ECE532 Project Proposal

Project Title:

Live action first person shooter game

Project Team:

Patrick Judd

Bao Le

Ian Katsuno

Project Description

The goal of our project will be to create a "live action" first person shooter game. It will use a stationary video camera connected to the Xilinx XUP Virtex 2 Pro FPGA board to capture the live video feed. This live video feed will be modified and displayed on a computer monitor using the VGA output of the FPGA board. Using a mouse connected to the FPGA, players of the game will look at the computer monitor and shoot at “targets” which move within the camera's field of view. To start the game the user must first take a reference shot of an empty section of a room to serve as the stage for the game. Any person who then appears on the stage in the cameras field of view will become a “target.” If the player is successful in 'shooting' (clicking on) a target then the target will be removed from the screen. Thus, it is similar to a first person shooter except for the fact that the both the game environment and the game targets will be live-action. This project is inspired by a past project which implemented 'invisibility' for specific colors by replacing specific pixel color values in a video feed.

The general strategy in architecting our design is to perform all image analysis and manipulation in hardware and everything else in software. User input, target tracking, and hit detection will be performed in software by a Microblaze processor on the FPGA while the target detection, target removal (for when the user 'hits' a target), and video coding/decoding will be performed strictly in hardware on the FPGA.

To ensure that we will be able to get an initial version of our system integrated and working sooner we will impose the restriction that there will be only one target on the screen at a time. Once the initial version of the system is working properly we will expand it to support multiple targets.

Farther into the project, depending on time constraints, we may attempt to add a light-gun user interface. The light gun would be similar to the Nintendo Zapper used in the game "Duck Hunt." Instead of using the mouse to aim at targets, the player would aim the light gun at the screen. The light gun would work by sending a signal to the FPGA when the trigger is pulled. The FPGA would then cause the screen to flash black, except for the target, which would flash white. During the period of the flash, the light sensor in the gun would either detect black or white, depending if it was aiming at the target or not, and send that signal to the FPGA to determine if the target was hit or missed.

Functional Requirements

The functional requirements define the minimum functionality that our project must have to be deemed successful. We have defined the following functional requirements:

1. The system must be able to handle 30 frames per second of video
2. The system must be able to handle a video feed with a 640x480 resolution
3. The system must be able to display the input video frames which have been modified by the internal logic
4. The system must be able to identify a “target” in the input video feed
5. The system must also be able to track the movement of the “target.” The system must decide if an identified “target” is the same “target” identified in the previous frame, or if it is a new target in the frame.
6. The system must be able to display the mouse cursor/crosshair on the output image
7. The system must allow targets to be selected using the mouse

Features

The features are the extra functionality we will add to our design once all the functional requirements are met. We have defined the following features:

- The system will support multiple targets in the frame at the same time
    - The system will track the position of these targets separately so that targets can be removed from the output image individually
- The system will use a light gun as an input device instead of a mouse
    - The processor will read the light guns interface
    - The system will produce the black and white flash needed by the light gun

Acceptance Criteria

For each functional requirement we have defined a simple acceptance test to determine whether the system meets the requirement. Many of the functions are performed by a single component and the corresponding acceptance test is intended to validate that component before the whole system is complete.

Function	Acceptance Test
1. 30 frames per second	This function cannot be tested until full system integration since meeting this requirement involves the timing of the whole system. However, timing estimates will be made and constraints will be put on individual components in order to meet the necessary full system timing.
2. 640x480 resolution	Similarly to the first function, this function cannot be tested until full system integration.
3. display modified video image	Display the input video image, but with one color replaced by another.
4. identify target	Draw a box around the target in the output image.
5. track the movement of a target	Output a counter to the LEDs of the number of unique targets detected. The counter should not increase for a target moving across the screen. The target should increase if a target disappears and a new target appears in a later frame.
6. display the mouse cursor	Draw a green cursor overtop of the input video image which follows the mouse movements.
7. select targets with mouse	If the mouse clicks on a target turn the cursor red, if the mouse clicks on the background turn the cursor blue.

Resource Requirements

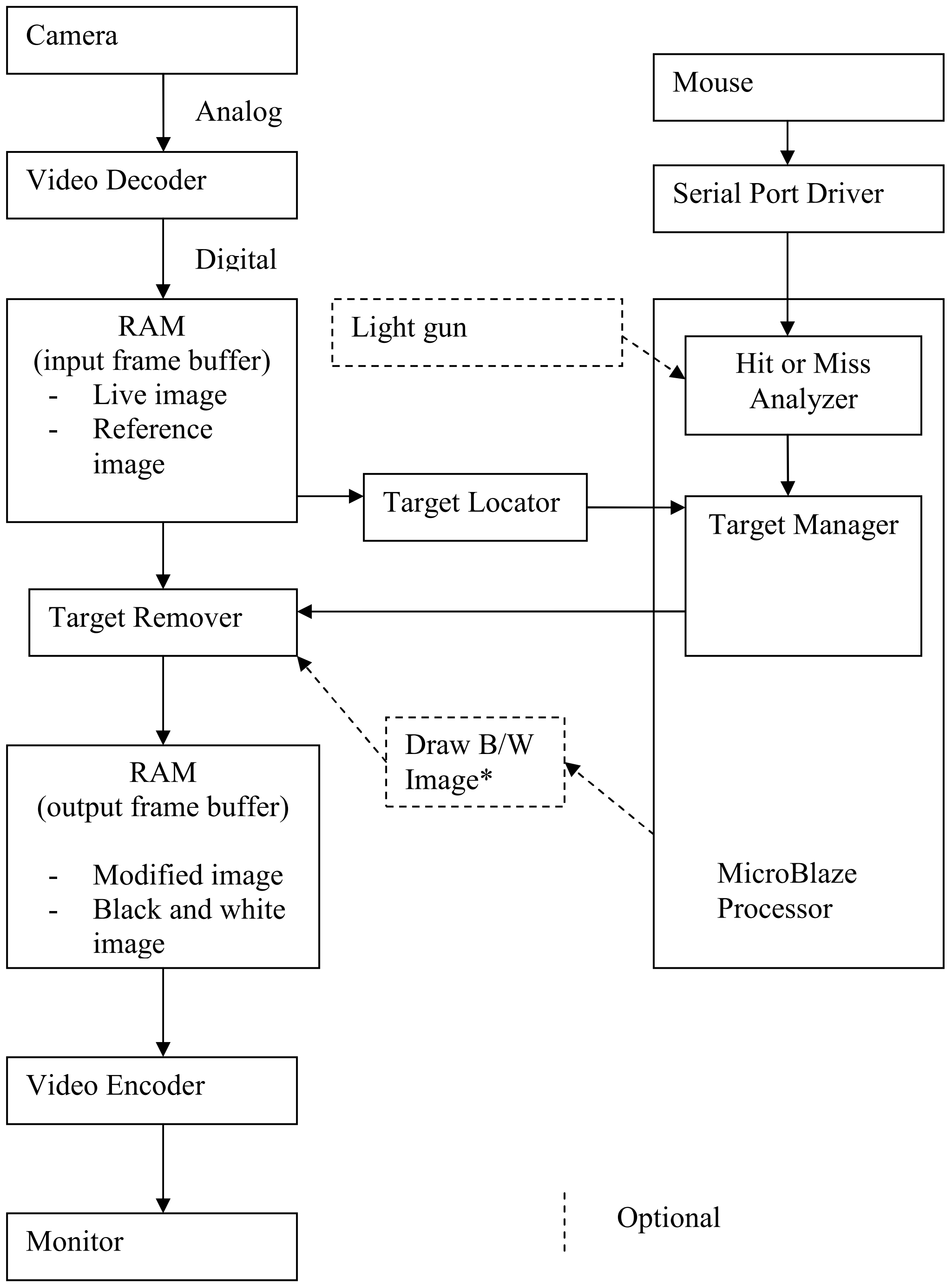
- CRT Monitor
    - The light gun is only able to detect light emitted from a CRT monitor, it will not work with the LCD monitors in the lab
- PS/2 Mouse
- Video Camera with RCA(composite) video output
- Light Gun

Milestones

The following milestones define the tasks we hope to complete for each week:

Feb 9	• Display an image stored in memory on the monitor • Setup the mouse interface o Demonstrate the mouse interface by displaying the mouse coordinates on the LEDs • Create a debug interface on the micro blaze processor o Demonstrate the debug interface by printing a debug statement
Feb 16	• Read image from a camera into memory and display it on the monitor • Implement the Target Remover block • Write code to store the target information and track the target
Mar 2	• Implement the Target Locator block • Draw the cursor on the screen corresponding to the mouse position • Implement hit detection in software • Implement the timeout function in software
Mar 9	• Integration and Testing
Mar 16	• Integration and Testing
Mar 23	• Add multiple target support to system o Modify target locator, target remover and software to support multiple targets
Mar 30	• Add light gun support o Create hardware interface to light gun o Modify hit detection software for light gun o Create block to draw black and white image needed by light gun

Block Diagram



Components

1. Camera: We will use a digital camera with composite video output.

2. Video decoder and encoder: this hardware is supplied with the tool kit, for the VGA module we will use the code that has been developed in the course previous years. This section will be implemented in FPGA chip.

3. RAM/Memory: SDRAM on the board is used as the main memory to store images in both phases, pre-modified (input frames) and post-modified (output frames). Also we will store one initial image which is used as a reference image.

4. Target Locator: This component takes in one frame image at a time from the memory. It then compares the image to the reference image; any objects that are not in the reference image but in the current frame image will be flagged as a target. The coordinates of the target then are sent to the target manager. The target coordinates will represent a rectangular box around the target. This component needs to do video processing; hence we implement it in hardware.

5. Target Manager: The processor is used to manage live, dead and new targets on the screen. First, it receives current target information from the target locator module. It checks to see if a user hit any targets. If the target is dead, it sends a signal to target remover to remove the target. If new target appears on the screen, its information is also stored in this module. If a user hit a target, the target is also transferred to list of dead target, this information is acknowledged from the hit/miss analyzer.

6. Target Remover: Receiving the signal from the target manager, this module will then produce appropriate frame image that later will output to the CRT monitor. A target is removed by copying image data from the reference image instead of the live image into the rectangular region defined by the target coordinates. Similar to the target locator, this component is implemented in hardware.

7. Hit/Miss Analyzer: This component is used to detect if a user hits or misses a target. Both target manager and hit/miss analyzer require significant amount of comparisons hence they will be implemented in software running on the MicroBlaze processor.

8. Mouse + Serial Port Driver: We will use IP that is available on Xilinx tool to implement the driver.

9. Light gun + Draw B/W Image: As explained, when we use the light gun as our input device, we will have to flash the screen black and white so that the light gun can work properly. These components will be added as extra feature and they will be implemented in hardware.

10. CRT Monitor: An output from the video will output the modified image to this monitor.