**LAB 02**

Digital Image & Video Processing – 19TGMT



**Teacher:**

* PhD. Lý Quốc Ngọc
* MS. Phạm Minh Hoàng
* MS. Phạm Thanh Tùng

**Student:**

* Chung Kim Khánh (19127644)

**Table of content**

[I. Information 2](#_Toc90996391)

[II. Solution 2](#_Toc90996392)

[III. User guide 8](#_Toc90996393)

[IV. Reference 12](#_Toc90996394)

# Information

|  |  |  |
| --- | --- | --- |
| student id | fullname | MAil |
| 19127644 | Chung Kim Khánh | ckkhanh19@clc.fitus.edu.vn |

# Solution

## Filter

Image filtering is used to:

* Remove noise
* Sharpen contrast
* Highlight contours
* Detect edges
* Other uses?

Image filters can be classified as linear or nonlinear.

Linear filters are also know as convolution filters as they can be represented using a matrix multiplication.

Thresholding and image equalisation are examples of nonlinear operations, as is the median filter.

### Average filtering

Average (or mean) filtering is a method of “smoothing” images by reducing the amount of intensity variation between neighbouring pixels.

The average filter works by moving through the image pixel by pixel, replacing each value with the average value of neighbouring pixels, including itself.

There are some potential problems:

* A single pixel with a very unrepresentative value can significantly affect the average value of all the pixels in its neighbourhood.
* When the filter neighbourhood straddles an edge, the filter will interpolate new values for pixels on the edge and so will blur that edge. This may be a problem if sharp edges are required in the output.

Consider the following 3 by 3 average filter:

We can write it mathematically as:

Normalizing is important because it keeps the image pixel values between 0 abd 255.

2D Average filtering example using a 3 x 3 sampling window:

Table

Description automatically generatedTable

Description automatically generatedA picture containing diagram

Description automatically generated

### Median filtering

Median filtering is a nonlinear method used to remove noise from images.

It is widely used as it is very effective at removing noise while preserving edges.

It is particularly effective at removing ‘salt and pepper’ type noise.

The median filter works by moving through the image pixel by pixel, replacing each value with the median value of neighbouring pixels.

The pattern of neighbours is called the "window", which slides, pixel by pixel, over the entire image.

The median is calculated by first sorting all the pixel values from the window into numerical order, and then replacing the pixel being considered with the middle (median) pixel value.

2D Median filtering example using a 3x3 sampling window:

Table

Description automatically generated with medium confidence

A picture containing chart

Description automatically generatedA picture containing table

Description automatically generated

### Gauss filtering

Gaussian filtering is used to blur images and remove noise and detail.

In one dimension, the Gaussian function is:

Where is the standard deviation of the distribution. The distribution is assumed to have a mean of 0.

Shown graphically, we see the familiar bell shaped Gaussian distribution.

Significant values:

A picture containing text

Description automatically generated

Gaussian kernel coefficients are sampled from the 2D Gaussian function.

Where σ is the standard deviation of the distribution.

The distribution is assumed to have a mean of zero.

We need to discretize the continuous Gaussian functions to store it as discrete pixels.

## Detect

### Sobel operator

The Sobel operator performs a 2-D spatial gradient measurement on an image and so emphasizes regions of high spatial frequency that correspond to edges. Typically it is used to find the approximate absolute gradient magnitude at each point in an input grayscale image.

In theory at least, the operator consists of a pair of 3×3 convolution kernels as shown below. One kernel is simply the other rotated by 90°. This is very similar to the Roberts Cross operator.

Table

Description automatically generated

These kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (call these Gx and Gy). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by:

Typically, an approximate magnitude is computed using: which is much faster to compute.

The angle of orientation of the edge (relative to the pixel grid) giving rise to the spatial gradient is given by:

In this case, orientation 0 is taken to mean that the direction of maximum contrast from black to white runs from left to right on the image, and other angles are measured anti-clockwise from this.

Often, this absolute magnitude is the only output the user sees --- the two components of the gradient are conveniently computed and added in a single pass over the input image using the pseudo-convolution operator shown below:

A picture containing text, clock

Description automatically generated

Using this kernel the approximate magnitude is given by:

### Prewitt operator

The Prewitt operator was developed by Judith M. S. Prewitt. Prewitt operator is used for edge detection in an image. Prewitt operator detects both types of edges, these are:

* Horizontal edges or along the x-axis.
* Vertical Edges or along the y-axis.

Wherever there is a sudden change in pixel intensities, an edge is detected by the mask. Since the edge is defined as the change in pixel intensities, it can be calculated by using differentiation. Prewitt mask is a first-order derivate mask. In the graph representation of Prewitt-mask’s result, the edge is represented by the local maxima or local minima.

* Both the first and second derivative masks follow these three properties:
* More weight means more edge detection.
* The opposite sign should be present in the mask. (+ and -)
* The Sum of the mask values must be equal to zero.

Prewitt operator provides us two masks one for detecting edges in the horizontal direction and another for detecting edges in a vertical direction.

Steps:

* Read the image.
* Convert into grayscale if it is colored.
* Convert into the double format.
* Define the mask or filter.
* Detect the edges along X-axis.
* Detect the edges along Y-axis.
* Combine the edges detected along the X and Y axes.
* Display all the images.

### Laplace operator

Steps:

* We will define a function for the filter
* Then we will make the mask
* Then we will define function to iterate that filter over the image(mask)
* We will make a function for checking the zeros as explained
* And finally a function to bind all this together

One thing to keep in mind is that to create a combination of the filters, i.e. Laplacian over gaussian filter (LoG), then we can use the following formula to combine both of them.

# User guide

## Filter

Source image:

A person wearing a hat

Description automatically generated with low confidence

With

### Average filtering

A person wearing a hat

Description automatically generated with medium confidence

### Median filtering

A person wearing a hat

Description automatically generated with medium confidence

### Gaussian filtering

A person wearing a hat

Description automatically generated with medium confidence

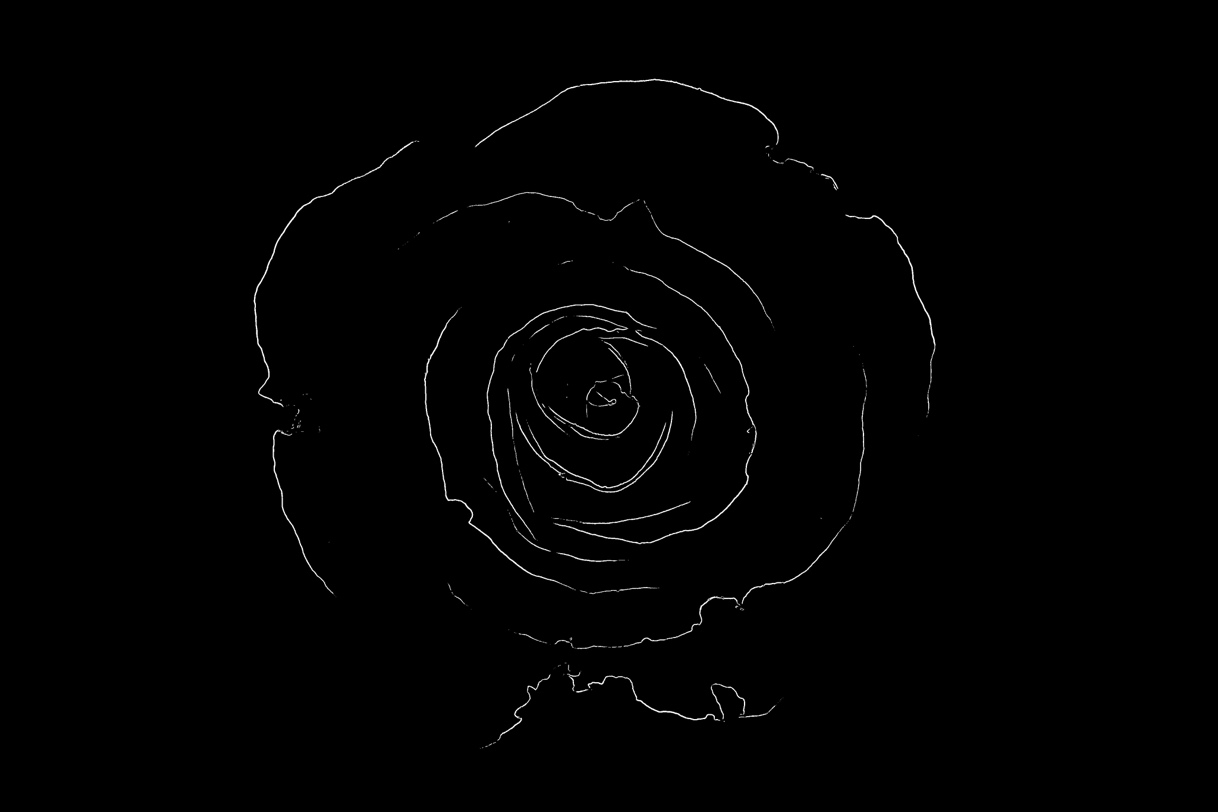
## Detect

Source image:

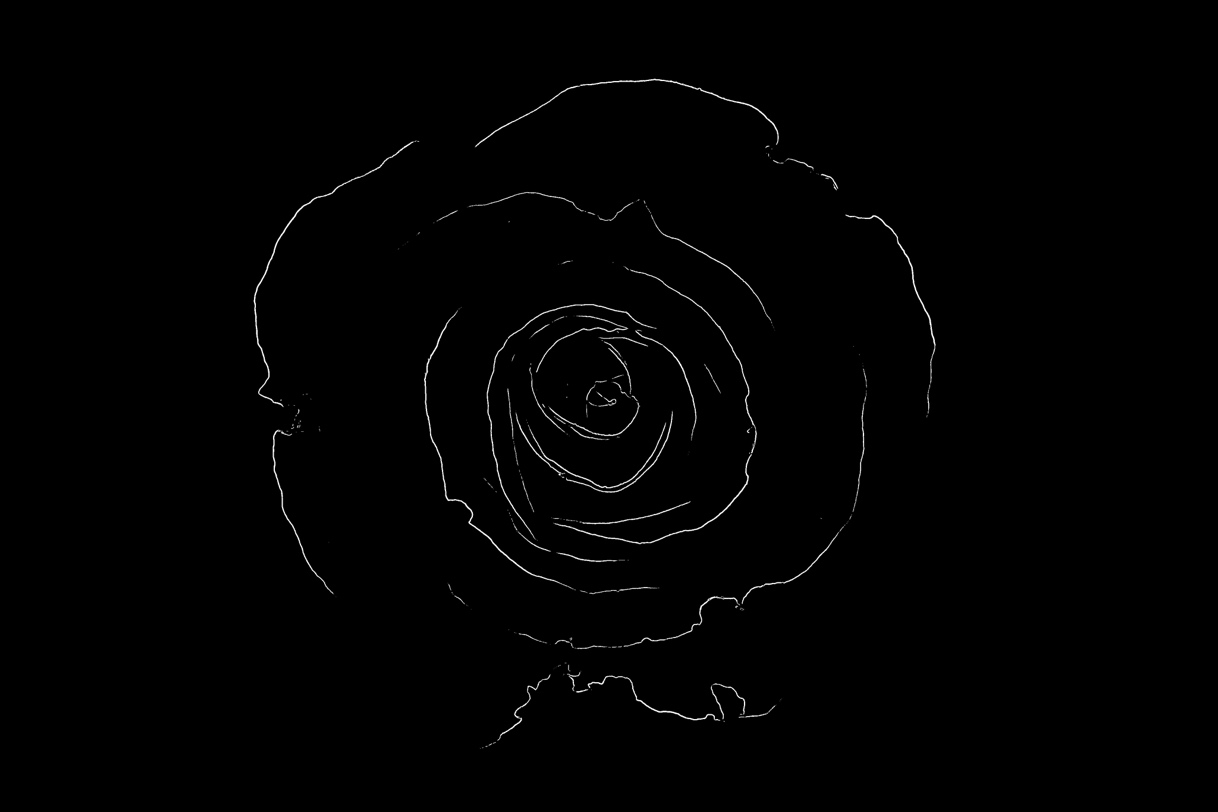
A close up of a rose

Description automatically generated with medium confidence

### Sobel operator



### Prewitt operator



### Laplace operator

A picture containing plant

Description automatically generated

# Reference

1. The University of Auckland New Zealand
2. https://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm#:~:text=The%20Sobel%20operator%20performs%20a,in%20an%20input%20grayscale%20image.
3. GeeksforGeeks