CSI 4133 – Lab 04

Colour based object detection

Contents

Introduction to

- Hue colour-correspondence experiment
- Colour-based object detection

Hue colour-correspodence experiment

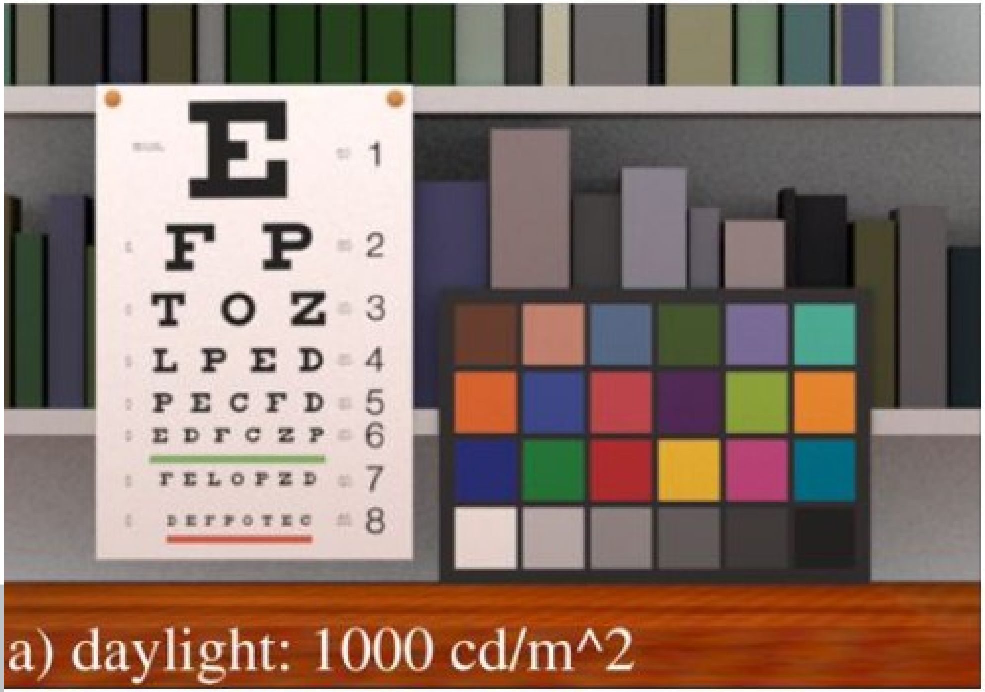
Procedure

- Load image (folder “images”)
- Convert image from RGB space into HSV space
- Isolate pixels with a specific hue value
  - Use a track bar to set the Hue value H_v
  - Use loops to get all the H,S,V values
    - If (H_current != H_v)
    - Then set H_current, S_current, V_current = 0
- Convert the image containing the isolated pixels from HSV space to RGB space
- Visualize the results

Hue colour-correspodence experiment

Analysis

- Change the Hue value using the track bar to find the min_H and max_H for the
  - Yellow-Green square (fifth column, second row)
  - Violet square (fourth column, second row)
  - Red square (third column, third row)

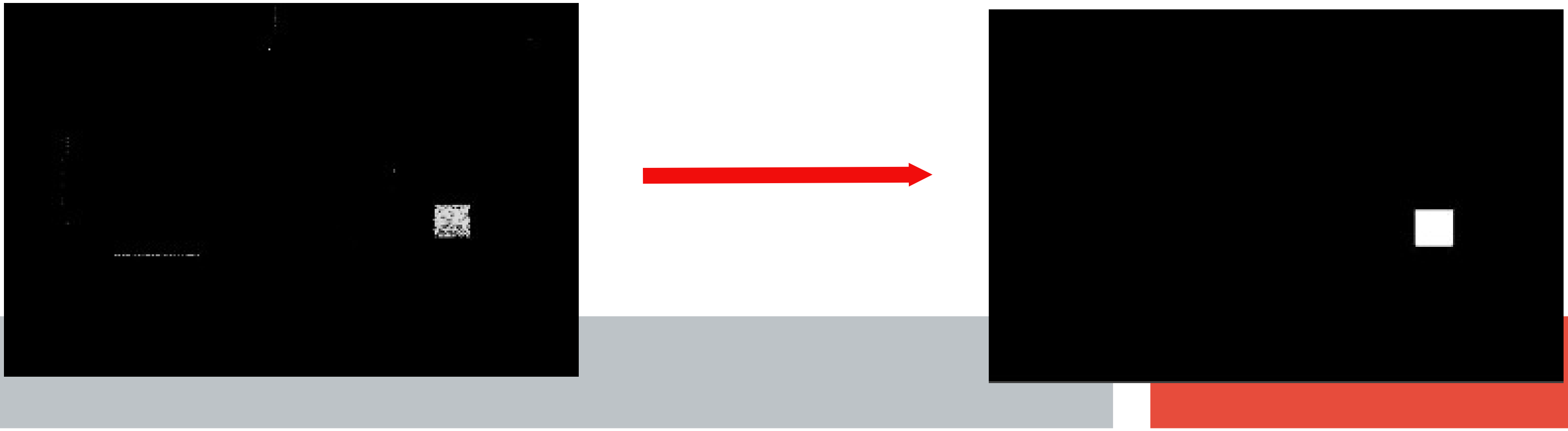


a) daylight: 1000 cd/m^2

Colour-based object detection

Procedure

- Get the appropriate Hue value/ranges for the yellow-green square/violet square/red square from Hue colour-correspondence experiment
- Generate the colour masks and refine the colour masks using
  - Erode function
  - Dilate function
  - Pay attention to the size of the kernel elements



Colour-based object detection

Procedure

- Isolate the yellow-green square, the violate square, and the red square in the grid (create a track bar to select among Yellow_Green(0), Violet(1), and Red(2))
- Show the isolated pixels (in their original colour RGB) in a window
- Show the isolated pixels (as a binary mask of all the detected pixels) in a window

Task

Goal: Experiment to see which hue-values correspond to which visible-spectrum colours in OpenCV.

Task (Part A)

Idea:

1. Load Image (folder “images”).
2. Convert Image from RGB space into HSV space.
3. Isolate pixels with a specific hue value.
4. Convert the image containing the isolated pixels from HSV space back into RGB space.
5. Visualize the result.

Hints:

- Your solution may require you to process each pixel individually.
- When displaying your results, use a window named "Processed Hue" (it is integrated with a track bar associated with the ‘hue’ variable)

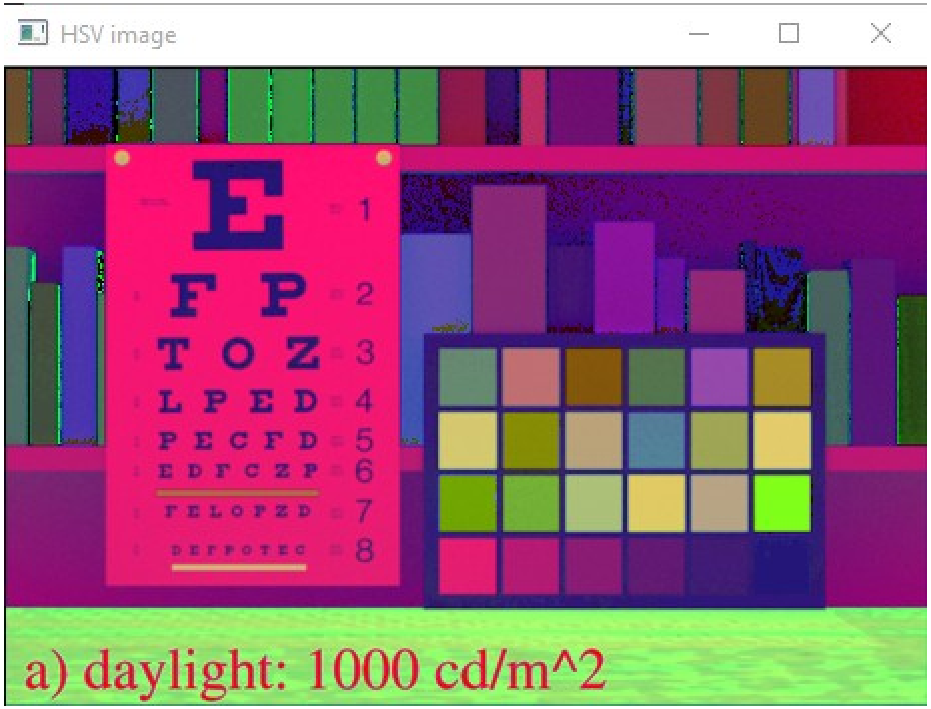
Task (Part A)

Hints:

- Your solution may require you to process each pixel individually.
- When displaying your results, use a window named "Processed Hue" (it is integrated with a track bar associated with the ‘hue’ variable)

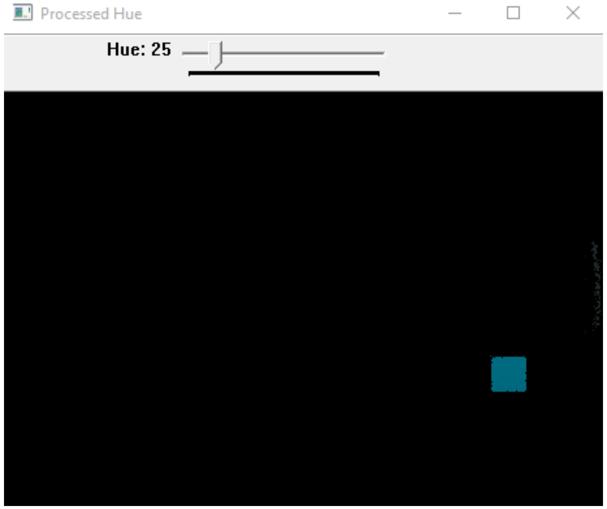
Task (Part A)

Examples: The HSV image (Picture3.png)

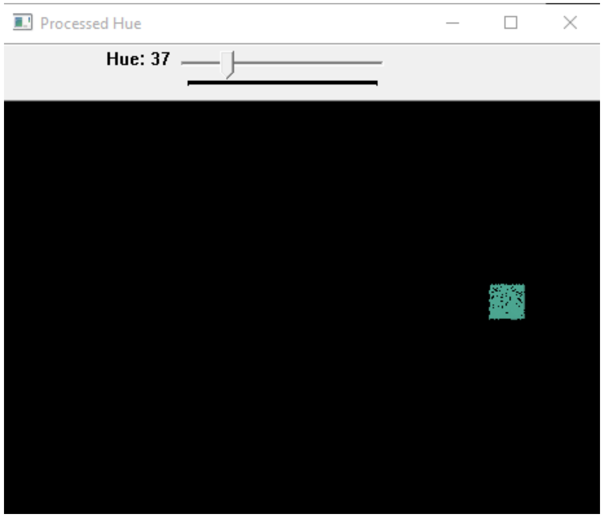


Task (Part A)

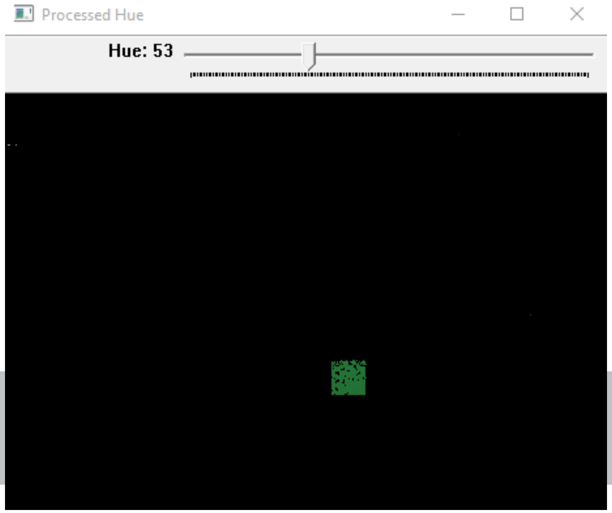
Examples: Different Hue Values


Processed Hue

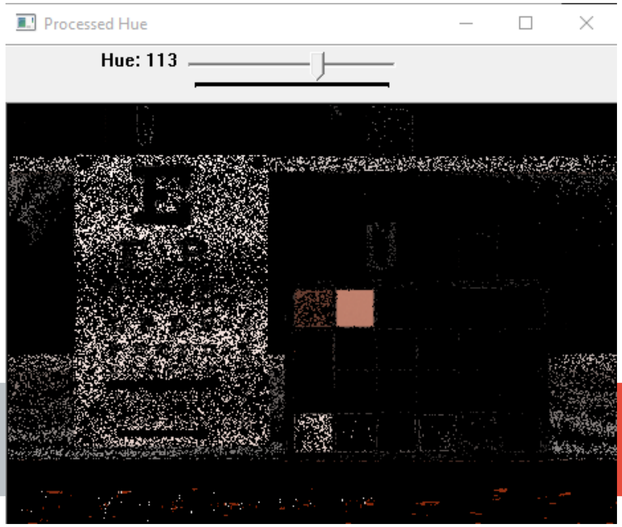
Hue: 25


Processed Hue

Hue: 37


Processed Hue

Hue: 53


Processed Hue

Hue: 113

Task (Part B)

Idea:

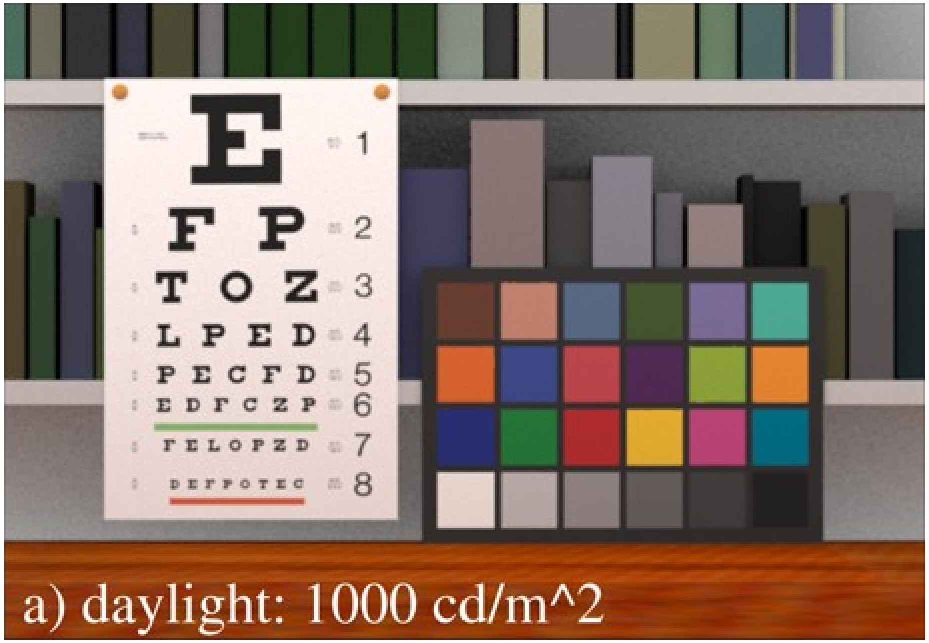
1. Yellow-Green object detection.
2. Violet object detection.
3. Red object detection.

Hints:

- Use Part A’s solution to help you pick appropriate hue values/ranges.
- Generate colour masks for different colours.
- Some creativity may be required in getting rid of the noise.

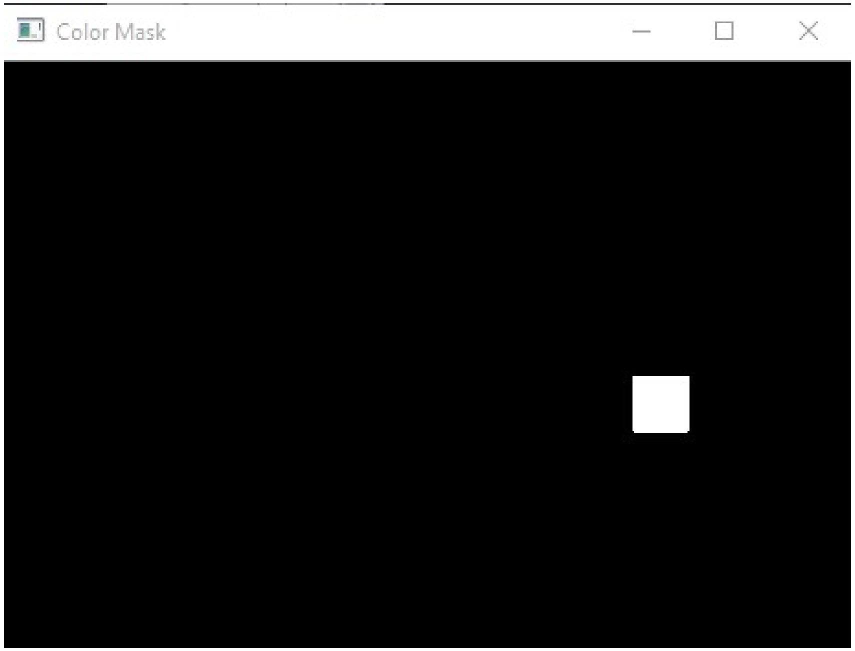
Task (Part B)

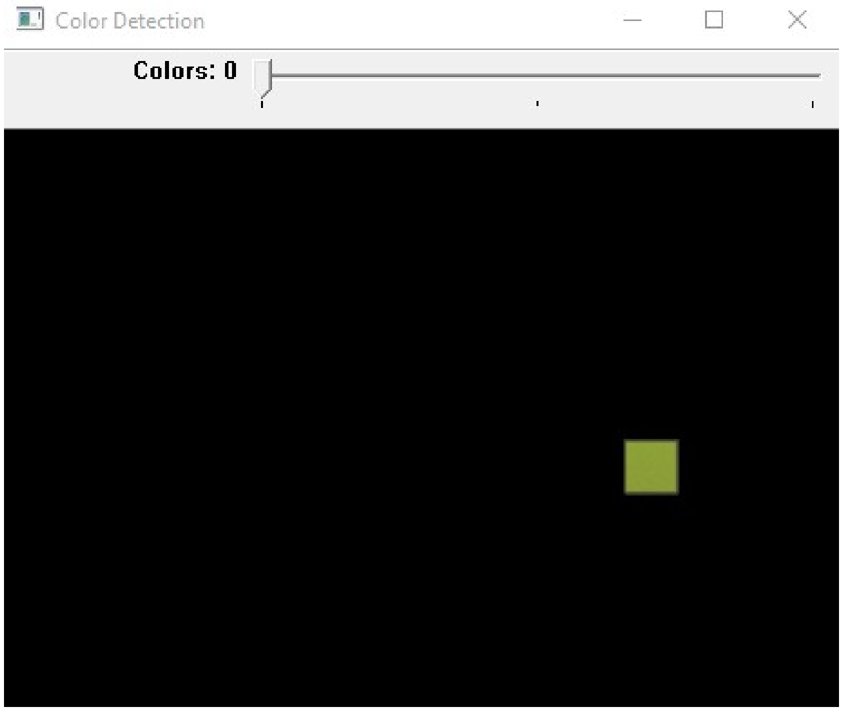
Examples: Original RGB Image



Task (Part B)

Examples: Yellow-Green and Violet object detection results (left: colour masks, right: colour detection results)

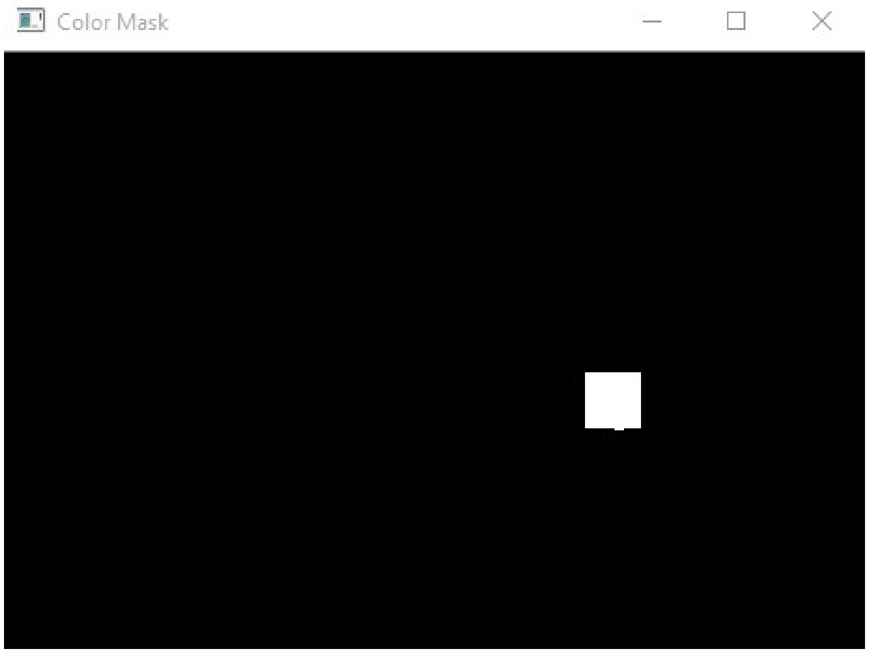


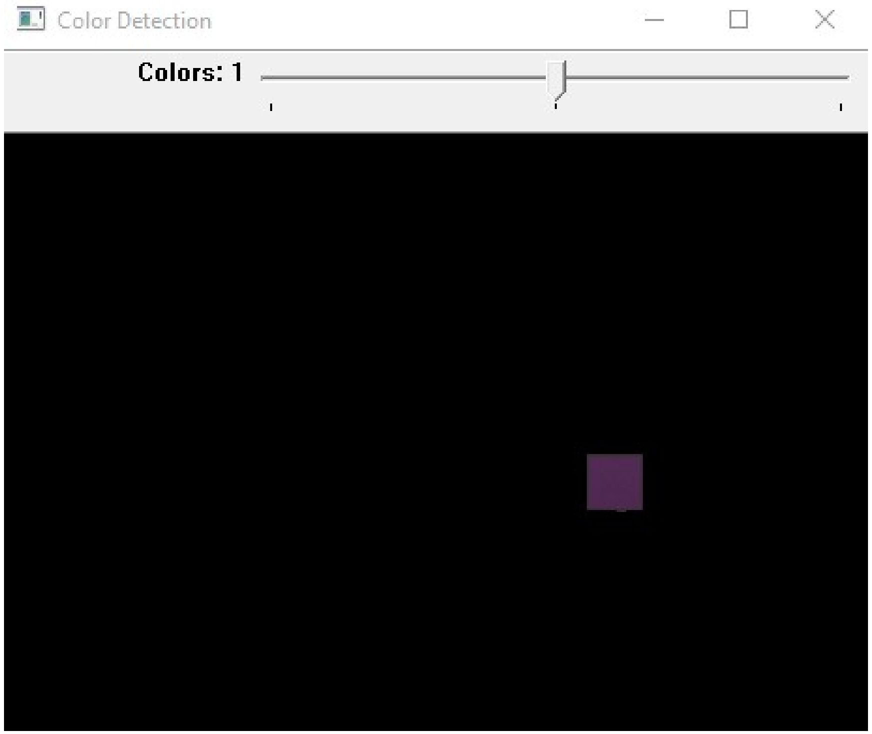


Colors: 0

Task (Part B)

Examples: Yellow-Green and Violet object detection results (left: colour masks, right: colour detection results)





Colors: 1

Task

Please submit a **lab report**, **source code**, and **screenshots** of your results.

END

THANK YOU