

머신비전시스템 과제3

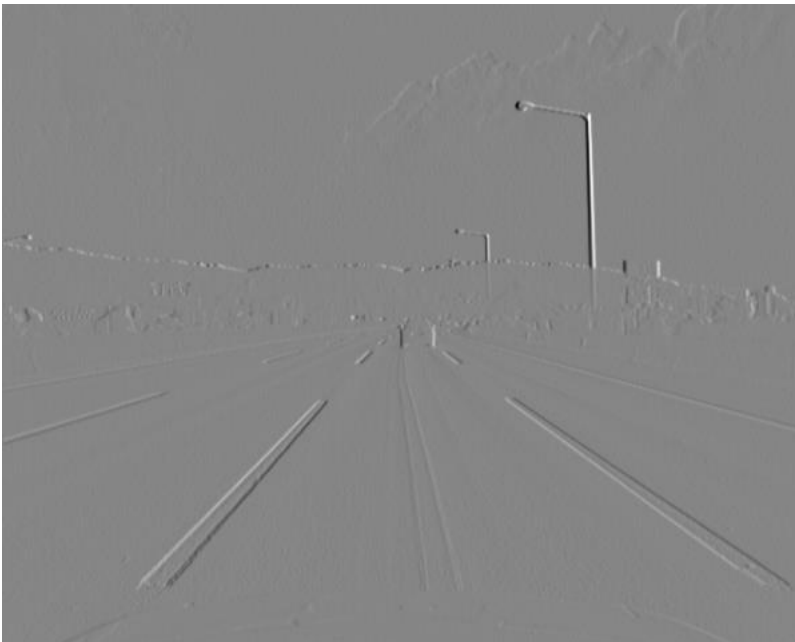
18011789 조혜수

1. Sobel filter Gradient

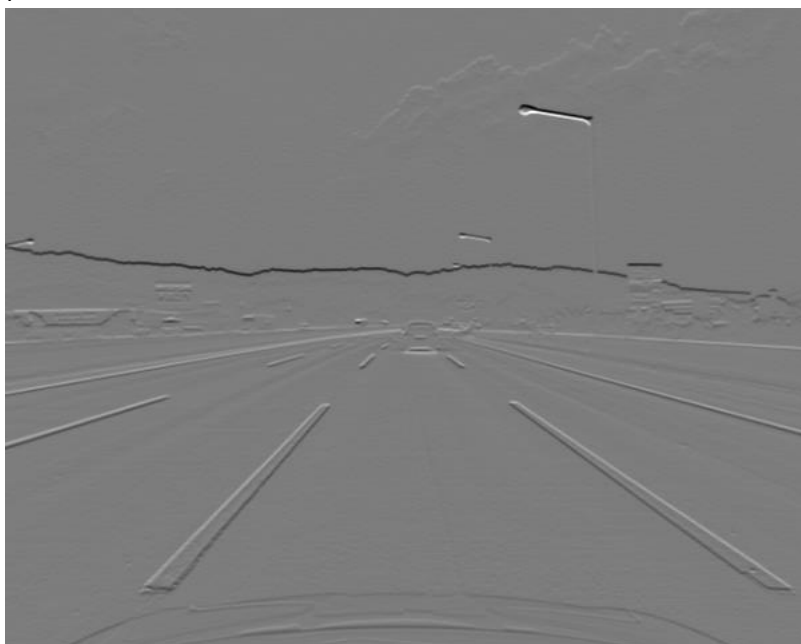
원본(흑백)



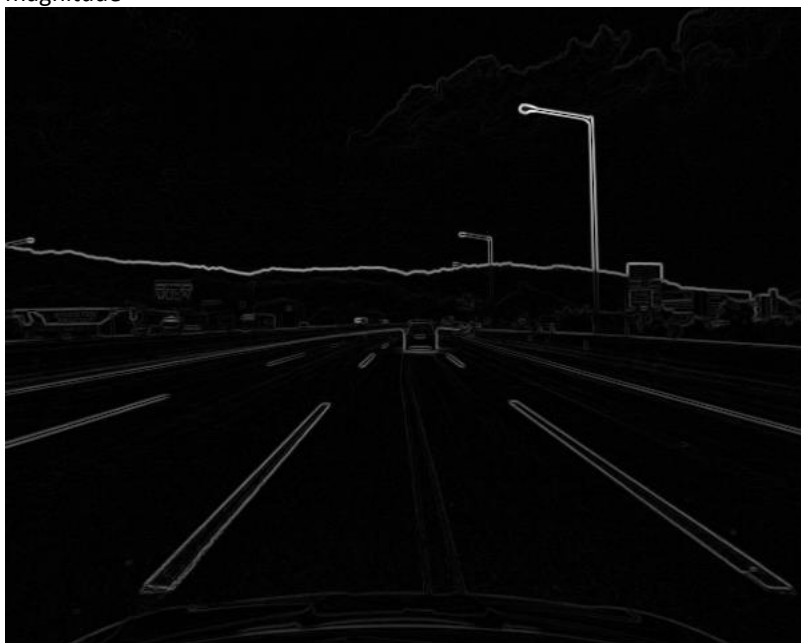
x방향 미분 영상



y방향 미분 영상



magnitude



orientation



코드

```
[1] from google.colab import drive
drive.mount('/content/gdrive')

Mounted at /content/gdrive

[2] import cv2 as cv
import numpy as np
from google.colab.patches import cv_imshow

img = cv.imread('/content/gdrive/MyDrive/MachineVision/lanes.bmp', cv.IMREAD_GRAYSCALE)

[3] Ix = cv.Sobel(img, ddepth=cv.CV_64F, dx=1, dy=0, ksize=3)
Iy = cv.Sobel(img, ddepth=cv.CV_64F, dx=0, dy=1, ksize=3)
mag = np.sqrt(np.square(Ix) + np.square(Iy))
ori = np.arctan2(Iy, Ix) * 180/np.pi

[4] # display 하기 위해 값의 범위 0-255로 변경
Ix_ = (Ix - Ix.min()) / (Ix.max() - Ix.min()) * 255
Iy_ = (Iy - Iy.min()) / (Iy.max() - Iy.min()) * 255
mag_ = (mag - mag.min()) / (mag.max() - mag.min()) * 255
ori_ = (ori - ori.min()) / (ori.max() - ori.min()) * 255

[9] cv_imshow(img)
cv_imshow(Ix_)
cv_imshow(Iy_)
cv_imshow(mag_)
cv_imshow(ori_)
```

A small grayscale image showing a close-up of a road lane marking, likely the result of the orientation estimation process.

2. 차선 검출

원본 (위와 동일)

좌측차선



우측차선



코드

```
[80] result_left = np.zeros(img.shape)
    id_L = np.where((mag>100) & (ori>30) & (ori<60))
    result_left[id_L] = 255

    #cv_imshow(result_left)

result_right = np.zeros(img.shape)
id_R = np.where((mag>100) & (ori>-60) & (ori<-30))
result_right[id_R] = 255

#cv_imshow(result_right)
```

3. RANSAC

원본 : 동일

좌측 라인 피팅



우측 라인 피팅



코드

```

✓ [135] import numpy as np
0초 import matplotlib.pyplot as plt

def f(x,a,b):
    return a*x+b

def ransac_line_fiting(x,y,r,t):
    iter = np.round(np.log(1-0.999) / np.log(1-(1-r)**2) + 1)
    num_max = 0
    for i in np.arange(iter):
        id = np.random.permutation(len(x))
        xs = x[id[:2]]
        ys = y[id[:2]]
        A = np.vstack([xs, np.ones(len(xs))]).T
        ab = np.dot(np.linalg.inv(np.dot(A.T, A)), np.dot(A.T, ys))
        dist = np.abs(ab[0]*x-y+ab[1])/np.sqrt(ab[0]**2+1)
        numInliers = sum(dist < t)
        if numInliers > num_max:
            ab_max = ab
            num_max = numInliers
    return ab_max, num_max

```

```

✓ [136] #작측차선 Line fitting
0초 abno_L, max = ransac_line_fiting(id_L[1],id_L[0],0.1,2)
    abno_L

array([ -1.01515152, 538.75757576])

```

```

✓ [137] id_L[1]
0초 array([392, 391, 20, 21, 490, 562, 6, 262, 263, 346, 323, 273, 272,
        273, 271, 272, 270, 271, 269, 270, 268, 269, 267, 268, 265, 266,
        267, 378, 237, 235, 236, 234, 235, 233, 234, 232, 233, 231, 232,
        230, 231, 229, 230, 228, 229, 227, 228, 226, 227, 228, 225, 226,
        227, 224, 225, 226, 223, 224, 225, 222, 223, 224, 221, 222, 223,
        220, 221, 222, 219, 220, 221, 218, 219, 220, 217, 218, 219, 217,
        218, 216, 217, 215, 216, 214, 215, 213, 214, 212, 213, 211, 212,

```

✓ [138] `img_tmp_L = img`
`y=f(100,abno_L[0],abno_L[1])`
`y1=f(270,abno_L[0],abno_L[1])`
`result_L=cv.line(img_tmp_L,(100,int(y)),(270,int(y1))),(0,255,255),5)`
`cv_imshow(result_L)`

✓ [139] #우측차선 Line fitting
0초 `abno_R, max = ransac_line_fiting(id_R[1],id_R[0],0.1,2)`
`abno_R`

`array([0.74226804, 14.46391753])`

✓ [140]  `id_R[1]`
0초

✓ [140] `img_tmp_R = img1`
`y=f(385,abno_R[0],abno_R[1])`
`y1=f(570,abno_R[0],abno_R[1])`
`result_R=cv.line(img_tmp_R,(385,int(y)),(570,int(y1))),(0,255,0),5)`
`cv_imshow(result_R)`