

Junction Detection and Grouping for Edge Classification Using Global Junction Analysis

Eugenia Levy Lozano

2600190433-6

Supervisors: Damon M. Chandler & Nicko R. Caluya

Graduation Research 3 (GF)

Department of Information Science and Engineering

Information Systems Science and Engineering Course

Ritsumeikan University

Fall Semester, 2023

Abstract

The human visual system uses information provided by junctions to obtain meaning from its observations. This research focuses on both the detection and grouping of junctions for edge classification. The author proposes that junctions can complement each other to obtain meaning from scenes instead of utilizing statistical analysis to complete said task (going from a local to a global analysis). The results of this thesis indicate that although junctions can indeed complement each other, junction detection and classification must be highly accurate to enable a useful global analysis.

Finally, the author encourages researchers to delve deeper into the numerical properties on junction interaction as perhaps these properties will allow for the creation of a more robust system which is unaffected by noise and shadows. The main contributions of this paper pertain to the exploration of the human visual system and the interactions of junctions themselves.

Keywords: mid-level vision, junctions, global junction analysis, edge classification, edges, line-segment detection

Contents

| | |
|--|-------------|
| List of Figures | vi |
| List of Tables | viii |
| 1 Introduction | 1 |
| 1.1 The Role of Junctions in Visual Psychology | 1 |
| 1.2 Junctions and Their Uses | 4 |
| 1.2.1 What are Junctions? | 4 |
| 1.2.2 Types of Junctions | 4 |
| 1.2.3 Global Analysis of Junctions | 7 |
| 1.3 Research Problem and Purpose | 7 |
| 1.4 Thesis Overview | 8 |
| 1.4.1 Research Contributions | 8 |
| 1.4.2 Thesis Structure | 8 |
| 2 Literature Review | 9 |
| 2.1 Junctions at Different Levels of Visual Processing | 9 |
| 2.2 Edge Detection and Classification | 10 |
| 2.2.1 Edge Detection | 10 |
| 2.2.2 Edge Classification | 10 |
| 2.3 Line Segment Detection | 12 |
| 2.4 Approaches to Junction Detection | 12 |
| 2.4.1 Skeleton Extraction | 12 |
| 2.4.2 Probabilities | 13 |
| 2.4.3 Machine Learning | 13 |
| 2.4.4 Line Segment Detection | 14 |
| 3 Methodology | 15 |
| 3.1 Preprocessing | 16 |

| | | |
|----------|---|-----------|
| 3.2 | Candidate Junction Detection | 18 |
| 3.2.1 | Line Intersections | 18 |
| 3.2.2 | Belonging Distance | 18 |
| 3.3 | Junction Clustering | 19 |
| 3.4 | Junction Scaling | 20 |
| 3.4.1 | Computing Search Range | 20 |
| 3.4.2 | Finding and Merging Branches | 21 |
| 3.5 | Junction and Branch Classification | 23 |
| 3.6 | Finding Junction Direction | 25 |
| 3.7 | Straight Edge Detection and Junction Grouping | 27 |
| 3.8 | Global Junction Analysis Testing: Edge Classification | 27 |
| 3.9 | Parameters | 28 |
| 4 | Results | 29 |
| 4.1 | Empirical Junction Detection Results | 29 |
| 4.2 | Edge Classification Results | 31 |
| 4.3 | Performance Metrics | 31 |
| 4.3.1 | Data Set | 32 |
| 4.3.2 | Edge Classification Case Studies | 32 |
| 4.3.2.1 | Case Study 1: Architectural Image | 32 |
| 4.3.2.2 | Case Study 2: Natural Image | 34 |
| 4.3.2.3 | Case Study 3: Indoor Image | 37 |
| 4.3.3 | Various Results | 38 |
| 5 | Discussion | 41 |
| 5.1 | Contributions | 41 |
| 5.2 | Limitations | 41 |
| 5.3 | Future Work | 42 |
| 5.3.1 | Boundary Type and Definition | 42 |
| 5.3.2 | Junction Order on Edge | 42 |
| 5.3.3 | Vision Statistics | 43 |

6 Conclusions 44

References 47

List of Figures

| | | |
|----|---|----|
| 1 | Junctions at different scales showing those which could be reduced to a single pixel in red and bigger junctions in green | 5 |
| 2 | Junction Types - Ψ , T , X , Y , and L Junctions | 6 |
| 3 | Types of edges - occlusion , surface change , reflectance change , and illumination edges shown in an image originally by Edward Adelson [1]. | 11 |
| 4 | Image derivative computation using Sobel operators. | 13 |
| 5 | Algorithm Overview: Simplified flow of the algorithm described in this thesis. | 15 |
| 6 | Comparison of edge detection algorithms. | 17 |
| 7 | Comparison of line segment detection algorithms. | 17 |
| 8 | Three possible cases when calculating b_d according to the location of the line intersection in regards to the line segment. | 19 |
| 9 | Junction Clustering: Bigger-scale junctions created from pixel-wide internal junctions. Grouping is performed according to proximity. . | 20 |
| 10 | Junction Search Ranges: Computed search range surrounding each junction. Branches are looked for within these areas. | 21 |
| 11 | Branch Detection: Visual aid showing how branches are found. Bresenham's algorithm fits a line to pixel locations; pixels in green are adjacent pixels which are also analyzed. Paths with sufficient hits are then marked as branches. | 22 |
| 12 | Inverted-Y-junction: A sub-type of Y-junctions where a single branch displays acute angles in relationship to other branches. | 24 |
| 13 | Branch Types: Classified spines and arms per relevant junction. Different branch types provide cues to different edge types as seen in Table 1. | 24 |
| 14 | Directed Ψ -junctions and their effects in regards to image interpretation. | 26 |
| 15 | Junction Directions: Junctions with their respective direction vectors. | 26 |

| | | |
|----|--|----|
| 16 | Classified junctions on a synthetic image (by Edward Adelson [1]). Junctions on corners of objects with no curves can be observed to be correctly categorized and provide relevant information. Junctions on the cylinder and letters are less relevant and sometimes incorrectly classified. | 30 |
| 17 | Classified junctions on an architectural image. Bigger-scale corners on the building seem to be correctly classified. Some junctions here are affected by the environment's shadows. | 30 |
| 18 | Classified junctions on a natural image. Shadows and curves affect accurate junction detection. However, some prominent junction points are still correctly identified. | 30 |
| 19 | Edge classification output of an image of a building at a low angle with some occluding natural elements. The base of the building cannot be seen, but it is beneficial for the algorithm to include all upper peaks and boundaries of the structure. OB, SC, and RC edges are shown in blue, green, and red respectively. | 34 |
| 20 | Edge classification output of a natural image. Said image shows a cut-off subject (long cactus) which can be seen hindering junction detection (object corners and boundaries are pivotal for junction detection). OB, SC, and RC edges are shown in blue, green, and red respectively. | 36 |
| 21 | Edge classification output of an indoor scene. OB, SC, and RC edges are shown in blue, green, and red respectively. | 38 |
| 22 | Various edge classification results. Junctions are shown in diagnostic capacity. OB, SC, and RC edges are shown in blue, green, and red respectively. | 40 |

List of Tables

| | | |
|---|---|----|
| 1 | Junction Types and Uses - Summary | 6 |
| 2 | Edge Type Scores: Score value of each junction and branch type contained in an e . The highest score determines edge class. | 28 |
| 3 | Parameters: All experimental values used when testing the algorithm described in this thesis. | 28 |
| 4 | Confusion matrix for the first case study (see Figure 19). These numbers are obtained pixel-wise by comparing the output to the ground truth. UC stands for “unclassified” which is represented by the color white in all outputs (here, Figure 19.f). Numbers on the main diagonal (highlighted in green) are the amount of true positives for each class. | 33 |
| 5 | Metrics for the first case study calculated using Table 4. Scores closer to one indicate better performance from the model. | 33 |
| 6 | Confusion matrix for the second case study. It can be seen that there is a considerable amount of unclassified SC edge pixels. Further, hits from OBs are scattered, but it is shown that most are correctly classified. | 35 |
| 7 | Performance metrics quantifying the results in Figure 20. Recall of SC edges is extremely low as the algorithm fails to retrieve most of the significant results pertaining to this class. | 36 |
| 8 | Confusion matrix for the third case study. This matrix shows there is an abundance of occlusion edges in Figure 21. RC edges are also abundant, but they show a higher rate of misclassification, affecting RC recall in Table 9. SC edges, however, show the worst performance. The reason seems to be their lower presence, leading to unstable scores as even lower numbers easily affect the overall results. | 37 |

| | | |
|----|--|----|
| 9 | Performance metrics quantifying the results shown in Figure 21. High performance here might be due to that very fact, as most identified junctions are T-junctions and most edge pixels pertain to the class OB. Although accurately classified to some extent, edge classification marking the contours of human subjects seems to depend more on chance and luck. Subjects in the image impair accurate analysis using this algorithm. | 38 |
| 10 | Average metrics after performing tests using 100 images from the BSDS-RIND [2]. Selected images for testing included mostly architectural images, similar to those found in Figure 22. | 39 |

1. Introduction

Junction detection and classification has been addressed by psychologists and engineers alike. Junctions provide important cues for image analysis, but they cannot provide meaningful conclusions on their own. Engineers have mostly focused on supplementing junction-based analysis using local statistics, however, visual psychologists have proved that junctions could complement each other. This research will attempt to detect and utilize junctions as the sole cues for edge classification by taking a global approach. To achieve that, this thesis will first discuss its foundations in visual psychology and the importance and usefulness of junctions; then it will attempt to detect and obtain meaning from these visual keys; next, it will evaluate the algorithm's performance using edge classification; and finally, it will discuss the experimental results and the ultimate usefulness of global junction analysis.

1.1 The Role of Junctions in Visual Psychology

The mammalian visual system attempts to interpret scenes based on luminance. However, luminance itself is not the only factor taken into account when trying to reach a meaningful conclusion based on an observed scene. If that was so, we would have a hard time understanding when we see the same scene (e.g. our own bedrooms) when in different illuminations and conditions (e.g. nighttime and daytime, when wearing sunglasses, when it is rainy, etc.). This visual adaptation to different atmospheres is called lightness constancy. [1]

It is known that vision is based on three measurable components: reflectance (amount of light reflected by objects based on their colors), illuminance (amount of light falling on the surface of an object), and luminance (amount of light reflected by an object and captured by the human eye). Luminance is the only information received by our bodies, and, based on those values, brightness and lightness are determined. Brightness and lightness are perceptual, which means that they are the conclusions we make about a scene which cannot be quantified and lead to

an internal interpretation. Brightness is perceived luminance while lightness is perceived reflectance.

In general, brightness and lightness are known to be the result of an internal computation. However, this computation stems from trying to obtain illuminance and reflectance from a single luminance value. Luminance is the product of reflectance and illuminance. This means that the photons captured by our eyes are a mix of two variables of unknown value. At first, the task of ‘un-multiplying’ two numbers might seem impossible. In practical mathematics, this would be impossible without prior information. However, visual processing contains what could be either innate or learned cues which help it achieve this task.

Visual processing has been divided into three levels of sophistication: Low, mid, and high. Low-level visual processing happens in the retina and it deals with simple tasks such as contrast. Ewald Hering, a leading visual psychologist from the 1800s, first related lightness to this level of processing, claiming that contrast is the lead indicator which allows a spectator to interpret reflectance in a scene.

On the opposite end, high-level visual processing deals with previous knowledge of objects themselves and their surroundings. The high-level approach to lightness was proposed by Hermann von Helmholtz in 1867, and it helps explain how geometry (and, therefore, our judgement of how light works in a 3D scene) in the *corrugated plaid illusion* aids computations in visual processing. However, a new version of the corrugated plaid proposed by Adelson [1] exhibits flaws in this logic. By themselves, even objects with similar placements and shapes in a 3D scene might be perceived as receiving different kinds and amounts of light, even as they are placed as though they should be under the same illumination.

As both low and high-level visual processing seem unable to fully explain how lightness is computed, a mid-level approach ensues. Mid-level visual processing takes on tasks related to contours and groupings. Gestalt psychology argues that groups in scenes are judged as independent sets, as continuity and proximity plays a vital role in determining lightness. Adelson [1] further backs this approach by proposing a model which assumes vision employs an adaptive window to evaluate atmospheres at different scales.

As described by Adelson [1], adaptive windows can grow larger or smaller to encase different groups of various perceived colors. Afterwards, statistics are used to determine the lightness of all items within the same window. Gilchrist et. al [3] discuss both local and global frameworks which form the base of the workings of these windows.

Global frameworks are hypothesized to contain anchors that attach to the highest luminance in a scene [4, 5]. As such, lightness is also related to the set of luminance values perceived, interpreting the item with the highest luminance as white, and proceeding down from there. This was tested by Li and Gilchrist [5] who created a space where subject's vision could only perceive two colors: black and gray. Subject interpretations showed the colors would be interpreted as white and gray, setting the original gray color as white following the highest luminance anchoring.

On the other hand, local frameworks work closer together with **junctions**. Junctions play a role in determining if a certain luminance is being affected by an atmosphere (i.e. overall lighting conditions, filters, etc.) or by an object's geometry (shading). Different types of junctions provide visual cues which help differentiate sections of objects (surfaces that form part of the same item) from filters. It is important to note that everything is within an atmosphere (there is no such thing as 'no atmosphere') which affects its illuminance and luminance, but junctions can help differentiate between two different atmospheres within the same scene. [1]

Adaptive windows use local frameworks to adapt in size and shape to sections of observed scenes when needed. In consequence, visual processing is done in parts, each segment resulting in a unique lightness and brightness. However, it is important to note that reflectance and illuminance might result in the same luminance. Just as $9 \times 2 = 18$, but so does 3×6 , lightness can also be perceived as 'equal' while our eyes arrive at similar conclusions regarding luminance multipliers (leaving behind the true value of luminance itself).

Levels of visual processing, lightness constancy, adaptive windows, frameworks at different scales, all of these concepts give us a small insight of the visual system's workings, however, every concept holds a plethora of information which can be

explored more profoundly. Given this brief introduction to visual psychology, this thesis will focus on one of the most important indicators which form part of mid-level visual processing: **junctions**. Junctions are formed from shapes of objects and how light interacts with them and their surroundings, and in turn, junctions provide visual cues which segment atmospheres and help interpret a scene's composition. By using junctions, this research attempts to obtain meaningful global analysis results in the form of edge classification.

1.2 Junctions and Their Uses

1.2.1 What are Junctions?

Junctions, in image processing, are points where lines or edges intersect. Lines which form them are considered the junction's *branches*. In research, these junctions are typically reduced to a single pixel, but in reality exist at different scales (see Figure 1). Notice, however, that junctions at bigger scales always contain smaller junctions which set their boundaries. For the purpose of this thesis, junctions will be defined as $J = \{x, y, r, \vec{d}, M, \{\theta\}_{i=1}^M\}$ where (x, y) are the center coordinates, r is the radius (scale of the junction), \vec{d} is the direction, M is the number of branches, and θ is the orientation of each of those branches.

1.2.2 Types of Junctions

Adelson [1] talks about the five most common types of junctions and their meanings pertaining edges. As validation for this research, junctions are used to attempt to classify edges. For the purpose of this research, edges will be divided into three types: occlusion (OB from *occlusion boundary*) - caused by objects overlapping, reflectance change (RC) - caused by a change in color, and surface change (SC) - caused by sharp turns in the geometry of the same object.

A junction's type is determined according to both the number of its branches and their configuration (see Figure 2). The following are the types of junctions which will be addressed within this thesis:



(a) Indoor Scene Corner

(b) Zoomed-in Corner and Junctions

Figure 1: Junctions at different scales showing those which could be reduced to a single pixel in **red** and bigger junctions in **green**.

- L-junctions have only two branches. These junctions signal OBs being found most commonly on lower object corners.
- T-junctions have three branches, two of which compose the *roof* of this junction. The third (called *spine*) branch could be angled at any direction with respect to the others. The roof of a T-junction indicates an OB and its spine could signal either an RC or an OB. T-junctions by themselves can be used for depth ordering as they are the most common indicators of occlusion.
- Y-junctions have three branches, none of which lie on the same line. These junctions signal corners of objects, meaning their branches could be either OBs or SCs. Y-junctions can be used to find the limits of geometrical objects.
- X-junctions are composed by only two lines which produce four branches (each pair being parallel to one another). These types of junctions are produced by either RCs or the presence of a filter (e.g. something transparent like sunglasses or stained glass).

- Ψ (Psi) junctions have four branches, two of which are parallel to each other (i.e. lie on the same line called *spine*) and two which have acute angles with respect to the same neighboring branch (called *arms*). Arms of a Ψ junction indicate RCs and the spine most commonly indicates an SC. Opposing Ψ junctions on the same edge negate the spine's usual indications.

All summarized junction information can be found on Table 1.

| Junction Type | Branch Number | Parallel Pairs | Uses |
|---------------|---------------|----------------|--|
| L | 2 | 0 | Locate corners and OBs. |
| T | 3 | 1 | Roof \rightarrow locate OBs. Spine \rightarrow locate OBs or RCs. |
| Y | 3 | 0 | Locate corners and OBs or SCs. |
| X | 4 | 2 | Locate filters or RCs. |
| Ψ (Psi) | 4 | 1 | Arms \rightarrow locate RCs. Spine \rightarrow locate SCs. |

Table 1: Junction Types and Uses - Summary

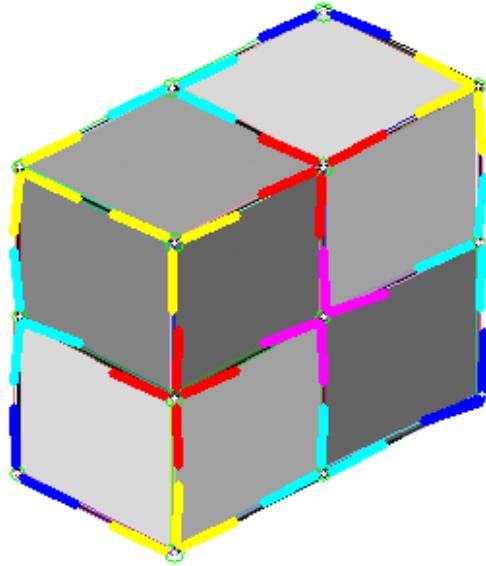


Figure 2: Junction Types - Ψ , T, X, Y, and L Junctions

1.2.3 Global Analysis of Junctions

As most research is centered on local analysis of junctions, it would be fitting to explain the usefulness pertaining to a global analysis of the same. By categorizing edges it would be possible to form a base for algorithms which address tasks such as:

- Image Depth Estimation - Depth ordering has previously been achieved using a local analysis of T-junctions [6], however, depth estimation could be further improved using information from multiple junctions to order even objects which are not occluded by others.
- Image Segmentation - By correctly categorizing occlusion edges, it would be possible to accurately segment a scene.
- Photogrammetry - Creating a three-dimensional model from multiple images would require less data if depth and surface queues could be found within individual images.

1.3 Research Problem and Purpose

The human visual system could be the key to providing simpler methods of global junction analysis. Most state-of-the-art research has focused on using machine learning to classify, segment, and detect items in pictures. However, weights adjusted by these models provide little to no explanation as to their efficacy. Further, there is little to no research with a deep focus on this approach, current methods dealing directly with more specific analysis pertaining to image processing tasks. This is, creating a generalized global junction analysis algorithm could potentially aid in not only one, but multiple tasks in this field as opposed to task-specific models.

The purpose of this research is to create an algorithm which detects and classifies junctions at a user-defined scale, and (as a key for global junction analysis) both assigns directions and groups junctions which lie on the same edge. The base of this

approach is formed by line-segment-based junction detection and edge classification (for global analysis validation).

1.4 Thesis Overview

1.4.1 Research Contributions

This research has accomplished the following contributions:

- Further developed line-segment-based junction detection.
- Proposed a method which allows for junctions to grow according to a user-defined scale.
- Developed a method to group junctions using line segments which are found on the same edge.
- Demonstrated junctions can complement each other in a global analysis.

1.4.2 Thesis Structure

This thesis is organized as follows: Chapter 2 talks about the thesis's foundation on visual psychology and multiple approaches to junction detection. Chapter 3 details the algorithm developed in this thesis. Chapter 4 analyzes the results of the algorithm and usefulness of junctions as a cue for edge classification. Finally, Chapter 5 discusses limitations and opportunities for future work.

2. Literature Review

2.1 Junctions at Different Levels of Visual Processing

As discussed in Chapter 1, visual psychology can provide insight into the usefulness of junctions. Adelson [1] explores the existing types of junctions and how humans use them to separate illuminance from reflectance. Since luminance is the product of both, it might seem impossible to, in a sense, ‘un-multiply’ them and perceive them as separate colors and lighting conditions. However, both junctions and statistics seem to make it possible to succeed at this task, which allow us to interpret scenes with a high degree of accuracy (the phenomenon of lightness constancy arises from this fact). Adelson [1] also performs multiple tests using visual illusions which demonstrate how junctions and local contrast can be manipulated to create scenes which cannot be accurately described by the human visual system because of its reliance on such cues.

Barton L. Anderson [7] explores how x-junctions can specifically be employed to detect cases of transparency depending on the polarity of boundaries immediately adjacent to said junction. This is, Anderson [7] analyzed how the contrast around a single x-junction signals filters, haziness or RCs depending on its polarity (and if it is reversed). [7] relies on both junctions and local statistics for scene analysis.

Josh McDermott [8] explores a high-level approach to edge classification. Anderson [7] seems to provide evidence of a bottom-up computation, but McDermott [8] concludes the opposite after having human subjects use junctions for edge classification. In his research, subjects are asked to classify edges after looking at windows containing junctions. The smaller the windows, the more the subjects struggled to accurately label each sample. This implies that a better understanding of a scene would be necessary to utilize junctions in the first place. As McDermott’s [8] and Anderson’s [7] conclusions conflict, Adelson’s [1] claim of junctions lying somewhere in between bottom-up and top-down visual processing

seems to be the most accurate.

In the end, it is clear that junctions by themselves cannot be used for scene analysis. Research so far has mostly focused on supplementing junctions with a local statistical analysis [1, 7, 9–11] (using mostly contrast). However, as Adelson’s [1] findings suggest, junctions could potentially supplement each other, turning global junction analysis into a Gestalt psychology issue which deals with grouping scene sections.

2.2 Edge Detection and Classification

2.2.1 Edge Detection

Edge detection is perhaps one of the most researched tasks in the field of computer vision. The Canny edge detection [12] has become a staple pre-processing step for virtually any task pertaining to image processing. Other edge detectors exist, however, such as James H. Elder’s [13] multi-scale edge detector, which sought to preserve more details in images and provide a more robust approach in regards to scale and parameter-tweaking. Depending on the task, it might be convenient to select one method or the other.

2.2.2 Edge Classification

Edges can be classified into multiple classes (see Figure 3), but most research has focused on separating occlusion edges from others. This bias is due to occlusion edges being the most useful in segmentation and image depth estimation tasks, however, accurately classifying other types of edges could help research in fields like photogrammetry. This thesis classifies edges into all types shown in Figure 3 except *illumination* edges, which are boundaries created by shadows. Shadows do not usually cause steps in contrast, making them difficult to find using the canny edge detector [12] or the MIT line segment detector [14]. As such, this paper is unfit to analyze such kinds of edges. Edge classification research also has its roots stemming from visual psychology, however, most state-of-the-art methods

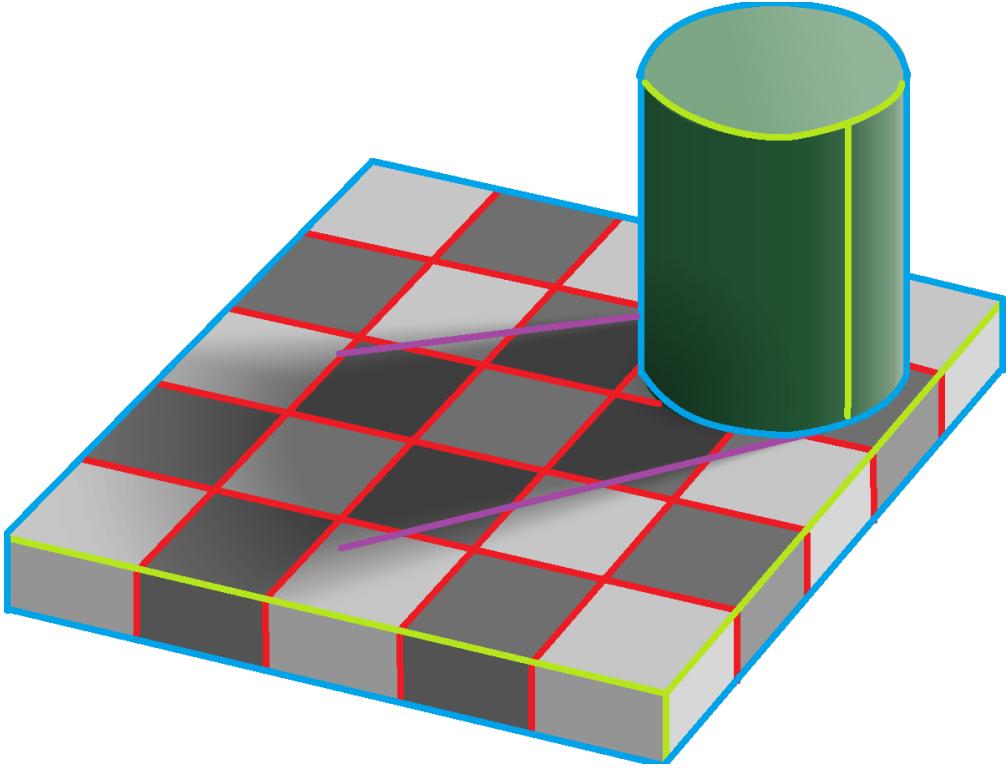


Figure 3: Types of edges - **occlusion**, **surface change**, **reflectance change**, and **illumination** edges shown in an image originally by Edward Adelson [1].

have turned to machine learning to tackle this task. [2, 10, 11, 15] all use CNNs at least as a part of their research. Interestingly, [10, 11, 15] employ models with a low number of hidden layers (going from one to three hidden layers) with only [2] relying on a more complex architecture. Even so, authors in [10, 11, 15] claim that their models match or even outperform human subjects. [11] found that humans and CNNs use different statistics for edge classification where the former relies more on scale-dependent statistics (such as texture) and the latter on scale-invariant cues (such as contrast). Notably, [11] also found that orientation was a cue employed by CNNs when classifying edges. It is possible this is due to a pattern in scene configuration and in occlusion-edge layouts; this, in turn, could potentially be related to junction directions.

Other edge classification methods use both chromatic and achromatic local statistics. [9, 10, 15] all employ contrast as their main indicator when classifying

occlusion edges. However, when classifying RCs and SCs, [9] found that contrast is not reliable. [10] uses blue-yellow and red-green differences to overcome said classification challenge.

So far, as edge classification research has relied heavily on local statistics extracted from small image patches, global junction analysis has not been employed to solve said task.

2.3 Line Segment Detection

This thesis relies heavily on line segment detection. As such, multiple line segment detection method were considered. [16] is a well-known algorithm which is available on OpenCV, however, [17] exposes the flaws of this approach as it only finds a single segment per globally-optimal line where, in truth, it is common to find co-linear line segments in images. [17] later attempts to fix this by using the standard Hough transform as a base which is then segmented using probabilities, but in the end it remains dependent on the global parameters of the algorithm.

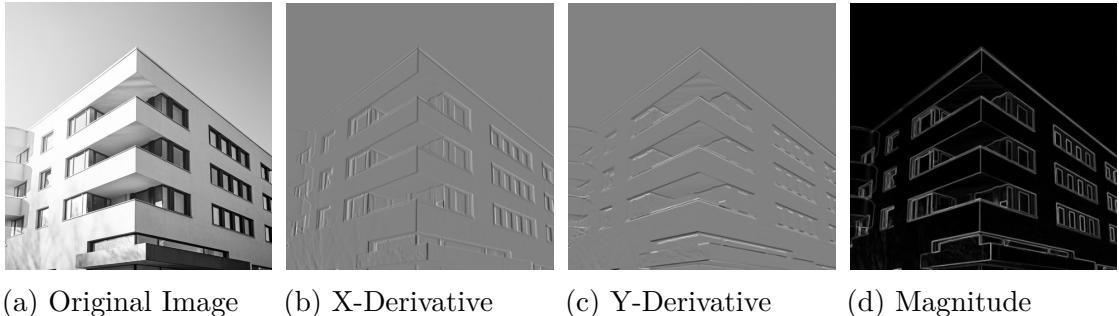
This research instead employs [14], which checks for gradient orientations inside rectangular regions. This method allows for detection of smaller-scale line segments which could be important to find junctions.

2.4 Approaches to Junction Detection

Junction detection research has mostly relied on using image derivatives (the most common method to create edge maps - see Figure 4) as a base [18–20], but those which further process edges obtained from those derivatives [21, 22] seem to have had more accurate results. Approaches which do not rely on image derivatives also exist, but these also rely on the creation of some sort of wireframe or edge map.

2.4.1 Skeleton Extraction

Skeleton extraction works by reducing structures to a single pixel in width. Research which employs skeleton extraction for junction detection does so exclusively on



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

Figure 4: Image derivative computation using Sobel operators.

binary images [23, 24]. This method is best applied on sketch-like images as it ensures finding a single junction per intersecting stroke. Further, this approach is hardly affected by noise as superimposed pixels are reduced together with the overall structure.

2.4.2 Probabilities

[21] relies on image derivatives, but further processes them at different scales using the *a contrario* detection theory. This theory relies on the Gestalt psychology view of grouping, stating that meaningful structures in images would be very unlikely to occur in a random sample. This approach reduces the number of false positive junction detection, but could also increase the number of false negatives.

2.4.3 Machine Learning

A recent approach [25] locates junctions by separating images into overlapping patches and optimizing assumed junctions in each patch. This approach is highly resilient to noise, but its parameters must be tweaked depending on the image being processed. [25] optimizes assumed junctions which possess three branches and, combining the results over multiple patches, determines the most probable junction location. This approach's greatest weakness is redundancy as it sometimes finds different junction locations from nearby patches which truly belong to a single junction.

[26] also employs a machine learning approach, but in the form of a CNN. This model was trained using human-labeled ground truth which marks both lines and junctions of man-made objects in both architectural and in-door scenes. Their results prove this approach to be best employed when extracting structures of overall scenes. However, smaller scale junctions and lines are ignored.

2.4.4 Line Segment Detection

[22] provides perhaps a more simple approach to junction detection. Relying on line segment detection, line intersections are computed and validated using an edge map. As junctions themselves are line intersections, this approach can directly be applied to both junction detection and classification. By relying on line segment detection, junction branches are more accurately detected and scaled. A weakness of this approach would be its use on natural scenes, as the lack of shapes which can be described by lines affects its results. This thesis also uses line segment detection as a base, which means it would be best applied to in-door and architectural scenes.

3. Methodology

The basic overview of the algorithm this thesis proposes can be seen in Figure 5. Simplified, this algorithm follows the following steps:

1. Detect: Detect line segment intersections which are set as junction candidates.
2. Scale: Cluster and find the scale of nearby junctions.
3. Classify: Classify junctions based on their branches.
4. Find direction: Find the direction of certain types of junctions for increased accuracy in global analysis.
5. Group: Group junctions lying on the same edges.

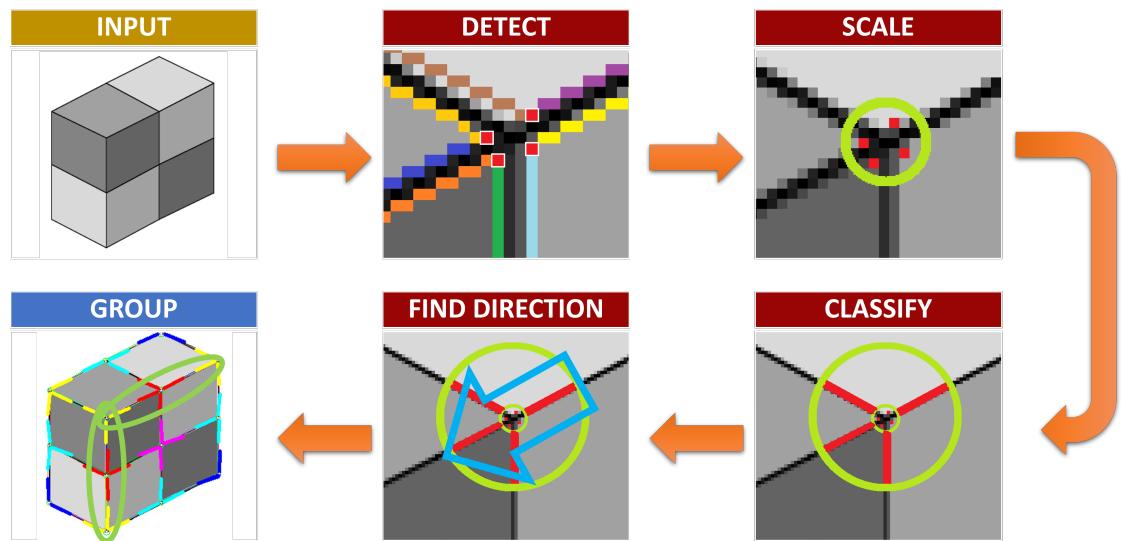


Figure 5: Algorithm Overview: Simplified flow of the algorithm described in this thesis.

An in-depth explanation of all steps and further sub-steps is provided throughout this chapter.

3.1 Preprocessing

As talked about in Chapter 2, both Canny edge detection [12] and the MIT line segment detector [14] are employed for image preprocessing. An edge map is employed to find junction branches and group junctions. On the other hand, line segments are utilized for junction detection itself.

Experimentally, it was found that the best results are obtained when the edge map and the detected line segments agree with each other. This is, the more edges which can be expressed with lines, the easier it is to detect and analyze junctions on said image. Further, thinner edges also allow for better results, as detected line segments are always a single pixel in width. Since Canny [12] provides such thin edges, it was chosen over the multi-scale edge detector [27] (see Figure 6).

On the other hand, detecting small line segments while ignoring shadow edges proved vital when handling junctions at smaller scales which is why the MIT line segment detector [14] was chosen over [16] and [17] which both depend on finding globally optimal lines, the former failing to find smaller lines and the latter providing many shadow-edge lines (see Figure 7). Note, however, that this thesis employed a threshold T_l to avoid using exceedingly small lines. This is due to the fact of the multiple iterations done over line segments which increases computational complexity.

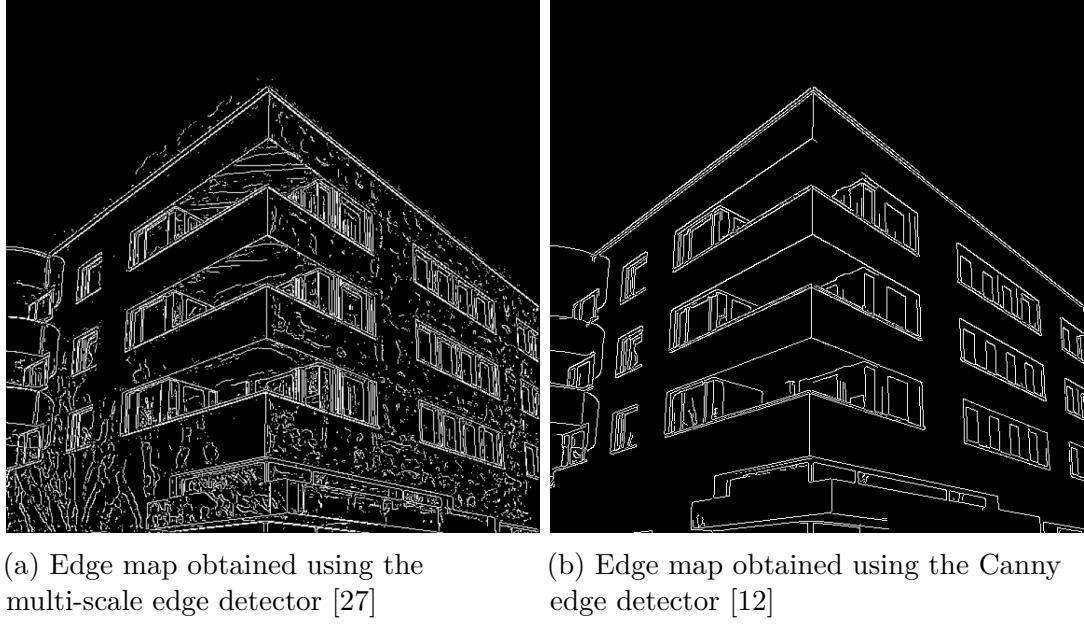


Figure 6: Comparison of edge detection algorithms.

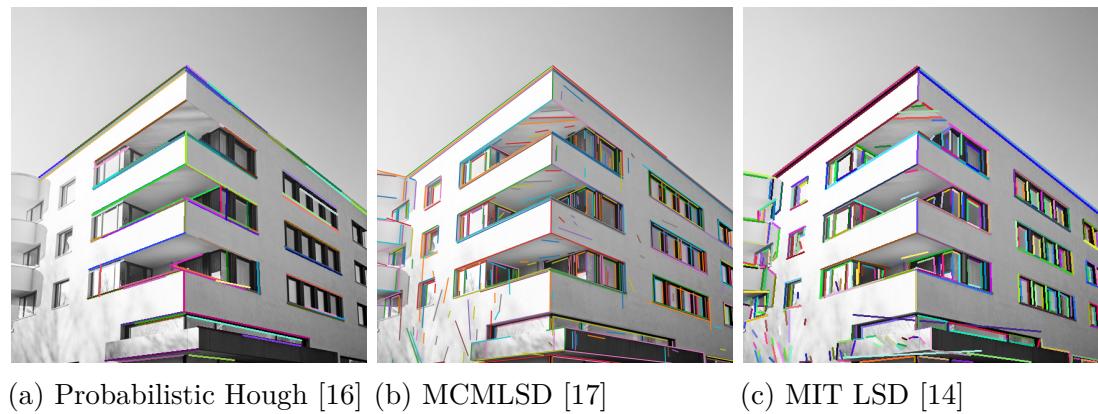


Figure 7: Comparison of line segment detection algorithms.

3.2 Candidate Junction Detection

Junction candidates (j) are identified from the intersections of line segments. However, most intersections must be filtered out. Intersections of parallel lines and intersections of lines which are beyond the threshold distance (T_d) from the actual segments are discarded.

3.2.1 Line Intersections

The angle between each pair of lines is calculated. If the angle between them is smaller than the angle threshold T_a , this intersection is ignored.

3.2.2 Belonging Distance

Belonging distance (b_d), originally described in [22], is a method to determine the usefulness of a line intersection for the task of junction detection. Belonging distance (see Equation 1) is computed for each j with coordinates (x, y) 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 $b_d < T_d$.

Figure 8 shows all possible cases when calculating belonging distance. Note, however, that case 8.b is not relevant in this stage of the methodology as each intersection between two lines will necessarily be either directly on the segment or away from it (not perpendicular to). Case 8.b can present itself when merging adjacent junctions, which will be important to keep in mind.

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

If the intersection is accepted as a junction candidate, as detected line segments might vary in accuracy, all pixels adjacent to said intersection are set as junction candidates as well.

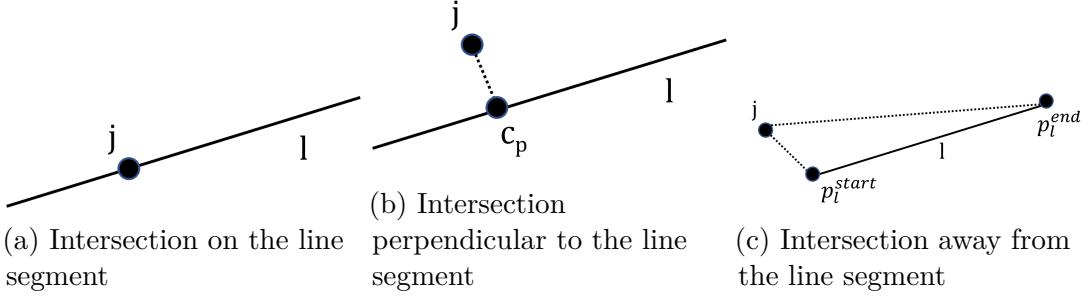


Figure 8: Three possible cases when calculating b_d according to the location of the line intersection in regards to the line segment.

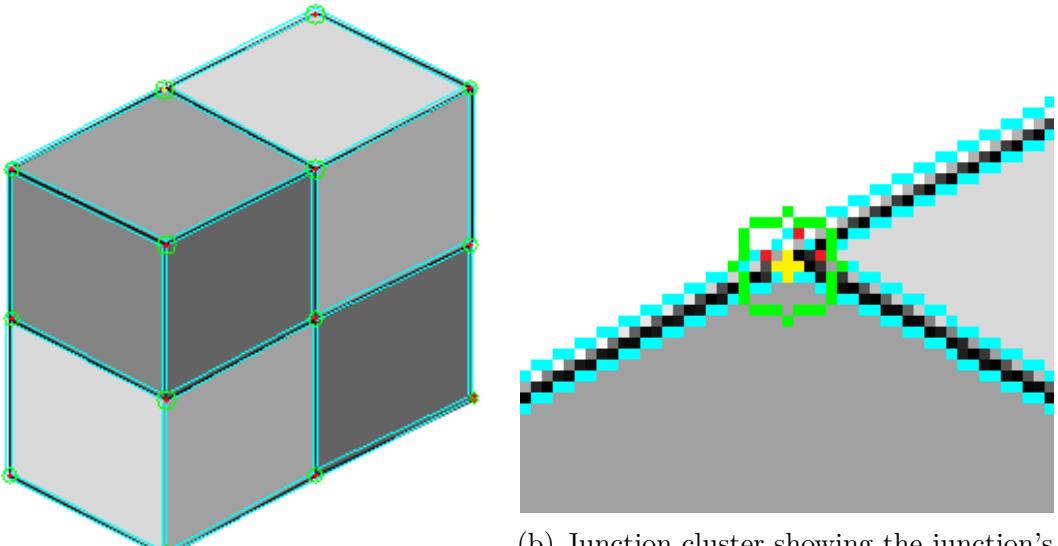
3.3 Junction Clustering

As mentioned in Chapter 1, junctions exist at different scales. Bigger junctions include more pixel-wide junctions within them, defining a wider circumference. This algorithm addresses this issue by setting a user-defined proximity threshold T_p and then clusters all nearby junctions. Default T_p is equal to T_d as this also aids in removing redundant (duplicate) js which arise from doubled line segments and adjacent js .

To cluster, *Euclidean distance* (see Equation 2) is calculated between each pair of js set as $d(p, q)$. All junctions within distance T_p from at least one junction in the cluster are grouped. The cluster's centroid is then computed and its distance d to the furthest j from it in the cluster is set as the junction's radius r .

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

As each junctions' r grows, these junction candidates go from being singular pixels to being circumferences (j_c). Center junction coordinates are set as the centroid of each cluster, updating (x, y) (see Figure 9). Further, the centroid's belonging distance is calculated again taking into account all line segments associated with the cluster, lines beyond T_d are unlinked from the cluster.



(a) Junction clusters on an example image

(b) Junction cluster showing the junction's
centroid, circumference, and pixel-wide
junctions

Figure 9: Junction Clustering: Bigger-scale junctions created from pixel-wide internal junctions. Grouping is performed according to proximity.

3.4 Junction Scaling

Although it might seem like a similar concept, junctions can be viewed as two different sizes. Junction clustering finds the size of the inner circumference's r , whereas junction scaling looks for the reach of the junction's branches.

3.4.1 Computing Search Range

Similar to [22], junction's branches are found and scaled according to their related line segments. For this, this algorithm first computes a search range which defines an *outer junction circumference*. Each search range r_l is found between each j and its N related lines¹ $\{l_i\}_{i=1}^N$ (considering len_i as the length of the i th line segment) as seen on Equation 3. The radius of the outer circumference r_o is set for each

¹Note that each junction might have more than two related line segments, as junction clusters could be formed by any number of them.

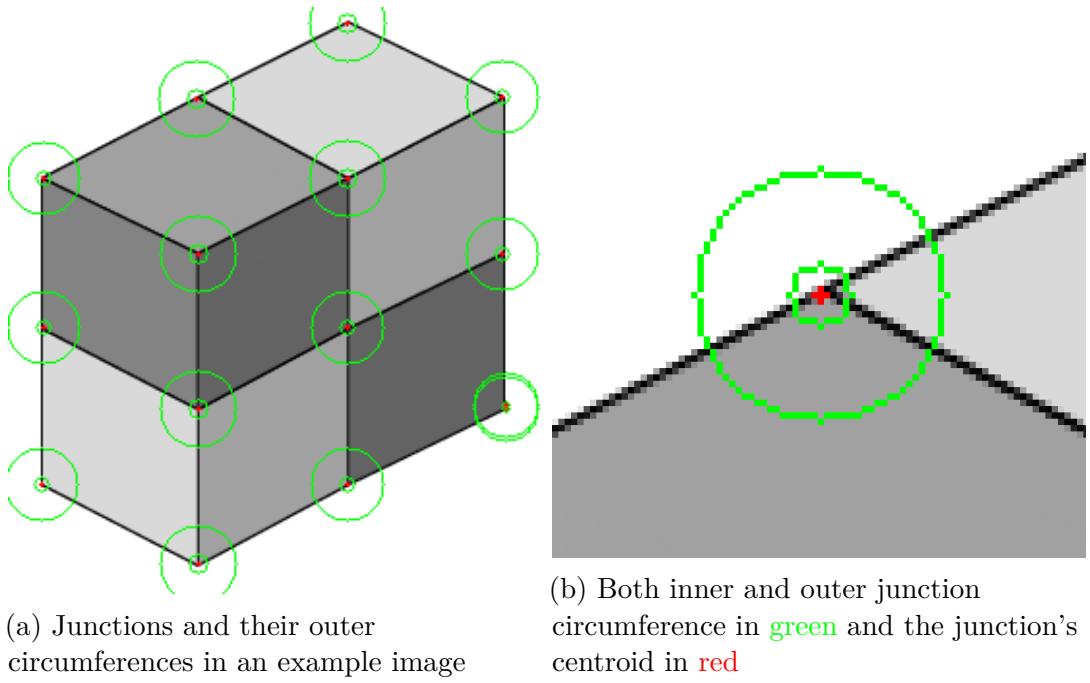


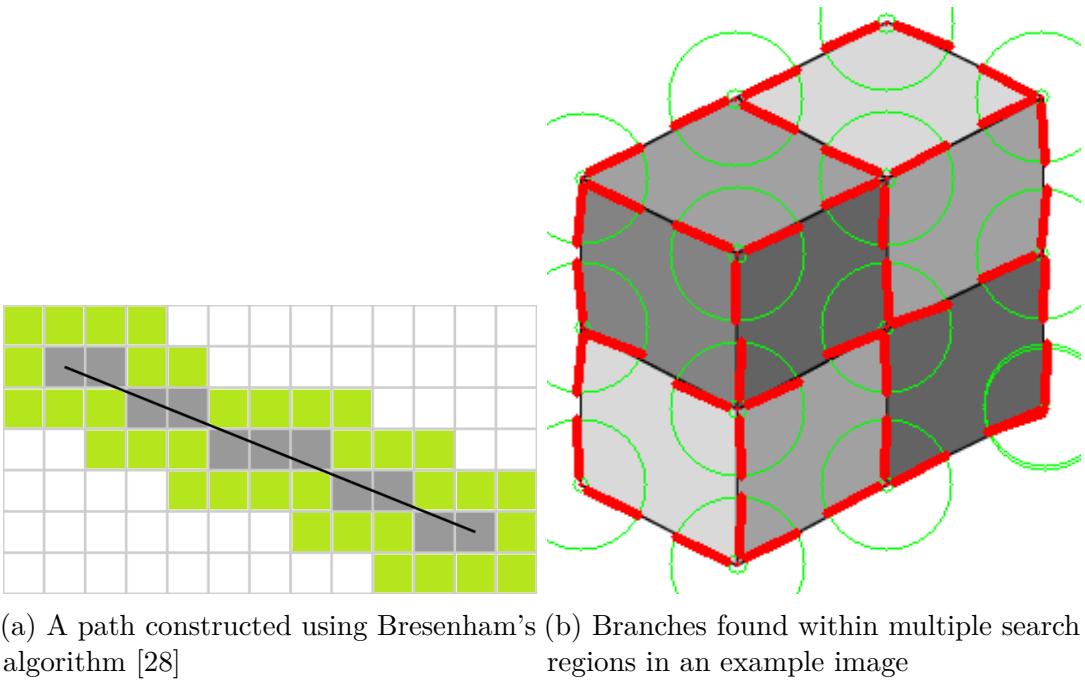
Figure 10: Junction Search Ranges: Computed search range surrounding each junction. Branches are looked for within these areas.

junction as $r_o = r + \min(r_{l_i}, \dots, r_{l_N}, T_r)$ using the threshold T_r to limit computational complexity (see Figure 10). This radius is considered to start from the centroid, but the actual search is performed starting from the inner circumference. For this reason, any j which has $r < r_o$ is discarded.

$$r_{l_i}(j, l_i) = \begin{cases} \max(T_d, \min(d(j, p_l^{start}), d(j, p_l^{end}))), & \text{if } j \text{ is not on } l, \\ len_i, & \text{otherwise} \end{cases} \quad (3)$$

3.4.2 Finding and Merging Branches

Once more, in a similar manner to [22], a path is set from the inner to the outer junction circumferences' and branches are detected. To reduce computational complexity, paths are only marked between pixels on both the inner and outer circumferences' which are also edge pixels. So far, simply working with line segments



(a) A path constructed using Bresenham's algorithm [28] (b) Branches found within multiple search regions in an example image

Figure 11: Branch Detection: Visual aid showing how branches are found. Bresenham's algorithm fits a line to pixel locations; pixels in green are adjacent pixels which are also analyzed. Paths with sufficient hits are then marked as branches.

might have seemed like enough, but as mentioned in Chapter 2, junctions only truly exist on image edges.

Each search path is set starting from each **edge** point on j_c to the closest edge point on the outer circumference using Bresenham's algorithm [28] (see Figure 11). Then, for each point in the path, said point is marked as a *hit* if any pixels in its pixel neighborhood are edges. After going through all points, if the branch is longer than $(r - r_o) \times 0.8$ and said branch overlaps only a single line segment, then it is accepted as a branch of j . When all branches are found, if two or more branches are separated by an angle T_a or less, only the line with the most *hits* is kept and its direction θ is stored as a component of j . Note that hits which pertain to the exact pixel in the path are worth double than those coming from adjacent pixels.

After this process, any j_s with only a single branch, two branches which lie on

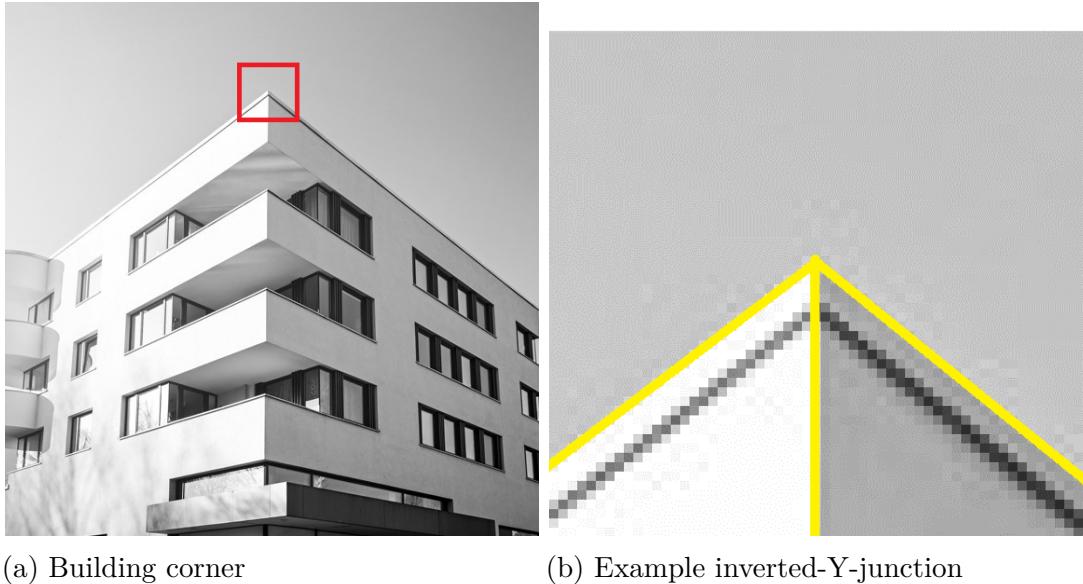
the same line, or more than four branches are discarded. Although junctions with more than four branches are still technically junctions, as no meaningful conclusion can come from their analysis, these are considered irrelevant for the purposes of this algorithm.

3.5 Junction and Branch Classification

All remaining junctions so far can be categorized according to the types mentioned in Table 1. Further, it is important to also record branch types as global analysis should consider which part of the junction lies on which edge of the image. Junctions are classified by checking their branch numbers and the parallel pairs of branches they contain (refer to Table 1). Junction branch classification is performed for all identified T, Ψ , and inverted-Y-junctions. Inverted-Y-junctions are a sub-type of Y-junctions (see Figure 12), which have an identifiable *spine*, this is, a single branch which has acute angles with respect to both other branches.

Branch classification is performed as follows (see Figure 13):

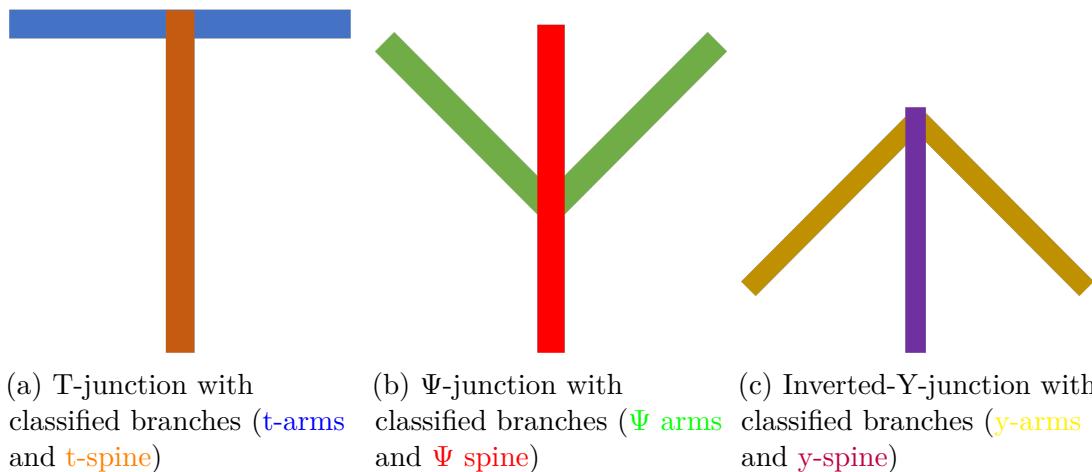
- T-junctions: the two parallel branches are classified as “t-arm” while the remaining branch is classified as “t-spine.”
- Ψ -junctions: the two parallel branches are classified as “ Ψ -spine” and the remaining branches as “ Ψ -arm.”
- Inverted-Y-junctions: the branch with acute angles with respect to all others is classified as “y-spine” and the rest as “y-arm.” Note that inverted-Y-junction spines are best at locating SCs and their arms are best at locating OBs. Regular Y-junctions usually just signal SCs.



(a) Building corner

(b) Example inverted-Y-junction

Figure 12: Inverted-Y-junction: A sub-type of Y-junctions where a single branch displays acute angles in relationship to other branches.



(a) T-junction with
classified branches (**t-arms**
and **t-spine**)

(b) Ψ -junction with
classified branches (**Ψ arms**
and **Ψ spine**)

(c) Inverted-Y-junction with
classified branches (**y-arms**
and **y-spine**)

Figure 13: Branch Types: Classified spines and arms per relevant junction.
Different branch types provide cues to different edge types as seen in Table 1.

3.6 Finding Junction Direction

In his analysis of junctions, Adelson [1] notes that junctions of the same type with incongruent directions will “cancel each other out.” For example, two Ψ -junctions with their spines aligned but mirrored arms cannot exist naturally on the same edge (see Figure 14). For this reason, the analysis of this specific situation would lead a viewer to interpret all edges around said junctions as RCs instead of the usual SC along the spine. Finding the direction of certain types of junctions is therefore pertinent when talking about global analysis.

No formal conventions exist to define junction direction. In this thesis, junction direction \vec{d} is set depending on junction type, always starting from the centroid towards the outer circumference (see Figure 15), as follows:

- L-junction: set as a vector bisecting the smaller of the two angles between its branches. This vector would normally point towards the *inside* of an object, rather than away (assuming the L-junctions are correctly found to be object corners).
- T-junction: set as a vector in the same direction as the spine.
- Ψ -junction: set as a vector in the same direction as the spine branch with acute angles in relationship to the arms.
- Inverted-Y-junction: set as a vector in the same direction as the spine.

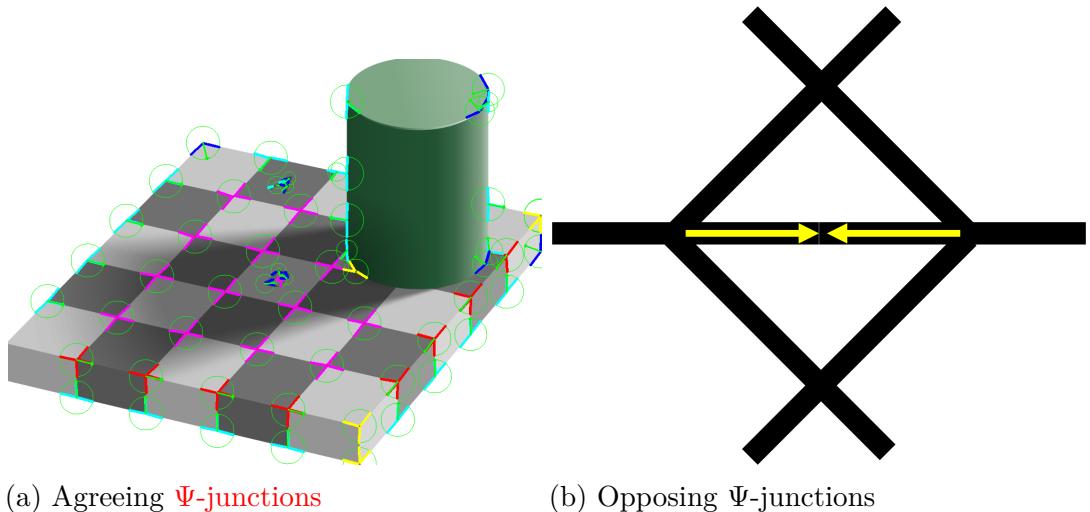


Figure 14: Directed Ψ -junctions and their effects in regards to image interpretation.

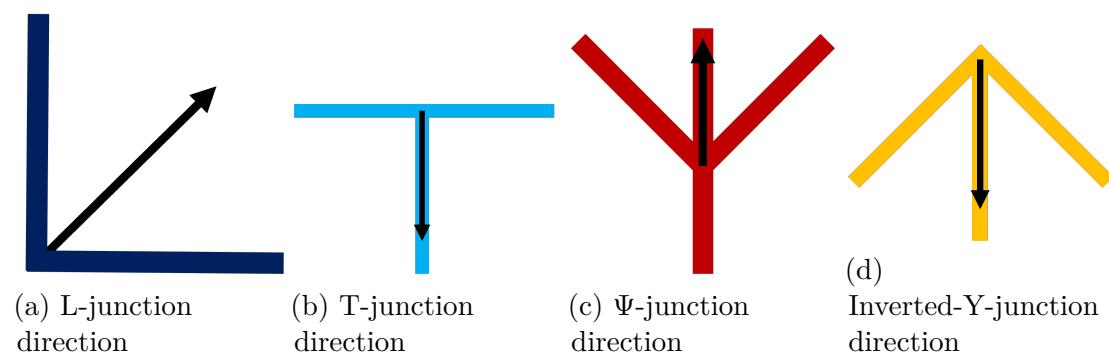


Figure 15: Junction Directions: Junctions with their respective direction vectors.

3.7 Straight Edge Detection and Junction Grouping

Junctions detected so far have associated line segments, but no edges beyond the hits contained in their branches. To group junctions, line segments themselves are grouped first. Any line segments whose edge points have a belonging distance (see Equation 1) below T_d from another parallel segment are grouped. These groups of line segments form longer lines along the image and are together considered a *straight edge*. For the purpose of this thesis, a straight edge e is defined as $e = O, P, Q, \{l\}_{i=1}^O, \{j\}_{j=1}^P, \{b\}_{k=1}^Q$ where O , P , and Q are the number of line segments, junctions and branch types respectively contained in e ; and where l , j , and b are the line segments, junctions, and branch types respectively.

Branch types are assigned to e s by iterating through each associated junction and matching junction branches with the line segments contained by the edge. It is important to separate branch types on each edge, as merely analysing junction types is not enough to make any conclusions about any e s.

3.8 Global Junction Analysis Testing: Edge Classification

As it is necessary to evaluate if this algorithm could be employed for global junction analysis, this thesis proceeds to perform edge classification by assigning an SC, OB, and RC score to each e . The highest score determines the edge type. The scores are calculated as seen in Table 2, for each relevant junction type or branch type associated to the e . These scores were determined empirically through brief prior testing. If no junctions are associated with e , the edge will be considered an RC. Further, the OB and SC scores have a default value of 0.02 and 0.01 respectively, as prior probability from [9] suggests that any random edge in an image is more likely to be an occlusion edge than other types, followed by SCs. This small weight is important in case of draws in the scores.

| 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 |

Table 2: Edge Type Scores: Score value of each junction and branch type contained in an e . The highest score determines edge class.

3.9 Parameters

To facilitate the replication of this algorithm, all parameters used when testing can be found in Table 3.

| Parameter | Value | Unit |
|-----------------------|-------|---------|
| Canny lower threshold | 85 | Votes |
| Canny upper threshold | 255 | Votes |
| T_d | 10 | Pixels |
| T_a | 15 | Degrees |
| T_p | 10 | Pixels |
| T_r | 30 | Pixels |

Table 3: Parameters: All experimental values used when testing the algorithm described in this thesis.

4. Results

4.1 Empirical Junction Detection Results

Each step in this algorithm has an influence on its results. As junctions are detected using line segments, the results are highly dependant on line segment detection itself. Shapes which cannot be expressed by lines and edge-maps with a lot of small curves make the task of both junction detection and analysis harder. For this reason, it is generally recommended that this algorithm be used on images which contain subjects of analysis which can be expressed in lines. Indoor and synthetic scenes provide the best kinds of results, followed by architectural scenes. Natural scenes are more challenging to evaluate and, although they would be possible to analyze, the author of this paper recommends that an alternative junction detection method be employed for said types of images. However, if junctions are correctly identified, classification and interaction could still proceed as this thesis has proposed. Figures 16, 17, and 18 show empirical results on synthetic, architectural, and natural images. These images possess no ground truth, but can be used to visualize which types of images are easier to work with.

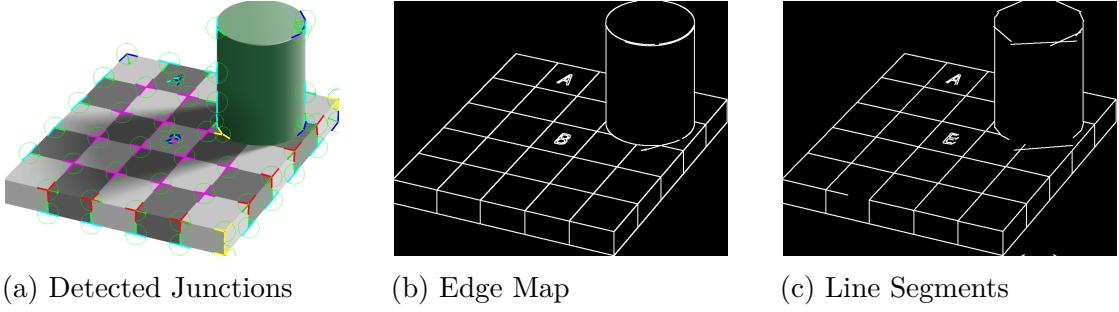


Figure 16: Classified junctions on a synthetic image (by Edward Adelson [1]). Junctions on corners of objects with no curves can be observed to be correctly categorized and provide relevant information. Junctions on the cylinder and letters are less relevant and sometimes incorrectly classified.

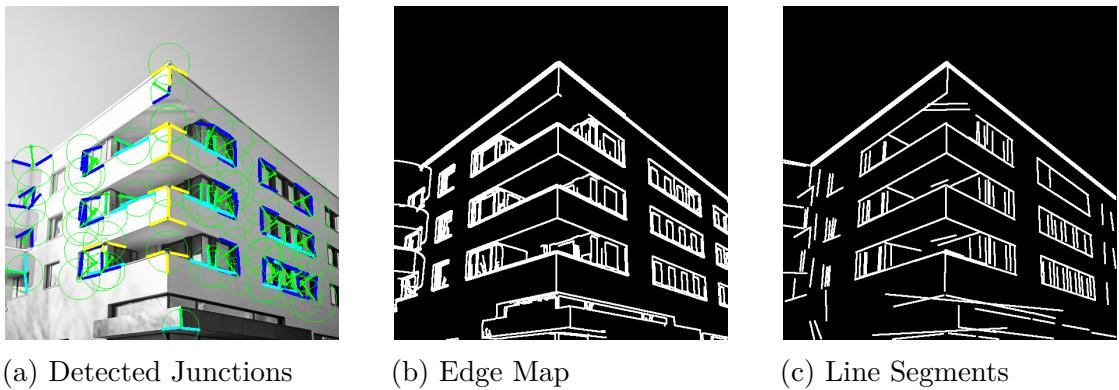


Figure 17: Classified junctions on an architectural image. Bigger-scale corners on the building seem to be correctly classified. Some junctions here are affected by the environment’s shadows.

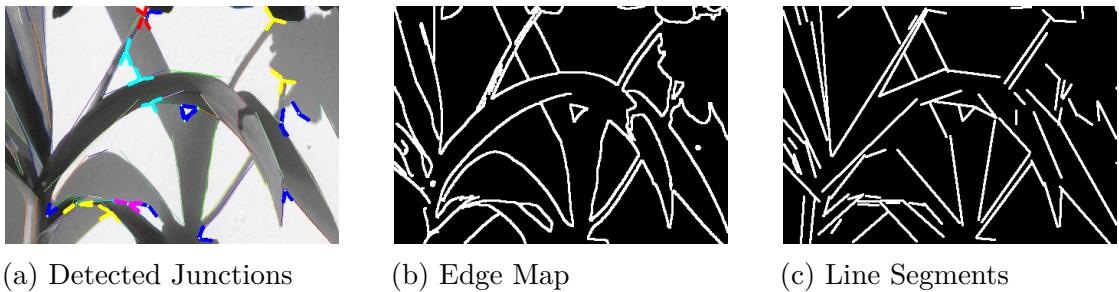


Figure 18: Classified junctions on a natural image. Shadows and curves affect accurate junction detection. However, some prominent junction points are still correctly identified.

4.2 Edge Classification Results

4.3 Performance Metrics

Although this thesis has a strong link to junction detection, merely evaluating the results of the detection itself would sidetrack from the main focus of this research: global junction analysis. To more accurately judge how useful this algorithm would be when grouping and analysing junctions as a group, this algorithm was applied to the task of edge classification. The results of edge classification itself provide a stronger insight on the potential applications of this algorithm, as well as assess junction interaction and scoring explored in this thesis.

In object classification, precision (see Equation 4) and recall (see Equation 5) are useful metrics which can provide pixel-wise results assessing a model's capability to correctly find and categorize instances of objects. This thesis too employs said metrics, however, pixel-wise evaluation is overly harsh for the purpose of edge classification. As it is not the main focus to accurately segment edges but merely assess if they can be classified correctly, the ground truth utilized for assessment was dilated when checking for true positives and it was returned to its normal state when checking for false negatives. These metrics and their harmonic mean (see Equation 6) can then more accurately assess the results of junction analysis using this algorithm.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True positive} + \text{False positive}} \quad (4)$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True positive} + \text{False Negative}} \quad (5)$$

$$\text{F1 score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (6)$$

Note that during edge classification testing, ground truth was fed into the system as an edge map. The reason for this is to focus result analysis on junction interaction. Performance here analyses results of this algorithm assuming an edge map which contains no noise or shadows.

4.3.1 Data Set

Mostly architectural images and images with shapes that could be expressed with lines were selected from the BSDS-RIND [2] data set. The original images come from the BSDS500 [29] data set which contains five hundred hand-labeled segmented images. Said segmentation was performed at various levels of detail, providing various edge-maps. The BSDS-RIND [2] further segments these annotations by categorizing edges as four different types: depth (OB), surface normal (SC), reflectance (RC) and illumination. This last category was discarded as these are edges produced by shadows, which was not explored here during research on junction interaction.

4.3.2 Edge Classification Case Studies

4.3.2.1 Case Study 1: Architectural Image

Figure 19 shows a case study using an architectural image. From the figure, it can be observed that RC edges are mostly accurately located, which is due to the building pattern which includes multiple X-junctions. Further SC edges are mostly classified correctly following Y and Ψ -junctions found in the picture. On the other hand, OBs around the building seem to lack in accuracy. Figure 19.d shows a Y-junction identified on the upper left corner of the building. This junction was classified incorrectly due to the presence of clouds, also leading its adjacent edges to misclassification.

Table 4 contains the confusion matrix for this case study. This matrix represents a pixel-wise true positive, false negative, and false positive results against a dilated ground truth. These values are used to obtain the performance metrics seen in Table 5. In this case, numerical results support the observations from Figure 19. Performance applying the algorithm on this picture shows results of over 0.8 on all instances, however, scores for OB are the lowest given the mislabeled edges on the left side of the building, which is one of the instances which drop its recall.

Overall performance of the algorithm in this image is better than expected. Given that flowers (such as the ones present in Figure 19.a) cannot be fully expressed

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

Table 4: Confusion matrix for the first case study (see Figure 19). These numbers are obtained pixel-wise by comparing the output to the ground truth. UC stands for “unclassified” which is represented by the color white in all outputs (here, Figure 19.f). Numbers on the main diagonal (highlighted in green) are the amount of true positives for each class.

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

Table 5: Metrics for the first case study calculated using Table 4. Scores closer to one indicate better performance from the model.

in terms of lines, getting results of over 0.8 is unexpected. However, as mentioned before, ground truth was dilated to favor whole-edge-classification over mere pixel-wise classification. This means that the results could contain bias beyond the one perceived. Another explanation for the results obtained on the flowers pertains to the edge-map being fed into the system. Edges are more relevant this way, and could help the algorithm locate more informative junctions. As such, edge-maps prove to be pivotal to the foundations of this algorithm.

Results on this image might indicate further applications in natural images. However, the task of obtaining relevant edge-maps could prove a challenge by itself.

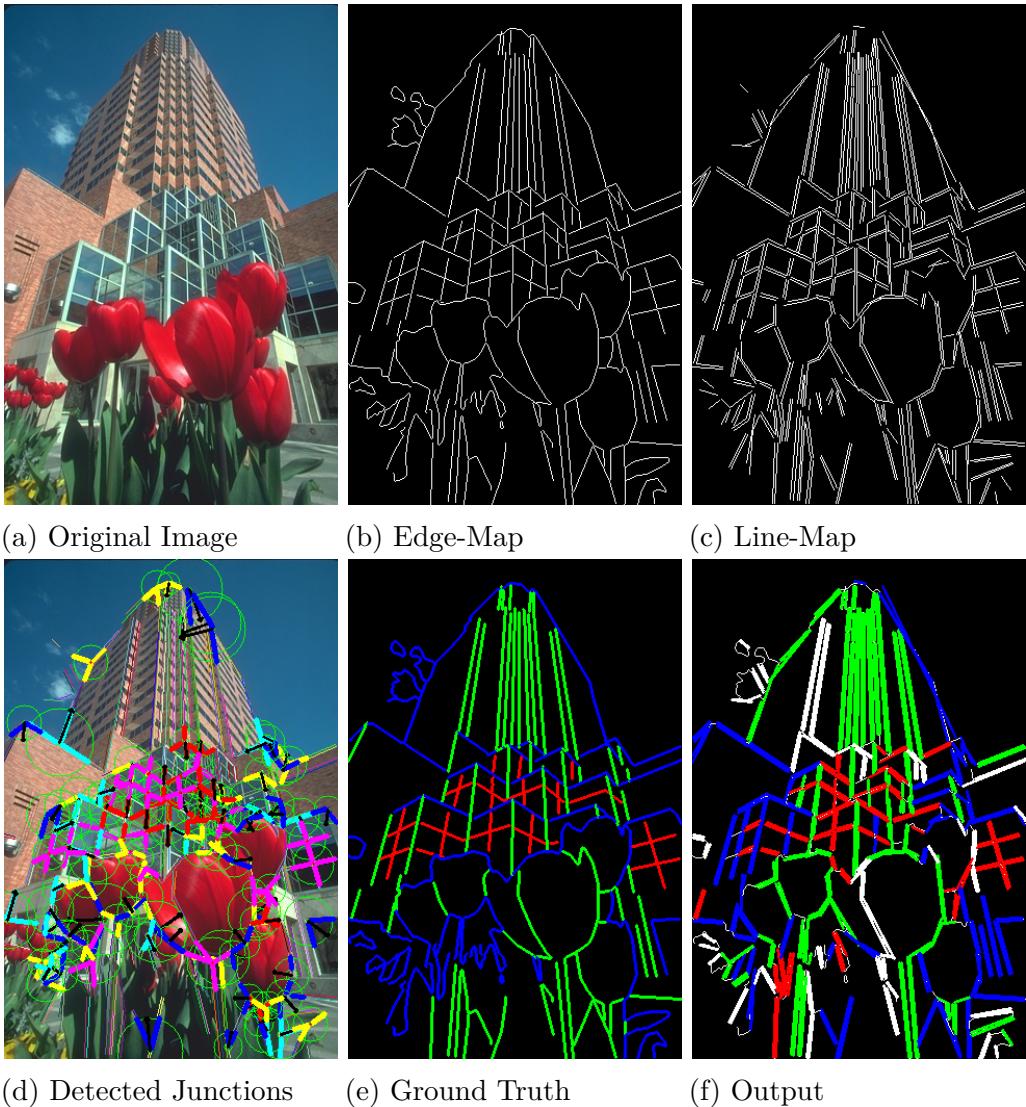


Figure 19: Edge classification output of an image of a building at a low angle with some occluding natural elements. The base of the building cannot be seen, but it is beneficial for the algorithm to include all upper peaks and boundaries of the structure. OB, SC, and RC edges are shown in blue, green, and red respectively.

4.3.2.2 Case Study 2: Natural Image

Natural images contain shapes and objects with curves. As this algorithm depends on lines for junction detection, natural images are expected to have both lower

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

Table 6: Confusion matrix for the second case study. It can be seen that there is a considerable amount of unclassified SC edge pixels. Further, hits from OBs are scattered, but it is shown that most are correctly classified.

precision and lower recall. Figure 20 shows an example of edge classification on a natural image. One might argue that some objects in said figure could be expressed with lines, however, and additional hindrance can be perceived. As most junctions appear in object corners and boundaries, the fact that one of the subjects (actus on the left) is cut off (neither the base nor its peak are visible in the image) greatly impairs this algorithm’s capabilities. As such, most edges in the image are categorized as false negatives.

Notice that Table 7 (constricted using Table 6) shows a relatively high performance despite what could be perceived in Figure 20. The reason for this is that, out of the relevant instances found, most of those instances are correctly classified (leading to a higher precision). Precision is a metric which assesses the quality of the instances which were classified at all, thus, OB and RC edges maintain a good performance. On the other hand, recall is greatly affected by false negatives, however, as this algorithm is most concerned “edge-wise” classification rather than pixel-wise classification, it would be best to interpret the result as whole sections. For this reason, recall from missing OBs is not greatly affected. Although lacking in performance regarding SC edges for the lack of corners, results show a positive outlook for the use of junctions as markers in natural images given a suitable junction detection method.

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

Table 7: Performance metrics quantifying the results in Figure 20. Recall of SC edges is extremely low as the algorithm fails to retrieve most of the significant results pertaining to this class.

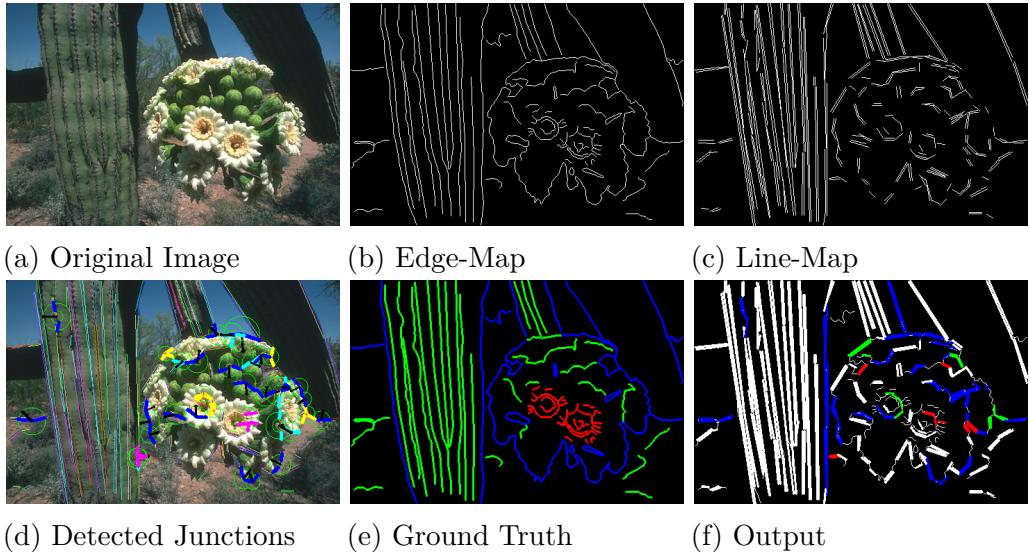


Figure 20: Edge classification output of a natural image. Said image shows a cut-off subject (long cactus) which can be seen hindering junction detection (object corners and boundaries are pivotal for junction detection). OB, SC, and RC edges are shown in blue, green, and red respectively.

4.3.2.3 Case Study 3: Indoor Image

Figure 21 shows the output applied to an indoor image. Although showing low performance when observing the natural elements in the picture (i.e. the people and the content of the paintings), edges pertaining borders and furniture seem to be correctly classified. When looking at the top-right of the output, the small painting's border shows once more how this algorithm can fail if an object's corners are not fully visible.

This case study and some examples found in Figure 22 show an abundance of T-junctions. It is worth to note that although these junctions are most commonly linked with occlusion edges, it seems that they can also be very present in both other types of edges. As such, T-junctions are not very reliable when they are the sole or most prominent type of junction in a scene being analyzed.

Table 9, created using Table 8, shows a lower overall performance than the first case study (see Table 5). This is somewhat unexpected, as indoor results were hypothesized to have the best performance. However, considering how crowded Figure 21 is with people and shapes in the paintings which cannot be expressed in lines, it is less surprising performance would fall.

Overall, it is once more recommended that this algorithm be used only on images with long-straight edges, unless junctions are provided as ground truth and only scoring and junction analysis is taken from this research.

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

Table 8: Confusion matrix for the third case study. This matrix shows there is an abundance of occlusion edges in Figure 21. RC edges are also abundant, but they show a higher rate of misclassification, affecting RC recall in Table 9. SC edges, however, show the worst performance. The reason seems to be their lower presence, leading to unstable scores as even lower numbers easily affect the overall results.

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

Table 9: Performance metrics quantifying the results shown in Figure 21. High performance here might be due to that very fact, as most identified junctions are T-junctions and most edge pixels pertain to the class OB. Although accurately classified to some extent, edge classification marking the contours of human subjects seems to depend more on chance and luck. Subjects in the image impair accurate analysis using this algorithm.

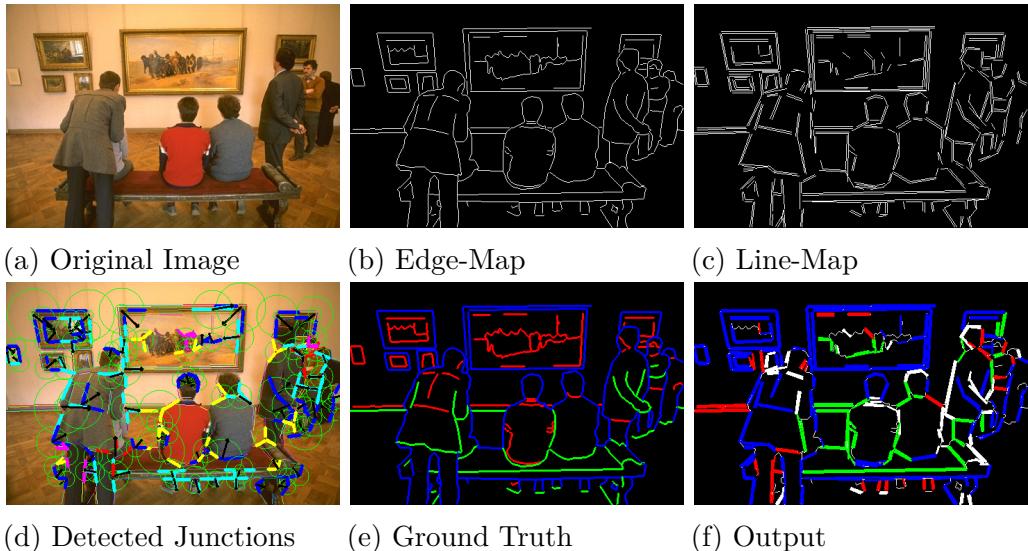


Figure 21: Edge classification output of an indoor scene. OB, SC, and RC edges are shown in **blue**, **green**, and **red** respectively.

4.3.3 Various Results

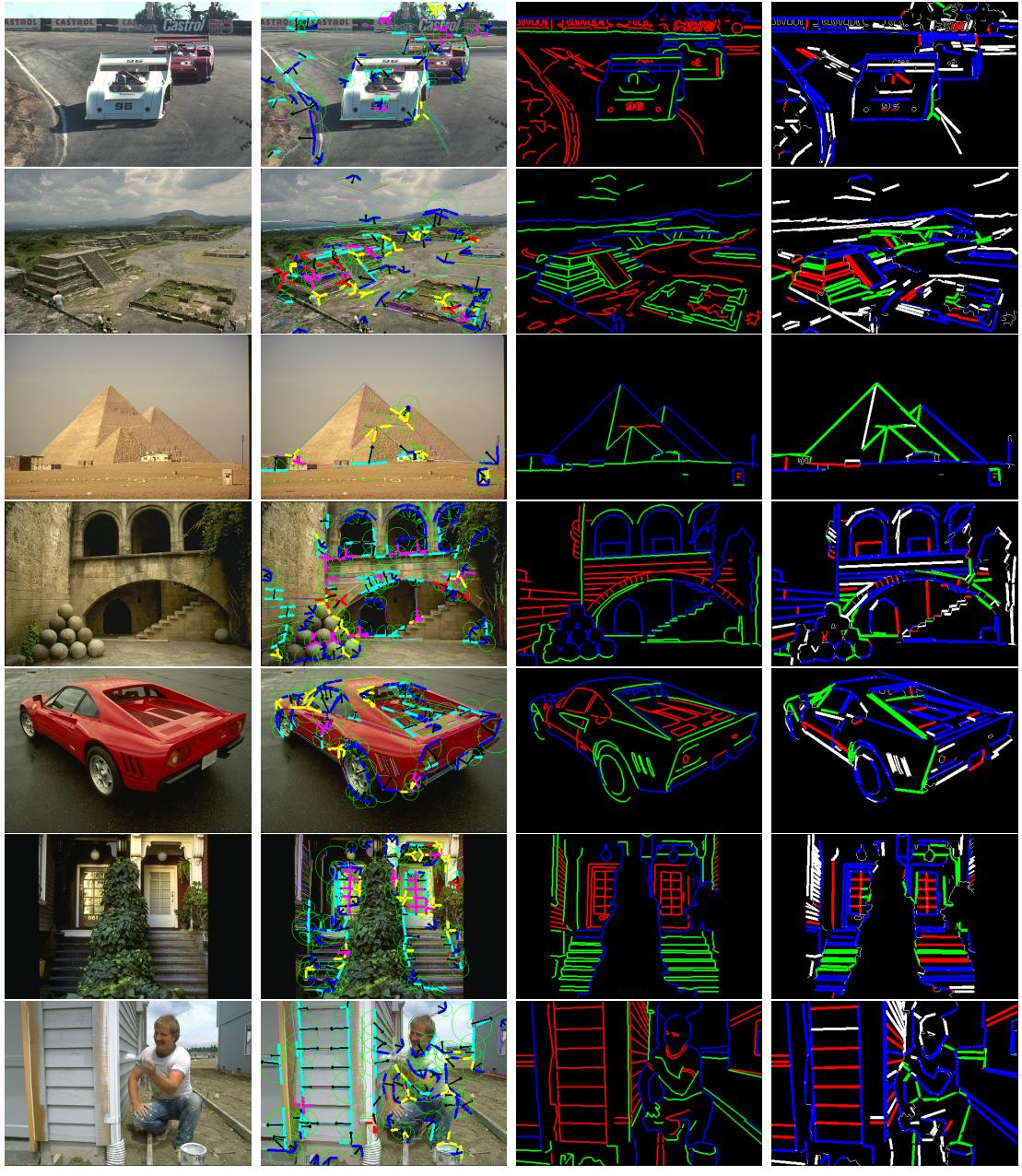
Figure 22 shows selected images from algorithm testing. In it, detected and classified junctions are shown as a diagnostic, allowing observers to determine the reason for

accurate or inaccurate classification.

Although only ten examples are shown in this thesis spanning Figures 19, 19, 21, and 22, actual testing was performed using a hundred select images from BSDS-RIND [2]. Table 10 shows the average metrics across all one hundred tests. Note that these tests focused on images with architectural elements or structures, leaving behind purely natural images showing landscapes, animals, or just people. For this reason, performance could drop if applied to other types of images.

| Class | Metric | | |
|-------|-----------|--------|----------|
| | 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 |

Table 10: Average metrics after performing tests using 100 images from the BSDS-RIND [2]. Selected images for testing included mostly architectural images, similar to those found in Figure 22.



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

Figure 22: Various edge classification results. Junctions are shown in diagnostic capacity. OB, SC, and RC edges are shown in blue, green, and red respectively.

5. Discussion

5.1 Contributions

This research explored the possibility of complementing junctions with other junctions to perform global junction analysis. It was found that junctions by themselves can be a strong cue for edge classification, however, overall performance heavily relies on having an accurate method of junction detection which can exclude noise and shadows from natural scenes.

As an experiment to analyze the findings of vision psychology, this method works well with simplified scenes, however, more complex scenes require additional cues for an accurate analysis. Lenient metrics and hand-picked images lead to results which show promise, but scoring by itself seems lacking, which is why local statistical analysis might still be needed.

Junctions and their interactions have been further explored during this paper, providing an initial scoring approach to classifying edges. These scores could be further adjusted in future research, but as a first step in defining the numerical properties of junction interaction, results from synthetic scenes provided a positive outlook. Edge classification was actually performed to some extent, although this specific algorithm would perhaps be best applied in artistic fields to create a possible interpretation of a drawing when seen by a person.

5.2 Limitations

The BSDS-RIND [2] data set contains mainly natural images and few indoor scenes. Results could improve greatly when applied to indoor scenes; however, perhaps it would be worth approaching the base junction detection differently to improve the results in natural scenes themselves.

5.3 Future Work

5.3.1 Boundary Type and Definition

Occlusion boundaries and surface change boundaries are often misclassified as each other. This is worthy of note as it might hint to the fact that objects with a complicated geometry are often made up of smaller objects (i.e. their parts). Further definition of fundamental differences between these two classes could help improve upon the views of how junctions interact with each other.

A similar case occurs with surface change and reflectance change boundaries. Often, parts of complex objects (e.g. buildings) change color as their surface changes. This means that RCs and SCs are not mutually exclusive, and it would be pertinent to perhaps devise an additional category which applies when both situations occur.

5.3.2 Junction Order on Edge

L-junctions and Y-junctions can be seen as 'limiters' as they are found on corners of objects, being the junctions which provide the strongest cues for image segmentation. However, junction misclassification might make it seem like these types of junctions can be found in the middle of an edge. As quality control, it would be worth to check the order of junctions on edges, and ignore any limiters found between others. These junctions are only found in extremes unless spotted due to reflectance changes.

Further concerning junction interaction, it would be worth to devise an algorithm which could identify continuous edges which are partially occluded. The MCMLSD [17] would be a good place to start for this research, as it works with segments of globally optimal lines. By doing this, more junctions could be grouped, analyzed perhaps not by scoring but merely by order and type, following vision research.

5.3.3 Vision Statistics

As this thesis sought to provide junction support with other junctions, no local statistics were employed. However, for the tasks of edge classification and beyond, performing a statistical analysis surrounding junctions might help improve results. Adelson [1] mentions how contrast is also a strong cue in mammalian vision. Further, differences in color could be analyzed to help accurately differentiate RCs.

Prior probability and a probabilistic model would also be worth exploring. [9] employs a probabilistic model, however, perhaps another study exploring the statistical nature of edges and junctions would provide further insight. Adelson [1] also speaks about the unique statistics of each atmosphere, however, no concrete research has bore fruit as to how to accurately determine said statistics.

6. Conclusions

This thesis builds upon a line-segment-detection-based method for junction detection. Further, this thesis utilizes the detected junctions to perform a global junction analysis for edge classification. Results indicate that although junctions can provide a promising analysis of architectural and indoor scenes, the nuances and statistics of the natural world require further cues to obtain useful results.

Ideally, this method would be used exclusively on synthetic, indoor, or architectural scenes unless a more robust junction detection method serves as its base. The author believes further research should focus on exploring the numerical properties of junction interaction, perhaps using hand-labeled junctions to limit the variables when testing.

References

- [1] E. H. Adelson, “Lightness perception and lightness illusions,” *The new cognitive neurosciences*, p. 339, 2000.
- [2] M. Pu, Y. Huang, Q. Guan, and H. Ling, “Rindnet: Edge detection for discontinuity in reflectance, illumination, normal and depth,” 2021.
- [3] A. Gilchrist, C. Kossyfidis, F. Bonato, T. Agostini, J. Cataliotti, X. Li, B. Spehar, V. Annan, and E. Economou, “An anchoring theory of lightness perception,” *Psychol. Rev.*, vol. 106, pp. 795–834, Oct. 1999.
- [4] E. H. Land and J. J. McCann, “Lightness and retinex theory,” *J. Opt. Soc. Am.*, vol. 61, pp. 1–11, Jan. 1971.
- [5] X. Li and A. L. Gilchrist, “Relative area and relative luminance combine to anchor surface lightness values,” *Percept. Psychophys.*, vol. 61, pp. 771–785, July 1999.
- [6] 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*, pp. 1229–1232, 2009.
- [7] B. L. Anderson, “A theory of illusory lightness and transparency in monocular and binocular images: the role of contour junctions,” *Perception*, vol. 26, no. 4, pp. 419–453, 1997.
- [8] J. McDermott, “Psychophysics with junctions in real images,” *Perception*, vol. 33, no. 9, pp. 1101–1127, 2004.
- [9] 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, pp. 13–13, 08 2014.
- [10] C. DiMattina, S. A. Fox, and M. S. Lewicki, “Detecting natural occlusion boundaries using local cues,” *Journal of Vision*, vol. 12, pp. 15–15, 12 2012.

- [11] 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)*, pp. 2681–2689, 2017.
- [12] 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.
- [13] J. Elder, “Are edges incomplete?,” *Computer Vision*, vol. 34, 07 1999.
- [14] 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.
- [15] C. DiMattina, J. Burnham, B. Guner, and H. Yerxa, “Distinguishing shadows from surface boundaries using local achromatic cues,” *bioRxiv*, 2022.
- [16] N. Kiryati, Y. Eldar, and A. Bruckstein, “A probabilistic hough transform,” *Pattern Recognition*, vol. 24, no. 4, pp. 303–316, 1991.
- [17] 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*, pp. 2031–2039, 2017.
- [18] 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.
- [19] C. G. Harris and M. J. Stephens, “A combined corner and edge detector,” in *Alvey Vision Conference*, 1988.
- [20] 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*, pp. 1–8, 2008.

- [21] 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.
- [22] Z. Tu and X. Chen, “Junction detection based on line segments,” in *2014 9th IEEE Conference on Industrial Electronics and Applications*, pp. 1231–1234, 2014.
- [23] 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.
- [24] 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)*, pp. 142–145, 2020.
- [25] 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)*, pp. 6869–6878, 10 2021.
- [26] K. Huang, Y. Wang, Z. Zhou, T. Ding, S. Gao, and Y. Ma, “Learning to parse wireframes in images of man-made environments,” *IEEE Conference on Computer Vision and Pattern Recognition (2018)*, pp. 626–635, 2020.
- [27] 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.
- [28] J. E. Bresenham, “Algorithm for computer control of a digital plotter,” *IBM Systems Journal*, vol. 4, no. 1, pp. 25–30, 1965.
- [29] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, “Contour detection and hierarchical image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, pp. 898–916, May 2011.