FPGA Camera-Controlled Traffic Lights

Jessica Quaye and Premila Rowles

6.111 Fall 2018

Table of Contents

1. Overview

Almost everybody interacts with traffic lights on a daily basis; whether as a pedestrian or as a driver, we use traffic lights in our daily commute. Most people have unpleasant experiences with traffic lights when they are not optimized; their inability to effectively handle rush hour coupled with large volumes of traffic causes frustration on the road.

In order to optimize traffic on roads, our project will create a traffic light controller that takes in information from cameras observing the room and synthesizes the information using the FPGA(field-programmable gate array). This data will be processed, analyzed, and sent to a finite state machine (FSM) that will control the traffic light signals. There are three main problems our project seeks to address. Firstly, the images provide information about the number of cars in the different directions and control traffic signals to prioritize denser regions in order to improve efficiency.
Also, using images provides a means to detect car accidents and send signals for help, especially in countries where communication is not easily accessible.
Finally, the images will help to detect cars which are breaking traffic laws through overspeeding and running red lights.
In addition to controlling the traffic lights, the information obtained is also used to draw a visualization mirroring the street on the Video Graphics Array (VGA) monitor (using color and graphics). This paper provides a detailed description of our project outline, including the main modules, block diagrams and timeline.

2. Design

2.1 Goals

In summary, our baseline goals will focus on object detection and logic controls for the traffic light, expected goals will focus on collision detection and visualization, and stretch goals will offer advanced visualizations.

2.2.1 Baseline

- **FSM:** Given number of cars in each direction, control traffic lights at an intersection.
- **Object Detection:** Given camera input, determine number of cars and corresponding direction on a two-way street.

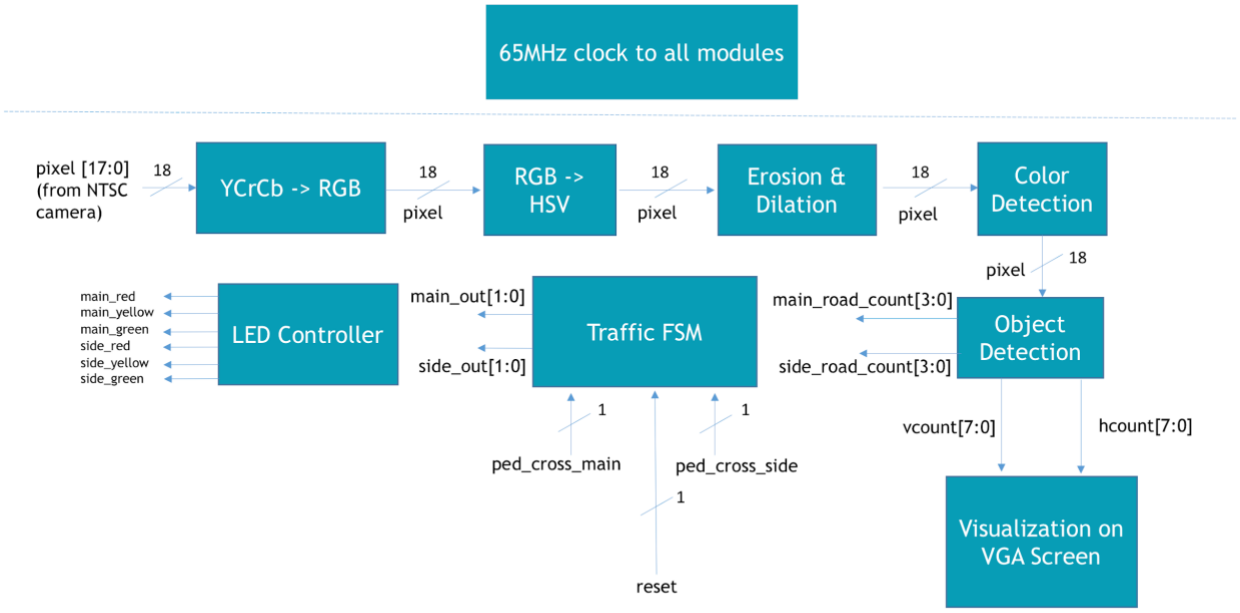
2.2.2 Expected

- **Object Detection:** Detect collisions on the street and send help signal.
- **Visualization:** Draw visualization by mirroring the street on a screen with cars moving and traffic light controls changing. When an accident occurs, the screen will display an ambulance approaching the collision region.
- **FSM:** Add logic to incorporate pedestrian "push-to-walk" signals to the FSM.

2.2.3 Stretch

- **Object Detection:** Detect reckless driving through processing car movement (swerving, overspeeding, etc). If a traffic warden is detected in the middle of a street for more than 5 seconds, turn off traffic lights to save energy.
- **FSM:** Increase number of lanes on roads to incorporate left and right turns, etc.
- **Visualization:** Add animations to ambulance during visualization on the screen as well as ambulance siren. Make it possible to playback accidents. That is, after two cars have collided, the system can give the user an option to playback the accident.
- **Sound:** The ambulance siren will go off when an ambulance appears on the screen. Add playback of screams for "help!" during collision.

2.2 Block Diagram



3. Implementation

For our setup, we will have a camera positioned at a bird eye view on the “street”. A huge 6ftx6ft piece of cardboard painted to simulate what happens on the road. We will then use remote-controlled cars to simulate cars following traffic rules and representing traffic densities.

3.1 Modules

We have three main modules in our system which can be further broken down into sub-modules. These modules are described in further detail in this section.

- 3.1.1 Image Processing
- 3.1.2 Traffic Control FSM
- 3.1.3 Graphics/Visualization

3.1.1 Image Processing

3.1.1.1 Object Detection in Image

Object detection will process a single pixel at a time and detect objects based on color. The input pixel values are in HSV space and have been eroded and dilated, so when multiple 1's are detected in a row, it is inferred as presence of a car.

We are using predetermined sizes of cars so we only need to keep track of the 'start' pixel of a car and add the size of the car to that 'start' pixel to find the 'stop' pixel. A collision is detected when the gap between cars equals zero, where gap = stop pixel of car 2 - start pixel of car 1 (car 2 is following car 1).

If a car is swerving or driving recklessly, this can be detected because the car pixels will not remain in the same row over a short period of time. Our system will have two remote-controlled cars and several 'cardboard painted' cars that will be placed on a wooden rod to simulate the presence of more cars.

3.1.1.2 Motion Detection

The main things we are going to identify are collisions, the direction of movement of the cars, overspeeding and running of red lights. Thus, the motion detection module is subdivided into 3 modules:

1. The position module - As we detect the x and y positions of cars in each frame, we store the information in a register. We don't need to account for speed if we are updating the x and y coordinates of the car in real time as they move. This information is then sent to the visualization FSM for plotting on the screen.

2. The illegal moves module - We can determine if a car ran a red light by looking at the traffic light; if our FSM says the traffic light is red and a car is in a certain range of the intersection, then we know the car has ran the light and we will send the hcount and vcount coordinates of the car as well as a signal to sound a siren to the visualization module.

3. The collision detection module - Given the locations of each car and the predetermined size of each car(in terms of pixels), if we detect that two different cars have overlapping pixels or are fairly close(pixels have touching edges) to one another, we will send a signal of a car crash to the visualization module.

3.1.2 Traffic Control FSM

Given inputs of the number of cars and their directions, the traffic control fsm determines what the output colors on the traffic lights should be and the duration for each color. It is subdivided into 2 modules:

1. The traffic control module - Given the count of cars on two different roads, it synthesizes the information to determine the optimal traffic control signals for the traffic lights.

2. The LED controller module - It takes in the color and direction of a traffic light and outputs individual signals for each component. For example, if input is “main traffic light red”, the output will be main traffic light red on, main traffic light yellow off and main traffic light green off.

3.1.3 Graphics/Visualization

All the image processing and object detection is done in previous modules. We will use an image ROM to store the images of the cars (so that they look real). We will then display moving cars, traffic lights, pedestrians, ambulances and police cars (when necessary) on the monitor.

Our stretch goals include:

1. Adding animations to display police cars or ambulances moving towards accident scenes.
2. Adding animations to show a pedestrian walking or a car collision in action.
3. If a collision occurs, we can play the visualization back upon request; this will help to detect the cause of the accident.

3.2 Module Testing & Design Verification

3.1.1 Image Processing

3.1.1.1 Object Detection in Image

We first need to test the interaction between the camera and the FPGA by checking that the output of the camera data is what we expect. We can display the images on the monitor to help identify locations of various objects.

Then, we need to make sure objects are detected correctly based on their color. We would feed in a single image at a time, and then check that all objects have been correctly detected and counted. If this is successful, we will test with multiple images.

3.1.1.2 Motion Detection

This module will require test benches for testing. We can simulate different car densities, traffic light states, speeding cars, and cars running red lights. Once the tests pass, we will test the interaction between this module and the real data. We can do this by taking pictures of our toy cars and simulating each of the different movements.

3.1.2 Traffic Control FSM

In order to test the timer and determine that the traffic light moves to the correct state with variable car densities, we will display the state and timer on the labkit hex display. We will also assign different LEDs to the different traffic signals (red, yellow, green) and test if they turn on or off at the correct time and for the correct duration.

3.1.3 Graphics/Visualization

We will use the monitor to test visualization- it's easiest to debug with the screen. Once this module can draw arbitrary cars and turn on the traffic lights using input signals, we can use actual data from the camera and observe real time that the visualization is accurate.

4. Responsibilities

Premila Rowles: Object Detection, Motion Detection
Jessica Quaye: Traffic Control FSM, Visualization

5. Timeline

Week	Premila Rowles	Jessica Quaye
Oct 29th	❑ Convert NTSC camera output from black and white to color (edit existing code)	➢ Complete baseline implementation of traffic control FSM
Nov 6th	❑ Convert YCrCb to RGB to HSV	➢ Incorporate two modes & pedestrian signal
Nov 13th	❑ Debug erosion and dilation ❑ Detect cars in the image	➢ Debug & add finishing touches ➢ Show visualization mirroring street on the screen
Nov 20th	❑ Motion detection algorithm: direction module, illegal moves module, collision detection module ❑ Integration	➢ Work on Visualization + Sound ➢ Integration
Nov 27th	❑ Reckless driving ❑ Other reach goals ❑ Debugging	➢ Debugging/other reach goals
Dec 4th	❑ Debugging ❑ Reach goals	➢ Debugging/other reach goals
Dec 11th	❑ Project Completed	➢ Project Completed

6. Supplies Needed

Need to be purchased

- 2 remote-controlled toy cars (~$30 total) : The two remote controlled cars will be used for simulating collisions and reckless driving.
- Traffic light LED and alternative

Need to be acquired

- Tension rod/rope: used to hold up traffic lights
- Wooden rod/pvc pipe to attach wood/cardboard cars
- Paint for cars/lights