

A Lightweight Confidence-Guided Adaptive Sobel FPGA Implemented Architecture with Spatially Supported Edge Decisions

Kuldeep Singh¹, Vinod Kumar², Ajay Kumar³, Rhythem⁴, Govind⁵

^{1,2,3,4,5}Department of Electrical & Electronics Engineering, Guru Jambheshwar University of Science & Technology, Hisar, India-125001

kuldeepsingh@gjust.org, vinodspec@yahoo.co.in; ajaypunia506@gmail.com;
rhythem2216@gmail.com; govindsaini00100@gmail.com

Abstract:

In the implementation of the emerging embedded computer-vision systems the real-time edge detection is becoming an essential preprocessing operation. The fixed-threshold Sobel realizations have advantages because they have simple arithmetic circuits with low complexity, but it is difficult for a single threshold to explicitly differentiate between strong responses, uncertain weak responses, and its background. In this paper a confidence-aware adaptive Sobel edge-detection architecture is presented and realized on FPGA with timing-closed performance and physically validated FPGA realization on a Zynq-7000 XC7Z020 Zed-Board. The architecture basically used a method that combines a conventional 3×3 Sobel gradient operator with three-level image recognition metric as confidence classification and surrounding neighborhood supported approval of intermediate-gradient responses. The recommended hardware architecture is based on utilization of synchronous dual-port frame buffers, a pipelined processing or display path, and a mixed-mode clock manager (MMCM)-derived VGA pixel clock. The behavioral verification experiment is executed for 1,679,998 pixel-clock samples, 2,099 raster lines, and four frame starts with zero assertion errors and zero monitor warnings; the calculated pixel-clock frequency was 25.174825 MHz. The post-synthesis timing shows WNS = -0.443 ns and TNS = -2.069 ns, whereas the routed implementation achieved improvement in WNS = +0.022 ns, TNS = 0 ns, and WHS = +0.064 ns. The implemented design utilizes 1,563 LUTs, 1,284 flip-flops, one RAMB36 and two RAMB18 blocks, and one MMCM. The total post-route analysis estimates the on-chip power of 0.254 W. The results obtained shows the practical hardware implementation feasibility and timing closure with the generated bitstream. The target hardware was programmed on the Zed-Board and shows validation using an externally connected VGA display. The expected three-panel display—input in grayscale, conventional Sobel as baseline, and confidence-aware adaptive as output—was reported, and a multiple SW [7:0] switch configurations was exercised and produced repeatable threshold-dependent changes in the detected-edge maps. The observed results establish the simulation correctness, timing closure, bitstream generation, and physical on-board VGA functionality test for the desired system.

Keywords: Adaptive edge detection, confidence-aware processing, FPGA, Sobel operator, Zynq-7000, VGA, timing closure, embedded vision.

I. Introduction

It has been acknowledged that edge detection rests a basic operation in image analysis as it converts the change in brightness into a condensed low dimensional representation of image for segmentation, feature extraction, object localization, inspection, and higher-level vision. The Sobel operator is highly desirable for FPGA implementation as its 3×3 kernels can be paired to additions, subtractions, shifts, line buffers, and comparators like operations. The past research on FPGA has often demonstrated the real-time Sobel processing described in HDL implementations with camera-to-VGA pipelines, and OpenCL or HLS-based implementations¹⁻⁹.

The gradient calculation is not the primary constraint of a traditional Sobel detector; rather, it is the final decision rule. When set at a high level, a fixed threshold can effectively suppress weak but spatially meaningful boundaries, while a low threshold can allow for isolated responses and noise. As a result, adaptive threshold level has been examined in canny variants, distributed block-based detectors, edge-proportion statistics, histogram techniques, fuzzy decision systems, and application-specific improved Sobel operators^{10-12, 15-22}. These experiments show that adaptive behavior is valuable, but they also illustrate a recurring hardware tradeoff: deeper threshold estimation, non-maximum suppression, hysteresis, optimization, or statistical analysis can increase memory, control complexity, and latency.

The present work gives solution for a miniaturizing hardware components where a light-weight confidence layer can be inserted in to a conventional Sobel pipeline in a way that strong gradients are passed immediately, whereas weak gradients are dropped, and intermediate gradients are passed only when a causal spatial-support condition is met out. The contribution is therefore positioned conservatively. The proposed work here is not claiming to be the very first adaptive edge detector, the first FPGA Sobel implementation view, or a replacement for the full Canny hysteresis. But the contribution is the integration of a three-level confidence decision within a compact Sobel datapath, along with a timing-closed Zynq-7000 Confidence-Guided Adaptive Sobel architecture using synchronous frame buffers and a pipelined VGA path.

II. Literature Review

A substantial body of work has mapped Sobel filtering to reconfigurable hardware. Chaple and Daruwala described a VHDL Sobel design for grayscale images¹, and Chaple et al. compared Roberts, Prewitt, and Sobel operators for FPGA-oriented real-time use². Jeyakumar et al. demonstrated FPGA Canny processing from real-time image capture to VGA display³, while Yuan and Xu proposed an adaptive Canny variant to recover weak-gradient edges through combined global and local processing⁴. Sateesh Kumar et al. demonstrated adaptive-threshold FPGA processing in an object/face-detection pipeline⁵. Development methodology has also evolved. You et al. implemented Sobel processing on FPGA using OpenCL⁶; Yusoff et al. used Sobel hardware for real-time Hevea leaf disease identification⁷; Yaman et al. integrated OV7670 capture, grayscale conversion, a Sobel IP core, and VGA display on a Basys3 platform⁸; and Chen et al. studied Sobel design and verification using high-level synthesis⁹. Collectively, these works establish that Sobel is a mature FPGA primitive. Consequently, the novelty of the present study is not the 3×3 convolution, VGA output, or FPGA mapping by themselves. For hardware-oriented adaptive detection, Possa et al. introduced a self-adapting FPGA architecture for Canny and Harris feature detection using adaptable neighborhood extraction and algorithm simplifications¹⁰, later extending the

approach to a multi-resolution parameterizable architecture with runtime threshold operators and FPGA/GPU/CPU comparisons¹¹. Xu et al. proposed a distributed Canny detector in which thresholds are computed from block type and local gradient distributions; their Virtex-5 architecture reduced latency by replacing frame-level threshold statistics with block-level processing¹². These works demonstrate that local or runtime adaptation is compatible with FPGA processing, but their target is a substantially richer Canny/feature-detection pipeline than the compact Sobel supervisory mechanism considered here. J. Canny formulated the criteria for edge detection, around detection, localization, and single-response¹³, while N. Otsu gives a standard principle for histogram-based threshold-selection¹⁴. Although the present architecture uses Sobel gradients rather than a full Canny pipeline, these works remain relevant because they distinguish gradient estimation from the decision mechanism applied to the gradient response.

The latest research has given a special attention on how the edge decision is modified along with the gradient kernel. Liu et al. has proposed an adaptive and long lasting strong detector based on edge-proportion statistics and two-dimensional entropy, stating that a single generic threshold is less suitable for diverse image contents¹⁵. Tan and He investigated the FPGA implementation flow for Canny video edge detection¹⁶. Chen and Yang combined the three (i) an adaptive Sobel formulation with (ii) an improved genetic algorithm and (iii) non-maximum suppression¹⁷. Gayathri and Remya Ajai realized an improved Sobel architecture with eight-direction and evaluated image-level metrics for synthesis¹⁸. Wei et al. demonstrated an FPGA accelerator design for real-time video edge detection technique¹⁹, while Hagara et al. reported and described FPGA histogram-based threshold selection mechanism²⁰. The emerging and demanding application-driven variants further highlighted the importance of threshold and post-gradient decisions. In this direction Luo et al. had combined a median filtering, an eight-direction Sobel operator, and iterative thresholding modeling for machine-vision defect edges²¹. Pandey et al. investigated the fuzzy-logic based edge decisions in comparison to simple empirical thresholding²². Tang et al. proposed an improved Sobel system for infrared target extraction for edge decisions²³, and Priyanka et al. reported a Sobel image-processing description and optimization for realization on FPGA²⁴. Most recently, Malik and Mirzaei demonstrated an optimized High Level Synthesis-based Canny system on FPGA, comprising of optimized gradient-direction and hysteresis processing²⁵.

The literature survey covers fixed Sobel pipelines architecture, adaptive Canny thresholds, statistical threshold approximation, fuzzy based decision logic, improved directional masks, and high processing capacity hysteresis/NMS techniques. Against these past events and records, the present Confidence-Guided Adaptive Sobel architecture fill up a deliberately small design point: it preserve the low-cost Sobel gradient, and introduces two thresholds to build three confidence classes, and performs a local support test only for the intermediate class. The projected merit is architectural inexpensiveness and a clear confidence gating.

III. System Experimental FPGA Architecture

A. System Organization

The system experimental organization with clock and memory structure of lightweight Confidence-Guided Adaptive Sobel implementation that targets the XC7Z020-1CLG484 device is shown in the Fig. 1. The image-processing placement and route operates in the 100-MHz system-clock domain. The mixed-mode clock manager (MMCM) creates a 25.175-MHz-class pixel clock for the VGA sub-system. Three

logical frame are used to hold the input image, baseline edge map, and a confidence-aware edge map. The synchronous dual-port memories isolate the processing accesses from reading the display.

The executional processing path is structured in a sequence of input frame buffer, 3×3 Sobel window, magnitude computation, pipeline register, confidence classifier, baseline/adaptive decisions, and output frame buffers in an indexed order. The display path reads the three memories in parallel and presents them as a trilogy on a 640×480 pixel grid screen.

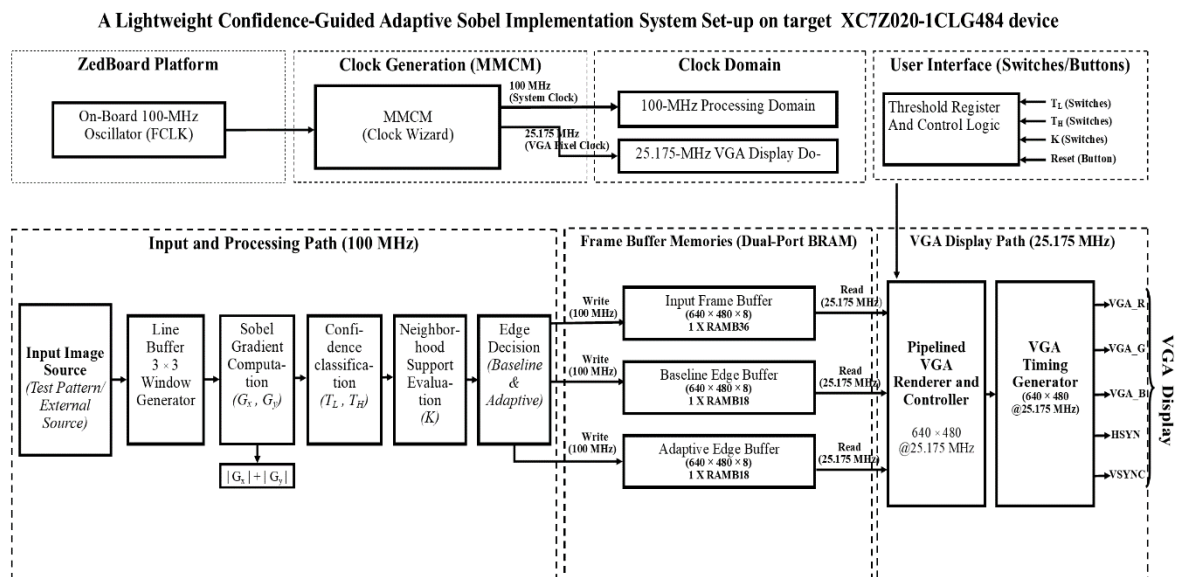


Fig.1 System experimental organization with clock and memory structure

B. Clocking and VGA Controller

The input system clock is 100 MHz, but the routed timing report recognizes an MMCM output period of 39.722 ns, corresponding to a 25.175 MHz clock. Hence, the behavioral monitoring is completed on 25.174825 MHz clock corresponds to 39.722222 ns period. Thus VGA controller is pipelined so that memory address generation, synchronous memory read latency, panel selection, and RGB synchronization alignment do not form a single long combinational path.

C. Memory and Arithmetic Structures

The routed architecture stands on one RAMB36 and two RAMB18 resources for frame storage without utilization of any DSP48 blocks. The absence of DSP blocks follows from the Sobel implementation is totally based on additions, subtractions, absolute values, and threshold comparisons. The confidence controller similarly consists of comparators, support counting, and Boolean decision logic.

D. FPGA Implementation Experimental Set-up

The system development and implementation is performed in Xilinx Vivado environment and integrated framework. Verification flow was divided into behavioral simulation and implementation signoff. The complete flow instantiates the system top-level design and sub-blocks integration, an assertion module, pixel-clock period, raster dimensions, horizontal and vertical synchronization, and any unknown values on the VGA outputs. A hard Tcl is designed that signoff gate blocks bitstream generation if the worst setup

slack is negative; at the same time a separate hold check requires nonnegative worst hold slack. The observatory conditional check was made on timing closure rather than successful placement and routing.

A 64x48 pixels image is used in the current test and the output result is displayed on 640x480 VGA frame. After completing the simulation and implementation signoff; the physical hardware validation was performed successfully using an external monitor. A photographic evidence shown in the Fig. 2, is reported for the experimental hardware setup and threshold settings. The validation criterion was functional and recorded as a stable VGA synchronization and rendering, the correct three-panel organization, and a visible configuration-dependent shifts in the baseline and confidence-aware edge maps.

IV. Results and Discussion

The behavioral simulation completed successfully and recorded in Table 1. The assertion monitor observed 1,679,998 pixel samples, 2,099 raster lines, and four frame starts. The average measured pixel period was 39.722222 ns, with observed extreme value of 39.722 and 39.723 ns, which is corresponding to 25.174825 MHz.

Table 1. Confidence-Guided Adaptive Sobel Architecture Behavioral Verification Report

Metric	Observed Values
System clock	100 MHz
Measured pixel lock	25.174825 MHz
Average pixel period	39.722222 ns
Observed period range	39.722 - 39.723 ns
Pixel samples	1,679,998
Raster lines checked	2,099
Frame starts observed	4
Assertion errors/warnings	0 / 0
Behavioral result	PASS

As shown in the Table 2, the post-synthesis timing did not initially meet the 100-MHz processing constraints. The WNS and TNS reporting was -0.443 ns and -2.069 ns respectively across eight failing setup endpoints. Moreover, after implementation, the routed design achieved the WNS = $+0.022$ ns and TNS = 0 ns with no failing setup endpoint. Hence, accordingly the worst hold slack remained positive and changing from $+0.097$ ns post-synthesis to $+0.064$ ns post-implementation. All user-specified timing constraints were reported as success.

Table 2. Comparison of Post-Synthesis and Post-Implementation Timing Results

Metric	Post-synthesis	Post-route
WNS (ns)	-0.443	+0.022
TNS (ns)	-2.069	0.000
Setup failing endpoints	8	0
WHS (ns)	+0.097	+0.064

THS (ns)	0.000	0.000
Hold failing endpoints	0	0
Constraint status	Not met	Met

The worst-path setup margin therefore improves by 0.465 ns relative to the post-synthesis estimate. The routed clock-domain summary shows that the 100-MHz system domain is the limiting domain (WNS +0.022 ns), whereas the VGA pixel-clock domain retains substantially larger setup margin. The positive final margin is small; therefore the result demonstrates timing closure but not a large guard band.

Table 3, reporting the hardware implementation optimization results and it shows that total LUT used were reduced by 64 cells (approximately 3.93%) while adding four flip-flops. No LUTRAM or DSP48 resource is utilized in the design.

Table 3. Hardware Resource Utilization Report for Post-Synthesis Vs. Post-Implementation

Resource	Post-synthesis	Post-route	Change in value
LUTs	1,627	1,563	-64(-3.93%)
FFs	1,280	1,284	+4
LUTRAM	0	0	0
RAMB36	1	1	0
RAMB18	2	2	0
DSP48	0	0	0
MMCM	1	1	0

A vector-less post-route power analysis estimates 0.254 W total on-chip power, comprising 0.145 W dynamic and 0.108 W device-static power, with an estimated junction temperature of 27.9 °C is listed in the Table 4.

Table 4. Report of Post-Route Estimated Power

Quantity	Value
Total on-chip power	0.254 W
Dynamic power	0.145 W
Device static power	0.108 W
Estimated junction temperature	27.9 °C
Clock power	0.005 W
Slice-logic power	0.006 W
Signal power	0.011 W
Block-RAM power	0.004 W
MMCM power	0.115 W
I/O power	0.005 W
Power confidence	Low (vector-independent)

The routed DRC reports clear the required implementation as correct default Zynq configuration. The presented demonstration is PL-centric and final bitstream was generated successfully after all the timing gates passed.

Following the successful timing closure and bitstream generation, the Confidence-Guided Adaptive Sobel design was programmed on a Zed-Board using the XC7Z020 device and connected to an external VGA monitor. The physical setup comprised the Vivado host used for design, implementation, and programming; the programmed Zed-Board; and the VGA display. The monitor produced the intended three-panel visualization. A complete hardware setup for system comprises the Vivado host ; an external VGA displaying the three-panel output: (a) input grayscale image, (b) conventional Sobel edge-detected baseline, and (c) confidence-aware adaptive threshold system with edge-detected output; and Zed-board for implementation is recorded in Fig. 2.

Threshold-control functionality was evaluated by changing configuration with the eight on-board switches. The representative recording was reported using different settings values on switch SW [7:0]. The processed edge maps changed with switch configuration while the input/reference panel remained visually unchanged. This provides direct physical evidence that the programmed threshold-control path affects the hardware result. These observations establish the qualitative functional validation of the hardware set-up; the quantitative image-quality superiority is not inferred in the photographs shown in Fig. 2.

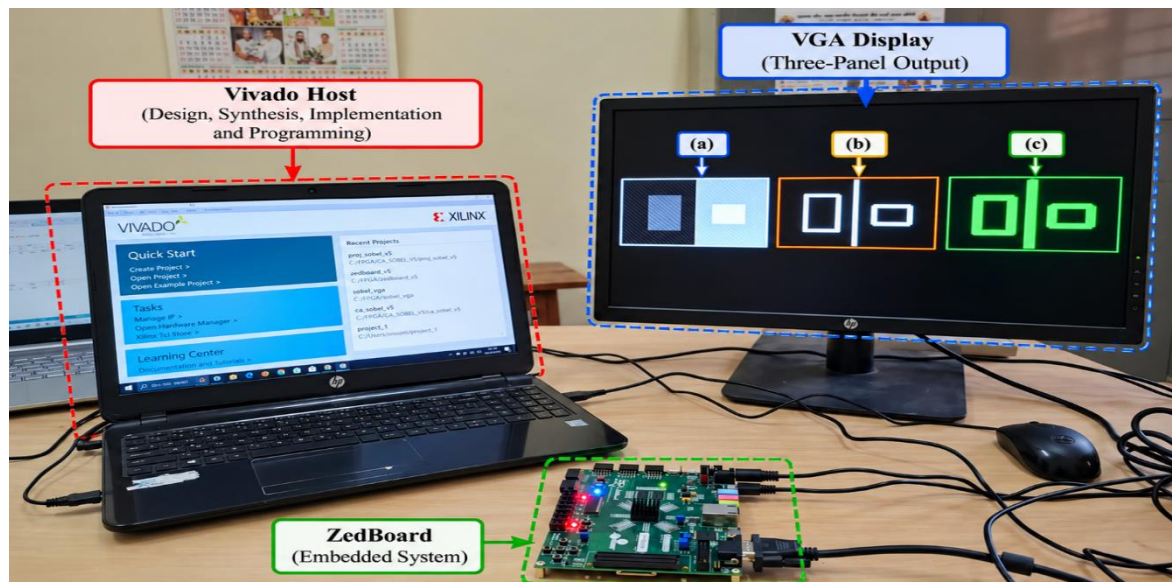


Fig. 2. Hardware validation setup and on-board VGA output of the proposed confidence-aware adaptive Sobel architecture

V. Conclusion

A confidence-aware adaptive Sobel edge-detection architecture has been implemented, timing-closed, and physically validated on a Xilinx Zynq-7000 XC7Z020 Zed-Board. The design combines a conventional Sobel gradient with three confidence classes and a local support condition for intermediate responses, while synchronous dual-port frame buffers and a pipelined VGA renderer provide a practical two-clock

hardware organization. Behavioral verification measured a 25.174825-MHz pixel clock with zero assertion errors over 1,679,998 pixel samples. Post-synthesis WNS of -0.443 ns was improved to $+0.022$ ns after implementation, with zero TNS and $+0.064$ -ns worst hold slack. The routed design uses 1,563 LUTs, 1,284 FFs, one RAMB36, two RAMB18 blocks, no DSP48 blocks, and one MMCM. A 0.254-W post-route power estimate is reported with the tool's Low-confidence qualification. The generated bitstream was successfully programmed on the Zed-Board, where stable VGA output displayed the input image, conventional Sobel baseline, and confidence-aware adaptive result simultaneously. The combined simulation, implementation, and board-level evidence therefore establishes end-to-end Confidence-Guided Adaptive Sobel implementation feasibility and functional operation.

REFERENCES:

1. G. N. Chaple and R. D. Daruwala, "Design of Sobel operator based image edge detection algorithm on FPGA," in Proc. Int. Conf. Communications and Signal Processing (ICCSP), 2014, doi: 10.1109/ICCSP.2014.6949951.
2. G. N. Chaple, R. D. Daruwala, and M. S. Gofane, "Comparisons of Robert, Prewitt, Sobel operator based edge detection methods for real time uses on FPGA," in Proc. Int. Conf. Technologies for Sustainable Development (ICTSD), Mumbai, India, 2015, pp. 1–4, doi: 10.1109/ICTSD.2015.7095920.
3. R. Jeyakumar, M. Prakash, S. Sivanantham, and K. Sivasankaran, "FPGA implementation of edge detection using Canny algorithm," in Proc. Online Int. Conf. Green Engineering and Technologies (IC-GET), Coimbatore, India, 2015, doi: 10.1109/GET.2015.7453798.
4. L. Yuan and X. Xu, "Adaptive image edge detection algorithm based on Canny operator," in Proc. 4th Int. Conf. Advanced Information Technology and Sensor Application (AITS), 2015, doi: 10.1109/AITS.2015.14.
5. H. C. Sateesh Kumar, S. Sarkar, and S. S. Bhairannawar, "An adaptive threshold based FPGA implementation for object and face detection," in Proc. Int. Conf. Image Information Processing (ICIIP), 2015, doi: 10.1109/ICIIP.2015.7414758.
6. B. You, W. Sheng, H. Ma, Y. Gu, and Y. Qin, "Implementation of Sobel edge detection on FPGA based on OpenCL," in Proc. IEEE 7th Annu. Int. Conf. CYBER Technology in Automation, Control, and Intelligent Systems, 2017, pp. 753–758, doi: 10.1109/CYBER.2017.8446103.
7. N. M. Yusoff, I. S. Abdul Halim, N. E. Abdullah, and A. A. Ab. Rahim, "Real-time Hevea leaves diseases identification using Sobel edge algorithm on FPGA: A preliminary study," in Proc. 9th IEEE Control and System Graduate Research Colloquium (ICSGRC), 2018, pp. 168–171, doi: 10.1109/ICSGRC.2018.8657603.
8. S. Yaman, B. Karakaya, and Y. Erol, "Real time edge detection via IP-Core based Sobel filter on FPGA," in Proc. Int. Conf. Applied Automation and Industrial Diagnostics (ICAAID), Elazig, Turkey, 2019, pp. 1–4, doi: 10.1109/ICAAID.2019.8934964.
9. P. Chen, Y. Wang, and J. Xing, "Research on design and verification of Sobel image edge detection based on high level synthesis," in Proc. 6th Int. Conf. Dependable Systems and Their Applications (DSA), 2019, pp. 491–492, doi: 10.1109/DSA.2019.00080.

10. P. Da Cunha Possa, S. A. Mahmoudi, N. Harb, and C. Valderrama, "A new self-adapting architecture for feature detection," in Proc. **22nd** Int. Conf. Field Programmable Logic and Applications (FPL), Oslo, Norway, 2012, pp. 643–646, doi: 10.1109/FPL.2012.6339230.
11. P. R. Possa, S. A. Mahmoudi, N. Harb, and P. Manneback, "A multi-resolution FPGA-based architecture for real-time edge and corner detection," IEEE Trans. Comput., vol. **63**, no. 10, pp. 2376–2388, Oct. 2014, doi: 10.1109/TC.2013.130.
12. Q. Xu, S. Varadarajan, C. Chakrabarti, and L. J. Karam, "A distributed Canny edge detector: Algorithm and FPGA implementation," IEEE Trans. Image Process., vol. **23**, no. 7, pp. 2944–2960, Jul. 2014, doi: 10.1109/TIP.2014.2311656.
13. J. Canny, "A computational approach to edge detection," IEEE Trans. Pattern Anal. Mach. Intell., vol. **PAMI-8**, no. 6, pp. 679–698, Nov. 1986, doi: 10.1109/TPAMI.1986.4767851.
14. N. Otsu, "A threshold selection method from gray-level histograms," IEEE Trans. Syst., Man, Cybern., vol. **9**, no. 1, pp. 62–66, Jan. 1979, doi: 10.1109/TSMC.1979.4310076.
15. Y. Liu, Z. Xie, and H. Liu, "An adaptive and robust edge detection method based on edge proportion statistics," IEEE Trans. Image Process., vol. **29**, pp. 5206–5215, 2020, doi: 10.1109/TIP.2020.2980170.
16. L. Q. Tan and B. He, "The research of implementation method of Canny edge detection of video on FPGA," in Proc. Int. Conf. Information Science, Parallel and Distributed Systems (ISPDS), 2020, pp. 1–4, doi: 10.1109/ISPDS51347.2020.00008.
17. S. Chen and X. Yang, "An enhanced adaptive Sobel edge detector based on improved genetic algorithm and non-maximum suppression," in Proc. China Automation Congress (CAC), 2021, pp. 8029–8034, doi: 10.1109/CAC53003.2021.9727626.
18. G. A. Gayathri and R. Ajai A. S., "VLSI implementation of improved Sobel edge detection algorithm," in Proc. Int. Conf. Communication, Control and Information Sciences (ICCISc), 2021, pp. 1–6, doi: 10.1109/ICCISc52257.2021.9485022.
19. X. Wei et al., "FPGA implementation of hardware accelerator for real-time video image edge detection," in Proc. IEEE **15th** Int. Conf. Anti-counterfeiting, Security, and Identification (ASID), Xiamen, China, 2021, pp. 16–20, doi: 10.1109/ASID52932.2021.9651710.
20. M. Hagara, P. Kubinec, A. Šatka, and R. Stojanović, "FPGA implementation of histogram-based thresholding," in Proc. **11th** Mediterranean Conf. Embedded Computing (MECO), 2022, doi: 10.1109/MECO55406.2022.9797132.
21. S. Luo, J. Hou, B. Zheng, X. Zhong, and P. Liu, "Research on edge detection algorithm of work piece defect in machine vision detection system," in Proc. IEEE **6th** Information Technology and Mechatronics Engineering Conf. (ITOEC), 2022, pp. 1231–1235, doi: 10.1109/ITOEC53115.2022.9734631.
22. A.K. Pandey, H. R. S. S. N. Chatla, M. Pandya, A. Farhan M. A., and A. S. Rana, "Image edge detection using fuzzy logic controller," in Proc. Int. Conf. Recent Advances in Electrical, Electronics & Digital Healthcare Technologies (REEDCON), New Delhi, India, 2023, pp. 508–513, doi: 10.1109/REEDCON57544.2023.10150762.
23. J. Tang, N. Liu, and H. Jian, "An edge detection method based on improved Sobel operator for infrared target extraction," in Proc. IEEE **5th** Int. Conf. Power, Intelligent Computing and Systems (ICPICS), 2023, pp. 119–123, doi: 10.1109/ICPICS58376.2023.10235635.



24. V. Priyanka, Y. S. Rama, K. Sravani, and B. Kavya, "Implementation of Sobel edge detection with image processing on FPGA," in Proc. 2nd World Conf. Communication & Computing (WCONF), 2024, pp. 1–5, doi: 10.1109/WCONF61366.2024.10692301.
25. S. H. K. Malik and S. Mirzaei, "An optimized FPGA based Canny edge detection system using high-level synthesis," in Proc. IEEE Int. Systems Conf. (SysCon), 2025, pp. 1–4, doi: 10.1109/SYSCON64521.2025.11014645.