

Design of Low Power Sobel Edge-8 Detection Algorithm using Verilog

Mr. G. Manga Rao¹, T. Chaithanya², M. Venkata Pavani³, M. Anu Sri⁴, K. Venkata Sai Chaithanya⁵,
Ch. Manikanta Venkata Sai⁶

¹Associate Professor, Dept. of ECE, PBR VITS, Kavali, Andhra Pradesh, India.

^{2,3,4,5,6}Department of Electronics and Communication Engineering,
PBR Visvodaya Institute of Technology & Science, Kavali (Autonomous),
SPSR Nellore (Dt.), Andhra Pradesh – 524201, India

Abstract - Edge detection is a fundamental operation in image processing, widely used in applications such as object recognition, medical imaging, pattern recognition, and computer vision. It identifies significant intensity changes in an image that correspond to object boundaries and structural details. Among various techniques, the Sobel operator is commonly used due to its simplicity and efficiency in approximating image gradients using horizontal and vertical convolution masks. However, the conventional Sobel method has limitations, including sensitivity to noise, restricted directional detection, and reduced accuracy in capturing fine or diagonal edges.

To address these issues, this project proposes a low-power Sobel Edge-8 detection algorithm implemented using Verilog. The enhanced approach computes gradients in eight different directions, enabling better detection of edges with varying orientations and finer details. The system is designed using Verilog HDL and evaluated through simulation and synthesis using Xilinx Vivado and MATLAB R2020a. Results show improved edge detection accuracy with low computational complexity, making the design suitable for real-time image processing and embedded vision applications due to its high speed, efficiency, reliability, and low power consumption.

Key words: Sobel edge detector, Vivado, Matlab.

1. INTRODUCTION

Edge detection is an essential image processing technique used to identify object boundaries and structural details in applications like object recognition, medical imaging, and computer vision. The Sobel Edge Detection Algorithm is widely used for its simplicity, but it is sensitive to noise and less effective in detecting fine edges. To overcome these limitations, this project proposes an improved Sobel algorithm using VLSI technology, enhancing gradient calculations for better accuracy and noise reduction. Implemented in Verilog HDL on FPGA and validated using Xilinx Vivado and ModelSim, the design achieves faster, more efficient, and reliable performance, making it suitable for real-time image processing applications.

2. LITERATURE SURVEY

The reviewed literature highlights advancements in secure image transmission, edge detection, and image processing techniques, emphasizing both performance and efficiency. The

VLSI implementation of the Blowfish algorithm provides a high-speed, low-latency solution for secure image transmission using FPGA-based parallelism and pipelining. In edge detection, simple methods like Sobel and Prewitt are fast but sensitive to noise and less effective for fine details, whereas the Canny edge detector offers superior accuracy, localization, and noise resistance at the cost of higher computational complexity, establishing a benchmark in computer vision. In image restoration, studies show that no single algorithm is universally optimal, with methods like Wiener filtering, median filtering, and constrained least squares performing best under specific noise conditions. Advanced techniques such as Structure Tensor Adaptive Total Variation (STA-TV), high-order total variation with inverse gradient, and Gaussian-adaptive bilateral filtering enhance denoising while preserving edges by adapting to local image features. Additionally, statistical noise estimation methods improve denoising without prior knowledge of noise characteristics. Overall, these developments improve image quality and processing efficiency, supporting applications in medical imaging, industrial inspection, computer vision, and emerging Industry 5.0 systems that integrate AI, edge computing, and human-machine collaboration.

3. EXISTING METHOD

Edge detection is a key technique in Digital Image Processing used to identify sharp changes in image intensity, which often correspond to object boundaries, texture variations, or surface details. It plays an important role in applications such as computer vision, medical imaging, object recognition, and image segmentation. Among various methods, the Sobel Edge Detection Algorithm is widely used because of its simplicity and efficiency. It is a gradient-based approach that approximates the first derivative of image intensity, detecting edges by measuring changes in pixel values in both horizontal and vertical directions.

The standard Sobel kernels are:

Horizontal Gradient Kernel (G_x)

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{---(5.1)}$$

Vertical Gradient Kernel (G_y)

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \text{---(5.2)}$$

These kernels are applied to the image through a convolution process. The resulting gradients represent the intensity change along the x-axis and y-axis respectively.

Once the gradients are computed, the magnitude of the gradient is calculated to determine the edge strength at each pixel location.

$$G = \sqrt{G_x^2 + G_y^2} \quad \text{----(5.3)}$$

In practical implementations, a simplified version is often used to reduce computational complexity:

$$G = |G_x| + |G_y|$$

The direction of the edge can also be estimated using the gradient direction:

$$\theta = \tan^{-1} \left(\frac{G_y}{G_x} \right)$$

Pixels with gradient magnitude values above a chosen threshold are classified as edges.

On performing convolution of the image matrix with the horizontal and vertical direction templates, we get the approximation of difference in the brightness as the gradients in X and Y directions.

Assume A as an input image,

$$A = \begin{bmatrix} p_0 & p_1 & p_2 \\ p_3 & p_4 & p_5 \\ p_6 & p_7 & p_8 \end{bmatrix} \quad \text{----(5.4)}$$

Using the Sobel operator, we have the gradient in X and Y direction, G_x and G_y as:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * A \quad \& \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * A \quad \text{----(5.5)}$$

To obtain the value of G, we must calculate the following equation:

$$G = \sqrt{G_x^2 + G_y^2} \quad \text{----(5.6)}$$

The equations required to apply the traditional Sobel filter are:

$$G_x = (p_2 - p_0) + ((p_5 - p_3) \ll 1) + (p_8 - p_6) \quad \text{----(5.7)}$$

$$G_y = (p_0 - p_6) + ((p_1 - p_7) \ll 1) + (p_2 - p_8) \quad \text{----(5.8)}$$

It is to be noted that some elements are not present in the above equations because those are multiplied to 0 in the X and Y Sobel matrices.

A) 8-Sobel Algorithm:

For 8-Sobel, the input matrix is of size 5x5, instead of 3x3 as in the case of Sobel and the elements of the matrix is from p_0 to p_{24} .

$$G_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -1 & -2 & -4 & -2 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 4 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad G_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & -4 & -2 & 0 \\ -1 & -4 & 0 & 4 & 1 \\ 0 & 2 & 4 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$G_3 = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 \\ 0 & -2 & -4 & 0 & 1 \\ 0 & -4 & 0 & 4 & 0 \\ -1 & 0 & 4 & 2 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad G_4 = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -2 & -4 & 2 & 0 \\ 0 & -4 & 0 & 4 & 0 \\ 0 & -2 & 4 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$G_5 = \begin{bmatrix} 0 & -1 & 0 & 1 & 0 \\ 0 & -2 & 0 & 2 & 0 \\ 0 & -4 & 0 & 4 & 0 \\ 0 & -2 & 0 & 2 & 0 \\ 0 & -1 & 0 & 1 & 0 \end{bmatrix} \quad G_6 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & -2 & 4 & 2 & 0 \\ 0 & -4 & 0 & 4 & 0 \\ 0 & 2 & -4 & 2 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

$$G_7 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 4 & 2 & 0 \\ 0 & -4 & 0 & 4 & 0 \\ 0 & -2 & -4 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 \end{bmatrix} \quad G_8 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 4 & 2 & 0 \\ -1 & -4 & 0 & 4 & 1 \\ 0 & -2 & -4 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The equations required to apply the 8-Sobel filter are:

$$G_1 = ((p_{19} + p_{15} - p_5 - p_9) + ((p_{16} + p_{18} - p_8 - p_6) \ll 1) + ((p_{17} - p_7) \ll 2)). \quad \text{----(5.9)}$$

$$G_2 = ((p_{14} - p_{10}) + ((p_{16} + p_{18} - p_6 - p_8) \ll 1) + ((p_{13} + p_{17} - p_{11} - p_7) \ll 2)). \quad \text{----(5.10)}$$

$$G_3 = ((p_{21} + p_9 - p_3 - p_{15}) + ((p_{18} - p_6) \ll 1) + ((p_{17} + p_{13} - p_7) \ll 2)). \quad \text{----(5.11)}$$

$$G_4 = ((p_{22} - p_2) + ((p_{18} - p_6) \ll 1) + ((p_{13} + p_{17} - p_7) \ll 2)). \quad \text{----(5.12)}$$

$$G_5 = ((p_{23} + p_3 - p_{21} - p_1) + ((p_{18} + p_8 - p_6 - p_{16}) \ll 1) + ((p_{13} - p_{11}) \ll 2)). \quad \text{----(5.13)}$$

$$G_6 = ((p_2 - p_{22}) + ((p_{16} + p_{18} - p_6 + p_8) \ll 1) + ((p_{13} - p_{17} - p_{11} + p_7) \ll 2)). \quad \text{----(5.14)}$$

$$G_7 = ((p_1 + p_{19} - p_5 - p_{23}) + ((p_8 - p_{16}) \ll 1) + ((p_7 + p_{13} - p_{11} - p_{17}) \ll 2)). \quad \text{----(5.15)}$$

$$G_8 = ((p_{14} - p_{10}) + ((p_{16} - p_{18} + p_6 + p_8) \ll 1) + ((p_{13} - p_{17} + p_{11} + p_7) \ll 2)). \quad \text{----(5.16)}$$

B) Algorithm used to implement Sobel and 8-Sobel using Verilog:

The given flowchart shows the implementation of traditional Sobel and the improved Sobel in eight directions.

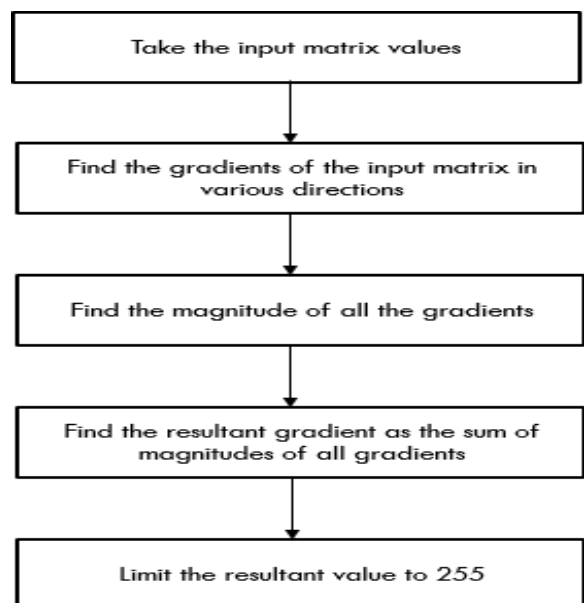


Fig. 1: Algorithm for Sobel and 8-Sobel.



Fig. 2: Existing Method

The above Fig 2 shows the workflow of a VLSI-based Sobel edge detection system. The diagram shows a simple process of edge detection. First, an input image is given to the system. It is then processed using the Sobel edge detection algorithm implemented in VLSI. After processing, the system produces an output image where only the edges (boundaries) of objects are highlighted.

Limitations: Sensitivity to Noise, Limited Directional Detection, Thick Edge Output, Poor Performance in Low-Contrast Images, Fixed Kernel Size, Lack of Adaptive Thresholding, Edge Localization Errors.

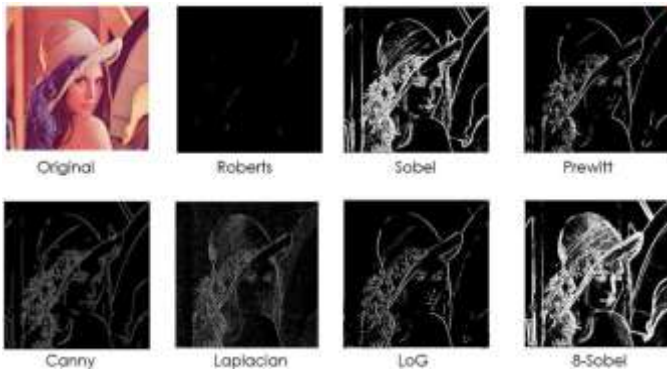


Fig. 3: Result For Lenna input image using various Algorithms.

4. PROPOSED METHOD

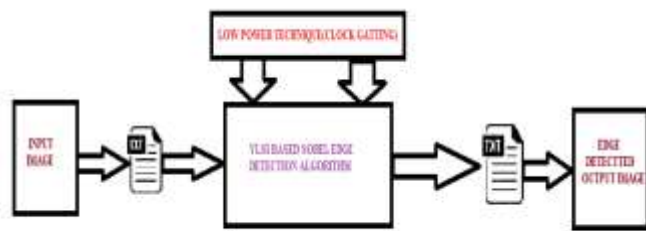


Fig. 4: Proposed Block design

The system is designed with scalability, allowing adjustment of image resolution and processing speed through configurable parameters, making it suitable for a wide range of devices and performance requirements. It begins with an input image, typically in grayscale, where each pixel represents intensity values used for edge detection. This image is converted into a text (.txt) format so that hardware simulation tools can process the numerical pixel data. The converted data is then fed into the VLSI-based Sobel Edge Detection algorithm, which serves as the core processing unit. In this stage, pixel values are analyzed using Sobel convolution masks to compute horizontal and vertical gradients, enabling accurate edge detection through efficient parallel processing in FPGA hardware.

To enhance energy efficiency, a low-power technique called clock gating is implemented, which reduces unnecessary switching activity by disabling inactive circuit components, thereby lowering dynamic power consumption. After processing, the edge-detected pixel values are stored again in a text file and later converted back into an image format. The final output image highlights edges as bright regions while non-edge areas remain darker, clearly revealing object boundaries and structural details, and demonstrating the effectiveness, speed, and reliability of the hardware-based improved Sobel edge detection system.

A) Comparison Between Existing method and Proposed method

Table 1: Comparison Table

METHODS	AREA(LUT'S)	POWER(W)	DELAY(ns)
BASE	38	14.424	15.591
PROPOSED	39	7.693	15.591

5. RESULTS



Fig. 5: Original Input Image



Fig. 6: Output Image



Fig. 7: Power Report of Proposed Algorithm



Fig. 8: Time Report of Proposed Algorithm

CONCLUSION

Edge detection is a key technique in image processing for identifying object boundaries and structural details in applications like computer vision and medical imaging. Although the Sobel algorithm is simple and efficient, it is sensitive to noise and less effective for fine or diagonal edges. This project proposes an improved Sobel method using VLSI technology, enhancing gradient computation for better accuracy and noise resistance. Implemented in Verilog HDL on FPGA, the design enables high-speed, low-power, and real-time processing. Simulation results using Xilinx Vivado show improved edge clarity and efficiency, making the system a reliable and scalable solution for advanced image processing applications.

DISCUSSION

The proposed low-power Sobel Edge-8 detection algorithm improves upon the traditional Sobel method by enabling gradient computation in eight directions, allowing better detection of fine and diagonal edges while reducing noise sensitivity. Implemented using Verilog HDL on FPGA and validated with Xilinx Vivado and MATLAB R2020a, the system achieves high speed, low latency, and efficient parallel processing. The results demonstrate improved accuracy with low computational complexity and power consumption, making it suitable for real-time and embedded image processing applications.

ACKNOWLEDGEMENT

The authors sincerely thank Mr. G. Manga Rao (Associate Professor, ECE, PBR VITS, Kavali) for his guidance, Dr. R. Sravanthi (Professor & HoD, ECE) for providing facilities, and Dr. V. Anil Kumar (Principal, PBR VITS Kavali) for the academic environment that enabled this work.

REFERENCES

1. Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6), 679-698.
2. Gota, A., & Min, Z. J. (2013). Analysis and Comparison on Image Restoration Algorithms Using MATLAB. *International Journal of Engineering Research & Technology (IJERT)* Vol, 2, 1350-1360.

3. Mahalakshmi, A.,& Shanthini, B. (2016, January). A survey on image deblurring. In 2016 International Conference on Computer Communication and Informatics (ICCCI) (pp. 1-5). IEEE.
4. Flusser, J., Farokhi, S., Hoschl, C., Suk, T., Zitov " a, B., ' & Pedone, M. (2015). Recognition of images degraded by Gaussian blur. *IEEE transactions on Image Processing*, 25(2), 790-806.
5. Ramya, S.,& Christial, T. M. (2011, March). Restoration of blurred images using Blind Deconvolution Algorithm. In 2011 International Conference on Emerging Trends in Electrical and Computer Technology (pp. 496-499). IEEE.
6. Sada, M. M., & Mahesh, M. G. (2018). Image deblurring techniques—a detail review. *Int. J. Sci. Res. Sci. Eng. Technol*, 4(2), 15.
7. Ansari, M. A., Kurchaniya, D., & Dixit, M. (2017). A comprehensive analysis of image edge detection techniques. *International Journal of Multimedia and Ubiquitous Engineering*, 12(11), 1-12.
8. Syahrian, N. M., Risma, P., & Dewi, T. (2017). Vision-based pipe monitoring robot for crack detection using canny edge detection method as an image processing technique. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 243- 250.
9. Jain, S., & Goswami, M. S. (2015). A comparative study of various image restoration techniques with different types of blur. *International Journal Of Research In Computer Applications And Robotics*.
10. Yadav, S., Jain, C., & Chugh, A. (2016). Evaluation of image deblurring techniques. *International Journal of Computer Applications*, 139(12), 32- 36.
11. Verma, R., & Ali, J. (2013). A comparative study of various types of image noise and efficient noise removal techniques. *International Journal of advanced research in computer science and software engineering*, 3(10).
12. Sekehravani, E. A., Babulak, E., & Masoodi, M. (2020). Implementing canny edge detection algorithm for noisy image. *Bulletin of Electrical Engineering and Informatics*, 9(4), 1404-1410.
13. Saini, S., Kasliwal, B., & Bhatia, S. (2013). Comparative study of image edge detection algorithms. *arXiv preprint arXiv:1311.4963*
14. (Jan. 2021). European Commission, *Industry 5.0: Towards a Sustainable, Human-Centric and Resilient European Industry*. [Online]. Available: <https://op.europa.eu/en/publication-detail/-/publication/468a892a-5097-11eb-b59f-01aa75ed71a1/>
15. P. K. R. Maddikunta, Q.-V. Pham, B. Prabadevi, N. Deepa, and K. Dev, "Industry 5.0: A survey on enabling technologies and potential applications," *J. Ind. Inf. Integr.*, vol. 26, pp. 1–31, Mar. 2022, doi: 10.1016/j.jii.2021.100257.