

5. Others

1. Сегментація зображення з перетворенням відстані та алгоритмом вододілу

КОД:

```
import cv2

import numpy as np

import random as rng

img = cv2.imread("/home/rodion/yuliia0/aboba/cv/others/cart2.jpg")

w=int(img.shape[0]*50/100)

h=int(img.shape[1]*50/100)

d=(h,w)

img=cv2.resize(img,d)

cv2.imshow('original', img)

img[np.all(img == 255, axis=2)] = 0

cv2.imshow('Black Background', img)

ker = np.array([[1, 1, 1], [1, -8, 1], [1, 1, 1]], dtype=np.float32)

imgLaplacian = cv2.filter2D(img, cv2.CV_32F, ker)

sharp = np.float32(img)

imgResult = sharp - imgLaplacian

imgResult = np.clip(imgResult, 0, 255)

imgResult = imgResult.astype('uint8')

imgLaplacian = np.clip(imgLaplacian, 0, 255)
```

```
imgLaplacian = np.uint8(imgLaplacian)

cv2.imshow('New Sharped Image', imgResult)

bw = cv2.cvtColor(imgResult, cv2.COLOR_BGR2GRAY)
_, bw = cv2.threshold(bw, 40, 255, cv2.THRESH_BINARY | cv2.THRESH_OTSU)

cv2.imshow('Binary Image', bw)

dist = cv2.distanceTransform(bw, cv2.DIST_L2, 3)

cv2.normalize(dist, dist, 0, 1.0, cv2.NORM_MINMAX)

cv2.imshow('Distance Transform Image', dist)

_, dist = cv2.threshold(dist, 0.4, 1.0, cv2.THRESH_BINARY)

dist_8u = dist.astype('uint8')

contours, _ = cv2.findContours(dist_8u, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

markers = np.zeros(dist.shape, dtype=np.int32)

for i in range(len(contours)):

cv2.drawContours(markers, contours, i, (i+1), -1)

cv2.circle(markers, (5,5), 3, (255,255,255), -1)

markers_8u = (markers * 10).astype('uint8')

cv2.imshow('Markers', markers_8u)

cv2.watershed(imgResult, markers)

mark = markers.astype('uint8')
```

```
mark = cv2.bitwise_not(mark)

colors = []

for contour in contours:

    colors.append((rng.randint(0,256), rng.randint(0,256),
rng.randint(0,256)))

dst = np.zeros((markers.shape[0], markers.shape[1], 3), dtype=np.uint8)

for i in range(markers.shape[0]):

    for j in range(markers.shape[1]):

        index = markers[i,j]

        if index > 0 and index <= len(contours):

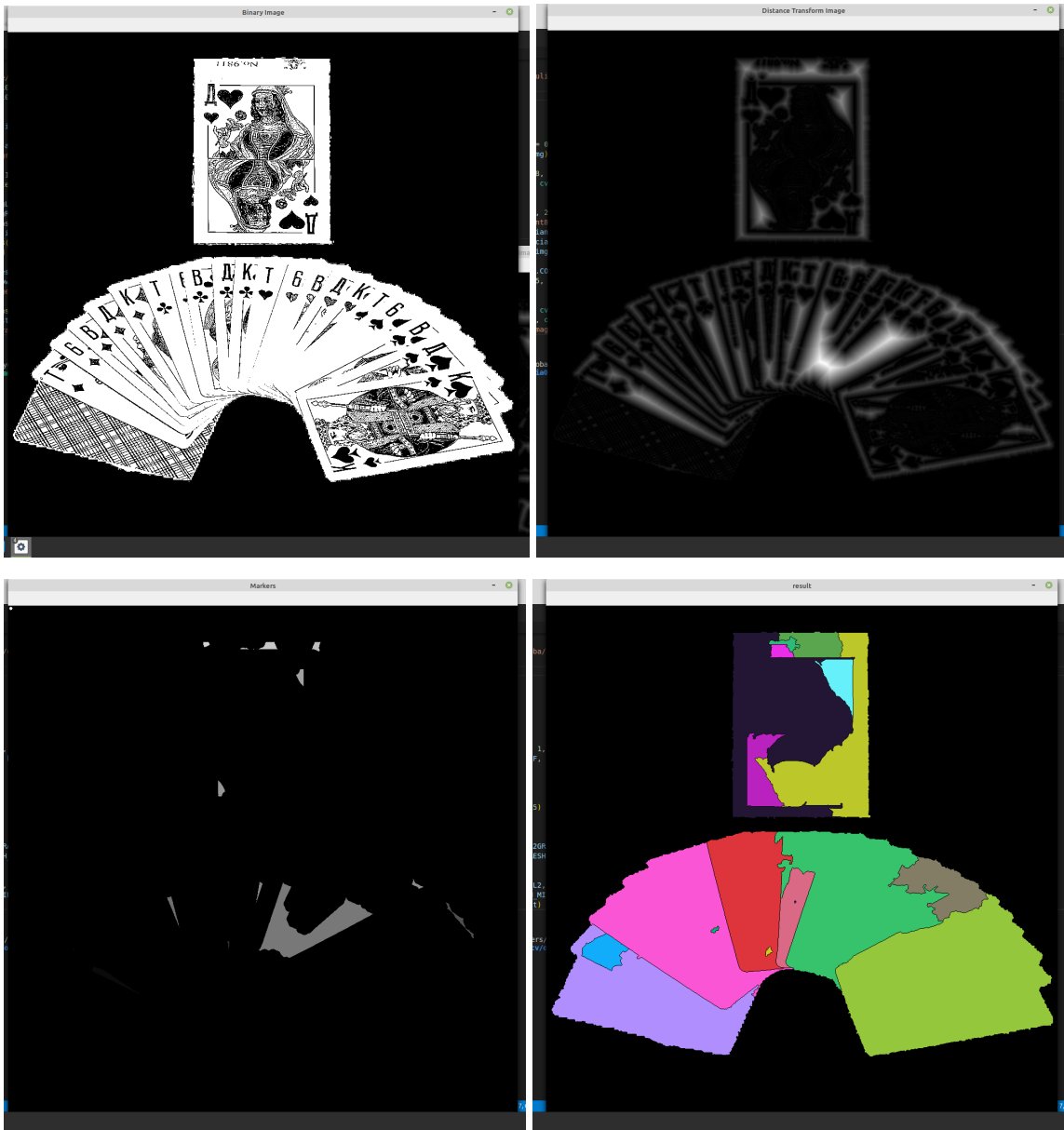
            dst[i,j,:] = colors[index-1]

cv2.imshow('result', dst)

cv2.waitKey(0)
```

результат:





2. Фільтр усунення розмиття поза фокусом

код: немає на python((

результат:

3. Фільтр усунення розмитості в русі

код:

результат:

4. Сегментація анізотропного зображення тензором градієнтної структури

КОД:

```
import cv2

import numpy as np

imgIn = cv2.imread('/home/rodion/yuliia0/aboba/cv/others/biba.jpg')

cv2.imshow('original', imgIn)

W = 52

def calcGST(inputIMG, w):

    img = inputIMG.astype(np.float32)

    imgDiffX = cv2.Sobel(img, cv2.CV_32F, 1, 0, 3)

    imgDiffY = cv2.Sobel(img, cv2.CV_32F, 0, 1, 3)

    imgDiffXY = cv2.multiply(imgDiffX, imgDiffY)

    imgDiffXX = cv2.multiply(imgDiffX, imgDiffX)

    imgDiffYY = cv2.multiply(imgDiffY, imgDiffY)

    J11 = cv2.boxFilter(imgDiffXX, cv2.CV_32F, (w,w))

    J22 = cv2.boxFilter(imgDiffYY, cv2.CV_32F, (w,w))

    J12 = cv2.boxFilter(imgDiffXY, cv2.CV_32F, (w,w))

    tmp1 = J11 + J22

    tmp2 = J11 - J22

    tmp2 = cv2.multiply(tmp2, tmp2)

    tmp3 = cv2.multiply(J12, J12)

    tmp4 = np.sqrt(tmp2 + 4.0 * tmp3)

    lambda1 = 0.5*(tmp1 + tmp4)

    lambda2 = 0.5*(tmp1 - tmp4)

    imgCoherencyOut = cv2.divide(lambda1 - lambda2, lambda1 + lambda2)
```

```
imgOrientationOut = cv2.phase(J22 - J11, 2.0 * J12, angleInDegrees =
True)

imgOrientationOut = 0.5 * imgOrientationOut

return imgCoherencyOut, imgOrientationOut


imgCoherency, imgOrientation = calcGST(imgIn, W)

_, imgCoherencyBin = cv2.threshold(imgCoherency, 0.43, 255,
cv2.THRESH_BINARY)

_, imgOrientationBin = cv2.threshold(imgOrientation, 35, 57,
cv2.THRESH_BINARY)

imgBin = cv2.bitwise_and(imgCoherencyBin, imgOrientationBin)

imgCoherency = cv2.normalize(imgCoherency, None, alpha=0, beta=1,
norm_type=cv2.NORM_MINMAX, dtype=cv2.CV_32F)

imgOrientation = cv2.normalize(imgOrientation, None, alpha=0, beta=1,
norm_type=cv2.NORM_MINMAX, dtype=cv2.CV_32F)

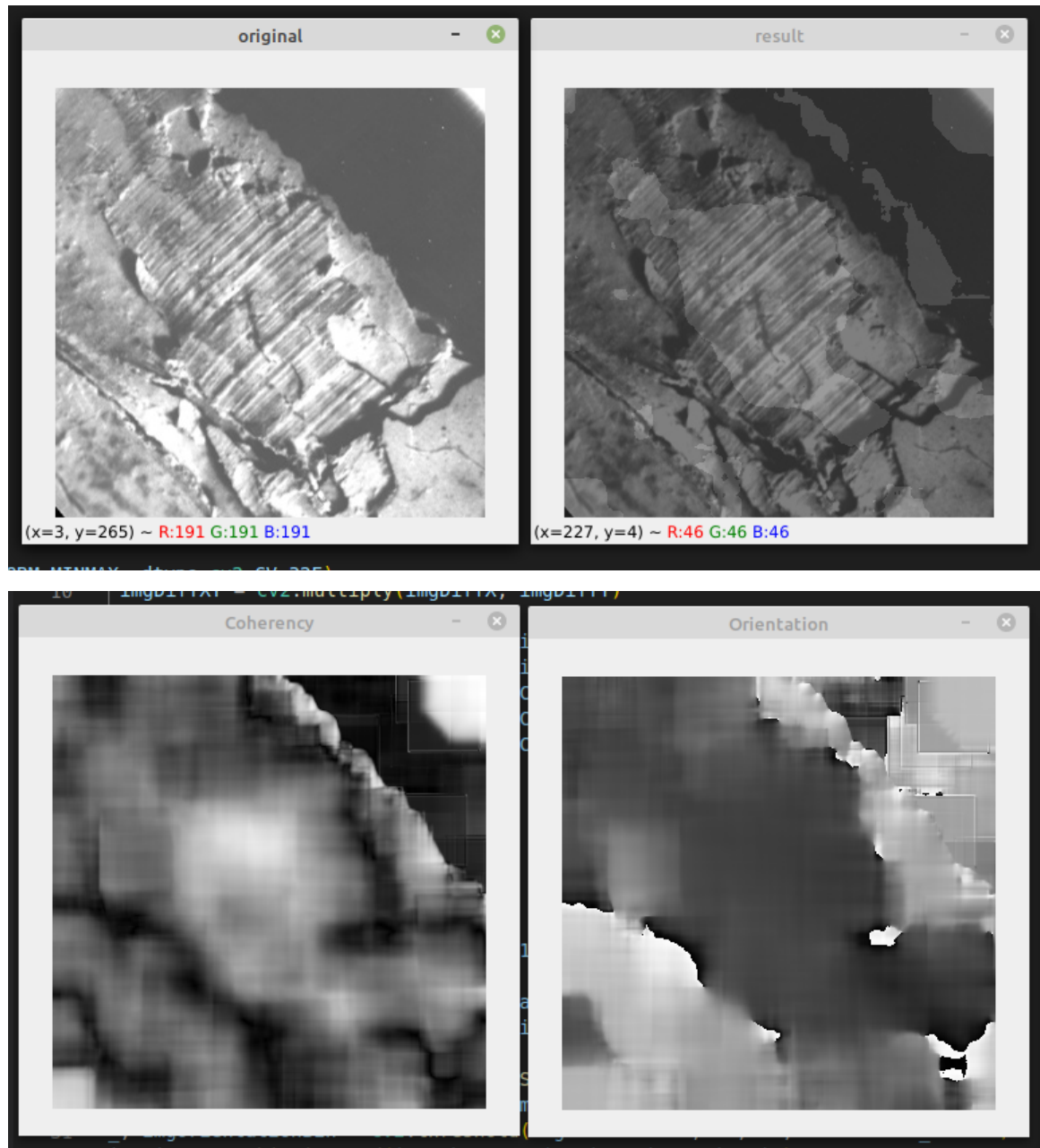

cv2.imshow('Coherency', imgCoherency)

cv2.imshow('Orientation', imgOrientation)

cv2.imshow('result', np.uint8(0.5*(imgIn + imgBin)))

cv2.waitKey(0)
```

результат:



5. Фільтр періодичного шумозаглушення

код:

результат: