Wooyoung Kim
323-834-4126
wkim9450@usc.edu

EE 569 Homework #2
February 22, 2026

Problem 1: Edge Detection

I. Abstract and Motivation

Edge detection is a fundamental computer vision task aimed at identifying object boundaries and structural transitions within an image. The motivation behind evaluating multiple algorithms (Sobel, Canny, and Structured Edge (SE)) to understand the trade-offs between computational simplicity, gradient-based localization, and semantic accuracy. This report applies these detectors to the BSDS 500 test images (*Bird* and *Deer*), tuning their respective parameters to optimize output. It evaluates their performance against human-generated ground truths using quantitative metrics such as Precision, Recall, and the F-measure to objectively measure boundary detection results.

II. Approach and Procedures

II..1 Sobel Edge Detector

The Sobel operator estimates the image gradient by convolving the grayscale image with two 3×3 kernels to approximate horizontal (G_x) and vertical (G_y) derivatives. RGB images are converted to grayscale using the formula $Y = 0.2989 * R + 0.5870 * G + 0.1140 * B$. The normalized gradient magnitude map is calculated using $M = \sqrt{G_x^2 + G_y^2}$. I normalize the gradients to a 0–255 scale. Finally, a threshold is applied to binarize the magnitude map into edges (0) and background (255).

II..2 Canny Edge Detector

The Canny edge detector improves upon basic gradient methods through two steps:

1. **Non-maximum Suppression:** This thinning technique ensures edges are exactly one pixel wide. The algorithm examines the gradient direction for each pixel. If the pixel's magnitude is not the strict local maximum along that exact gradient direction, its value is suppressed.

2. **Double Thresholding:** Two threshold values are used in a hysteresis process. Pixels above the high threshold are classified as strong edges. Pixels below the low threshold are discarded as noise. Pixels falling between the two thresholds are weak edges and are only preserved if they connect to a strong edge pixel.

II..3 Structured Edge

1. **Algorithm and Flowchart:** The SE algorithm extracts overlapping local patches from the image and computes multi-channel features (color, gradient magnitude, and orientation). A Random Forest classifier evaluates these features to predict a local, structured edge mask for the patch. Predictions from overlapping patches are averaged to produce a final probability map. Then, the map will be binarized.

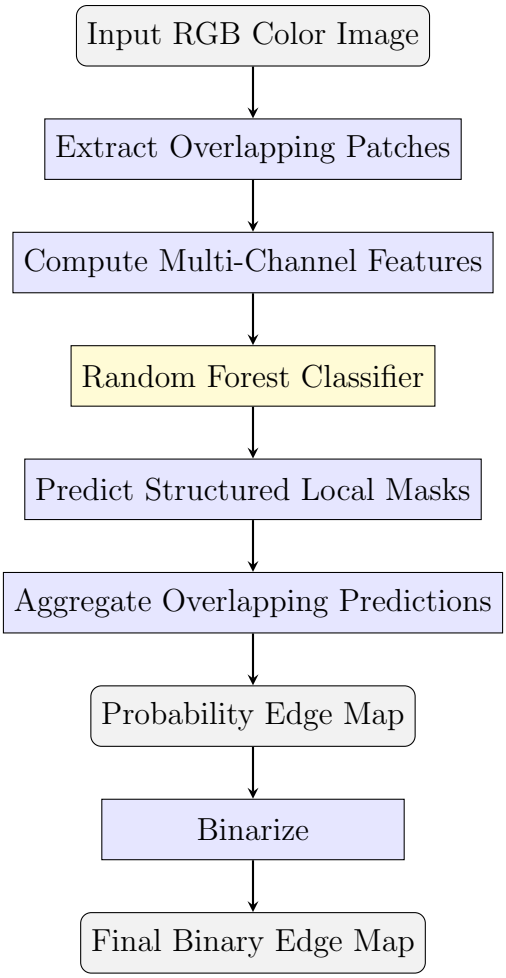


Figure 1: Flowchart of the Structured Edge algorithm.

2. **Random Forest Principle:** A decision tree recursively splits training data at nodes based on feature thresholds to maximize information gain. The Random Forest classifier utilizes an ensemble of these decision trees, where each tree is trained on a random subset of data and features. The RF integrates their outputs by averaging the individual predictions into a probability vector, which prevents overfitting.

II..4 Performance Evaluation

Performance is evaluated by comparing generated binary edge maps to human ground truths (GT). Precision ($P = \frac{TP}{TP+FP}$) measures edge accuracy, while Recall ($R = \frac{TP}{TP+FN}$)

measures edge completeness. The F-measure is the harmonic mean: $F = 2 \cdot \frac{P \cdot R}{P+R}$. I compute mean P and R across all thresholds and plot the F-measure curve.

III. Experimental Results

III..1 Sobel Edge Detector

The optimal threshold magnitude for Sobel edge detection was 15 when the gradient magnitude was normalized to the 0–255 range. At 15, weak texture responses are suppressed while meaningful object boundaries are preserved. Therefore, this threshold provides a good balance between noise reduction and edge preservation in natural images.

Figure 2 displays the final Bird images by threshold.

Figure 2: Sobel Results by Threshold.

Figure 3 displays the final Deer images by threshold.

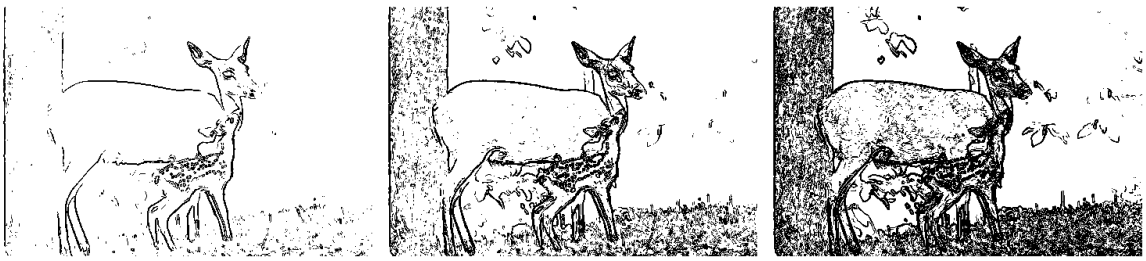


Figure 3: Sobel Results by Threshold.

Figure 5 displays the normalized x-gradient, y-gradient, magnitude map, and the final binarized output for the Bird image. The optimal visual threshold was found at the top 15% of magnitude values.

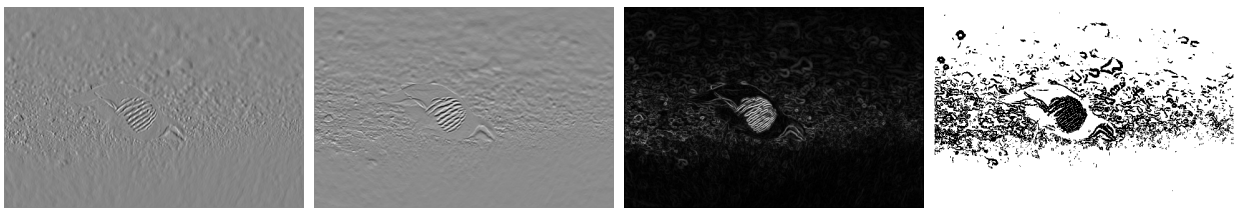


Figure 4: Sobel Results (Left to Right): Normalized X-Gradient, Y-Gradient, Magnitude Map, and Binary Edge Map (Threshold = 15%).

Figure 5 displays the normalized x-gradient, y-gradient, magnitude map, and the final binarized output for the Deer image. The optimal visual threshold was found at the top 15% of magnitude values.

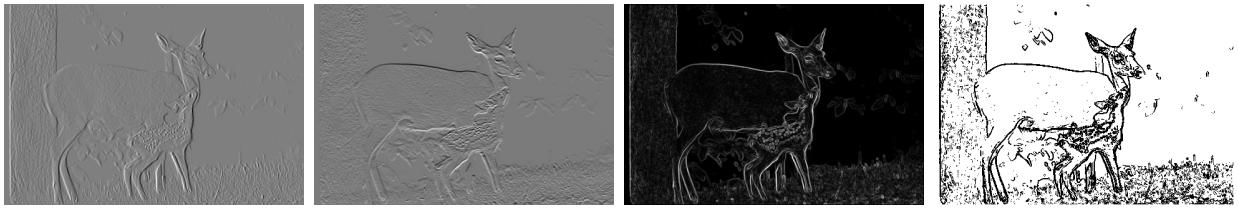


Figure 5: Sobel Results (Left to Right): Normalized X-Gradient, Y-Gradient, Magnitude Map, and Binary Edge Map (Threshold = 15%).

III..2 Canny Edge Detector

Figure 6 and Figure 7 demonstrates the effect of varying the thresholds.

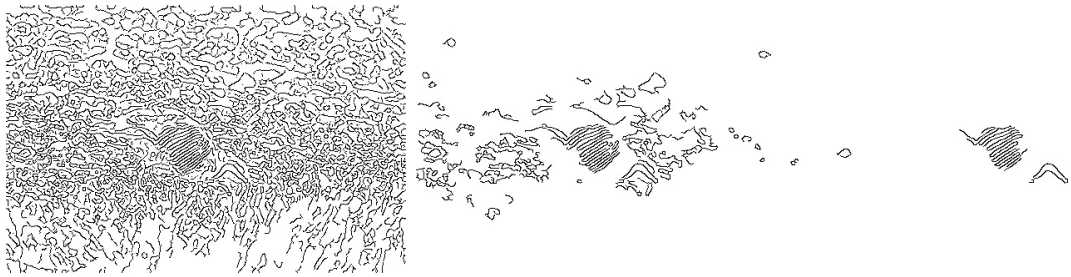


Figure 6: Bird Canny Edge Maps. Left: Low Thresholds (10/30). Middle: Optimal Thresholds (60/180). Right: High Thresholds (120/360).

Figure 7: Deer Canny Edge Maps. Left: Low Thresholds (10/30). Middle: Optimal Thresholds (60/180). Right: High Thresholds (120/360).

III..3 Structured Edge

The SE detector was applied to the *Bird.raw* and *Deer.raw* images using OpenCV. To accommodate the raw output distribution of the OpenCV port, the probability maps were Min-Max normalized to $[0.0, 1.0]$ scale, and Non-Maximum Suppression (NMS) was applied to achieve one-pixel-wide edges.

Through tuning, the optimal thresholds were found to be image-dependent. For the *Bird* image, the best visual performance was achieved at a threshold of $t = 0.30$. For the *Deer* image, the optimal threshold was significantly lower, yielding the best results at $t = 0.15$.

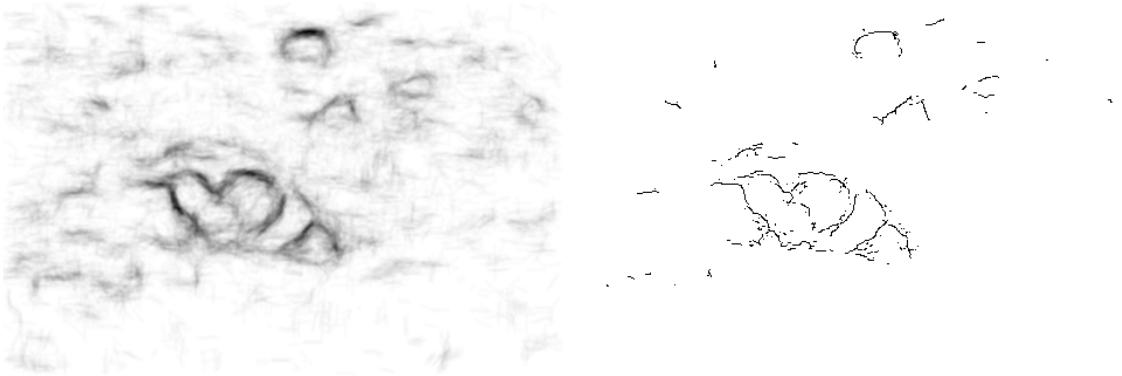


Figure 8: SE Results for Bird. Left: Normalized Probability Map. Right: Binary Edge Map ($t = 0.30$).

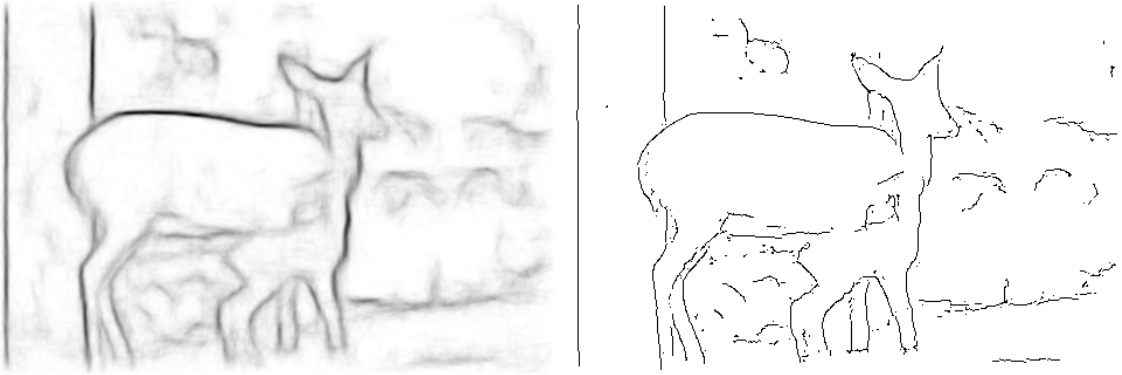


Figure 9: SE Results for Deer. Left: Normalized Probability Map. Right: Binary Edge Map ($t = 0.15$).

III..4 Performance Evaluation

The evaluation was performed using a strict pixel-wise matching algorithm against the provided Ground Truth annotations.

Table 2 presents the Precision (P), Recall (R), and F-measure (F) evaluated individually against each of the five ground truths, along with the overall means and the final calculated F-measure for the *Bird* and *Deer* images at the optimal threshold for each detector.

	Sobel			Canny (Optimal)			SE		
GT Index	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
GT 1	0.0408	0.5309	0.0757	0.0461	0.1355	0.0688	0.0786	0.3633	0.1292
GT 2	0.0431	0.4948	0.0793	0.0432	0.1119	0.0624	0.0782	0.3185	0.1255
GT 3	0.0758	0.5852	0.1342	0.0983	0.1714	0.1249	0.1631	0.4471	0.2390
GT 4	0.0621	0.5178	0.1109	0.0727	0.1367	0.0949	0.1330	0.3937	0.1989
GT 5	0.0304	0.6090	0.0579	0.0551	0.2491	0.0902	0.0901	0.6410	0.1580
Overall Mean	**0.0504**	**0.5475**	**0.0924**	**0.0631**	**0.1609**	**0.0906**	**0.1086**	**0.4327**	**0.1736**

Table 1: Performance evaluation of Sobel, Canny, and SE detectors on the Bird image against 5 Ground Truths.

	Sobel			Canny (Optimal)			SE		
GT Index	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
GT 1	0.1376	0.6038	0.2241	0.2236	0.2657	0.2429	0.3279	0.4319	0.3728
GT 2	0.1154	0.7634	0.2004	0.1989	0.3563	0.2553	0.2777	0.5514	0.3693
GT 3	0.1548	0.7547	0.2569	0.2777	0.3665	0.3160	0.3968	0.5806	0.4714
GT 4	0.2079	0.6781	0.3182	0.3264	0.2881	0.3061	0.4172	0.4084	0.4128
GT 5	0.2023	0.5551	0.2966	0.3366	0.2499	0.2869	0.4198	0.3457	0.3791
Overall Mean	**0.1636**	**0.6710**	**0.2630**	**0.2726**	**0.3053**	**0.2881**	**0.3679**	**0.4636**	**0.4102**

Table 2: Performance evaluation of Sobel, Canny, and SE detectors on the Deer image against 5 Ground Truths.

Furthermore, the overall F-measure was computed across a full sweep of thresholds from 0.01 to 0.99 for the continuous probability and magnitude maps. Figure 10 and Figure 11 illustrates these F-measure vs. Threshold curves, pinpointing the maximum F-measure achieved by each respective edge detector.

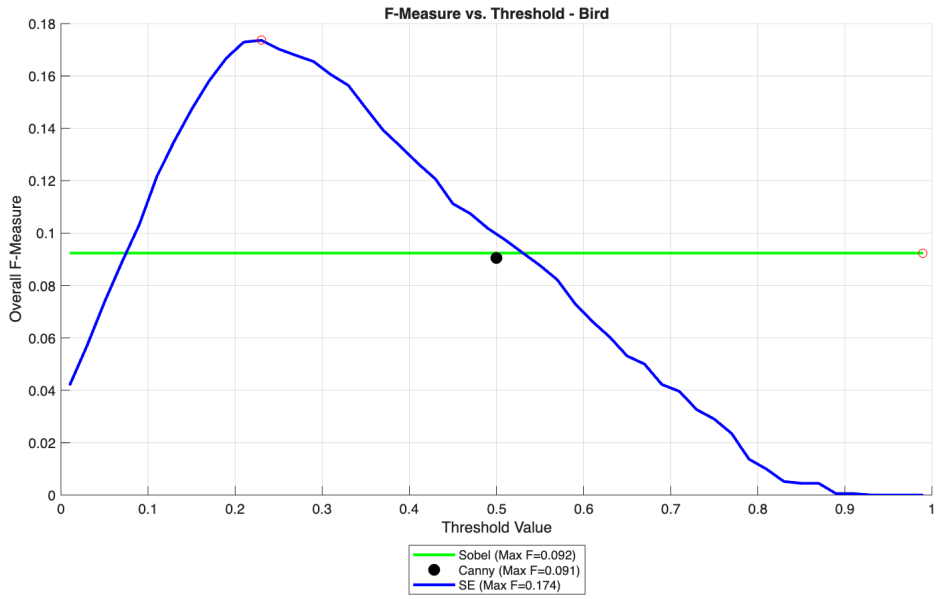


Figure 10: F-measure vs. Threshold curves for the Sobel, Canny, and SE detectors on the Bird image.

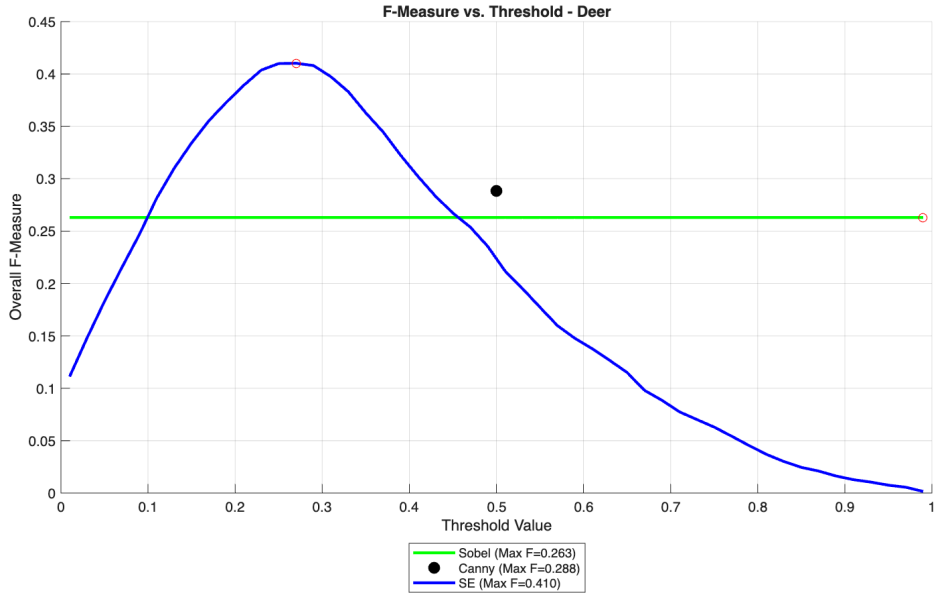


Figure 11: F-measure vs. Threshold curves for the Sobel, Canny, and SE detectors on the Deer image.

IV. Discussion

IV..1 Sobel Edge Detector

The Sobel operator is computationally efficient but highly sensitive to noise. The lack of non-maximum suppression results in thick, imprecise boundaries. While tuning the percentage threshold helps isolate stronger edges, it inevitably loses boundaries, leading to poor performance.

IV..2 Canny Edge Detector

Non-maximum suppression proves critical, allowing Canny to produce sharp, one-pixel-wide lines. Double thresholding effectively bridges gaps in edge contours. However, Canny relies entirely on local gradients. it frequently mistakes high contrast background textures (ex. grass) for object boundaries, negatively impacting Precision. It outperforms Sobel but falls short of SE.

IV..3 Structured Edge

The SE detector demonstrates superior visual performance compared to both Sobel and Canny, successfully utilizing its learned Random Forest classifier to suppress high-gradient background noise and focus on holistic object contours.

A key observation from this experiment is the discrepancy between the example threshold ($p > 0.8$) suggested in the assignment for the MATLAB implementation and the practical thresholds acquired for the OpenCV. Even after normalization, the optimal thresholds were much lower ($t = 0.30$ and $t = 0.15$).

Furthermore, these results highlight the image-dependent nature of edge detection thresholds. The *Bird* image features a highly textured, pebbly background that generates significant gradient noise. To suppress these false positives and maintain a reasonable Precision score, a stricter threshold of 0.3 was required. Conversely, the *Deer* image features brightly lit subjects against a deeply shadowed, nearly uniform dark background. Because the dark shadow generates almost no false positives, the algorithm permits a much more lenient threshold of 0.15.

IV..4 Performance Evaluation

Performance of Different Edge Detectors (Pros and Cons): Based on the overall result including Bird and Deer images, the Sobel detector yielded the lowest F-measure. Its primary pro is mathematical and computational simplicity, but its con is extreme noise sensitivity and thick, non-localized edges. The Canny detector achieved a moderate F-measure than Sobel detector. Its pro is the generation of thin, continuous lines via non-maximum suppression, but its con is an inability to distinguish true structural boundaries from background textures. The Structured Edge detector achieved the highest F-measure. Its pro is robust, semantic edge detection learned from human annotations, which drastically reduces false positives, while its con is high computational complexity.

F-measure vs. Threshold Curves: As illustrated in Figure 10 and Figure 11, the F-measure changes with the threshold. At low thresholds, the algorithms generate excessive False Positives. At high thresholds, they miss true edges, generating False Negatives. The Sobel curve is very flat and low, reflecting poor performance. Canny exists as a single highly-optimized point due to its binary nature. The SE detector curve is significantly higher and broader, indicating it achieves the highest absolute maximum F-measure.

Image Dependence (Bird vs Deer): The F-measure is highly image-dependent. Based on visual characteristics, the Deer image is significantly easier to achieve a high F-measure on. The Deer image features brightly lit subjects against a massive, deeply shadowed, almost uniform dark background. Because solid black areas generate zero gradients, there is no risk of background noise, allowing the detector to utilize a lenient threshold to maximize Recall without introducing False Positives. In contrast, the Bird image is positioned on a highly textured ground. This high frequency background generates thousands of weak false edges. This forces a harsh trade-off: lowering the threshold to capture the bird's full outline simultaneously captures the pebbles, heavily penalizing the Precision score.

Rationale Behind the F-measure Definition: The F-measure is defined as the harmonic mean of Precision and Recall. It is impossible to achieve a high F-measure if either metric is significantly higher than the other (ex. $P = 0.99, R = 0.10$), because the harmonic mean is heavily weighted toward the lowest value. Both must be simultaneously high, ensuring the algorithm neither misses true edges nor invents fake ones.

Mathematical Proof ($P = R$ maximizes F): Given that the sum of Precision and Recall is a constant C ($P + R = C$), I substitute $R = C - P$ into the standard F-measure formula:

$$F = \frac{2P \cdot (C - P)}{P + (C - P)} = \frac{2CP - 2P^2}{C}$$

To find the maximum, we take the first derivative with respect to P and set it to zero:

$$\frac{dF}{dP} = \frac{d}{dP}\left(2P - \frac{2P^2}{C}\right) = 2 - \frac{4P}{C} = 0$$

$$2 = \frac{4P}{C} \implies 2C = 4P \implies P = \frac{C}{2}$$

Substituting P back returns $R = C - \frac{C}{2} = \frac{C}{2}$. Therefore, the F-measure reaches its absolute maximum when Precision exactly equals Recall.

Problem 2: Digital Half-toning

I. Abstract and Motivation

The objective is to convert an 8-bit continuous-tone grayscale image into a binary halftone image. Digital halftoning is essential for displaying or printing images on devices that only

support binary outputs. This section explores two primary categories of halftoning algorithms: Dithering and Error Diffusion. By implementing and comparing these methods, I can evaluate their effectiveness in preserving image details, smooth gradients, and overall visual quality while minimizing noticeable artifacts.

II. Approach and Procedures

II..1 Dithering

Dithering algorithms determine pixel values by comparing the original grayscale intensity against a threshold. Three variations were implemented:

1. **Fixed Thresholding:** A constant threshold $T = 128$ is applied globally. Pixels < 128 become 0 (black), and others become 255 (white).

2. **Random Thresholding:** A uniformly distributed random threshold between 0 and 255 is generated for each pixel to break up the harsh monotones of fixed threshold.

3. **Dithering Matrix:** Bayer index matrices (I_n) are recursively generated and normalized into threshold matrices $T(x, y)$. The matrix is periodically tiled across the image, and each pixel is compared to its corresponding threshold. Matrices of sizes 2×2 (I_2), 8×8 (I_8), and 32×32 (I_{32}) were applied.

II..2 Error Diffusion

Unlike dithering, error diffusion incorporates spatial neighborhood information. When a pixel is quantized to 0 or 255, the resulting quantization error is distributed to adjacent, unprocessed pixels using a specific kernel map. Three kernels were tested:

1. **Floyd-Steinberg:** Utilizes a 2×3 kernel with a divisor of 16. This was implemented with serpentine scanning to alternate the direction of error propagation line by line.

2. **Jarvis, Judice, and Ninke (JJN):** Utilizes a larger 3×5 kernel with a divisor of 48, applying standard raster scanning.

3. **Stucki:** Utilizes a 3×5 kernel with a divisor of 42, applying standard raster scanning.

III. Experimental Results

III..1 Dithering

Fixed thresholding results in a high-contrast image with a severe loss of detail in shadowed and highlighted areas. Random thresholding introduces a heavy white-noise effect, restoring some gradient perception but resulting in a highly grainy image. The ordered dithering matrices successfully introduce structured patterns that simulate gradients much more effectively than fixed or random methods.

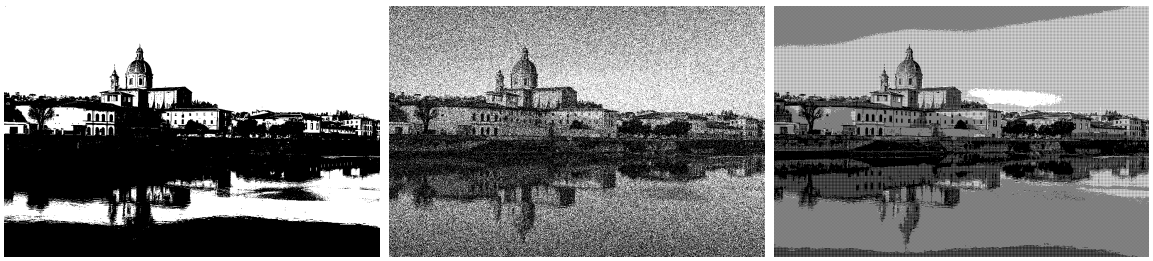
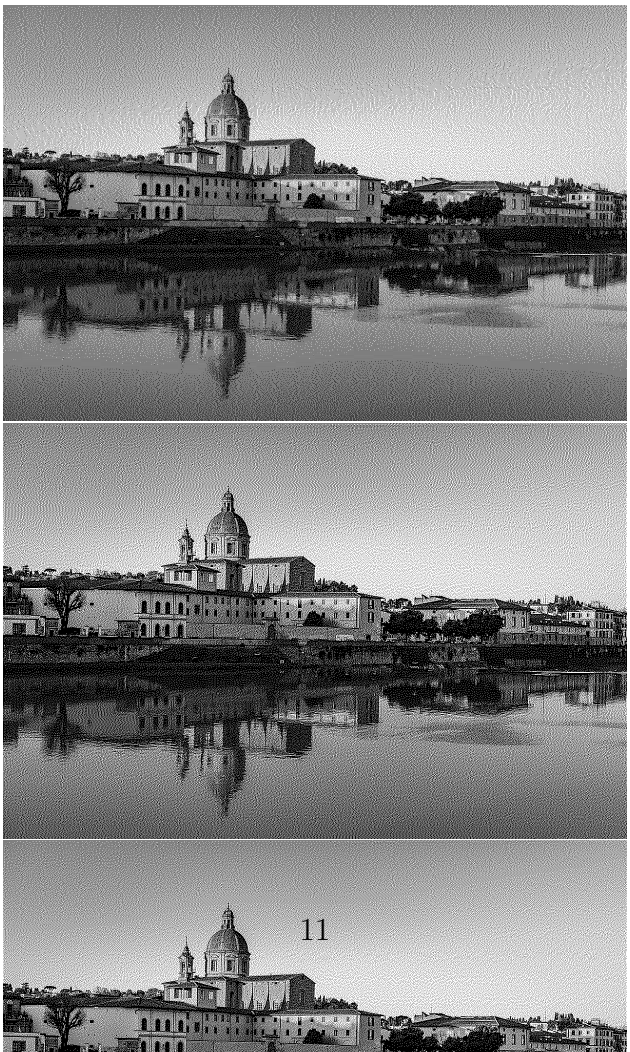


Figure 12: Left to Right: Fixed Threshold, Random Threshold, Dithering Matrix I_2

Figure 13: Left to Right: Dithering Matrix I_8, Dithering Matrix I_{32}

III..2 Error Diffusion



The error diffusion methods produce significantly sharper images compared to ordered dithering. The gradients are smoother, and the structural edges of the buildings in the Reflection image are highly preserved.

IV. Discussion

IV..1 Dithering

When comparing the results obtained by the different dithering matrices (I_2, I_8, I_{32}), the differences lie in the number of simulated gray levels and the spatial frequency of the resulting patterns.

- I_2 **Matrix:** Produces a high-frequency checkerboard pattern. Because it only supports 5 effective intensity levels, smooth transitions in the sky and water reflections suffer from severe contouring.
- I_8 **Matrix:** Offers 65 intensity levels. It provides a much smoother gradient representation of the sky while maintaining a relatively tight, high-frequency dot pattern that is less distracting to the human eye.
- I_{32} **Matrix:** Supports 1025 effective levels, offering the most accurate mathematical reproduction of continuous gradients. However, because the matrix is physically larger, the structured cross-hatching becomes much more apparent. This can introduce visible texture artifacts that distract from the underlying image details.

IV..2 Error Diffusion

Among the tested error diffusion algorithms, the Stucki method is the preferred choice.

While Floyd-Steinberg is computationally cheaper, it is prone to distinct, structured artifacts in mid-tone areas. The JJN kernel, being larger, diffuses the error over a wider area, creating a softer but slightly blurred image. The Stucki method is at the best balance. It uses a large kernel to disperse artifacts effectively, but its specific weighting coefficients tend to produce sharper edges and higher local contrast.

Idea for Better Results: *Space-Filling Curve Error Diffusion (Hilbert Curve).*

Standard error diffusion algorithms push quantization errors in rigid and uniform directions. This directional propagation is the root cause of the repeating artifacts in uniform areas. Traversing the image array using a space-filling curve like the Hilbert curve distributes the quantization error in a much more directionless manner. Because the curve twists and turns in 2D space, the quantization error is dispersed across all adjacent pixels in varying directions, effectively breaking up linear artifacts and yielding smoother halftones.

Problem 3: Color Half-toning with Error Diffusion

I. Abstract and Motivation

Color halftoning is a crucial process for rendering continuous tone images with a limited color palette, such as printers. The challenge is to create the illusion of continuous color gradients using only a small set of printable colors. This section explores two error diffusion techniques for color halftoning: Separable Error Diffusion method and Minimum Brightness Variation Quadrants (MBVQ) based Error Diffusion method. The motivation is to demonstrate how addressing color channel correlation reduces visual artifacts and luminance noise compared to independent channel processing.

II. Approach and Procedures

II..1 Separable Error Diffusion

The separable approach treats color halftoning as three independent grayscale halftoning tasks. The procedure is as follows:

1. The 24-bit RGB input image is converted into the CMY color space by subtracting the RGB values from 255.
2. The Floyd-Steinberg error diffusion algorithm is applied to each of the C, M, and Y channels separately.
3. For each channel at a given pixel, the continuous value is quantized to either 0 or 255 (a threshold of 128).
4. The quantization error is distributed to adjacent, unprocessed pixels using the standard Floyd-Steinberg weights.
5. The resulting binary CMY channels form one of eight possible colors at each pixel, corresponding to the vertices of the CMY cube.

II..2 MBVQ-based Error diffusion

The MBVQ-based method processes the color of a pixel as a single point in a 3D RGB color space rather than independent channels. The procedure is as follows:

1. The RGB color cube is partitioned into Minimum Brightness Variation Quadrants (MBVQs).
2. For a given pixel, the algorithm evaluates the original RGB values to determine which of the six MBVQ tetrahedrons the pixel belongs to.
3. The accumulated quantization error is added to the pixel's original RGB value.
4. The algorithm calculates the Euclidean distance to the four vertices of the specifically assigned MBVQ tetrahedron and outputs the closest vertex.
5. The 3D vector quantization error is calculated and distributed to neighboring pixels using the Floyd-Steinberg weights.

III. Experimental Results

III..1 Separable Error Diffusion

Figure 15: Halftoned Flowers image using Separable Error Diffusion.

The separable error diffusion algorithm successfully generated a halftoned representation of the Flowers image, utilizing only the eight primary and secondary colors. However, the result exhibit noticeable noise and dot patterns.

III..2 MBVQ-based Error diffusion

Figure 16: Halftoned Flowers image using MBVQ-based Error Diffusion.

Applying the MBVQ-based error diffusion to the same Flowers image resulted in a cleaner output. The transition between colors is smoother and the overall visual fidelity closely mimics the original continuous-tone image.

IV. Discussion

IV..1 Separable Error Diffusion

The shortcoming of the separable error diffusion approach is its disregard for the correlation between color channels and the overall luminance of the pixel. Because the C, M, and Y channels fire dots independently, complementary colors are often placed adjacently or overlapping in an uncoordinated manner. This produces salt and pepper artifacts. Furthermore, because the separable method does not control for local luminance, the independent channel quantizations cause the local brightness to fluctuate wildly, resulting in a noisy texture.

IV..2 MBVQ-based Error diffusion

The MBVQ method overcomes the shortcomings of the separable approach by prioritizing local luminance consistency. The key idea establishing this method is that human eyes are more sensitive to spatial variations in brightness than in color. By restricting the quantization targets for any given pixel to the four vertices of its specific MBVQ tetrahedron, this method ensures that the selected colors have the minimum possible brightness difference between them. This prevents harsh complementary colors from being placed in the same local neighborhood if a smoother mid-tone is required. Consequently, the MBVQ method eliminates the aggressive salt-and-pepper noise and preserves smooth gradients.