# fix_blue

December 4, 2020

[1]:
```
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt
from ROOT import TGraph2D,TRandom,TCanvas,TF2
import math
```

PIL : pillow , 이미지 불러오기

Numpy, plt, ROOT, math

```
Welcome to JupyROOT 6.22/02
```

[2]:
```
img=('20201204_170717.jpg')
img
```

[2]: '20201204_170717.jpg'

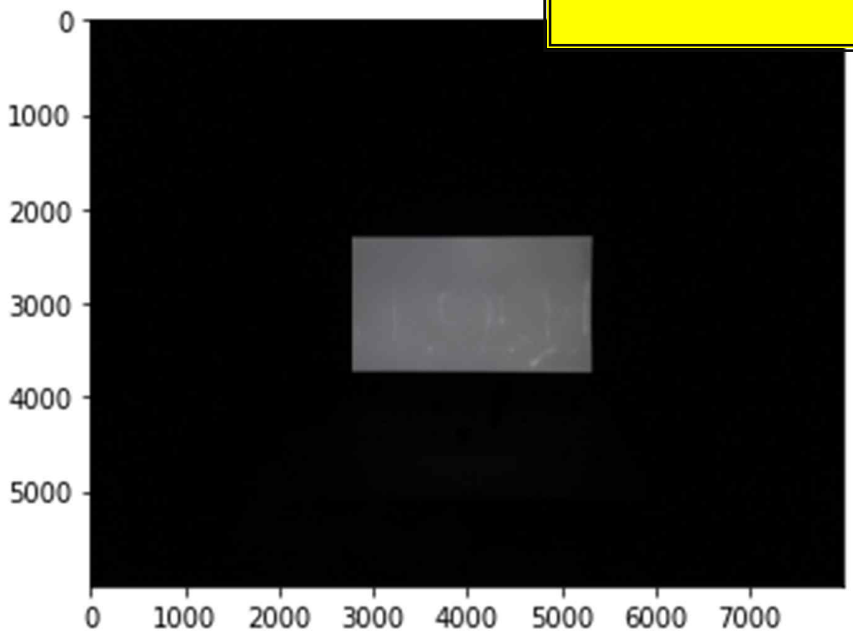
[3]:
```
im=Image.open(img)

pix = np.array(im)

pix.size
```

[3]: 144000000

[4]:
```
plt.imshow(im, cmap='gray')
```

[4]: <matplotlib.image.AxesImage at 0x7fed0831

불러온 픽셀 데이터



[5]:

```
pix
```

[5]:
```
array([[[1, 1,
1], [1, 1, 1],
      [1, 1, 1],
      …,
      [2, 2, 0],
      [2, 2, 0],
      [1, 1, 0]],

     [[1, 1, 1],
      [1, 1, 1],
      [1, 1, 1],
      …,
      [2, 2, 0],
      [2, 2, 0],
      [1, 1, 0]],

     [[0, 0, 0],
      [1, 1, 1],
      [2, 2, 2],
      …,
      [2, 2, 0],
      [2, 2, 0],
      [1, 1, 0]],
     …,

     [[0, 2, 1],
      [0, 2, 1],
      [0, 2, 1],
      …,
      [2, 2, 2],
      [1, 1, 1],
      [0, 0, 0]],

     [[0, 2, 1],
      [0, 2, 1],
      [0, 2, 1],
      …,
      [2, 2, 2],
      [1, 1, 1],
      [0, 0, 0]],

     [[0, 1, 0],
      [0, 1, 0],
```

```
        [0, 1, 0],
        …,
        [1, 1, 1],
        [1, 1, 1],
         [0, 0, 0]]], dtype=uint8)
```

[6]:
```
pix[1500][2500]
```

[6]:
```
array([1, 1, 1], dtype=uint8)
```

[7]:
```
pix[2500][5300][1]
```

[7]:
```
84
```

[8]:
```
np.min(pix[:][:][2])
```

[8]:
```
0
```

[9]:
```
np.max(pix[:][:][2])
```

[9]:
```
5
```

[10]:
```
r=len(pix)
```

[11]:
```
c=len(pix[0])
```

[ ]:

[ ]:

[ ]:

불러온 픽셀 데이터에서 노트북화면에 해당하는 영역을 인식

밝기가 어둡지 않은 부분의 픽셀 중 이미지의 가운데에서 가장 멀리 있는 점을 찾는다

[12]:
```
D=0
r=len(pix)
c=len(pix[0])
point0=np.zeros(2)
point1=np.zeros(2)
point2=np.zeros(2)
point3=np.zeros(2)
for i in range(int(r/2)):
    for j in range(int(c/2)):
        if pix[i][j][2] > 60:
            D=max([D,(i-r/2)**2+(j-c/2)**2])
for i in range(int(r/2)):
    for j in range(int(c/2)):
        if pix[i][j][2] > 60 and ((i-r/2)**2+(j-c/2)**2) == D:
            point0=[j,i]
D=0
for i in range(int(r/2)):
    for j in range(int(c/2),int(c)):
        if pix[i][j][2] > 60:
            D=max([D,(i-r/2)**2+(j-c/2)**2])
for i in range(int(r/2)):
    for j in range(int(c/2),int(c)):
        if pix[i][j][2] > 60 and ((i-r/2)**2+(j-c/2)**2) == D:
            point1=[j,i]
D=0
for i in range(int(r/2),int(r)):
    for j in range(int(c/2)):
        if pix[i][j][2] > 60:
            D=max([D,(i-r/2)**2+(j-c/2)**2])
for i in range(int(r/2),int(r)):
    for j in range(int(c/2)):
        if pix[i][j][2] > 60 and ((i-r/2)**2+(j-c/2)**2) == D:
            point2=[j,i]
D=0
for i in range(int(r/2),int(r)):
    for j in range(int(c/2),int(c)):
        if pix[i][j][2] > 60:
            D=max([D,(i-r/2)**2+(j-c/2)**2])
for i in range(int(r/2),int(r)):
    for j in range(int(c/2),int(c)):
        if pix[i][j][2] > 60 and ((i-r/2)**2+(j-c/2)**2) == D:
```

```
            point3=[j,i]
point0, point1, point2,point3
```

[12]: ([2777, 2301], [5336, 2291], [27

[13]:
```
import cv2
r,c
```

[13]: (6000, 8000)

cv2 :
OpenCV, 이미지의 일부를 자르고 보정함

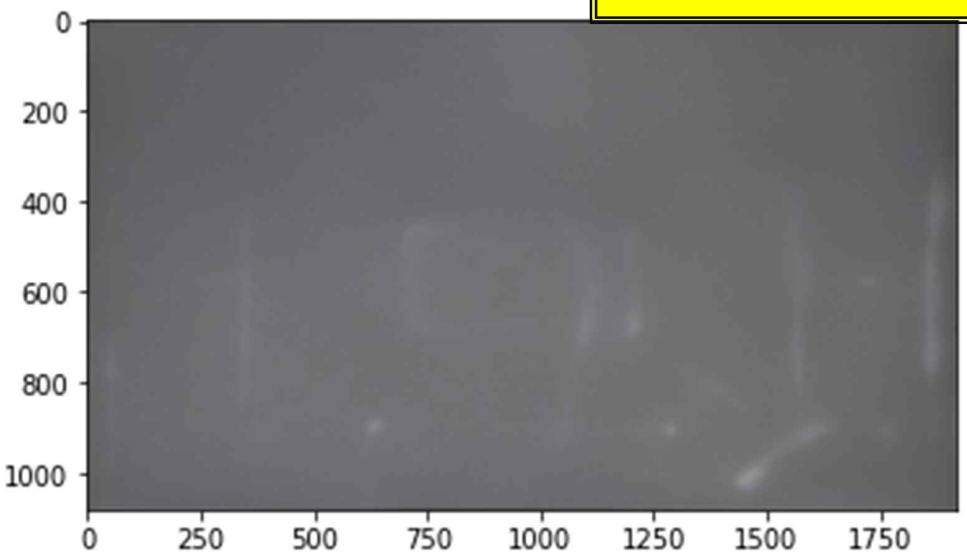
[14]:
```
scr_np=np.array([point0,point2,point3,point1],dtype=np.float32)
dst_np=np.array([[0,0],[0,1080],[1920,1080],[1920,0]],dtype=np.float
32)
```

[15]:
```
M= cv2.getPerspectiveTransform(scr_np ,
dst_np) get_scr =
cv2.warpPerspective(pix,M,(1920,1080))
```

[16]:
```
get_scr=np.array(get_scr)
plt.imshow(get_scr, cmap='gray')
```

배경은 잘라내고 노트북화면에 해당하는 데이터만 얻어냄

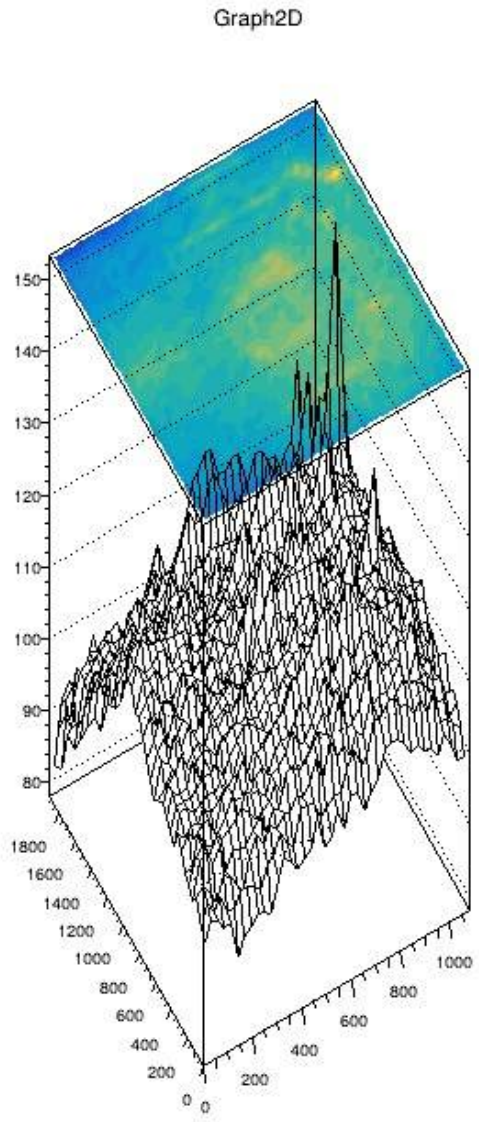
[16]: <matplotlib.image.AxesImage at 0x7fecdf42




[17]:
```
r=len(get_scr)
c=len(get_scr[0])
get_scr[727][965]
```

[17]: array([115, 115, 117], dtype=uint8)

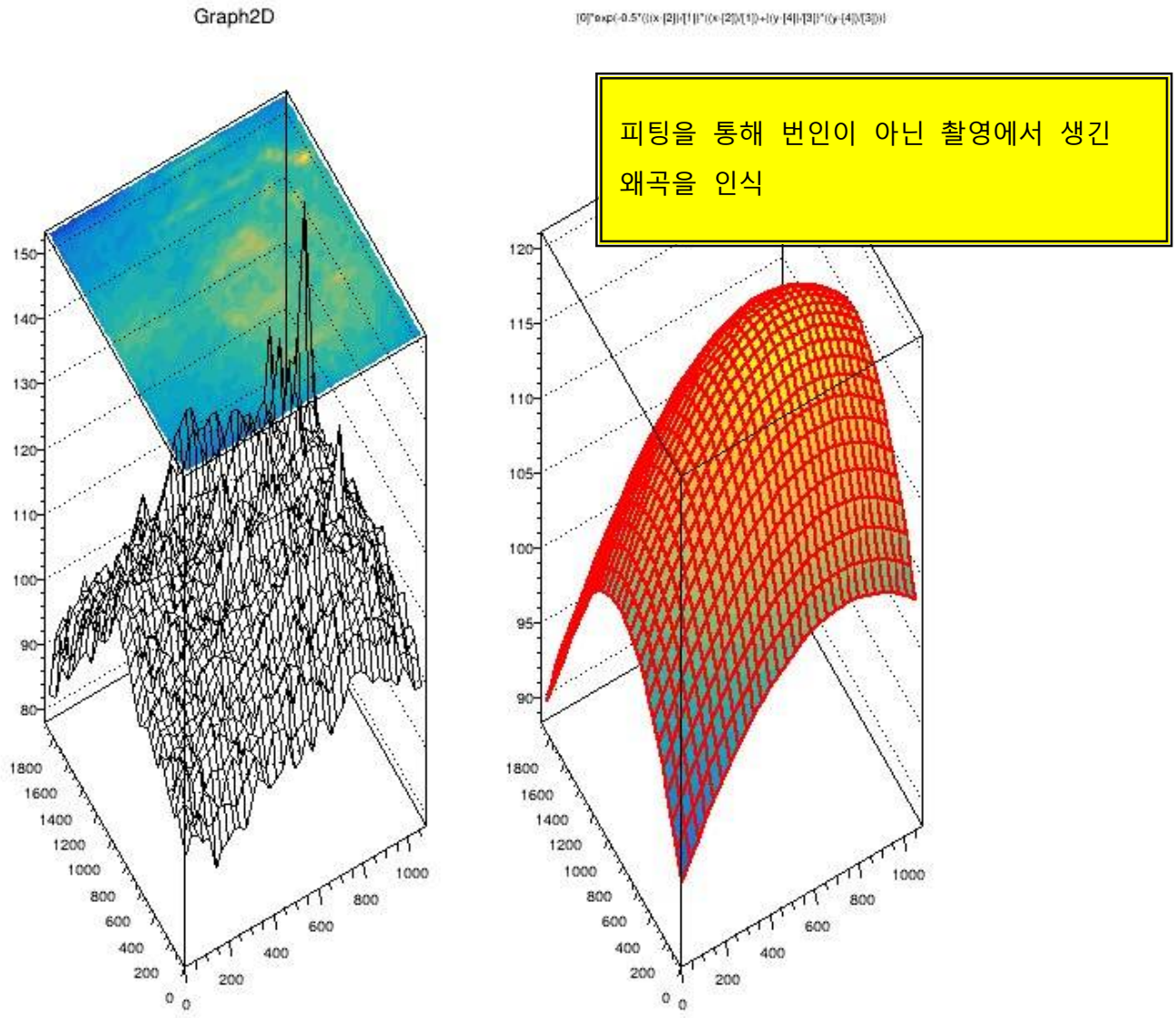
[18]:
```
c2d = TCanvas("c2d","Graph2D example",0,0,600,700)
c2d.Divide(2)

dt = TGraph2D()
n=0
for i in range(r):
    for j in range(c):
        x=i
        y=j
        z = get_scr[i][j][2]
        n=n+1
        dt.SetPoint(n,x,y,z);

c2d.cd(1)
dt.Draw("surf3")#use "surf1" to generate the left picture
c2d.Draw()
```




노트북 화면의 밝기를 3 차원으로 PLOT

[19]:
```
f=TF2('f',"[0]*exp(-0.5*(((x-[2])/[1])*((x-[2])/[1])+((y-[4])/[3])*((y-[4])/
 ↪[3])))",0,1080,0,1920)
f.SetParameters(160,600,1000,1000,2000)
c2d.cd(2)
dt.Fit(f,'R')
f.Draw("surf1")
c2d.Draw()
```

```
****************************************
Minimizer is Minuit / Migrad
Chi2                      = 6.13049e+07
NDf                       =     2073596

Edm                       = 1.06049e-05

NCalls                    =         289

p0                        =     118.125       +/-0.00696268
p1                        =     1474.68       +/-1.2912
p2                        =     745.771       +/-0.424256
p3                        =     1817.77       +/-0.776389
p4                        =     887.284       +/-0.223214
```




[20]:
```
par=np.array([f.GetParameters()[0],f.GetParameters()[1],f.GetParameters()[2],f.
 ↪GetParameters()[3],f.GetParameters()[4]]) par
```

[20]: array([ 118.12488537, 1474.67801616, 745.77089451, 1817.77439697,
887.28387919])

[21]:
```
i=727
j=965
fit=par[0]*math.exp(-0.5*(((i-par[2])/par[1])*((i-par[2])/par[1])+((j-par[4])/
 ↪par[3])*((j-par[4])/par[3])))
data=get_scr[j][i][2]
fit,data
```

[21]: (118.00741665276256, 120)

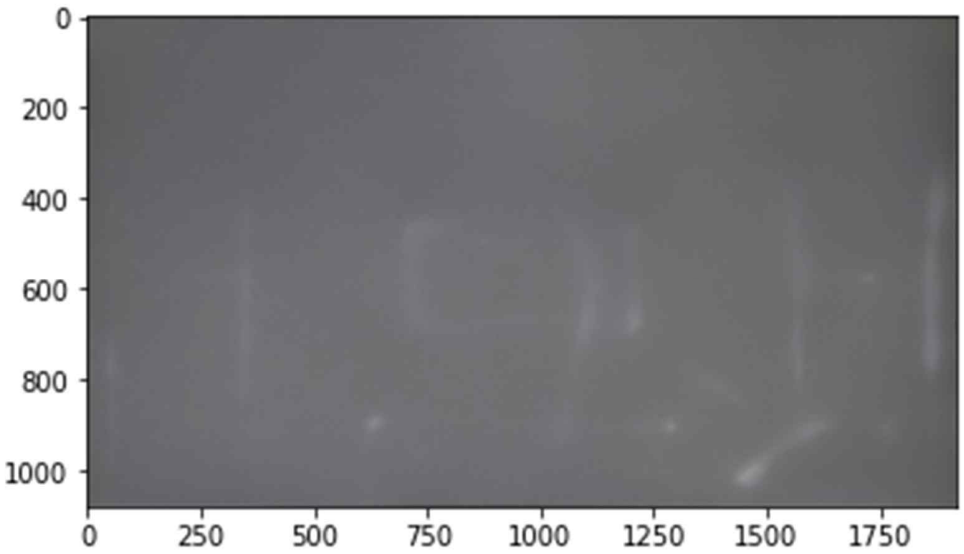
[22]:
```
scr_np
```

[ ]:

[22]:
```
array([[2777., 2301.],
       [2781., 3732.],
       [5335., 3741.],
       [5336., 2291.]], dtype=float32)
```

[23]:
```
plt.imshow(get_scr)
```

[23]: <matplotlib.image.AxesImage at 0x7fec9c583670>




[24]:
```
cv2.imwrite("new_blue.jpg",get_scr)
get_scr
```

[24]:
```
array([[[ 67,  67,
    67], [ 96,  96,  96],
```

```
 [100, 100, 100],
 …,
 [ 67, 67, 67],
 [ 70, 70, 70],
 [ 72, 72, 72]],

[[ 78, 78, 78],
 [104, 104, 104],
 [ 97, 97, 97],
 …,
 [ 76, 76, 74],
 [ 81, 81, 81],
 [ 89, 89, 89]],

[[ 78, 78, 78],
 [100, 100, 100],
 [ 93, 93, 92],
 …,
 [ 75, 75, 73],
 [ 78, 78, 76],
 [ 86, 86, 86]],
…,

[[100, 102, 101],
 [108, 110, 109],
 [ 96, 97, 98],
 …,
 [ 95, 95, 97],
 [ 99, 99, 101],
 [112, 112, 114]],

[[ 98, 100, 99],
 [110, 112, 111],
 [102, 103, 103],
 …,
 [ 98, 98, 100],
 [102, 102, 104],
 [111, 111, 113]],

[[ 94, 96, 95],
 [108, 110, 109],
 [104, 105, 106],
 …,
 [107, 107, 109],
 [106, 106, 108],
 [108, 108, 110]]], dtype=uint8)
```

[25]:
```
new_average=(np.average(np.average(get_scr, axis =
0),axis=0))[1] new_stv=((np.average(np.var(get_scr, axis
=0),axis=0))**(1/2))[1] new_average, new_stv
c=len(get_scr[0]) r=len(get_scr) r,c,new_average,
new_stv
```

[25]: (1080, 1920, 106.11293354552467, 6.19345632013888)

[26]:
```
            fix_blue=np.zeros((r,c,3),
dtype="float16") for i in range(r):
for j in range(c):
       if get_scr[i][j][2]<(new_average-6*new_stv):
           fix_blue[i][j][2]=1
       elif get_scr[i][j][2]>(new_average-6*new_stv) :
           fix_blue[i][j][2]=(par[0]*math.exp(-0.5*(((i-par[2])/
 ↪par[1])*((i-par[2])/par[1])+((j-par[4])/par[3])*((j-
 par[4])/par[3])))))/ ↪get_scr[i][j][2]
```

[27]:
```
np.min(fix_blue[fix_blue>0])
```

[27]: 0.7095

[28]:
```
np.max(fix_blue)
```

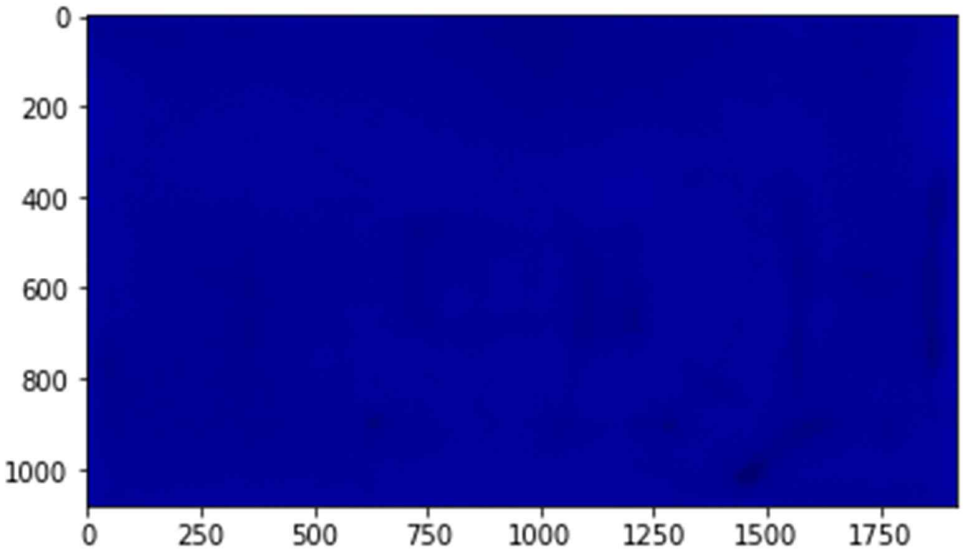
[28]: 1.669

[29]:
```
normalize_blue=fix_blue*255/np.max(fix_blue)
```

[30]:
```
fix_blue_img=normalize_blue.astype(np.uint8)
```

[31]:
```
plt.imshow(fix_blue_img, cmap='gray')
im= Image.fromarray(fix_blue_img)
im.save("fix_blue_img.jpg")
```

데이터와 피팅그래프를 이용하여, 화면을 보정할 행렬(화면 픽셀비율)을 만들어냄




[32]:
```
D=0

point0=np.zeros(2)
point1=np.zeros(2)
point2=np.zeros(2)
point3=np.zeros(2)
for i in range(int(r/2)):
    for j in range(int(c/2)):
        if fix_blue_img[i][j][2] > 10:
            D=max([D,(i-r/2)**2+(j-c/2)**2])
for i in range(int(r/2)):
    for j in range(int(c/2)):
        if fix_blue_img[i][j][2] > 10 and ((i-r/2)**2+(j-c/2)**2) == D:
```

```
            point0=[j,i]
D=0
for i in range(int(r/2)):
    for j in range(int(c/2),int(c)):
        if fix_blue_img[i][j][2] > 10:
            D=max([D,(i-r/2)**2+(j-c/2)**2])
for i in range(int(r/2)):
    for j in range(int(c/2),int(c)):
        if fix_blue_img[i][j][2] > 10 and ((i-r/2)**2+(j-c/2)**2) == D:
            point1=[j,i]
D=0
for i in range(int(r/2),int(r)):
    for j in range(int(c/2)):
        if fix_blue_img[i][j][2] > 10:
            D=max([D,(i-r/2)**2+(j-c/2)**2])
for i in range(int(r/2),int(r)):
    for j in range(int(c/2)):
        if fix_blue_img[i][j][2] > 10 and ((i-r/2)**2+(j-c/2)**2) == D:
            point2=[j,i]
D=0
for i in range(int(r/2),int(r)):
    for j in range(int(c/2),int(c)):
        if fix_blue_img[i][j][2] > 10:
            D=max([D,(i-r/2)**2+(j-c/2)**2])
for i in range(int(r/2),int(r)):
    for j in range(int(c/2),int(c)):
        if fix_blue_img[i][j][2] > 10 and ((i-r/2)**2+(j-c/2)**2) == D:
            point3=[j,i]
point0, point1, point2,point3
```

[32]:
```
([0, 0], [1919, 0], [0, 1079], [1919, 1079])
```

[33]:
```
scr_np=np.array([point0,point2,point3,point1],dtype=np.float32)
dst_np=np.array([[0,0],[0,1080],[1920,1080],[1920,0]],dtype=np.float
32)
```

[34]:
```
M= cv2.getPerspectiveTransform(scr_np , dst_np)
fix_complete =
cv2.warpPerspective(fix_blue_img,M,(1920,1080))
fix_complete[1000][500]
```

[34]:
```
array([  0,   0, 154], dtype=uint8)
```

[35]:
```
im= Image.fromarray(fix_complete)
im.save("blue_complete.jpg")
```

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: