histogram_stretching_technical_report

introduction

Grayscale 이미지를 stretching하고, stretching에 사용된 전이함수와 결과 이미지의 PDF, 오리지널 이미지의 PDF를 plot한다.

코드 구현

hist_stretching.cpp

main 함수

```
1)
  Mat input = imread("input.jpg", IMREAD_COLOR);
    Mat input_gray;

    cvtColor(input, input_gray, COLOR_RGB2GRAY);    // convert RGB to Grayscale

    Mat stretched = input_gray.clone();

    // PDF or transfer function txt files
    FILE *f_PDF;
    FILE *f_stretched_PDF;
    FILE *f_trans_func_stretch;

    fopen_s(&f_PDF, "PDF.txt", "w+");
    fopen_s(&f_stretched_PDF, "stretched_PDF.txt", "w+");
    fopen_s(&f_trans_func_stretch, "trans_func_stretch.txt", "w+");

    G trans_func_stretch[L] = { 0 };

2)
    float *PDF = cal_PDF(input_gray);

3)
    linear_stretching(input_gray, stretched, trans_func_stretch, 50, 110, 10, 110);  // histogram stret
    float *stretched_PDF = cal_PDF(stretched);
```

1. input 이미지 로드, rgb를 grayscale로 변환

   stretched할 이미지를 오리지널 이미지를 clone해서 생성
2. grayscale로 변환된 오리지널 이미지의 PDF 계산
3. linear_stretching 진행.

```
void linear_stretching(Mat &input, Mat &stretched, G *trans_func, G x1, G x2, G y1, G y2) {

    float constant = (y2 - y1) / (float)(x2 - x1);

1)
    // compute transfer function
    for (int i = 0; i < L; i++) {
        if (i >= 0 && i <= x1)
            trans_func[i] = (G)(y1 / x1 * i);
        else if (i > x1 && i <= x2)
            trans_func[i] = (G)(constant * (i - x1) + y1);
        else
            trans_func[i] = (G)((L - 1 - x2) / (L - 1 - y2) * (i - x2) + y2);
    }

2)
    // perform the transfer function
    for (int i = 0; i < input.rows; i++)
        for (int j = 0; j < input.cols; j++)
            stretched.at<G>(i, j) = trans_func[input.at<G>(i, j)];
}
```

1. 전이함수 계산

   x1, x2는 인풋 이미지 intensity의 최솟값과 최댓값을 정의하고

   y1, y2는 아웃풋 이미지 intensity의 최솟값과 최댓값을 정의한다.

$$y = \frac{b_{i+1} - b_i}{a_{i+1} - a_i}(x - a_i) + b_i$$

   const는 위의 식에서 x의 계수에 해당한다.

   해당 공식을 이용해 transfer function을 계산한다.

2. 전이함수 적용

   인풋 이미지의 row, col의 픽셀값을 trans_func에 대입하여, 아까 클론한 이미지 픽셀값에 대응시킨다.

위의 두 과정을 통해 이미지 stretching을 할 수 있다.

stretched_PDF_trans.py

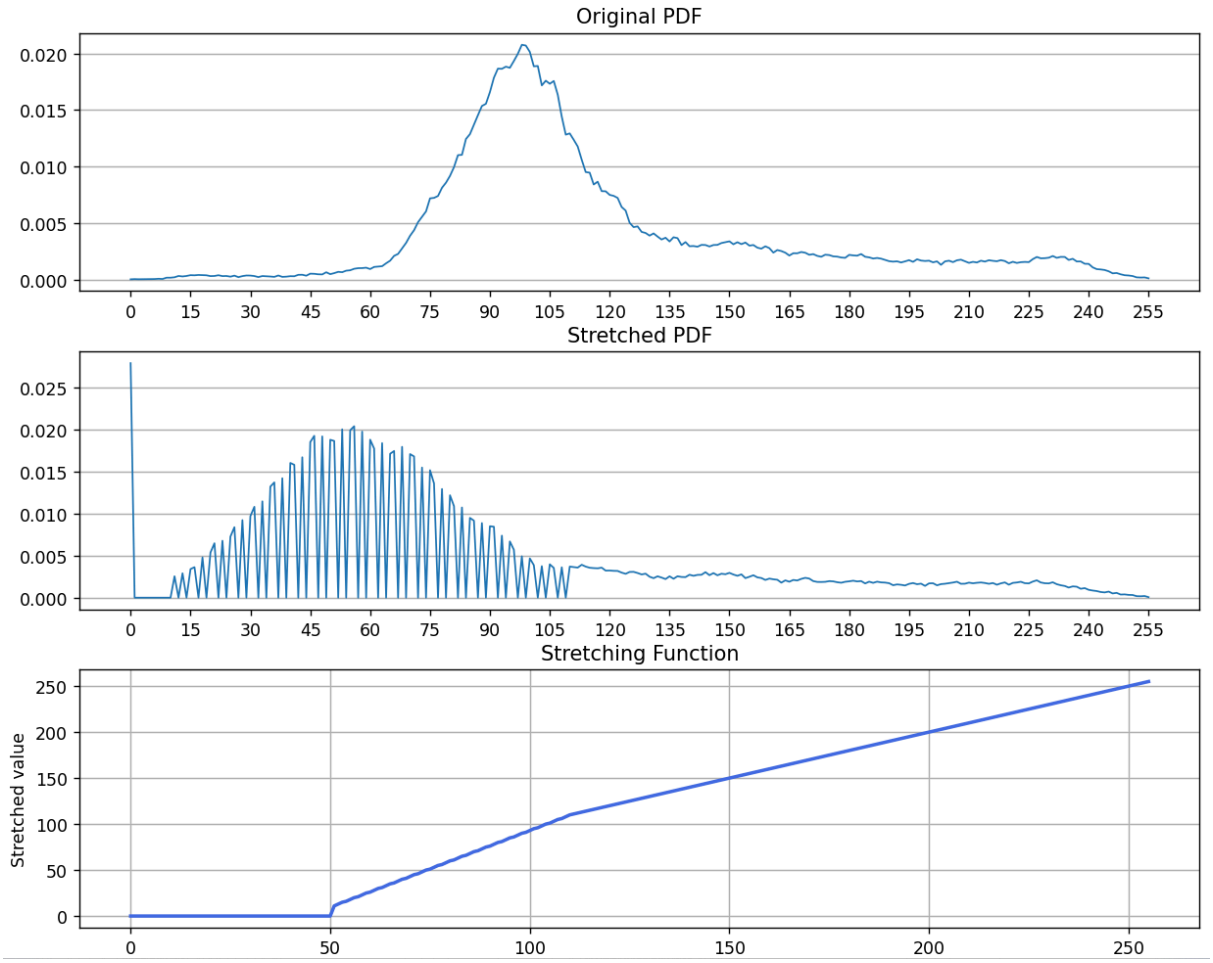
```
import matplotlib.pyplot as plt
import numpy as np

# 데이터 불러오기
```

```
pdf = np.loadtxt("PDF.txt", delimiter="\t", usecols=1)
st_pdf = np.loadtxt("stretched_PDF.txt", delimiter="\t", usecols=1)
trans_func = np.loadtxt("trans_func_stretch.txt")
x = np.arange(256)
y_trans = trans_func[:, 1]


fig, axs = plt.subplots(3, 1, figsize=(10, 10))

# Original PDF
axs[0].plot(x, pdf, linewidth=1)
axs[0].set_title("Original PDF")
axs[0].set_xticks(range(0, 256, 15))
axs[0].grid(True, axis='y')

# Stretched PDF
axs[1].plot(x, st_pdf, linewidth=1)
axs[1].set_title("Stretched PDF")
axs[1].set_xticks(range(0, 256, 15))
axs[1].grid(True, axis='y')

# Stretching Function
axs[2].plot(x, y_trans, linewidth=2, color='royalblue')
axs[2].set_title("Stretching Function")
axs[2].set_xlabel("Original value")
axs[2].set_ylabel("Stretched value")
axs[2].grid(True)


plt.tight_layout()
plt.show()
```

오리지널 이미지의 PDF, Stretching을 거친 이미지의 PDF, 전이함수를 시각적으로 표현하는 python code

Result

plot한 결과는 다음과 같다.



원본 이미지의 PDF

PDF 분석 결과, 대부분의 intensity가 90과 105 사이에 몰려있다. 그렇기 때문에 픽셀이 중간보다 약간 어두운 영역에 몰려있고, 전체적인 이미지의 대비가 낮다고 볼 수 있다.

stretched 된 이미지의 PDF

histogram stretching을 진행할 때, output image intensity의 min값을 10, max 값을 110으로 놓았는데, 그 range 상에서 골고루 intensity들이 존재함을 알 수 있다.

Stretching function

linear한 함수 형태로 나타난다

결론

차례로 input 이미지와 stretching 진행한 이미지다.

좀 더 선명해지고, 밝은 부분의 이미지가 더 어둡게 진해지고 어두운 부분은 더 어두워졌다.