Mansib Ahmed

CMPSC 497

10/17/2023

Red & Blue Color Detection with OpenCV

Objective: To use Python’s OpenCV library to implement color detection algorithms for red and blue objects.

Materials:

1. Python OpenCV library
2. Jupyter Notebook
3. Red and blue objects

Successful cases of blue object detection:

A pair of scissors in a shape of a heart

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Finding largest blue object if there are multiple:

A close up of a computer screen

Description automatically generated

Failure case of blue object detection:

A screenshot of a computer

Description automatically generated

Code for blue object detection:

cap = cv.VideoCapture(0)

while(1):

*# Take each frame*

    \_, frame = cap.read()

*# Convert BGR to HSV*

    hsv = cv.cvtColor(frame, cv.COLOR\_BGR2HSV)

    lower\_blue = np.array([110, 50, 50])

    upper\_blue = np.array([130, 255, 255])

    mask = cv.inRange(hsv, lower\_blue, upper\_blue)

    largest\_mask = np.zeros\_like(mask)

*# find all the blue objects*

    contours, \_ = cv.findContours(mask, cv.RETR\_EXTERNAL, cv.CHAIN\_APPROX\_SIMPLE)

    if len(contours) > 0:

*# find and draw largest object*

        largest\_contour = max(contours, key=cv.contourArea)

        cv.drawContours(largest\_mask, [largest\_contour], -1, (255), thickness=cv.FILLED)

*# Bitwise-AND mask with largest object and original image*

    res = cv.bitwise\_and(frame, frame, mask=largest\_mask)

    cv.imshow('frame', frame)

    cv.imshow('mask', largest\_mask)

    cv.imshow('res', res)

    k = cv.waitKey(5) & 0xFF

    if k == 27:  *# escape key is 27; space is 32, etc*

        break

cv.destroyAllWindows()

Successful cases of red object detection:

A screenshot of a computer

Description automatically generated

A screen shot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

Finding the largest red object if there are multiple:

A screenshot of a computer

Description automatically generated

Failure case:

A screen shot of a computer

Description automatically generated

Code for red object detection:

cap = cv.VideoCapture(0)

while(1):

*# Take each frame*

    \_, frame = cap.read()

*# Convert BGR to HSV*

    hsv = cv.cvtColor(frame, cv.COLOR\_BGR2HSV)

    lower\_red = np.array([0, 100, 100])

    upper\_red = np.array([10, 255, 255])

    mask = cv.inRange(hsv, lower\_red, upper\_red)

    largest\_mask = np.zeros\_like(mask)

    contours, \_ = cv.findContours(mask, cv.RETR\_EXTERNAL, cv.CHAIN\_APPROX\_SIMPLE)

    if len(contours) > 0:

*# find and draw largest object*

        largest\_contour = max(contours, key=cv.contourArea)

        cv.drawContours(largest\_mask, [largest\_contour], -1, (255), thickness=cv.FILLED)

*# Bitwise-AND mask with largest object and original image*

    res = cv.bitwise\_and(frame, frame, mask=largest\_mask)

    cv.imshow('frame', frame)

    cv.imshow('mask', largest\_mask)

    cv.imshow('res', res)

    k = cv.waitKey(5) & 0xFF

    if k == 27:  *# escape key is 27; space is 32, etc*

        break

cv.destroyAllWindows()

Conclusion:

Overall, this program was able to successfully detect red and blue objects and display the largest one. Certain test cases with objects that had a navy-blue color were not detected as well as objects with regular blue color. Moreover, objects that had an unusual shape or configuration were also not detected as well.