

## NA HW#8 Resampling of image

2013012148 이재일

### Bilinear Interpolation

```
void bilinear_interpolation(Mat &img1, Mat &img2, float xrate, float yrate)
{
    for (int y = 0; y < img2.rows; y++)
    {
        for (int x = 0; x < img2.cols; x++)
        {
            int px = (int)(x / xrate);
            int py = (int)(y / yrate);

            double fx1 = (double)x / (double)xrate - (double)px;
            double fx2 = 1 - fx1;
            double fy1 = (double)y / (double)yrate - (double)py;
            double fy2 = 1 - fy1;

            double w1 = fx2*fy2;
            double w2 = fx1*fy2;
            double w3 = fx2*fy1;
            double w4 = fx1*fy1;
        }
    }
}
```

px에 x값에 확대 배율을 나눈 값을 int로 형 변환 한 value를 assign한다. py도 같은 작업을 해준다. 그 후 fx1에 x값에 확대 배율의 나눈 값에 px를 뺀 value를 assign한다. fy1도 같은 작업을 해준다(x,y가 기존 image에 대응되는 위치와 px, py와의 거리). fx2와 fy2는 1에 fx1,fy1를 각각 뺀 값을 assign한다(px+1, py+1과의 거리).

```
    if (img1.channels() == 1)
    {
        uchar P1 = img1.at<uchar>(py, px);
        uchar P2 = img1.at<uchar>(py, px+1);
        uchar P3 = img1.at<uchar>(py+1, px);
        uchar P4 = img1.at<uchar>(py+1, px+1);
        img2.at<uchar>(y, x) = w1*P1 + w2*P2 + w3*P3 + w4*P4;
    }
    else if (img1.channels() == 3)
    {
        Vec3b P1 = img1.at<Vec3b>(py, px);
        Vec3b P2 = img1.at<Vec3b>(py, px+1);
        Vec3b P3 = img1.at<Vec3b>(py+1, px);
        Vec3b P4 = img1.at<Vec3b>(py+1, px+1);
        img2.at<Vec3b>(y, x) = w1*P1 + w2*P2 + w3*P3 + w4*P4;
    }
}
```

P1~P4에 각 좌표 pixel의 RGB값을 저장시킨다(칼라일 경우, channel=3). 그 후 새로운 image의 y,x 좌표에 각 가중치 w1~w4를 각각 P1~P4로 곱한 값의 value를 집어넣는다.

## Main

c++ skeleton을 사용하였다.

```
51 int main(int argc, char *argv[])
52 {
53     // jpg, png, bmp과 같은 이미지 관련 확장자들은 모두 사용할 수 있습니다.
54     cv::Mat image = cv::imread("lena.png", CV_LOAD_IMAGE_COLOR);
55     if (image.empty())
56     {
57         printf("No Image Found\n");
58         return -1;
59     }
60     namedWindow("Original Image", WINDOW_AUTOSIZE);
61     cv::imshow("Original image", image);
62
63     int h = image.rows;
64     int w = image.cols;
65
66     int imgstate = (image.channels() == 1) ? CV_8UC1 : CV_8UC3;
67     Mat newimage(h, w, imgstate, Scalar(0));
68
69     float n, m;
70
71     std::cout << "input x scale and y scale" << std::endl;
72     std::cin >> m;
73     std::cin >> n;
74
75     bilinear_interpolation(image, newimage, m, n);
76
```

```
cv::imshow("new image", newimage);
// cv::waitKey()는 imshow를 위해서 쓴다
//안그러면 image를 그리고 보여주기 전
cv::waitKey();
// cv::waitKey(); --> 아무 파라미터도
// cv::waitKey(mills); --> 넣은 숫자
destroyAllWindows();
```

## Execution Result

