FPGA design and implementation for vertex extraction of polygonal shapes

Jorge Martínez-Carballido Jorge Guevara-Escobedo Juan M. Ramírez-Cortés
Instituto Nacional de Astrofísica, Óptica y Electrónica
jmc@inaoep.mx *jorgege@inaoep.mx* *jmram@inaoep.mx*
Rubén Alejos-Palomares
Universidad de las Américas Puebla
ruben.alejos@udlap.mx

Abstract

This work focuses on developing systems of blocks in SIMULINK and VHDL to reuse on design of applications involving the recognition of polygonal objects. Usage of this work reduces time in the development of prototype solutions using FPGAs.

The vertex extraction algorithm uses contour's corners as the candidate set to select vertices by using local properties. SIMULINK and VHDL implementations were tested to be equal at I/O level. Using a Spartan 3E FPGA Starter kit for the hardware implementation probed that the VHDL implementation synthesizes. Binary images are loaded into the FPGA through a microSD memory card and the resulting data from the FPGA process is visualized through the Starter Kit built-in alphanumeric LCD.

Test cases consider artificial cases to ensure wide case testing for case combination of the algorithm. These sets of components in software and hardware contribute by easing implementations on computer vision applications using polygons for object identification.

1. Introduction

A polygon is a closed multi-linear curve. Polygons are basis for many of the objects of interest in the field of computer vision. On industrial applications there is an increased need to integrate; camera and processing on the same device, where usage of FPGA is becoming a good choice given the advantages of re-configurability and the possibility of high performance implementations, using parallel architectures.

Computer vision is a rich topic for research and study; increasingly, it has a commercial future. While the goal of computer vision is to make useful decisions about physical objects and scenes based on sensed images; computer vision systems become important whenever automation or improvement of industrial or human activities is a goal.

Computer vision research is an interdisciplinary field closely related to human vision system, with applications in a variety of areas such as: mechanical piece inspection, agricultural quality, electronic circuit board inspection, biometrics, medical diagnostics [1], and automotive safety [2] [3].

Along the years, starting on the 1960's, there has been specialized hardware to accelerate and/or integrate functionality to computer vision applications. Some of this are: array and vector processors, FPGA, and GPU based hardware implementations.

Computer vision applications make use of object representation; many object representations include shape representation. We present design and implementation on FPGA that extracts vertices of a binary image containing a contour of a polygon. This work uses as hardware platform a Xilinx Spartan 3E (500K) Starter kit with microSD for image loading.

The following sections describe design, development, testing, and results of this work.

2. Algorithm

On the design phase for vertex location on a binary image of a polygonal object, a decision on the contour's representation and on the vertex location method has to be taken.

Contour based representation is divided in the following three classes [4]:

- *Parametric Contours:* the shape outline is represented by a parametric curve implying a sequential order along it.
- *Set of Contour Points:* the shape outline is simply represented as a set of points, without any special order among them;
- *Curve Approximation:* a set of geometric primitives like straight line segments are fitted to the shape outline.

Polygons can naturally be represented by its set of vertices, where representation is independent to vertex order; thus, this work takes a set of contour points to represent a resulting polygon.

978-1-4244-9557-3/11/$26.00 ©2011 IEEE

Vertex detection and polygonal approximation methods can be global or local:

- Global methods use an error function to estimate quality of the resulting approximation [5].
- Local methods go directly to identify vertex location, using contour properties [6] [7] [8].

Given that FPGA implementation is the final target of this work, it is of high relevance to consider simplicity of elements with preference to use integer arithmetic and/or logical binary components for the vertex location method. This leads to use a local method to identify vertex location on the contour.

From the above it can be said that this solution approach uses a set of contour points, locates vertices on the contour by using local properties and represent the resulting polygon with a set of vertices.

3. Algorithm developing

Software based tools are good for quick development and test cycles while trying options for algorithm development. Here we used MATLAB for this stage. The details of the algorithm is subject of another report, let just say that we used a list of corners as representation of the binary image contour and with local properties, a subset of this corners are identified as vertices of the contour.

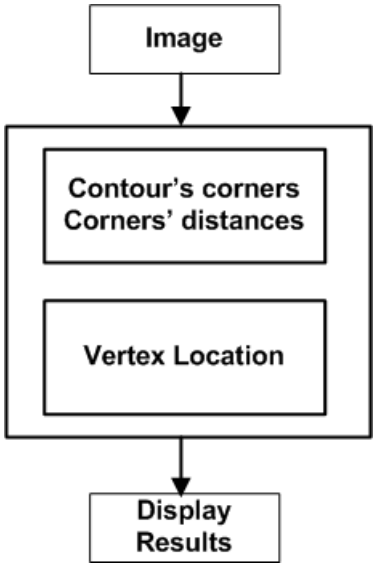

Figure 1. System diagram

4. Implementation

Development was performed concurrently using SIMULINK and VHDL implementations for each of the internal blocks, while integrating them to form the system. Equivalence between systems is at the IO level, meaning that given the same image, one gets the same set of vertices representing the polygon on each system.

The system has two subsystems: contour's local properties and vertex location. With the use of corners on the contour, instead of sets of contour points that form a straight line segment [7], and the difference between corners locate a vertex. The system needs peripherals as means to get an image into memory to process it and one to present the resulting vertices. Diagram on Figure 1 represents this.

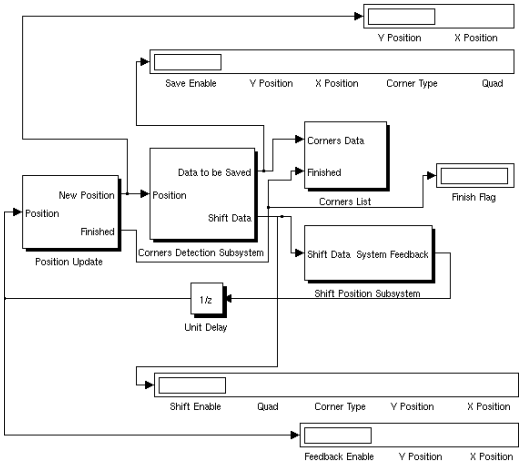

Figure 2. SIMULINK Contour's local properties system

Once the algorithm for the system was defined by coding and testing on 'M' language from MATLAB, a system decomposition to define blocks that will be more related to a final development on a digital design was performed using level-2 S functions of SIMULINK: a) Figure 2 presents the contour's local properties, extracting corners and their differences. b) Figure 3 presents the block system in SIMULINK for the vertex location system.

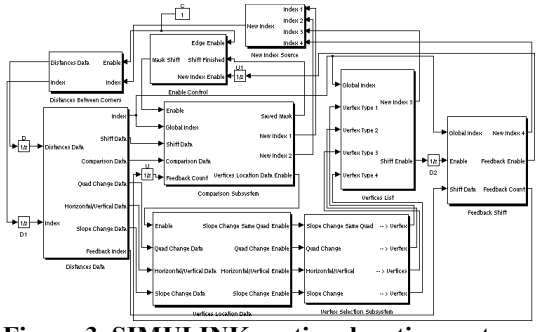

Figure 3. SIMULINK vertices location system

978-1-4244-9557-3/11/$26.00 ©2011 IEEE

Concurrently the SIMULINK and VHDL systems were developed, resulting on systems of VHDL entities to integrate the contour's local properties and vertex location systems; whose corresponding diagrams are shown on Figure 4 and Figure 5. The fact that SIMULINK blocks were first implemented then the VHDL ones, eased the process to have working VHDL entities to the synthesized system on the FPGA Spartan 3 starter kit. Given that by using SIMULINK to try design options can be done in less time than on a VHDL simulation and synthesis.

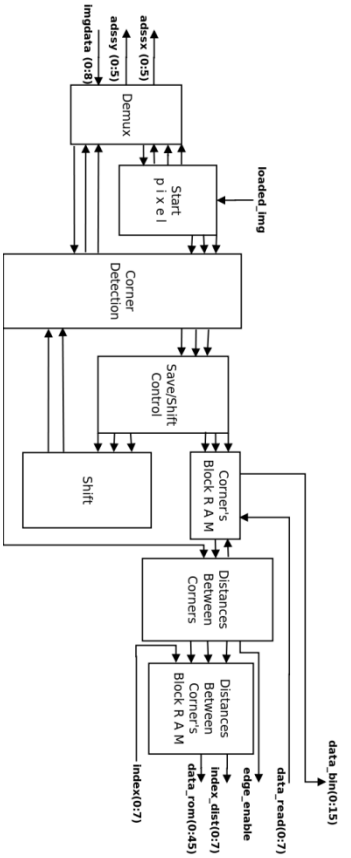

Figure 4. VHDL Contour's local properties system

A difference between the SIMULINK and VHDL implementations is that IO subsystem in SIMULINK is readily available on the computer and that on the synthesized VHDL to FPGA, IO needs to be implemented. For this system a microSD was used as storage and an LCD display for vertex location output. Therefore subsystems for each peripheral were developed.

For the microSD memory card, the FAT32 file system was used to be able to read images form images saved on a microSD card on a personal computer; this ensures that the image written by the an external application can be read into the FPGA system from the microSD memory card with the 'BMP' standard uncompressed image file format.

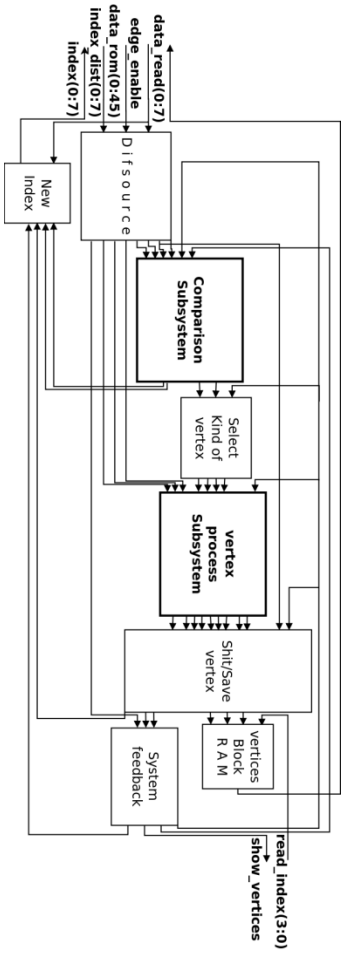

Figure 5. VHDL Vertex location system

5. Tests and Results

This section presents results obtained in the three different implementations for the implemented algorithm (MATLAB function, SIMULINK block system and VHDL system). A group of binary images in 'BMP' format of size 64x64 - 574 bytes (512 bytes raw image + 62 bytes bmp format header) representing a combination of all possible type of vertices to be found were selected as examples; this test cases include cases found on images of polygonal objects in a real environment as obtained from an industrial case.

Careful selection of test cases considered each type of vertex and combination of each type of vertex, giving seven test cases. Figure 6 shows the results generated by the SIMULINK implementation

978-1-4244-9557-3/11/$26.00 ©2011 IEEE

for each of the seven cases, for each case a list of the vertices located and its corresponding image with their vertices are shown.

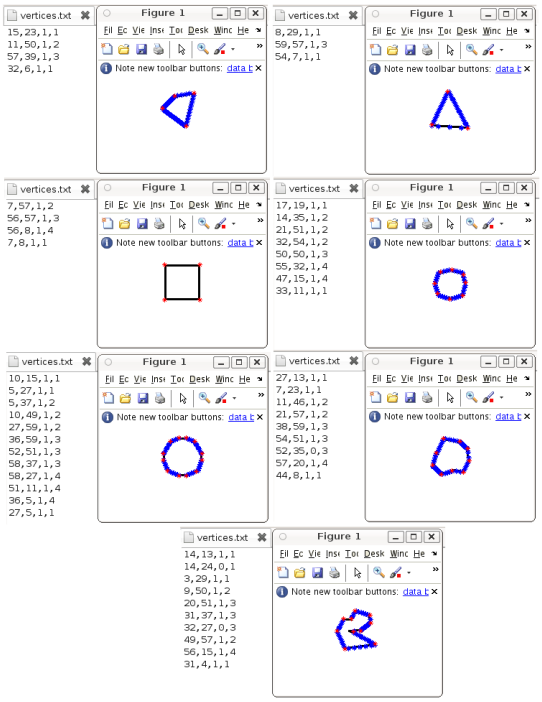

Figure 6. SIMULINK tests and results

For the VHDL implementation case, a Spartan 3E (500K) starter kit with proto-board expansion for the microSD connection was used and shown on Figure 7. The rotating knobs of the kit was used to read vertices location and are displayed on the LCD screen, Figure 8 illustrates the usage of the rotating knob for a four vertices polygon.

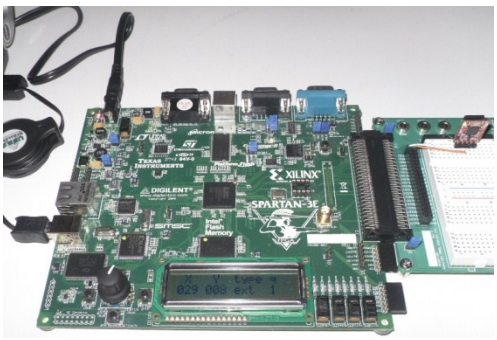

Figure 7. FPGA system

Using Xilinx ISE to synthesize the VHDL system for the Spartan 3E (500K) starter kit, the RTL diagram obtained is shown on Figure 9.

MATLAB and Simulink Testing was performed by using MATLAB R2008a running under Linux Operating System (Ubuntu 9.10) in a Intel Core 2 Duo processor at 1.83Ghz of clock frequency, with 2 GBytes of RAM.

Using the MATLAB implementation time execution is estimated for the implemented algorithm; Table 1 shows the obtained results along

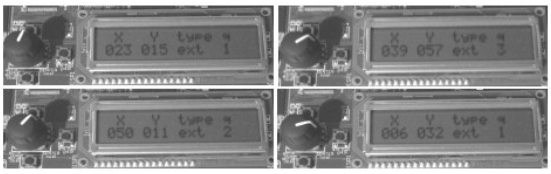

Figure 8. LCD display with sample vertices location

with the number of pixels forming the polygon's contour, the number of contour pixels that are corners, the number of vertices, and execution time in milliseconds found.

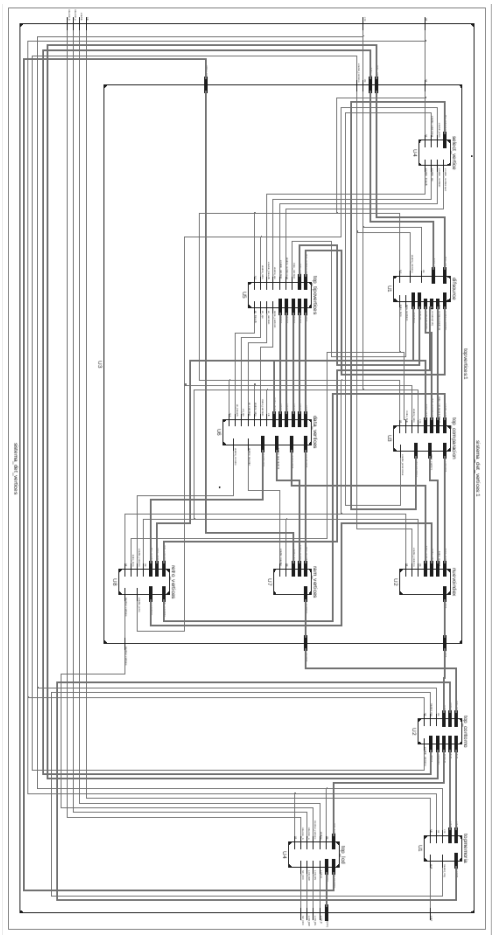

Figure 9. Vertices' system RTL

978-1-4244-9557-3/11/$26.00 ©2011 IEEE

Results show that timing is closely related to the number of corners of the contour, given that the algorithm uses the list of corners to extract vertices by using local properties.

Table 1. Summary of tests results

Image	Pixels	Corners	Vertices	Time ms
pol1.bmp	181	115	4	11
pol2.bmp	203	107	3	12
pol3.bmp	197	4	4	2
pol4.bmp	169	77	8	9
pol5.bmp	215	115	12	11
pol6.bmp	207	117	9	12
pol7.bmp	261	139	10	14

6. Conclusions

A SIMULINK and VHDL component library was designed and implemented by refactoring a MATLAB algorithm. Both systems were tested for equivalence at the I/O level with a set of designed cases.

The FPGA implementation used microSD memory and Starter Kit built-in LCD; thus, images could be loaded into the FPGA and results could be visualized through the LCD.

A set of components in software and hardware, was designed, developed and implemented to contribute on computer vision implementations of applications using polygons for object identification.

7. References

[1] Xiaolei Huang and Dimitris N. Metaxas, "Metamorphs: Deformable Shape and Appearance Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 8, pp. 1444-1459, August 2008.

[2] Carlos Filipe Paulo and Paulo Lobato Correia, "Traffic Sign Recognition Based on Pictogram Contours," in *Ninth International Workshop on Image Analysis for Multimedia Interactive Services*, Klagenfurt University, Austria, 2008, pp. 67-70.

[3] Gareth Loy, David Shaw Nick Barnes, "The regular polygon detector," *PatternRecognition*, vol. 43, p. 592–602, 2010.

[4] Roberto Marcondes Cesar Jr. Luciano da Fontoura Costa, *Shape Analysis and Classification*.: CRC Press LLC, 2001.

[5] Alexander Kolesnikov and Pasi Fränti, "Polygonal approximation of closed discrete curves," *Pattern Recognition*, vol. 40, pp. 1282-1293, 2007.

[6] Denise Guliato, Juliano D. de Carvalho, Sérgio A. Santiago Rangaraj M. Rangayyan, "Polygonal approximation of contours based on the turning angle function," *Journal of Electronic Imaging*, vol. 17, no. 2, pp. 023016-1 - 023016-14, Apr-Jun 2008.

[7] Partha Bhowmick and Bhargab B. Bhattacharya, "Fast Polygonal Approximation of Digital Curves Using Relaxed Straightness Properties," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 9, pp. 1590-1602, September 2007.

[8] Lars Kulik, Mike Worboys,Antony Galton Matt Duckham, "Efficient generation of simple polygons for characterizing the shape of a set of points in the plane," *Pattern Recognition*, vol. 41, pp. 3224 - 3236, 2008.

978-1-4244-9557-3/11/$26.00 ©2011 IEEE