Digital shooting range

Emmanuel Azuh
Mubarik Mohamoud
6.111 Final project proposal

Overview

There were several 6.111 shooting game projects over the years including the Rifle Arcade game from 2014 and Beat Gunner from 2009. Rifle Arcade and Beat Gunner are quite interesting and immersive games but they both have significant limitations in terms of range. Beat Gunner uses a photodiode as the tip of the gun to determine whether the bullet missed or hit the target based on the brightness of the incident point, the screen goes dark during the clock cycle following a shot except the target stays bright. For Rifle Arcade, they used a camera to detect the tip of the gun which is a small, blue ball then they used a gyroscope to determine the angular rotations of the gun to estimate the point of incident.

In Beat Gunner, there is a hit and there is a miss but nothing in between and the photodiode limits their range to about a meter. For Riffle Arcade, determining the tip of the gun using the camera can be problematic. The camera looks up a blob of a certain color in space, which works well only if the background was picked to avoid false positives. Otherwise, it would require more complex, resource intensive, multistep algorithms like background removal to distinguish the tip from other static objects that may look the same. Additionally the tip of the gun needs to be big for longer distance tracking.

Digital Shooting Range determines the point of incident of a shot in a simple and intuitive way, that allows long range tracking. Like Rifle Arcade, we are using a camera to determine where the player is aiming, but in this case the camera is the tip of the gun which we use to determine the location of the target.

This is intuitive because it is the shooter that's determining where the target is instead of the target determining where the shooter is. And it is simple because all we have to find is the center of the target with respect to the view of the camera, and then we simply determine our score based on the distance between the center of the camera view and the center of the target.

In addition to the simplicity of our approach, we are expecting both an improved detection range and more robust detection. The target is a large screen VGA monitor, which we believe we can detect from distances of up to 6 meters and from a wide range of angles. The unique patterns displayed on our target will improve the reliability of our detection quite a lot.

The primary goal of our design is to implement a reliable detection of the target, have our IR communication transmit and receive states correctly and reliably, and display the target and scores as well as the "bullet holes" on the screen of VGA monitor. Possible extensions we are considering are: creating an immersive background for the target, adding moving targets, adding IMU to predict where the screen may be, given where it in the previous state, adding recoil, and adding sound. Our goal is to create something that resembles the experience of actual shooting range as closely possible (May be one day it will replace range shooting with live bullets and save the hundreds of lives we lose per year in shooting range training).

Design

In deciding our final project, we wanted to work on a mixture of skills we have acquired from labs and new topics. Our skills from lab5B and lab3 will help us with the IR transmission and receiving as well display modules, and we will explore new ideas like image processing along the way.

When we came up with Digital Shooting Range, we knew we wanted to optimize for mobility and accuracy so we can mimic an actual shooting range as closely as possible. Putting the camera on the gun was "killing two birds with one stone". With the camera as the tip of the gun, we can move around and shoot from wide range of angles and distances, but also we will be able to determine where we will be hitting with pixel accuracy.

Implementation

Figure 1

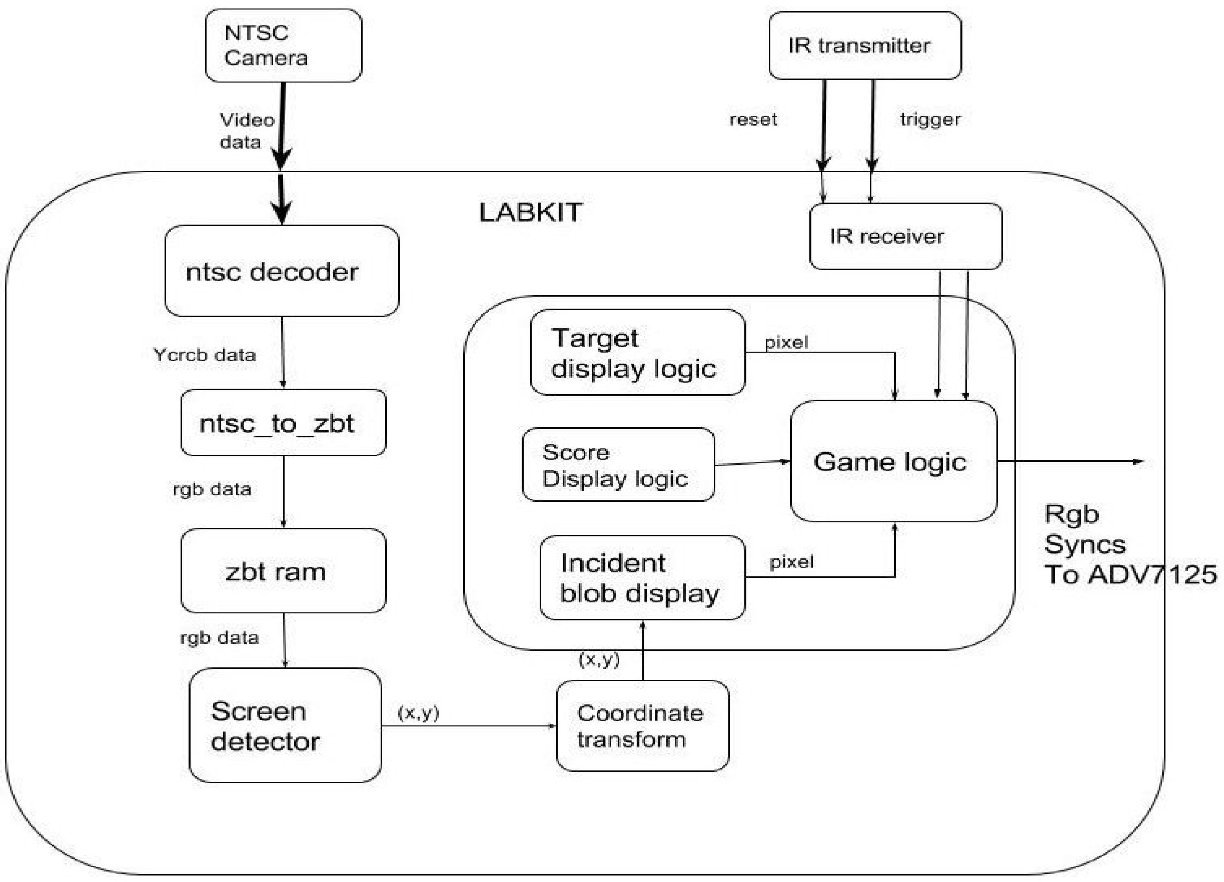


Figure 1 shows the high level block diagram of our implementation with all the major modules and their connections shown. For brief explanation of each module, keep reading.

NTSC Camera

The NTSC camera, which was available in the lab, provides us with analog video data in the YUV (ycrcb) domain as well as synchronization signals. We will be transforming this information to domains we are more familiar with and easier to manipulate like RGB and HSV. We will then use the images in those domains to detect the target. More about the NTSC camera.

NTSC Decoder

The NTSC Decoder module is the interface with the NSTC camera that extracts the video data and the sync signals from the video jack input of the labkit. This module has been implemented already by Javier Castro as part of the NTSC sample verilog available on the 6.111 website.

NTSC To ZBT

This module provides an interface for writing to the Labkit ZBT RAM. It takes video data from the NTSC decoder and stores on the Labkit's ZBT RAM which we can then read that data from and use it for our object detection modules. Most of this module is also available here but we will need to modify it store RGB or HSV data instead of the Black and White which is what it currently outputs.

ZBT RAM Reader

This module which is also available as part of the NTSC sample verilog will be used to read the video data and signals from the ZBT memory. We may or may not need to modify it to tailor it for our needs but any changes will be discussed on the report.

Screen Detector

This will be a multi-module verilog to determine the position of the target in the camera view. The details of the implementation are yet to be decided but we will be manipulate image data in RGB or HSV colorspace to detect objects of known pattern of colors. The final output will be 2D pose indicating the position of the center of the target in camera perspective as well as the height and the width of the target.

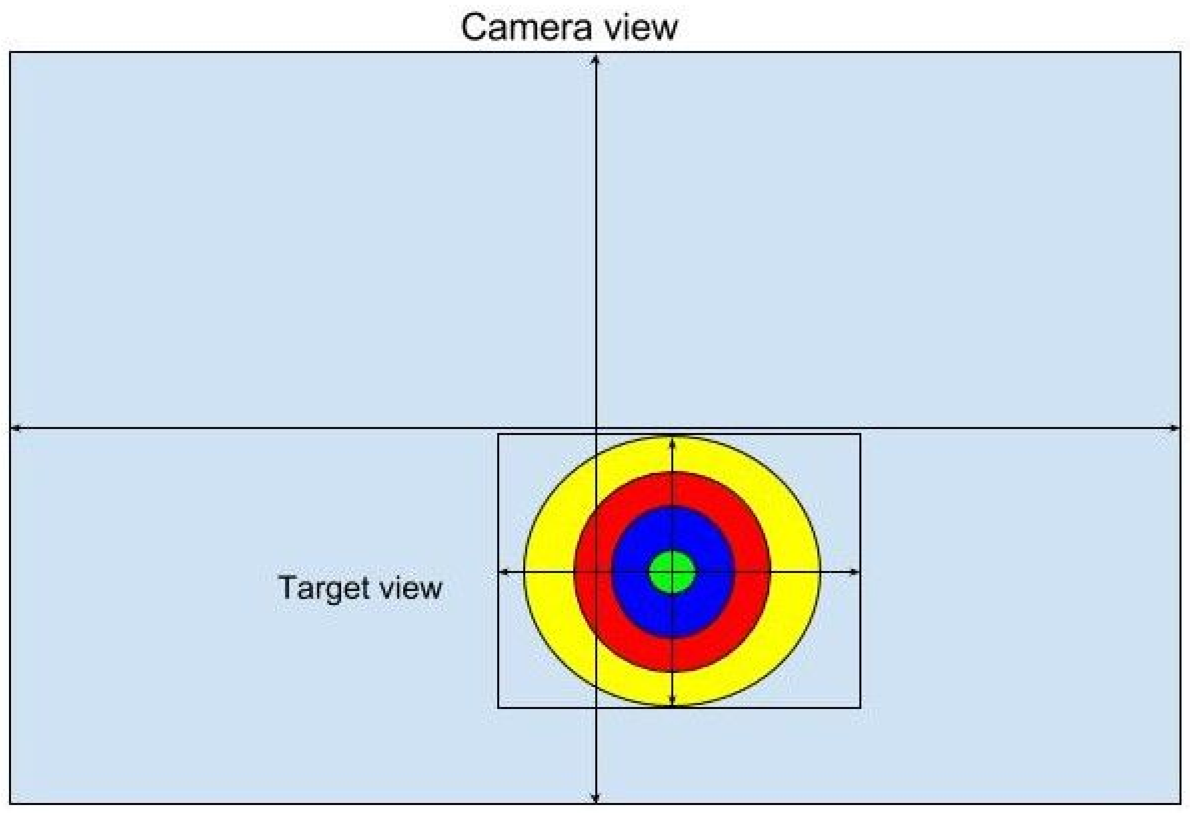
Coordinate Transform

This module takes the final output of the screen detector and does basic coordinate transformation to find the heading of the gun(camera). The following two equations will give us the center point of the camera view, using the parameters outputted by the previous module .

$$y = \frac{h_{cv}}{h_t}\left|y_t - \frac{h_{cv}}{2}\right| \qquad x = \frac{w_{cv}}{w_t}\left|x_t - \frac{w_{cv}}{2}\right|$$

h_{cv} : Height of the camera frame.
h_t : Height of the target with respect to the camera view.
w_{cv} : Width of the camera frame.
w_t : width of the target with respect to the camera view.
(x_t, y_t) : Center of the target.

Figure 2



Incident Blob Display

Module to draw a blob at the position of bullet if the bullet is to land on the target. The blobs will accumulate on the target as long as the same player is shooting. When player is done, they assert the clear switch so their bullet blobs can be erased.

Score Display

The game logic determines the cumulative score of a player and sends the score this module which displays the score. The display will be at the bottom of the screen, overlaying the sprites using alpha-blending.

Target Display Logic

The target display will be implemented to accommodate the image processing module. Preset colors (similar to those in figure 2) in concentric circles will fill the screen. As the image detection algorithm gets better, we may have multiple targets or a moving target.

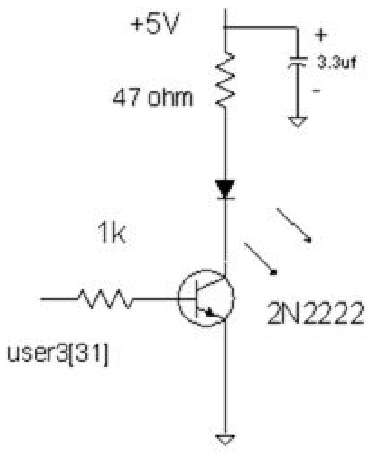
Game Logic

The game logic will bring together all the different parts of the project. Inputs will include the coordinates from the image processing unit, data from the IR receiver, and the scores from the score module. Based on the inputs, the game logic updates the state of the game including the new score, which images to display (blob when appropriate) and a moving target logic (as a stretch goal). The output of the logic module ultimately updates the UI.

IR Transmitter

Vishay TSKS5400S (Infrared Emitting Diode, 950 nm, GaAs) from Lab 5 will be the transmitter for this project. The circuiting for the transmitter will be mostly identical to figure 3. The data stream will have a preamble followed by 11 bits, 10 to describe the x,y position on the screen and the last bit to tell whether clear has been pressed. The implementation will be an 11 bit serial pulse width modulation (the number of bits will be changed if necessary).

Figure 3



IR receiver

Infrared receiver (RPM7140-R) will be used to detect and read the output of the IR transmitter. The signal received by the receiver is parsed according to the established protocol from the transmitter and and the data is sent over to the game logic module.

Technical Challenges

One potential main challenge could arise using an IMU to predict the position of the target before the vision detection. Using an IMU is a work itself, but the main challenge would be doing something with that prediction. One way we could use that prediction is would be reduce our search when detecting the target. We could be looking at only some regions of the image to find the target, this would mean we have to be really careful with keeping the original structure of the frames and sync signals.

Another challenge would be adjusting the image processing to accommodate a moving target. The initial task of detecting the screen and figuring out the the center of the target becomes more complex. Extra computation power will go into more advanced algorithms to detect the targets. The computations however, will be fast enough relative to the frame update rate from the camera such that we can still afford to implement relatively complex computations. We will also use more clever algorithms that require fewer matrix operations like those presented in the image processing guest lecture.

Timeline and Responsibilities

Timeline:

	10/31	11/7	11/14	11/21	11/28	12/5
IR Transmission	E					
Screen and Color Detection	M	M				
Optimizing Image Processing			M	M		
UI and Game Logic Implementation		E	E	E		
Stretch Goals and Debugging					M,E	M,E

Key: Emmanuel - E, Mubarik -M

IR Transmission: A basic setup to transmit trigger and clear information from the gun to labkit.

Screen and Color Detection: Image processing algorithms to detect stationary and moving targets (as a stretch) to high degree of accuracy.

UI and Game Logic Implementation: An interface to provide targets and display scores, with moving targets as a stretch. Game logic will determine when a game ends, update scores, and in the case of moving targets, keep track of objects moving around.

Stretch Goals: Implement a shooting game with moving targets and add sound. Stretch goals will start as soon as earlier implementations are completed.

Responsibilities:

The implementation of this game has two main parts. The game implementation and image recognition from NTSC camera. Mubarik's main focus will be to create a very reliable image recognition system to provide a range of about 6 meters from the monitor. Emmanuel will handle the implementation of the game logic and interface.

Testing

Testing modules will happen alongside development. Main modules including the coordinate transform and game logic will be tested using test benches. Image recognition modules will be tested with a combination of test benches and making estimates of change in gun direction and the change reported by the module. For the UI, testing will include test benches and verifying the VGA output to the monitor.

Required Resources

The hardware for this project include IR sensor, IR receiver, NTSC camera and an Arduino Nano board. All the hardware with the exception of Arduino are already available at the lab. The arduino will be used to modulate the IR emitted from the gun (since the labkit will be far away). An arduino nano board costs ~ $10.

Conclusion

Digital shooting range is a good pastime to help rifle and pistol lovers do what they love in the comfort of their rooms. We believe we can learn a lot from this project because we will be exposed to image processing in hardware as well as sprite-based game design. The project also has room for stretch goals which will allow to also learn more about signal processing if we are able to add a sound feature. Our hope is to utilize the room for creativity presented in the making of this game to learn about essential trade offs in digital circuits design, while testing the bounds of our ingenuity.