Objective

The objective of this lab was to use the live feed from a webcam to perform analysis on an image and draw a bounding box over the largest red object.

Code

below is a modified version of my lab 3 code. Specifically, I created an infinite loop to capture frames from the webcam. Within this while loop, i have loop that is used to iterate over all detected red objects in a given frame and select the largest one. This is done by finding the area the bounding box the program would put around each object and then selecting the largest one.

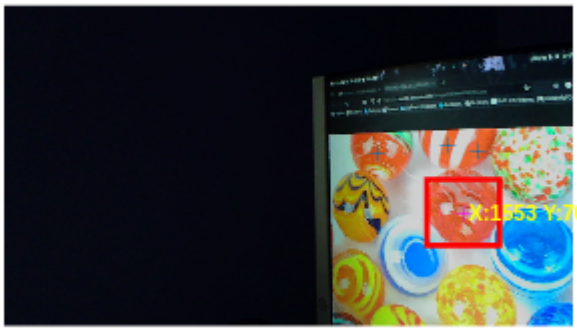
```
cam = webcam

% preview(cam) - launches camera screen
% snapshot(cam) - takes a pic

% processing images through a loop
for i=1:5
    % Acquire a single image from video.
    im = snapshot(cam);

    r=im(:,:,1);
    g=im(:,:,2);
    b=im(:,:,3);

    % hueristic function for determining red only
    red = (r>2*g) & (r>2*b) & (r>40);

    % group all red objects within 5 pixels of each other as one
object (this one should be tweaked)
    se = strel('disk', 9);
    red = imclose(red,se);

    % remove all objects smaller than 35 pixels in area (adjust this
```

```
threshold as needed)
    red = bwareaopen(red, 2000);

        % Centroid

    % separate red and NOT red
    s = regionprops(bwlabel(red), 'centroid');
    S = vertcat(s.Centroid);

    % plotting...
    figure
    imshow(im)

    hold on
    plot(S(:,1), S(:,2), '+') % identify red objects with +

    zoom on

        % Bounding Box
    % get all red objects (appear as white blobs from intensity
graph)
    stats = regionprops(red, 'BoundingBox', 'Centroid') % passing in
red intensity binarized image

    % finding largest element
    if numel(stats) > 1

        largest_bb = 0;
        largest_area = 0;
        largest_bc = 0;

        % This is a loop to bound the red objects in a rectangular
box.
        for object = 1:length(stats)
            bb = stats(object).BoundingBox;
            bc = stats(object).Centroid;

            bb_w = bb(3)
            bb_h = bb(4)

            bb_area = bb_w * bb_h;
```

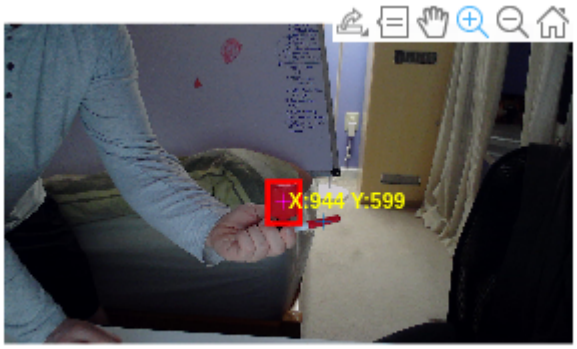
```
            if bb_area > largest_area
                largest_bb = bb
                largest_area = bb_area
                largest_bc = bc
            end
        end

            if ~isempty(largest_bb) && ~isempty(largest_bc)
                rectangle('Position', largest_bb, 'EdgeColor', 'r',
'LineWidth', 2); %  bounding box
                plot(largest_bc(1), largest_bc(2), '-m+'); % mark
the centroid with '+'
                a = text(largest_bc(1)+15, largest_bc(2), strcat('X:
', num2str(round(largest_bc(1))), ' Y: ',
num2str(round(largest_bc(2)))));
                set(a, 'FontName', 'Arial', 'FontWeight', 'bold',
'FontSize', 8, 'Color', 'yellow');
            end
    end

end
```

Images

Simple Detection



Successful Identification of largest red object



Unsuccessful Identification or False Positives

In this image, the the red markers are distinct but the algorithm treats them as a single entity

The rgb lights change fairly rapidly but it seems that the white mixed with the red confused it enough to select this portion

Video Code

In order to create a video, i needed to modify the code a little to include a `VideoWriter()`. This was done since the original program just treats the input as a

stream of images that are separate.

```
% setup video writer
video = VideoWriter('tracking_output.avi');
video.FrameRate = 10;
open(video);

% display figure for video
hFig = figure;
```