AN IMPROVED OBJECT RECOGNITION SYSTEM DESIGN USING BACKGROUND SUBTRACTION ALGORITHM

Salunke Kaweri Nanasaheb*1, Ms. D. Asha2

1M.Tech, Mallareddy College of Engineering & Technology, Secunderabad, Telangana, India.
2Asst. Prof, Mallareddy College of Engineering & Technology, Secunderabad, Telangana, India.

ABSTRACT

For various computer vision applications, background subtraction (BS) is a “quick and dirty” way of localizing moving objects in a video shot by a static camera. In this paper a new technique for developing a hardware for background subtraction with morphological filter using Spartan 3 EDK FPGA. Field Programmable Gate Array (FPGA) has become a new device for implementation of algorithms of video image process applications. In Spartan3 EDK we IMPLEMENTATION of algoirthms through pipelined architecture through the soft core processor Micro Blaze which in deed used for developing a Hardware structure to Image Processing Applications.

Keywords: FPGA, Filtering, Image processing.

1. INTRODUCTION

Field Programmable Gate Arrays (FPGA) are majorly used as a reconfigurable device, which can used in the field of Image Processing. FPGA generally consists of large no of digital components such as look up tables, logic gates, flip-flops and many more, and its consists of memory, and all there are interconnected through many interconnecting wires. All of the logic in an FPGA can be rewired, or reconfigured, with many different designs and according to the own needs. Normally Image Processing application can be implemented by using MATLAB software but in this paper our Background subtraction algorithm was implemented by using Spartan 3 FPGA Which consists Micro blaze processor which increases the speed of operation and it consists of high no of MAC units compare to the DSP processors so that we can achieve the speed of operation in the FPGA. The main process was because the software results are not accurate than the Hardware results to implement a hardware to existing Image Processing applications we are coming for FPGA implementation. In this project a high configurable Micro blaze processor was used, our algorithm was written in the system C coding an synthesized using the XILINX Platform Studio 10.1 and our output are seen through VB application which reads the pixels values of the image that comes from the FPGA to computer through UART communication.

Background subtraction could be a technique within the fields of image processing in the visual surveillance vision whereby an image's foreground is extracted for additional process (object recognition etc.), typically an image's regions of interest are objects (humans, cars, text etc.) in its foreground, once the stage of image preprocessing (which could embody image denoising etc.) object localisation is needed which can create use of this method. Background subtraction could be a wide used approach for detective work moving objects in videos from static cameras. The explanation within the approach is that of detective work the moving objects from the distinction between this frame and a system, typically known as “background image”, or “background model” [1]. Background subtraction is generally done if the image in question could be a part of a video stream.

Background subtraction could be a category of techniques for segmenting out objects of interest in an exceedingly scene for applications like police investigation. There area unit several challenges in developing a decent background subtraction algorithmic rule. First, it should be strong against changes in illumination. Second, it ought to avoid detection non-stationary background objects and shadows solid by moving objects. A decent background model ought to additionally react quickly to changes in background and adapt itself to
accommodate changes occurring within the background like moving of a stationary chair from one place to a different. It ought to even have a decent foreground detection rate and also the time interval for background subtraction ought to be period.

2. THE REVIEWED APPROACH

The approach reviewed in the paper is

- Background subtraction
- Applying Edge Detection on the subtracted Image
- Applying morphological filter to the above step to remove noise
- Again applying a linear filter technique

![Fig 1: Flow of Implementation](image)

We consider two images from the Video sequences then we convert the two image files into header files ten we add both the header file and Source C code file XPS then we dump the bit stream into FPGA and we view output Result through VB display.

**Background subtraction:** Background subtraction method is a general method of motion detection method which uses the difference of the current image and the background image to detect moving objects. The key of this method is the initialization and update of background image and detection of moving object is also accurate.

**Edge Detection**

**Edge Detection Techniques**

Majorly Edge detection can be done using three operators

- Prewitt operator
- Sobel Operator
- Canny Operator

In this paper we are going to implement Sobel operator for finding an Edge detected image.

**Sobel Operator**

The operator consists of a pair of 3×3 convolution kernels as shown in Fig 1. One kernel is simply the other rotated by 90°.
These kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (call these $G_x$ and $G_y$). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by:

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

Typically, an approximate magnitude is computed using:

$$|G| \approx |G_x| + |G_y|$$

which is much faster to compute.

The angle of orientation of the edge (relative to the pixel grid) giving rise to the spatial gradient is given by:

$$\theta = \arctan(G_y/G_x)$$

### Image Enhancement Using Morphological Filter

The goals of image enhancement include the improvement of the visibility and perceptibility of the various regions into which an image can be partitioned and of the detectability of the image features inside these regions. These goals include tasks such as: cleaning the image from various types of noise; enhancing the contrast among adjacent regions or features; simplifying the image via selective smoothing or elimination of features at certain scales and retaining only features at certain desirable scales. While traditional approaches for solving the above tasks have used mainly tools of linear systems, there is a growing understanding that linear approaches are not well suitable or even fail to solve problems involving geometrical aspects of the image. Thus there is a need for nonlinear approaches. A powerful nonlinear methodology that can successfully solve the above problems is mathematical morphology.

The to processing in the morphology are

1. Dilation
2. Erosion

#### Dilation

This dilation is essential for the removal of the clouded, catar... other drugs, we can use different, longer-lasting dilation eye drops or micro-hooks to keep the pupil completely dilated during surgery. The dilation and curettage procedure is called a D&C. The D stands for dilation, which means enlarging. Curettage (the C) means scraping. Together, this procedure involves expanding or enlarging the entrance of a woman’s uterus so that a thin, sharp instrument can scrape or suction away the lining of the uterus and take tissue samples.

#### Erosion

**Common Names: Erode, Shrink, Reduce**
Brief Description

Erosion is one of the two basic operators in the area of mathematical morphology, the other being dilation. It is typically applied to binary images, but there are versions that work on grayscale images. The basic effect of the operator on a binary image is to erode away the boundaries of regions of foreground pixels (i.e., white pixels, typically). Thus areas of foreground pixels shrink in size, and holes within those areas become larger.

3. MICROBLAZE PROCESSOR DESIGN

Field-Programmable Gate Arrays (FPGA's) are unit versatile and reusable high-density circuits that may be simply re-configured by the designer, sanctioning the VLSI style/validation/simulation cycle to be performed additional quickly and fewer expensive. Increasing device densities have prompted FPGA makers, like Xilinx and Altera, to include larger embedded parts, as well as multipliers, DSP blocks and even embedded processors. one in every of the recent subject enhancements within the Xilinx Spartan, Virtex family architectures is that the introduction of the MicroBlaze (Soft IP) and PowerPC405 hard-core embedded processor. The MicroBlaze processor may be a 32-bit Harvard Reduced Instruction Set computer (RISC) design optimized for implementation in Xilinx FPGAs with separate 32-bit instruction and knowledge buses running at full speed to execute programs and access knowledge from each on-chip and external memory at an equivalent time.

3.1 Background

The backbone of the design could be a single-issue, 3-stage pipeline with thirty two all-purpose registers (does not have any address registers just like the Motorola 68000 Processor), Associate in Nursing Arithmetic Logic Unit (ALU), a shift unit, and 2 levels of interrupt. This basic style will then be organized with additional advanced options to tailor to the precisewants of the target embedded application such as: barrel shifter, divider, multiplier, single exactitude on floating-point unit (FPU), instruction and information caches, exception handling, rectify logic, quick Simplex Link (FSL) interfaces et al.

This flexibility permits the user to balance the specified performance of the target application against the logic spacevalue of the soft processor MicroBlaze additionally supports reset, interrupt, user exception, and break hardware exceptions. For interrupts, MicroBlaze supports only 1 external interrupt supply (connecting to the Interrupt input port). If multiple interrupts are required, Associate in Nursing interrupt controller should be accustomed handle multiple interrupt requests to MicroBlaze shown in fig 2. An interrupt controller is offered to be used with the Xilinx Embedded Development Kit (EDK) code tools. The processor can solely react to interrupts if the Interrupt alter (IE) bit within the Machine standing Register (MSR) is ready to one. On Associate in Nursing interrupt the instruction within the execution stage can complete, whereas the instruction within the decipher stage is replaced by a branch to the interrupt vector (address Ox 10). The interrupt address (the laptop related to the instruction within the decipher stage at the time of the interrupt) is mechanically loaded into all-purpose register. additionally, the processor additionally disables future interrupts by clearing the id est bit within the MSR. The id est bit is mechanically set once more onecorporal punishment the RTID instruction.

Fig 2: MicroBlaze architecture block diagram

Due to the advancement within the fabrication technology and therefore the increase within the density of logic blocks on FPGA, the utilization of FPGA instr'srestricted any longer to debugging and prototyping digital electronic circuits. as a result of the big similarity doable on FPGA and therefore the increasing density of logic.
blocks, it's getting used currently as a replacement to ASIC solutions during a few applications wherever the time to plug is vital and conjointly entire embedded processor systems square measure enforced on these devices with soft core processors embedded within the system. Soft cores square measure technology freelance and uniquely simulation and temporal order verification when synthesized to a target technology. This reduces the planning (the look) cycle development time by a significant issue as compared to the event cycle for a tough core processor and has the advantage of customizing the soft core design for a selected application.

3.2 Features

The MicroBlaze soft core processor is very configurable, permitting you to pick a particular set of options needed by your style.

The fastened feature set of the processor includes:

- 36-bit general purpose registers
- 32-bit instruction word with 3 operands and 2 addressing modes
- 32-bit address bus
- Single issue pipeline

In addition to those fastened options, the MicroBlaze processor is parameterized to permit selective sanctioning of extrapracticality. Older (deprecated) versions of MicroBlaze support a set of the facultative options delineate here. So, the newest (preferred) version of MicroBlaze (v7.00) supports all choices. Xilinx recommends that each one new styles use the newest well-liked version of the MicroBlaze processor.

3.3 Pipeline Architecture

MicroBlaze execution is pipelined. for many directions, every stage takes one clock cycle to finish. Consequently, the quantity of clock cycles necessary for a fixed instruction to finish is adequate to the quantity of pipeline stages, and one instruction is completed in each cycle. A number of directions need multiple clock cycles within the execute stages to finish, this is often achieved by stalling the pipeline.

When death penalty from slower memory, instruction fetches might take multiple cycles. This extra latency directly affects the potency of the pipeline. MicroBlaze implements an instruction prefetch buffer that reduces the impact of such multi-cycle instruction memory latency. Whereas the pipeline is stalled by a multi-cycle instruction within the execution stage, the prefetch buffer continues to load ordered directions, once the pipeline resumes execution, the fetch stage will load new directions directly from the prefetch buffer rather than awaiting the instruction operation to finish.

4. IMPLEMENTATION

4.1 Xilinx Platform Studio

The Xilinx Platform Studio (XPS) is that the development atmosphere or user interface used for planning the hardware portion of your embedded processor system. B. Embedded Development Kit Xilinx Embedded Development Kit (EDK) is associate integrated software system tool suite for developing embedded systems with Xilinx MicroBlaze and PowerPC CPUs. EDK includes a spread of tools associated applications to help the designer to develop associate embedded system right from the hardware creation to final implementation of the system on an FPGA. System style consists of the creation of the hardware and software system parts of the embedded processor system and also the creation of a verification element is elective. A typical embedded system style project involves: hardware platform creation, hardware platform verification (simulation), software system platform creation, software system application creation, and software system verification. Base System Builder is that the wizard that automatically generate a hardware platform in keeping with the user specifications that's defined by the MHS (Microprocessor Hardware Specification) file. The MHS file defines the system design, peripherals and embedded processors. The Platform Generation tool creates the hardware platform mistreatment the MHS file as input. The software system platform is defined by MSS (Microprocessor software system Specification) file that defines driver and library customization parameters for peripherals,
processor customization parameters, customarray one hundred ten devices, interrupt handler routines, and different software system connected routines. The MSS file is associate input to the Library Generator tool for personalisation of drivers, libraries and interrupts handlers.

The creation of the verification platform is facultative and is predicated on the hardware platform. The MHS file is taken as Associate in Nursing input by the Simgen tool to make simulation files for a particular machine. 3 varieties of simulation models will be generated by the Simgen tool: behavioural, structural and temporal arrangement models. another helpful tools on the market in EDK ar Platform Studio that provides the GUI for making the MHS and MSS files. produce / Import IP Wizard that permits the creation of the designer's own peripheral and import them into EDK comes. Platform Generator customizes and generates the processor system within the sort of hardware netlists. Library Generator tool configures libraries, device drivers, file systems and interrupt handlers for embedded processor system. Bitstream Initializer tool initializes the instruction memory of processors on the FPGA shown in figure 2. antelope Compiler tools ar used for collection and linking application executables for every processor within the system [6]. There ar2 choiceson the market for debugging the appliance created victimisation EDK namely: Xilinx micro chipcorrect (XMD) for debugging the appliance package employing amicro chipcorrect Module (MDM) with the embedded processor system, and package program that invokes the package programme appreciate the compiler getting used for the processor. C. package Development Kit Xilinx Platform Studio package Development Kit (SDK) is Associate in Nursing integrated development atmosphere, complimentary to XPS, that's used for C/C++ embedded package application creation and verification. SDK is made on the Eclipse opensource framework. Soft Development Kit (SDK) may be a suite of tools that allows you to style a package application for elite Soft IP Cores within the Xilinx Embedded Development Kit (EDK). The package application will be written during a "C or C++" then the entire embedded processor system for user application are completed, else correct&download the bit file into FPGA. Then FPGA behaves like processor implemented on it in a Xilinx Field Programmable Gate Array (FPGA) device.

5. TABULATION RESULTS

The Algorithm is implemented in Microblaze Processor and the results are furnished in the tabulation below.

<table>
<thead>
<tr>
<th>Device utilization summary:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selected Device: 3s2000eq144-4</td>
</tr>
<tr>
<td>Number of Slices: 1663 out of 1940 85%</td>
</tr>
<tr>
<td>Number of Slice Flip Flops: 3119 out of 3940 85%</td>
</tr>
<tr>
<td>Number of LUTs: 2971 out of 3860 77%</td>
</tr>
<tr>
<td>Number used as logic: 3415</td>
</tr>
<tr>
<td>Number used as Shift registers: 207</td>
</tr>
<tr>
<td>Number used as RAMs: 256</td>
</tr>
<tr>
<td>Number of IOs: 55</td>
</tr>
<tr>
<td>Number of bonded I/Os: 62 out of 97 65%</td>
</tr>
<tr>
<td>20X Flip Flops: 12</td>
</tr>
<tr>
<td>Number of BRAMs: 4 out of 12 33%</td>
</tr>
<tr>
<td>Number of MULs: 3 out of 22 14%</td>
</tr>
<tr>
<td>Number of DLLs: 4 out of 8 50%</td>
</tr>
<tr>
<td>Number of CBRs: 1 out of 4 25%</td>
</tr>
</tbody>
</table>

Fig 4: Synthesis report
Fig 5: Background Image reading in VB window

Fig 6: Foreground Image

Fig 7: Background subtracted image

Fig 8: Morphological Output
6. CONCLUSION

In this work a moving object motion detection on background subtraction algorithmic rule was developed. This system works on a period pipelined flow on the Micro Blaze architecture of Spartan3 EDK. On the opposite hand, synthesis results show that space consumption is low, using simply 100 percent of logic components of FPGA for moving object detection system, permitting the implementation of this method over inexpensive FPGAs.

REFERENCES


