

Curs 10 - camera web

December 14, 2020

Catalin Stoean

catalin.stoean@inf.ucv.ro

<http://inf.ucv.ro/~cstoean>

1 Transformata Hough pentru detectie de cercuri

```
[8]: import cv2
import numpy as np
from matplotlib import pyplot as plt

imgColor = cv2.imread('D:\\ban.jpg')
imgColor = cv2.cvtColor(imgColor, cv2.COLOR_BGR2RGB)
img = cv2.cvtColor(imgColor, cv2.COLOR_BGR2GRAY)
img = cv2.medianBlur(img, 5)

cercuri = cv2.HoughCircles(img, cv2.HOUGH_GRADIENT, 1, 20, param1=200,
↳param2=130, minRadius=0, maxRadius=0)

#pentru a le desena, valorile cercurilor trebuie sa fie intregi
cercuri = np.uint16(np.around(cercuri))

print('Numarul de cercuri detectate: ', len(cercuri))

for c in cercuri[0,:]:
    # cercul detectat
    cv2.circle(imgColor, (c[0], c[1]), c[2], (0, 255, 0), 15)
    # centrul cercului detectat
    cv2.circle(imgColor, (c[0], c[1]), 10, (255, 0, 0), -1)

fig = plt.figure(figsize=(6, 5))
plt.imshow(imgColor)
plt.title('Detectare de cercuri')
plt.axis('off')

plt.show()
```

Numarul de cercuri detectate: 1

Detectare de cercuri



2 Detectare obiect de o anumita culoare intr-o poza

Valorile aproximative din HUE Portocaliu 0-22 Galben 22-38 Verde 38-75 Albastru 75-130 Violet 130-160 Rosu 160-179

```
[12]: import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread("D:\\poza.jpg")
numeFereastră = 'Initiala'
imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
formatHSV = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)

#cautam verdele in poza
lowH = 38
highH = 75

lowS = 0
highS = 255
```

```

lowV = 0
highV = 210

masca = cv2.inRange(formatHSV, np.array([lowH, lowS, lowV]), np.array([highH,
↪highS, highV]))

kernel = np.ones((5,5),np.uint8)
deschisa = cv2.morphologyEx(masca, cv2.MORPH_OPEN, kernel)
inchisa = cv2.morphologyEx(deschisa, cv2.MORPH_CLOSE, kernel)

fig = plt.figure(figsize=(14, 5))
plt.subplot(131)
plt.imshow(imgRGB)
plt.title('Initiala')
plt.axis('off')
plt.subplot(132)
plt.imshow(masca, cmap = 'gray')
plt.title('Masca pentru verde')
plt.axis('off')
plt.subplot(133)
plt.imshow(inchisa, cmap = 'gray')
plt.title('Masca deschidere-inchidere pentru verde')
plt.axis('off')

plt.show()

```



3 Detectare obiect de o anumita culoare intr-o poza cu trackbar

```

[1]: import cv2
import numpy as np

def detecteaza(_x):
    lowH = cv2.getTrackbarPos("lowH", numeFereastră)
    highH = cv2.getTrackbarPos("highH", numeFereastră)

```

```

lowS = cv2.getTrackbarPos("lowS", numeFereastră)
highS = cv2.getTrackbarPos("highS", numeFereastră)
lowV = cv2.getTrackbarPos("lowV", numeFereastră)
highV = cv2.getTrackbarPos("highV", numeFereastră)

formatHSV = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)

masca = cv2.inRange(formatHSV, np.array([lowH, lowS, lowV]), np.
↪array([highH, highS, highV]))

kernel = np.ones((5,5),np.uint8)
deschisa = cv2.morphologyEx(masca, cv2.MORPH_OPEN, kernel)
inchisa = cv2.morphologyEx(deschisa, cv2.MORPH_CLOSE, kernel)

cv2.imshow('Masca', inchisa)

img = cv2.imread("D:\\poza.jpg")
numeFereastră = 'Initiala'
cv2.namedWindow(numeFereastră, cv2.WINDOW_AUTOSIZE)
cv2.imshow(numeFereastră, img)

cv2.createTrackbar("lowH", numeFereastră, 0, 179, detecteaza)
cv2.createTrackbar("highH", numeFereastră, 0, 179, detecteaza)

cv2.createTrackbar("lowS", numeFereastră, 0, 255, detecteaza)
cv2.createTrackbar("highS", numeFereastră, 0, 255, detecteaza)

cv2.createTrackbar("lowV", numeFereastră, 0, 255, detecteaza)
cv2.createTrackbar("highV", numeFereastră, 0, 255, detecteaza)

cv2.waitKey(0)
cv2.destroyAllWindows()

```

4 Citire frame-uri de la camera

```

[3]: import numpy as np
import cv2

cap = cv2.VideoCapture(0)

while(True):
    #citim frame
    _ret, frame = cap.read()

    #Aici putem procesa frame-ul curent

```

```

# Afisam frame
cv2.imshow('frame', frame)

#apasam q pentru a incheia citirea
if cv2.waitKey(1) == 27:
    break

cap.release()
cv2.destroyAllWindows()

```

5 Detectare obiect de o anumita culoare folosind intrarea de la camera

```

[4]: import numpy as np
import cv2

def nimic(_x):
    pass

numeFereastră = 'Masca'

cv2.namedWindow(numeFereastră, cv2.WINDOW_AUTOSIZE)

cv2.createTrackbar("lowH", numeFereastră, 0, 179, nimic)
cv2.createTrackbar("highH", numeFereastră, 0, 179, nimic)

cv2.createTrackbar("lowS", numeFereastră, 0, 255, nimic)
cv2.createTrackbar("highS", numeFereastră, 0, 255, nimic)

cv2.createTrackbar("lowV", numeFereastră, 0, 255, nimic)
cv2.createTrackbar("highV", numeFereastră, 0, 255, nimic)

cap = cv2.VideoCapture(0)

while(True):
    #citim frame
    _ret, frame = cap.read()

    # Afisam frame
    cv2.imshow('frame', frame)

```

```

lowH = cv2.getTrackbarPos("lowH", numeFereastră)
highH = cv2.getTrackbarPos("highH", numeFereastră)
lowS = cv2.getTrackbarPos("lowS", numeFereastră)
highS = cv2.getTrackbarPos("highS", numeFereastră)
lowV = cv2.getTrackbarPos("lowV", numeFereastră)
highV = cv2.getTrackbarPos("highV", numeFereastră)

formatHSV = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

masca = cv2.inRange(formatHSV, np.array([lowH, lowS, lowV]), np.
↪array([highH, highS, highV]))

kernel = np.ones((3,3), np.uint8)
deschisa = cv2.morphologyEx(masca, cv2.MORPH_OPEN, kernel)
inchisa = cv2.morphologyEx(deschisa, cv2.MORPH_CLOSE, kernel)

cv2.imshow(numeFereastră, inchisa)

#apasam ESC pentru a incheia citirea
if cv2.waitKey(1) == 27:
    break

cap.release()
cv2.destroyAllWindows()

```

6 Gasire cel mai mare contur dintr-o imagine binara

```

[5]: import numpy as np
import cv2
from matplotlib import pyplot as plt

img = cv2.imread("D:\\poza.jpg")
numeFereastră = 'Initiala'
imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
formatHSV = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)

#cautam verdele in poza
lowH = 38
highH = 75

lowS = 0
highS = 255

lowV = 0
highV = 210

```

```

masca = cv2.inRange(formatHSV, np.array([lowH, lowS, lowV]), np.array([highH,
↳highS, highV]))

kernel = np.ones((5,5),np.uint8)
deschisa = cv2.morphologyEx(masca, cv2.MORPH_OPEN, kernel)
inchisa = cv2.morphologyEx(deschisa, cv2.MORPH_CLOSE, kernel)

#Gasire contururi din "inchisa"

contours, hierarchy = cv2.findContours(inchisa, cv2.RETR_TREE, cv2.
↳CHAIN_APPROX_SIMPLE)

ariile = []
for c in contours:
    ariile.append(cv2.contourArea(c))
maxLoc = ariile.index(max(ariile))

x, y, w, h = cv2.boundingRect(contours[maxLoc])
cv2.rectangle(imgRGB, (x, y), (x + w, y + h), (255, 0, 0), 3)

fig = plt.figure(figsize=(12, 5))
plt.subplot(121)
plt.imshow(inchisa, cmap = 'gray')
plt.title('Imaginea binara')
plt.axis('off')
plt.subplot(122)
plt.imshow(imgRGB, cmap = 'gray')
plt.title('Dreptunghiul care include conturul')
plt.axis('off')

plt.show()

```

Imaginea binara



Dreptunghiul care include conturul



7 Gasire cel mai mare contur de la camera

7.1 Pornim de la codul cu trackbar si camera de mai sus si punem dreptunghi pe cel mai mare contur gasit

```
[5]: import numpy as np
import cv2

def nimic(_x):
    pass

def gasesteDreptunghi(inchisa):
    contours, hierarchy = cv2.findContours(inchisa, cv2.RETR_TREE, cv2.
↳CHAIN_APPROX_SIMPLE)
    x = 0
    y = 0
    w = 0
    h = 0
    if len(contours) > 0:
        ariile = []
        for c in contours:
            ariile.append(cv2.contourArea(c))
        maxLoc = ariile.index(max(ariile))

        imgRGB = cv2.cvtColor(inchisa, cv2.COLOR_BGR2RGB)
        x, y, w, h = cv2.boundingRect(contours[maxLoc])
    return x, y, w, h;

numeFereastră = 'Masca'

cv2.namedWindow(numeFereastră, cv2.WINDOW_AUTOSIZE)

cv2.createTrackbar("lowH", numeFereastră, 0, 179, nimic)
cv2.createTrackbar("highH", numeFereastră, 0, 179, nimic)

cv2.createTrackbar("lowS", numeFereastră, 0, 255, nimic)
cv2.createTrackbar("highS", numeFereastră, 0, 255, nimic)

cv2.createTrackbar("lowV", numeFereastră, 0, 255, nimic)
cv2.createTrackbar("highV", numeFereastră, 0, 255, nimic)

cap = cv2.VideoCapture(0)
```



```

while(True):
    #citim frame
    _ret, frame = cap.read()

    lowH = cv2.getTrackbarPos("lowH", numeFereastra)
    highH = cv2.getTrackbarPos("highH", numeFereastra)
    lowS = cv2.getTrackbarPos("lowS", numeFereastra)
    highS = cv2.getTrackbarPos("highS", numeFereastra)
    lowV = cv2.getTrackbarPos("lowV", numeFereastra)
    highV = cv2.getTrackbarPos("highV", numeFereastra)

    formatHSV = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

    masca = cv2.inRange(formatHSV, np.array([lowH, lowS, lowV]), np.
↪array([highH, highS, highV]))

    kernel = np.ones((3,3),np.uint8)
    deschisa = cv2.morphologyEx(masca, cv2.MORPH_OPEN, kernel)
    inchisa = cv2.morphologyEx(deschisa, cv2.MORPH_CLOSE, kernel)

    x, y, w, h = gasesteDreptunghi(inchisa)
    cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 3)

    # Afisam frame
    cv2.imshow('frame', frame)
    cv2.imshow(numeFereastra, inchisa)

    #apasam ESC pentru a incheia citirea
    if cv2.waitKey(1) == 27:
        break

cap.release()
cv2.destroyAllWindows()

```

8 Citire fisier video

```

[6]: import numpy as np
import cv2

cap = cv2.VideoCapture('D:\\Dota2.mp4')

while(cap.isOpened()):
    ret, frame = cap.read()

```

```
cv2.imshow('frame', frame)
if cv2.waitKey(1) == 27:
    break

cap.release()
cv2.destroyAllWindows()
```

[]: