Iranian Journal of Science and Technology, Transactions of Electrical Engineering
https://doi.org/10.1007/s40998-020-00333-5

RESEARCH PAPER

Adaptive Edge Detection Technique Implemented on FPGA

Saeid Taslimi[1] · Rasoul Faraji[1] · Ali Aghasi[1,2] · Hamid Reza Naji[1]

Received: 17 February 2019 / Accepted: 12 March 2020

Abstract

This article proposes an adaptive model that improves the edge detection operation in digital images. One of the disadvantages of traditional edge detection operators such as Prewitt, Sobel and Canny is using fixed-size masks which limit the edge detection operation in images with different degrees of the gray levels. In the proposed model, the image histogram is used as a criterion for selecting a suitable mask according to image specifications. In this model, the mask size of the edge detection operator is determined by measuring the contrast of the image to improve the quality of edge detection. For this purpose, a mask with a small size is used for images that have high contrast to increase processing speed, and for the images that do not have high contrast larger masks are used to increase the accuracy of edge detection. The proposed model is implemented utilizing real-time processing and parallelism capabilities of the Xilinx-Virtex6 FPGA. And, in order to reduce the effect of parallelism on increasing the area size of implemented hardware resources, existing resources are reused appropriately. The results show that with the use of the proposed method the edge detection is done in just 2.2 ms and a trade-off between the quality of detected edge and the hardware resources is established.

Keywords Image processing · Edge detection · Real-time processing · FPGA · Parallelism

1 Introduction

Edge detection is one of the most fundamental issues and plays an important role in image processing, which is done in order to extract the salient features of images. Also, it is one of the main tasks of low-level image processing which is directly related to performance, accuracy and runtime. So, with regard to the time-consumption of image processing due to massive data processing, optimum methods in processing and implementation should be utilized to achieve real-time processing (Khan et al. 2015; Li et al. 2015; Peker et al. 2016; Zolfaghari and Yazdi 2014; Khosravi and Momeni 2018; Saryazdi 2016; Roy and Pal 2019).

Derivative operators are simple edge detection operators which have different characteristics and detect different edges depending on the incoming input image. Sobel operator is one of these operators which operates based on differentiation. This operator uses to detect edges. Sobel operator is more applicable than other traditional edge detection operators because of the simplicity and higher precision (Hou et al. 2011). However, it has low accuracy in edge detection because of the use of two fixed masks in both horizontal and vertical directions. To increase its accuracy, a broader set of masks with different directions and larger sizes can be used (Burger and Burge 2009). Another problem of the conventional Sobel operator is the use of a fixed mask without considering the features of the image. To solve this problem, an appropriate mask can be selected by specifying a threshold according to specifications of different parts of the image.

As mentioned earlier, simple edge detection methods only use a fixed-size mask which reduces image quality. On the other hand, using multiple masks increase processing time because of increasing the complexity of computation and disable real-time processing especially by using sequential processors as an arithmetic logic unit (ALU) platform. But, with developing technology and manufacturing the advanced hardware description platforms such as FPGA with features like parallel processing

✉ Rasoul Faraji
R.Faraji@ec.iut.ac.ir

[1] Department of Electrical and Computer Engineering, Graduate University of Advanced Technology, Kerman, Iran

[2] Department of Computer Science, Shahid Bahonar University of Kerman, Kerman, Iran

Published online: 20 March 2020

and reconfiguration capability, executing complex computations have been feasible in the shortest time (Wang and McAllister 2016; Vourvoulakis et al. 2016; Sun et al. 2015; Amiri et al. 2017; Nausheen et al. 2018).

In this proposed method, the image histogram is calculated to evaluate the image features. Then according to the intensity distribution of image pixels, the mask size is selected. Then, the selected mask is called from hardware platform memory and placed in the processing path. To improve the execution time and implemented hardware resources, four masks sweep the image in a parallel way, and after sweeping each row, the same mask is used to sweep the next ones. Finally, the combined image and the desired results are displayed.

The rest of this paper is organized as follows: In Sect. 2, edge detection improvement methods and previous studies are presented. Using histogram as a threshold is described in Sect. 3. Section 4 explains the operation of the Sobel operator. Section 5 presents the proposed adaptive edge detection model. And finally, the conclusion is provided in Sect. 6.

2 Edge Detection Improvement Methods and Review of Literature

So far, many methods have been suggested to improve edge detection operators that can divide them into two general frames: improving processing methods and improving the way of executing and implementation. In this regard, some related works which are implemented in FPGA are investigated in this section.

Digital images sometimes contain noises (Goel and Aggarwal 2019), so before any processing operations, images should be denoised (Işik and Özkan 2016). In articles (Marsi and Ramponi 2013; Dandekar et al. 2007), a low-pass filter is used in the spatial domain to smooth and to reduce the noise. Hou et al. (2011) studies the performance of six different edge detection operators and concludes that the Sobel operator compared to other operators is more appropriate in terms of hardware resources for implementing on FPGA. In Jiang et al. (2018), in order to reduce the implementation complexity of the Steger edge detection algorithm, an optimized parallel-pipeline hardware structure is suggested. Through this hardware optimization, the implementation complexity of the algorithm is reduced and the timing performance is enhanced. Possa et al. (2014) tries to optimize the architecture of the Canny edge detection algorithm. In this paper, the hysteresis thresholding method with two different values (T_{low} and T_{high}) is used to define the threshold criteria for edge detection. T_{high} saturates every pixel with a gradient value greater than its threshold value, and also $\mathrm{T}_{\mathrm{low}}$ bypasses the pixel that has less value than T_{low} defines. In Qian et al. (2014), the Canny edge detection algorithm is used to improve the performance of the traditional Canny algorithm in smooth and high-detailed regions of the images. This algorithm computes the edge detection thresholds according to the black and the local distribution of the gradients in the image block adaptively. These thresholds are determined with a non-uniform gradient magnitude histogram. And in Singh et al. (2014) and Abbasi and Abbasi (2007), a new architecture for hardware implementation on FPGA is proposed. These works focus on improving the implementation of the Sobel operator, using hardware resources like using optimum memory addressing and also using extra flip-flops to speed up the process. Due to FPGA's inherent parallelism, and since the architectures of image processing have the potential to run in a parallel way, Hsiao et al. (2010) is improved the processing performance by parallel implementing the processing operators. Given that in image processing, especially in methods that use adaptive approaches during executing the program, many different hardware resources are used. Thus, in order to optimize the hardware resources, the FPGA reconfiguration feature is used in Quinn et al. (2007) and Kessal et al. (2008).

3 Histogram

The histogram is a graph which indicates the number of pixels in each image brightness level. In each digital image, pixel values represent the properties of the image. In literature, the histogram is used to select the threshold for edge detection (Sen and Pal 2010). It is used to identify the brightness level of different parts of the image to indicate the segments of an image. This method is known as image segmentation (Pare et al. 2019). In general, using the threshold criteria is one of the easiest methods to apply in the image segmentation method. But, automatic identification of threshold value is difficult and it is only suitable for images that have a large light intensity difference between the object and its background (i.e., the images which object is bright and the background is dark and vice versa) (Gonzalez and Woods 2008). Figure 1 shows an image with different contrasts and histograms.

4 Sobel Operator

There are many operators for edge detection, but most traditional and conventional methods use image derivative operators. These approaches are divided into two general categories, such as using the first order and second order of the derivative. Given that using the second-order derivative

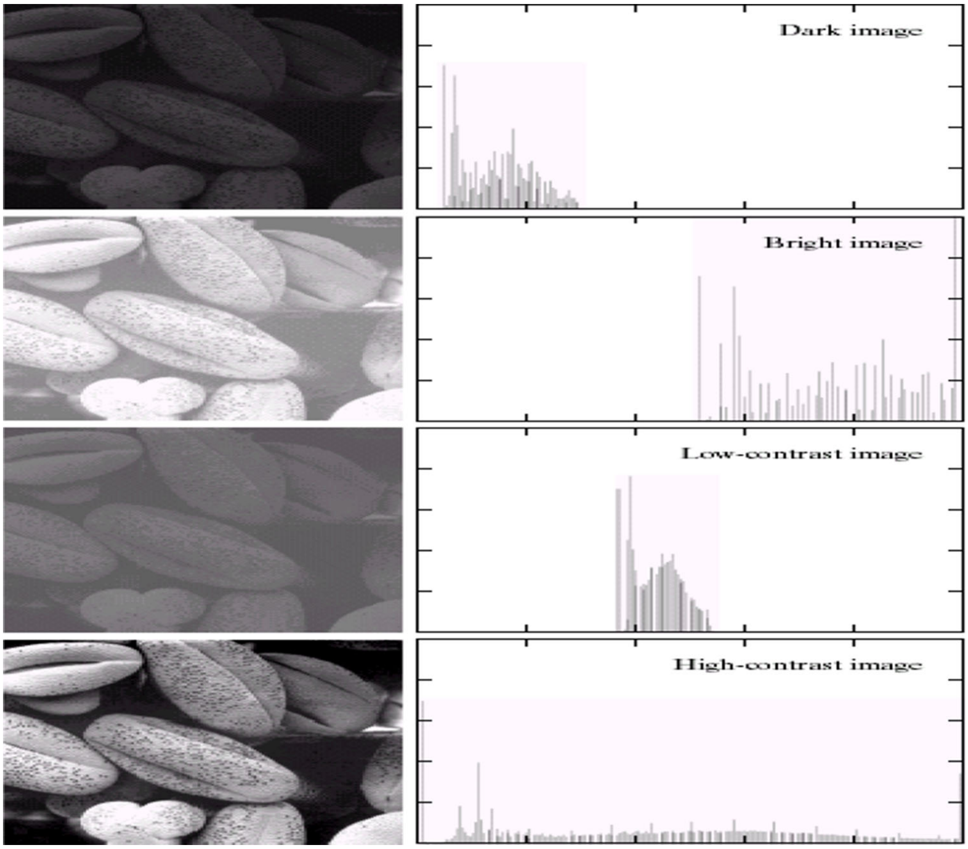


Fig. 1 An image with different contrasts and histograms (Gonzalez and Woods 2008)

Fig. 2 Sobel operator coefficient mask

2	1	0
1	0	-1
0	-1	-2

NW_SE

+45°

-1	0	1
-2	0	2
-1	0	1

E_W

Vertical

1	2	1
0	0	0
-1	-2	-1

N_S

Horizontal

0	1	2
-1	0	1
-2	-1	0

NE_ SW

-45°

has a lot of complex equations, methods based on first-order derivatives are used more frequently. One of the methods that uses first-order derivatives is the Sobel model. This model is not only simple, but it also gives a more accurate image than other operators of this group (Hou et al. 2011).

The traditional Sobel model only has two masks with vertical and horizontal directions (Yangli 2005; Sanduja and Patial 2012) which is not accurate for complicated edges, and it is sensitive to noise. Four masks can be used in 4 different directions to improve Sobel operator. Figure 2 shows four masks in four different directions.

In the following, Eqs. (1)–(4) are obtained according to the coefficient of masks in Fig. 3. So, the coordination of them would be achieved in four directions.

$$\text{Drive NE\_SW} = (\text{table }[0,1] + 2 * \text{table }[0,2] + \text{table}[1,2]) - (\text{table }[1,0] + 2 * \text{table}[2,0] + \text{table}[2,1]) \tag{1}$$

$$\text{Drive N\_S} = (\text{table }[0,0] + 2 * \text{table }[0,1] + \text{table}[0,2]) - (\text{table }[2,0] + 2 * \text{table}[2,1] + \text{table}[2,2]) \tag{2}$$

$$\text{Drive E\_W} = (\text{table }[0,2] + 2 * \text{table }[1,2] + \text{table}[2,2]) - (\text{table }[0,0] + 2 * \text{table}[1,0] + \text{table}[2,0]) \tag{3}$$

$$\text{Drive NW\_SE} = (\text{table }[1,0] + 2 * \text{table }[0,0] + \text{table}[0,1]) - (\text{table }[2,1] + 2 * \text{table}[2,2] + \text{table}[1,2]) \tag{4}$$

[0,0]	[0,1]	[0,2]
[1,0]	[1,1]	[1,2]
[2,0]	[2,1]	[2,2]

Fig. 3 Coordinates of Sobel operator coefficients

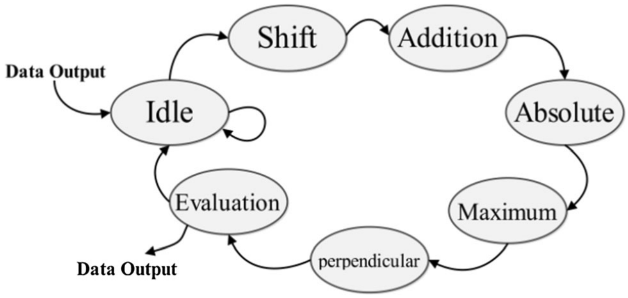


Fig. 4 Sobel state machine

[0,0]	[0,1]	[0,2]	[0,3]	[0,4]
[1,0]	[1,1]	[1,2]	[1,3]	[1,4]
[2,0]	[2,1]	[2,2]	[2,3]	[2,4]
[3,0]	[3,1]	[3,2]	[3,3]	[3,4]
[4,0]	[4,1]	[4,2]	[4,3]	[4,4]

Fig. 5 Coordinates of a 5 × 5 Sobel mask coefficient

And in the end, to accomplish the above tasks and to operate the edge detection process a state machine is designed which defines the main body of the designed operator (Fig. 4).

All the above equations have been explained for a 3 × 3 mask. For a 5 × 5 Sobel operator, the mask of coordinates of Fig. 5 and the mask of coefficients of Figs. 6 and 7 can be used.

Tables 1 and 2 show details of hardware resources implementations and the executing time of 3 × 3 and 5 × 5 edge detection models, respectively.

As illustrated, using a 5 × 5 mask increases processing time. But by considering more details of the image, it improves the quality of the image. The higher speed is attained by using a 3 × 3 mask, but in situations where the contrast of the image is low, edge detection would be difficult. In Fig. 8, the results of using a Sobel operator with a 3 × 3 mask and a 5 × 5 one are shown.

As can be seen, in some images, the image quality of 5 × 5 operator is much better than 3 × 3 operator and vice versa. Therefore, edge detection operator specification should be selected based on characteristics of the image, so that output image quality is acceptable.

1	2	0	-2	-1
4	8	0	-8	-4
6	12	0	-12	-6
4	8	0	-8	-4
1	2	0	-2	-1

Vertical

1	4	6	4	1
2	8	12	8	2
0	0	0	0	0
-2	-8	-12	-8	-2
-1	-4	-6	-4	-1

Horizontal

Fig. 6 Mask of vertical and horizontal coefficients

0	2	1	4	6
-2	0	8	12	4
-1	-8	0	8	1
-4	-12	-8	0	2
-6	-4	-1	-2	0

-45

6	4	1	2	0
4	12	8	0	-2
1	8	0	-8	-1
2	0	-8	-12	-4
0	-2	-1	-4	-6

+45

Fig. 7 Coefficients mask of + 45, − 45

Table 1 Hardware implementation details of a 3 × 3 edge detection model

Details	Used	Available	Utilization
Number of slices register	193	437,600	0%
Number of slice LUTs	372	218,800	0%
Number of fully used LUT FF pairs	105	1446	22%
Power dissipation (mW)	8.27		
Maximum frequency (MHz)	250.69		
Minimum period (nS)	3.989		
Execution time (mS)	8.323		

Table 2 Hardware implementation details of a 5 × 5 edge detection model

Details	Used	Available	Utilization
Number of slices register	417	437,600	0%
Number of slice LUTs	1260	218,800	0%
Number of fully used LUT FF pairs	231	1446	15%
Power dissipation (mW)	7.69		
Maximum frequency (MHz)	131.745		
Minimum period (nS)	7.59		
Execution time (mS)	12.435		

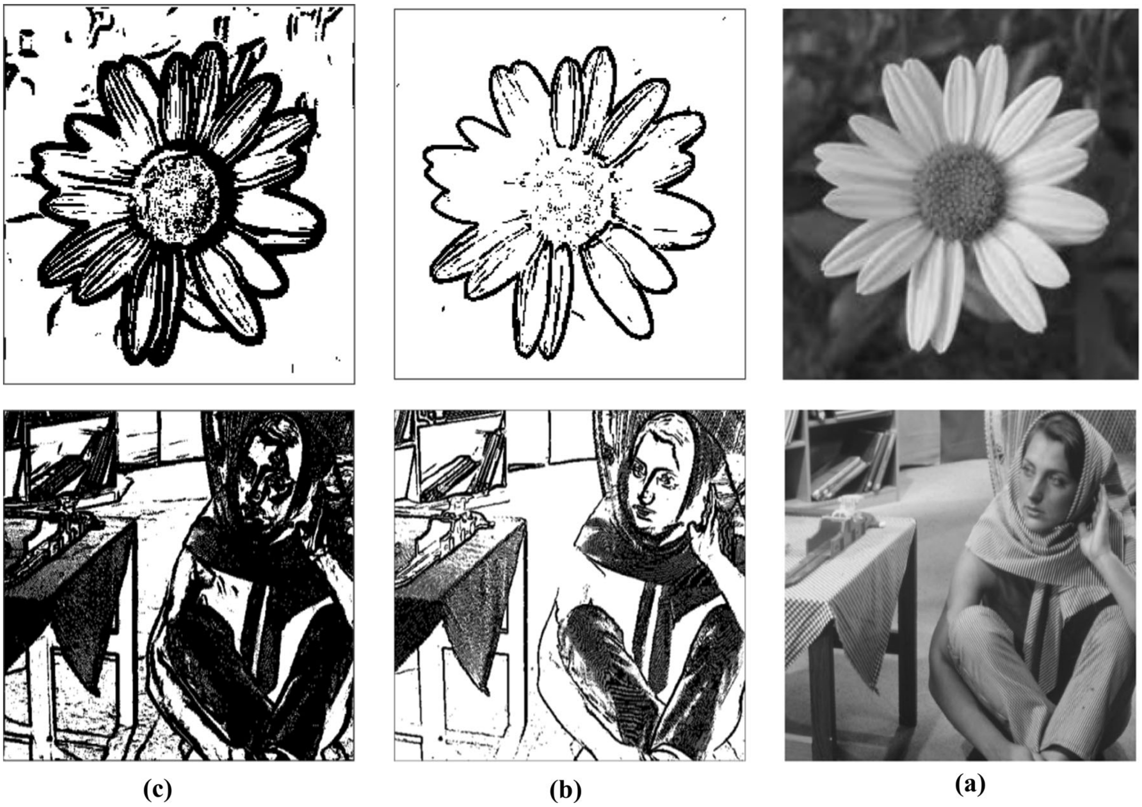
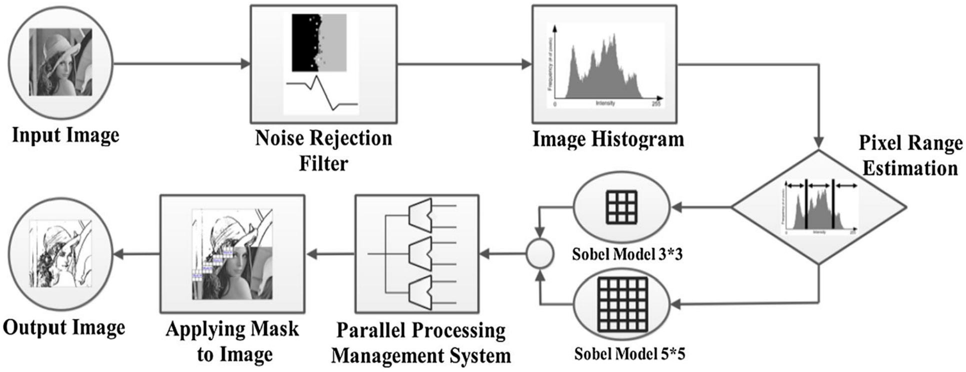


Fig. 8 **a** Main image **b** output with a 3 × 3 Sobel operator **c** output with a 5 × 5 operator

Fig. 9 The proposed model



5 Proposed Adaptive Edge Detection Model and the Results

Figure 9 shows the proposed model which is implemented on an FPGA. As illustrated, proposed model has noise removal, image histogram, comparator, 3 × 3 or 5 × 5 Sobel models selector and a parallel processing management system blocks.

In the proposed method, the image histogram is used as criteria to check the contrast of the image. And its results are used as a decision-maker to select the size of the desired mask to detect edges. To check the conditions of contrast, brightness levels should be divided into multiple parts. In this work, to define the appropriate range for brightness levels, the image histogram is averaged and divided into three parts (Table 3).

Table 3 Ranges of histogram

Part number	Lower side	Upper side
1	0	95
2	96	159
3	160	255

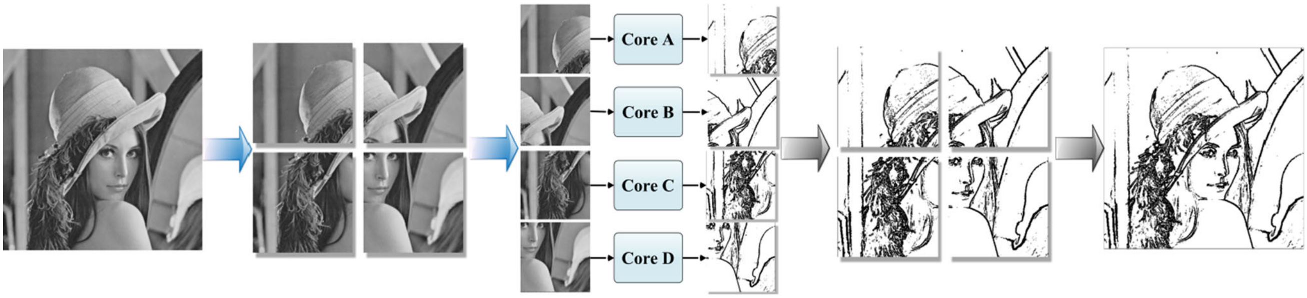


Fig. 10 Graphics of performing a parallel process

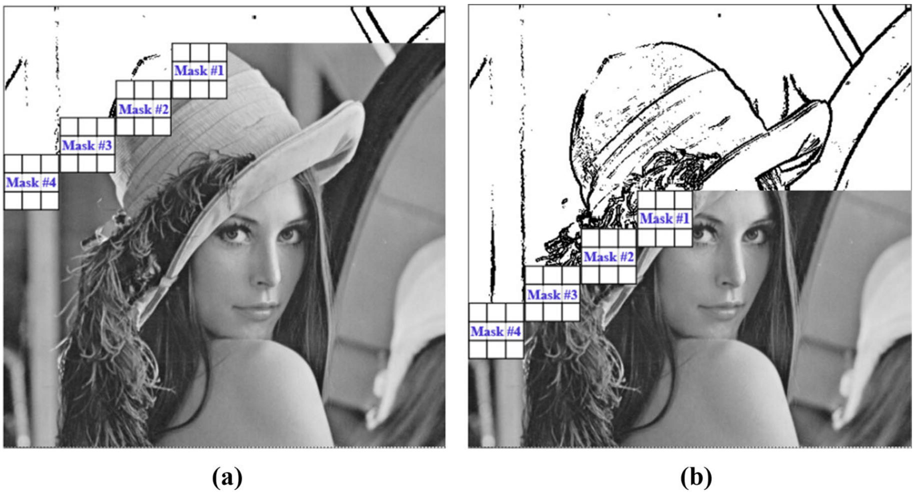


Fig. 11 Reusing masks in parallel processing **a** sweeping the first row by indicating masks, **b** sweeping the others rows by the same masks

Image contrast is measured, and after averaging, it is compared to the values of histogram. If the intensity of pixels is spread at all levels, the image has high contrast. And, if the pixels that are concentrated in a particular part of the image do not have proper contrast, a model with a larger mask size is needed to achieve accurate edge detection. And, if the contrast is high, proper edge detection with lower execution time would be available with a 3 × 3 mask. This can establish a trade-off between the accuracy and processing time.

The real-time processing plays an important role in image processing, and many software platforms do not have a good timing performance (Hong 2008). On the other hand, using two different-sized masks in the Sobel operator increases the hardware area. To overcome these problems, hardware platforms that have high processing speed and reconfiguration ability can be used. FPGA is a hardware platform which increases the processing speed by using descriptive hardware capability and parallel processing. Moreover, it can manage the area of the implemented hardware by utilizing reconfiguration capability. Using parallel processing can increase the processing speed, and it also gives the possibility to share implemented hardware resources, and therefore, hardware area can be managed well (Dammak et al. 2015; Stojilovic et al. 2013; Zuluaga and Topham 2009; Faraji and Naji 2014; Faraji et al. 2014). Thus, FPGA can be used as a platform for the proposed model.

Parallel processing capability can increase the processing speed. There are several ways to use parallel processing capability in image processing. One way is to divide the image into different parts and then use a separate processing core for each part (Zakerhaghighi and Naji 2013) (Fig. 10). Because a complete processing core is needed for each part, this method causes a severe increase in hardware area size. Thus, instead of implementing a complete processing core in each part, the proposed model applies only the masks of the processing core to the image in a parallel way to use the parallel processing capability and reuse the existing hardware resources. The operation of this method is shown in Fig. 11.

At first, four parallel masks are applied to the image and edge detection is performed for four rows; then, these four masks are used to sweep other rows and this process continues until the end of the last row of the image. With this method, not only the processing speed becomes almost four times better but also by reusing the existing hardware resources, the area of implemented hardware decreases dramatically.

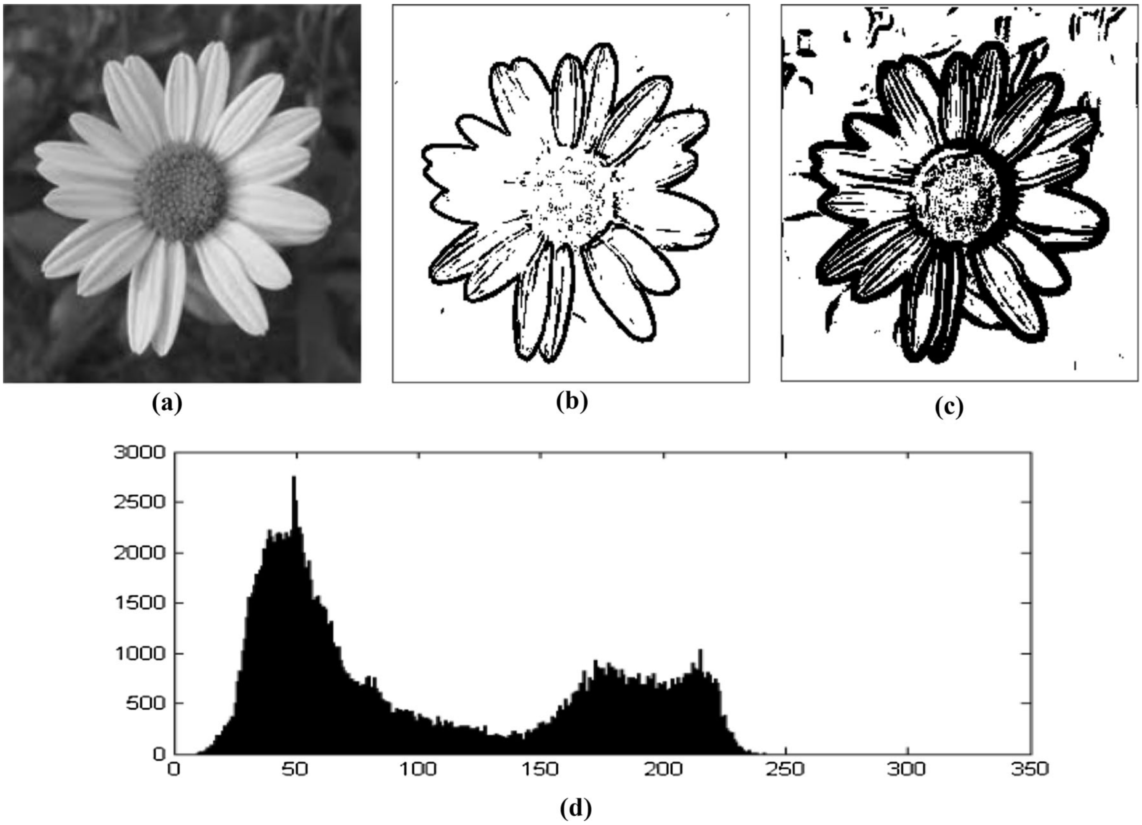


Fig. 12 **a** Original image, **b** image obtained from the 3 × 3 Sobel mask, **c** output of the proposed model, **d** histogram image

6 Results and Discussion

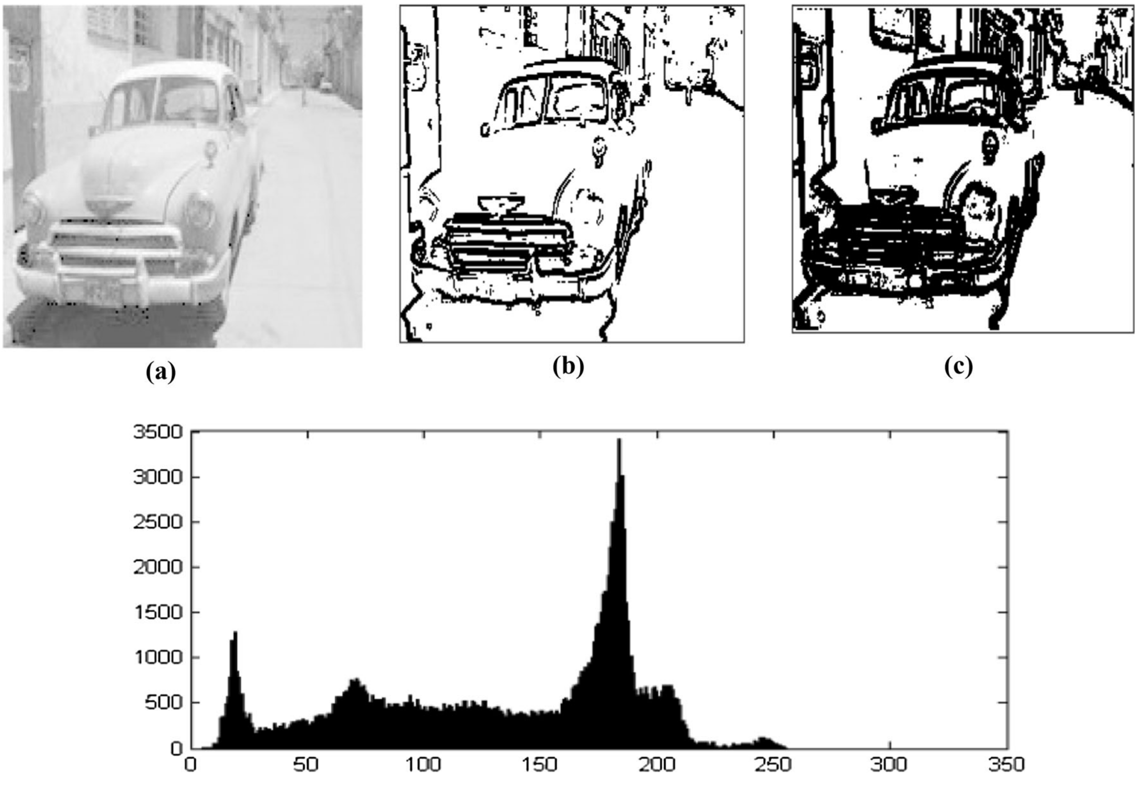
Figures 12, 13 and 14 compare the output of the proposed model with the traditional edge detection model with fixed masks in different sizes.

In Fig. 12, the average contrast goes in the first position of Table 3, and in Fig. 13, the average contrast goes in the third position of Table 3. Accordingly, for edge detection a 5 × 5 mask should be used. As illustrated, using a 3 × 3 mask shows fewer edge details and many edges are missed. But, edge detection with a 5 × 5 mask shows details as well.

In Fig. 14, due to the proximity of the average contrast of each part with the others, using a 5 × 5 mask has not improved the detected edge quality, and it has also made wide lines and spike noise in the image because of errors in edge detection. But, by using a 3 × 3 mask, better edge quality is obtained and also processing speed is increased. Using the proposed method when several consecutive frames of an image with different contrasts are received they can be detected the edges uniformly and can be removed unwanted changes in edge detection in consecutive frames of the image.

Table 4 shows the results of hardware resources and the execution time of the proposed model obtained from the synthesis. Hardware resources are increased negligibly compared to use a 5 × 5 Sobel model, but the processing time is 3.7 times less than a 3 × 3 Sobel model.

Another advantage of using the proposed adaptive method is establishing a trade-off between the size of the mask operator (based on the quality of the received image) and the amount of energy consumption. In Table 5, the proposed method and the conventional Sobel operator method in terms of energy consumption are compared. According to the results and by assuming that the proposed method operates the edge detection with probability of 50


(a)

(b)

(c)

(d)

Fig. 13 **a** Original image, **b** image obtained from the 3 × 3 Sobel mask, **c** output of the proposed model, **d** histogram image

percentage via the 5 × 5 mask size and with probability of 50 percentage via the 3*3 mask size, it can be seen that the proposed method compared with the conventional Sobel operator with a 5*5 mask size can improve energy consumption by 4.29%. And because of the parallelism reduces the execution time drastically, energy consumption of the proposed method compared with the conventional Sobel operator with 3 × 3 mask size, improved by 42.74%.

Since the synthesis tools determine the maximum operating frequency based on the maximum path delay, the maximum operating frequency of the proposed design is close to the conventional method with the 5 × 5 mask size. On the other hand, if it is required to use only the 3 × 3 mask size, the frequency can be increased. Using the proposed method, in addition to the size of the mask, can be determined. Also, the operating frequency can be changed so that the power consumption improves based on the image quality and processing time.

In Table 6, the proposed model is compared with some of the recent works in terms of hardware resources, power consumption and timing performance. As can be seen, the proposed model using parallelism has better execution time than most of the related works And without the use of parallelism, there is no significant increase in hardware resources. On the other hand, by determining the number of parallelism units, a trade-off between the execution time and hardware resources can be established.

Note that, in the design procedure of the proposed model the edge detection processor module is designed as an IP core so that it is a part of a comprehensive system that includes processors, memory and other parts which are connected together by the bus interconnect system. Synthesis results are presented in this article not included the memory blocks, and the designed structure for the proposed model is very similar to the structure which is introduced in Abbasi and Abbasi (2007).

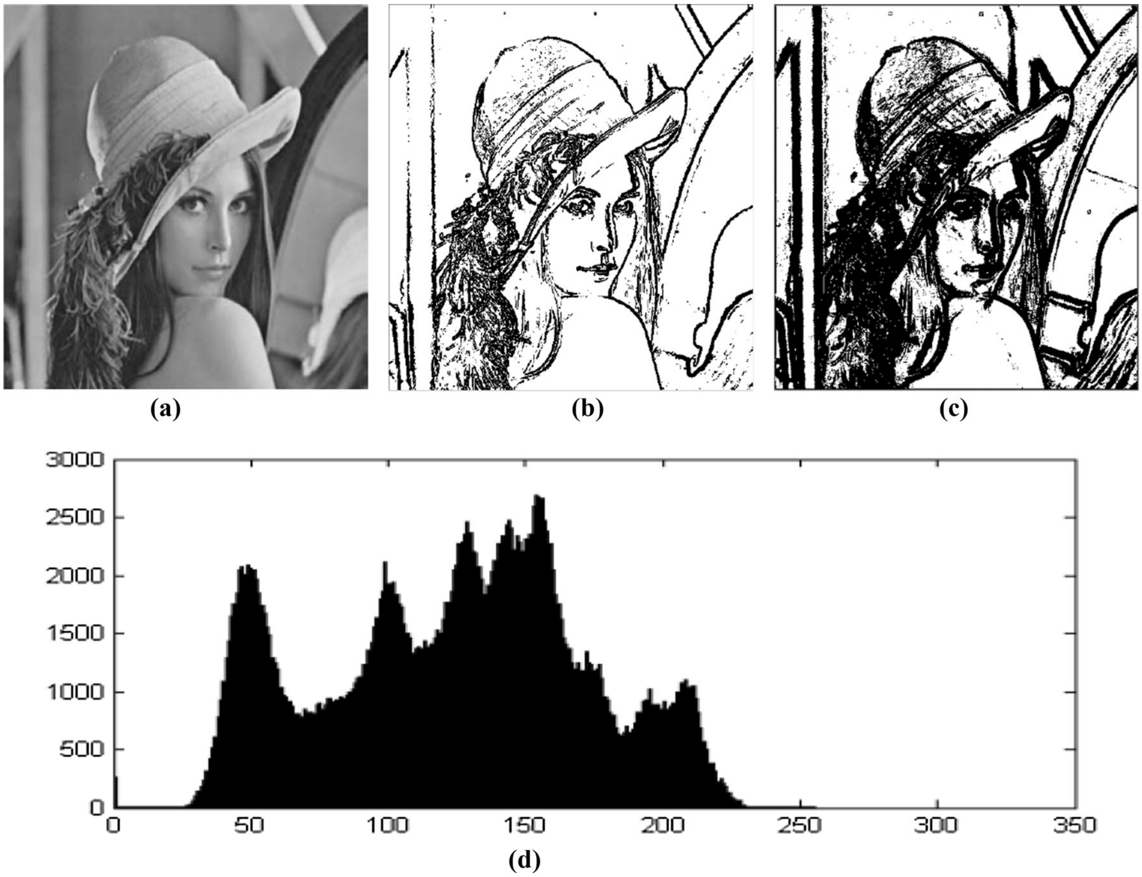


Fig. 14 **a** Original image, **b** output image of the designed model, **c** image obtained from Sobel model with a 5 × 5 mask, **d** histogram image

Table 4 Hardware resources implementation details of the proposed model on FPGA (Xc6vlx240t)

Details	Proposed model (without parallelism)			Proposed model (with parallelism)		
	Used	Available	Utilization	Used	Available	Utilization
Number of slices register	455	437,600	0%	1421	437,600	0%
Number of slice LUTs	1542	218,800	1%	3510	218,800	1%
Number of fully used LUT FF pairs	285	1446	20%	561	1446	38%
Instant power dissipation (mW)	8.45			14.31		
Maximum frequency (MHz)	129.45			126.812		
Minimum period (nS)	7.72			7.88		
Execution time of 3 × 3 (mS)	8.65			2.219		
Execution time of 5 × 5 (mS)	13.01			3.29		

Table 5 Energy comparison of the proposed model with conventional Sobel operator

	Conventional Sobel (3 × 3)	Conventional Sobel (5 × 5)	Proposed design without parallelization	Proposed design with parallelization
Maximum instant power dissipation (mW)	8.27	7.69	8.45	14.31
Maximum frequency (MHz)	250.69	131.745	126.812	129.45
Execution time	8.323	12.435	8.65 (Mask 3 × 3) 13.01 (Mask 5 × 5)	2.219 (Mask 3 × 3) 3.29 (Mask 5 × 5)
Energy (μJ)	68.83	95.62	73.09 (Mask 3 × 3) 109.93 (Mask 5 × 5)	31.75 (Mask 3 × 3) 47.07 (Mask 5 × 5)
Energy (μJ) (mask size selection probability = 50%)	–	–	91.51	39.41

Table 6 Comparison of proposed model with some related works

	Hou et al. (2011)	Jiang et al. (2018)	Singh et al. (2014)	Abbasi and Abbasi (2007)	Kessal et al. (2008)	Sanduja and Patial (2012)	Guo et al. (2010)	Kumar et al. (2017)	Chaple and Daruwala (2014)	Mehra and Verma (2012)	Zhang et al. (2018)	Proposed model (without parallelism)	Proposed model (with parallelism)
Number of slice	470	5714	119		–	1987	216	396	1685	71	481	455	1421
Number of slice flip-flops	–	–	–	328	–	836	289	–	–	482	–	–	–
Number of slice LUTs	–	4623	142	306	–	–	–	462	339	–	–	1542	3510
Number of fully used LUT FF pairs	–	–	177	–	–	–	–	315	–	–	–	285	561
Number of 4 input LUTs	–	–	–	–	–	3901	346	–	3166	353	–	–	–
Number of BRAMs	–	–	–	517	–	–	2	3	2	–	–	–	–
Power (mW)	–	–	–	32	–	–	–	–	–	103	–	8.45	14.31
Execution time	218 ms	2.62 ms	4.48 ms		36 ms	Size × 400 µs	21 ms	80.47 ms	6.412 ms		–	8.65 ms 13.01 ms	2.219 ms 3.29 ms
PIC size	–	512 × 512	512 × 512	2.65 ms	256 × 256	–	1024 × 1024	320 × 240	640 × 480	–	640 × 480	512 × 512	512 × 512
Maximum frequency (MHz)	–	100	117	512 × 512	12.5	200	50	276	48	148		129	126
FPGA model	Cyclone II EP2C70	Vertex 5 LX330	ML510	XC3S1500	Ardoise	XC4VLX200	XC3S200	xc7z020-1clg484	XC3S400	xc2vp30	CycloneII EP2C35F672C6	Xc6vlx240t	Xc6vlx240t

7 Conclusion

This paper proposes a new adaptive edge detection method to increase the accuracy of edge detection operator based on image characteristics and to improve the processing speed by considering the area of the hardware. This idea selects the size of the applied mask based on contrast to detect edges which can remove the lack of continuity in edge detection with small size masks by using larger masks which consider more details. On the other hand, it can improve the processing speed by using smaller masks in images with proper contrast. Finally, using parallel processing and share & reusing hardware resources in FPGA improve the performance of the proposed design such that the edges of the 512 × 512 image are detected in 2.2 ms and due to use a less amount of hardware resources just about 14 mW energy is consumed.

References

Abbasi T, Abbasi M (2007) A novel FPGA-based architecture for Sobel edge detection operator. Int J Electron 94(9):889–896

Amiri M, Siddiqui FM, Kelly C, Woods R, Rafferty K, Bardak B (2017) FPGA-based soft-core processors for image processing applications. J Signal Process Syst 87:1–18

Burger W, Burge MJ (2009) Digital image processing: an algorithmic introduction using Java. Springer, Berlin

Chaple G, Daruwala R (2014) Design of Sobel operator based image edge detection algorithm on FPGA. In: 2014 International conference on communication and signal processing (ICCSP). IEEE, pp 788–792

Dammak B, Baklouti M, Benmansour R, Niar S, Abid M (2015) Hardware resource utilization optimization in FPGA-based heterogeneous MPSoC architectures. Microprocess Microsyst 39(8):1108–1118

Dandekar O, Castro-Pareja C, Shekhar R (2007) FPGA-based real-time 3D image preprocessing for image-guided medical interventions. J Real Time Image Process 1(4):285–301

Faraji R, Naji HR (2014) An efficient crossover architecture for hardware parallel implementation of genetic algorithm. Neurocomputing 128:316–327

Faraji R, Rouholamini A, Naji HR, Fadaeinedjad R, Chavoshian MR (2014) FPGA-based real time incremental conductance maximum power point tracking controller for photovoltaic systems. IET Power Electron 7(5):1294–1304

Goel N, Aggarwal P (2019) Enhanced modified decision-based unsymmetric trimmed adaptive neighborhood winsorized mean filter for removing 1–99% levels of salt-and-pepper noise. Iran J Sci Technol Trans Electr Eng 43(3):459–468. https://doi.org/10.1007/s40998-019-00186-7

Gonzalez R, Woods R (2008) Digital image processing. Pearson prentice hall, Upper Saddle River

Guo Z, Xu W, Chai Z (2010) Image edge detection based on FPGA. In: 2010 Ninth international symposium on distributed computing and applications to business, engineering and science. IEEE, pp 169–171

Hong SFW (2008) An improved thread edge detection method based on Sobel algorithm. Microcomput Inf 1:111

Hou X, Shang Y, Liu H, Song Q (2011) Research on the real-time image edge detection algorithm based on FPGA, chapter 33. In: Shen G, Huang X (eds) Advanced research on computer science and information engineering, vol 153. Communications in computer and information science. Springer, Berlin, pp 200–206

Hsiao P-Y, Lu C-L, Fu L-C (2010) Multilayered image processing for multiscale Harris corner detection in digital realization. IEEE Trans Ind Electron 57(5):1799–1805

Işik Ş, Özkan K (2016) A new approach for edge detection in noisy images based on the LPGPCA technique. Turk J Electr Eng Comput Sci 24(4):2789–2805

Jiang J, Liu C, Ling S (2018) An FPGA implementation for real-time edge detection. J Real-Time Image Proc 15:787–797. https://doi.org/10.1007/s11554-015-0521-7

Kessal L, Abel N, Karabernou SM, Demigny D (2008) Reconfigurable computing: design methodology and hardware tasks scheduling for real-time image processing. J Real Time Image Process 3(3):131–147

Khan TM, Bailey D, Khan MA, Kong Y (2015) Real-time edge detection and range finding using FPGAs. Optik Int J Light Electron Opt 126(17):1545–1550

Khosravi I, Momeni M (2018) Presenting an object-based approach using image edges to detect building boundaries from high spatial resolution images. Iran J Sci Technol Trans Electr Eng 42(1):95–105. https://doi.org/10.1007/s40998-018-0051-y

Kumar V, Asati A, Gupta A (2017) Hardware implementation of a novel edge-map generation technique for pupil detection in NIR images. Eng Sci Technol Int J 20:694–704

Li Y, Wang S, Tian Q, Ding X (2015) A survey of recent advances in visual feature detection. Neurocomputing 149:736–751

Marsi S, Ramponi G (2013) A flexible FPGA implementation for illuminance–reflectance video enhancement. J Real Time Image Process 8(1):81–93

Mehra R, Verma R (2012) Area efficient FPGA implementation of Sobel edge detector for image processing applications. Int J Comput Appl 56(16):7–11

Nausheen N, Seal A, Khanna P, Halder S (2018) A FPGA based implementation of Sobel edge detection. Microprocess Microsyst 56:84–91. https://doi.org/10.1016/j.micpro.2017.10.011

Pare S, Kumar A, Singh GK, Bajaj V (2019) Image segmentation using multilevel thresholding: a research review. Iran J Sci Technol Trans Electr Eng. https://doi.org/10.1007/s40998-019-00251-1

Peker M, Altun H, Karakaya F (2016) Hardware implementation of a scale and rotation invariant object detection algorithm on FPGA for real-time applications. Turk J Electr Eng Comput Sci 24(5):4368–4382

Possa PR, Mahmoudi SA, Harb N, Valderrama C, Manneback P (2014) A multi-resolution FPGA-based architecture for real-time edge and corner detection. IEEE Trans Comput 63(10):2376–2388. https://doi.org/10.1109/tc.2013.130

Qian X, Varadarajan S, Chakrabarti C, Karam LJ (2014) A distributed canny edge detector: algorithm and FPGA implementation. IEEE Trans Image Process 23(7):2944–2960. https://doi.org/10.1109/tip.2014.2311656

Quinn H, Leeser M, King LS (2007) Dynamo: a runtime partitioning system for FPGA-based HW/SW image processing systems. J Real Time Image Process 2(4):179–190

Roy S, Pal AK (2019) A hybrid domain color image watermarking based on DWT–SVD. Iran J Sci Technol Trans Electr Eng 43(2):201–217. https://doi.org/10.1007/s40998-018-0109-x

Sanduja V, Patial R (2012) Sobel edge detection using parallel architecture based on FPGA. Int J Appl Inf Syst 3(4):20–24

Saryazdi S (2016) An optimal recursive step edge detector. Iran J Sci Technol Trans Electr Eng 40(1):35–44. https://doi.org/10.1007/s40998-016-0006-0

Sen D, Pal SK (2010) Gradient histogram: thresholding in a region of interest for edge detection. Image Vis Comput 28(4):677–695. https://doi.org/10.1016/j.imavis.2009.10.010

Singh S, Saini AK, Saini R, Mandal A, Shekhar C, Vohra A (2014) A novel real-time resource efficient implementation of Sobel operator-based edge detection on FPGA. Int J Electron 101(12):1705–1715

Stojilovic M, Novo D, Saranovac L, Brisk P, Ienne P (2013) Selective flexibility: creating domain-specific reconfigurable arrays. IEEE Trans Computer-Aided Des Integr Circuits Syst 32(5):681–694

Sun X, Shen Y, Li W, Wu H (2015) A PWM and PFM hybrid modulated three-port converter for a standalone PV/battery power system. IEEE J Emerg Sel Topics Power Electron 3(4):984–1000

Vourvoulakis J, Kalomiros J, Lygouras J (2016) Fully pipelined FPGA-based architecture for real-time SIFT extraction. Microprocess Microsyst 40:53–73

Wang P, McAllister J (2016) Streaming elements for FPGA signal and image processing accelerators. IEEE Trans Very Large Scale (VLSI) Integr Syst 24(6):2262–2274. https://doi.org/10.1109/tvlsi.2015.2504871

Yangli Y (2005) Study of FPGA based parallel processing of Sobel operator. AI Modern Electronics Technique

Zakerhaghighi MR, Naji HR (2013) Whale algorithm for image processing, a hardware implementation. In: 2013 8th Iranian conference on machine vision and image processing (MVIP). IEEE, pp 355–359

Zhang K, Zhang Y, Wang P, Tian Y, Yang J (2018) An improved Sobel edge algorithm and FPGA implementation. Procedia Comput Sci 131:243–248. https://doi.org/10.1016/j.procs.2018.04.209

Zolfaghari M, Yazdi M (2014) A new edge-preserving algorithm based on the CIE-Lu’v’color space for color contrast enhancement. Turk J Electr Eng Comput Sci 22(4):990–1006

Zuluaga M, Topham N (2009) Design-space exploration of resource-sharing solutions for custom instruction set extensions. IEEE Trans Computer-Aided Des Integr Circuits Syst 28(12):1788–1801