**LPR Algorithm Design**

Plate Finding

Part one is estimating a possible plate location (center point). Part two is refining the plate location estimate by doing the best job possible of locating the plate’s four edges.

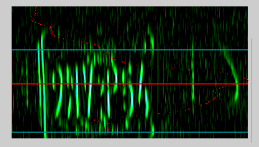
Part one is taking the input image (640x480 for example) and sub-scaling by a factor of 5 on each axis (resulting in a 128 x 96 image in this example). The 2-nd order edge contour is created. The term 2-nd order is used to indicate that neighborhood edge energy is accumulated for each pixel. That is, if in the edge contour map a pixel is hot (over a threshold), its neighbors hotness is added to it. This causes plates (lots of edges in an area) to get really bright compared to other things (generally speaking). This work is done in the function: findBrightestSpot().

Once the second order edge contour map is created on the sub-scaled image, finding candidate plates is a matter of finding the hottest spots in regions (or in super sized plate windows). Looping on hotspots, starting with the brightest, each bright spot center point is passed to the findExactEdges function (its name is a misnomer because its edge finding isn’t very exact). Then the loop comes back to the second hottest spot, as a second plate candidate, and so on until MAX\_CANDIDATE\_PLATES is reached.

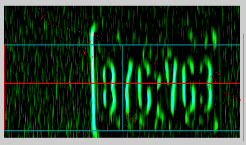
Plate Edge Finding

Once bright spots (candidate plate locations) are determined, the second step, to isolate the plate edges, is to create a 1st order edge contour map from the full resolution image, in a region around the center point (brightest spot).

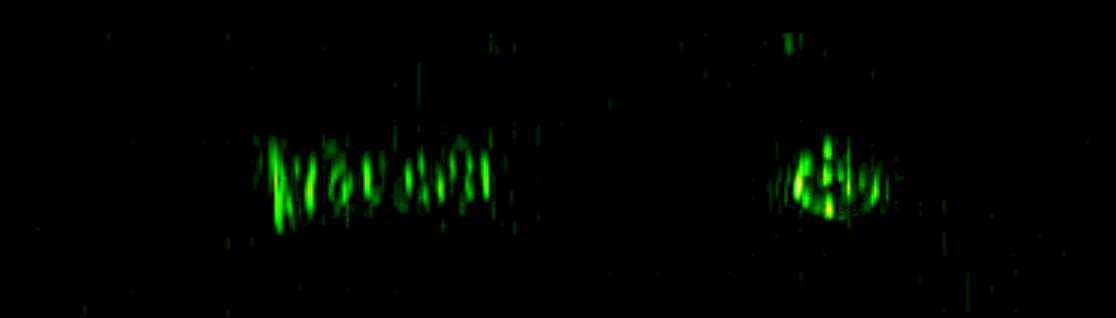
Below are Edge Contour Maps of MAX WIDTH x MAX HEIGHT plate windows. Need to find exact (or closer) edges of the plates. The maps are created by taking the plate center point estimates from the findBrightestSpot function and drawing a box around these center points. The size of the box is MAX PLATE WIDTH x MAX PLATE HEIGHT pixels.



Case 1. Plate on the left, part of a bumper to its right. Need to separate the right edge of the plate. The right edge of the plate itself does not appear in the image.



Case 2. Very large (200 pixels wide x 90% of MAX HEIGHT) plate on the right. No other objects which can be confused as the plate. However, lots of edge noise around the plate.



Case 3. Very small plate on the left and right tail light on the right. In this case, the plate/car is far away and relatively small such that half of the rear end of the car fits in a window of MAX PLATE WIDTH x MAX PLATE HEIGHT. Need to separate the plate from the tail light. Challenge: do not know if this is a large plate with a whitespace in the center (e.g: “123 FHG”) or a small plate and some other object.

**Solution Design**

Generally speaking, the findBrightestSpot () method goes a good job getting the vertical center right; that is, if we draw a horizontal line through the center point (y value) provided by findBrightestSpot function, then generally that line cuts through the letters on the plate.

But the algorithm often provides a really bad estimate of the horizontal center point (x value). Objects to the left or right of the plate (bumper artifacts, tail lights, etc) often are included in the plate are energy accumulation and skew the center point to one side or the other. This is very difficult to control as the relative size of the car/plate can vary greatly.

So the real challenge is finding the left and right edges of the plate.

**Top/Bottom finding**

Using the high resolution edge contour map, sum the edge energy on horizontal scan lines. There will be MAX PLATE HEIGHT scan lines. Find the peak sum of all the lines. Start in the center of the plate and move up, looking at each line. When the line sum value drops below a threshold (like 30%) of the peak value, call that the top. Repeat to find the plate bottom.

**Left/Right finding**

Using the newly determined topEdge and bottomEdge as the bounds for the range in the following calculations.

1. Generate the Image Integral over the region window.
2. Find the center of mass by running a test box over the image integral. The test box will be MIN PLATE WIDTH pixels wide by MIN PLATE HEIGHT pixels tall. This represents a middle ground size: the smallest allowed plate will take up this entire box if properly aligned. Larger plates will contain two to 2.5 of these test boxes. This ensures that a smaller plate will be located while with larger plates we will identify a point within that plate, but not necessarily the center.
3. Use the center of mass location to find left and right edges using the same method used for finding top and bottom edges.