Andrew Kozempel

Professor Avanzato

CMPSC 497

5 October 2023

**Lab 3: Color Tracking with Images**

**Objective:** Design and test a MATLAB algorithm and code to identify a red object in an image and display the centroid (location) and bounding box.

**Materials:**

* Red Glasses Wipes
* Red Ghost Code
* Blue Ghost Code
* Green Gum Pack (with red spot)
* Purple Contact Lens Case
* Blue Chapstick
* Blue Philips Zoom Packaging

**MATLAB Script:**

| **%{ Andrew Kozempel CMPSC 497 Fall 2023 LAB #3: Color Tracking with Images %}  % load images images = {'img1.jpg', 'img2.jpg', 'img3.jpg'};  % loop through images for i = 1:length(images)   % read image  RGB = imread(images{i});  fprintf('\nImage %d: ', i);  fprintf('\nSize: %d x %d', size(RGB, 1), size(RGB, 2));   % split into R, G, B planes  r = RGB(:, :, 1);  g = RGB(:, :, 2);  b = RGB(:, :, 3);    % create a binary mask for red objects  redMask = (r > 2\*g) & (r > 2\*b) & (r > 30);    % apply closing  se = strel('disk', 35);  redClosed = imclose(redMask, se);    % remove small objects  redObjects = bwareaopen(redClosed, 35);   % display binary image  figure;  imshow(redObjects);  title(sprintf('Binary Image %d', i));    % properties of labeled regions  properties = regionprops(bwlabel(redObjects), 'Centroid', 'BoundingBox');  numRedObjects = numel(properties);    % display original image  figure;  imshow(RGB);  title(sprintf('Image %d', i));  hold on;    % loop through red objects to display centroids and bounding boxes  for k = 1:numRedObjects  centroid = properties(k).Centroid;  boundingBox = properties(k).BoundingBox;    % mark centroid (x,y,+)  plot(centroid(1), centroid(2), 'b+');    % draw bounding box  rectangle('Position', boundingBox, 'EdgeColor', 'b', 'LineWidth', 2);  end    zoom on;   % print the number of red objects  fprintf('\nDetected %d red objects.\n', numRedObjects);   end** |
| --- |

**Results:**

| **Image #** | **Binary Image** | **Original Image with Overlay** |
| --- | --- | --- |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |

| **Image #** | **Output** |
| --- | --- |
| 1 | Image 1:  Size: 3024 x 4032  Detected 2 red objects. |
| 2 | Image 2:  Size: 3024 x 4032  Detected 1 red objects. |
| 3 | Image 3:  Size: 3024 x 4032  Detected 2 red objects. |

**Conclusion:**

This was a pretty straightforward lab. I was able to achieve the objective of identifying the red objects within an image.

I had to mess with the size of the structural element, but everything ended up working with a value of 35, which I thought was pretty high. It would detect multiple objects within one object, which was probably because of the lighting and reflections. I did not have to increase the value too high to fix that. I think the value was about 10 or 15 for that issue. However, the instructions said to completely close the red object, so the glasses wipes package is the reason for the value of 35. It was to completely solidify the object without the white strip in the middle. I am not entirely sure if that is what you were asking in the instructions, but the results turned out just fine either way.

Another interesting result was the little red piece on the green gum pack in image 3. I was not even thinking about that as a red object, not because it is relatively small, but because I did not even notice it when I chose that object for the image. So, while I was thinking that image only had one red object, it turned out to have 2.