High-performance VLSI architecture for edge detection in advanced image processing applications

Bandari Srilekha[1], Swathi Katta[1], Durgaprasad Anagandula[1]
[1]Department of Electronics and Communication Engineering
[1]Sree Dattha Group of Institutions, Sheriguda, Hyderabad, Telangana.

ABSTRACT

Edge detection has a wide range of applications, including object recognition, image segmentation, and scene analysis. It is crucial in autonomous vehicles, robotics, medical imaging, surveillance systems, and various industrial automation processes. Traditional edge detection methods, like the Sobel and Canny operators, have been widely employed due to their simplicity and effectiveness. The computation of gradients and thresholds in conventional edge detectors requires extensive processing, resulting in high computational costs. The proposed VLSI implementation of the Sobel edge detectors utilizes hybrid accumulations to address the limitations of traditional methods. By using hybrid accumulations, the hardware resource requirements are optimized, making it feasible to implement the edge detectors on resource-constrained platforms.

Keywords: Image Processing,MATLAB, Pixel Analysis, Image Recognition, Sub-window, Very Large-Scale Integration.

DOI Number: 10.48047/nq.2020.18.12.NQ20264 NeuroQuantology 2020; 18(12):298-304

1. INTRODUCTION

Edge detection stands as a foundational process within the domain of image processing, acting as a gateway to revealing the intricate structures inherent in digital images. In essence, the primary objective of edge detection is to pinpoint areas within an image where there is a sudden and significant change in intensity. These changes often correspond to boundaries or transitions between distinct objects or features, forming the building blocks for more advanced image analysis and interpretation.In the visual realm, the notion of an "edge" encapsulates the boundaries that define various elements within an image. It could be the outline of an object, the contour of a shape, or any region where there is a discernible shift in pixel intensity. Edges, therefore, serve as crucial cues for understanding the underlying structure and content of an image. By isolating and highlighting these edges, the complexity of visual data is distilled into a more comprehensible form, facilitating subsequent tasks such as object recognition, segmentation, and feature extraction.The concept of edge detection finds its roots in the broader field of computer vision, where the objective is to endow machines with the ability to interpret and understand visual information akin to human perception. In the absence of such computational tools, the vast amount of visual data captured by cameras and sensors remains largely untapped. Edge detection acts as a gateway to unlock the potential encoded within this visual data, transforming it into meaningful information that can be harnessed across various applications and industries.The challenge in edge detection lies in developing algorithms and methodologies that can discern these subtle transitions in intensity amidst the noise and complexity of real-world images. This complexity is compounded by factors such as varying lighting conditions, image artifacts, and the presence of textures. Over the years,

researchers and practitioners have devised a plethora of techniques, ranging from simple gradient-based operators like Sobel and Prewitt to more sophisticated methods like Canny edge detection and Laplacian of Gaussian filtering.

2. LITERATURE SURVEY

Midde, Venkata Siva,et.al [1] proposed three pipeline VLSI architectures for high-quality image scaling. The proposed architectures used a low-complexity v-model spatial sharpening filter, modified edge detection, and simplified bilinear interpolation. The low-complexity v-model spatial sharpening filter was obtained by modifying the 3x3 sharpening kernel to x-model and then to v-model. Sivasankari, B.,et.al [2] discussed Boundary recognition played a vital role in real-world scenarios like medical imaging and surveillance. The classical edge detector failed to preserve minute details in the image. The abdominal CT images were first analyzed using two edge detection techniques based on gauss gradient, proposed here for real-time data processing. Upadhyay, Bharat Bhushan,et.al [3] discussed Hazy weather degraded the vividness of the images captured by real-time systems for applications such as object detection, remote sensing, and surveillance systems. This affected the performance of such real-time systems. The hardware implementation of a real-time haze removal system was imperative to solve these problems.

Sivanandam, K,et.al [4] observed to avoid the carry resulting from the shortened component, this study suggested a truncation-based approximate multiplier with a compensation circuit created by selected k-map adjustments. Output error reduction and hardware trimming were accomplished concurrently. For error correction, 16x16 truncated multipliers based on approximate multipliers were created using streamlined NAND gate circuits. Kashyap, Sidharth,et.al [5] proposed a resource-efficient architecture of Sobel edge detector, Prewitt edge detector, Robert's edge detector, and Laplacian edge detector on the field-programmable gate arrays (FPGAs). The proposed architecture was less complex as well as used lower resources compared to the conventional architecture. Yun, Heuijee,et.al [6] introduced with the development of various chips, such as VLSI chips, and the progress in semiconductor technology and artificial intelligence, autonomous driving technology advanced daily. Lane recognition technology, considered crucial in implementing autonomous vehicles, required a substantial amount of computation and processing time, as data needed to be received from a camera attached to the vehicle and processed in real time.

Joshi, Rajeev,et.al [7] proposed a simulated annealing (SA)-based neural network (NN) optimization approach for designing an area and power-efficient hardware model for multilayer perceptron's. Our approach targeted optimizing the hidden layer neuron weights to integer values. For each SA move, a subset of the hidden weights was perturbed at random. Marimuthu, Mohanapriya,et.al [8] introduced Continuous improvements in very-large-scale integration (VLSI) technology and design software have significantly broadened the scope of digital signal processing (DSP) applications. Li, Tao,et.al [9] developed A hybrid signed convolution module with the architecture of an unsigned divide-and-conquer (UDC) multiplier was proposed to improve the functional diversity and energy efficiency of artificial intelligence (AI) accelerators based on spin-transfer-torque magnetic random-access memory (STT-MRAM).

Nishio, Kimihiro,et.al [10] illustrated Novel basic analog–digital edge detection circuits in this research were proposed based on the vertebrate retina. The vertebrate retina serves as a pre-processor for image processing in the brain and possesses superior functions such as edge detection and motion detection. Bharadwaj, N. Akash,et.al [11] developed the technique of segmenting a picture involves breaking it up into its component parts, such as boundaries and regions. Image edges are found via edge detection algorithms. Edges of the picture are detected when the pixels' intensities abruptly shift. Sobel, Prewitt, and Robert are three well-known gradient-based

edge detection algorithms that operate on the 3-by-3 and 2-by-2 kernels in the X and Y axis.Nagaraj, P.,et.al [12] proposed a method is introduced to automate or regulate green traffic signal allocation time based on traffic conditions at the roadside using the Canny Edge Detection Algorithm. The most recent traffic picture is sent to the program, and the application extracts edges using this strategy. Arbeletche, Yuri,et.al [13] presented MAxPy, a framework for bridging approximate computing (AxC) circuit design to its applications. MAxPy was an application-agnostic framework capable of automatically building a cycle-accurate Python model of an approximate hardware design. This model could easily be emulated and integrated as a module in Python-based applications.

Anurag, Ranga,et.al [14] illustrated the external system was a subsystem that constituted a part of a receiver application. The Composite Blanking Cover Pulse module (CBCP) was one of the major functional modules that needed to be designed and developed for the External Sub-system. The external system (target hardware) was designed and developed based on Virtex 5 FPGA. Parmar, Rushik,et.al [15] introduced Atrial fibrillation (AF) was a recurrent and life-threatening disease leading to a rapid increase in the mortality rate due to cardiac abnormalities. Manual diagnosis of AF using electrocardiogram (ECG) signals was challenging due to the complex and varied changes in its characteristics.

3.PROPOSED METHODOLOGY

Figure 1 shows the proposed system architecture. The process begins with the input image, a digital representation of visual content. This image serves as the canvas for subsequent processing. The characteristics of the input image, such as its resolution and color depth, impact the overall performance of the edge detection unit.

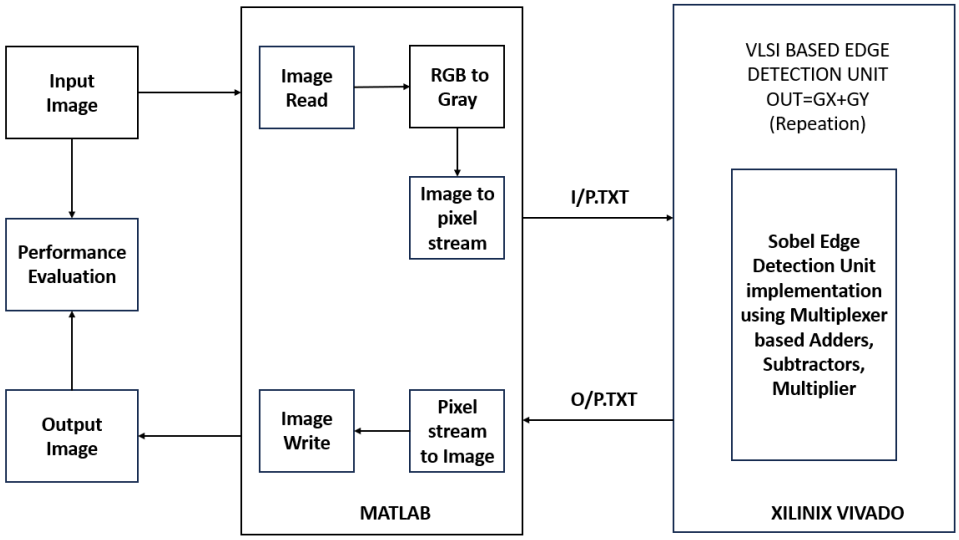

Figure 1. Proposed block diagram.

Image Read: The image read operation is pivotal for bringing the digital input image into the VLSI-based system's memory. This step involves retrieving pixel values from external sources, such as image files or imaging devices, and loading them into the system for further analysis.

RGB to Gray: Following image read, the RGB to Gray conversion is often employed as a preprocessing step. This operation simplifies subsequent computations by transforming the color image from the RGB color space to grayscale. This reduction in dimensionality facilitates more efficient edge detection processes.

Image to Pixel Stream: To enable streamlined processing within the VLSI architecture, the image is converted into a pixel stream. This involves organizing pixel values into a sequential flow, optimizing the system for parallel processing. The pixel stream format is conducive to efficient hardware operations in subsequent stages.

Input TXT: In certain scenarios, the edge detection unit may require additional parameters or configurations. This

information is often stored in a text file (input TXT) and serves as a guide for the subsequent stages of the edge detection process. It could include values such as filter coefficients, threshold parameters, or other essential settings.

Sobel Edge Detection Unit: The Sobel edge detection unit takes center stage in the processing pipeline. This specialized VLSI hardware component applies the Sobel operator, a convolution kernel, to compute gradients and accentuate areas with significant intensity changes. The Sobel unit plays a critical role in identifying potential edges within the image.

Output Text: The results of the edge detection process are often recorded in an output text file. This file may contain valuable information such as the coordinates of detected edges, statistical measures, or any other relevant data generated during the Sobel operation and subsequent stages.

Pixel Stream to Image: Post-edge detection, the pixel stream is transformed back into a two-dimensional image format. This conversion is crucial for visualizing the identified edges and interpreting the results. The reconstructed image can be further analyzed or utilized for downstream applications.

Image Write: The processed image or its results are written to an external storage device or memory. The image write operation finalizes the edge detection unit's processing, making the output available for additional analysis, display, or integration into broader image processing workflows.

Output Image: The culmination of the entire process is the output image, where edges are distinctly highlighted. This visual representation encapsulates the success of the edge detection unit, providing a clear depiction of the identified features within the original input image. The output image serves as a valuable output for applications in computer vision, pattern recognition, and other fields that leverage image processing techniques.

4.Results and Discussion

Figure 2shows the outcome of simulating both the existing and proposed edge detection methods using VLSI. Figure 3provides a design summary of the VLSI implementations for edge detection. Figure 4 presents a power summary, showcasing the power consumption metrics of the VLSI implementations for edge detection. It could include information on static power (power consumed when the circuit is idle) and dynamic power (power consumed during operation), providing insights into the energy efficiency of the designs.

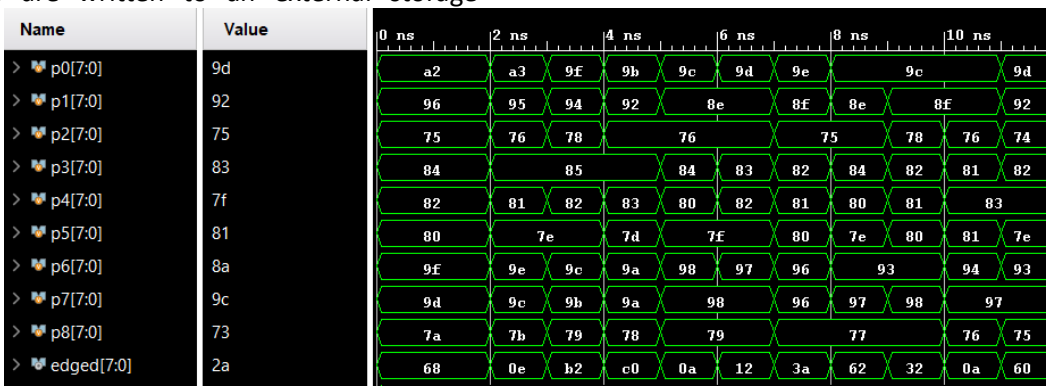

Figure 2. Simulation Outcome.

Resource	Utilization	Available	Utilization %
LUT	117	133800	0.09
IO	73	500	14.60

Figure3. Design Summary.

Figure 5 depicts a time summary, detailing the time-related metrics of the VLSI implementations for edge detection. Figure 6 illustrate the performance of Sobel edge detection, a common technique

used in image processing, with the proposed method. Part (a) shows an input image, while part (b) displays the edge-detected output image generated by the proposed Sobel method. This comparison visually demonstrates the effectiveness of the proposed method in detecting edges in the input image.

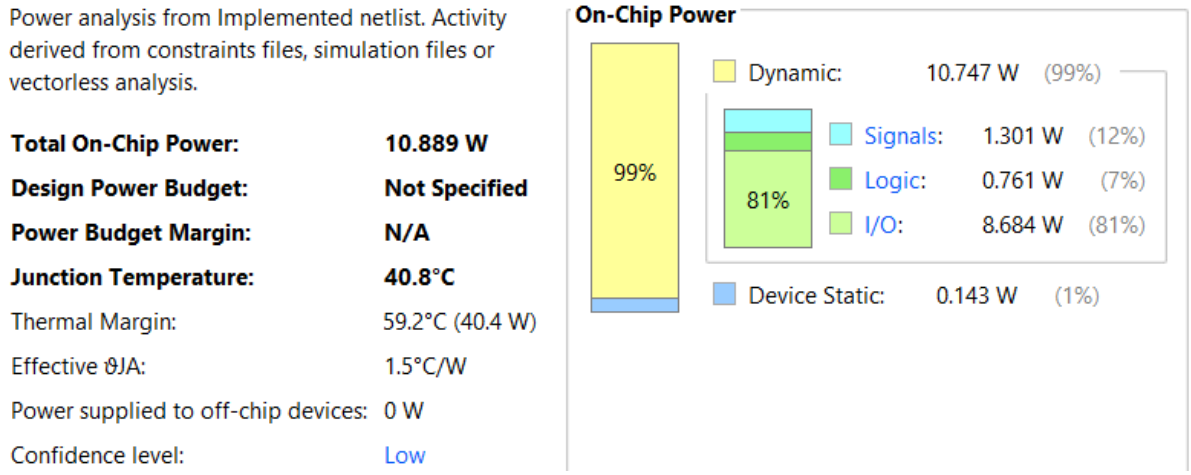

Figure 4. Power summary.

Unconstrained Paths - NONE - NONE - Hold

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock
Path 10	∞	5	3	3	p2[7]	gradient_magnitude[8]	3.501	2.001	1.500	-∞	input port clock
Path 11	∞	5	3	4	p6[5]	gradient_magnitude[6]	3.686	2.041	1.645	-∞	input port clock
Path 12	∞	6	4	5	p6[3]	gradient_magnitude[5]	3.751	2.174	1.577	-∞	input port clock
Path 13	∞	5	3	3	p2[7]	gradient_magnitude[7]	3.752	1.998	1.753	-∞	input port clock
Path 14	∞	6	4	5	p6[3]	gradient_magnitude[4]	3.784	2.158	1.627	-∞	input port clock
Path 15	∞	5	3	5	p6[0]	gradient_magnitude[1]	3.833	2.009	1.824	-∞	input port clock
Path 16	∞	5	3	5	p6[0]	gradient_magnitude[2]	3.859	2.060	1.799	-∞	input port clock
Path 17	∞	5	3	5	p6[3]	gradient_magnitude[3]	3.894	2.024	1.870	-∞	input port clock
Path 18	∞	5	3	5	p6[0]	gradient_magnitude[0]	3.901	1.976	1.925	-∞	input port clock

Figure 5. Time summary.

Table 1 provides a comparison table between the existing threshold-based edge detection method and the proposed Sobel method across various parameters or performance metrics. These metrics include Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Gradient Error (GE). The comparison highlights the differences in edge detection performance between the two methods, with the proposed Sobel method showing significant improvements in several metrics, such as MSE, PSNR, and SSIM.

(a) (b)

Figure 6. (a) Input Image. (b) Proposed Sobel Edge Detection Output Image.

Table 1. Comparison Table.

Parameters	Existing Threshold Method	Proposed Sobel Method
Mean Square Error (MSE)	0.1096964	0.0307747
Peak Signal to Noise Ratio (PSNR)	9.94547	15.1911
SSIM	4.055944	0.8747
Gradient Error (GE)	0.886877	0.23615

5.CONCLUSION

The VLSI implementation of edge detection for image processing systems mainly focuses on the design and simulation System of the Sobel edge detection method. This method uses two 3×3 convolution masks to estimate gradient in X and Y-direction and which is easier to implement than other operators. The Sobel edge detection method calculates 2-D spatial gradient of image intensity at each point of an image by convolution with small and integer valued filters therefore relatively less expensive in terms of computations. It has proven to be a significant advancement in enhancing the efficiency of real-time image analysis. By leveraging the capabilities of VLSI technology, we have successfully translated complex edge detection algorithms into hardware circuits that can swiftly identify and highlight edges in images. This not only speeds up the image processing tasks but also reduces the computational burden on the overall system.The optimized hardware design allows for quicker response times and increased accuracy in detecting edges, contributing to the overall improvement of image processing systems. As technology continues to advance, the integration of VLSI in edge detection paves the way for more efficient and powerful image processing solutions, offering practical benefits across various industries.

References

[1] Midde, Venkata Siva, and E. P. Jayakumar. "Low-cost low-power approximated VLSI architecture for high-quality image scaling in mobile devices." Journal of Real-Time Image Processing 20, no. 1 (2023): 11.

[2] Sivasankari, B., A. Ahilan, and S. N. Kumar. "FPGA Implementation of Gauss Gradient Edge Detectors for Medical Images." Wireless Personal Communications (2023): 1-18.

[3] Upadhyay, Bharat Bhushan, and Kishor Sarawadekar. "VLSI Design of Saturation-Based Image Dehazing Algorithm." IEEE Transactions on Very Large-Scale Integration (VLSI) Systems (2023).

[4] Sivanandam, K., R. Jagadheesh, SN Kavi Bharath, and S. Manoj. "Low Power Design of Edge Detector using Static Segmented Approximate Multipliers." In 2023 7th International Conference on Computing Methodologies and Communication (ICCMC), pp. 739-744. IEEE, 2023.

[5] Kashyap, Sidharth, Ashish Kumar Bhandari, and Pushpa Giri. "Low resource FPGA implementation based efficient image edge detector architecture." Multimedia Tools and Applications (2023): 1-21.

[6] Yun, Heuijee, and Daejin Park. "FPGA Realization of Lane Detection Unit using Sliding-based Parallel-Segment Detection for Buffer Memory Reduction." In 2023 IEEE International Conference on Consumer Electronics (ICCE), pp. 1-2. IEEE, 2023.

[7] Joshi, Rajeev, Lakshmi Kavya Kalyanam, and Srinivas Katkoori. "Area Efficient VLSI ASIC Implementation of Multilayer Perceptrons." In 2023 International VLSI Symposium on Technology, Systems and Applications (VLSI-TSA/VLSI-DAT), pp. 1-4. IEEE, 2023.

[8] Marimuthu, Mohanapriya, Santhosh Rajendran, Reshma Radhakrishnan, Kalpana Rengarajan, Shahzada Khurram, Shafiq Ahmad, Abdelaty Edrees Sayed, and Muhammad Shafiq. "Implementation of VLSI on Signal Processing-Based Digital Architecture

Using AES Algorithm." Computers, Materials & Continua 74, no. 3 (2023).

[9] Li, Tao, Yitao Ma, Ko Yoshikawa, and Tetsuo Endoh. "Hybrid signed convolution module with unsigned divide-and-conquer multiplier for energy-efficient STT-MRAM-based AI accelerator." IEEE Transactions on Very Large Scale Integration (VLSI) Systems (2023).

[10] Nishio, Kimihiro, and Natsumi Kuroda. "Basic analog–digital integrated circuit for edge detection based on the vertebrate retina." Artificial Life and Robotics 28, no. 4 (2023): 797-802.

[11] Bharadwaj, N. Akash, Muhammad Afsar, Kapish Kumar Khaitan, Rahul, and C. Gururaj. "Efficient FPGA-Based Implementation of Image Segmentation Algorithms for IoT Applications." In Low Power Architectures for IoT Applications, pp. 25-45. Singapore: Springer Nature Singapore, 2023.

[12] Nagaraj, P., V. Muneeswaran, K. Saiteja, D. Abhishaek Yadav, K. Manikanta, and S. Krishna Aditya. "Comparative Analysis of Canny Edge Detection-Based Smart Traffic Control System Using Multi-Type Image Processing." In 2023 International Conference on Computer Communication and Informatics (ICCCI), pp. 1-11. IEEE, 2023.

[13] Arbeletche, Yuri, Guilherme Paim, Brunno Abreu, Sérgio Almeida, Eduardo Costa, Paulo Flores, and Sergio Bampi. "MAxPy: A Framework for Bridging Approximate Computing Circuits to its Applications." IEEE Transactions on Circuits and Systems II: Express Briefs (2023).

[14] Anurag, Ranga, and AS Keerthi Nayani. "VLSI Implementation of Blanking Cover Pulse and CBCP module Propagation for External System Interface." In 2023 International Conference for Advancement in Technology (ICONAT), pp. 1-6. IEEE, 2023.

[15] Parmar, Rushik, MeenaliJanveja, Jan Pidanic, and Gaurav Trivedi. "Design of DNN-Based Low-Power VLSI Architecture to Classify Atrial Fibrillation for Wearable Devices." IEEE Transactions on Very Large Scale Integration (VLSI) Systems 31, no. 3 (2023): 320-330.