HDL Implementation of Digital Image Display on VGA through FPGA

Rohit Raj
Student/Research Scholar
BGIET, Sangrur
Distt. Sangrur-148001, India

Gaurav Mittal
Assistant Prof.
BGIET, Sangrur
Distt. Sangrur-148001, India

Monika Aggarwal
Associate Prof.
BGIET, Sangrur
Distt. Sangrur-148001, India

ABSTRACT
For rigorous computer vision applications for image processing on embedded platforms is still a very challenging task. For a customized hardware Field-programmable gate arrays (FPGAs) offer a suitable technology to accelerate image processing. Most recent available image processing frameworks are concentrated on pixel-based modules for simple preprocessing tasks which are mainly defined using MATLAB. This presented paper deals with the aims to implementation of image/digital data into real time hardware using HDLs with the motto of relatively inexpensive and adaptable technology. Thus, it offers modules and interfaces to perform operations and incorporate software defined operations. This paper will show to how the digital image (of any source/formats) can be stored into a memory(RAM/ROM) onto the FPGA and how it can be further processed for larger display onto the TV (i.e. VGA).

Keywords
FPGA, Simulation, Synthesis, Core generation, VGA, HDLs, Xilinx.

1. INTRODUCTION
The efficiency and capability of available FPGAs is growing rapidly in recent decades. Today, low or mid range devices allow the implementation of Networks on chip (NoC) or even complex Systems on Chip (SoC) on a single device of FPGA. Field Programmable Gate Arrays (FPGAs) has the great flexibility to generate the required hardware structures that optimally fit a given algorithm and to efficiently exploit parallelism. In addition the growing capacities make FPGAs a dominated platform with the capability of integration of even more complex image processing tasks in the near future. Thus it makes the Field Programmable Gate Array (FPGAs) to become an alternative for the software algorithms implementation. The major example of this is in many applications from video surveillance to remote sensing image applications. Field programmable gate arrays are the large-scale integrated circuit that can be reconfigured where "field programmable" refers to ability of changing the operation of the device as per the application demands. While gate array refers to the basic internal architecture that makes reconfiguration possible.

As previously mentioned, the implementations of real-time image processing algorithms can be done on general purpose microprocessors or microcontrollers. Due to the high computational density and parallel processing capability of the FPGAs it has a major impact on image or video processing when compared to a general purpose microprocessor or microcontrollers. If both of these features coupled together with the ability of FPGA of being reprogrammable it leads to the flexibility in the development of image processing algorithms on FPGA. That's why during the recent years FPGAs have become the dominant form of programmable logic implementation.

Real time computer vision is a fast growing area of research and development with lots of applications in the avionics, automotive, consumer electronics industries and medical systems. The involved algorithms range from simple image preprocessing to high degree of complexity interpretations. Particularly in the field of an embedded system, the available processing power, memory bandwidth and power resources are very limited, and all of which must be needed to perform real-time image processing thus using dedicated hardware is a well-known way to accelerate signal processing. In this project we would start with storing of an image which might be taken from any source such as HDDs, MicroSds etc. or any other media file in either BMP, GIF or Jpeg format and then will convert that image information into digital information form which is said to be a coefficient file, for this we must have some familiarity with MATLAB and then that information would be sent to store into a FPGA ROM, for this purpose we can choose Xilinx CORE Generator, that delivers the cores optimized for Xilinx FPGAs. CORE Generator is mainly used to create high density, high performance designs in Xilinx FPGAs in less time. The CORE Generator is comes with almost all of the XILINX ISE version(s) with an extensive library of Xilinx LogiCORE Intellectual Property Block. These could include DSP functions, memories, storage elements, math functions and a variety of basic elements to high speed transceivers.

2. RELATED WORK
Image processing on embedded platform is still a very challenging work as stated by **Michael Schaeferling et. al. (2015),** [1]. His work basically related to the development of frame work for image processing with FPGAs. They also stated that major available image processing frameworks for FPGAs concentrated only on pixel-based modules for simple preprocessing tasks. His work presents the framework which also aims to cover the integration of higher-level algorithms. Thus, his work offers modules and interfaces to perform window-oriented filter operations and incorporate software defined operations. They also incorporated several complex and higher-level algorithms, such as undistortion and rectification, natural feature description, edge detection and Hough transform have been adopted and integrated into the frameworks image processing implementation. His work completely describes the framework with several interfaces and some of the existing modules used. Finally, some applications which were already implemented using this framework are also presented.

In another research given by **Mariangela Genovese and Ettore Napoli (2014),** [2] as per them the background identification/clarification is a most common feature in many video processing systems. There work proposed two hardware implementations of the OpenCV version of the Gaussian mixture model (GMM), a background identification algorithm. There implemented versions of the algorithm allows a fast initialization of the background model while an hardware-oriented, innovative and the formulation of the GMM equations makes the proposed circuits able to perform real-time background identification on high definition (HD) video sequences. The proposed circuit is designed with commercial field-programmable gate-array (FPGA) devices as target at first with implementation on Virtex6 vlx75t, the proposed circuit process 91 HD fps (frames per second) and uses 3% of FPGA logic resources while the second circuit is oriented to the implementation in UMC-90 nm CMOS standard cell technology.

There was another work presented by **Carlos Gonzalez et. al. (2013),** [3] on hyper spectral imaging that is a growing area in remote sensing in which an imaging spectrometer that collects hundreds of images (at different wave lengths) for the same area on the surface of the Earth. These images are extremely high-dimensional, and require most advanced on-board processing algorithms able to satisfy near real time constraints in applications such as mapping of oil spill, wild land fire monitoring and sand chemical contamination, etc. The most widely used techniques for analyzing hyper spectral images is spectral unmixing, which allows for sub pixel data characterization. Since the available spatial resolution in hyper spectral images is typically of several meters, and therefore it is affordable to assume that several spectrally pure substances (called end members in hyper spectral imaging terminology) can be found within each imaged pixel, thus it is particularly very important. In this work they explore the role of hardware accelerators in hyper spectral remote sensing missions and further provided the comparison of the two types of solutions: field programmable gate arrays (FPGAs) and graphics processing units (GPUs). They had implemented and tested the full spectral unmixing chain, using both types of accelerators, in the context of areal hyperspectral mapping application using hyper spectral data collected by NASA's Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS). There work gives a thoughtful impression on the capability and existing challenges of applying these types of accelerators in remote sensing missions, indicating that the reprogram ability of FPGA systems and the low cost of GPU systems may open various new perspectives toward fast on-board and on-the-ground processing of hyper spectral images used in remote sensing applications.

One another different image processing approaches are provided by **Horace Josh et. al., (2013),** [4] that for the Monash Vision Group prosthetic vision device. In a companion work they described that is developing a bionic eye based on the implantation of stimulation tiles on the primary visual cortex. And they lead up to an expected 2014 first in-human trial of this device, potential image processing techniques and intuitive user interfaces need to be developed and checked. The main bottleneck of this approach is the (often reduced) bandwidth connection between the station and satellite, which drastically limits the information that can be sent and processed in real time by **Carlos González et. al., (2012),** [5]. They provide the possible way to overcome this problem by including onboard computing resources able to preprocess the data, reducing its size by magnitude orders. Thus Reconfigurable field-programmable gate arrays (FPGAs) are most promising platform that allows hardware /software codesign and the potential to provide powerful onboard computing capability and flexibility at the same time of frame. And also FPGAs can implement custom hardware solutions; they can reach very high performance levels of degree.

3. METHODOLOGY

Image/Video processing algorithms are normally classified into one of three modes: started with Low-level algorithms always operate on individual pixels or neighborhoods pixel data especially for concurrency. Then going upto intermediate-level algorithms either convert/change pixel data into a different representation depending on the application need, such as coordinate or chain code, a histogram, or to operate on one of any higher representations that previously done. And at last the High-level algorithms aim to extract meaning from the image using information from the previous defined levels. This may be in the form of adding, rejecting a component or identifying where an object is within an image. While coming from low to the high-level data representations there are a significant decrease in credulous parallelism due to the change from pixel data to more informative representations. And also there is a reduction in the sufficient amount of data that must be processed, thus allowing more time to do the processing.

The FPGAs are most adaptable, due to their internal structure, and provide ease of use for computationally intensive tasks which form the vast majority of low and intermediate-level operations for image/video processing. Due to this the large data sets and regular repetitive nature of the operations can be easily imposed. For this reason it has been become tradition in most of the systems for the FPGAs to handle the low-level operations and then pass to the microprocessor or microcontroller to process the data to which then executes the high-level operations. Now with the advent of newer technologies, the size of FPGAs are increasing relatively, so it is now possible to implement processor architectures on the reconfigurable fabric, which means that the FPGA can form the core of the system which can be further used in any processing applications.

The application of image or video processing in FPGAs is a rapidly growing research area due to the recent increases in the power and structures of these programmable devices. Although there are various challenges to implement the working algorithms despite of the potential gains in speed, make it an attractive topic for research in image/video processing. Due to these factors there is a rapid growth in the field of VLSI especially while dealing with FPGAs, mostly newcomers consider simply porting an existing software algorithm to an FPGA implementation.

This Project basically aims to help those researchers who wishing to use FPGAs to accelerate image/video processing algorithms through some of the basics, and provide a range of techniques that result in an efficient implementation, both algorithmatically, computationally and in terms of resource requirements and utilizations. This work will start from the basics of the image that can be found in any type of storage media or system file(s) etc.

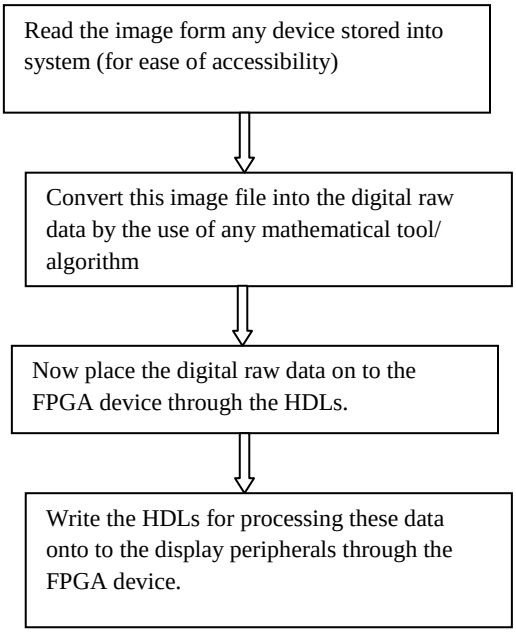

Fig.1: Block Diagram of Proposed Work.

3.1 Implementation

The steps involved for this work is like below:

A. Select an image from a media file stored into system (for ease of access) in either JPEG/BMP/ GIF format.

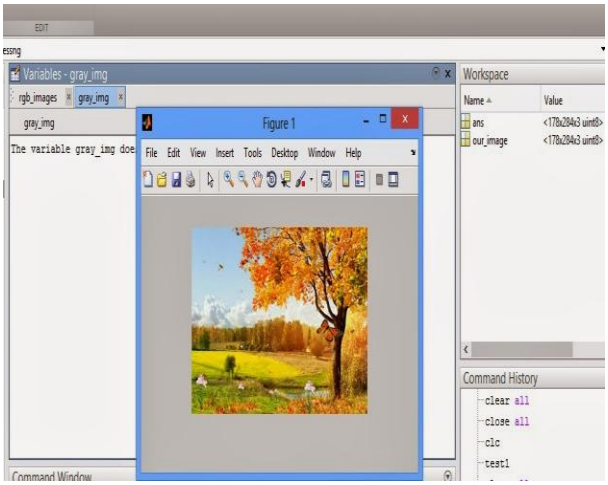

Fig.2: Image read from MATLAB

B. Now the second step would be involved in writing the algorithm for converting these image data into the digital data as the most HDLs behaves very well with digital information.

C. There are various algorithm / procedure available to perform this in this particular work we would prefer MATLAB to convert those image information into the raw digital data which is often said to as the "Coefficient File" with extension ".COE" for Xilinx FPGAs and ".MIF" for Altera FPGAs.

```
memory_initialization_radix=16;
memory_initialization_vector=
201F, A21F, E420, 6681, 1801, 5A2F, 993F, 301D,
7283, B81D, 201E, C000, F025, 201C, 0201, F025,
F025, 0000, 0000, 0000, 0000, 0000, 0000, 0000,
0000, 0000, 0000, 0000, 0000, 0000, 0000, 0000,
1234, 3022, 4321, DEAD, 1111, DEAD, DEAD, 3028,
DEAD, 300D, 0001;
```

Fig 3: COE format of an Image

D. After generating the Coe file of an image the next step is to provide these information into the FPGA ROM. For this purpose we will Xilinx Core generator will be use. Xilinx core is nothing but a Logic Core System that generates and delivers cores optimized for targeted Xilinx FPGAs. CORE Generator is mainly used to create high density, high performance designs in Xilinx FPGAs in very less time.

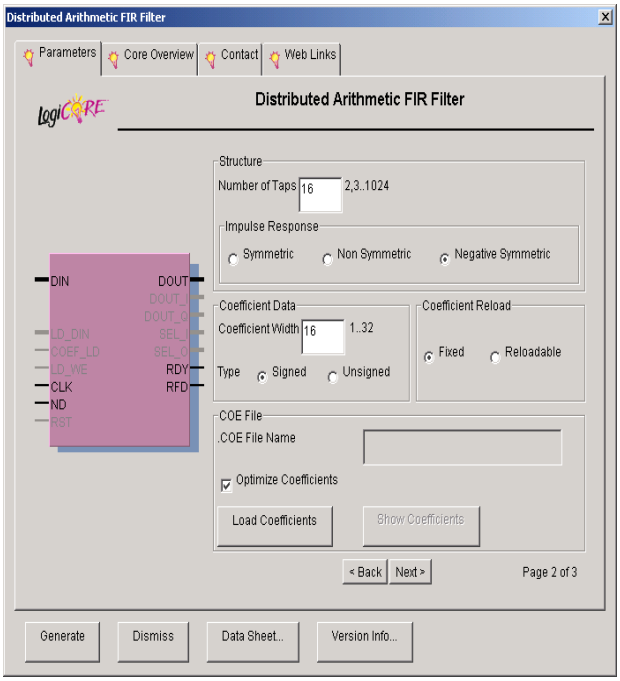

Fig 4: Core Generator block form Xilinx ISE

E. Now the foremost step would be to write the HDL for VGA connector.

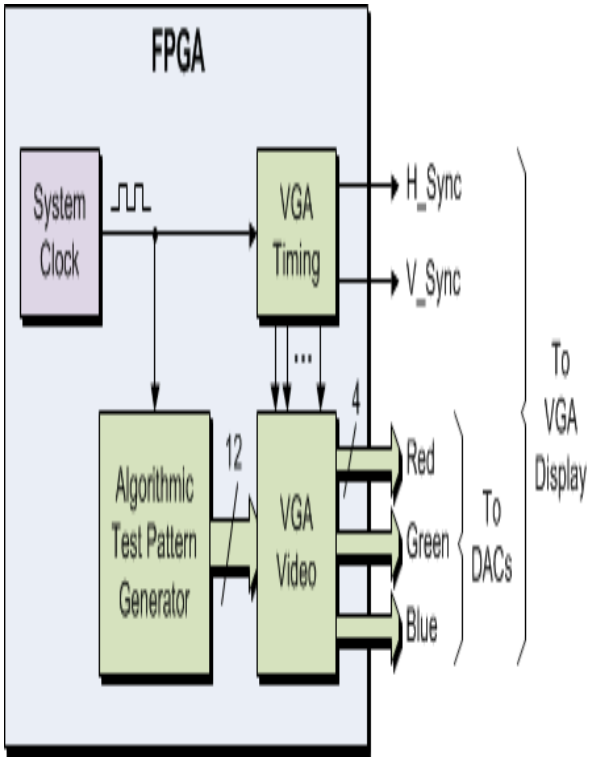

Fig5: FPGA to VGA connector signal interfacing

4. RESULT & DISCUSSION

The reference image for this work is Bart Simpson. The size of image is totally depends on the available Memory size available on the specified FPGA device. The image upto 148 Kb can be used to display on any VGA with good reflections on screen for Xilinx Spartan-3Edevice.

Fig6: Reference Image

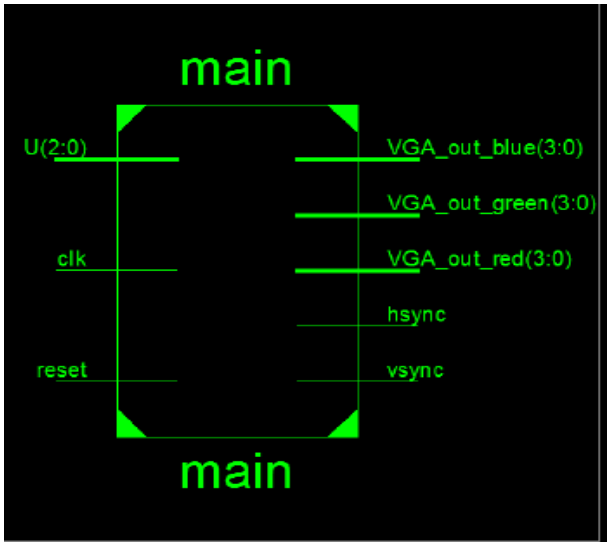

Fig7: RTL view of Our work

The above figure shows the Register Transfer level of the complete design. This RTL view has been carried out from the Xilinx ISE tool after synthesis process. This figure shows the major Input/output pins/signals for real world communication/handshaking. In this, the CLK and Reset are two input pins for the synchronous and initialization of each and every part of the circuit, respectively. While U(2:0) is a 3-bit user input ,works as a input switch, for generating the different colors depends on the choice of the switch patterns.

On the other hand Blue, Red, Green are the various output signals for generating the colors on VGA monitor. Hsync and Vsync are for horizontal and vertical synchronous pulses for VGA signals. These signals are going to be connected directly with the VGA connector pins in order to Interface any VGA monitor with the FPGA board.

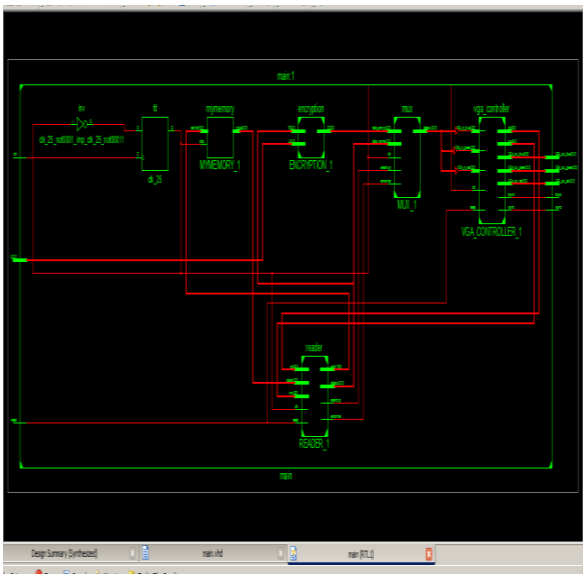

Fig8: Internal RTL view

4.1 Experimental Setup

In order to setup the workbench the VGA monitor (it might be a TFT screen or Computer monitor), Xilinx FPGA device as a Hardware and XILINX ISE as a software tool for Synthesis, Simulation and Implementation.

Fig9: Spraten3E starter Kit

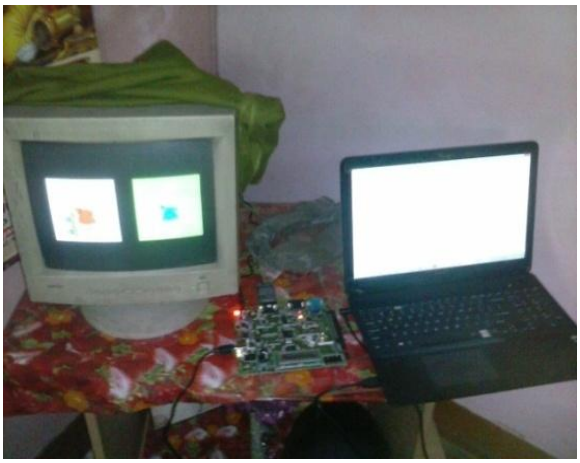

Fig10: Complete Hardware Setup

The FPGA board is the main medium through which the the VGA signals are carried to VGA monitor. The Laptop/computer is the main source on which the complete HDL source is running. After generating the BIT file in the Laptop/PC the corresponding BIT file is dumped onto the FPGA board through Xilinx Impact Tool. Once downloaded the BIT file, simply connect the VGA cable on to the VGA connector part of Board and turn on the TV or Monitor. The Monitor will show now the Two Images. Let's say left hand image and right hand image. The left hand image is nothing but the original image i.e. reference image while right hand image is the duplication of the reference image. As shown in the next figures that the color of right hand image is changing depending upon the button pressed by the user on the board.

Fig 11: Final Result 1

Fig 12: Final Result 2

Fig 13: Final result 3

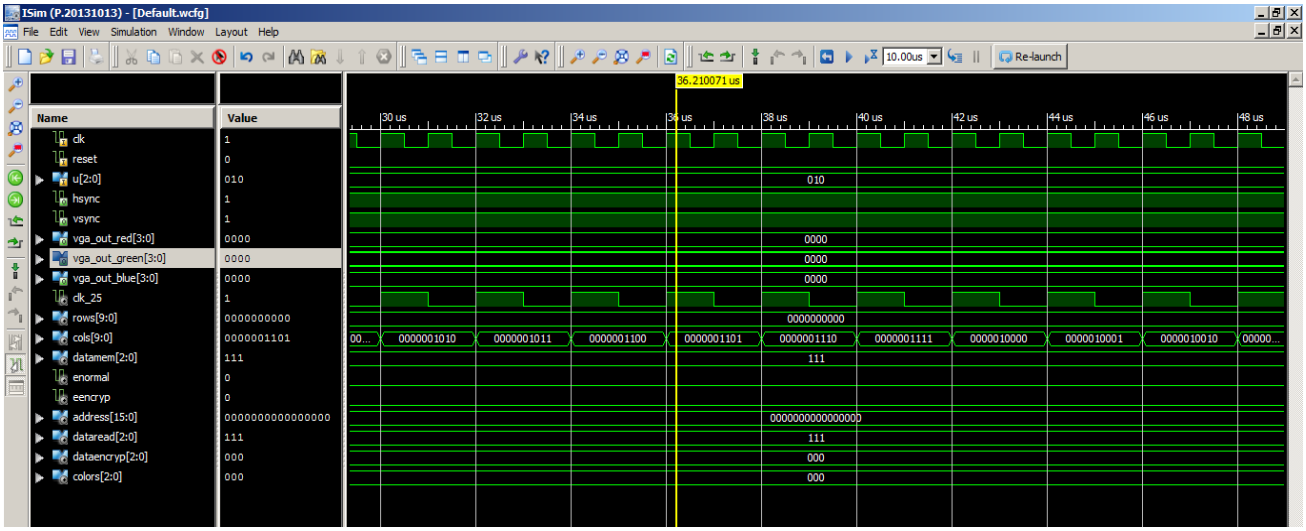

Fig.14 : Simulation Waveforms

5. CONCLUSION

The work has been successfully carried out using Xilinx SIE , Spartan 3E FPGA board and a VGA Monitor. The work can be further accelerate to processing with the Real Time Video or Moving Picture(s) implementations. But for this, one must have to use the reusable code efficiently in order to increase the software development productivity. However, for video or animated pictures processing there is need a lots of Filtering and larger ROM area. The more filtering need more DSP slices and hence more Area required. Also for large memory bandwidth the more burden on the budget of the project. However, still there are lots of happening in this domain is going on. The use of Simulink blocks with the Hardware-Software co-simulation may reduce the burden of writing the large codes as well as the design cycle time. Also the designer needs to be more efficient and specific while working on this type of application(s) especially on real Time embedded Platform.

6. REFERENCES

[1] Michael Schaeferling et. al., "ASTERICS - An Open Toolbox for Sophisticated FPGA-Based Image Processing" embedded world Conference 2015.

[2] Mariangela Genovese and Ettore Napoli, "ASIC and FPGA Implementation of the Gaussian Mixture Model Algorithm for Real-Time Segmentation of High Definition video" IEEE transactions on very large scale integration (VLSI) systems, vol. 22, no. 3, march 2014

[3] Carlos Gonzalez et. al., " Use of FPGA or GPU-based architectures for remotely sensed hyper spectral image processing" INTEGRATION, theVLSIjournal46(2013)89–103.

[4] Horace Josh et. al., " Psychophysics Testing Of Bionic Vision Image Processing Algorithms Using An FPGA Hatpack" 978-1-4799-2341-0/13/$31.00 ©2013 IEEE.

[5] Carlos González et. al., "FPGA Implementation of the N-FINDR Algorithm for Remotely Sensed Hyperspectral Image Analysis" IEEE transactions on geoscience and remote sensing, vol. 50, no. 2, February 2012.

[6] G. Bradski, "The OpenCV Library," Dr. Dobb's J. Software Tools, vol. 25, pp. 120-126, 2000.

[7] J. Dı´az, E. Ros, F. Pelayo, E. Ortigosa, and S. Mota, "FPGA-Based Real-Time Optical-Flow System," IEEE Trans. Circuits and Systems for Video Technology, vol. 16, no. 2, pp. 274-279, Feb. 2006.

[8] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," Int'l J. Computer Vision, vol. 60, no. 2, pp. 91-110, 2004.

[9] S. Sabatini, G. Gastaldi, F. Solari, K. Pauwels, M. Van Hulle, J. Dı´az, E. Ros, N. Pugeault, and N. Kru¨ ger, "A Compact Harmonic Code for Early Vision Based on Anisotropic Frequency Channels," Computer Vision and Image Understanding, vol. 114, no. 6, pp. 681- 699, 2010.

[10] A. Bruhn, J. Weickert, and C. Schnorr, "Lucas/Kanade Meets Horn/Schunck: Combining Local and Global Optic Flow Methods," Int'l J. Computer Vision, vol. 61, no. 3, pp. 211-231, 2005.

[11] A. Bruhn, J. Weickert, T. Kohlberger, and C. Schnoerr, "A Multigrid Platform for Real-Time Motion Computation with Discontinuity-Preserving Variational Methods," Int'l J. Computer Vision, vol. 70, no. 3, pp. 257-277, 2006.

[12] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani, "Hierarchical Model-Based Motion Estimation," Proc. European Conf. Computer Vision (ECCV), pp. 237-252, 1992.