhai, Scott Workman, Nathan Jacobs. CVPR 2016

Poor Accuracy of CNNs on VP Detection



Design Philosophy of NeurVPS

- The overall approach has the advantages of
  - ***accuracy*** of traditional line clustering algorithms
  - ***robustness*** of neural network-based algorithms
- Neural networks should be trained **end-to-end**
  - without relying on line segment detectors
- New operators that captures geometric cues
  - vanishing points are the intersections of lines
  - operators should be *local* and *stackable*

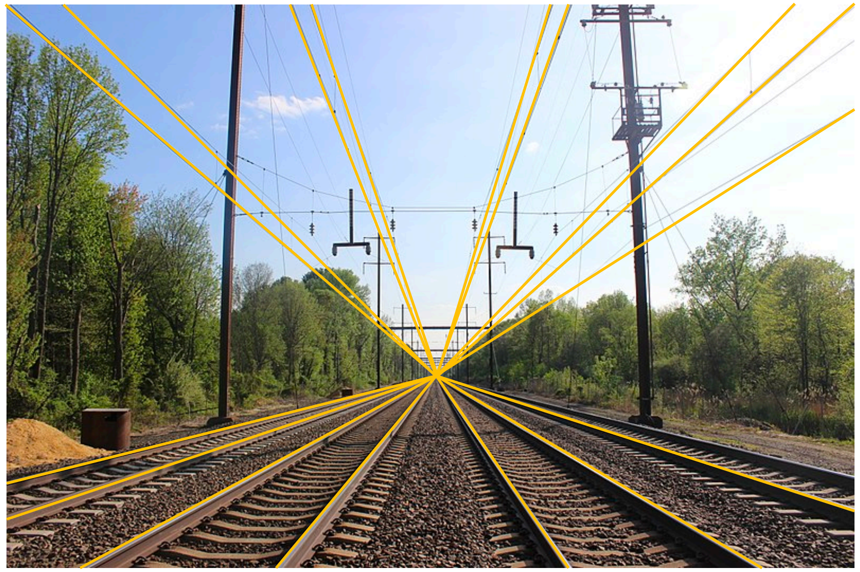
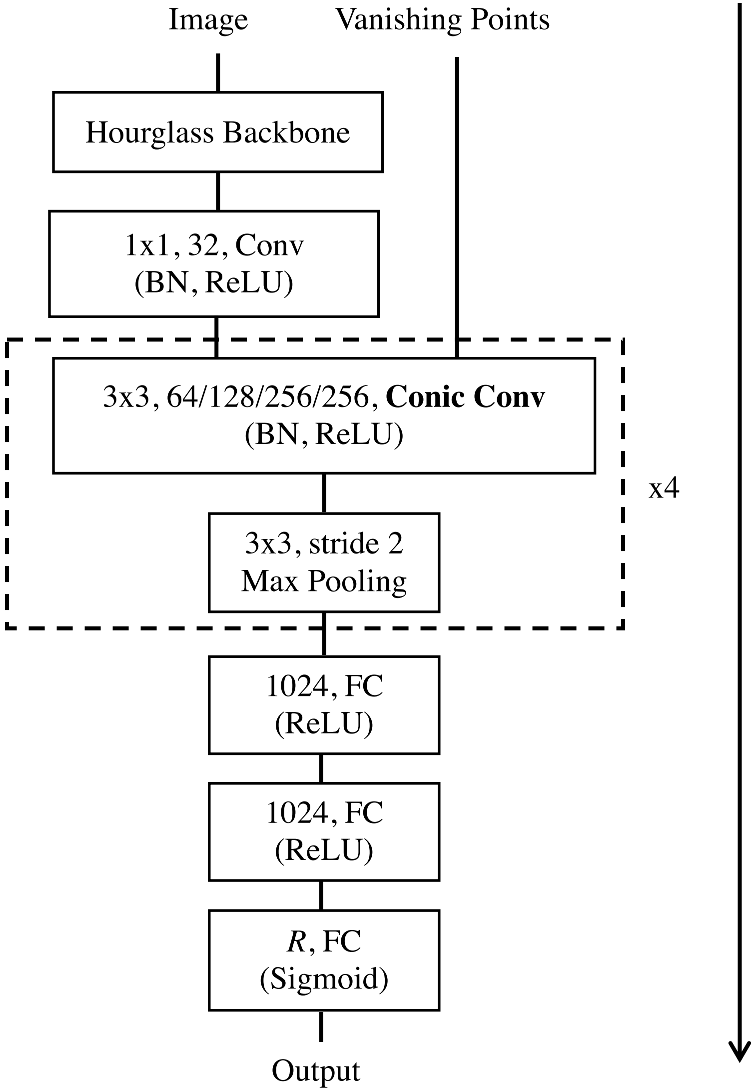


Image Source: Wikipedia

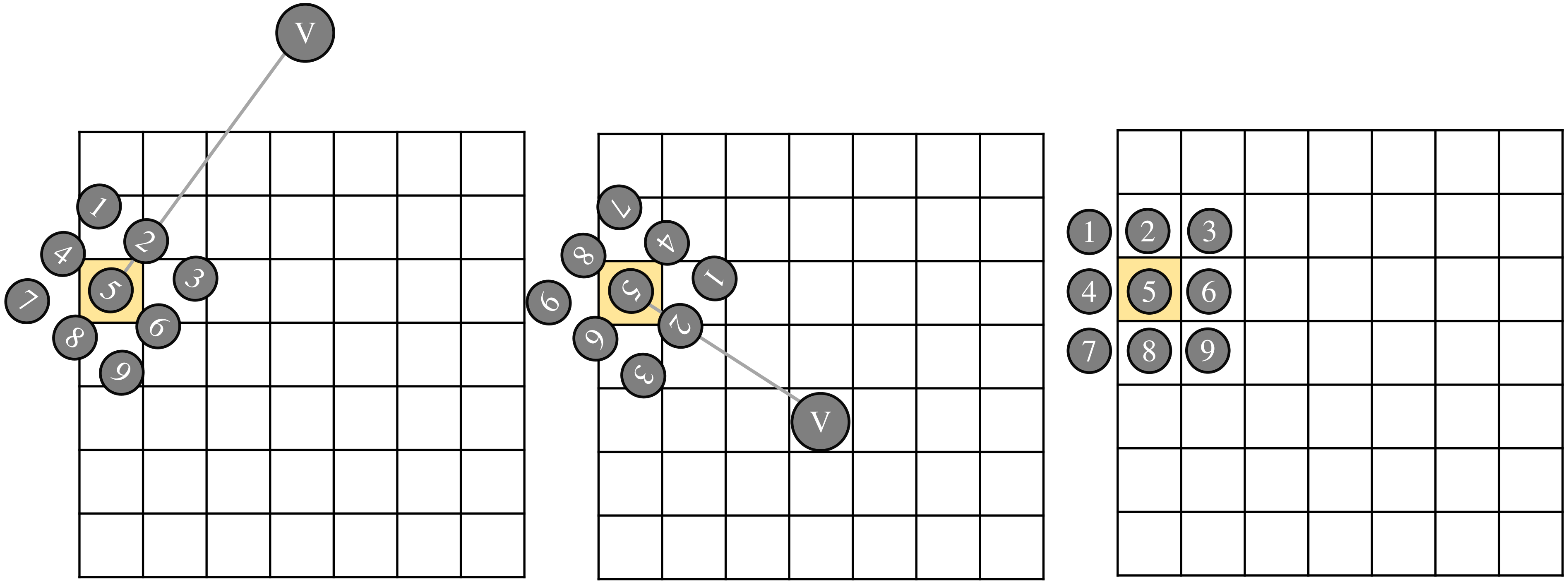
Our Methods

- Input
  - An image
  - A coordinate (vanishing point candidate)
- Output
  - likelihood of the existence of a vanishing point near that coordinate.
- Key Component
  - Conic Convolution



Conic Convolution

- Guided by vanishing point candidates (convolution center)



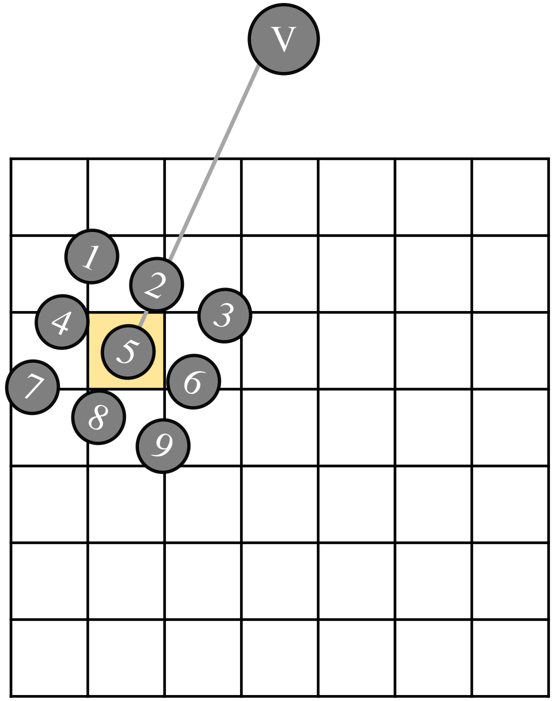
Conic Convolution
(vanishing point outside image plane)

Conic Convolution
(vanishing point inside image plane)

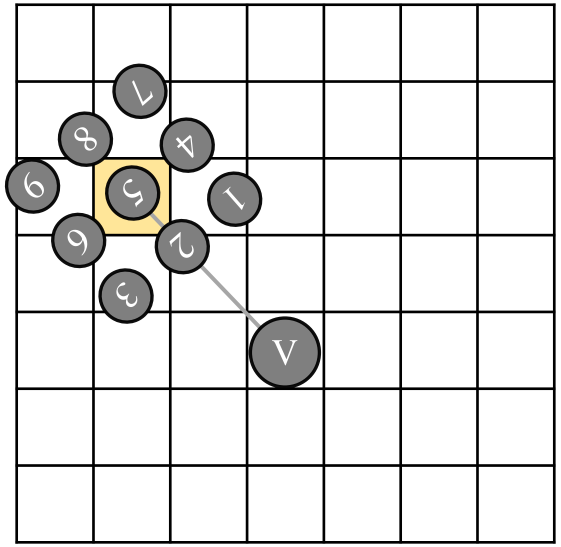
Plain Convolution

Conic Convolution

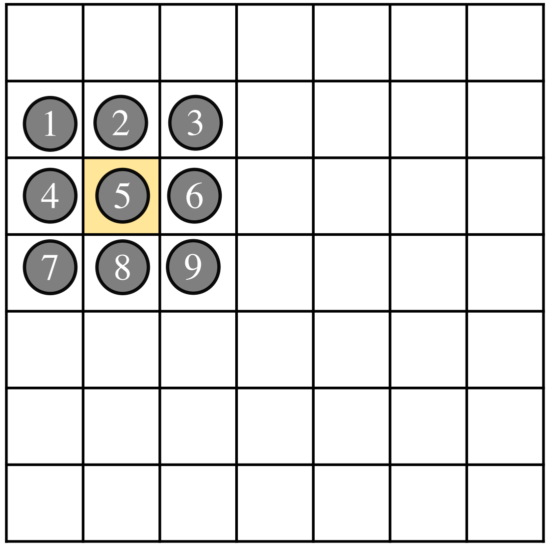
- Guided by vanishing point candidates (convolution center)



Conic Convolution
(vanishing point outside image plane)



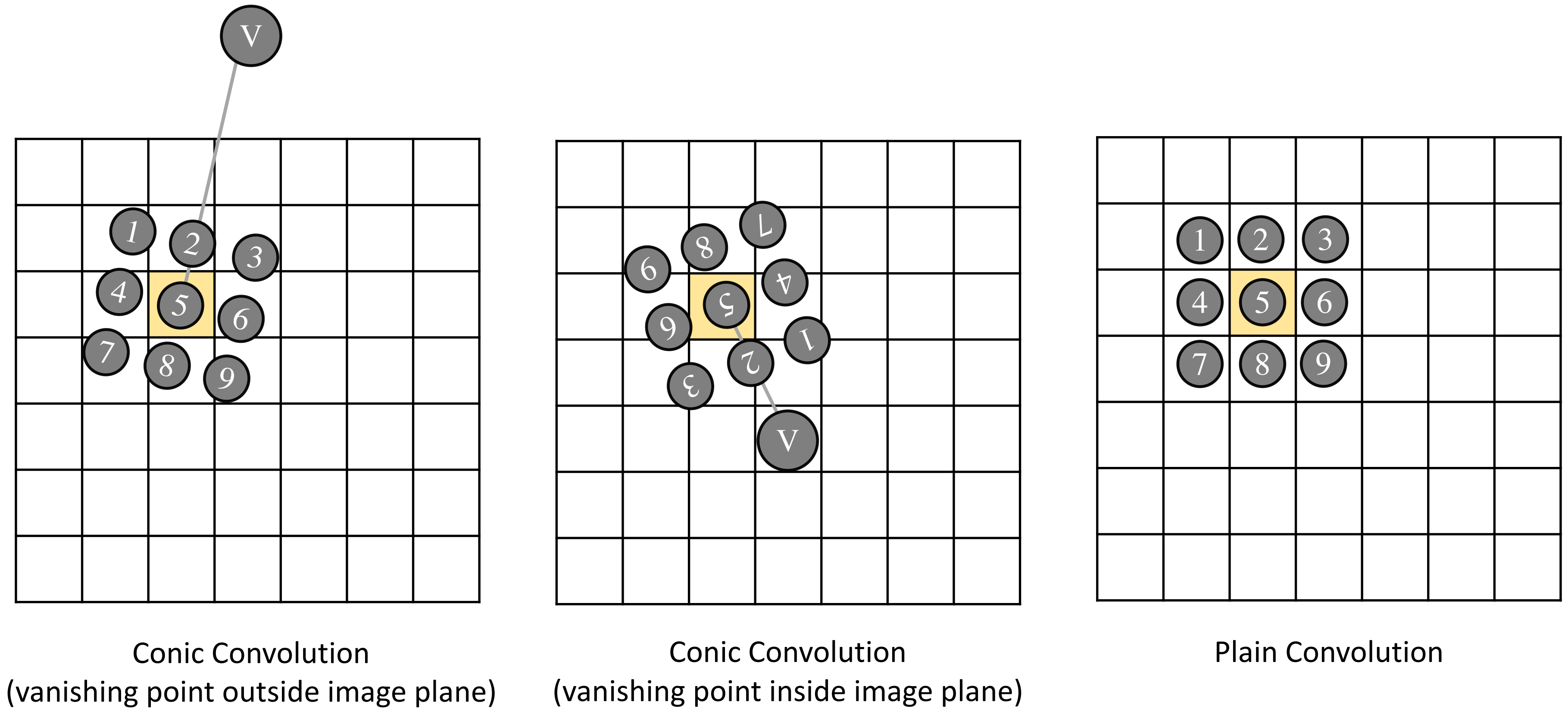
Conic Convolution
(vanishing point inside image plane)



Plain Convolution

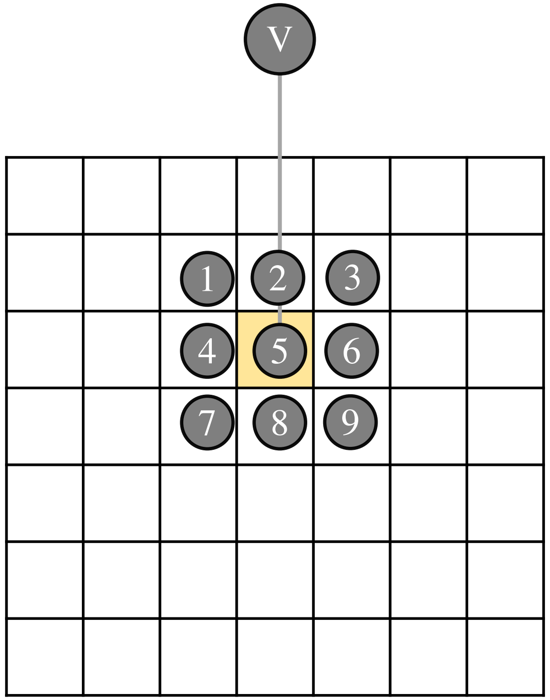
Conic Convolution

- Guided by vanishing point candidates (convolution center)

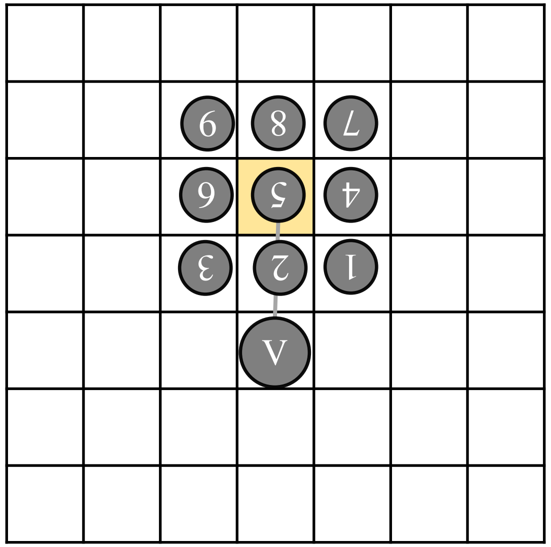


Conic Convolution

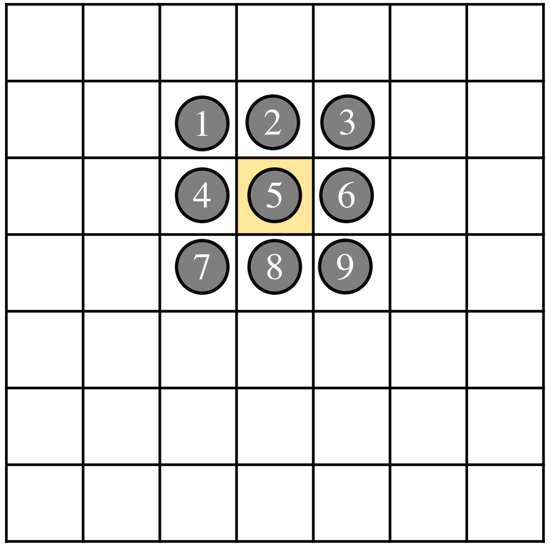
- Guided by vanishing point candidates (convolution center)



Conic Convolution
(vanishing point outside image plane)



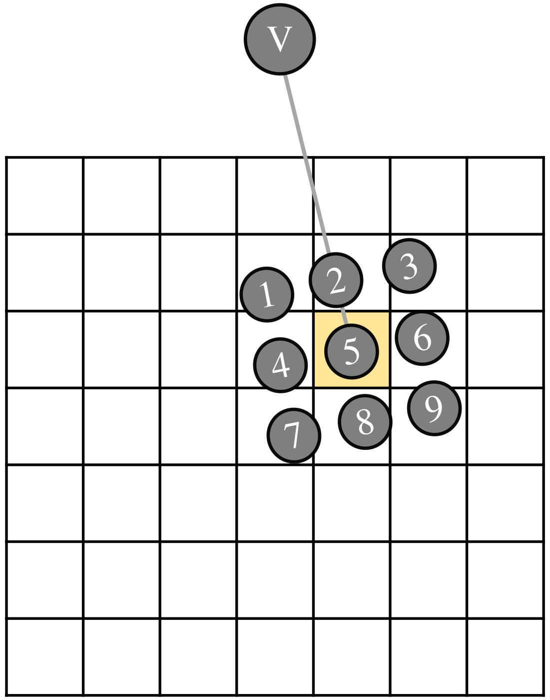
Conic Convolution
(vanishing point inside image plane)



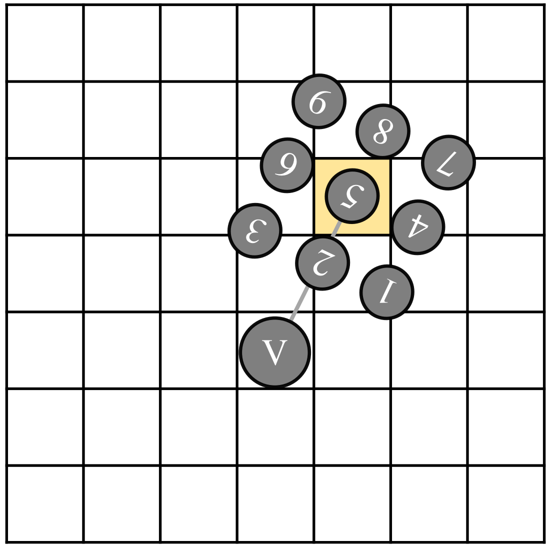
Plain Convolution

Conic Convolution

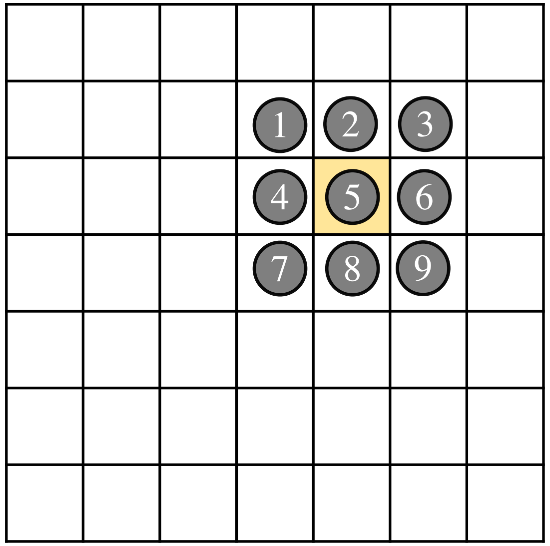
- Guided by vanishing point candidates (convolution center)



Conic Convolution
(vanishing point outside image plane)

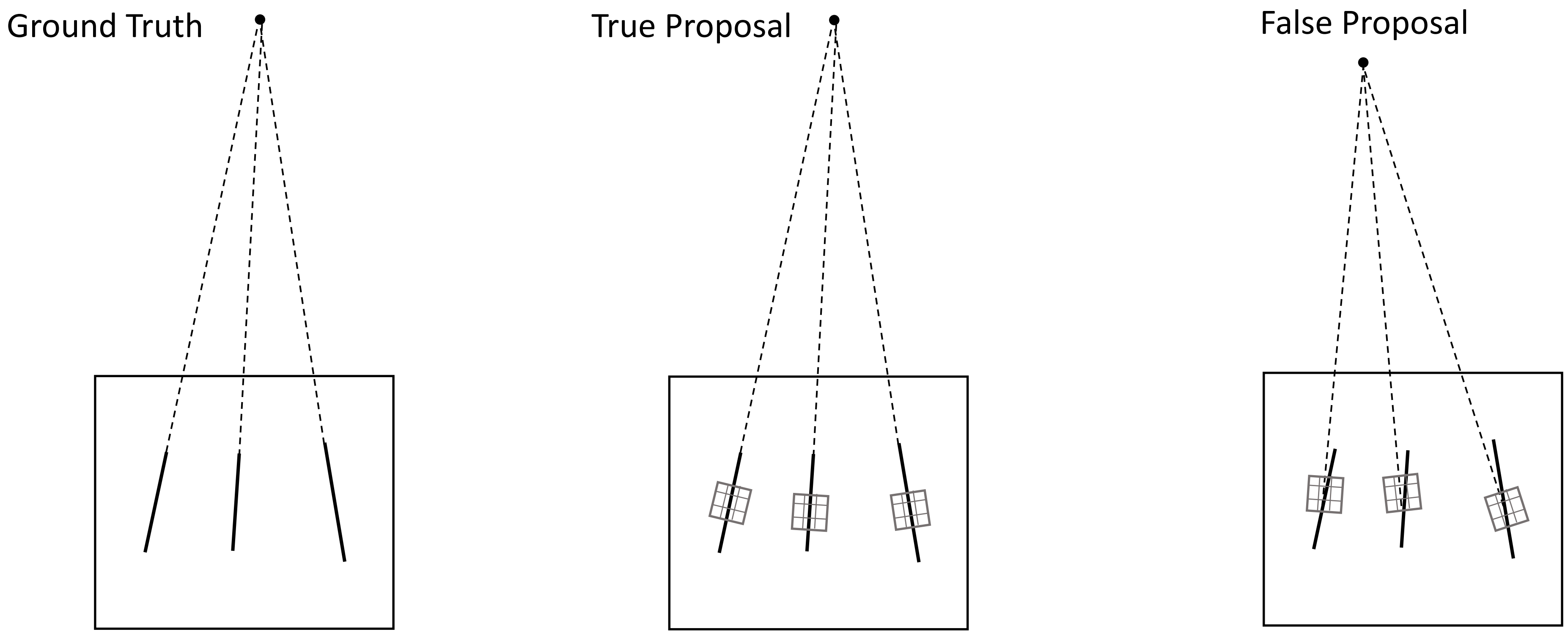


Conic Convolution
(vanishing point inside image plane)



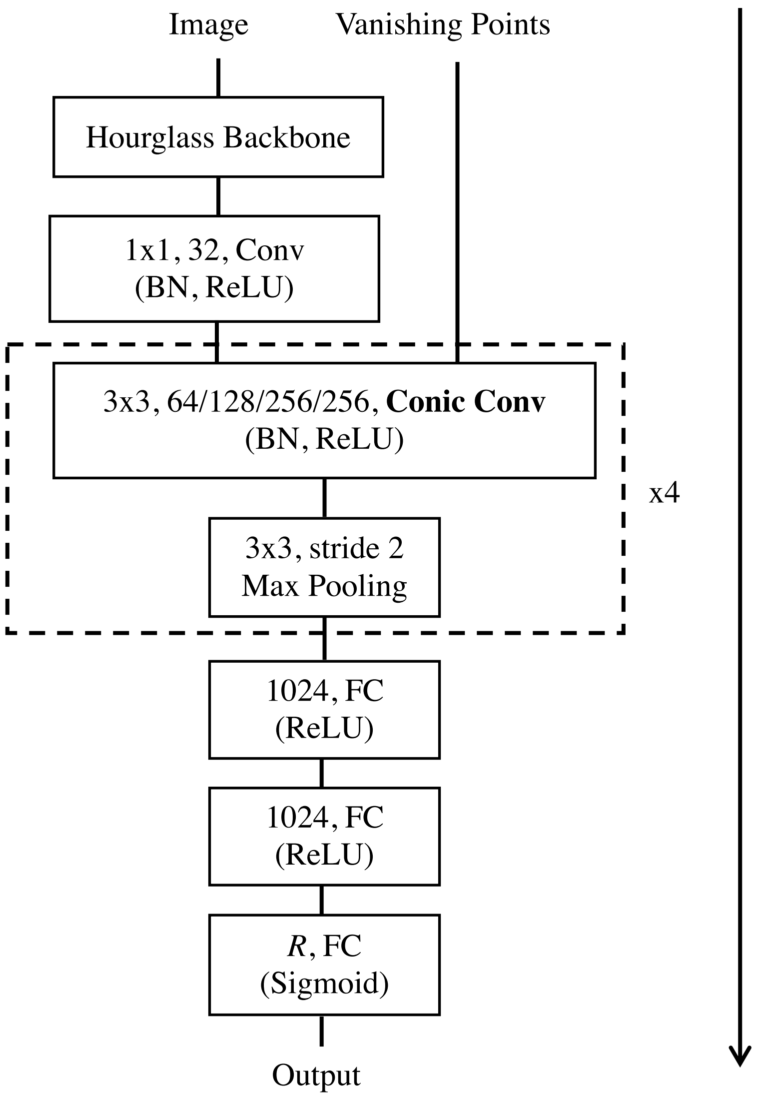
Plain Convolution

Intuition Behind Conic Convolution



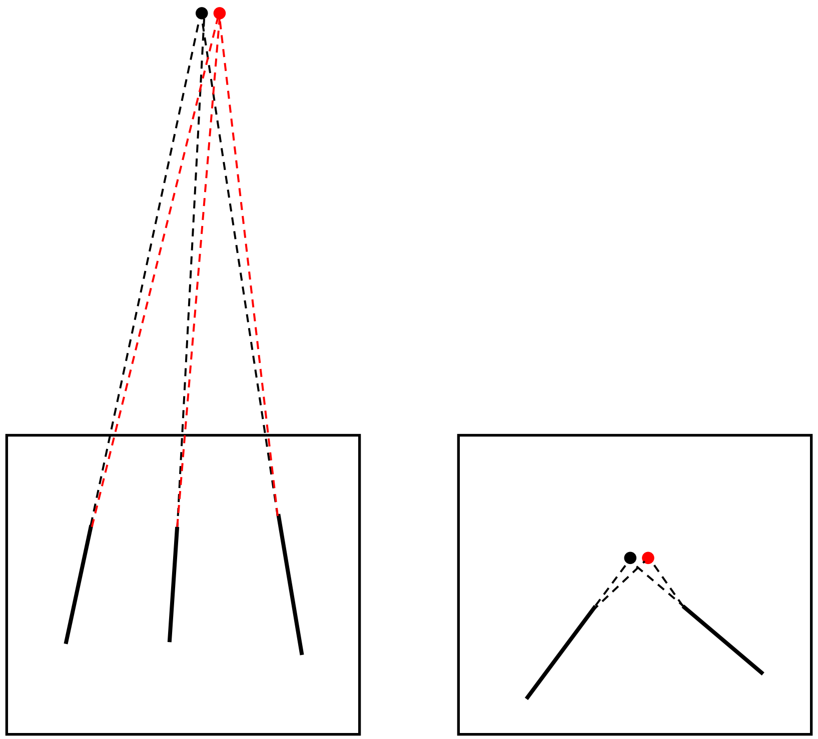
Coarse-to-Fine Inference

- Our network is essentially a vanishing point classifier
- During evaluation
  1. Sample vanishing points
  2. Test it with our network classifier
- How to sample vanishing points?



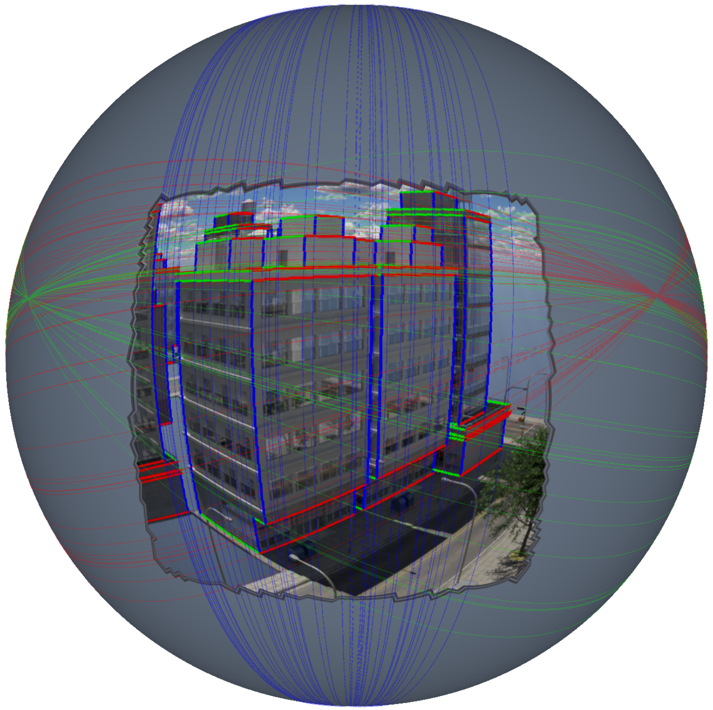
A very brief review of Gaussian Sphere

- How to do uniform sampling for vanishing point?

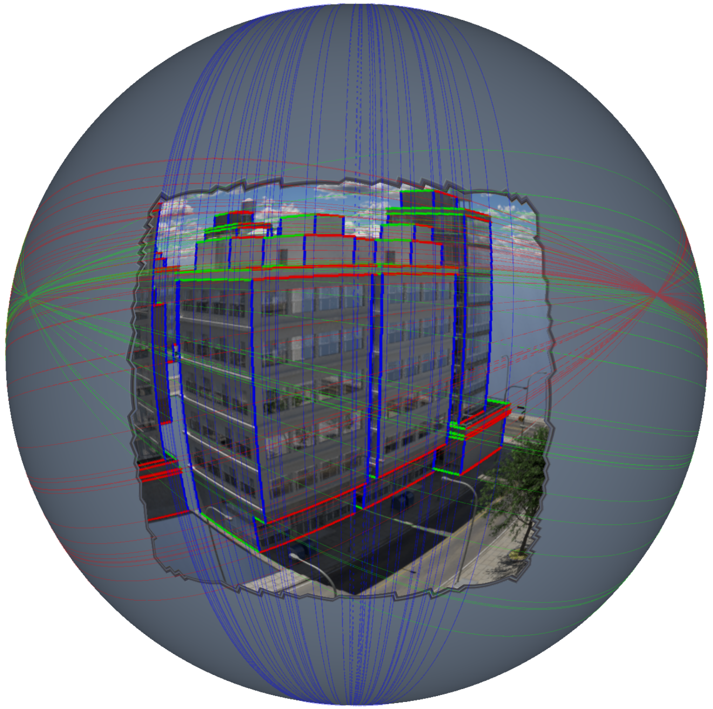


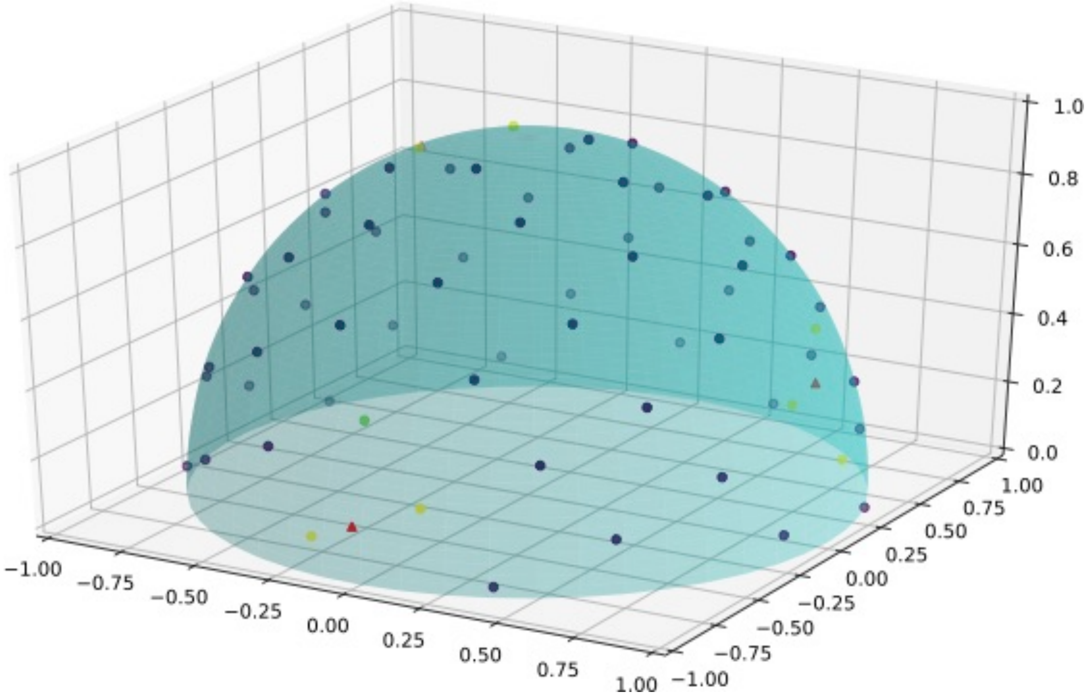
A very brief review of Gaussian Sphere

- How to do uniform sampling for vanishing point?
- We put the image on a sphere (Gaussian Sphere Representation)

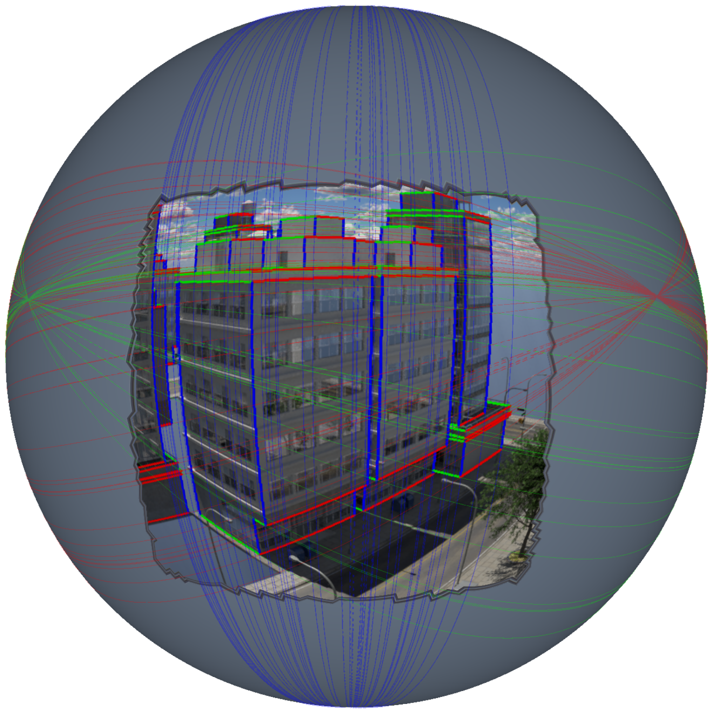


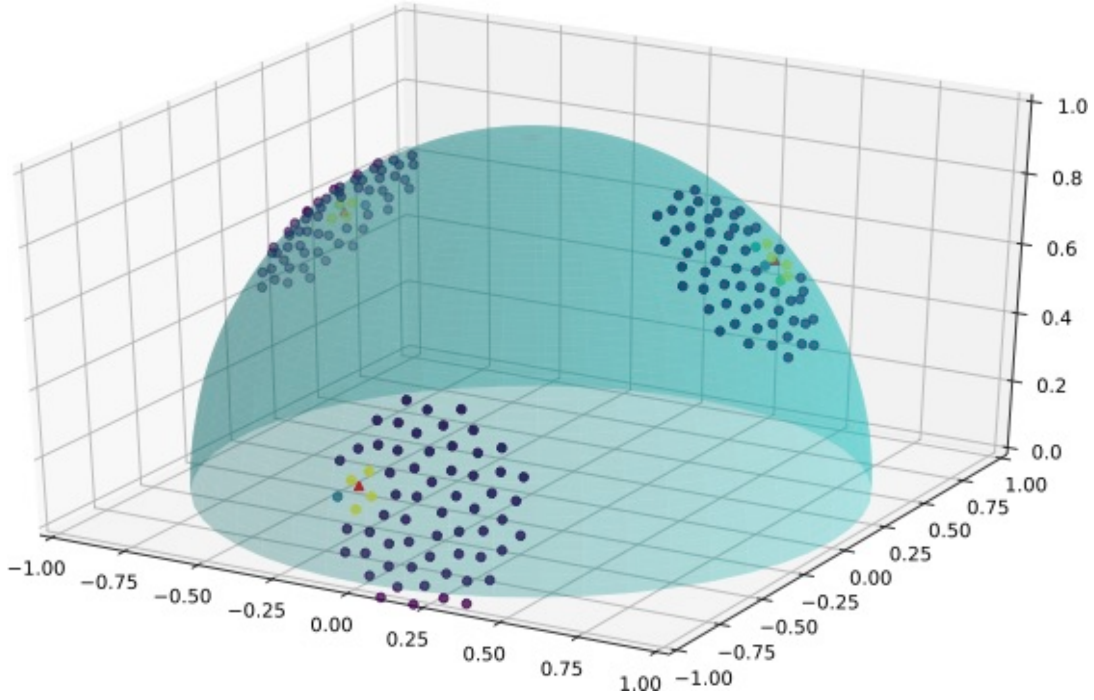
Hierarchical Inference



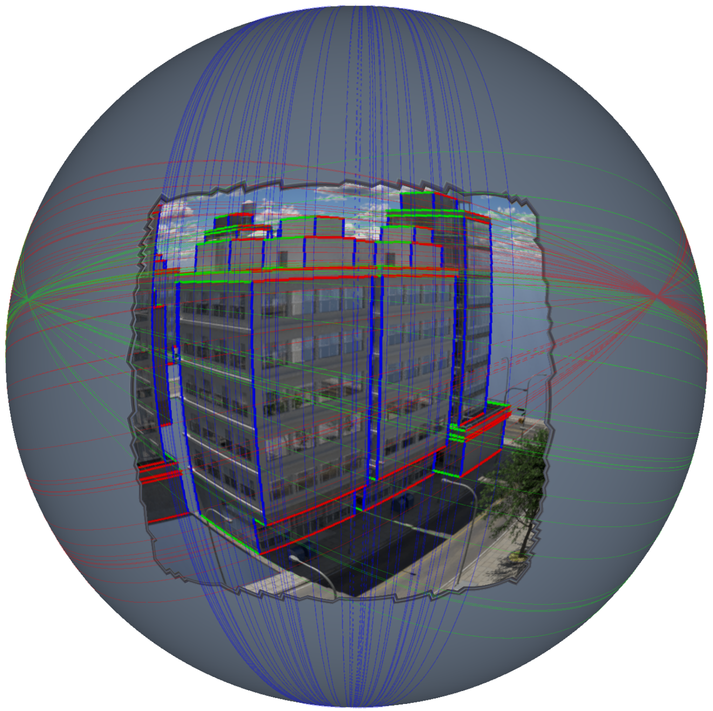


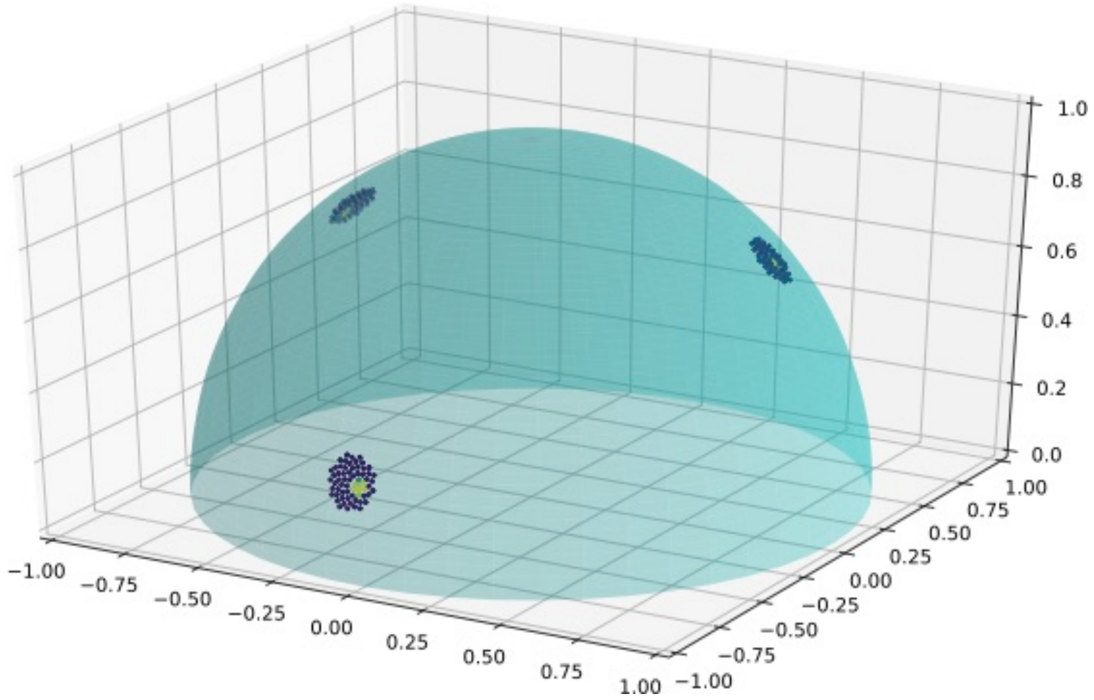
Hierarchical Inference





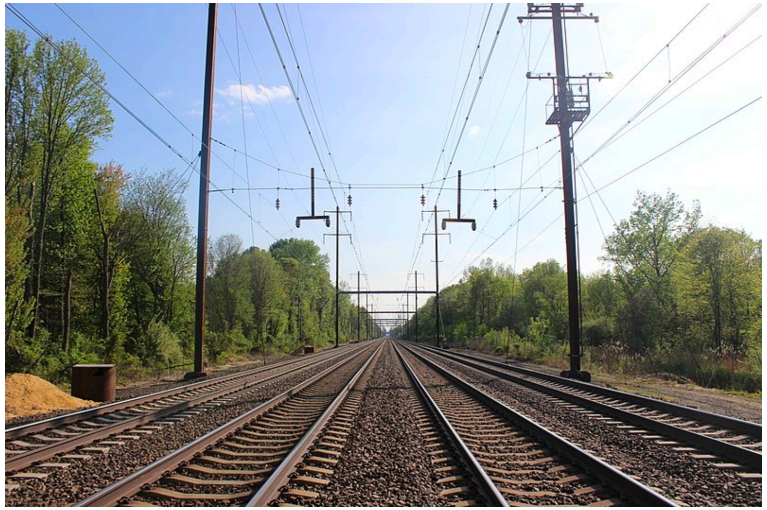
Hierarchical Inference

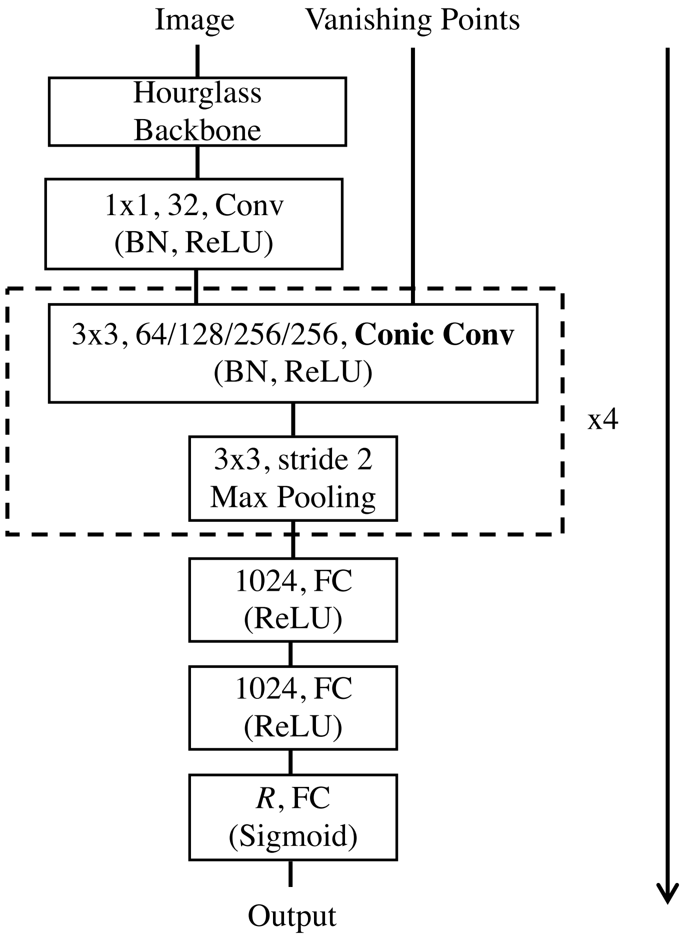




Training

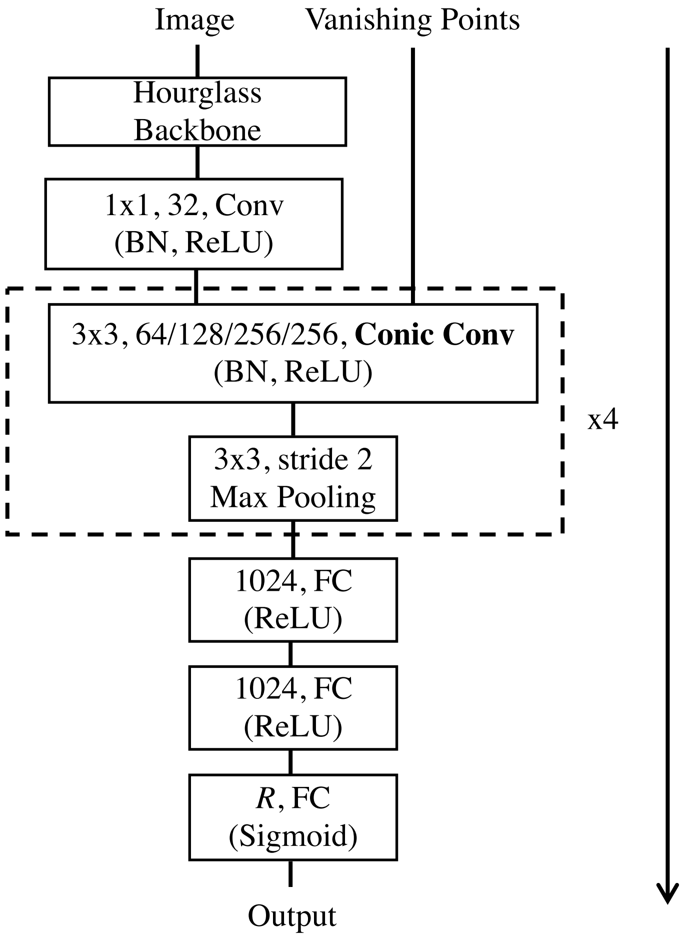
- We train multiple classifiers, each of which corresponds to a different threshold;
- Sample one positive & one negative vanishing points for each threshold;
- Randomly sample three vanishing points to reduce bias.





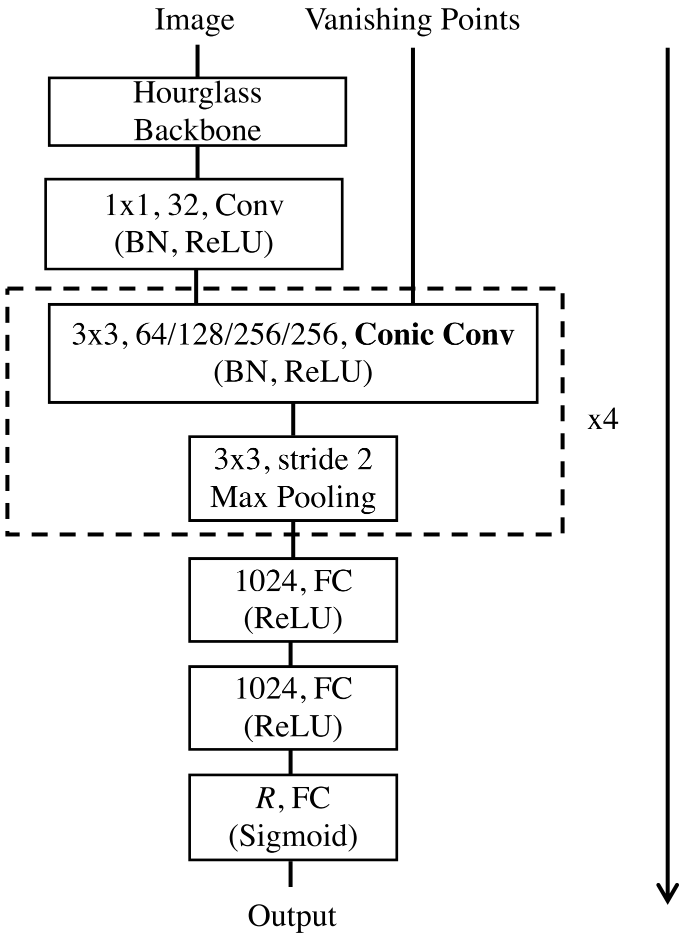
Training

- We train multiple classifiers, each of which corresponds to a different threshold;
- Sample one positive & one negative vanishing points for each threshold;
- Randomly sample three vanishing points to reduce bias.



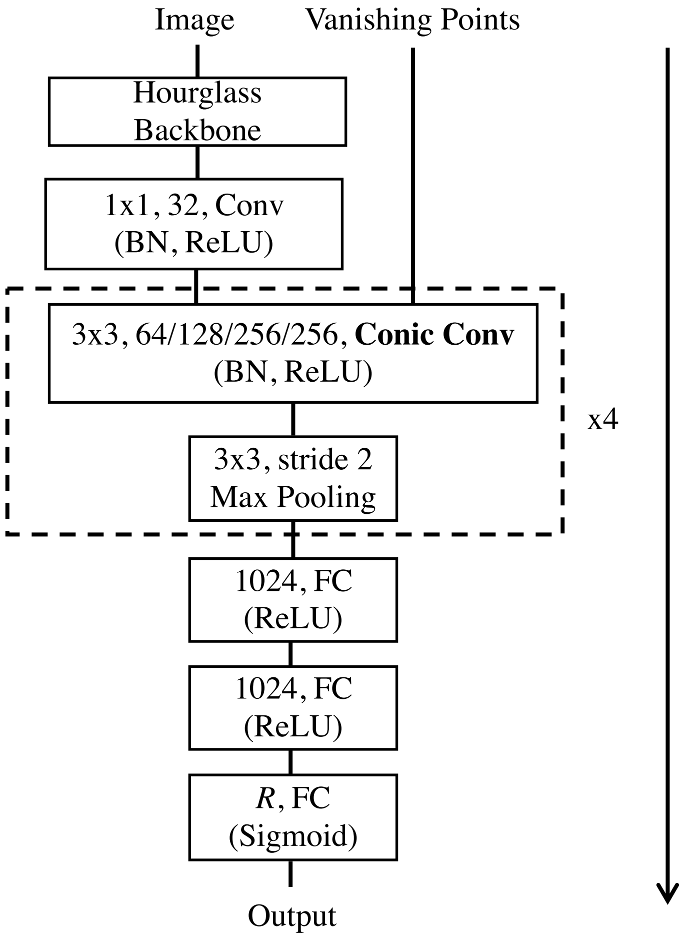
Training

- We train multiple classifiers, each of which corresponds to a different threshold;
- Sample one positive & one negative vanishing points for each threshold;
- Randomly sample three vanishing points to reduce bias.



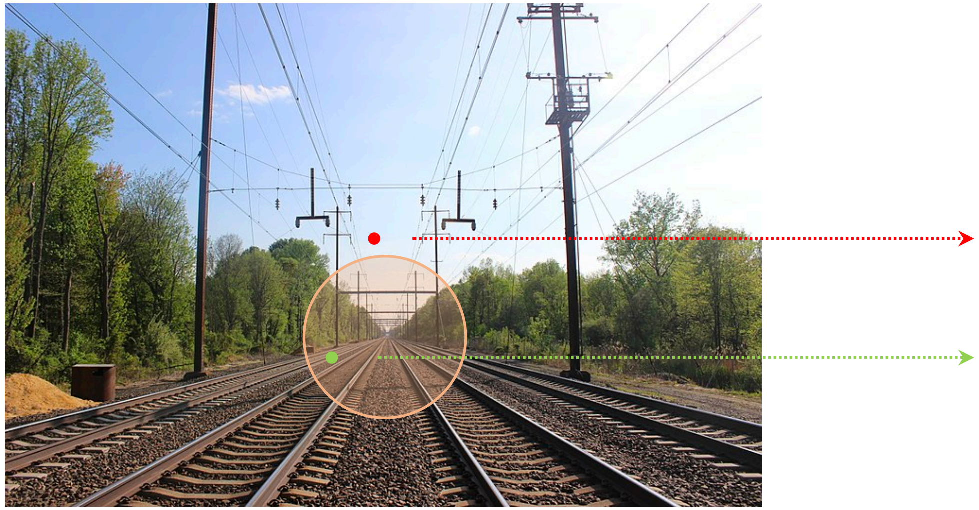
Training

- We train multiple classifiers, each of which corresponds to a different threshold;
- Sample one positive & one negative vanishing points for each threshold;
- Randomly sample three vanishing points to reduce bias.



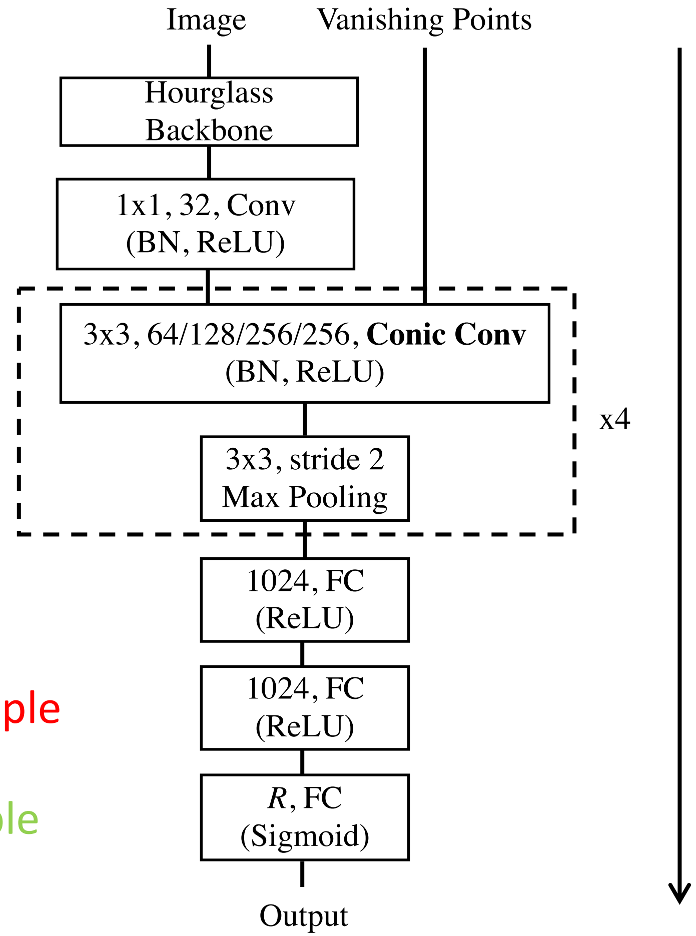
Training

- We train multiple classifiers, each of which corresponds to a different threshold;
- Sample one positive & one negative vanishing points for each threshold;
- Randomly sample three vanishing points to reduce bias.



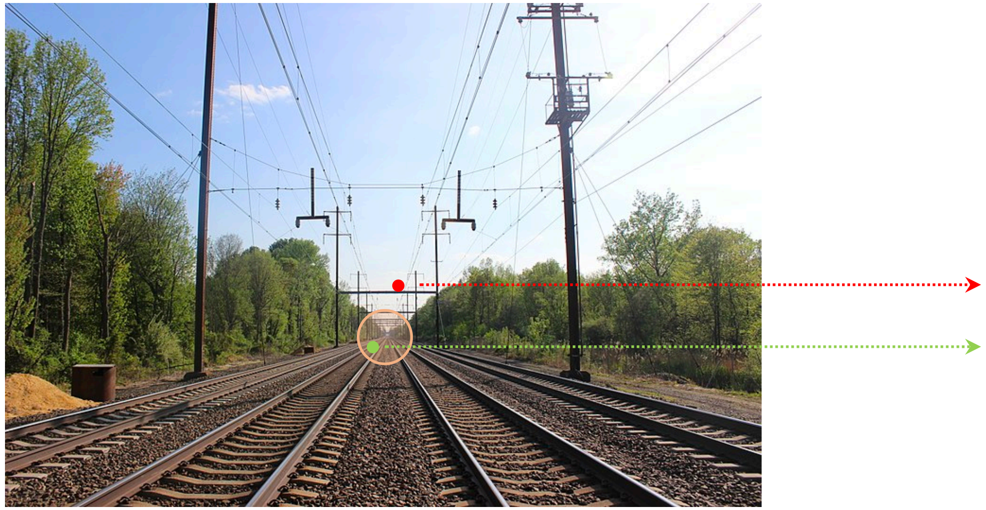
Negative Sample

Positive Sample

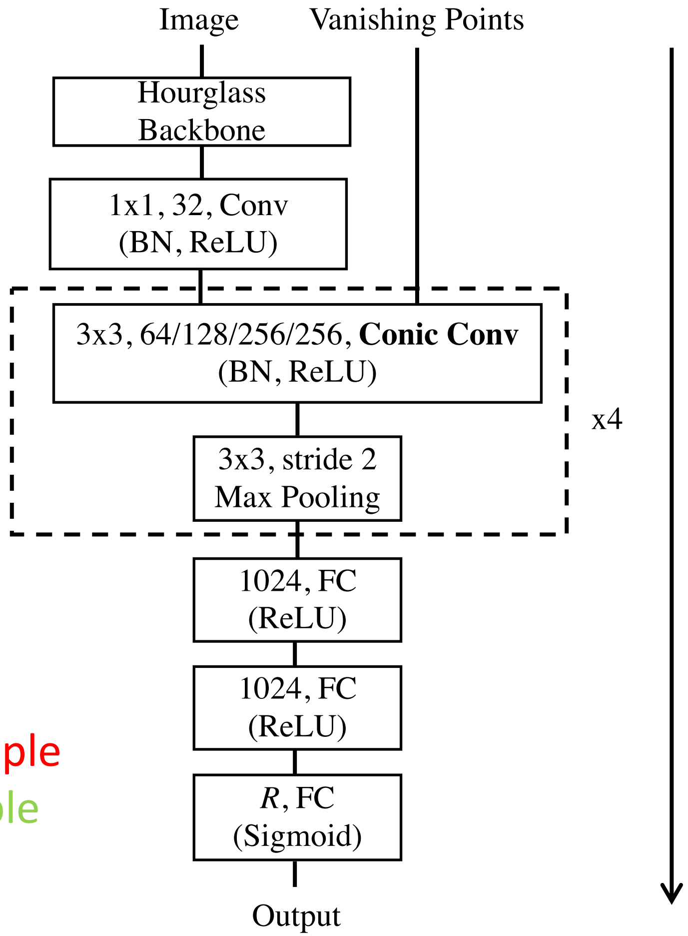


Training

- We train multiple classifiers, each of which corresponds to a different threshold;
- Sample one positive & one negative vanishing points for each threshold;
- Randomly sample three vanishing points to reduce bias.

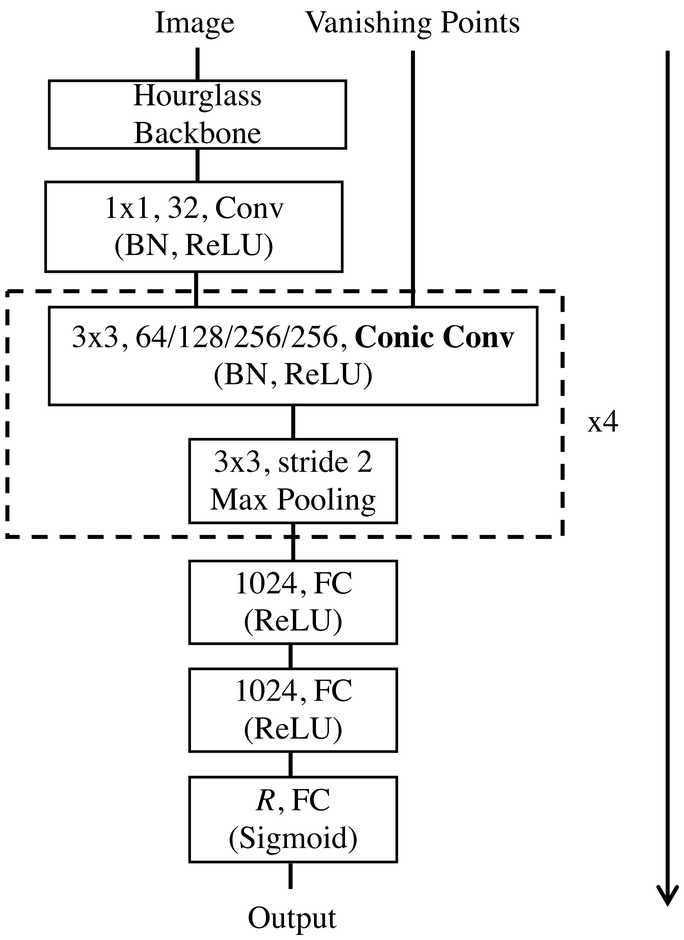


Negative Sample
Positive Sample



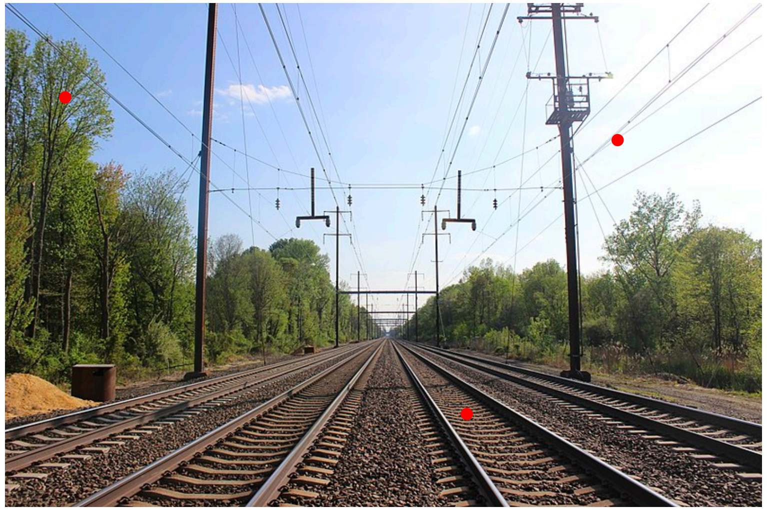
Training

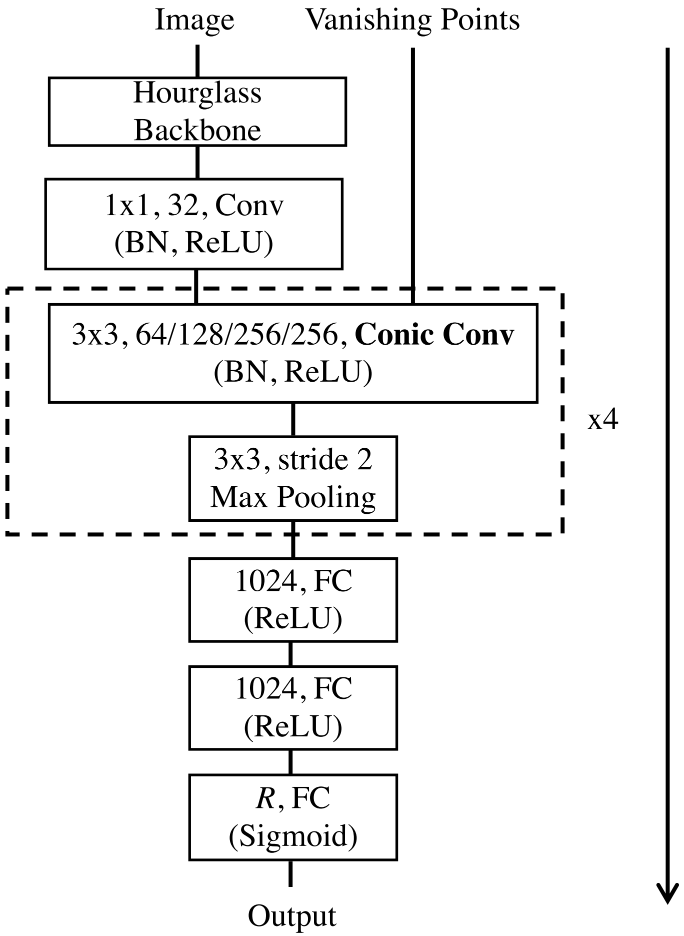
- We train multiple classifiers, each of which corresponds to a different threshold;
- Sample one positive & one negative vanishing points for each threshold;
- Randomly sample three vanishing points to reduce bias.



Training

- We train multiple classifiers, each of which corresponds to a different threshold;
- Sample one positive & one negative vanishing points for each threshold;
- Randomly sample three vanishing points to reduce bias.





Experiments

- Datasets
  - Synthetic Urban 3D Dataset [1]
  - Natural Scene Dataset [2]
  - ScanNet Dataset [3]
- Evaluation Metric
  - Angle Accuracy Curves Introduced
- Algorithms:
  - 7 different vanishing point detection methods

[1] Zhou, Yichao, et al. "Learning to Reconstruct 3D Manhattan Wireframes from a Single Image." arXiv preprint arXiv:1905.07482 (2019).

[2] Zhou, Zihan, Farshid Farhat, and James Z. Wang. "Detecting dominant vanishing points in natural scenes with application to composition-sensitive image retrieval." IEEE Transactions on Multimedia 19.12 (2017): 2651-2665.

[3] Dai, Angela, et al. "Scannet: Richly-annotated 3d reconstructions of indoor scenes." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.

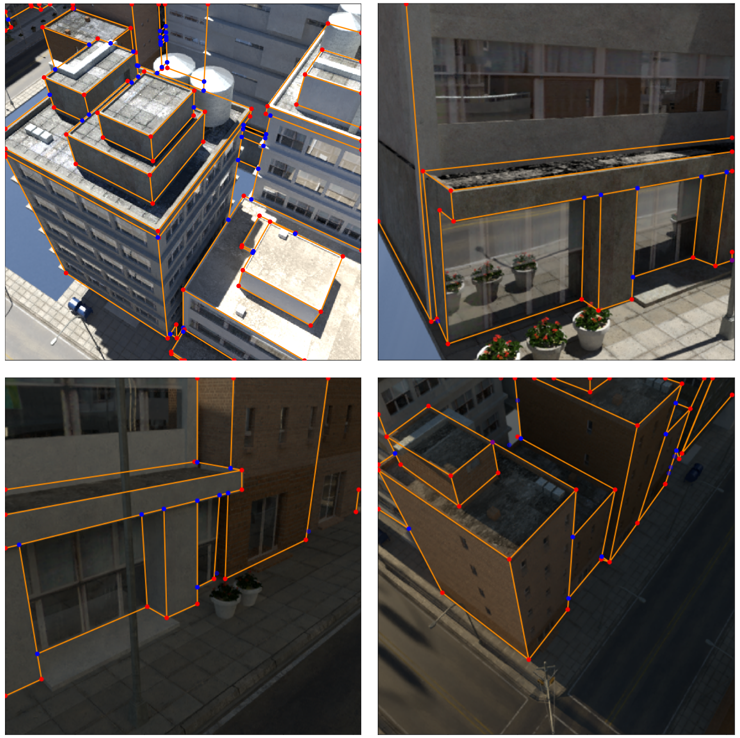
Experiment Settings

- ConicConv
  - Testing with different number of layers
    - **2 conic** convolution layers
    - **4 conic** convolution layers
    - **6 conic** convolution layers
- Line clustering baselines
  - **LSD + J-Linkage [1]**
  - **Contour + J-Linkage [2]** (Only for dominating vanishing point detection)
- Deep learning baselines:
  - Use the same number of parameters as 4x ConicConv
  - **REG**: directly regress the vanishing point coordinates
  - **CLS**: use vanishing point coordinates as features and do classification

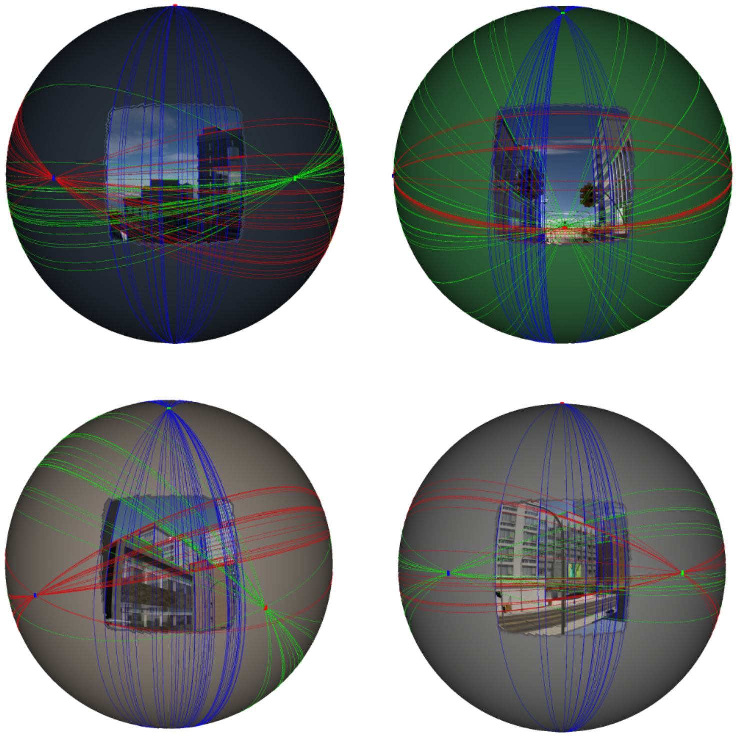
[1] “Semi-automatic 3D Reconstruction of Piecewise Planar Building Models From Single Image ” Chen Feng, Fei Deng, Vineet R. Kamat.

[2] “Detecting Dominant Vanishing Points in Natural Scenes with Application to Composition-Sensitive Image Retrieval” Zihan Zhou, Farshid Farhat, and James Z. Wang

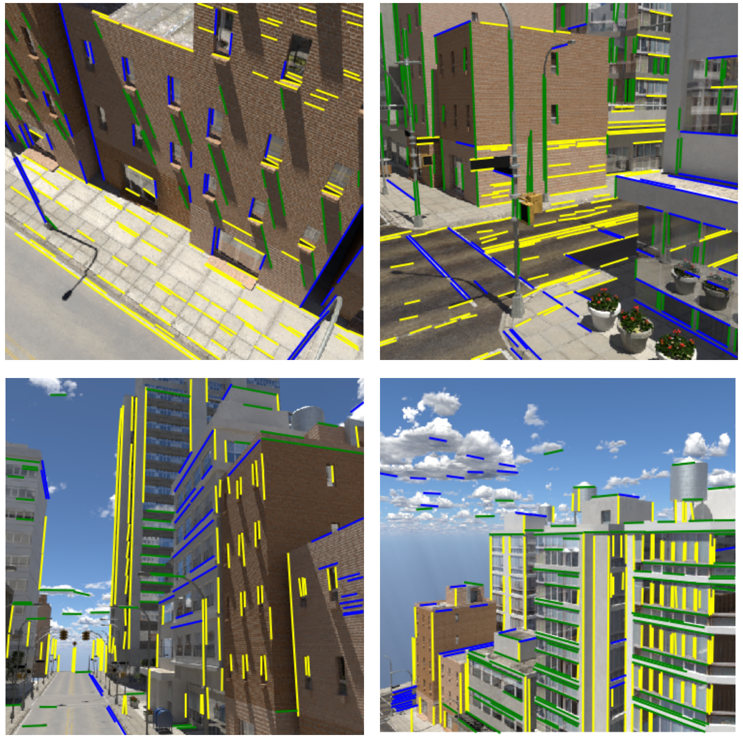
Synthetic Urban 3D Dataset [1] Visualization



Ground Truth Geometric Lines

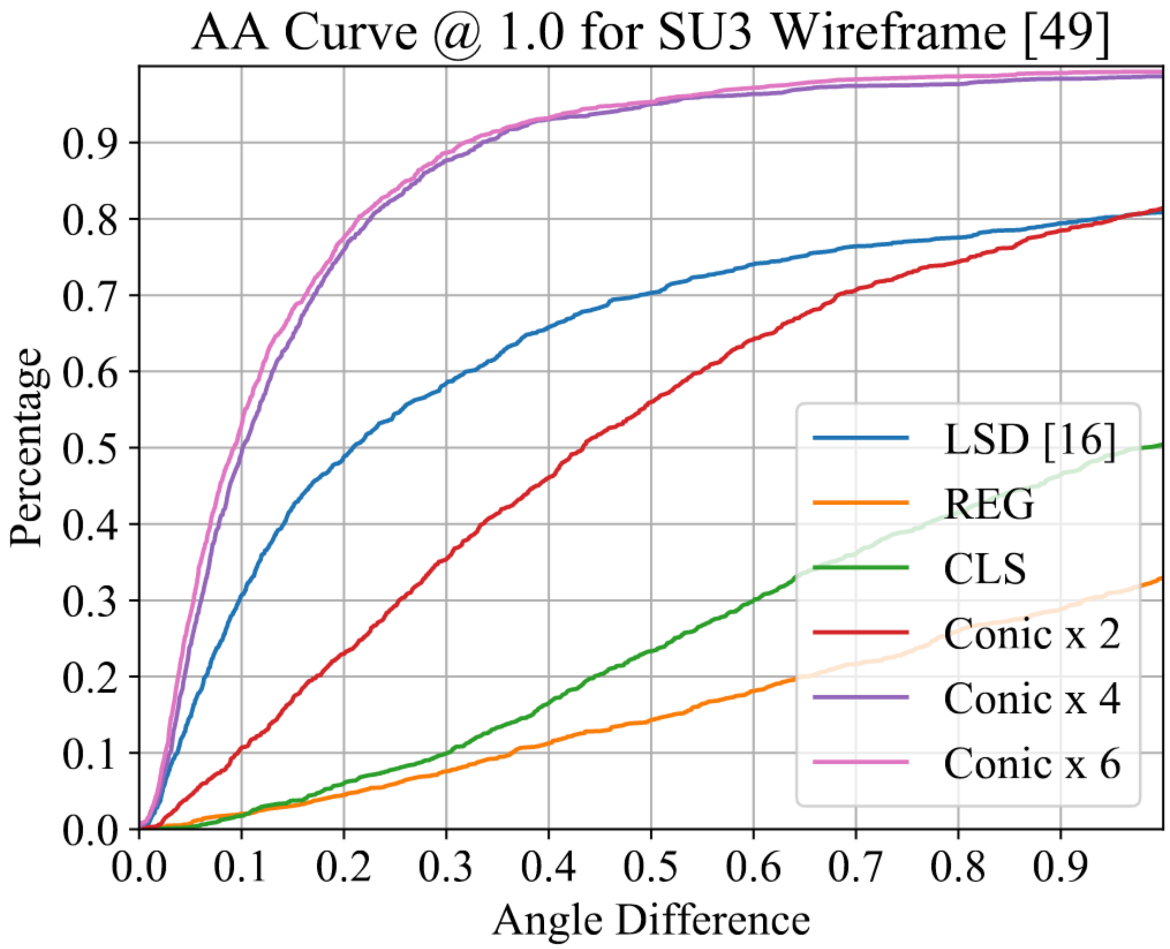


NeurVPS Results

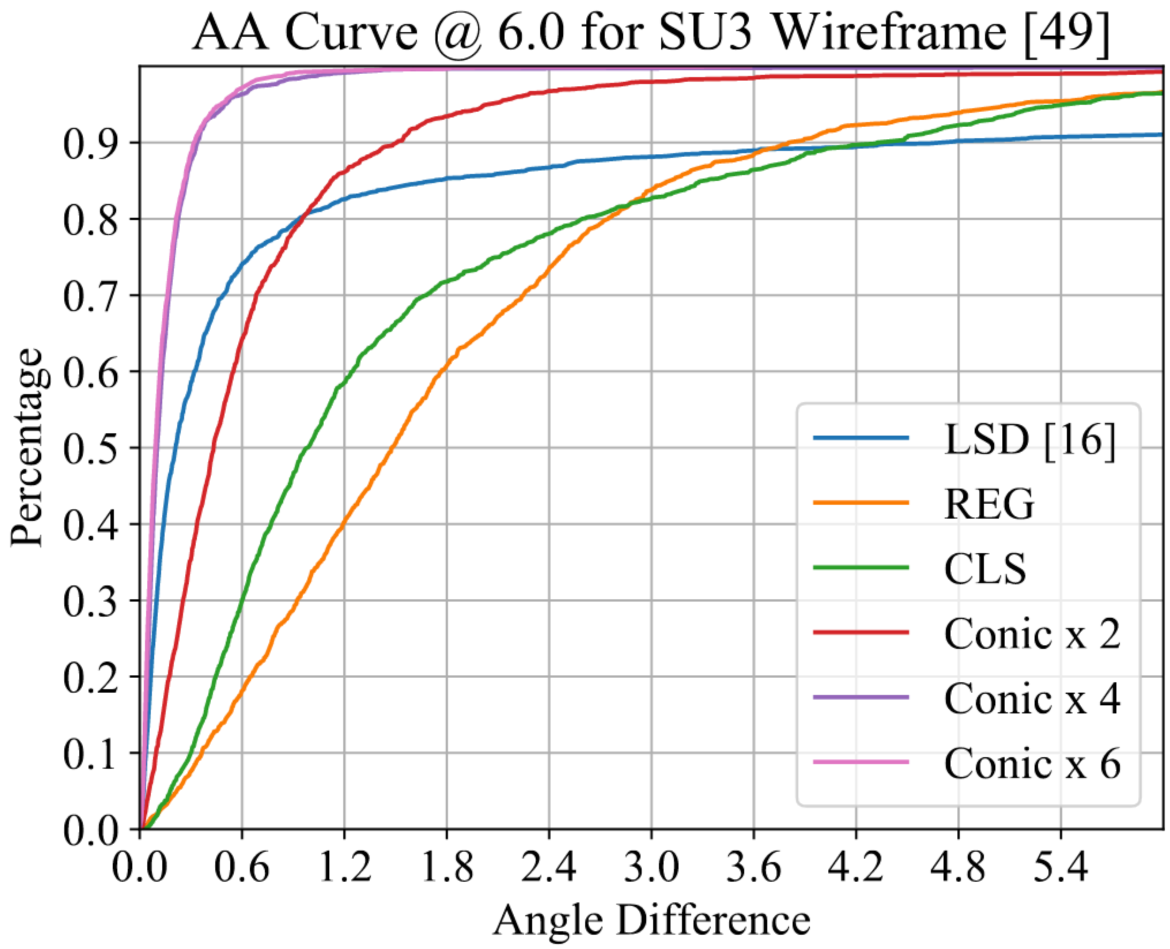


LSD + J-Linkage Results

[1] Zhou, Yichao, et al. "Learning to Reconstruct 3D Manhattan Wireframes from a Single Image." arXiv preprint arXiv:1905.07482 (2019).



(a) Angle difference ranges from 0° to 1°.

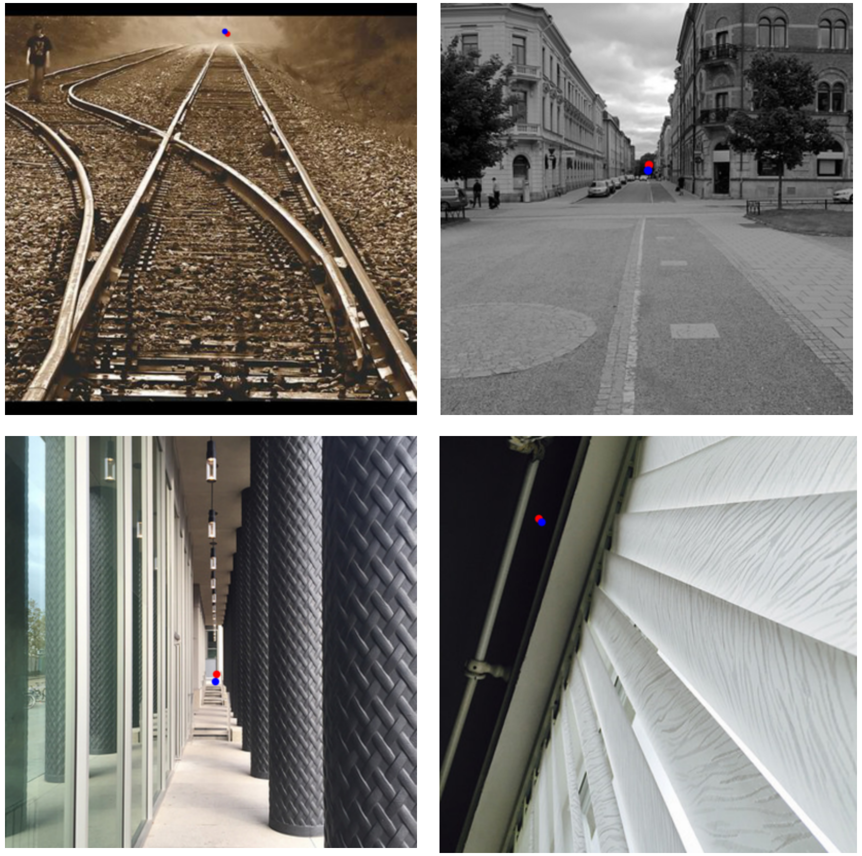


(b) Angle difference ranges from 0° to 6°.

Figure 5: Angle accuracy curves for different methods on the SU3 wireframe dataset [49].

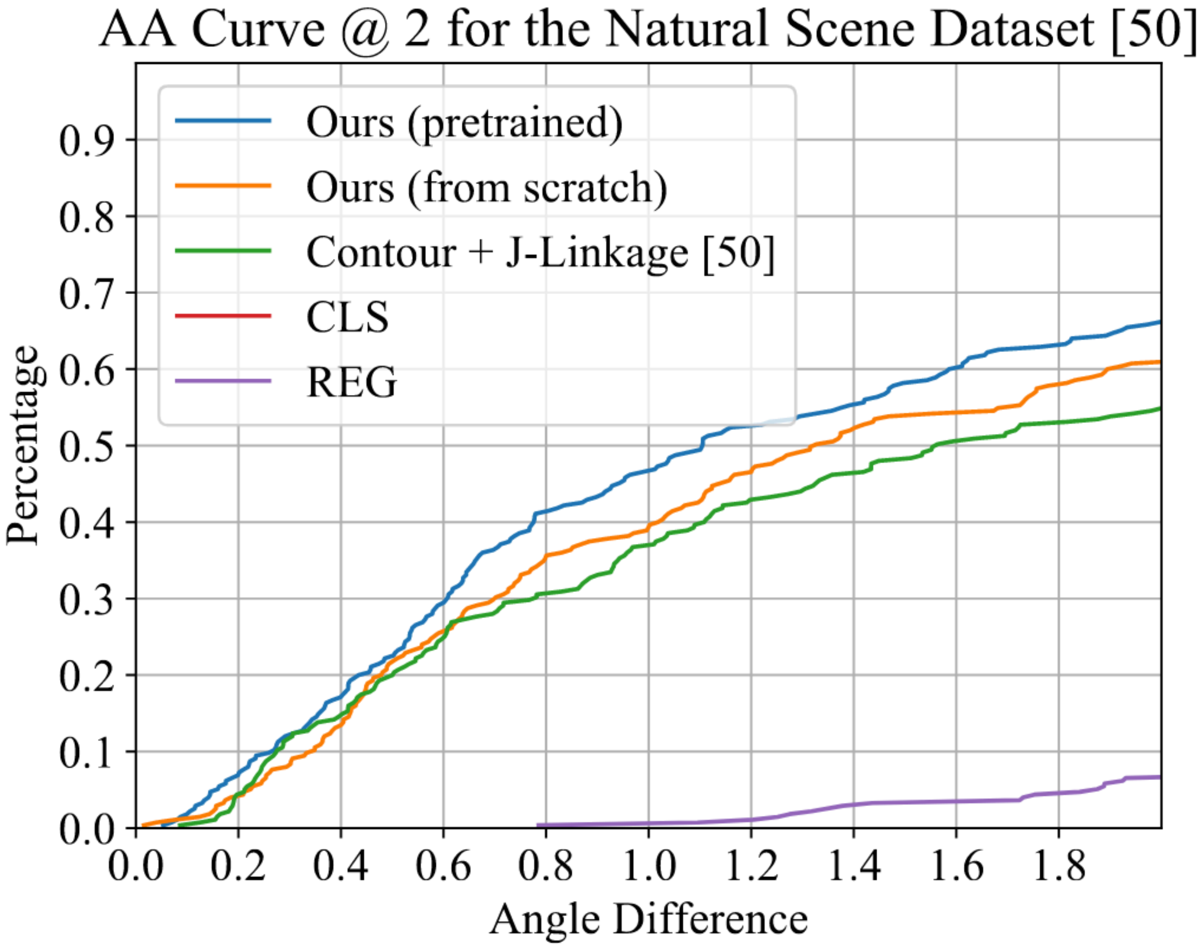
Natural Scene Dataset [1] Visualization

Labeled Ground Truth Lines for Vanishing Points

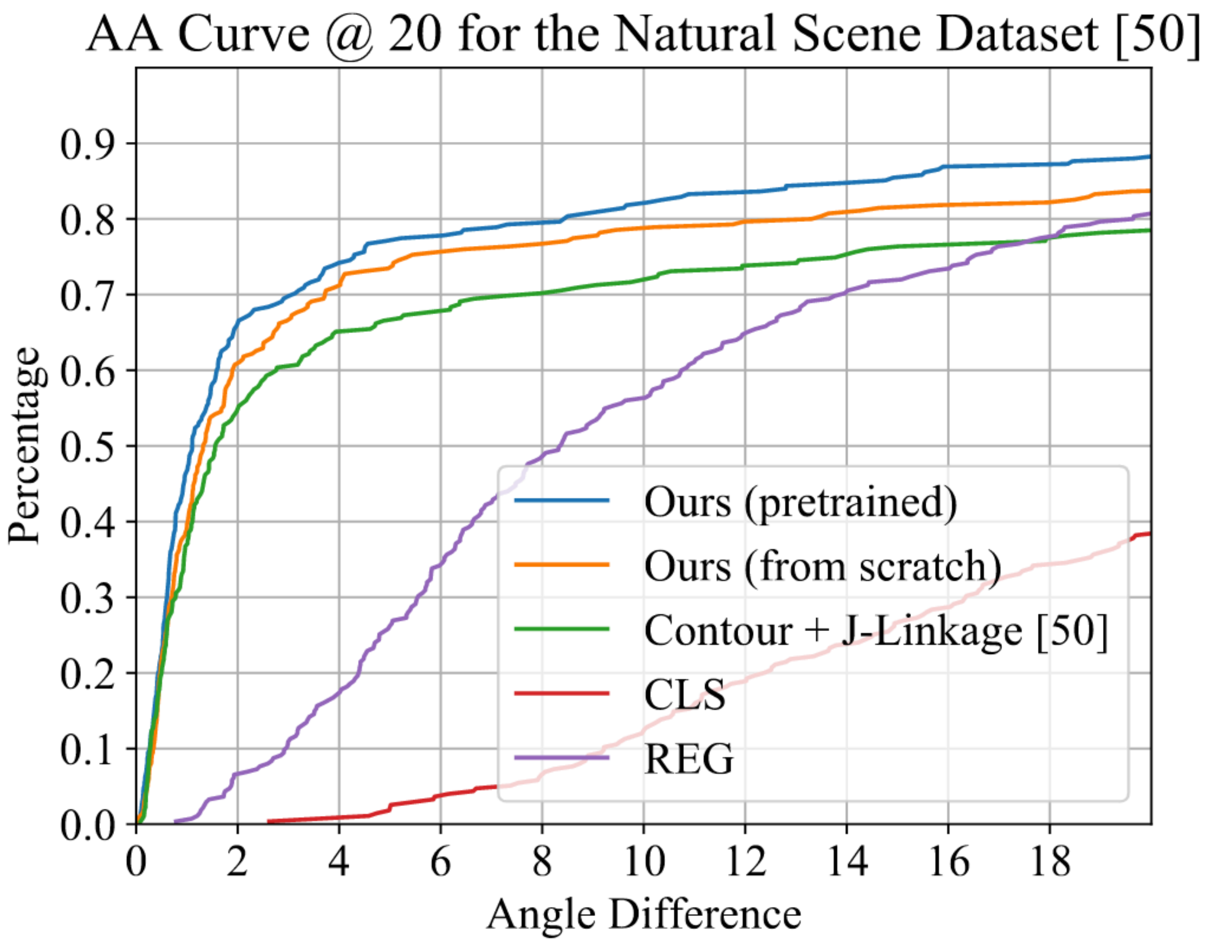


NeurVPS Results (Blue: Pred, Red: GT)

[1] Zhou, Zihan, Farshid Farhat, and James Z. Wang. "Detecting dominant vanishing points in natural scenes with application to composition-sensitive image retrieval." IEEE Transactions on Multimedia 19.12 (2017): 2651-2665.



(a) Angle difference ranges from 0° to 2°.

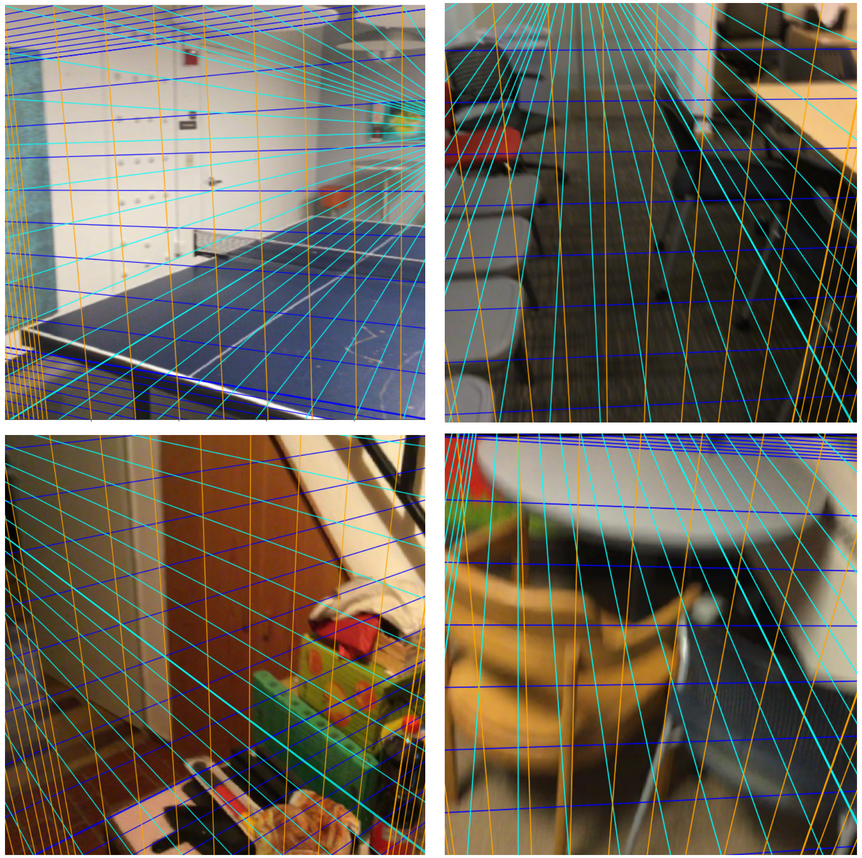


(b) Angle difference ranges from 0° to 20°.

Figure 8: Angle accuracy curves for different methods on the natural scene dataset [50].

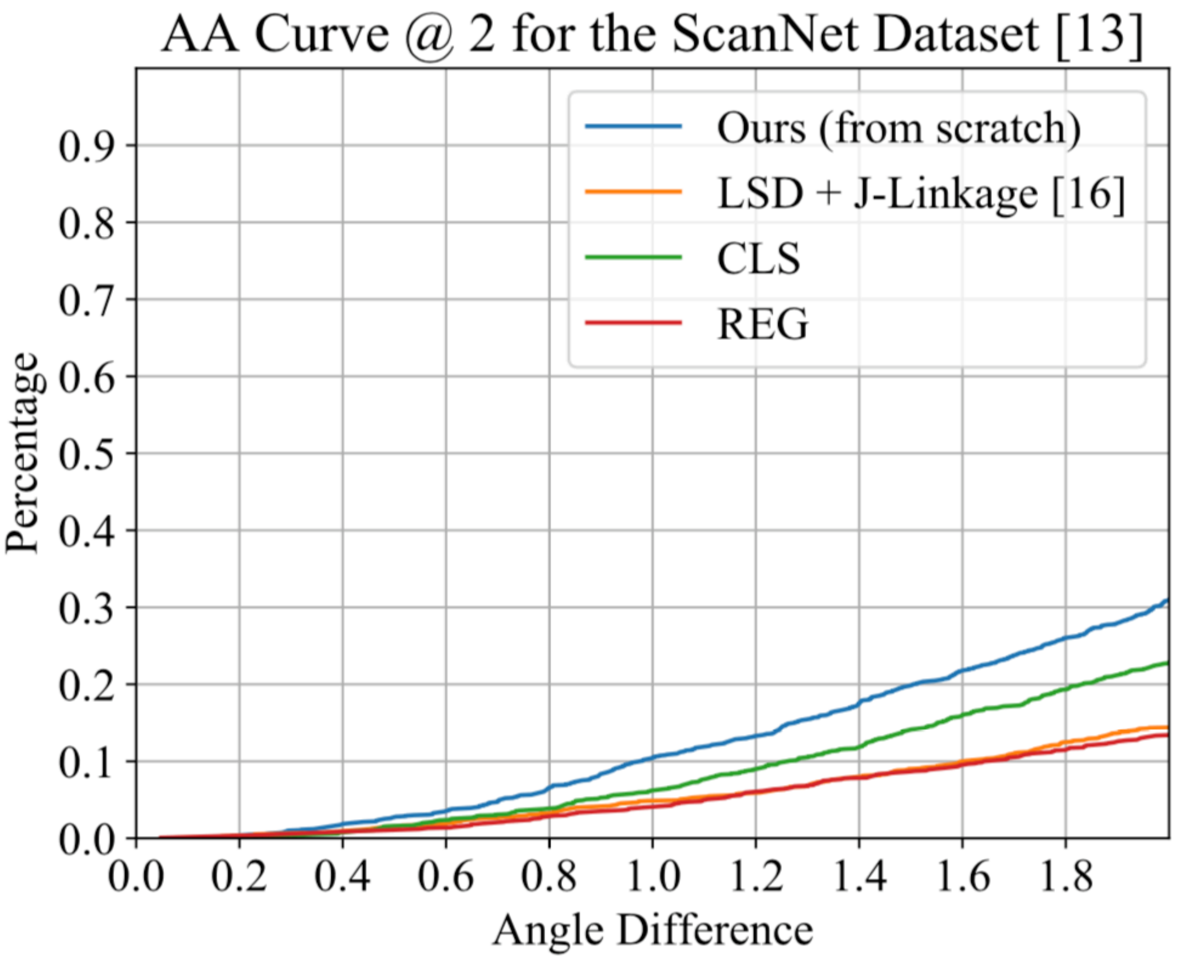
ScanNet Dataset [1] Visualization

ScanNet Image Samples

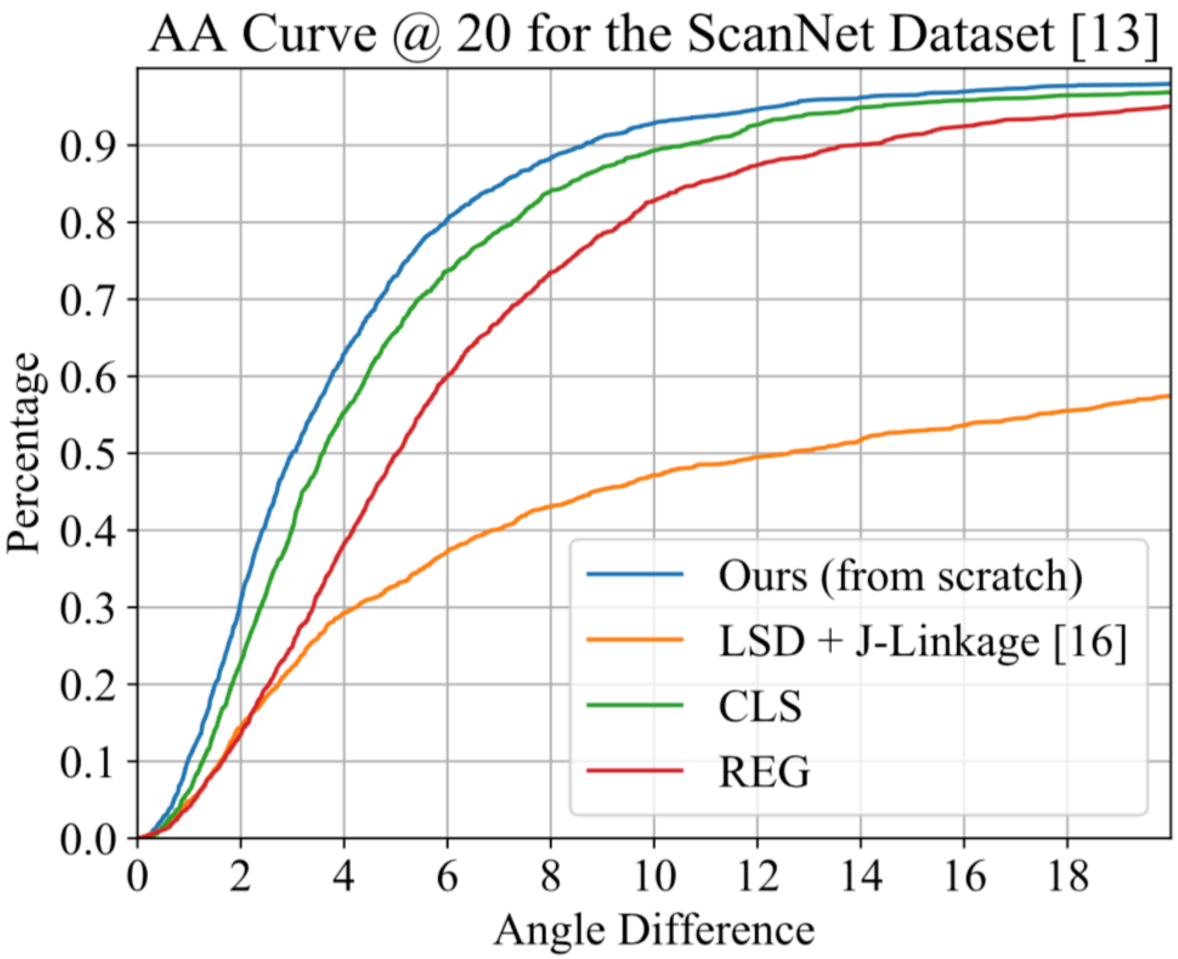


Ground Truth Vanishing Points

[1] Dai, Angela, et al. "Scannet: Richly-annotated 3d reconstructions of indoor scenes." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.



(a) Angle difference ranges from 0° to 2°.



(b) Angle difference ranges from 0° to 20°.

Figure 9: Angle accuracy curves for different methods on the ScanNet dataset [13].