Digital Image Processing - Exercise 5

Group C

Ahmad Alabaji
Erik Dietz
Eunbi Park
Timur Pratama Wiradarma

[1] Practice

1) Gradient in x- and y direction

```
// creates kernel representing fst derivative of a Gaussian kernel in x-direction
/*
sigma   standard deviation of the Gaussian kernel
return  the calculated kernel
*/
Mat Dip5::createFstDevKernel(double sigma){
    int kSize = round(sigma * 3) * 2 - 1;

    Mat gaussKernel = getGaussianKernel(kSize, sigma, CV_32FC1);
    gaussKernel = gaussKernel * gaussKernel.t();

    Mat Gx = Mat::zeros(kSize, kSize, CV_32FC1);

    for (int i = 0; i < kSize; i++){
        for (int j = 0; j < kSize; j++){
            Gx.at<float>(i, j) = (-(i-1) / (sigma*sigma))*gaussKernel.at<float>(i, j);
        }
    }
    return Gx;

}
```

```
Mat FDKernel = createFstDevKernel(sigma);
Mat resultsImg;
Mat Gx, Gy;

//Calculate directional gradients
filter2D(img, Gx, -1, FDKernel);
//imshow("resultsX", Gx);
//imwrite("resultsX.png", Gx);

filter2D(img, Gy, -1, FDKernel.t());
//imshow("resultsY", Gy);
//imwrite("resultsY.png", Gy);
```

2) Average (Gaussian Window)
3) Trace of structure tensor

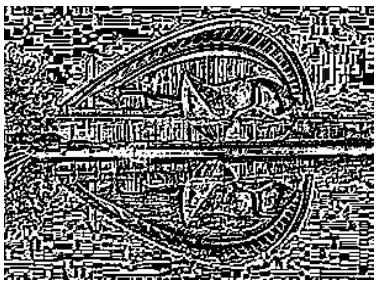
```
Mat Gxx = Gx.mul(Gx);
Mat Gyy = Gy.mul(Gy);
Mat Gxy = Gx.mul(Gy);

//Average (Gaussian Window)
GaussianBlur(Gxy, Gxy, cv::Size(3,3), 1);
//imshow("resultsxy", Gxy);
//imwrite("resultsxy.png", Gxy);

//Trace of structure tensor
Mat trace;
add(Gxx, Gyy, trace);
//imshow("structure tensor", trace);
//imwrite("structure tensor.png", trace);
```

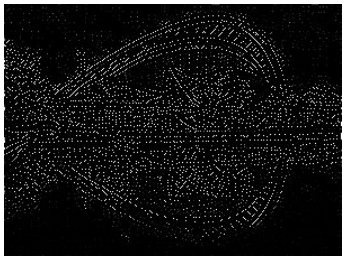
4) Determinant of structure tensor

```
//Determinant of structure tensor
Mat de1, de2, determinant;
de1 = Gxx.mul(Gyy);
de2 = Gxy.mul(Gxy);
determinant = de1 - de2;
//imshow("determinant", determinant);
//imwrite("determinant.png", determinant);
```



5) Weight calculation

```
//weight calculation
Mat weight = determinant / trace;
float meanW = mean(weight)[0];
weight = nonMaxSuppression(weight);
threshold(weight, weight, 0.5*meanW, 1, THRESH_TOZERO);
//imshow("weight", weight);
//imwrite("weight.png", weight);
```



6) Isotropy calculation

```
//isotropy calculation
Mat iso;
iso = 4 * determinant / trace.mul(trace);
iso = nonMaxSuppression(iso);
threshold(iso, iso, 0.5, 255, THRESH_TOZERO);
//imshow("iso", iso);
//imwrite("iso.png", iso);
```

7) Keypoints found