

A Real Time Hardware and Software Co-Simulation of Edge Detection for Image Processing System

Obili Ramesh¹, P. V. Krishna Mohan Gupta², B. Sreenivasu³

^{1,2}M.Tech (Digital System & Computer Electronics)

³Associate Professor,^{1,2,3}Sreyas Institute of Engg., & Technology,
Affiliated to JNTU, Hyderabad, Andhra Pradesh, INDIA-500015,

Abstract---A methodology for implementing real-time DSP applications on a field programmable gate arrays (FPGA) using Xilinx System Generator (XSG) for MATLAB is presented in this paper. It presents architecture for Edge Detection using Sobel Filter & Canny Filter for image processing using Xilinx System Generator. The design was implemented targeting a Spartan3A DSP 3400 device (XC3S200-4TQ144). The Edge Detection method has been verified successfully with no visually perceptual errors in the resulted images and also comparing the both filters with an individually performance is justified clearly with Practical and Theoretical.

Keywords

System generator, Simulink, Sobel, Canny, Edge Detection.

I. Introduction

In Presently Global market for video processing systems requires high-performance digital signal processing as well as low device costs appropriate for a volume application. Xilinx FPGA devices provide a platform with which to meet these two contrasting requirements. A Xilinx tool, the System Generator for DSP [1] offers an efficient and straightforward method for transitioning from a PC-based model in Simulink to a real-time FPGA based hardware implementation. The system model can be simulated in the Simulink environment. This higher abstraction level reduces the analysis and debugging time. For real hardware testing, Xilinx System Generator supports the possibility to perform hardware in-the-loop co-simulation [2]. This methodology provides easier hardware verification and implementation compared to HDL based approach. The Simulink simulation and hardware-in-the loop approach presents a far more cost efficient solution than other methodologies. The ability to quickly and directly realize a control system design as a real-time embedded system greatly facilitates the design process[3]. The goal of this project was to implement an image processing algorithm applicable to Edge Detection system in a Xilinx FPGA using System Generator for DSP, with a focus on achieving overall high performance, low cost and short development time. After introducing, In section II. Image Processing [5] and Types of Image formats. In section III.

Digital Images and Segmentation. In section IV. Edge Detection. In section V. Descriptions of Sobel and Canny Operators. In section VI. System Design and Functional Architecture of Sobel and Canny. In Section VII, Implementation Of Image Edge Detection Using FPGA. In section VIII, Hardware/Software Co-Design in System Generator. In Section IX,. Conclusion and Results In Section X. References.

II. Image Processing

Digital image processing is the use of computer algorithms to perform on digital images .As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing. Since images are defined over two dimensions (perhaps more) digital image processing may be modeled in the form of multidimensional systems. Image Processing has different types are significant features .some of them are given below

a. Low-level: Edge detection, Corner detection , Blob detection Ridge detection Scale-invariant feature transform.

b. Image motion: detection area based, differential approach, optical flow

c. Shape Based: Thresholding, Blob, Templates matching and Hough Transform.

d. Flexible Methods: Deformable, parameterized shapes and Active contours (snakes).

IMAGE: An image is a two-dimensional picture, which has a similar appearance to some subject usually a physical object or a person. Image is a two-dimensional, such as a photograph, screen display, and as well as a three-dimensional, such as a statue. They may be captured by optical devices—such as cameras, mirrors, lenses, telescopes, microscopes, etc. and natural objects and phenomena, such as the human eye or water surfaces.

The word image is also used in the broader sense of any two-dimensional figure such as a map, a graph, a pie chart, or an abstract

painting. In this wider sense, images can also be rendered manually, such as by drawing, painting, carving, rendered automatically by printing or computer graphics technology, or developed by a combination of methods, especially in a pseudo-photograph.



Fig.2.1

An image is a rectangular grid of pixels. It has a definite height and a definite width counted in pixels. Each pixel is square and has a fixed size on a given display. However different computer monitors may use different sized pixels. The pixels that constitute an image are ordered as a grid (columns and rows); each pixel consists of numbers representing magnitudes of brightness and color.

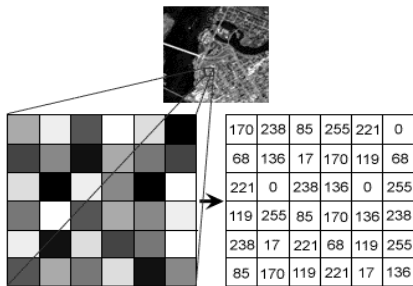


Fig 2.2

Each pixel has a color. The color is a 32-bit integer. The first eight bits determine the redness of the pixel, the next eight bits the greenness, the next eight bits the blueness, and the remaining eight bits the transparency of the pixel.



Fig 2.3

Types of Image File Formats:

Image file formats are standardized means of organizing and storing images. This entry is about digital image formats used to store photographic and other images. Image files are composed of either pixel or vector (geometric) data that are rasterized to pixels when displayed (with few exceptions) in a vector graphic display. Including proprietary types, there are hundreds of

image file types. The PNG, JPEG, and GIF formats are most often used to display images on the Internet.

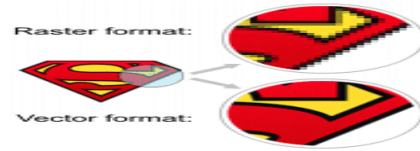


Fig.2.4

In addition to straight image formats, Metafile formats are portable formats which can include both raster and vector information. The metafile format is an intermediate format. Most Windows applications open metafiles and then save them in their own native format.

RASTER FORMATS:

These formats store images as bitmaps (also known as pixmaps. Examples are Jpeg/Jfif, Exif, Tiff, Png , Gif, Bmp

VECTOR FORMATS:

As opposed to the raster image formats above (where the data describes the characteristics of each individual pixel), vector image formats contain a geometric description which can be rendered smoothly at any desired display size. At some point, all vector graphics must be rasterized in order to be displayed on digital monitors. However, vector images can be displayed with analog CRT technology such as that used in some electronic test equipment, medical monitors, radar displays, laser shows and early video games. Plotters are printers that use vector data rather than pixel data to draw graphics. Examples are CGM,SVG,

III. DIGITAL IMAGE

A digital image is a numeric representation (normally binary) of a two-dimensional image. Depending on whether the image resolution is fixed, it may be of vector or raster type. Without qualifications, the term "digital image" usually refers to raster images also called bitmap images.

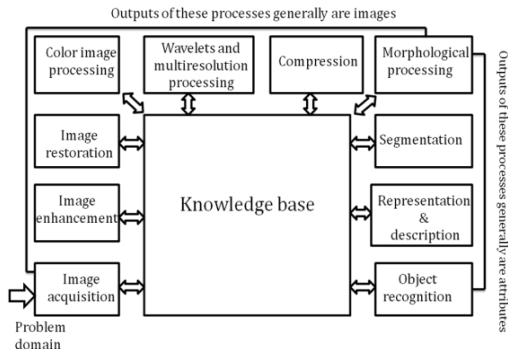


Fig 3.1

IMAGE ACQUISITION:

Image Acquisition is to acquire a digital image. To do so requires an image sensor and the capability to digitize the signal produced by the sensor.



Fig.3.2

Scanner produces a two-dimensional image. If the output of the camera or other imaging sensor is not in digital form, an analog to digital converter digitizes it. The nature of the sensor and the image it produces are determined by the application.



Fig.3.3

IMAGE ENHANCEMENT:

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interesting an image.



Fig.3.4

Image enhancement improves the quality (clarity) of images for human viewing. Removing blurring and noise, increasing contrast, and revealing details are examples of enhancement operations.

IMAGE RESTORATION:

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.



Fig.3.5

COLOR IMAGE PROCESSING:

The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from a scene. Second, humans can discern thousands of color shades and intensities, compared to about only two dozen shades of gray. This second factor is particularly important in manual image analysis.



Fig.3.6

WAVELETS AND MULTIREOLUTION PROCESSING:

Wavelets are the formation for representing images in various degrees of resolution. Although the Fourier transform has been the mainstay of transform based image processing since the late 1950's, a more recent transformation, called the wavelet transform, and is

now making it even easier to compress, transmit, and analyze many images.



Fig.3.7

Wavelets were first shown to be the foundation of a powerful new approach to signal processing and analysis called Multi-resolution theory.

COMPRESSION:

Compression, as the name implies, deals with techniques for reducing the storage required saving an image, or the bandwidth required for transmitting it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. Image compression is familiar to most users of computers in the form of image file extensions, such as the jpg file extension used in the JPEG (Joint Photographic Experts Group) image compression standard.

MORPHOLOGICAL PROCESSING:

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape. The language of mathematical morphology is set theory. As such, morphology offers a unified and powerful approach to numerous image processing problems.



Fig.3.8

SEGMENTATION:

In computer vision, image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as super pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to

locate objects and boundaries (lines, curves, etc.) in images.

BASED ON INTENSITY VALUES

Intensity values used for thresholding unless detected appropriately can result in segmentation errors. Thresholding and stretching images separate foreground pixels from background pixels and can be performed before or after applying a morphological operation to an image. While a threshold operation produces a binary image and a stretch operation produces a scaled, grayscale image, both operations rely upon the definition of an *intensity value*.

IV. Edge detection

It is the name for a set of mathematical methods which aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed edges. The same problem of finding discontinuities in 1D signals is known as step detection and the problem of finding signal discontinuities over time is known as change detection. Edge detection is a fundamental tool in image processing, machine vision and computer vision, particularly in the areas of feature detection and feature extraction. Region boundaries and edges are closely related, since there is often a sharp adjustment in intensity at the region boundaries. Edge detection techniques have therefore been used as the base of another segmentation technique. The edges identified by edge detection are often disconnected. To segment an object from an image however, one needs closed region boundaries. The desired edges are the boundaries between such objects.

V. Descriptions of Sobel and Canny Edge Operators:

Sobel Operator:

The Sobel operator is used in image processing, particularly within edge detection algorithms. Technically, it is a discrete differentiation operator, computing an approximation of the gradient of the image intensity function. At each point in the image, the result of the Sobel operator is either the corresponding gradient vector or the norm of this vector. The Sobel operator is based on convolving the image with a small, separable, and integer valued filter in horizontal and vertical direction and is therefore relatively inexpensive in terms of computations. On the other hand, the gradient

approximation that it produces is relatively crude, in particular for high frequency variations in the image. 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. If we define A as the source image, and G_x and G_y are two images which at each point contain the horizontal and vertical derivative approximations, the computations are as follows:

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * A \quad \text{and} \quad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A$$

where $*$ here denotes the 2-dimensional convolution operation. Since the Sobel kernels can be decomposed as the products of an averaging and a differentiation kernel, they compute the gradient with smoothing. For example, G_x can be written as

$$\begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} +1 & 0 & -1 \end{bmatrix}$$

The x-coordinate is defined here as increasing in the "right"-direction, and they-coordinate is defined as increasing in the "down"-direction. At each point in the image, the resulting gradient approximations can be combined to give the gradient magnitude, using:

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

Using this information, we can also calculate the gradient's direction:

$$\Theta = \text{atan2}(G_y, G_x)$$

where, for example, Θ is 0 for a vertical edge which is darker on the right side.



Fig.5.1 Original image

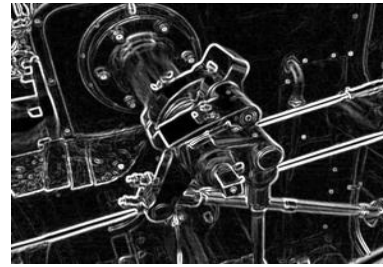


Fig.5.2 Sobel operator applied

Canny:

The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images.

Canny's aim was to discover the optimal edge detection algorithm. In this situation, an "optimal" edge detector means:

1. **Good Detection** – the algorithm should mark as many real edges in the image as possible.
2. **Good Localization** – edges marked should be as close as possible to the edge in the real image.
3. **Minimal Response** – a given edge in the image should only be marked once, and where possible, image noise should not create false edges.

To satisfy these requirements Canny used the calculus of variations – a technique which finds the function which optimizes a given functional. The optimal function in Canny's detector is described by the sum of four exponential terms, but it can be approximated by the first derivative of a Gaussian.

Steps involve in canny algorithm mainly :

Noise reduction

Because the Canny edge detector is susceptible to noise present in raw unprocessed image data, it uses a filter based on a Gaussian (bell curve), where the raw image is convolved with a Gaussian filter. The result is a slightly blurred version of the original which is not affected by a single noisy pixel to any significant degree.

Finding the intensity gradient of the image

An edge in an image may point in a variety of directions, so the Canny algorithm uses four filters to detect horizontal, vertical and diagonal edges in the blurred image. The edge detection operator (Roberts, Prewitt, Sobel for example) returns a value for the first derivative in the horizontal direction (G_x) and the vertical direction (G_y). From this the edge gradient and direction can be determined:

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

$$\Theta = \arctan\left(\frac{G_y}{G_x}\right).$$

The edge direction angle is rounded to one of four angles representing vertical, horizontal and the two diagonals (0, 45, 90 and 135 degrees for example).

Non-maximum suppression

Given estimates of the image gradients, a search is then carried out to determine if the gradient magnitude assumes a local maximum in the gradient direction.

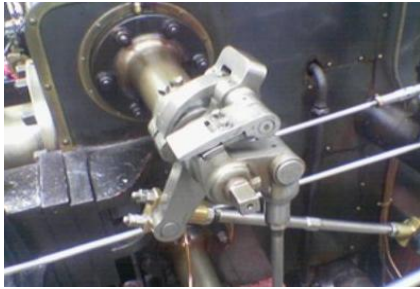


Fig.5.3Original image

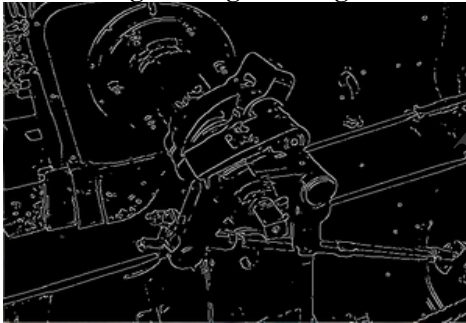


Fig.5.4Canny operator applied

VI. System Design and Functional Architecture of Sobel and Canny

The purpose of the design phase is to plan a solution of the problem specified by the requirement document. This phase is the first step in moving from the problem domain to the solution domain. The design of the system is perhaps the most critical factor affecting the quality of the hardware implementation. Here we build the System Block Diagram that is helpful to understand the behavior of the system.

In the proposed work entire system is divided into following:

1. Conversion of image to text file using MATLAB.

2. Image Edge detection using Sobel & Canny Operator.
3. Image Segmentation using Intensity Funtion.
4. Conversion of text file to image in MATLAB.

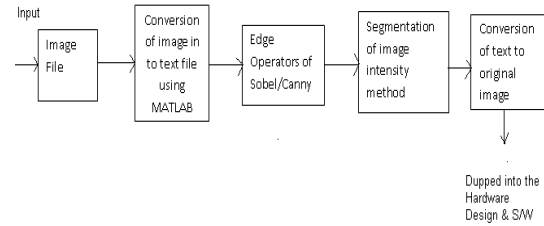


Fig.6.1

VII. IMPLEMENTATION OF IMAGE EDGE DETECTION USING FPGA

The edges of image are considered to be most important image attributes that provide valuable information for human image perception. The edge detection is a terminology in image processing particularly in the areas of feature extraction to refer to algorithms which aim at identifying points in a digital image at which the image brightness changes sharply. The data of edge detection is very large so the speed of image processing is a difficult problem. FPGA can overcome it. Sobel operator is commonly used in edge detection. Sobel operator has been researched for parallelism but Sobel operator locating complex edges are not accurate. It has been researched for the Sobel enhancement operator in order to locate the edge more accurate and less sensitive to noise but the software can not meet the real-time requirements . The popular Canny edge detector uses the following steps to find contours presents in the image. The first stage is achieved using Gaussian smoothing. The resulting image is sent to the PC that sends it back to the gradient filter, but here we modified our gradient filter a bit because this time we don't only need the gradient magnitude that is given by our previous operator, but we need separately G_x and G_y . We also need the phase or orientation of our gradient which is obtained using the following formula: $\theta = \arctan[G_y/G_x]$ and for Sobel Modulus of $G_x + \text{Modulus of } G_y$.

VIII. Hardware/Software Co-Design in System Generator

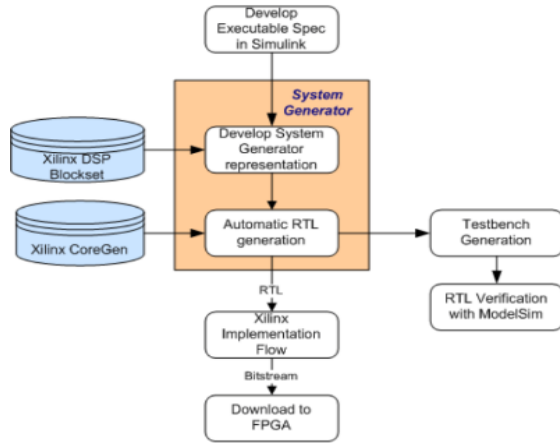


Fig.8.1

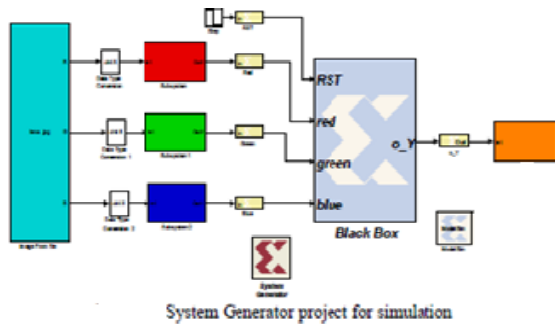


Fig.8.2

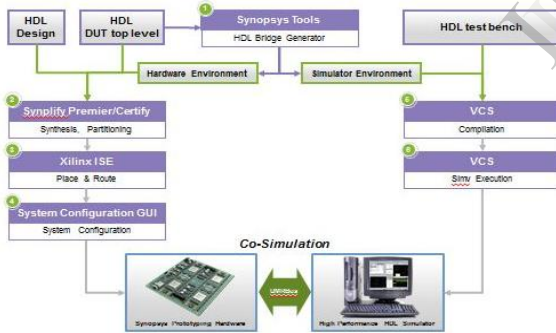
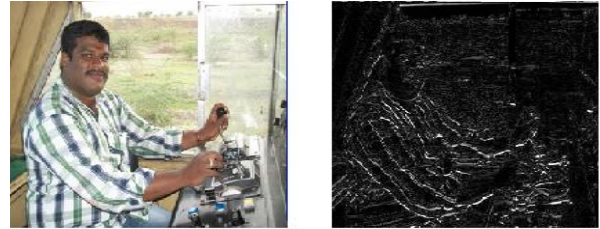


Fig.8.3

IX. Results



Original Image

Sobel Operator used



Canny Operator used

Conclusion

This paper is present Real Time Hardware And Software Co-Simulation Of Edge Detection For Images Processing System .Comparing the results using Xilinx Spartan 3 with two familiar edge detection methods Sobel and Canny . By observing the synthesis results we concluded that Canny edge detection method gives sharp edge image compare to sobel method.Future works include the use of the Xilinx System. Generator development tools for the implementation of other blocks used in computer vision like feature extraction and object detection on Xilinx Programmable Gate Arrays (FPGA).

X. REFERENCES

1. Xilinx System Generator User's Guide, www.xilinx.com
2. K.Van Beeck, F.Heysten, J.Meel, T.Goedemé, "Comparative study of Model-based hardware design tools", *In proceedings of European Conference on the Use of Modern Electronics in ICT, ECUMICT 2010*, Ghent, Belgium, 25-26 March, 2010
3. T. Saidani, D. Dia, W. Elhamzi, M. Atri and R. Tourki, "Hardware Co-simulation For Video Processing Using Xilinx System Generator" *Proceedings of the World Congress on Engineering 2009 Vol I, WCE 2009*, July 1 - 3, 2009, London, U.K.
4. <http://www.mathworks.com/>
5. Tim Morris (2004). *Computer Vision and Image Processing*. Palgrave Macmillan. ISBN 0-333-99451-5.
6. Bernd Jähne (2002). *Digital Image Processing*. Springer. ISBN 3-540-67754-2.