

Meeting Minutes: Sept. 26th, 2013

We have already been doing client interviews....

Talked about using background subtraction for the majority of the image processing and neural networks for blobbing

Talked about MIT paper using background subtraction → used gaussians to model every single pixel...got way less results from way more advanced hardware...basic image processing from paper but not a lot else

Talk to john lam and peter oman about image processing? (CS guys)

We need to do everything real time...harder than what other people are doing.

Run program for our background subtraction and algorithm on raspberry pie and demonstrate at expo in December.

Are we familiar with raspberry pi graphics processing stuff? We are not. They have a lot of stuff dedicated to image processing.

Our algorithm – talked about low pass filter to slowly change pixel for background. Dynamically change the pixel thresholds for how to change background. We can only store four frames in memory (128k).

Use a single bit to denote whole frame average.

Use time stamp as input for neural network so we can detect headlights etc.

Use filter for headlight/sun removal so they don't wash out the entire frame.

Communications format for traffic control. Dr. Ay has a folder for what protocols are available in traffic control. There is a standard for traffic control. We will need to look at it to decide what we want our system to output.

Right now, don't focus on the outside world interface. Just get the position, speed, acceleration down and then rely on controller technology to catch up with us.

Ismail's Idea: Could we just subtract two frames from each other and only detect the difference in pixels? That way we would only see the moving objects.

High power operation. Definitely worth a try. Maybe need two FPGA's and twin channels, etc. Will cars be moving fast enough for 100 frames per second? Prolly not. At 25 mph they will only move 1cm.

So for right now, only try to capture one car moving. Don't worry about multiple cars and stuff for now. We don't have money or time to do those complicated now.

We should think about zoning. Split the field of view into ten zones or whatever. Maximum number of objects entering zone will be the number of lanes on the road. Extract and send information out for each

zone. Maybe speed and position for each zone over time? Maybe we should just try to detect the lane into ten zones and try to catch a car passing through that zone. Use checkpoints and only read those lines of the pixel array.

See attached picture of Dr. Ay's Drawing...Assumes that the controller will calculate speed and position and everything. We will just send it the information necessary to calculate these things and then have a lookup table or something available for it to make the calculations quickly.

We can have two modes. One is the video mode where we spit out ten frames per second, and then another mode that only read out the ten pixel rows (zones)

Distance would be 500 meters for the camera to see out in front of the intersection

Scale the pixels so that the foreground and background pixels have the same number of square foot of road assigned to them. Pixels are square and thus when we blow them up and represent them on the monitor, it will look like the car never changes size. It will look like we are looking straight down on the car the whole time.

Action items:

Fix how we are going to try to detect and calculate how many frames we would need and find how many zones. How many bits we need. How many frames we need to have a good resolution. Assume the

distance on the road is 500 meter. Get how many bits we need to send.

