ECE7410/ENGI 9804
Image Processing and Applications

Laboratory 2

Spatial Filtering and Thresholding

Introduction

The objective of this exercise is to familiarize you with thresholoding and filtering in the spatial domain.

Spatial Filtering

Spatial domain filtering operates directly on the pixel intensities. In general spatial filters are define as a neighbourhood region centred at the working pixel in an image, and they operate on a portion of the image that is the same size as the filter neighbourhood. The filter is typically called a *kernel, mask, template* or *window* and the values (coefficients) in the filter determine the function of the filter. The filtering process is carried out by sliding the filter mask from point to point in an image; every point in the image is filtered. At each point (x, y), the output of the filter is calculated using a predefined relationship.

Spatial filters can be divided in to two main categories: *linear filters* and *non-linear filters*. The output of linear filters at a point (x, y) is the sum of products of the filter coefficients and the corresponding image pixels in the neighbourhood area. In general the width and height of the mask are odd numbers to ensure symmetry. Spatial filters have different functions (e.g., blurring, sharpening) depending on the coefficients of the kernel being used. Common examples are averaging filters, the sobel operators, Laplacian, Gaussian, etc.

The general expression for a linear correlation filter at location (x, y)can be expressed as:

$$g(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x + s, y + t)$$

where g is the output of the filter at (x, y), f is the original image, w is the filter mask and the size of the mask is $(2a + 1, 2b + 1)$.

Non-linear filters, also known as order-statistics filters, have a response based on the ordering of the pixels contained in the image area encompassed by the filter. Some examples of non-linear filters are median filter, max filter, min filter, etc.

Thresholding

Thresholding is the simplest method of image segmentation. When used with grayscale images the result of thresholding is a binary image (two intensities, (0) and (1)). Thegeneral expression for thresholding can be expressed as:

$$g(x,y) = \begin{cases} 1 : f(x,y) \geqslant T \\ 0 : f(x,y) \leqslant T \end{cases}$$

where T is the cuttoff threshold intensity.

Even though the thresholding process is relatively simple to implement, determining the correct threshold value is difficult and in many cases requires manual user specification. Otsu's method is a widely used method to determine the appropriate threshold value of an image.

Otsu's Method

This method assumes that the image contain's two classes (object and background) and determines most suited intensity to separate the two classes. Otsu shows that the most suited threshold will maximize the inter-class variance. Therefore we need to find the intensity level that maximizes the inter-class variance. In order to calculate inter-class variance we define the class probability and class mean as follows.

$$\begin{aligned} \omega_0(t) &= \sum_{i=0}^{t-1} p(i) \\ \omega_1(t) &= \sum_{i=t}^{L-1} p(i) \\ \mu_0(t) &= \sum_{i=0}^{t-1} ip(i)/\omega_0 \\ \mu_1(t) &= \sum_{i=t}^{L-1} ip(i)/\omega_1 \end{aligned} \tag{1}$$

where ω_0, ω_1 are class probabilities, μ_0, μ_1 are class means, t threshold value, $p(i)$ is the probability of the intensity i, and L are number of gray levels. The inter-class variance is defined as:

$$\sigma_b^2(t) = \omega_0\omega_1[\mu_0 - \mu_1]^2 \tag{2}$$

The Otsu's algorithm is as follows

1. Compute the histogram and determine the probabilities of each intensity level;

2. Initialize $\omega_0(0), \omega_1(0)$ and $\mu_0(0), \mu_1(0)$;
3. Step through all possible thresholds $t = 1\ to\ L-1$, update $\omega_0(t), \omega_1(t), \mu_0(t), \mu_1(t)$ and $\sigma_b^2(t)$.
4. The `optimal` threshold corresponds to the maximum value of $\sigma_b^2(t)$.

Procedure

Spatial Filtering

1. Download the test images (img1.png and img2.png) from Brightspace under *Lab 2* and save them in your working directory.
2. Read img1.png and convert it to a grayscale image. Develop a function that will perform the spatial filtering operations below. The function should take an image and the kernel as inputs and return the filtered image. Do **NOT** use internal functions for this (e.g., MATLAB's **imfilter** function). (**hint:** you might need to add zero padding around the image).

   i Kernel_1 = (1/9)*ones(3)

   ii Kernel_2 = (1/49)*ones(7)

   iii Kernel_3 = fspecial('average',[7,7]);

   iv Kernel_4 = fspecial('gaussian',[3,3],0.5);

   v Kernel_5 = fspecial('gaussian',[7,7],1.2);

3. Develop a function that performs median filtering. A median filter will replace the pixel intensity with the median intensity of the pixels overlapped with the kernel. The function should take the image and the size of the kernel as inputs. Apply the following median filters to image img1.png:

   i Median filter with kernel size 3×3.

   ii median filter with kernel size 5×5

4. The following kernels implement linear sharpening filters which are used to enhance the edges in images. Use the function developed in question **2** and apply the filters to image img2.png.

i Kernel_6 = $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$

ii Kernel_7 = $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$

iii Kernel_8 = fspecial('log',3);

Thresholding

1. Download the test images (img3.png and img4.png) from Brightspace under *Lab 2* and save them in your working directory.
2. Implement equations 1 and 2 as a function to calculate the threshold value for any given image.
3. Calculate and report the threshold values for im3.png and im4.png found using your code in question **2**.
4. Develop code to convert a grayscale image to a binary image using a given threshold value. Do **NOT** use internal functions for this (e.g., the MATLAB function **im2bw**).
5. Use the threshold values found in question **3** and generate binary image for img3.png and img4.png.

Discussion

1. Compare the smoothing effects obtained by Kernel_1 to Kernel_5 and the median filters.
2. Discuss the effect the kernel size has on the output images in your experiments.
3. Compare the outputs obtained by Kernel_6 to Kernel_8.
4. Discuss the effect noise has on edge detection.
5. Global thresholding can fail if the lighting conditions in an image are not uniform. Review adaptive thresholding ad explain how it can address uneven lighting conditions.