**NWTM Interface - The Next Wave Technologies and Markets
Programme of the Department of Trade and Industry**

FPGA Based Video Processing System for Ubiquitous Applications

H Meng, N E Pears and C Bailey
University of York, UK

Printed and published by the IEE, Michael Faraday House, Six Hills Way,
Stevenage, Herts SG1 2AY, UK

FPGA BASED VIDEO PROCESSING SYSTEM FOR UBIQUITOUS APPLICATIONS

H Meng, N E Pears and C Bailey

University of York, UK

ABSTRACT

In this paper, a novel FPGA based video/image processing architecture is introduced that is designed to deal with advanced computer vision applications in ubiquitous systems. As basic components of this architecture, some low-level image feature extraction algorithms including the Sobel edge detector and SUSAN edge and corner detectors were designed and implemented using FPGA software tools. A simple PC-FPGA combined platform was proposed and designed for testing the performance of our design. Experimental results showed these components worked very well on an FPGA chip.

INTRODUCTION

A long-standing problem with computer vision applications is the difficulty in attaining real-time performance of visually driven tasks due to the computationally intensive and data intensive nature of the processes, particularly on small, low-power devices such as mobile or ubiquitous devices.

One solution is to use special purpose DSP processors, designed to execute certain low-level image processing operations extremely quickly. However, this approach can sometimes be over specialised, when one has a large range of possible applications in mind, since this requires a commensurately large range of low and medium level operations. An alternative solution to a software implementation (running on either general or specialised hardware) is the design of specific hardware for specific computer vision processes, in order to perform a high rate of operations per second. If only a small number of specific processes are required for a particular application, then only those particular processes need to be implemented in the hardware.

Continuing growth in silicon chip capability is rapidly reducing the number of chips in a typical system, and increasing the size, performance and power benefits of System-on-Chip (SoC) integration. However, the design and test of a miniaturised vision system can be quite strenuous, owing to many technology and financial constraints that often restrict the developer's pool of resources.

Meanwhile, advances in programmable logic devices have resulted in the development of Field Programmable Gate Arrays (FPGA) that allow integration of large numbers of programmable logic elements in a single chip. The size and speed of FPGAs are comparable to ASICs, but FPGAs are more flexible and their design cycle is shorter. It is possible that FPGA architectures will allow generic real-time image processing, computer vision and pattern recognition techniques to be packaged with a relatively low power CPU and an image sensor.

In contrast to developing a specific vision SoC, we aim to develop a general-purpose architecture of video/image processing for ubiquitous systems. The most important advantage of this general-purpose architecture is its flexibility, which promotes diversity of potential applications and longer product lifetime. Ubiquitous systems include devices around the home with embedded processing power, which allows them to monitor user activity and interact in an intelligent way with users. We aim to implement a library of generic image processing, computer vision and pattern recognition algorithms in an FPGA-based architecture. The low-level, high bandwidth processes, such as smoothing and feature extraction, will be implemented directly in hardware, whilst higher level, lower bandwidth processes, such as task-oriented combination of visual cues, will be implemented in software architecture. Thus part of the FPGA will be configured as a relatively low power CPU.

The rest of this paper is organised as follows: In Section 2, we will give a brief introduction to our video processing architecture for ubiquitous systems. In Section 3, some low-level image feature extraction algorithms, as the basic vision components of our architecture, will be introduced. In Section 4, FPGA design of these components and the testing platform are presented. In Section 5, some experimental results are presented. Finally, we present some discussion and the conclusions.

VIDEOWARE ARCHITECTURE

We propose an FPGA-based architecture named "Videoware" for real time video/image processing in ubiquitous systems. An abstraction of the data-flow in the framework that we propose to develop is shown in figure 1. The essential feature is that multiple sources of visual information must be simultaneously extracted and integrated in order to generate a robust interpretation of video data. On the bottom row of the diagram, we have generic features, such as edges, corners and 2D blobs of colour and texture. These generic features can be extracted and used to form more complex constructs, shown on the second row from the bottom of the diagram, which are informative visual cues. These include motion, parallax, invariants, segmentation and 3D structure.

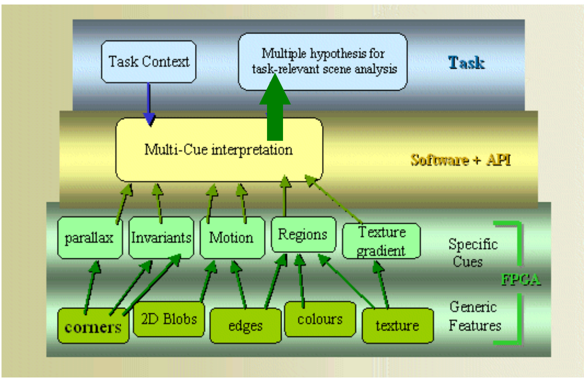


Figure 1 - Videoware architecture

In the context of a particular visual task, we must determine which cues are informative and which are not. In addition, we must determine how the informative cues should be combined, in order to output one or more informed hypotheses for task relevant scene representation.

This architecture will be used for development of real time video/image processing applications in ubiquitous systems. Besides its flexibility, it also benefits from ease of interfacing and can be tested using high-level languages.

LOW-LEVEL IMAGE FEATURE EXTRACTION COMPONENTS

In the early stages of our project, we were focussed on the development of the basic image processing components implemented as intellectual property (IP) cores in our proposed architecture. We selected the Sobel edge detector and the SUSAN edge and corner detectors as our initial low-level feature extraction algorithms for edge and corner information.

Sobel edge detector

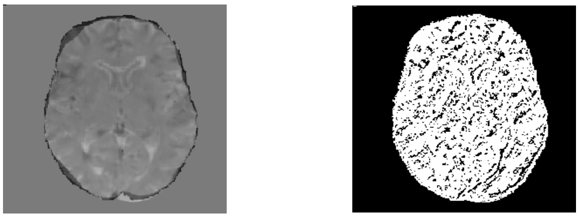
The Sobel edge detector (Gonzalez and Woods (1)) works by applying two 3x3 kernels to the input image, one designed for detecting vertical edges and one for horizontal edges. These are shown below in figure 2.

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

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

Figure 2 - Filters for Sobel edge detector

The responses of the two kernels are added together (not strictly correct, but fast and computationally cheap) and this value gives the overall edge response. Figure 3 shows an example result of the Sobel edge detector.



(a) Original image (b) Edges of the image

Figure 3 - Sobel edge detection of an image

SUSAN edge/corner detector

The SUSAN algorithm, for edge and corner detection, was developed by Smith and Brady (2). The algorithm follows the usual method of taking an image and, using a predetermined window centred on each pixel in the image, applying a locally acting set of rules to give an edge response. The response is then processed to give a set of edges as the output.

The SUSAN edge detector works by finding the differences between the centre pixel and all the other pixels in the kernel. These differences are then compared to a threshold value. The number of pixels that pass the threshold determines the response. The motivation for this is that the more pixels that are different from the centre pixel, the higher or stronger the edge. The meaning of ‘different’ is set by a threshold. There is also a final thresholding step that takes this initial response and applies a second threshold to it. This reduces the effect of noise in the input image. The SUSAN filter is written to use a 7x7 circular kernel as showed in figure 4. The difference between the SUSAN edge and corner detection algorithms is the locally acting set of rules used.

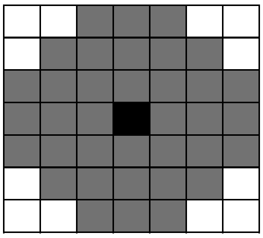
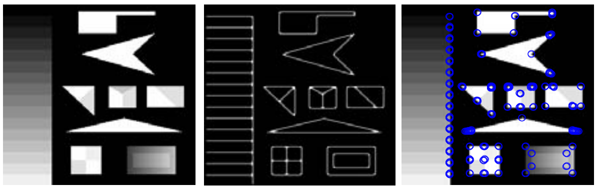


Figure 4 - Circular kernel of SUSAN algorithm

Figure 5 shows the experimental results of the SUSAN edge and corner detectors. Corner pixels are marked by circles.



(a) Original image (b) Edges (c) Corners

Figure 5 - SUSAN edge/corner detection

FPGA DESIGN OF COMPONENTS

For the above edge/corner detectors, a uniform architecture was designed for the implementation in VHDL, as shown in figure 6. The image is input to the FPGA chip line by line. Only a very small number of lines of image are stored in the data buffer. Then the edge/corner detector works as a filter on the stored data and the resultant image is output in the same order, line by line.

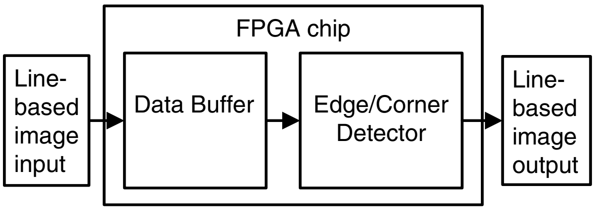


Figure 6 - Diagram of component architecture

For the Sobel edge detector, the image signal was input line by line with only three lines stored in the FPGA chip. The two filters of the Sobel algorithm are combined in the implementation on the FPGA chip. The output image is also line-based and is output in the same order as it was input.

The SUSAN edge/corner detector is more complex to implement than the Sobel edge detector. There are at least 6 lines stored in the FPGA chip and the processing is more involved than the Sobel algorithm, thus the used chip area is a little bit bigger. Table 1 showed the resources used by the three algorithms.

Table 1 - Comparison of resources used by Sobel edge detector, SUSAN edge detector and SUSAN corner detector in the FPGA chip Xilinx SPARTAN 2E.

	Sobel edge detector	SUSAN edge detector	SUSAN corner detector
Number of occupied slices	181 (5%)	885 (28%)	848 (27%)
Equivalent gate count	26,280	83,534	82,466

EXPERIMENT RESULTS

Testing platform

In order to test the basic low-level image processing components, designed using a hardware description language (VHDL), a simple PC-FPGA test harness was implemented. In this system, there is an FPGA chip on a "BurchED" prototyping board. It is a Xilinx SPARTAN 2E chip with 300K gates. The FPGA board is shown in figure 7. It is connected to a PC by a parallel cable through a JTAG connector. The VHDL codes were developed in the free Xilinx WebPACK design software (Xilinx ISE 7.1i) and were synthesized and implemented by Xilinx tools. Finally, the FPGA design was downloaded to the FPGA board using the iMPACT tool, through the parallel cable.

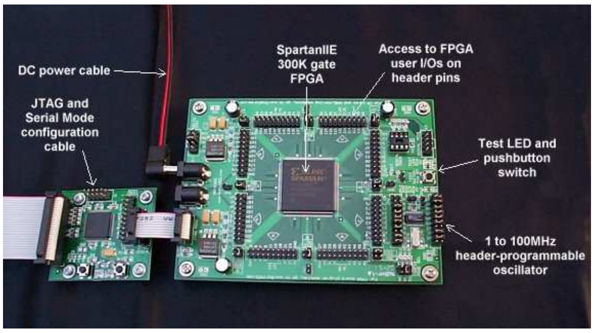

DC power cable
SpartanIIE 300K gate FPGA
Access to FPGA user I/Os on header pins
JTAG and Serial Mode configuration cable
Test LED and pushbutton switch
1 to 100MHz header-programmable oscillator

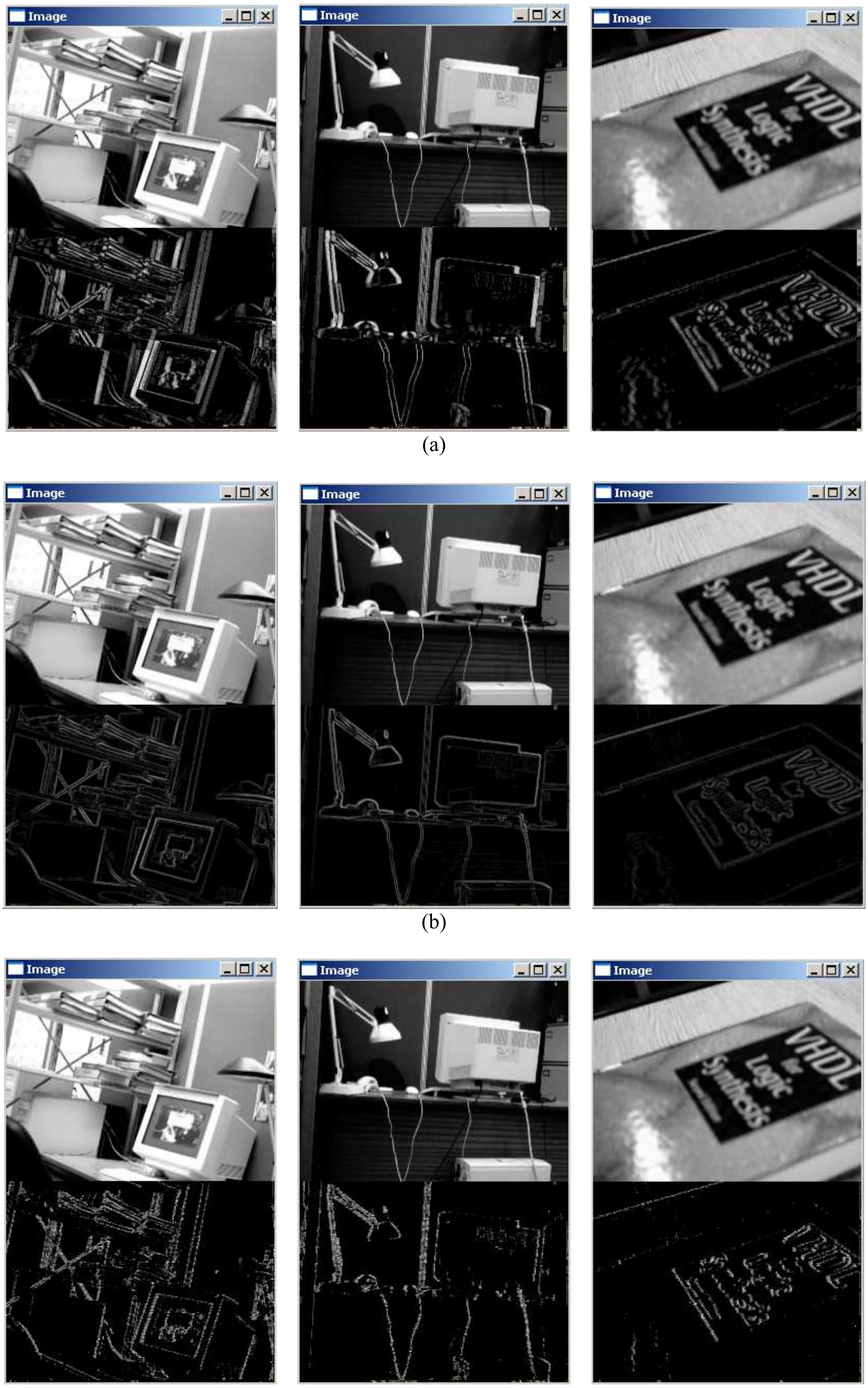
Figure 7 - BurchED BX-300 FPGA board

It should be mentioned here that the parallel cable can also be easily used as a data communication cable using the pushbutton on the board. On the PC side, we have developed a friendly interface using Visual C++. In it, image data can be sent to the FPGA board through the parallel port and the results are received through the same parallel cable. Finally, both original image and output image can be displayed on the screen of the PC.

It should also be mentioned here that we do not intend to use the BurchED board for our final Videoware architecture, rather we will use an in house development board with a much larger FPGA. The BurchED board was simply used for testing the hardware IP components, as a convenient test harness.

Testing results

For each algorithm, VHDL codes were synthesized and implemented in Xilinx software and then they were downloaded into the FPGA chip SPARTAN 2E. Finally, the parallel cable was switched to data communication mode. On the PC side, the interface software starts and the images were sent to the parallel port and the resultant output images were received. Figure 8 shows the outputs of our experiments. For each algorithm, three example results are shown. In each picture, the above half is the original image and second half is the edge or corner output image



(a)

(b)

(c)

Figure 8 - Screen shoots of three example results using (a) Sobel edge detector (b) SUSAN edge detection and (c) SUSAN corner detectors. All results based on the hardware implementations on an FPGA chip

generated by the algorithms directly implemented in hardware on the FPGA chip.

For edge detection, the Sobel edge detector extracted more detailed information than the SUSAN edge detector, whilst the SUSAN edge detector got a more clearly defined outline of the objects in the images. For the SUSAN corner detector, most of the corner information in the images has been successfully extracted, but the algorithm can give many false positives, particularly along edges in the image.

CONCLUSION AND DISCUSSIONS

In this paper, we have proposed an FPGA-based architecture named "Videoware" for real time video/image processing. It is a different approach to other architectures aiming for a complete vision SoC. It aims to meet the growing demand of real-time image processing in ubiquitous systems and takes advantage of the rapid development of performance/cost in the FPGA chip industry.

In the early stages of our project, we selected and implemented three basic low-level feature extraction algorithms, as some of the basic components within our architecture and we have shown that these components work well on an FPGA chip. We are continuing to develop further low-level image processing IP cores and aim for a basic image processing IP core library to form the lowest layer of our architecture.

In terms of performance, the SUSAN edge detector has achieved better results than the Sobel algorithm for edge detection. Of course, the cost on hardware resources is also bigger. For corner detection, although the SUSAN corner detector worked to some extent, there were many false positives along edges in the image and we intend to compare SUSAN with other algorithms for corner detection, both in terms of performance and gate count. At the same time, we will find good algorithms for other low-level image features, such as texture features, in our proposed Videoware architecture.

Our BurchED testing platform could be used to test these new components. However, there are some disadvantages. Firstly, the parallel cable has very limited communication ability. This makes this approach inappropriate, when the images are input and output at a high frequency in real-time applications. Secondly, the SPARTAN 2E chip is too small, which would be a significant constraint when we develop more complex algorithms. However, we are in the early stages of our project. Ultimately, the whole architecture will be deployed on an in-house platform, with a larger FPGA, allowing larger and more complex components to be added into the system.

ACKNOWLEDGEMENTS

The authors would like to thank DTI and Broadcom Ltd. for the financial support for this research. We also would like to thank Mr. Peter Stock for his previous work on this project.

REFERENCES

1. Gonzalez R. and Woods R., 1992. "Digital image processing", Addison Wesley, pp 414 – 428.

2. Smith S. and Brady J., 1995. "SUSAN – A new approach to low level image processing", Technical Report, Oxford Centre for Functional Magnetic Resonance Image of the Brain (FMRIB).