Junction Detection and Grouping for Edge Classification Using Global Junction Analysis

Thesis Defense

Eugenia Levy Lozano

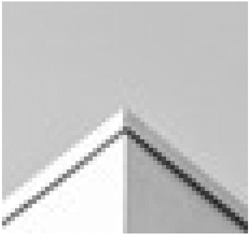
Information Systems Science and Engineering (ISSE)
Ritsumeikan University

February 1st, 2023

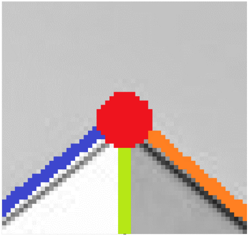
What is a junction?

- A *junction* is the point where two or more lines join.
- The lines which form a junction are called *branches*.

(a) Architectural Picture



(b) Zoomed Building Corner



(c) Highlighted Junction and Branches

Figure 1: A junction[1]

[1] Original image license: creative Commons BSD-3 License Agreement: https://opensource.org/licenses/BSD-3-Clause

Junction Types[3]

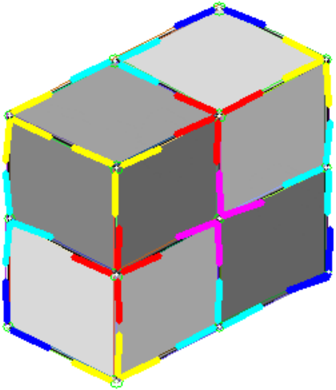


Figure 2: Junction Types - Ψ (Psi), T, X, Y, and L Junctions [2]

[2] Creative Commons Attribution-ShareAlike 4.0 International: https://creativecommons.org/licenses/by-sa/4.0/legalcode

[3] E. H. Adelson, "Lightness perception and lightness illusions," *The new cognitive neurosciences*, p. 339, 2000.

How are junctions useful?

- Junctions can be used to perform **edge classification**, which in turn can be used as a base for algorithms in tasks such as:
  - Image segmentation
  - Image depth estimation
  - Photogrammetry

Types of Edges

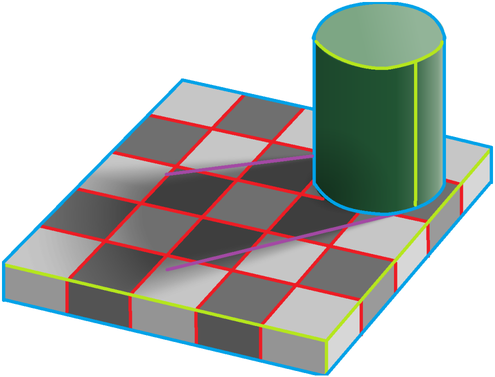


Figure 3: Edge Types - **Occlusion** (OB), **surface change** (SC), **reflectance change** (RC), and **illumination** edges.[4]

[4] Creative Commons Attribution-ShareAlike 4.0 International: https://creativecommons.org/licenses/by-sa/4.0/legalcode

Research Purpose

- Create an algorithm which:
  - **Detects** junctions
  - **Classifies** junctions
  - **Groups** junctions on the same lines or edges (novel)
  - Can perform **edge classification** using global junction analysis (novel)
- **Overall purpose:** Create an algorithm which can identify and group junctions to perform edge classification.

Junction Detection Methods

- Skeleton/topology extraction[5, 6]
- Optimization[7]
- Image derivatives (gradients/energy/edges)[8, 9, 10, 11]

---

[5] H. S. Al-Khaffaf and A. Z. Talib, "Three-stage junction detection in document images," in *2020 International Conference on Computer Science and Software Engineering (CSASE)*, 2020, pp. 142–145. DOI: 10.1109/CSASE48920.2020.9142083.

[6] T.-A. Pham, M. Delalandre, S. Barrat, *et al.*, "Accurate junction detection and characterization in line-drawing images," *Pattern Recognition*, vol. 47, no. 1, pp. 282–295, 2014, ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2013.06.027. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320313002793.

[7] D. Verbin and T. Zickler, "Field of junctions: Extracting boundary structure at low SNR," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 6869–6878.

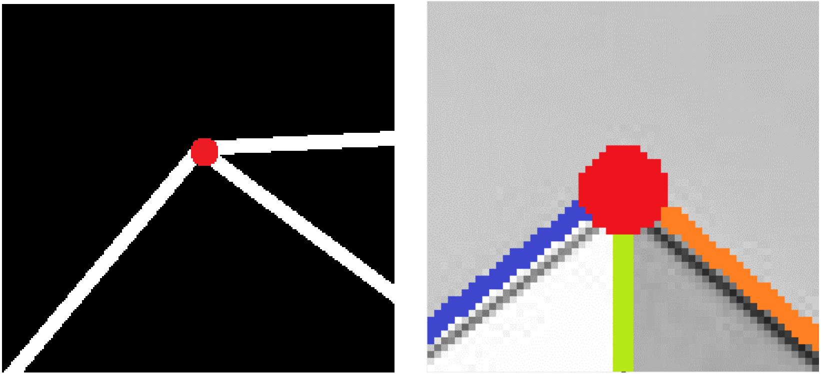
[8] R. Elias and R. Laganiere, "Judoca: Junction detection operator based on circumferential anchors," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2109–2118, 2012. DOI: 10.1109/TIP.2011.2175738.

[9] C. G. Harris and M. J. Stephens, "A combined corner and edge detector," in *Alvey Vision Conference*, 1988.

[10] M. Maire, P. Arbelaez, C. Fowlkes, *et al.*, "Using contours to detect and localize junctions in natural images," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8. DOI: 10.1109/CVPR.2008.4587420.

[11] G.-S. Xia, J. Delon, and Y. Gousseau, "Accurate junction detection and characterization in natural images," *International Journal of Computer Vision*, vol. 106, pp. 31–56, 2013.

Junction Detection Methods - Line Segment Intersection[12]



(a) Intersection of Line Segments (b) A junction

Figure 4: Intersecting Lines Example

[12] Z. Tu and X. Chen, "Junction detection based on line segments," in *2014 9th IEEE Conference on Industrial Electronics and Applications*, 2014, pp. 1231–1234. DOI: 10.1109/ICIEA.2014.6931354.

Edge Classification Methods

- *Local* statistics
  - Contrast[13, 14, 15]
  - B-Y and R-G differences [7]
  - Prior probability [6]
- Machine learning (CNNs)[7, 8, 16, 17]

[13] K. P. Vilankar, J. R. Golden, D. M. Chandler, *et al.*, "Local edge statistics provide information regarding occlusion and nonocclusion edges in natural scenes," *Journal of Vision*, vol. 14, no. 9, pp. 13–13, Aug. 2014, ISSN: 1534-7362. DOI: 10.1167/14.9.13. eprint: https://arvojournals.org/arvo/content_public/journal/jov/933550/i1534-7362-14-9-13.pdf. [Online]. Available: https://doi.org/10.1167/14.9.13.

[14] C. DiMattina, S. A. Fox, and M. S. Lewicki, "Detecting natural occlusion boundaries using local cues," *Journal of Vision*, vol. 12, no. 13, pp. 15–15, Dec. 2012, ISSN: 1534-7362. DOI: 10.1167/12.13.15. eprint: https://arvojournals.org/arvo/content_public/journal/jov/932801/i1534-7362-12-13-15.pdf. [Online]. Available: https://doi.org/10.1167/12.13.15.

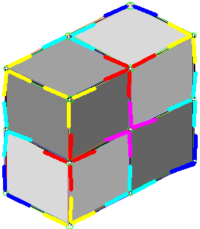
[15] C. DiMattina, J. Burnham, B. Guner, *et al.*, "Distinguishing shadows from surface boundaries using local achromatic cues," *bioRxiv*, 2022. DOI: 10.1101/2022.03.08.483480. eprint: https://www.biorxiv.org/content/early/2022/03/08/2022.03.08.483480.full.pdf. [Online]. Available: https://www.biorxiv.org/content/early/2022/03/08/2022.03.08.483480.

[16] K. A. Ehinger, E. W. Graf, W. J. Adams, *et al.*, "Local depth edge detection in humans and deep neural networks," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017, pp. 2681–2689. DOI: 10.1109/ICCVW.2017.316.

[17] M. Pu, Y. Huang, Q. Guan, *et al.*, *Rindnet: Edge detection for discontinuity in reflectance, illumination, normal and depth*, 2021. DOI: 10.48550/ARXIV.2108.00616. [Online]. Available: https://arxiv.org/abs/2108.00616.

Edge Classification Methods - Visual Psychology Foundations

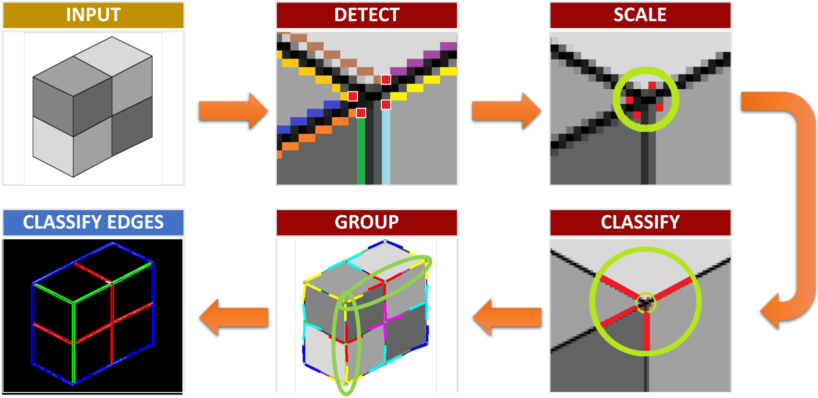
- **This (novel) approach: global junction analysis**[18]



Junction Type	Uses
L	Locate corners and OBs.
T	Roof → locate OBs. Spine → locate OBs or RCs.
Y	Locate corners and OBs or SCs.
X	Locate filters or OBs.
Ψ (Psi)	Arms → locate RCs. Spine → locate SCs.

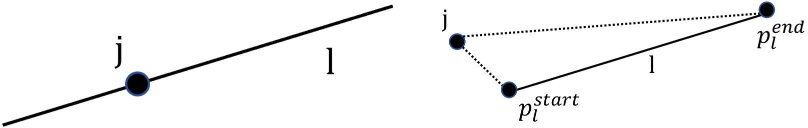
Table 1: Acronyms: OB → occlusion boundary; SC → surface change; RC → reflectance change.

[18] E. H. Adelson, "Lightness perception and lightness illusions," *The new cognitive neurosciences*, p. 339, 2000.


INPUT
DETECT
SCALE
CLASSIFY EDGES
GROUP
CLASSIFY

DETECT - Line Intersections

1. Detect line segments[19]
2. Calculate each line pair intersection
3. Check if the intersection is within threshold distance from line segment
4. Set these as **potential junctions**



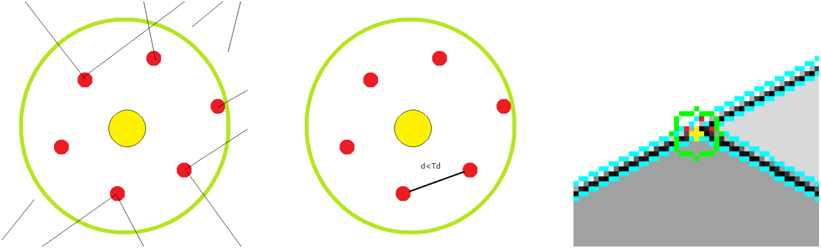
(a) Intersection on Segment   (b) Intersection Away from Segment

Figure 5: Measuring Distance From an Intersection to a Line Segment

[19] R. Grompone von Gioi, J. Jakubowicz, J.-M. Morel, *et al.*, "Lsd: A fast line segment detector with a false detection control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 722–732, 2010. DOI: 10.1109/TPAMI.2008.300.

SCALE - Clustering

1. Find distance between each junction candidate
2. Cluster all junctions within the distance threshold T_d **(user defined scale)**



(a) Intersection Cluster (Centroid in Yellow)
(b) Intersection Cluster (Distance)
(c) Clustering in Algorithm

Figure 6: Finding Branches Around Junctions

CLASSIFY - Classify Junctions Based on Branch Number and Angles

1. Check branch number N
2. Check branches for parallel pairs (same line)
3. **Classify** junctions according to the following table:

Junction type	Branch number	Parallel pairs
L	2	0
T	3	1
Y	3	0
X	4	2
Psi (Ψ)	4	1

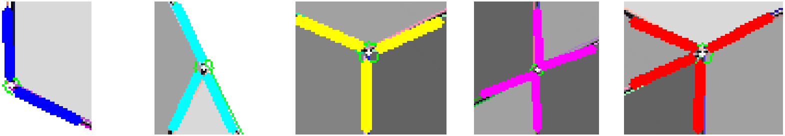
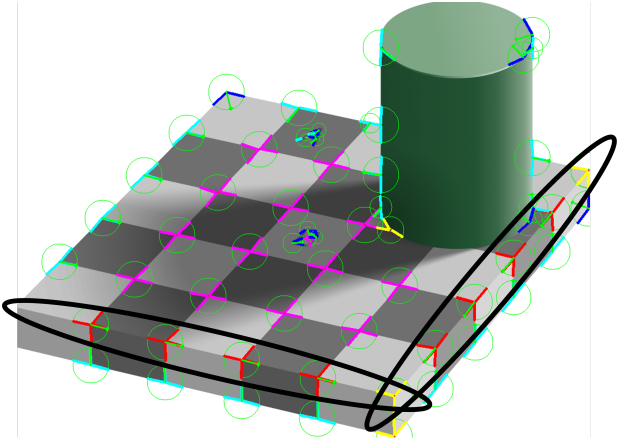


Figure 7: 5 Different Junction Types

GROUP - Group Junctions on the Same Edge/Line

1. **Group** *parallel* line segments within T_d of each other.
2. Set each group of lines as a single straight edge.



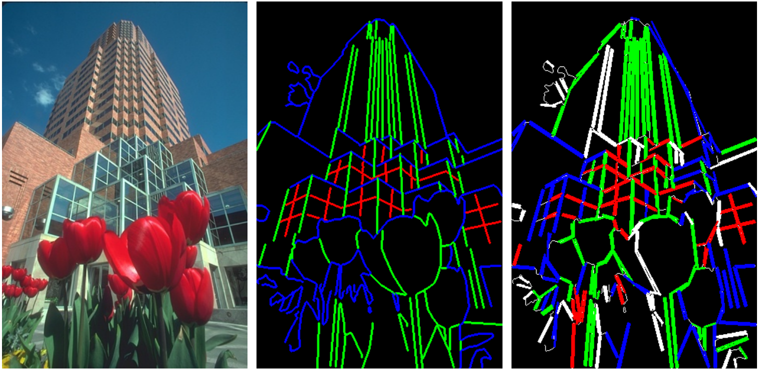
CLASSIFY EDGES

1. Check all junction and branch types associated to each edge.
2. Give a class score to each edge according to the table below.
3. Classify each edge as OB, RC or SC according to its highest class score.

Junction/Branch	Type	OB Score	RC Score	SC score
Junction	L	+1	0	+0.8
Junction	X	+0.8	0	0
Junction	Y	0	0	+1.3
Branch	T-arm	+1	0	0
Branch	T-spine	+0.8	+1	0
Branch	Ψ-arm	0	+1	0
Branch	Ψ-spine	0	0	+1
Branch	Inverse-Y-arm	+1	0	0
Branch	Inverse-Y-spine	0	0	+1

1. Introduction
2. Research Purpose
3. Literature Review
4. Methodology
5. **Results**
6. Conclusions

Architectural Image - Output



(a) Original Image (b) Ground Truth (c) Output

Figure 8: OB, SC, and RC edges. Image taken from the BSDS-RIND[20].

[20] M. Pu, Y. Huang, Q. Guan, *et al.*, *Rindnet: Edge detection for discontinuity in reflectance, illumination, normal and depth*, 2021. DOI: 10.48550/ARXIV.2108.00616. [Online]. Available: https://arxiv.org/abs/2108.00616.

Architectural Image - Performance Metrics

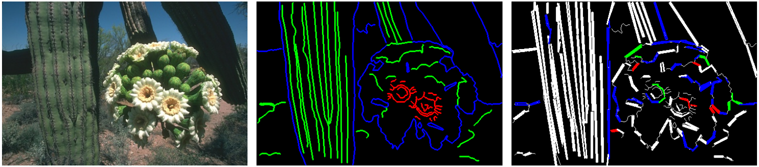
	Ground Truth				
Predicted Label		**OB**	**SC**	**RC**	**UC**
	OB	714	80	11	0
	SC	103	2118	0	0
	RC	30	5	466	0
	UC	76	60	0	0

Table 2: Confusion matrix

	Metric			
Class		**Precision**	**Recall**	**F1-score**
	OB	0.887	0.774	0.826
	SC	0.954	0.936	0.945
	RC	0.930	0.977	0.953

Table 3: Performance metrics

Natural Image - Output



(a) Original Image (b) Ground Truth (c) Output

Figure 9: **OB**, **SC**, and **RC** edges. Image taken from the BSDS-RIND[21].

[21] M. Pu, Y. Huang, Q. Guan, *et al.*, *Rindnet: Edge detection for discontinuity in reflectance, illumination, normal and depth*, 2021. DOI: 10.48550/ARXIV.2108.00616. [Online]. Available: https://arxiv.org/abs/2108.00616.

Natural Image - Performance Metrics

	Ground Truth				
Predicted Label		**OB**	**SC**	**RC**	**UC**
	OB	610	49	1	0
	SC	8	10	3	0
	RC	2	3	28	0
	UC	244	1152	7	0

Table 4: Confusion matrix

	Metric			
Class		**Precision**	**Recall**	**F1-score**
	OB	0.925	0.706	0.801
	SC	0.476	0.008	0.016
	RC	0.848	0.717	0.778

Table 5: Performance metrics

Indoor Image - Output

(a) Original Image (b) Ground Truth (c) Output

Figure 10: **OB**, **SC**, and **RC** edges. Image taken from the BSDS-RIND[22].

[22] M. Pu, Y. Huang, Q. Guan, *et al.*, *Rindnet: Edge detection for discontinuity in reflectance, illumination, normal and depth*, 2021. DOI: 10.48550/ARXIV.2108.00616. [Online]. Available: https://arxiv.org/abs/2108.00616.

Indoor Image - Performance Metrics

	Ground Truth				
Predicted Label		**OB**	**SC**	**RC**	**UC**
	OB	1577	101	96	0
	SC	43	504	29	0
	RC	23	16	920	0
	UC	71	32	9	0

Table 6: Confusion matrix

	Metric			
Class		**Precision**	**Recall**	**F1-score**
	OB	0.889	0.920	0.904
	SC	0.875	0.772	0.820
	RC	0.959	0.873	0.914

Table 7: Performance metrics

Average Performance Metrics

- 100 images from the BSDS-RIND[23]
- Mostly outdoor images with architectural subjects.

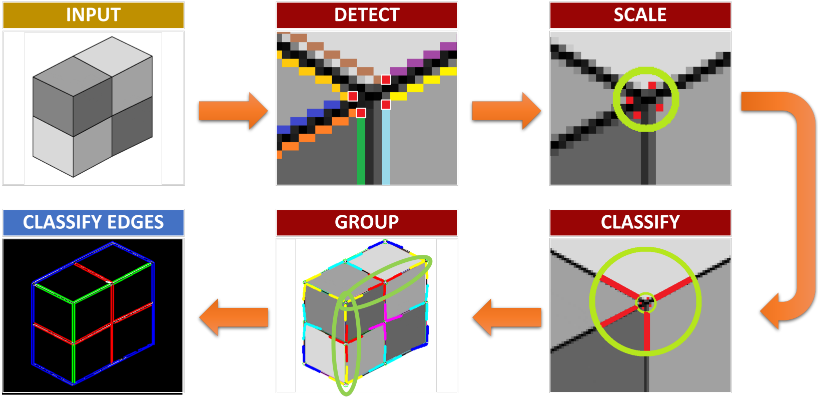
	Metric			
Class		**Precision**	**Recall**	**F1-score**
	OB	0.750	0.863	0.803
	SC	0.805	0.536	0.643
	RC	0.638	0.562	0.598

[23] M. Pu, Y. Huang, Q. Guan, *et al.*, *Rindnet: Edge detection for discontinuity in reflectance, illumination, normal and depth*, 2021. DOI: 10.48550/ARXIV.2108.00616. [Online]. Available: https://arxiv.org/abs/2108.00616.

Summary

- Algorithm: **detects**, **classifies**, and **groups** junctions to perform **edge classification**.
- Purpose: **Classify edges** using *global* junction analysis.
  - Applications: Image depth estimation, image segmentation, photogrammetry, etc.
- Contributions: Showed junctions could be used as the sole indicator for edge classification.
- Future work: Focus on numerical junction weights and interactions from an engineering stand point.

Q&A



Extended Slides

Introduction - Image Segmentation

(a) Original Picture

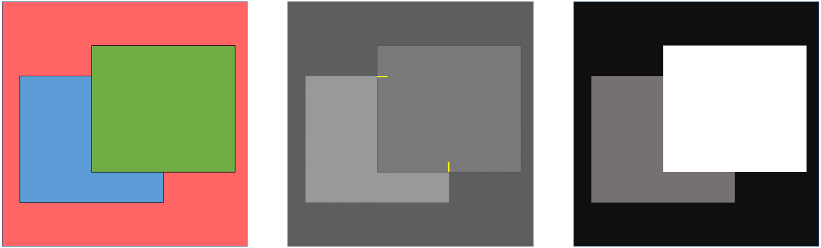
(b) Segmented Image

Figure 11: Image Segmentation[24, 25]

[24] B. Zhou, H. Zhao, X. Puig, *et al.*, "Scene parsing through ade20k dataset," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017.

[25] Creative Commons BSD-3 License Agreement: https://opensource.org/licenses/BSD-3-Clause

Introduction - Image Depth Estimation[26]



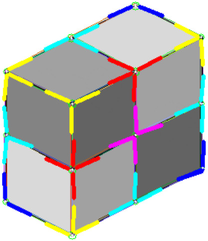
(a) Original Picture (b) T-Junctions (c) Depth Ordering

Figure 12: Using Junctions for Depth Estimation

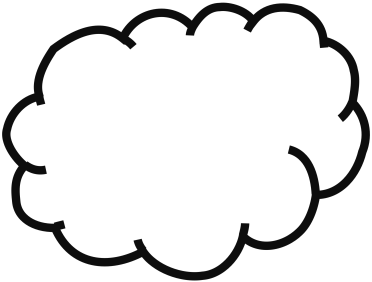
[26] M. Dimiccoli and P. Salembier, "Exploiting t-junctions for depth segregation in single images," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 1229–1232. DOI: 10.1109/ICASSP.2009.4959812.

Literature Review - Visual Psychology - Junctions: Types and Uses [1] [20]

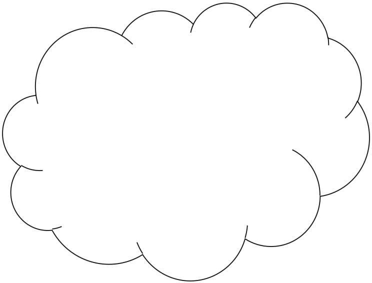
- Ψ (Psi) junctions: Spine -> Luminance boundaries; Arms -> Reflectance boundaries
- T junctions: Top -> Occlusion boundaries; Spine -> Reflectance or occlusion boundaries [17] [18]
- X junctions: Reflectance boundaries or filters [19]
- Y junctions: Occlusion or surface-change boundaries
- L junctions: Occlusion boundaries [18]



Literature Review - Junction Detection - Skeleton/Topology Extraction[27, 28]



(a) Original Image



(b) Skeleton

Figure 13: Skeleton Extraction Example

[27] H. S. Al-Khaffaf and A. Z. Talib, "Three-stage junction detection in document images," in *2020 International Conference on Computer Science and Software Engineering (CSASE)*, 2020, pp. 142–145. DOI: 10.1109/CSASE48920.2020.9142083.

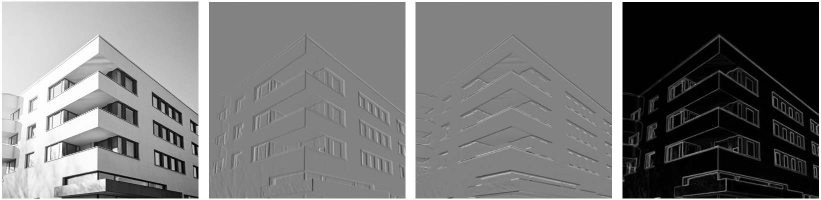
[28] T.-A. Pham, M. Delalandre, S. Barrat, *et al.*, "Accurate junction detection and characterization in line-drawing images," *Pattern Recognition*, vol. 47, no. 1, pp. 282–295, 2014, ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2013.06.027. [Online]. Available:

Literature Review - Junction Detection - Optimization[29]

- Assuming three-branch junction existence in image patches
- Optimizing junction -
    - Angles
    - Position
- Using gradient descent

[29] D. Verbin and T. Zickler, "Field of junctions: Extracting boundary structure at low SNR," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 6869–6878.

Literature Review - Junction Detection - Image Derivatives (Gradients/Energy/Edges)[30, 31, 32, 33]



(a) Original Image (b) X-Derivative (c) Y-Derivative (d) Magnitude

Figure 14: Image Derivative Computation Example

[30] R. Elias and R. Laganiere, "Judoca: Junction detection operator based on circumferential anchors," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2109–2118, 2012. DOI: 10.1109/TIP.2011.2175738.

[31] C. G. Harris and M. J. Stephens, "A combined corner and edge detector," in *Alvey Vision Conference*, 1988.

[32] M. Maire, P. Arbelaez, C. Fowlkes, *et al.*, "Using contours to detect and localize junctions in natural images," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8. DOI: 10.1109/CVPR.2008.4587420.

[33] G.-S. Xia, J. Delon, and Y. Gousseau, "Accurate junction detection and characterization in natural images," *International Journal of Computer Vision*, vol. 106, pp. 31–56, 2013.

Literature Review - Pre-Processing - Edge Detection

Figure 15: Multi-Scale Edge Detector [21]

- Keeps detail
- Finds shadow edges
- Keeps noise

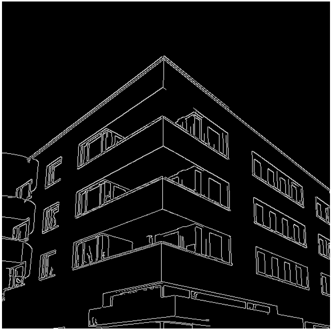


Figure 16: Canny Edge Detector [22]

- Cleaner edges
- Parameters have to be adjusted
- Provides better control

Literature Review - Pre-Processing - Line Segment Detection

Figure 17: MCMLSD [23]

- Best to group long edges
- Can fail to reach corners
- Slowest

Figure 18: LSD [15]

- Best to find short segments
- Could detect redundant lines
- Fastest

Figure 19: PHT [24]

- Easily adjusted
- Can fail to reach corners
- Detects redundant lines

Methodology - Pre-processing

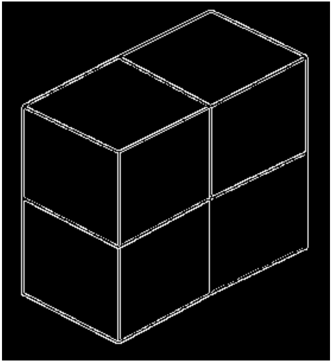


Figure 20: Edge Detection (Canny)[22]

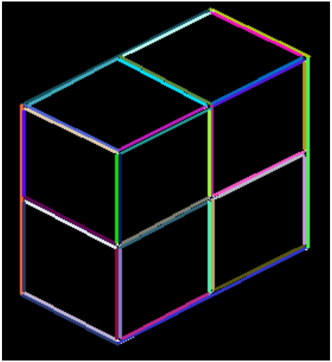
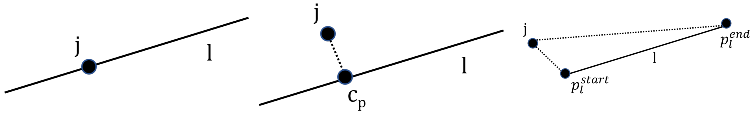


Figure 21: Line Segment Detection (LSD)[15]

Methodology - Line Intersection

1. Calculate each line pair intersection
2. Calculate the belonging distance b_d from each intersection j with coordinates (m, n) to the line segment l defined by the endpoints p_l^{start} and p_l^{end} of a line $Ax + By + C = 0$ whose closest point to j is c_p and set as a junction candidate if $b_d < T_d$

$$b_d(j,l) = \begin{cases} 0, & \text{if } j \text{ is on } l, \\ \dfrac{|Am + Bn + C|}{\sqrt{A^2 + B^2}}, & \text{if } c_p \text{ is on } l \\ min(d(j, p_l^{start}), d(j, p_l^{end})), & \text{otherwise} \end{cases} \tag{1}$$

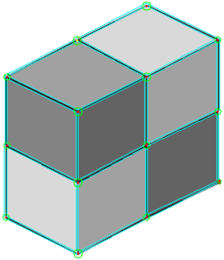


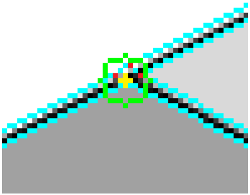
Methodology - Clustering

1. Find distance between each junction candidate using *Euclidean distance*:

$$d(p, q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2}$$

2. Cluster all junctions within a distance threshold T_d
3. Find the cluster's centroid j
4. Find the furthest junction candidate p_f from the centroid
5. Set the junction as the circle c_j of radius r_j where $r_j = d(j, p_f)$



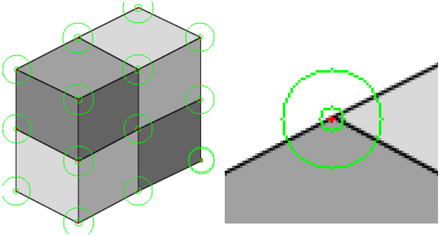


Methodology - Computing Search Range

1. Compute the search range r_l between each j and its N related lines $\{l_i\}_{i=1}^{N}$ (considering len_i as the length of the ith line segment)

$$r_{l_i}(j, l_i) = \begin{cases} max(T_d, min(d(j, p_l^{start}), d(j, p_l^{end}))), & \text{if } j \text{ is not on } l, \\ len_i, & \text{otherwise} \end{cases} \tag{2}$$

2. Set the search range r for each junction as $r = j_r + min(r_{l_i}, ..., r_{l_N}, T_r)$ using the threshold T_r to avoid



Methodology - Finding and Merging Branches (1)

1. Set a search range to look for branches:
   - Check distance from intersections to segment end points
2. Find edge paths connected from the junction to the circumference of the search range
3. Set all junction candidates with two or more branches as **detected junctions**

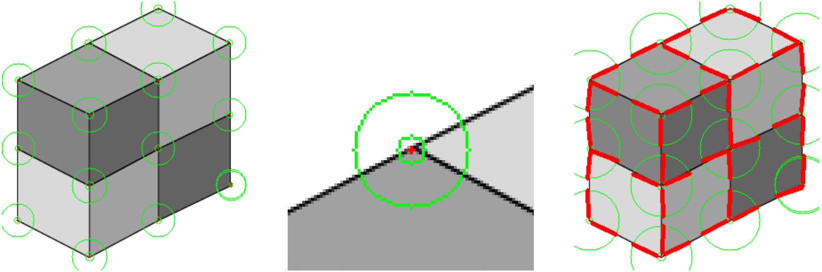
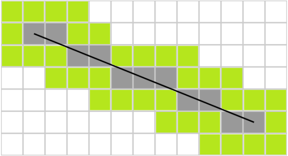
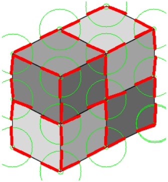


Figure 22: Finding Branches Around Junctions

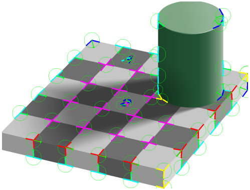
Methodology - Finding and Merging Branches (2)

1. Create a search path from each **edge** point on a circumference of radius r with center j to the closest edge point on c_j using Bresenham's algorithm [25]
2. For each point in the path, append said point to the branch candidate if any pixels in its pixel neighborhood are edges.
3. After going through all points, if the branch is longer than $(r - j_r) \times 0.8$ and it is accepted as a branch of j.
4. When all branches are found, if two or more branches are separated by an angle T_a or less, they are compared to a line fitted to their points and only the closest branch to it is kept.

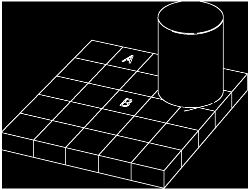



Results - Synthetic Images

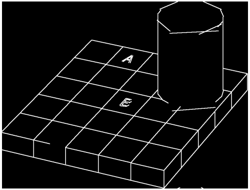
- Most junctions (scale-wise) are correctly identified



(a) Detected Junctions



(b) Edge Map



(c) Line Segments

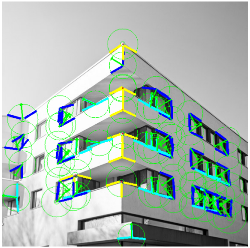
Figure 23: Results on a Synthetic Image (by Edward Adelson[34, 35])

[34] E. H. Adelson, "Lightness perception and lightness illusions," *The new cognitive neurosciences*, p. 339, 2000.

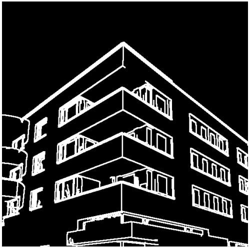
[35] Creative Commons Attribution-ShareAlike 4.0 International: https://creativecommons.org/licenses/by-sa/4.0/legalcode

Results - Architectural Images

- Large scale junctions are correctly identified but some are misclassified



(a) Detected Junctions



(b) Edge Map

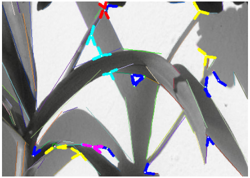
(c) Line Segments

Figure 24: Results on an Architectural Image[36]

[36] Creative Commons BSD-3 License Agreement: https://opensource.org/licenses/BSD-3-Clause

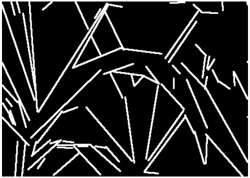
Results - Natural Images

- Many false negatives and misclassified junctions



(a) Detected Junctions

(b) Edge Map



(c) Line Segments

Figure 25: Results on a Natural Image[37]

Results - Junction Detection Discussion

- Natural images have the worst results
  - This algorithm is **purpose specific** - looking to group and analyze junctions on straight edges.
- Architectural images deal with some misclassification.
  - However, analysis might still work depending on the specific task given accurate junction grouping.
- Synthetic images have the best results.
  - However, smaller-scale junctions still suffer from misclassification and false negatives.
  - Works best on images (and objects) with long straight edges.

References I

[1] E. H. Adelson, "Lightness perception and lightness illusions," *The new cognitive neurosciences*, p. 339, 2000.

[2] H. S. Al-Khaffaf and A. Z. Talib, "Three-stage junction detection in document images," in *2020 International Conference on Computer Science and Software Engineering (CSASE)*, 2020, pp. 142–145. DOI: 10.1109/CSASE48920.2020.9142083.

[3] T.-A. Pham, M. Delalandre, S. Barrat, and J.-Y. Ramel, "Accurate junction detection and characterization in line-drawing images," *Pattern Recognition*, vol. 47, no. 1, pp. 282–295, 2014, ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2013.06.027. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320313002793.

References II

[4] D. Verbin and T. Zickler, "Field of junctions: Extracting boundary structure at low SNR," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 6869–6878.

[5] R. Elias and R. Laganiere, "Judoca: Junction detection operator based on circumferential anchors," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2109–2118, 2012. DOI: 10.1109/TIP.2011.2175738.

[6] C. G. Harris and M. J. Stephens, "A combined corner and edge detector," in *Alvey Vision Conference*, 1988.

References III

[7] M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik, “Using contours to detect and localize junctions in natural images,” in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8. DOI: 10.1109/CVPR.2008.4587420.

[8] G.-S. Xia, J. Delon, and Y. Gousseau, “Accurate junction detection and characterization in natural images,” *International Journal of Computer Vision*, vol. 106, pp. 31–56, 2013.

[9] Z. Tu and X. Chen, “Junction detection based on line segments,” in *2014 9th IEEE Conference on Industrial Electronics and Applications*, 2014, pp. 1231–1234. DOI: 10.1109/ICIEA.2014.6931354.

References IV

[10] K. P. Vilankar, J. R. Golden, D. M. Chandler, and D. J. Field, "Local edge statistics provide information regarding occlusion and nonocclusion edges in natural scenes," *Journal of Vision*, vol. 14, no. 9, pp. 13–13, Aug. 2014, ISSN: 1534-7362. DOI: 10.1167/14.9.13. eprint: https://arvojournals.org/arvo/content_public/journal/jov/933550/i1534-7362-14-9-13.pdf. [Online]. Available: https://doi.org/10.1167/14.9.13.

References V

[11] C. DiMattina, S. A. Fox, and M. S. Lewicki, "Detecting natural occlusion boundaries using local cues," *Journal of Vision*, vol. 12, no. 13, pp. 15–15, Dec. 2012, ISSN: 1534-7362. DOI: 10.1167/12.13.15. eprint: https://arvojournals.org/arvo/content_public/journal/jov/932801/i1534-7362-12-13-15.pdf. [Online]. Available: https://doi.org/10.1167/12.13.15.

References VI

[12] C. DiMattina, J. Burnham, B. Guner, and H. Yerxa, "Distinguishing shadows from surface boundaries using local achromatic cues," *bioRxiv*, 2022. DOI: 10.1101/2022.03.08.483480. eprint: https://www.biorxiv.org/content/early/2022/03/08/2022.03.08.483480.full.pdf. [Online]. Available: https://www.biorxiv.org/content/early/2022/03/08/2022.03.08.483480.

[13] K. A. Ehinger, E. W. Graf, W. J. Adams, and J. H. Elder, "Local depth edge detection in humans and deep neural networks," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017, pp. 2681–2689. DOI: 10.1109/ICCVW.2017.316.

References VII

[14] M. Pu, Y. Huang, Q. Guan, and H. Ling, *Rindnet: Edge detection for discontinuity in reflectance, illumination, normal and depth*, 2021. DOI: 10.48550/ARXIV.2108.00616. [Online]. Available: https://arxiv.org/abs/2108.00616.

[15] R. Grompone von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: A fast line segment detector with a false detection control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 722–732, 2010. DOI: 10.1109/TPAMI.2008.300.

[16] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017.

References VIII

[17] M. Dimiccoli and P. Salembier, "Exploiting t-junctions for depth segregation in single images," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 1229–1232. DOI: 10.1109/ICASSP.2009.4959812.

[18] J. McDermott and E. H. Adelson, "Junctions and cost functions in motion interpretation," *Journal of Vision*, vol. 4, no. 7, pp. 3–3, Feb. 2004, ISSN: 1534-7362. DOI: 10.1167/4.7.3. eprint: https://arvojournals.org/arvo/content_public/journal/jov/933506/jov-4-7-3.pdf. [Online]. Available: https://doi.org/10.1167/4.7.3.

References IX

[19] B. L. Anderson, "A theory of illusory lightness and transparency in monocular and binocular images: The role of contour junctions," en, *Perception*, vol. 26, no. 4, pp. 419–453, 1997.

[20] J. McDermott, "Psychophysics with junctions in real images," *Perception*, vol. 33, no. 9, pp. 1101–1127, 2004.

[21] J. H. Elder and S. W. Zucker, "Local scale control for edge detection and blur estimation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 20, no. 7, pp. 699–716, 1998.

References X

[22] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986. DOI: 10.1109/TPAMI.1986.4767851.

[23] E. J. Almazan, R. Tal, Y. Qian, and J. H. Elder, "Mcmlsd: A dynamic programming approach to line segment detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2031–2039.

[24] R. S. Stephens, "A probabilistic approach to the hough transform," in *Proceedings of the British Machine Vision Conference*, doi:10.5244/C.4.12, BMVA Press, 1990, pp. 12.1–12.6.

References XI

[25] J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Systems Journal*, vol. 4, no. 1, pp. 25–30, 1965. DOI: 10.1147/sj.41.0025.