

EECS297A LAB5 OUTLINE. VISUAL AID FOR THE BLIND

by Nasser Abbasi

Nov 18, 2004

The problem: Per the lab assignment handout, the system is supposed to deliver the following verbal output (both spoken and text) upon recognizing the respective objects: "*You are wearing a horizontally striped shirt today*" or "*You're wearing a vertically striped shirt today*" or "*I see something red*" or "*This is blue like the sky*".

In this outline, we will concentrate only on how to accomplish the last 2 goals shown above.

We need to be able to recognize objects which we consider to have a predominant specific color. Here we are interested in red or blue (but the color itself is not important, the method will work for any color). So, this is a problem of image segmentation based on color content.

I see 2 main problems to solve. The first is to decide if a particular pixel belongs to the class of pixels that are considered to have the color of interest, call it color X for now. This problem can be solved as follows.

Convert the RGB channels to HSL to make it easier to classify colors. (Color classification can be done directly with RGB, eliminating the need to do the conversion, but I think using HSL for this is more natural). For an example to classify a pixel as of being of color X , do the following:

```
IF pixel hue value is within a certain degree range (depends on the color, for red, use something like  $0^0 \dots 10^0$ )
  AND the pixel Lightness value in 40%  $\dots$  60% AND pixel saturation  $> 70\%$  THEN
    tag pixel to class  $X$ 
END IF
```

The further away the Hue angle is from its true angle (example, 0^0 for red), the higher the saturation range needs to be and the lower the lightness range needs to be for the color to be still be classified as X . This will complicate the algorithm when the Hue angle range becomes larger. So using a smaller range for Hue angle will make the algorithm simpler, but might miss shades of color to the color we are trying to detect.

Once color classification is completed, we obtain an image where each pixel is tagged as either BLUE or RED or neither.

The second problem to solve, is that we need to decide which of the pixels of color X constitute an object. The reason is that we need to differentiate between random isolated and small clusters of pixels of color X (could be noise) as compared to large connected sets of pixels which would indicate main objects in the image which is of color X .

The problem specification did not indicate what the requirements for this are. A simple solution to this is to check that we obtained at least a certain percentage of pixels of color X in the image, we would then claim that we saw this color in the image. A better solution would be as follows:

For each pixel (x,y) in the image which belongs to the color X

Find the set of pixels which are connected to it using 8-adjacency neighbors. This set is called the connected component of pixel (x,y)

We now have a set of sets of connected components. Sort the set based on the size of the connected components. Now we have an ordered sequence of sets. Use a threshold cutoff below which we will discard those sets in the sequence. What is left are sets which represents objects of significant size in the image which are all separate, and all are of color X . If no sets are left, then we claim we did not detect color X .

In the above, we did not differentiate between the goal of producing the output "*I see something red*" and the goal of producing "*This is blue like the sky*". Other than the obvious difference of the color, the main difference between these is the purity of color. For the "blue" output, the statement seems to indicate that we need to check for a very pure blue color, hence we need to classify the pixel to have a higher saturation value. So for blue, I would make the check for pixel saturation $> 90\%$ for an example, instead of using a lower value for checking for RED.

Summary of the algorithm

1. Read the image file. (Support for reading GIF or JPEG exist in Mathematica, as well as for other image files.).
2. Set values for parameters: Threshold cutoff for size of connected component set, ranges for color Hue values, Ranges for Saturation and ranges for Lightness. These parameters could be hardcoded in the program initially or read as input from user for flexibility if needed.
3. Obtain the RGB color channels from the data read and convert to HSL (algorithm exist).
4. For each pixel, classify if RED or BLUE or neither. Use HSL model for this classification as shown above. For BLUE use larger saturation value that we used for RED.
5. For each pixel of class RED find its connected component set. Sort the sets based on size, apply size threshold. IF what is left is non-empty, then output "*I see something red*"
6. Repeat steps 5 but apply to pixels of color BLUE.
7. IF steps 5 and 6 did not product any output, then output "Unable to recognize a color of interest".