RBE 549: Homework Assignment 4

Aditya Mehrotra
amehrotra@wpi.edu

October 9, 2021

Problem 1

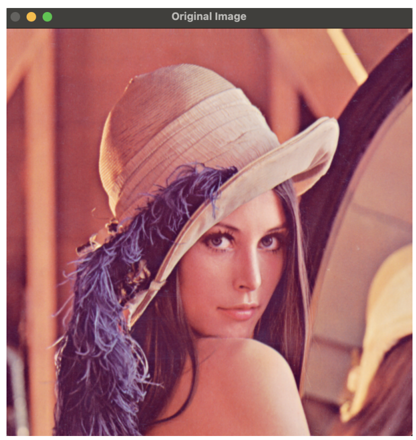


Figure 1: Original Image

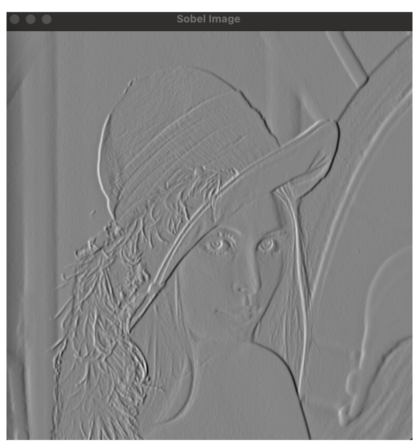


Figure 2: Sobel Edge Detection

a

```
#include "opencv2/core/core.hpp"
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include "iostream"

using namespace cv;
using namespace std;

int main(int argc, char const *argv[])
{
    Mat src;
    src = imread("../Data/lenna.png", IMREAD_COLOR);
    namedWindow( "Original Image", WINDOW_AUTOSIZE);
    imshow( "Original Image", src);

    Mat grey;
    cvtColor(src, grey, COLOR_BGR2GRAY);

    Mat sobelx;
    Sobel(grey, sobelx, CV_32F, 1, 0);

    double minVal, maxVal;
    minMaxLoc(sobelx, &minVal, &maxVal);
    cout << "minVal : " << minVal << endl << "maxVal : " << maxVal << endl;

    Mat draw;
    sobelx.convertTo(draw, CV_8U, 255.0/(maxVal - minVal),
            -minVal * 255.0/(maxVal - minVal));

    namedWindow("Sobel Image", WINDOW_AUTOSIZE);
    imshow("Sobel Image", draw);

    waitKey(0);
    return 0;
}
```

b As the σ value increases, the features of the image diminish, and in turn, the edges are lost. We get larger closed contours with higher values of σ.

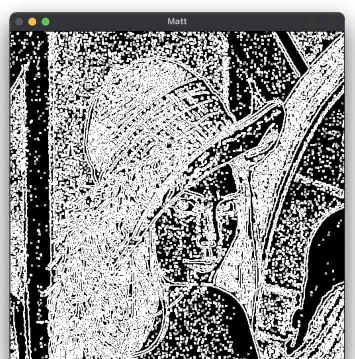
```
#include "opencv2/imgproc.hpp"
#include "opencv2/imgcodecs.hpp"
#include "opencv2/highgui.hpp"
#include <iostream>

using namespace cv;
using namespace std;

Mat MarrEdgeDetection(Mat src, int kernelDiameter, double sigma) {
    int kernel_size = kernelDiameter / 2;
    Mat kernel(kernelDiameter, kernelDiameter, CV_64FC1);
    for (int i = -kernel_size; i <= kernel_size; i++) {
        for (int j = -kernel_size; j <= kernel_size; j++) {
            kernel.at<double>(i + kernel_size, j + kernel_size) =
            exp(-((pow(j, 2) + pow(i, 2)) /
                (pow(sigma, 2) * 2)))
                * (((pow(j, 2) + pow(i, 2) - 2 *
                    pow(sigma, 2)) / (2 * pow(sigma, 4))));
        }
    }
    Mat laplacian(src.rows - kernel_size * 2, src.cols - kernel_size * 2, CV_64FC1);
    Mat dst = Mat::zeros(src.rows - kernel_size * 2,
    src.cols - kernel_size * 2, CV_8UC1);
    for (int i = kernel_size; i < src.rows - kernel_size; i++) {
        for (int j = kernel_size; j < src.cols - kernel_size; j++) {
            double sum = 0;
            for (int x = -kernel_size; x <= kernel_size; x++){
                for (int y = -kernel_size; y <= kernel_size; y++) {
                    sum +=  src.at<uchar>(i + x, j + y) * kernel.at<double>(x +
                    kernel_size, y + kernel_size);
                }
            }
            laplacian.at<double>(i - kernel_size, j - kernel_size) = sum;
        }
    }
    for (int i = 1; i < dst.rows - 1; i++) {
```

```
        for (int j = 1; j < dst.cols - 1; j++) {
            if ((laplacian.at<double>(i - 1, j) *
            laplacian.at<double>(i + 1, j) < 0) ||
            (laplacian.at<double>(i, j + 1) *
            laplacian.at<double>(i, j - 1) < 0) ||
                (laplacian.at<double>(i + 1, j - 1) *
                laplacian.at<double>(i - 1, j + 1) < 0) ||
                (laplacian.at<double>(i - 1, j - 1) *
                laplacian.at <double> (i + 1, j + 1) < 0)) {
                dst.at<uchar>(i, j) = 255;
            }
        }
    }
    return dst;
}

int main() {
    Mat src = imread("../Data/lenna.png", 0);
    Mat dst = MarrEdgeDetection(src, 6, 1);
    imshow("Matt", dst);
    waitKey(0);
    return 0;
}
```



(a) $\sigma = 1$

Figure 3: Marr-Hildreth Edge Detection computed on the original image

Problem 2

Let us define

$$F(w) = \int_{-\infty}^{\infty} f(t)e^{-iwt}dt$$

$$F(-w) = \int_{-\infty}^{\infty} f(t)e^{iwt}dt$$

Now let $t = -u$, and $dt = -du$. So,

$$\Rightarrow F(-w) = \int_{\infty}^{-\infty} f(-u)e^{-iwu}(-du)$$

$$\Rightarrow F(-w) = \int_{-\infty}^{\infty} f(-u)e^{-iwu}du$$

So, $f(-t) = \pm f(t)$ implies $F(-w) = \pm F(w)$

Recall that $e^{iwt} = coswt + isinwt$. Also, sine is an odd function whereas cosine is an even function. We know from elementary calculus that the integral of an odd function on $(-\infty, \infty)$ vanishes.

Thus, when f is an odd and purely imaginary, then $f(t)cos(wt)$ is odd and imaginary $-f(t)isin(wt)$ is odd and purely real. It follows that the imaginary part of Fourier transform vanishes. Consequently, $F(w)$ is odd and real.

$$F(w) = \int_{-\infty}^{\infty} f(t)e^{-iwt}dt$$

$$= \int_{-\infty}^{\infty} f(t)[cos(wt) - isin(wt)]dt$$

$$= \int_{-\infty}^{\infty} f(t)[cos(wt)]dt - \int_{-\infty}^{\infty} f(t)isin(wt)]dt$$

$$= -i \int_{-\infty}^{\infty} f(t)sin(wt)dt$$

$$= -2i \int_{0}^{\infty} f(t)sin(wt)dt$$

Problem 3

We know that, $f(x) * g_{\sigma_1}(x) = h_1(x)$. And similarly, $f(x) * g_{\sigma_2}(x) = h_2(x)$

a Since convolution is distributive,

$$h_3(x) = \frac{h_2(x) - h_1(x)}{\sigma_2 - \sigma_1}$$

$$\Rightarrow \frac{f(x) * g_{\sigma_2}(x) - f(x) * g_{\sigma_1}(x)}{\sigma_2 - \sigma_1}$$

$$\Rightarrow f(x) * \left(\frac{g_{\sigma_2}(x) - g_{\sigma_1}(x)}{\sigma_2 - \sigma_1} \right)$$

$$\Rightarrow h_3(x) = f(x) * g(x))$$

Now considering just $g(x)$,

$$= \left(\frac{g_{\sigma_2}(x) - g_{\sigma_1}(x)}{\sigma_2 - \sigma_1} \right)$$

$$\Rightarrow \frac{1}{\sigma_2 - \sigma_1} \left(\frac{1}{2\pi{\sigma_2}^2} e^{\frac{-|x|^2}{2{\sigma_2}^2}} - \frac{1}{2\pi{\sigma_1}^2} e^{\frac{-|x|^2}{2{\sigma_1}^2}} \right)$$

From the above, it is observed that $g(x)$ is the difference of gaussians (DoG) operator.

b If $F(x) = 1$ we will get the diagram in 1D as below. The difference of gaussian (DoG) filter is equivalent to the bandpass filter.

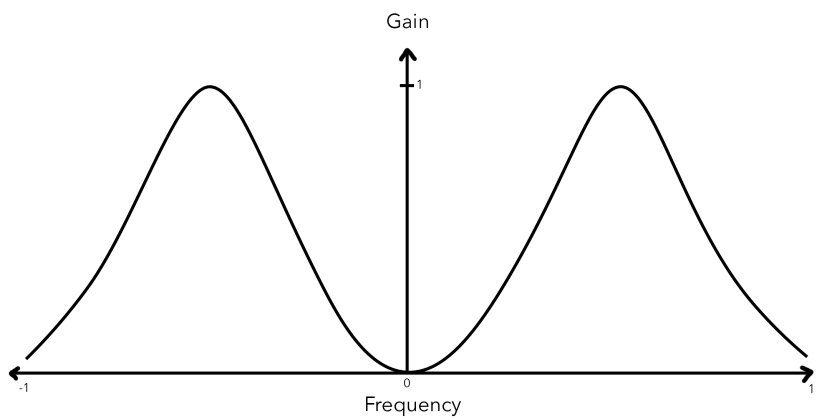


Figure 4: Bandpass Filter

c Now if, $g_{\sigma_1}(x) = \frac{1}{2\pi{\sigma_1}^2} e^{\frac{-|x|^2}{2{\sigma_1}^2}}$, the Fourier transform can be calculated analytically as:

$$G_1(u) = \int_{-\infty}^{\infty} g(x) e^{i2\pi ux} dx$$

$$= \frac{1}{2\pi{\sigma_1}^2} \int_{-\infty}^{\infty} e^{\frac{-|x|^2}{2{\sigma_1}^2}} e^{i2\pi ux} dx$$

$$= \frac{1}{2\pi{\sigma_1}^2} \int_{-\infty}^{\infty} e^{\frac{-[|x|^2 - 4i\pi{\sigma_1}^2 ux]}{2{\sigma_1}^2}} dx$$

$$= \frac{1}{2\pi{\sigma_1}^2} \int_{-\infty}^{\infty} e^{\frac{-[|x|^2 - 4i\pi ux + (2i\pi{\sigma_1}^2 u)^2 - (2i\pi{\sigma_1}^2 u)^2]}{2{\sigma_1}^2}} dx$$

$$= \frac{1}{2\pi{\sigma_1}^2} \left(\int_{-\infty}^{\infty} e^{-\frac{([|x| - 2i\pi{\sigma_1}^2 u)^2}{2{\sigma_1}^2}} dx \right) e^{-2\pi^2{\sigma_1}^2}$$

$$= \frac{1}{2\pi{\sigma_1}^2} . \sqrt{(2\pi{\sigma_1}^2)} . e^{-2\pi^2{\sigma_1}^2 u^2}$$

$$G_1(u) = \frac{e^{-2\pi^2{\sigma_1}^2 u^2}}{\sqrt{(2\pi{\sigma_1}^2)}}$$

Similarly, $G_2(u) = \frac{e^{-2\pi^2{\sigma_2}^2 u^2}}{\sqrt{(2\pi{\sigma_2}^2)}}$
So,

$$G(u) = \frac{G_2(u) - G_1(u)}{\sigma_2 - \sigma_1}$$

$$G(u) = \frac{\frac{e^{-2\pi^2{\sigma_2}^2 u^2}}{\sqrt{(2\pi{\sigma_2}^2)}} - \frac{e^{-2\pi^2{\sigma_1}^2 u^2}}{\sqrt{(2\pi{\sigma_1}^2)}}}{\sigma_2 - \sigma_1}$$

d If $\sigma_2 \to \sigma_1$ then, $g(x) \to 0/0$. So, we apply L'Hôpital's rule.

$$\frac{dG_1(u)}{d\sigma_1} = \frac{d(e^{-2\pi^2{\sigma_1}^2 u^2})/d\sigma_1}{d(\sigma_2 - \sigma_1)/d\sigma_1}$$

$$\Rightarrow \frac{d(e^{-2\pi^2{\sigma_1}^2 u^2})/d\sigma_1}{1}$$

$$\Rightarrow -4\pi^2 u^2 \sigma_1 e^{-2\pi^2{\sigma_1}^2 u^2}$$

We only consider 1 term, i.e. $G_1(u)$ as the difference would result in zero. Hence, this is a good edge detector. Here, a scalar is multiplied with the Fourier Transform of the gaussian function and the zero crossing will occur at the edges.

Problem 4

a The NW Operator is given as $\begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}$, while the NE Operator is $\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix}$

Similarly, the Sobel H Operator is $\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$ and the Sobel V Operator is $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$

So, [H + V] can be given as, $\begin{bmatrix} 0 & 2 & 2 \\ -2 & 0 & 2 \\ -2 & -2 & 0 \end{bmatrix}$

Similarly, [H - V] can be given as, $\begin{bmatrix} 2 & 2 & 0 \\ 2 & 0 & -2 \\ 0 & -2 & -2 \end{bmatrix}$

Clearly, NW Operator = $\frac{1}{2}$ [H + V] and the NE Operator = $\frac{1}{2}$ [H - V].

b Two ways in which the NW and NE Operators can be combined into a single measure are:

1. $\sqrt{(NE * f)^2 + (NW * f)^2}$
   (a) Advantage: High accuracy and sharpness
   (b) Disadvantage: Takes longer to compute
2. $|(NE * f) - (NW * f)| + |(NE * f) + (NW * f)|$
   (a) Advantage: Takes shorter to compute
   (b) Disadvantage: Low accuracy and is an approximation

c The NW Operator can be expressed as the convolution of the following operators:
A = $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ and B = $\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$, i.e. $A * B = NW$

Problem 5

a $V * B$ can be given by $\begin{bmatrix} -2 & -3 & 0 & 3 & 2 \\ -7 & -10 & 0 & 10 & 7 \\ -10 & -14 & 0 & 14 & 10 \\ -7 & -10 & 0 & 10 & 7 \\ -2 & -3 & 0 & 3 & 2 \end{bmatrix}$

b Applying B first then V produces the same results since convolutions is commutative.

c The filter is non-separable since it is orientation selective.