Indian Journal of Computer Science and Technology
Volume 5, Issue1 (January-April 2026), PP: 73-77.
www.indjcst.com **ISSN No: 2583-5300**

Enhancements and Optimization of the Canny Edge Detection Algorithm

Shantanu Bahadur Karki
Master's Student, Software Engineering, Nepal.

To Cite this Article: *Shantanu Bahadur Karki, "Enhancements and Optimization of the Canny Edge Detection Algorithm", Indian Journal of Computer Science and Technology, Volume 05, Issue 01 (January-April 2026), PP: 73-77.*

Abstract: *The canny edge detection algorithm is a trusted and robust algorithm used to identify edges in the image. The algorithm processes a greyscale image to detect the edges in the image by processing it through multiple steps. In this paper, we shall discuss the steps and what we can do to make those steps better. Canny edge detection is useful for object detection, gesture recognition, and product quality control.*

Key Words: *Image Processing, Gradient Calculation, Non-Maximum Suppression, Algorithm Optimization.*

I. INTRODUCTION

Edge detection is a fundamental task in image processing and computer vision, serving as the basis for applications such as object recognition, gesture analysis, and industrial quality inspection. Among the various algorithms developed, the Canny edge detection algorithm, proposed by John F. Canny in 1986 [1], remains one of the most widely used due to its robustness and accuracy.

The Canny algorithm follows a multi-stage process designed to maximize edge detection while minimizing noise and false detections. First, the input image is converted to grayscale, reducing computational complexity by focusing solely on intensity variations. A Gaussian filter is then applied to suppress noise while preserving important edge structures. The gradient magnitude and direction are computed, typically using the Sobel operator, to highlight regions of rapid intensity change. To refine the detected edges, non-maximum suppression is applied, which thins edges to a single-pixel width along the gradient direction. Finally, a double-thresholding and hysteresis process classifies edges as strong, weak, or irrelevant, thereby producing a clean and continuous edge map.

Despite its effectiveness, the classical canny detector exhibits limitations when applied to noisy or complex images. Its performance is highly sensitive to parameter selection and may fail to preserve fine structures under certain conditions. This motivates research into optimizing individual steps of the algorithm, such as noise reduction, gradient estimation, and edge refinement, to improve accuracy and robustness in real-world applications.

II.METHODOLOGIES

The proposed approach builds upon the classical canny edge detection framework by introducing enhancements in three key stages: noise reduction, gradient calculation, and edge refinement. The overall process consists of six sequential steps: grayscale conversion, noise reduction, gradient computation, non-maximum suppression, double thresholding, and edge tracking by hysteresis. Each step is described in detail below.

A. Image Grey Scaling

Since most input images are in RGB format, the first step converts the image into a single-channel grayscale representation. This reduces computational complexity while preserving essential luminance information. The luminosity method was used for this transformation, defined as:

$$.299*R + .587*G + .114*B$$

Where R, G, B denote the red, green, and blue color channels, respectively. This weighting reflects human visual sensitivity and provides a perceptually accurate grayscale image as seen in Fig. 1.

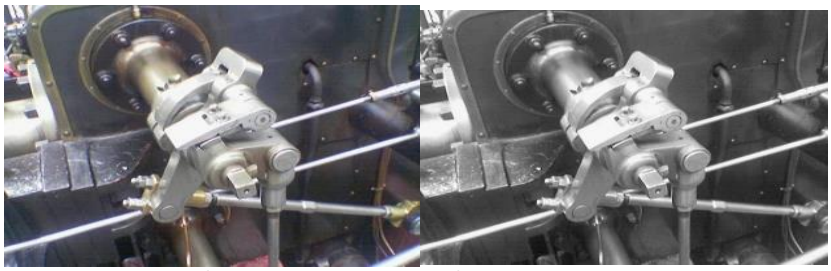

Fig. 1 Original Image of a mechanical valve on left and the grey scaled on the right (Source)

B. Noise Reduction

Noise reduction is critical for accurate edge detection, as spurious noise can result in false or broken edges. While the classical canny algorithm relies on Gaussian smoothing, this study evaluates three alternative denoising techniques:

- **Gaussian Filtering:** Effective for reducing high-frequency noise but prone to blurring fine edges.
- **Morphological Filtering:** Useful for structural preservation but less effective in homogeneous regions.
- **Non-Local Means (NL-Means) Denoising:** Demonstrated superior performance by preserving edge details while effectively suppressing noise.

Experimental results, as seen in Fig. 2-4, confirm that NL-Means denoising produced the sharpest and most consistent edges among the tested methods.

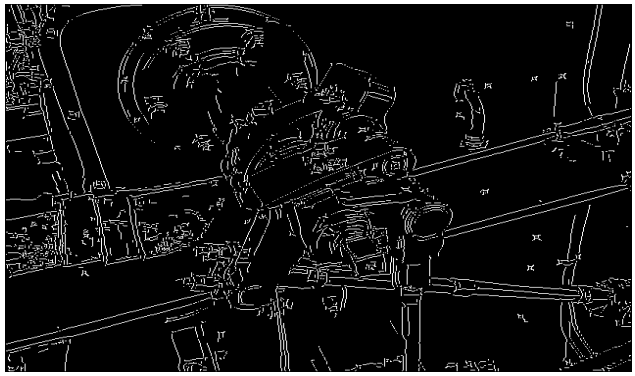

Fig. 2 The final edges image with NL-means filtering with sigma_h as 5, patch size as 7, neighborhood size as 23 and low and high threshold as .05 and .09 respectively

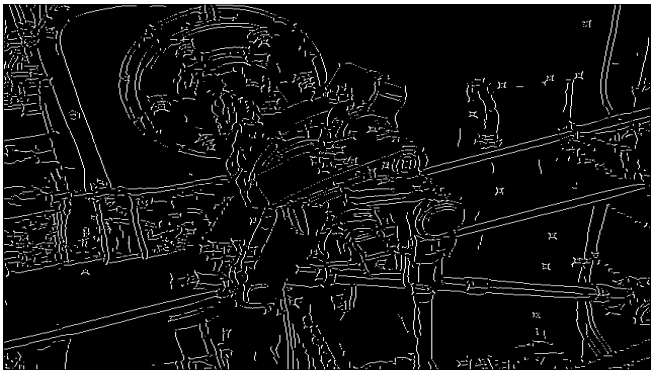

Fig. 3 The final edges image with Gaussian filtering with sigma as 2, kernel size as 5 and low and high threshold as .05 and .09 respectively

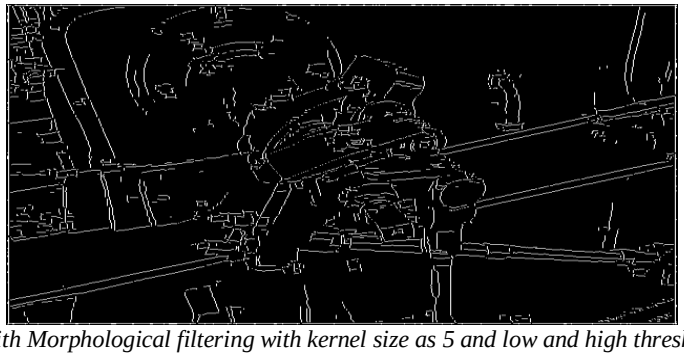

Fig. 4 The final edges image with Morphological filtering with kernel size as 5 and low and high threshold as .05 and .09 respectively

C. Gradient Calculation

Gradient magnitude and direction are essential for identifying edge strength and orientation. The classical canny detector employs the Sobel operator; however, this study compared three operators:

- **Sobel Operator:** Widely used for its simplicity and balance between accuracy and efficiency.
- **Prewitt Operator:** Like Sobel but less sensitive to diagonal edges.
- **Laplacian Operator:** Captures edges effectively but does not provide gradient direction, limiting its usability in later stages.

Based on experimental analysis, the Sobel operator was selected, results can be seen in Fig. 5. To further improve accuracy, a 5×5 convolution kernel was used instead of the standard 3×3 kernel, yielding more stable gradient estimates.

The gradient magnitude is given by:

$$\mathbf{Gx = I * Ix}$$
$$\mathbf{Gy = I * Iy}$$

Where **I** is the image, **Ix** and **Iy** are the 5x5 convolution matrix across the x and y axis, and * is the convolution operator.

$$\mathbf{G} = \sqrt{\boldsymbol{Gx^2 + Gy^2}}$$

Where **G** is the gradient edges, G_x and G_y represent the horizontal and vertical gradient components, respectively.

$$\boldsymbol{\theta} = \mathbf{tan^{-1}}\left(\frac{\mathbf{Gy}}{\mathbf{Gx}}\right)$$

Where **θ** is the gradient angle.

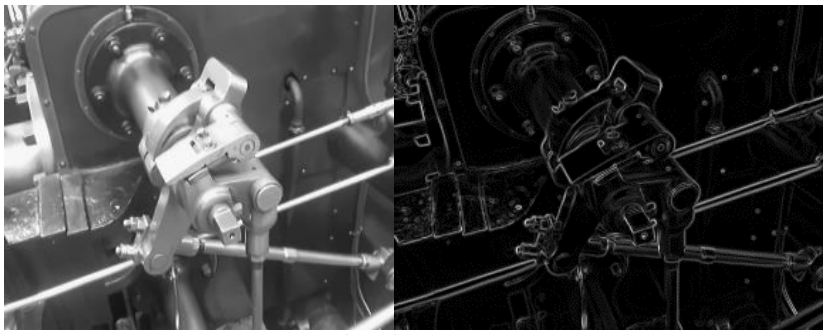

Fig. 5 Gradient edges of the filtered image with Sobel operator

D. Non-Max Suppression

To refine edge localization, non-maximum suppression (NMS) was applied. For each pixel, the algorithm compares its gradient magnitude with neighboring pixels along the gradient direction. Only pixels with the highest magnitude are retained, resulting in thin, single-pixel-wide edges, as seen in Fig. 6. This directional suppression reduces edge blurring and enhances continuity.

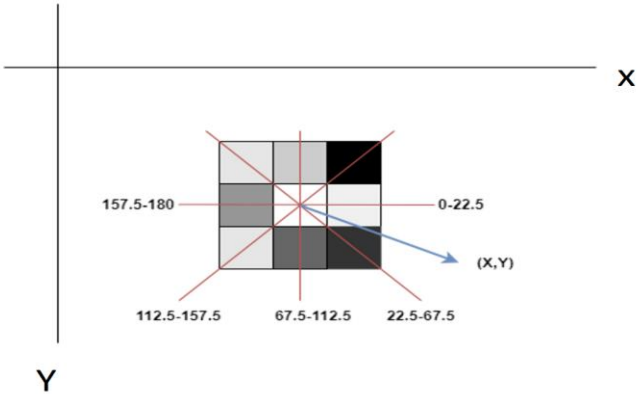

Fig. 6 Non-Max Suppression

- **Between (0-22.5) or (157.5-180):**

If the gradient angle of the pixel is between these angles, the neighboring pixels will be (x, y-1) and (x, y+1).

- **Between (22.5-67.5):**

If the gradient angle of the pixel is between these angles, the neighboring pixels will be (x+1, y-1) and (x-1, y+1).

- **Between (67.5-112.5):**

If the gradient angle of the pixel is between these angles, the neighboring pixels will be (x+1, y) and (x-1, y).

- **Between (112.5-157.5):**

If the gradient angle of the pixel is between these angles, the neighboring pixels will be (x-1, y-1) and (x+1, y+1).

E. Double Thresholding

The thinned edge map is subjected to double thresholding, which classifies pixels as strong, weak, or irrelevant based on intensity values. Pixels with gradient magnitude above the high threshold are retained as strong edges, while those below the low threshold are suppressed. Pixels falling between the two thresholds are marked as weak candidates.

F. Edge Tracking by Hysteresis

Finally, hysteresis is applied to connect weak edges to strong ones if they are spatially adjacent, as shown in Fig. 7. This ensures that true edges are preserved while isolated weak pixels caused by noise are eliminated. The result is a robust and continuous edge map suitable for higher-level vision tasks.

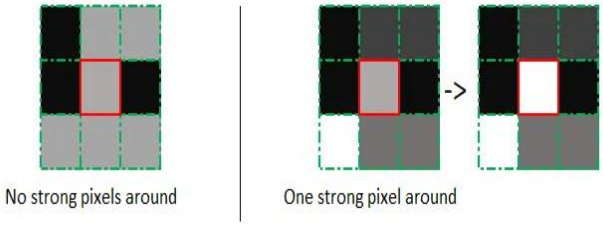

Fig. 7 Edge Tracking by Hysteresis

III.RESULTS

The proposed enhancements were evaluated on a set of grayscale and color images containing diverse textures and structural details. Figures 2–8 illustrate representative results obtained from different noise reduction and gradient calculation methods.

A. Effect of Noise Reduction Techniques

The effectiveness of Gaussian filtering, morphological filtering, and NL-Means denoising was compared. As shown in Fig. 2–4, NL-Means consistently produced sharper edges while reducing background noise more effectively than Gaussian or morphological filtering. Gaussian filtering tended to blur fine structures, whereas morphological filtering preserved some structural details but was less robust in homogeneous regions.

B. Gradient Calculation Operators

Gradient maps generated using Sobel, Prewitt, and Laplacian operators are compared in Fig. 5. The Sobel operator provided the most reliable balance between edge sharpness and directional accuracy. Although the Laplacian operator achieved stronger edge responses, the absence of orientation information limited its usefulness in subsequent non-maximum suppression. Prewitt was found to be less sensitive to diagonal edges, reducing overall detection accuracy.

C. Edge Refinement and Continuity

Figure 6 demonstrates the effect of non-maximum suppression in producing thin, continuous edges. The use of a 5×5 Sobel kernel further improved edge smoothness compared to the standard 3×3 implementation. Double thresholding and hysteresis (Fig. 7) effectively distinguished between strong and weak edges, reducing false positives caused by noise.

D. Final Results

Figure 8 presents final outputs for different test images, highlighting the superior performance of the proposed modifications. In particular, the combination of NL-Means denoising and 5×5 Sobel gradient computation produced well-defined and continuous edge maps. These results demonstrate the effectiveness of the optimized canny implementation in preserving fine details while suppressing noise.

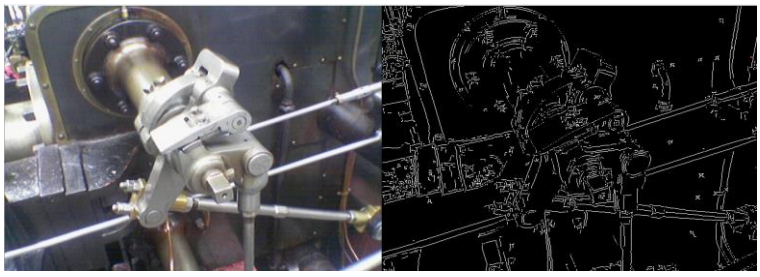

Fig. 8 The original image alongside final image after using NL-means filtering with sigma_h as 5, patch size as 7, neighborhood size as 23 and low and high threshold as .05 and .09 respectively. (Source 1, Source 2)

IV.CONCLUSION

In this work, we investigated enhancements to the classical canny edge detection algorithm by refining its key stages: noise reduction, gradient computation, and edge refinement. Our experiments demonstrated that NL-Means denoising outperformed Gaussian and morphological filters by preserving meaningful edges while effectively reducing noise. Similarly, the use of a 5×5 Sobel operator improved gradient estimation and edge continuity compared to traditional 3×3 kernels.

The proposed approach also maintained the strengths of non-maximum suppression, double thresholding, and hysteresis, resulting in thinner, sharper, and more robust edge maps. These improvements enhance the algorithm's applicability to tasks such as object detection, gesture recognition, and industrial quality inspection.

Future work will focus on integrating additional filtering methods, such as bilateral and guided filtering, as well as alternative edge-thinning strategies like morphological thinning and contour-based methods. These extensions have the potential to further improve edge localization and robustness in complex visual environments.

V.SOURCE CODE

You can find project's source codes in the link below. The code is written in Python version 3.12.0 using the dependencies

- **NumPy (1.26.2):** We used NumPy to do mathematical operations on the matrix.
- **OpenCV (4.8.1.78):** We have used the OpenCV library to read, write and show the image and some functions provided by the library.
- **Scipy (1.11.3):** We have used the Scipy library to use the convolution function.

You can find a detailed explanation of installing and running the program in the readme.md file provided in the repository.
https://gitlab.com/shantanukarki/Cannny-Edge_Detection

References

1. J. Canny, "A Computational Approach to Edge Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986, doi: 10.1109/TPAMI.1986.4767851.
2. A. Buades, B. Coll, and J.-M. Morel, "A Non-Local Algorithm for Image Denoising," in Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA, 2005, vol. 2, pp. 60–65, doi: 10.1109/CVPR.2005.38.
3. R. C. Gonzalez and R. E. Woods, Digital Image Processing, 4th ed. Hoboken, NJ, USA: Pearson, 2018.
4. C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," in Proc. 6th IEEE Int. Conf. Computer Vision (ICCV), Bombay, India, 1998, pp. 839–846, doi: 10.1109/ICCV.1998.710815.
5. H. Farid and E. H. Adelson, "Separating Reflections and Lighting Using Independent Components Analysis," J. Opt. Soc. Am. A, vol. 16, no. 9, pp. 2136–2145, Sept. 1999, doi: 10.1364/JOSAA.16.002136.
6. P. Perona and J. Malik, "Scale-Space and Edge Detection Using Anisotropic Diffusion," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 7, pp. 629–639, July 1990, doi: 10.1109/34.56205.
7. S. Paris, P. Kornprobst, J. Tumblin, and F. Durand, "Bilateral Filtering: Theory and Applications," Foundations and Trends in Computer Graphics and Vision, vol. 4, no. 1, pp. 1–73, 2009, doi: 10.1561/0600000020.
8. M. Heath, S. Sarkar, T. Sanocki, and K. Bowyer, "Comparison of Edge Detectors: A Methodology and Initial Study," Computer Vision and Image Understanding, vol. 69, no. 1, pp. 38–54, Jan. 1998, doi: 10.1006/cviu.1997.0544.
9. J. Weickert, "Anisotropic Diffusion in Image Processing," European Mathematical Society, 1998.
10. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600–612, Apr. 2004, doi: 10.1109/TIP.2003.819861.