CS512 Assignment 2

Lin Zhuo

A20379609

1. Noise and Filtering

(a) Explain how to estimate the signal to noise ratio in an image.

It is basically measured as the energy of the signal divided by the energy of noise. It can be written as the following formula:

$SNR = \frac{E_s}{E_n} = \frac{{\sigma_s}^2}{{\sigma_n}^2}$

where ${\sigma_s}^2$ represents the variance of pixels in a sequence of images and ${\sigma_n}^2$ has the meaning of the variance in uniform area.

When measured in db, the formula is:

$SNR[db] = 10\, log \frac{E_s}{E_n}$

(b) Explain the difference between Gaussian and impulsive noise. Which filter handles better impulsive noise: an averaging filter or a median filter?

Gaussian noise is statistical noise with a probability density function equal to that of the normal distribution (Gaussian distribution) while impulsive noise is abruptly happened, accidentally introduced to the signal, and it is normally sharp and short.

A median filter handles better the impulsive noise.

(c) Give an image having the value of 2 in each cell, write the value of the pixels in this image after applying a 3x3 convolution filter having all 1-s in its entries.

For example we have a 5-by-5 matrix having the value 2 in each cell:

$$\begin{bmatrix} 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 \end{bmatrix}$$

By the end of applying the 3-by-3 convolution filter having all 1-s, it becomes:

$$\begin{bmatrix} 2 & 2 & 2 & 2 & 2 \\ 2 & 18 & 18 & 18 & 2 \\ 2 & 18 & 18 & 18 & 2 \\ 2 & 18 & 18 & 18 & 2 \\ 2 & 2 & 2 & 2 & 2 \end{bmatrix}$$

The image having the other dimensiona are similar: the outmost edge of the image remain the same while all the pixels inside are multiplied by 9 times.

(d) Given that we need the derivative of an image convolved with a filter, explain how the operation can be applied more efficiently.

Compute the derivative of the filter first and then convolve with image.

(e) Explain the three different ways to handle boundaries during convolution.

- Expand - the nearest pixels on the border are extended to provide values for the convolution. Corner pixels are extended in 90° wedges. Other edge pixels are extended in lines.
- Wrap - The image is conceptually wrapped such that the values can be taken from the opposite edge or corner.
- Crop - Any pixel in the output image which would require values from beyond the edge is skipped. This method can result in the output image being slightly smaller, with the edges having been cropped.

(f) Write a basic 3 x 3 smoothing filter. What is the sum of all entries in this filter? Explain the reason for the sum to be selected as it is.

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

The sum of all entries in this filter is 1, because we want to maintain the average intensity.

(g) Explain how to implement a 2D convolution with a Gaussian using two 1D convolution filters. Which option is more efficient? Is it possible to implement any 2D filter in this way?

The convolution with 2D Gaussian filter $G(x, y)$ can be decomposed to applying a 1D filter along the x direction and then another 1D filter along y direction, such as:

$I * G(x, y) = G(y) * (I * G(x))$

The Gaussian convolution with two 1D filters is more efficient.

It is possible to do in this way with Gaussian filter because Gaussian filter is seperable, which is:

$G(x, y) = exp(-\frac{x^2+y^2}{2\sigma^2}) = exp(-\frac{x^2}{2\sigma^2}) * exp(-\frac{y^2}{2\sigma^2})$

but for some other filters that are not able to seperate x and y components, it is not possible to implement in this way.

(h) Given a 1D Gaussian Filter with $\sigma = 2$, what should be the size of this filter?

The filter size is generally 3 times the standard deviation, that is, 6.

(i) Explain how a Gaussian image pyramid is produced. What is the reason for producing such pyramids? What is the amount of additional processing done in a pyramid compared with a single image?

When the most proper window size cannot be determined, the multiple scale analysis of image is needed with the same window size but with different scales of images.

The Gaussian pyramid is produced in the way that every upper level of image has half of the height and width of the lower level. The upper bound of the number of layers is the logarithmic of the image size.

The processing cost is: $m^2 + (\frac{1}{2}m)^2 + (\frac{1}{4}m)^2 + (\frac{1}{8}m)^2 + ... <= \frac{4m^2}{3}$ where m is the size of image.

One third of additional amount of processing is needed.

(j) Explain how the Laplacian pyramid is produced and its use.

A Laplacian pyramid is similar to a Gaussian pyramid but saves the difference image of the blurred versions between each levels. Only the smallest level is not a difference image to enable reconstruction of the high resolution image using the difference images on higher levels.

The Laplacian pyramid is useful image compression.

2. Edge Detection

(a) Why is edge detection useful? What are the desired properties of edge detection?

Edge detection can provide useful local features besides the global features of an image.

The desired properties are: - Correspond to scene elements. - Invariant. - Reliable and consistent in detection.

(b) Explain the basic steps of edge detection and the need for them: smoothing, enhancement and localization.

Capture the discontinuities in the image, it means to detect: - discontinuities of depth - discontinuities of illumination - discontinuities of texture - discontinuities of normal - noise - ...

In the end, find the derivative.

- Because derivatives are sensitive to noise, smoothing is needed in order to filter out the noise but not affect the edges.
- Enhancement makes the edges more pronounced for detection.
- Localization can give information where the edges are.

(c) Describe two filters for computing the image gradient. What is the meaning of image gradient? What is it used for?

The image gradient can be computed by applying the filter for x-direction and for y-direction, which are:

$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$ and $\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$, respectively.

Image grdient is the direction in which the image pixel has maximum change in colors. It can be used for edge detection becuase on the edge we can detect the significant image gradient.

(d) Explain how the Sobel filter can be produced from a smoothing and derivative filters.

For example, when compute the image gradient in x-direction,

$I * G * \frac{\partial}{\partial x}$

we can compute the convolution of the smoothing filter G and the derivative filter $\frac{\partial}{\partial x}$ first:

$I * G * \frac{\partial}{\partial x} = I * (G * \frac{\partial}{\partial x})$

$G * \frac{\partial}{\partial x} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} * \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$

(e) Explain how to generate a more accurate derivative filter with an arbitary σ. Write the elements of a filter for more accurate derivative computation with $\sigma = 2$.

Reconstruct the original image (continuous function) from the sample image and get the derivative of this continuous function, then resample it.

Original image $f[x]$ -> Reconstruct: $f[x] * h(x)$ -> Compute derivative: $f[x] * h'(x)$ -> Resample: $f[x] * h'[x]$

We could use Gaussian filter to approximate the sinc function, thus the more accurate derivative becomes $f[x] * G'[x]$.

$I_x = I * G_y * G'_x$

$I_y = I * G_x * G'_y$

$G_x = exp(\frac{-x^2}{2\sigma^2}) => G'_x = \frac{-x}{\sigma^2} exp(\frac{-x^2}{2\sigma^2})$

With $\sigma = 2$, we have $G'_x = \frac{-x}{4} exp(\frac{-x^2}{8})$

Similarly, $G'_y = \frac{-y}{4} exp(\frac{-y^2}{8})$

(f) Explain how an edge can be localized using the first or second order derivative of the image.

By computing the first derivative of the image function, we can detect the edge where the first derivative is larger than a specific threshold τ.

While by computing the second derivative, the edge can be localized at zero-crossing of second derivative.

(g) Let $\sigma = 1$. Write the Laplacian of Gaussian (LOG) filter using this σ. Explain how to use LOG to detect edges.

- The Lapacian operator:

$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$

$LoG(x, y) = \frac{\partial^2 G}{\partial x^2} + \frac{\partial^2 G}{\partial y^2},\ where\ G(x, y) = \frac{1}{2\pi\sigma^2 exp(-\frac{x^2+y^2}{2\sigma^2})}$

$LoG(x, y) = -\frac{1}{\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} (1 - \frac{x^2+y^2}{2\sigma^2})$

Let $\sigma = 1$,

$LoG(x, y) = -\frac{1}{\pi} e^{-\frac{x^2+y^2}{2}} (1 - \frac{x^2+y^2}{2})$

- The procedure of using LOG to detect edges:

- Compute the LOG filter and convolve the image with LOG. ($\Delta(I * G) = I * \Delta G$)

- Thresholding. Assign 1 to the pixels where $I * LoG > 0$ and 0 to those where $I * LoG \leq 0$

- Mark the edges at transitions (from 0 to 1 or from 1 to 0).

(h) Explain the main differences between the Canny edge detection algorithm and a standard edge detection that does not use directional derivatives. What is the condition for detecting an edge candidate in Canny?

The Canny edge detection first detects the direction of image gradient and compute the derivative in this direction, while the standard edge detection computes the derivatives in x and y directions and sums them up.

The condition for detecting an edge candidate is the gradient is larger than a certain threshold.

(i) Explain the non-maximum suppression and hysteresis thresholding parts of the Canny algorithm.

Non-maximum suppression: for each direction only keep the maximum and suppress the others in its neighbors.

Hysteresis: use two thresholds (a higher threshold τ_H and a lower one τ_L) in the edge detection. First start detecting using τ_H. Secondly, of the detected pixels, continue tracking using τ_L.