FPGA IMPLEMENTATION OF DISTRIBUTED CANNY EDGE DETECTOR

B Madhu
Research Scholar Dept. of ECE (VLSI)
Vignan Institute Of Technology And Science
Deshmukhi,Hyderabad
bmadhu.408@gmail.com

Mr. Vijaykumar R. Urkude,(Ph.D)
Associate Professor, Dept of ECE
Vignan Institute Of Technology And Science
Deshmukhi,Hyderabad
cool.viju1721@gmail.com

***ABSTRACT*- Edges represent the significant local changes in the intensity of an image. Image Edge detection significantly reduces the amount of data and filters out useless information, while preserving the important structural properties of an image. A distributed canny edge detection algorithm is used which results in the reduced memory requirements, decreased latency and increased throughput with no loss in the performance of the edge detection, when compared to that of the original Canny algorithm. An FPGA-based hardware architecture of the proposed algorithm is synthesized on Xilinx SPARTEN-3 EDK.**

Keywords:FPGA,CANNY EDGE DETECTION,

CANNY ALGORITHM.

1.INTRODUCTION

Edge detection is the process of identifying and locating the discontinuities present in an image are the immediate changes in pixel intensity which characterize boundaries of an objects in a image. It is widely used in image analysis, make use of algorithms for enhancing and detecting the edges. An edge is the boundary between an object and the image. It is widely used in image analysis, make use of algorithms for enhancing and detecting the edges An edge is the boundary between an object and the background, and indicates the boundary between overlapping objects.

Edge detection is a very important area in the field of Computer Vision. Edges define the boundaries between the regions in an image, which helps with segmentation and object recognition. An edge detector accepts a digital image as input and produces an edge map as output. The quality of edge detection is highly dependent on lighting conditions, the presence of objects with the similar intensities, density of edges in the image, and noise.The majority of the methods may be grouped into two categories:

Gradient: The gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image.

Laplacian: The Laplacian method searches for zero crossings in the second derivative of the image to find edges.

The edge representation of an image drastically reduces the amount of data to be processed,but still it retains the important information about the shapes of objects.this description of image is easy to integrate into a large number object recognition algorithms used into a large number of object recognition algorithms used in computer vision and other image processing application.An important property of the edge detection method is its ability to extract the accurate line with good orientation.

The rest of this paper is arranged as follows:section 2 gives a brief introduction of the edge detection methods,section3presentscanny edge detector, ection 4 gives the brief methodology,section 5 shows simulation results and section 6 gives the conclusions of this paper.

II.EDGE DETECTION METHODS

There are many edge detection methods but the most Commonly used are the classical methods and Gaussian methods.

Classical edge detectors have no smoothing filter, and they are only based on a discrete differential operator.The earlier algorithms developed by Roberts, Sobel and Prewitt computes an estimation of gradient for the pixels, and look for local maxima to localize step edges. Typically, they are simple in computation and capable to detect the edges and their orientation, but due to lack of smoothing stage, they are very sensitive to noise and inaccurate.

***A.* Roberts cross operator:**

The Roberts cross operator is used in the image processing for edge detection. It was the first edge detector and was initially proposed by Lawrence Roberts in 1963.

According to Roberts, the edge detector should have the following properties:

- The produced edges should be well-defined.
- noise should be as low as possible.
- Intensity of edges should be closely related to that of a original image.

Roberts Cross operator is very easy and quick to compute. There are no additional parameters to set,and its main limitation is that it uses a small kernel, which is very sensitive to noise and produces weak responses to genuine edges.

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

$Gx\ \ mask$

1	2	1
0	0	0
1	2	1

$Gy\ mask$

Fig1: Roberts cross convolution kernels Gradient magnitude is given by

Canny convolution kernels gradient magnitude is given by

$$|G| = \sqrt{G_x^2 + G_y^2}$$

Gradient direction is given by

$$\theta = a\tan\left(\frac{G_y}{G_x}\right)$$

B. Sobel edge operator:

The Sobel edge operator used in image processing, convolves the image with a small filter in horizontal and vertical directions and is therefore relatively inexpensive in terms of computations.

Mathematically, the operator uses two 3×3 kernels which are convolved with the original image to calculate approximations of the derivatives one for horizontal changes, and one for vertical.

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

1	2	1
0	0	0
1	2	1

Fig2:sobels cross convolution kernels Gradient magnitude is given by

Canny convolution kernels gradient magnitude is given by

$$|G| = \sqrt{G_x^2 + G_y^2}$$

Gradient direction is given by

$$\theta = a \tan\left(\frac{G_y}{G_x}\right)$$

C**. Prewitts edge operator:**

The Prewitt operator used in image processing,helps in computing an approximation of the gradient of the image intensity function. Mathematically, the operator uses two 3×3 kernels which are convolved with the original image to calculate approximations of the derivatives one for horizontal changes, and one for vertical responses to genuine edges.

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

$Gx\ \ mask$

1	2	1
0	0	0
1	2	1

$Gy\ mask$

Fig3: Prewitts cross convolution kernels Gradient magnitude is given by

Canny convolution kernels gradient magnitude is given by

$$|G| = \sqrt{G_x^2 + G_y^2}$$

Gradient direction is given by

$$\theta = a \tan\left(\frac{G_y}{G_x}\right)$$

Gaussian filter is a filter whose impulse response is a Gaussian function. Gaussian filters are the most widely used filters in image processing and extremely useful as detectors for edge detection. There are advantages for the Gaussian filter that make it unique and so important in edge detection they are:

- Gaussian smoothening is very effective for
- removing Gaussian noise.
- It reduces edge blurring.
- They are linear low pass filters.

III. CANNY EDGE DETECTOR

Canny edge detector which is also well known as an optimal edge detector is one of the most efficient and successful edge detection methods. It operates on the gray-scale version of the image under consideration.

He considered three criteria desired for any edge detector:

• Good detection: The algorithm should mark as many real edges as possible in the image.

• Good localization: Edges marked should be as close as possible to the edges in the real image.

• Minimal response: A given edge in the image should only be marked once and noise should not create false edges.responses to genuine edges.

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

$Gx\ \ mask$

1	2	1
0	0	0
1	2	1

$Gy\ mask$

Fig4: Canny convolution kernels Gradient magnitude is given by

Canny convolution kernels gradient magnitude is given by

$$|G| = \sqrt{G_x^2 + G_y^2}$$

Gradient direction is given by

$$\theta = a \tan\left(\frac{G_y}{G_x}\right)$$

The effect of the Canny operator is determined by three parameters:

- Width of the Gaussian kernel used in the smoothing stage, and the upper and lower thresholds used by the edge tracker.

- By increasing the width of the Gaussian kernel,reduces the edge detector's sensitivity to noise
- The localization error in the detected edges increases as the Gaussian width is increased.

IV. CANNY EDGE DETECTOR

The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images, and it is regarded as a near optimal edge detection technique because it produces reliable, thin edges even in the presence of noise in the image. It is developed by John Canny in 1986.

The Canny algorithm has 5 separate steps:

- Smoothing: removal of noise from image
- Gradient magnitude and direction: the image having larger agnitudes should be marked as an edge.
- Non-maximum suppression: Only local maxima should be marked as edges.
- Double thresholding: strong edges are determined by thresholding.
- Edge tracking by hysteresis: Final edges are determined by suppressing all the edges that are not connected to strong edges.

There are 2 types of canny edge detector:

D. Traditional canny edge detector

Canny algorithm gives good edge detection results when compared to other edge detectors. But Canny algorithm depend on a correct setting of the threshold. Threshold setting affects the edge detecting results greatly. In a canny edge detector it is difficult to set the threshold to a proper value for a given image, and it is usually done by the manual process.

As a result we miss some edges or detect some spurious edges when the threshold is not set to a proper value. Therefore, these algorithms do not suit for robot vision systems. Another limitation of a commonly used Canny algorithm is the computational cost of the algorithm which is very high and it cannot be implemented in real time to meet the needs of robot vision system. To overcome these limitations of a canny edge detector, a distributed canny edge detector is proposed.

E. Distributed canny edge detector

In this algorithm, the threshold is calculated by the histogram of gradient magnitude. This improved Canny algorithm is also implemented on an FPGA to meet the needs of real time processing, compared with the Canny edge detector, the distributed canny edge detector can give good edge detection results even when there is large envirormental changes. The implementation of the improved algorithm on FPGA has also reduced the implantation time to a large extent so that it can be used for the robot vision system.

Benefits of distributed canny edge detector:

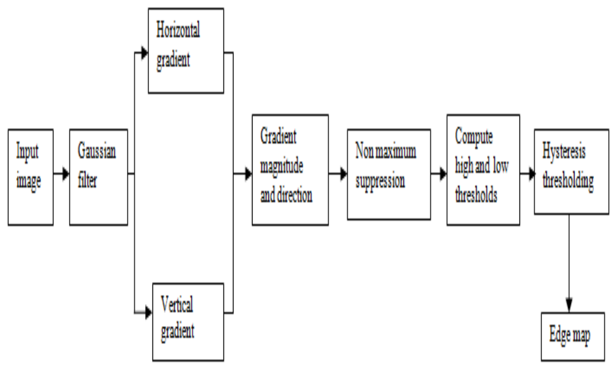
- An increase in the throughput when multiple core architectures are available
- A decrease in expensive on-chip memory for a fixed throughput, and

The possibility of pipelining the Canny edge detector with other image processing modules of an application.

The distributed Canny algorithm contains a number of adjustable parameters which can effect the computation and effectiveness of the algorithm The size of the Gaussian filter: The smoothing filter affects the results of the Canny algorithm. Smaller filters cause less blurring, and allow detection of small, sharp lines. A larger filter causes more blurring, and loosing out the value of a given pixel over a larger area of the image.

Thresholds: The use of two thresholdswith hysteresis allows more flexibility than in a single-threshold approach. A threshold set too high can miss important information. On the other hand, a threshold set too low will identify irrelevant information.

V. METHODOLOGY



A block diagram of the Canny edge detection algorithm is shown in Fig.5

The original Canny algorithm consists of the following steps:

F. Noise reduction

Canny edge detector filters out any noise present in the original image before trying to locate and detect any edges. Gaussian filter can be computed using a simple mask, it is used exclusively in the Canny algorithm. Once a suitable mask has been calculated, the Gaussian smoothing can be performed. A convolution mask is usually much smaller than the actual image. As a result, the mask is slid over the image, manipulating a square of a pixels at a time.

G. Gradient magnitude and direction

The gradient of the image is one of the fundamental building blocks in image processing. The gradient is a vector which has magnitude and direction. The gradient of an image measures how it is changing. It provides two pieces of information .The magnitude of the gradient tells us how quickly the image is changing, while the direction of the gradient tells us the direction in which the image is changing most rapidly.

The canny operator uses a pair of 3x3 convolution masks, one estimating the gradient in the x-direction (columns) and the other estimating the gradient in the y-direction (rows). The Gx and Gy masks will estimate gradient in x direction and y direction respectively. After smoothing the image and eliminating the noise, the next step is to find the gradient magnitude of the image. The gradient magnitude is given by

$$|G| = \sqrt{G_x^2 + G_y^2}$$

Gx is the gradient in x direction.

Gy is the gradient in y direction.

Once the edge direction is known, the next step is to relate the edge direction to a direction that can be traced in an image.

x x x x x

x x x x x

x x c x x

x x x x x

x x x x x

By looking at the centre pixel "c", there are four possible directions:

0 degrees (in the horizontal direction),

45 degrees (along the positive diagonal),

90 degrees (in the vertical direction), or

135 degrees (along the negative diagonal),

Therefore the edge direction calculated will be rounded up to the closest angle possible.

H. Non maximum suppression

This step converts the “blurred” edges in the image to “sharp” edges. It is used to determine if the gradient magnitude assumes a local maximum in the gradient direction. If the rounded angle is zero degrees then the point will be considered to be on the edge if its intensity is greater than the intensities in the north and south directions, if the rounded angle is 90 degrees then the point will be considered to be on the edge if its intensity is greater than the intensities in the east and west directions, if the rounded angle is 135 degrees then the point will be considered to be on the edge if its intensity is greater than the intensities in the north east and south west directions, if the rounded angle is 45 degrees then the point will be considered to be on the edge if its intensity is greater than the intensities in the south east and north west directions.

I. Double thresholding

The pixels remaining after the non-maximum suppression are marked with their strength pixel-by pixel.Many of these are the true edges in the image,but some are caused by noise or color variations for instance due to rough surfaces. The simplest way to decide between these variations would be to use a threshold, so that only edges stronger than a certain value would be preserved. The Canny edge detection algorithm uses double thresholding. Edge pixels stronger than the high threshold are marked as strong;edge pixels weaker than the low threshold are suppressed and edge pixels between the two thresholds are marked as weak.

J. Edge tracking by hysteresis

Thresholding with hysteresis requires two thresholdsi.e high and low. Strong edges can be immediately included in the final edge image. Weak edges will be included if and only if they are connected to strong edges. Strong edges will be present only due to true edges in the original image. The weak edges can be due to true edges or noise or colour variations.

V.EXPERIMENTAL RESULTS

Experiments are performed on gray level images to verify the proposed method. These images are represented by 8bits/pixel and size is 128 x 128. Image used for experiments are shown in below figure.

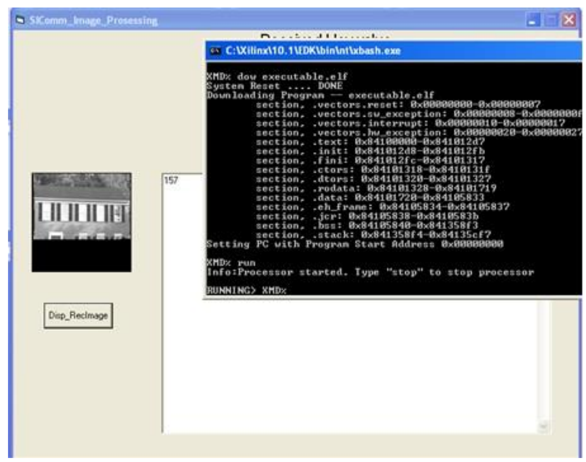


Fig.6input image

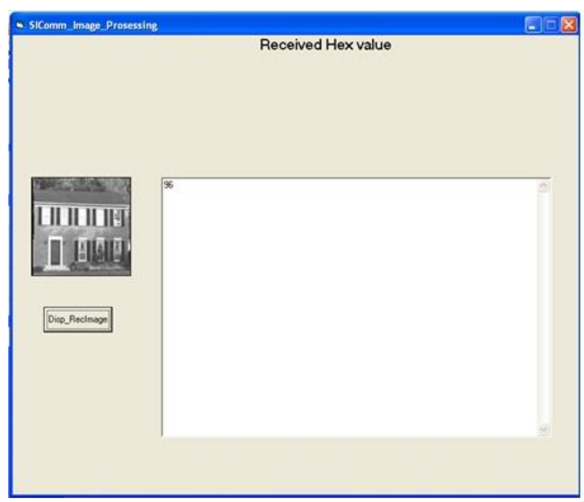


Fig.7 display of image

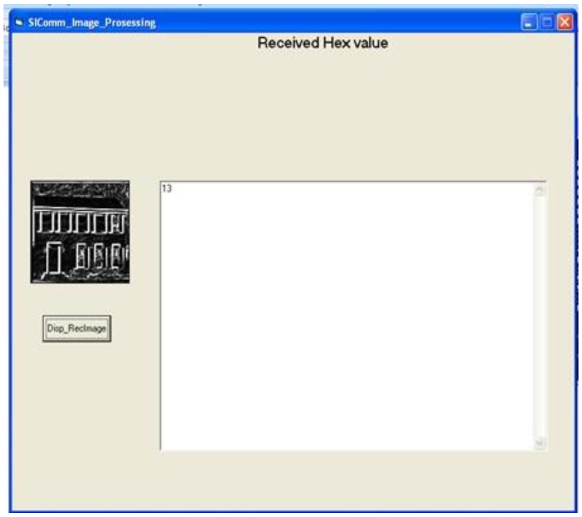


Fig.8 edge detection image

And the synthesis report is below

```
Selected Device : 3s200tq144-4

 Number of Slices:                    1880  out of   1920    97%
 Number of Slice Flip Flops:          2118  out of   3840    55%
 Number of 4 input LUTs:              2971  out of   3840    77%
    Number used as logic:             2418
    Number used as Shift registers:    297
    Number used as RAMs:               256
 Number of IOs:                         62
 Number of bonded IOBs:                 62  out of     97    63%
    IOB Flip Flops:                     64
 Number of BRAMs:                        4  out of     12    33%
 Number of MULT18X18s:                   3  out of     12    25%
 Number of GCLKs:                        4  out of      8    50%
 Number of DCMs:                         1  out of      4    25%
```

Timing report:

```
Speed Grade: -4

   Minimum period: 15.304ns (Maximum Frequency: 65.342MHz)
   Minimum input arrival time before clock: 6.569ns
   Maximum output required time after clock: 16.181ns
   Maximum combinational path delay: No path found
```

Fig. 7 Synthesis report

VI. CONCLUSIONS

A distributed Canny edge detection algorithm presented in this paper results in a significant speed up without sacrificing the edge detection performance. A novel non uniform quantized histogram calculation method is proposed in order to reduce the computational cost of the hysteresis threshold selection. As a result, the computational cost of the proposed algorithm is very low compared to the original Canny edge detection algorithm. The algorithm is mapped to onto a Xilinx sparten-3 EDK platform and tested using VISUAL BASICS. It is capable of supporting fast real-time edge detection for images and videos with various spatial and temporal resolutions. In this new Canny edge detector, instead of setting the threshold by manual, it can be set automatically by the algorithm itself. Therefore, this new algorithm is more adaptable to the changes of environments and illumination conditions.

To meet the needs of real time processing, design of a pipelined implementation on FPGA for this improved edge detection algorithm is considered. Compared with the implementation in a PC based system, this pipelined implementation on FPGA takes much less implementation time and can therefore be used for the mobile robot vision system which is very strict for the real-time performance of its vision system.

REFERENCES

[1]. F. M. Alzahrani and T. Chen, “A real-time edge detector algorithm and VLSI architecture,” Real-Time Imaging, vol.3, no. 5, pp. 363 – 78, 1997.

[2]. L. Torres, M. Robert, E. Bourennane, and M. Paindavoine,“Implementation of a recursive real time edge detector using retiming techniques,” VLSI, pp. 811 –816, Aug. 1995.

[3]. D. V. Rao and M. Venkatesan, “An efficient reconfigurable architecture and implementation of edge detection algorithm using Handle-C,” ITCC, vol. 2, pp. 843 – 847, Apr. 2004.

[4]. S. Varadarajan, C. Chakrabarti, L. J. Karam, and J. M. Bauza,“A distributed psycho-visually motivated Canny edge detector,” IEEE ICASSP, pp. 822 –825, Mar. 2010.

