Artificial Vision and Pattern Recognition

Laboratory Report

Edge Detection, Corner Detection, and Morphological Operations

Audigier Matteo

October 29, 2025

Contents

1 Introduction

Computer vision and pattern recognition are fundamental fields in artificial intelligence that enable machines to interpret and understand visual information. This laboratory session focuses on three essential low-level image processing techniques:

- **Edge Detection**: Identifying significant transitions in image intensity that often correspond to object boundaries.
- **Corner Detection**: Locating points where two or more edges meet, which are highly distinctive features for image analysis.
- **Morphological Operations**: Applying set theory-based operations to enhance and extract specific image features.

These techniques form the foundation for higher-level vision tasks such as object recognition, image segmentation, and feature extraction. The source code for the implementations can be found at https://github.com/audigiem/AVPR_labs.

2 Task 1: Edge Detection

2.1 Objective

The objective of this task was to implement and compare different edge detection filters to understand their characteristics and performance on grayscale images.

2.2 Methodology

Edge detection identifies points in an image where the brightness changes sharply. We implemented three gradient-based edge detection filters:

2.2.1 Sobel Filter

The Sobel operator uses two 3×3 convolution kernels to compute approximations of the horizontal and vertical derivatives:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * I \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * I$$

The magnitude is computed as: $G = \sqrt{G_x^2 + G_y^2}$

2.2.2 Prewitt Filter

The Prewitt operator uses simpler kernels with uniform weights:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} * I \quad G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} * I$$

2.2.3 Scharr Filter

The Scharr operator uses optimized 3×3 kernels for better rotational symmetry:

$$G_x = \begin{bmatrix} -3 & 0 & 3 \\ -10 & 0 & 10 \\ -3 & 0 & 3 \end{bmatrix} * I \quad G_y = \begin{bmatrix} -3 & -10 & -3 \\ 0 & 0 & 0 \\ 3 & 10 & 3 \end{bmatrix} * I$$

2.3 Results

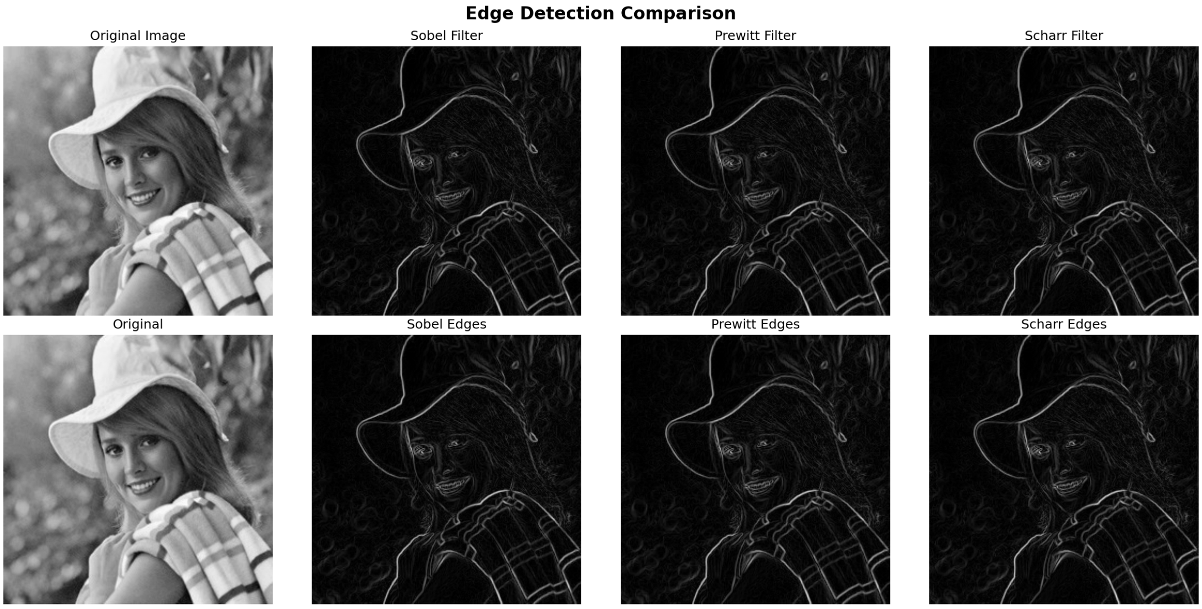

Figure 1: Comparison of edge detection filters: Sobel, Prewitt, and Scharr

Figure 1 shows the comparison of all three edge detection filters applied to the same input image. The results demonstrate subtle but important differences in edge detection quality.

Figure 2: Directional edge detection showing horizontal and vertical edges separately

Figure 2 illustrates the directional sensitivity of each filter, showing how edges in different orientations are detected.

2.4 Analysis and Discussion

Table 1: Comparison of Edge Detection Filters

Characteristic	**Sobel**	**Prewitt**	**Scharr**
Sensitivity	Medium	Low	High
Noise Resistance	Good	Medium	Lower
Computational Cost	Medium	Low	Medium
Edge Localization	Good	Medium	Excellent
Rotational Symmetry	Good	Medium	Excellent

Key Findings:

- **Sobel Filter**: Provides the best balance between noise suppression and edge detection accuracy. Most widely used in practice.
- **Prewitt Filter**: Simpler implementation with slightly more noise sensitivity. Suitable for applications where computational efficiency is critical.
- **Scharr Filter**: Superior rotational invariance and edge localization. Better for detecting small-scale edges and when precise edge orientation is important.
- All three filters effectively detect edges but with different trade-offs between sensitivity, noise handling, and computational cost.

3 Task 2: Corner Detection

3.1 Objective

The objective was to implement the Harris Corner Detection method and investigate how different parameters affect corner detection performance.

3.2 Methodology

The Harris corner detector identifies corners by analyzing the local intensity structure of an image. The algorithm computes the structure tensor (also called the second moment matrix):

$$M = \sum_{(x,y)\in W} \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix}$$

where I_x and I_y are image derivatives, and W is a local window.
The Harris corner response is computed as:

$$R = \det(M) - k \cdot \text{trace}(M)^2 = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2$$

Points where R exceeds a threshold are classified as corners.

3.2.1 Parameters

The Harris detector has several key parameters:

- **Block Size**: Size of the neighborhood considered for corner detection
- **Aperture Size (ksize)**: Size of the Sobel kernel used for gradient computation
- **k**: Harris detector free parameter (typically 0.04-0.06)
- **Threshold**: Minimum corner response value to accept

3.3 Results

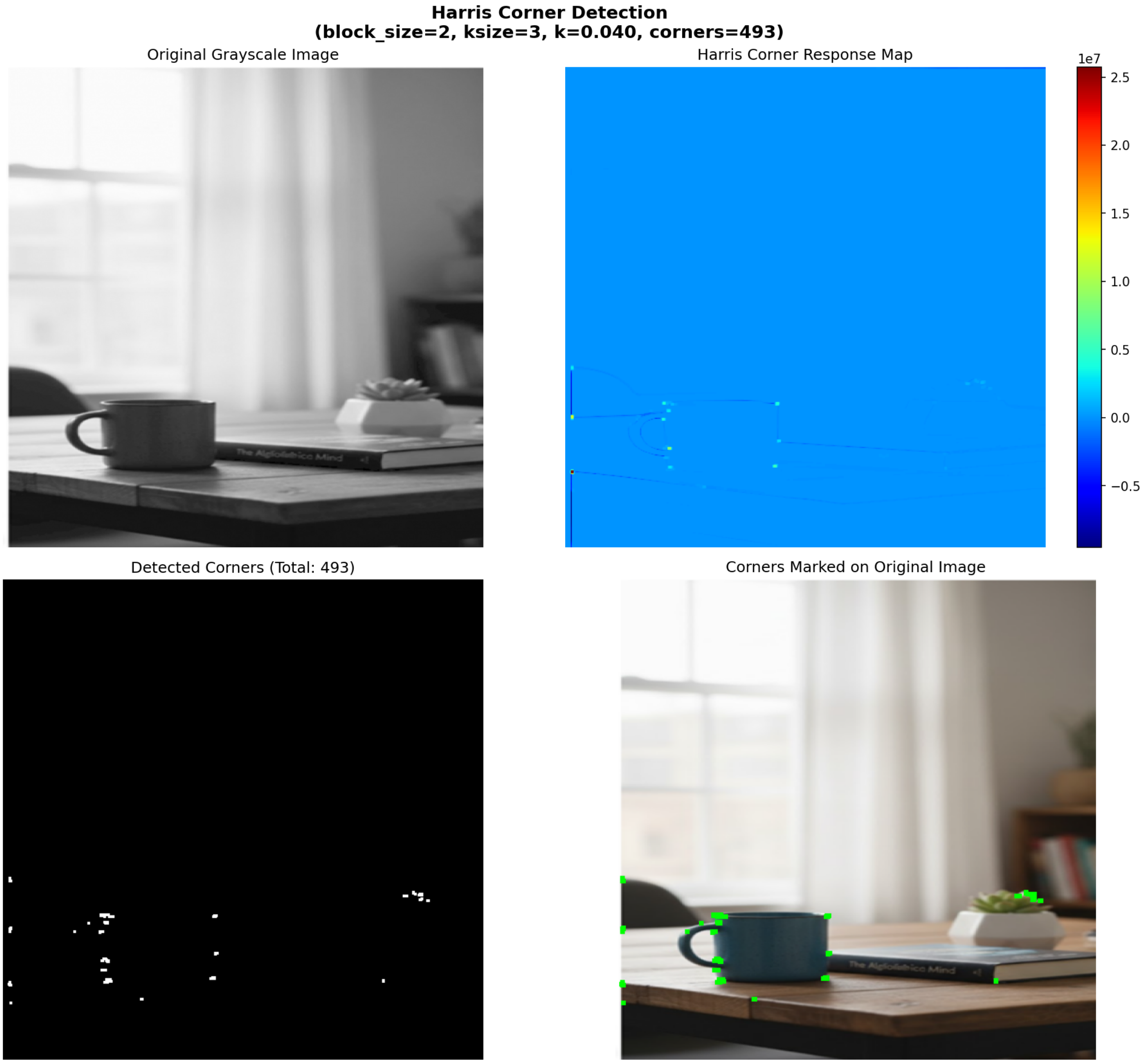

Figure 3: Harris corner detection with default parameters: block_size=2, ksize=3, k=0.04

Figure 3 shows the Harris corner detection results with default parameters, including the corner response map and detected corner locations.

Figure 4: Comparison of Harris corner detection with different parameter configurations

Figure 4 demonstrates how varying parameters affects the number and quality of detected corners.

3.4 Analysis and Discussion

3.4.1 Effect of Block Size

- **Smaller blocks (2-3)**: Detect finer corners and small-scale features. More sensitive to noise and texture variations.
- **Larger blocks (5-7)**: Detect more prominent, stable corners. Better noise resistance but may miss fine details.

3.4.2 Effect of Aperture Size

- **Smaller aperture (3)**: Faster computation, sharper gradient estimates, more sensitive to noise.
- **Larger aperture (5-7)**: More smoothing, better noise handling, may blur fine corners.

3.4.3 Effect of k Parameter

- **Lower k (0.02)**: More sensitive, detects more corners including weaker ones. Risk of false positives.
- **Higher k (0.06)**: More selective, detects only strong corners. May miss subtle features.

3.4.4 Effect of Threshold

- **Lower threshold (0.005)**: More corners detected, including weaker responses.
- **Higher threshold (0.02)**: Fewer, higher-quality corners. Better for feature matching.

Recommendations:

- For general-purpose corner detection: block_size=2, ksize=3, k=0.04, threshold=0.01
- For noisy images: Increase block_size and ksize
- For fine detail preservation: Use smaller block_size with moderate threshold
- For robust feature matching: Increase threshold and k to select only strong corners

4 Task 3: Morphological Operations

4.1 Objective

The objective was to implement morphological operations for image enhancement and explore the effects of different structuring elements, sizes, and iteration counts.

4.2 Methodology

Morphological operations are based on set theory and process images based on their shapes. They use a structuring element (kernel) to probe and interact with image structures.

4.2.1 Basic Operations

Erosion: Shrinks bright regions by removing pixels at object boundaries.

$$A \ominus B = \{z \in E | B_z \subseteq A\}$$

Dilation: Expands bright regions by adding pixels at object boundaries.

$$A \oplus B = \{z \in E | (\hat{B})_z \cap A \neq \emptyset\}$$

Opening: Erosion followed by dilation. Removes small objects and smooths contours.

$$A \circ B = (A \ominus B) \oplus B$$

Closing: Dilation followed by erosion. Fills small holes and connects nearby objects.

$$A \bullet B = (A \oplus B) \ominus B$$

4.2.2 Advanced Operations

Morphological Gradient: Difference between dilation and erosion. Highlights edges.

$$\text{Gradient}(A) = (A \oplus B) - (A \ominus B)$$

Top Hat: Difference between original and opening. Extracts small bright features.

$$\text{TopHat}(A) = A - (A \circ B)$$

Black Hat: Difference between closing and original. Extracts small dark features.

$$\text{BlackHat}(A) = (A \bullet B) - A$$

4.3 Results

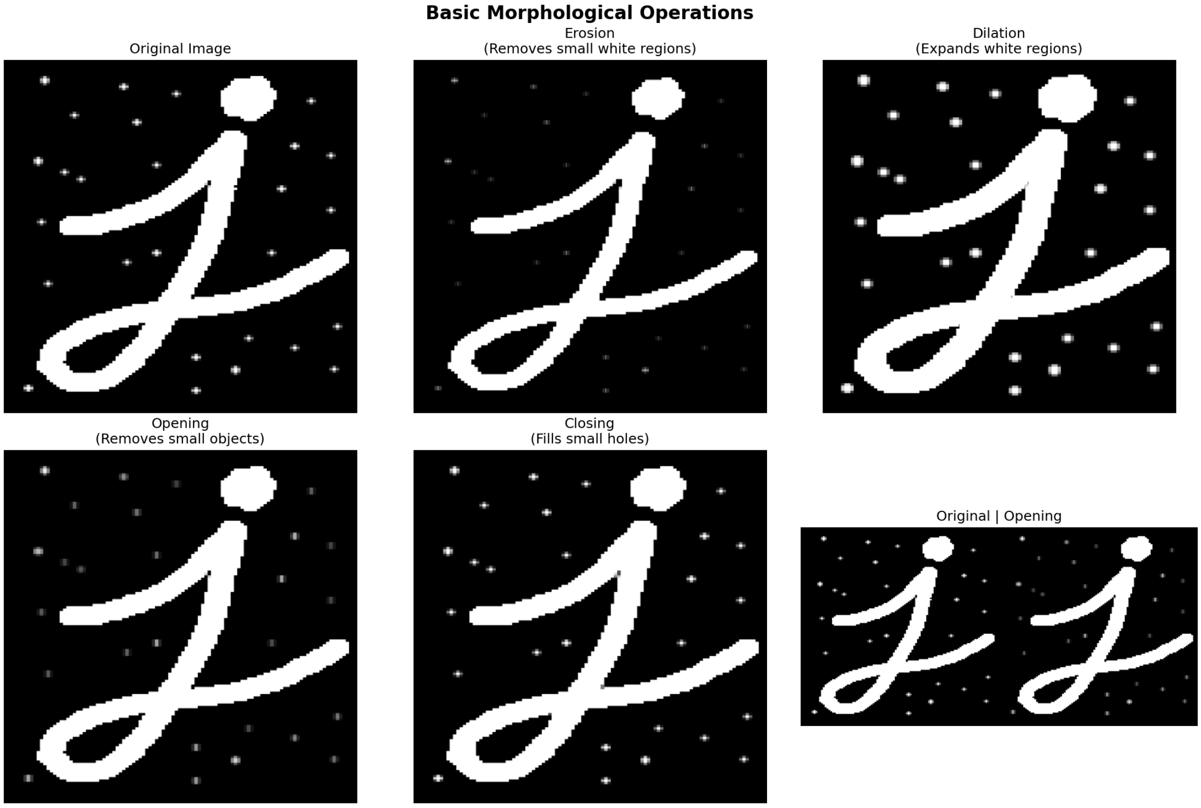

Figure 5: Basic morphological operations: erosion, dilation, opening, and closing

Figure 5 shows the four fundamental morphological operations and their effects on the input image.

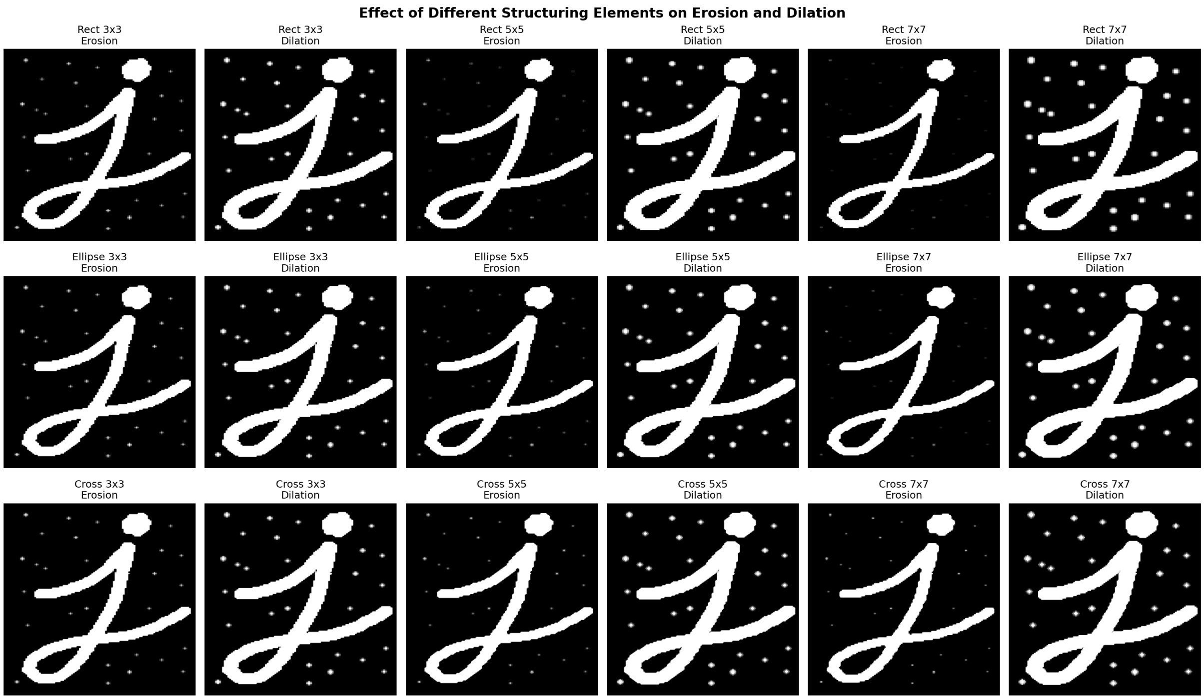

Figure 6: Effect of different structuring elements (rectangle, ellipse, cross) and sizes

Figure 6 compares how different structuring element shapes and sizes affect erosion and dilation operations.

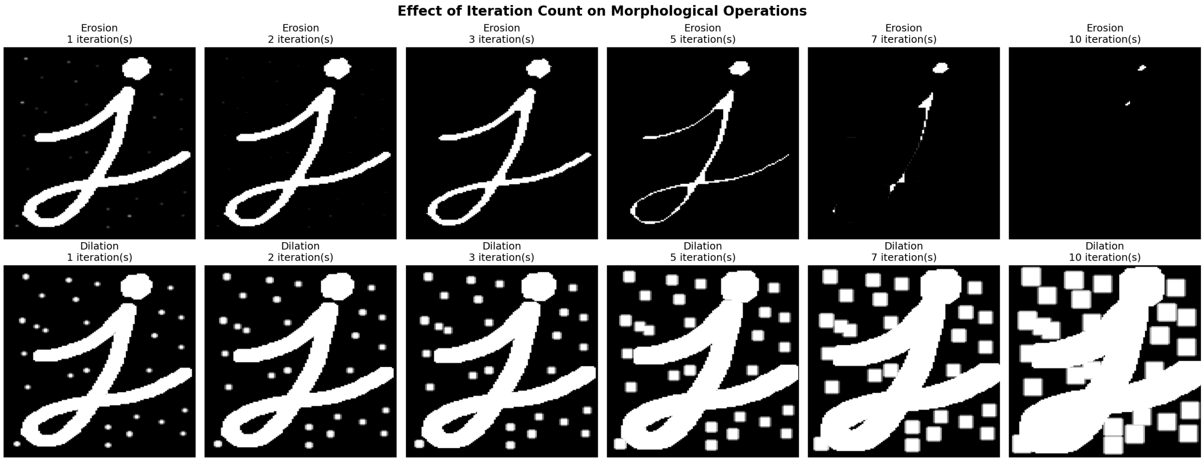

Figure 7: Effect of iteration count on erosion and dilation operations

Figure 7 illustrates the progressive effect of multiple iterations on morphological operations.

4.4 Analysis and Discussion

4.4.1 Structuring Element Selection

Table 2: Characteristics of Different Structuring Elements

Element	Characteristics and Applications
Rectangle	Directional effect. Good for linear features and text. Preserves horizontal/vertical structures.
Ellipse	Isotropic (rotation-invariant). Good for circular and blob-like features. Most natural for general use.
Cross	Minimal effect. Preserves corners better. Good for preserving shape details while reducing noise.

4.4.2 Size and Iteration Effects

Size Impact:

- Larger structuring elements create stronger effects
- 3×3: Subtle changes, preserves fine details
- 5×5: Moderate effects, good balance
- 7×7 and larger: Strong effects, may remove desired features

Iteration Impact:

- Each iteration applies the operation again to the result
- Erosion iterations: Progressive shrinking, eventually objects disappear
- Dilation iterations: Progressive expansion, objects may merge
- Multiple iterations with small kernels Single iteration with large kernel
- More control with iterations, but higher computational cost

4.4.3 Practical Applications

Table 3: Applications of Morphological Operations

Operation	Applications
Erosion	Remove salt noise, separate touching objects, thin lines
Dilation	Remove pepper noise, fill gaps, thicken lines, connect components
Opening	Remove small bright objects, smooth contours, background removal
Closing	Fill holes, connect nearby objects, remove small dark regions
Gradient	Edge detection, boundary extraction, feature emphasis
Top Hat	Extract small bright features, uneven illumination correction
Black Hat	Extract small dark features, detect defects

Key Insights:

- Opening is effective for removing noise (small bright spots) while preserving object shapes
- Closing is excellent for filling gaps and connecting broken text or lines
- Morphological gradient provides a simpler alternative to differential edge detectors
- Top hat and black hat transforms are powerful for feature extraction in non-uniform lighting
- Combined operations (e.g., opening + gradient) can create sophisticated enhancement pipelines

Acknowledgments

This work was completed as part of the Artificial Vision and Pattern Recognition course. The implementation utilized OpenCV, NumPy, SciPy, and Matplotlib libraries.

References

[1] Sobel, I., & Feldman, G. (1968). *A 3x3 isotropic gradient operator for image processing.* Presented at the Stanford Artificial Intelligence Project.

[2] Harris, C., & Stephens, M. (1988). *A combined corner and edge detector.* Alvey vision conference, 15(50), 10-5244.

[3] Gonzalez, R. C., & Woods, R. E. (2008). *Digital image processing* (3rd ed.). Prentice Hall.

[4] Scharr, H. (2000). *Optimal operators in digital image processing.* PhD thesis, University of Heidelberg.

[5] Bradski, G. (2000). *The OpenCV Library.* Dr. Dobb's Journal of Software Tools.

[6] Prewitt, J. M. (1970). *Object enhancement and extraction.* Picture processing and Psychopictorics, 10(1), 15-19.

[7] Serra, J. (1982). *Image analysis and mathematical morphology* (Vol. 1). Academic Press.

[8] Noble, J. A. (1989). *Finding corners.* Image and Vision Computing, 6(2), 121-128.