Homework#8

2015004302 곽상원

목차

---

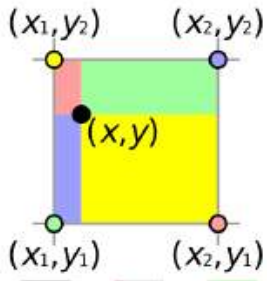
1. Purpose of program

- 이미지를 **bilinear interpolation**을 이용해 resampling한다.
- **Bilinear interpolation**: 2차원 평면에서 f(x,y)를 구하기 위해, 주변 4개의 주어진 data를 이용해 interpolation하는 방식이다.

2 .Experimental process

- lena.jpg를 이미지 파일로 이용한다.

▶bilinear interpolation 구현



가로, 세로 S배된 resampled image에 대해, (x', y')위치의 data는 원래 image의 (x'/S, y'/s)의 data와 같다. 즉, (x'/S, y'/s)의 data를 bilinear interpolation을 이용해서 구한다.

$x'/S - x_1 = a, y'/S - y_1 = b$라고 하면, 해당 위치에서의 $data$는

$$f(x'/S,y'/S) = (1-a)(1-b)f(x_1,y_1) + a(1-b)f(x_2,y_1) + (1-a)bf(x_1,y_2) + abf(x_2,y_2)$$

이때, 각 꼭짓점 쪽 가장자리에서 x1 = x2 or y1 = y2인 경우가 발생한다. 이런 경우는 bilinear대신 linear interpolation을 적용한다.

< Code >

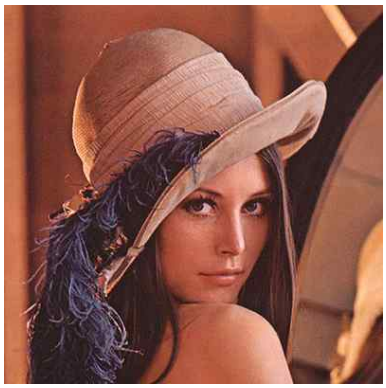
```
void BilinearInterpolation(cv::Mat &src, cv::Mat &dst,double scale_rate){
        double s = scale_rate; //scale_rate short notation
        int x[2], y[2]; //주위의 4점의 좌표
        double a, b; // 거리
        for (int j = 0; j < dst.cols; j++){
                for (int i = 0; i < dst.rows; i++) {
                        x[0] = (int)(i / s);
                        y[0] = (int)(j / s);
                        a = i / s - (double)x[0];
                        b = j / s - (double)y[0];

                        if (x[0] < src.rows-1) x[1] = x[0] + 1;
                        else x[1] = x[0];
                        if (y[0] < src.cols-1) y[1] = y[0] + 1;
                        else y[1] = y[0];

                        for (int k = 0; k < 3; k++) {
                                dst.at<cv::Vec3b>(j, i)[k] = (1.0 - a) * (1.0 - b) *
src.at<cv::Vec3b>(y[0], x[0])[k]
                                        + a * (1.0 - b) * src.at<cv::Vec3b>(y[0], x[1])[k]
                                        + (1.0 - a) * b * src.at<cv::Vec3b>(y[1], x[0])[k]
                                        + a * b * src.at<cv::Vec3b>(y[1], x[1])[k];
                        }
                }
        }
}
```

3.Result

< 원본 image >



▶scale_rate = 0.5

▶scale_rate = 1.5

4.Discussion

- scale_rate가 1보다 큰 경우에 기존의 이미지보다 화질이 깨지는 느낌이 제법 강하다. Bicubic interpolation등 더 좋은 기법을 사용하면 이미지 변질을 최소화 할 수 있을 것이다.