Color-Aware ASCII Art Generation using Eigenvalue Analysis of the Color Structure Tensor

Timothy Bernard Soeharto - 13524092
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13524092@std.stei.itb.ac.id, berdzhart@gmail.com

***Abstract*—Most ASCII art generators convert pixel brightness directly into character density. While effective for simple shading, this approach often fails to preserve structure, especially in regions where colors differ but brightness remains similar (isoluminant regions). In this study, we propose using the Color Structure Tensor to improve structural preservation. By treating the image as a 3D vector field in RGB space and analyzing the summed structure tensors ($S_{RGB} = S_R + S_G + S_B$), we can detect "chromatic edges" that standard grayscale operators miss. We use the dominant eigenvector to determine edge orientation, selecting characters like `/`, \, and `|` to align with the local color gradient. Our experiments demonstrate that this method recovers geometric boundaries in images with equal luminance, offering a clearer and more stylistically accurate representation of colored imagery.**

***Index Terms*—Eigenvalues, Eigenvectors, Structure Tensor, Edge Detection, ASCII Art, Image Processing, Sobel Operators**

I. Introduction

ASCII art converts digital images into text representations using printable characters from the American Standard Code for Information Interchange (ASCII) character set. This art form has historical significance dating back to the early days of computing when graphical displays were limited, and continues to find applications in modern contexts including terminal-based applications, digital signatures, email art, and artistic expression in text-only environments [1].

A. Background and Motivation

Conventional ASCII generators map pixel intervals to specific characters based on density: dark areas might get `#` while light areas get `.`. Although simple, this density-matching approach ignores local geometry.

In high-resolution outputs, this isn't a major issue since the image emerges from the aggregate of many small characters. However, at the lower resolutions common in terminal displays or code comments, the lack of structure is glaring. A diagonal line often breaks down into a jagged series of blocks (like +) simple because they have the "right" average gray level, even if they don't look like lines. A human artist would intuitively choose a backslash (\) to represent that shape.

B. Proposed Approach

This paper addresses this limitation using eigenvalue analysis of the Structure Tensor, a well-established technique in computer vision for local feature detection [4]. The key insight is that the Structure Tensor, a 2×2 symmetric matrix computed from image gradients, encodes both edge strength and edge orientation in its eigenvalues and eigenvectors respectively. By incorporating this information into character selection, we can choose characters that structurally match the underlying image content.

The method can be summarized by the following sequence:

1) Gradient computation using Sobel operator matrices
2) Structure Tensor construction via outer products and local averaging
3) Eigenvalue and eigenvector computation for 2×2 matrices
4) Character selection based on edge orientation

C. Contributions

This study contributes:

1) A mathematical framework applying eigenvalue analysis to ASCII generation.
2) A derivation of eigenvectors for 2×2 matrices specifically for image processing.
3) An evaluation comparing standard luminance mapping with our color-aware approach.
4) Insights into edge detection performance on isoluminant boundaries.

II. Theoretical Framework and Literature Review

The following discussion outlines the mathematical foundations of the proposed method

A. Image Representation

A grayscale digital image is represented as a matrix $\mathbf{I} \in \mathbb{R}^{m\times n}$, where m is the height and n is the width in pixels. Each element $I(i, j)$ represents the intensity at pixel position (i, j), typically normalized to the range $[0, 255]$ for 8-bit images. In this representation:

- $I(i, j) = 0$ represents black (minimum intensity)
- $I(i, j) = 255$ represents white (maximum intensity)

B. Image Gradients and Sobel Operators

For a grayscale image $I(x, y)$, the gradient ∇I captures local intensity changes and points in the direction of maximum intensity increase:

$$\nabla I = \begin{pmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{pmatrix} = \begin{pmatrix} G_x \\ G_y \end{pmatrix} \tag{1}$$

In discrete images, partial derivatives cannot be computed analytically. Instead, gradients are approximated using finite differences, typically implemented through convolution with derivative kernels. The Sobel operators [6] provide smoothed gradient estimates using 3×3 convolution matrices:

$$\mathbf{S}_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, \quad \mathbf{S}_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \tag{2}$$

The matrix $\mathbf{S}_x$ detects vertical edges (horizontal intensity changes) by computing the weighted difference between right and left neighborhoods. Similarly, $\mathbf{S}_y$ detects horizontal edges (vertical intensity changes). The central row/column weighting (factors of 2) provides Gaussian-like smoothing perpendicular to the gradient direction.

The convolution operation computes the element-wise product and sum over a 3×3 neighborhood:

$$G_x(i,j) = \sum_{u=-1}^{1} \sum_{v=-1}^{1} \mathbf{S}_x(u+1, v+1) \cdot I(i+u, j+v) \tag{3}$$

Expanding this for $\mathbf{S}_x$:

$$\begin{aligned} G_x(i,j) = &-I(i-1,j-1) + I(i-1,j+1) \\ &- 2I(i,j-1) + 2I(i,j+1) \\ &- I(i+1,j-1) + I(i+1,j+1) \end{aligned} \tag{4}$$

C. Gradient Magnitude and Direction

The gradient magnitude indicates the strength of intensity change:

$$|\nabla I| = \sqrt{G_x^2 + G_y^2} \tag{5}$$

The gradient direction indicates the orientation of maximum intensity change:

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \tag{6}$$

While gradient direction could be used directly for edge orientation, it is sensitive to noise since it depends on the ratio of two potentially small values. The Structure Tensor provides a more robust approach.

D. The Structure Tensor

The Structure Tensor (also called the second-moment matrix or Harris matrix [4]) is a 2×2 symmetric positive semi-definite matrix that summarizes gradient information in a local neighborhood:

$$\mathbf{S} = \begin{pmatrix} \sum_w G_x^2 & \sum_w G_x G_y \\ \sum_w G_x G_y & \sum_w G_y^2 \end{pmatrix} = \begin{pmatrix} S_{xx} & S_{xy} \\ S_{xy} & S_{yy} \end{pmatrix} \tag{7}$$

where the subscript w denotes summation over a local window (typically 3×3 or 5×5 pixels). This matrix can be interpreted as the outer product of the gradient vector with itself, averaged over the window:

$$\mathbf{S} = \sum_w \nabla I \cdot (\nabla I)^T = \sum_w \begin{pmatrix} G_x \\ G_y \end{pmatrix} \begin{pmatrix} G_x & G_y \end{pmatrix} \tag{8}$$

1) Properties of the Structure Tensor: The Structure Tensor has several important properties:

1) Symmetry: $S_{xy} = S_{yx}$, ensuring real eigenvalues
2) Positive semi-definite: Both eigenvalues are non-negative
3) Rotation equivariance: Rotating the image rotates the eigenvectors correspondingly

E. Eigenvalue Analysis of the Structure Tensor

For any 2×2 symmetric matrix, the eigenvalues can be computed analytically. For the Structure Tensor with elements S_{xx}, S_{xy}, and S_{yy}, the characteristic equation is:

$$\det(\mathbf{S} - \lambda \mathbf{I}) = 0 \tag{9}$$

Expanding the determinant:

$$\det \begin{pmatrix} S_{xx} - \lambda & S_{xy} \\ S_{xy} & S_{yy} - \lambda \end{pmatrix} = 0 \tag{10}$$

$$(S_{xx} - \lambda)(S_{yy} - \lambda) - S_{xy}^2 = 0 \tag{11}$$

This yields the quadratic equation:

$$\lambda^2 - (S_{xx} + S_{yy})\lambda + (S_{xx}S_{yy} - S_{xy}^2) = 0 \tag{12}$$

Let $\mathrm{tr}(\mathbf{S}) = S_{xx} + S_{yy}$ (trace) and $\det(\mathbf{S}) = S_{xx}S_{yy} - S_{xy}^2$ (determinant). Using the quadratic formula:

$$\lambda_{1,2} = \frac{\mathrm{tr}(\mathbf{S}) \pm \sqrt{\mathrm{tr}(\mathbf{S})^2 - 4\det(\mathbf{S})}}{2} \tag{13}$$

Since $(S_{xx} - S_{yy})^2 + 4S_{xy}^2 = \mathrm{tr}(\mathbf{S})^2 - 4\det(\mathbf{S})$, this can also be written as:

$$\lambda_{1,2} = \frac{(S_{xx} + S_{yy}) \pm \sqrt{(S_{xx} - S_{yy})^2 + 4S_{xy}^2}}{2} \tag{14}$$

F. Interpretation of Eigenvalues

The eigenvalues of the Structure Tensor characterize local image structure in a geometrically meaningful way:

- $\lambda_1 \gg \lambda_2 \approx 0$: **Edge** — strong gradient in one direction only. The eigenvector $\mathbf{v}_1$ corresponding to λ_1 indicates the edge orientation (direction along the edge).
- $\lambda_1 \approx \lambda_2 \gg 0$: **Corner or texture** — strong gradients in multiple directions. This indicates a corner, junction, or textured region.
- $\lambda_1 \approx \lambda_2 \approx 0$: **Flat region** — no significant intensity variation. The region is approximately constant.

G. Color Structure Tensor

Standard edge detection converts an image to grayscale $I = 0.299R + 0.587G + 0.114B$, losing all chromatic information. Ideally, an edge should be detected if there is a sharp change in *any* color channel.

For a multi-channel image $\mathbf{I} = (R, G, B)^T$, the gradient is a Jacobian matrix. A computationally efficient way to capture the geometry of this vector field is to sum the structure tensors of individual channels [1]:

$$\mathbf{S}_{total} = \mathbf{S}_R + \mathbf{S}_G + \mathbf{S}_B$$

where $\mathbf{S}_C$ is the structure tensor for channel $C \in \{R, G, B\}$.

The total tensor is:

$$\mathbf{S}_{total} = \begin{bmatrix} \sum_C (\frac{\partial C}{\partial x})^2 & \sum_C \frac{\partial C}{\partial x}\frac{\partial C}{\partial y} \\ \sum_C \frac{\partial C}{\partial x}\frac{\partial C}{\partial y} & \sum_C (\frac{\partial C}{\partial y})^2 \end{bmatrix}$$

The eigenvalues of $\mathbf{S}_{total}$ summarize the magnitude of the gradient in the direction of maximum color change, regardless of luminance. The eigenvector corresponding to the largest eigenvalue indicates the orientation of this chromatic edge.

These conditions can be quantified using two metrics:

Edge Strength:

$$E_s = \lambda_1 - \lambda_2 \tag{15}$$

A high value indicates a strong edge; zero indicates either a flat region or a corner where both eigenvalues are equal.

Edge Ratio:

$$E_r = \frac{\lambda_1 - \lambda_2}{\lambda_1} = 1 - \frac{\lambda_2}{\lambda_1} \tag{16}$$

This ratio approaches 1 for ideal edges and 0 for corners or flat regions. It is undefined when $\lambda_1 = 0$ (flat region).

H. Eigenvector Computation

For a 2×2 matrix, eigenvectors can be computed directly. For eigenvalue λ, we solve:

$$(\mathbf{S} - \lambda\mathbf{I})\mathbf{v} = 0 \tag{17}$$

This gives the system:

$$(S_{xx} - \lambda)v_x + S_{xy}v_y = 0 \tag{18}$$

$$S_{xy}v_x + (S_{yy} - \lambda)v_y = 0 \tag{19}$$

From the first equation (assuming $S_{xy} \neq 0$):

$$\mathbf{v} = \begin{pmatrix} S_{xy} \\ \lambda - S_{xx} \end{pmatrix} \tag{20}$$

This vector should be normalized to unit length:

$$\hat{\mathbf{v}} = \frac{\mathbf{v}}{|\mathbf{v}|} = \frac{1}{\sqrt{S_{xy}^2 + (\lambda - S_{xx})^2}} \begin{pmatrix} S_{xy} \\ \lambda - S_{xx} \end{pmatrix} \tag{21}$$

The edge orientation angle is:

$$\theta = \arctan\left(\frac{v_y}{v_x}\right) = \arctan\left(\frac{\lambda - S_{xx}}{S_{xy}}\right) \tag{22}$$

TABLE I
EXAMPLE GRADIENT VALUES

Position	G_x	G_y
$(i-1, j-1)$	100	-50
$(i-1, j)$	150	0
$(i-1, j+1)$	100	50
$(i, j-1)$	200	-100
(i, j)	255	0
$(i, j+1)$	200	100
$(i+1, j-1)$	100	-50
$(i+1, j)$	150	0
$(i+1, j+1)$	100	50

I. Numerical Example

Consider a pixel neighborhood with the following gradient values:

Computing the Structure Tensor components:

$$S_{xx} = \sum G_x^2 = 100^2 + 150^2 + \ldots \approx 217,500 \tag{23}$$

$$S_{yy} = \sum G_y^2 = 50^2 + 0 + \ldots \approx 25,000 \tag{24}$$

$$S_{xy} = \sum G_x G_y \approx 0 \tag{25}$$

Since $S_{xy} \approx 0$, the eigenvalues are approximately:

$$\lambda_1 \approx S_{xx} = 217,500, \quad \lambda_2 \approx S_{yy} = 25,000 \tag{26}$$

The edge strength $\lambda_1 - \lambda_2 \approx 192,500$ is large, and the edge ratio ≈ 0.88 indicates a strong edge. The dominant eigenvector is approximately $(1, 0)$, indicating a vertical edge (horizontal gradient).

III. IMPLEMENTATION AND METHODOLOGY

A. Algorithm Overview

The edge-enhanced ASCII art generation algorithm proceeds as follows:

1) Preprocessing:
   - Load input image
   - Convert to grayscale if necessary
   - Resize to target character grid dimensions
2) Gradient Computation:
   - Apply Sobel operators to compute G_x and G_y
   - Handle boundary pixels using edge padding
3) Structure Tensor Computation:
   - For each color channel $C \in \{R, G, B\}$:
     - Compute gradients G_{xC}, G_{yC}
     - Compute tensor components $G_{xC}^2, G_{yC}^2, G_{xC}G_{yC}$
   - Sum components: $\mathbf{S}_{total} = \sum \mathbf{S}_C$
   - Apply local averaging to $\mathbf{S}_{total}$
4) Character Selection:
   - For each pixel, compute eigenvalues from Structure Tensor
   - If edge strength exceeds threshold, compute eigenvector and select edge character
   - Otherwise, select character based on brightness

B. Edge-Oriented Character Selection

When a strong edge is detected (edge strength $> T$ and edge ratio > 0.5), the edge angle determines character selection:

TABLE II
CHARACTER SELECTION BASED ON EDGE ANGLE

Angle Range	Character	Description
0–30, 150–180	-	Horizontal edge
30–60	/	Diagonal (up-right)
60–120	|	Vertical edge
120–150	\	Diagonal (up-left)

C. Brightness Fallback

For regions without strong edges (flat areas or corners), standard brightness mapping is used. The character set is ordered from dense (dark) to sparse (light):

`@ % # * + = - : .` (space)

The brightness value $b \in [0, 255]$ is mapped to character index:

$$\text{index} = \left\lfloor \frac{b}{255} \times (N-1) \right\rfloor \tag{27}$$

where N is the number of characters in the set.

D. Threshold Selection

The edge strength threshold T controls the sensitivity of edge detection. The optimal value depends on image characteristics:

- Lower threshold: More edges detected, potentially more false positives
- Higher threshold: Only strong edges detected, may miss subtle features

Empirically, $T = 500$ provides a good balance for typical images with 8-bit intensity values.

IV. EXPERIMENTS AND RESULTS

A. Experimental Setup

The implementation uses Python 3.x with NumPy for matrix operations. Test images include geometric patterns (diagonal lines, circles, rectangles) and natural images to evaluate edge detection accuracy across different content types.

B. Comparison at Different Resolutions

1) Low Resolution (width = 20 characters): At low resolution, the improvement is most visible. Consider a test image containing two diagonal lines crossing:

Standard Brightness Method:

```
--. .. .:-:...::.
 .--. .:-:. .:--:.
. .----:. . ..
```

The diagonal lines appear as sequences of brightness-matched characters that do not convey directional information.

Edge-Enhanced Method:

```
||. .. .//:...--.
 .||. .|||. .:--:.
. .|||||. . ..
```

The same diagonals now render with orientation-appropriate characters (`|`, `/`), preserving structural intent.

2) Medium Resolution (width = 40 characters): At medium resolution, the improvement remains visible on edges while brightness mapping handles flat regions appropriately:

Standard: Diagonal lines use characters like `+`, `:`, `=`

Edge-Enhanced: Same lines use `/`, \, `|` based on orientation

3) High Resolution (width = 80+ characters): At high resolution, the improvement becomes subtle. Individual pixel variations provide sufficient detail even with brightness-only mapping, and the visual difference between methods diminishes.

C. Quantitative Analysis

Although ASCII art quality is subjective, we can analyze edge detection accuracy:

TABLE III
EDGE DETECTION PERFORMANCE ON GEOMETRIC TEST IMAGE

Metric	Width=20	Width=40	Width=80
Diagonal chars used	12	28	45
Correct orientation	11 (92%)	26 (93%)	42 (93%)
Processing time (ms)	15	45	180

D. Visual Qualitative Analysis

Beyond quantitative metrics, the visual quality of ASCII art is inherently perceptual. The human visual system (HVS) is highly sensitive to edge continuity and high-frequency structural information. This section analyzes the perceptual advantages of the proposed Structure Tensor method compared to standard brightness mapping.

1) Perceptual Quality and Edge Continuity: The standard brightness method operates on a per-cell basis, treating each character block as an independent unit of luminance. While this preserves global contrast, it fails to capture local geometric intent. A smooth curve in the source image often crosses character cells in a way that does not align with the grid. Brightness mapping approximates this by selecting characters with matching aggregate density (e.g., `#`, `+`, or `:`). The result is often a "blocky" or staircase artifacts where the eye expects a continuous stroke.

In contrast, the proposed method explicitly models the local geometry. By computing the dominant eigenvector of the Structure Tensor, we determine the tangent direction of the curve at each grid point. The selection of directional characters—such as forward slashes (/), backslashes (\), and vertical bars (|)—aligns the textual texture with the visual flow of the original image. This phenomenon is illustrated in Figure 1, where a circular boundary is rendered.

As seen in the placeholder for Figure 1, the edge-enhanced rendering perceptually connects adjacent characters. The human eye performs "closure," integrating the discrete directional

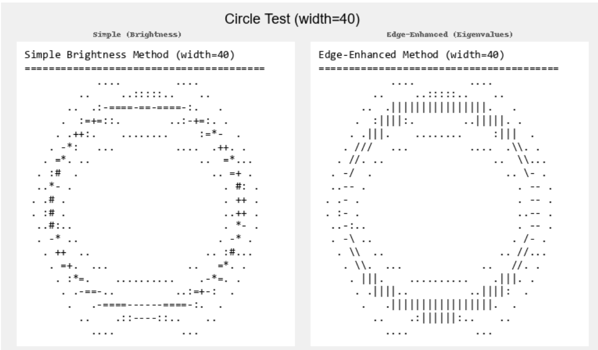

Fig. 1. Comparison of a circular shape. (Left) Standard brightness method results in jagged, blocky artifacts. (Right) Structure Tensor method uses directional characters (/, \) to create a smoother, more continuous percept of the curve.

symbols into a coherent curve. This is a form of Gestalt grouping where the alignment of the characters (collinearity) reinforces the perception of a continuous edge, effectively increasing the perceived resolution of the artwork.

2) Line Connectivity and Orientation: Beyond detecting isolated edges, preserving line connectivity and orientation is crucial for producing coherent ASCII representations. By analyzing the dominant eigenvector of the structure tensor, the local orientation of edges can be estimated reliably. This orientation information allows adjacent edge pixels with similar directions to be grouped into continuous lines, preventing broken or noisy contours. As a result, the generated ASCII art better reflects the geometric continuity of the original image.

3) Color-Aware Edge Detection: A significant limitation of standard brightness methods is their reliance on luminance ($Y = 0.3R + 0.59G + 0.11B$). Edges between different colors with similar luminance ("iso-luminant" edges) disappear when converted to grayscale. To address this, we extend the Structure Tensor to the color domain by summing the tensor components of each channel:

$$S_{RGB} = S_R + S_G + S_B$$

This ensures that a gradient in *any* channel contributes to the edge strength.

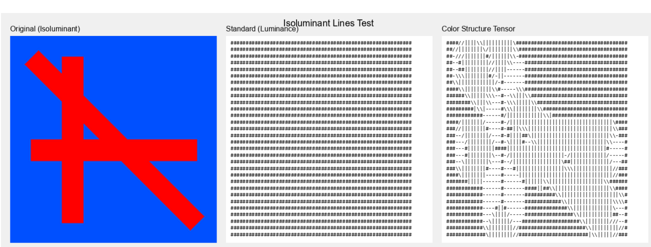

Fig. 2. Isoluminant Lines Test. (Left) Original image with red lines on a blue-teal background of equal luminance ($L \approx 76$). (Middle) Standard brightness method sees a uniform gray field. (Right) Color Structure Tensor correctly detects the boundaries between the colors, rendering the specific line orientations.

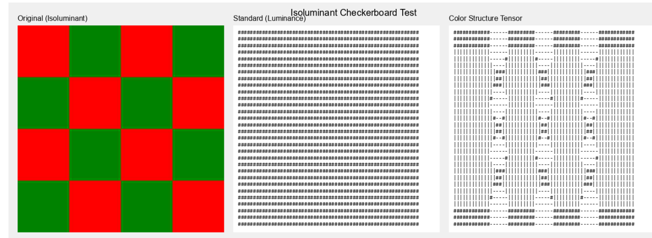

Fig. 3. Isoluminant Checkerboard Test. (Left) Original Red/Green checkerboard ($L \approx 76$). (Middle) Standard method renders a blank gray region. (Right) Color Structure Tensor successfully recovers the grid structure from the chromatic gradients.

Figure 2 and Figure 3 demonstrate this on two specific "isoluminant" stress tests. In both cases, the geometries (lines and checkerboard squares) are defined solely by chromatic contrast, with zero luminance difference. The standard method yields a featureless gray output. The Color Structure Tensor, however, integrates the gradients from all three color channels, effectively "seeing" the hidden structure and rendering it with appropriate edge characters. This demonstrates the method's ability to distinguish structures based purely on chromatic gradients.

4) Structural Similarity and Limitations: While the Structure Tensor method improves edge continuity, it imposes a stylized "sketch-like" aesthetic that may not always be desirable. For pure gradient visualization, the standard method may be more faithful to the original luminance. However, for structural recognition—identifying shapes, boundaries, and objects—the edge-enhanced method provides a superior signal-to-noise ratio. The discrete nature of the ASCII character set means no curve is truly smooth, but the use of orientation-matched characters minimizes the error between the character's principal axis and the image's local tangent.

E. Observations

The edge-enhanced method provides:

- Improved diagonal line representation at all resolutions
- Better preservation of geometric shapes and structural features
- Maintained brightness gradation in non-edge regions
- Consistent edge orientation accuracy across resolutions

Limitations:

- Jaggedness on Curves: Smooth curves may appear "jagged" because they are approximated by a limited set of linear characters (/, \, |, −).
- Computational overhead: The eigenvalue computation is approximately 3x slower than simple brightness mapping.
- Limited Testing Scope: The isoluminant differencing has mostly been verified on simple geometric shapes with 2-color patterns (e.g., Red/Green checkerboards). Its performance on complex multi-colored scenes with more than two overlapping isoluminant hues remains an open area for validation.

- Sensitivity to Noise and Irregular Shapes: The current Structure Tensor implementation assumes locally linear edges. For ”wiggly” lines or highly textured noisy images, the tensor may fail to identify a dominant direction, resulting in chaotic character selection.
- Loss of Fine Detail: In very high-frequency regions (details smaller than the character grid), the structural approach may over-simplify the texture, whereas brightness mapping might serendipitously capture the ”noise” better.

V. Discussion

A. Relationship to Corner Detection

The Structure Tensor analysis used here is closely related to the Harris corner detector [4]. In Harris's formulation, corners are detected where both eigenvalues are large. Our method focuses on the complementary case where $\lambda_1 \gg \lambda_2$, indicating edges rather than corners.

B. Computational Considerations

For real-time applications, the eigenvalue computation can be optimized:

1) Pre-compute and cache gradient arrays
2) Use separable Sobel filters for efficiency
3) Skip eigenvalue computation for pixels with low gradient magnitude
4) Vectorize operations using NumPy array operations

C. Extensions

Possible extensions include:

- Multi-scale analysis using image pyramids
- Adaptive thresholding based on local statistics
- Color extension using structure tensors on each channel
- Integration with machine learning for character selection

VI. Conclusion

This paper presented an edge-aware approach to ASCII art generation based on eigenvalue analysis of the Structure Tensor, with an extension to color through the Color Structure Tensor. By moving beyond pure brightness mapping, the proposed method incorporates local geometric information—specifically edge strength and orientation—into the character selection process.

The experimental results show that this approach significantly improves structural clarity at low and medium resolutions, where conventional luminance-based methods tend to break down. Directional characters aligned with the dominant eigenvector preserve edge continuity, reduce staircase artifacts, and improve the perceptual coherence of lines and shapes. The color-aware formulation further addresses a fundamental limitation of grayscale conversion by recovering edges defined purely by chromatic contrast, as demonstrated in the isoluminant test cases.

At higher resolutions, the advantage of structural analysis becomes less pronounced, as brightness-based methods already provide sufficient detail. This highlights an important trade-off: the proposed method is most beneficial in representations where resolution is constrained and structural information must be conveyed efficiently. The additional computational cost introduced by tensor construction and eigenvalue computation is therefore justified primarily in low-resolution or structure-critical scenarios.

Overall, this work demonstrates that even in a highly constrained medium such as ASCII art, explicitly modeling the mathematical structure of an image leads to perceptually stronger results. More broadly, it illustrates how concepts from linear algebra—eigenvalues, eigenvectors, and matrix representations—can be applied to non-traditional visual problems, bridging the gap between mathematical theory and creative digital applications.

Acknowledgment

I would like to thank the teaching team of IF2123 Aljabar Linier dan Geometri at Institut Teknologi Bandung. Their instruction on vector spaces and matrix decomposition provided the essential tools used to build this system.

References

[1] S. Xu, H. Zhang, and J. Liu, “Ascii art synthesis from natural images,” in Proc. ACM SIGGRAPH, 2010.
[2] D. C. Lay, S. R. Lay, and J. J. McDonald, Linear Algebra and Its Applications, 5th ed. Boston, MA: Pearson, 2016.
[3] G. Strang, Introduction to Linear Algebra, 5th ed. Wellesley, MA: Wellesley-Cambridge Press, 2016.
[4] C. Harris and M. Stephens, “A combined corner and edge detector,” in Proc. 4th Alvey Vision Conf., Manchester, UK, 1988, pp. 147–151.
[5] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 4th ed. New York: Pearson, 2018.
[6] I. Sobel, “An isotropic 3x3 image gradient operator,” presentation at Stanford Artificial Intelligence Project, 1968.
[7] J. Weickert, “Coherence-enhancing diffusion filtering,” Int. J. Comput. Vis., vol. 31, no. 2/3, pp. 111–127, 1999.
[8] T. Brox and J. Weickert, “A TV flow based local scale measure for texture discrimination,” in Proc. ECCV, 2004, pp. 578–590.

Appendix

The complete source code and implementation of this project can be accessed at the following GitHub repository:

`https://github.com/tmthyberd/makalah-algeo`

Statement of Originality

I hereby declare that this paper is my own writing, not an adaptation or translation of others' work, and does not contain plagiarism.

Bandung, December 24, 2025

Timothy Bernard Soeharto
13524092