Lecture 04: Image Filtering

1 Image Filtering

Definition 1. The word **filter** comes from frequency-domain processing, where filtering refers to the process of accepting or rejecting specific frequency components.

In general, once can distinguish between low-pass and high-pass filtering:

- **Low-pass** filters smooth an image (retain low-frequency components).
- **High-pass** filters retain the edges of an image (retain high-frequency components).

1.1 Low-pass Filtering

As previously introduced, low-pass filters retain low-frequency components of an image, while blocking the high-frequency ones. As noise is considered an high-frequency signal, low-pass filters are typically used to reduce noise. Noise can occur in different forms:

- **Salt and pepper noise**: randum occurences of black and white pixels.
- **Impulse noise**: random occurences of white pixels.
- **Gaussian noise**: variations in intensity drawn from a Gaussian normal distribution. This can be written as
$$f(x, y) = \tilde{f}(x, y) + \eta(x, y), \tag{1.1}$$
where $f(x, y)$ represents the real observed image, $\tilde{f}(x, y)$ represents the ideal image without noise and $\eta(x, y) \sim \mathcal{N}(\mu, \sigma)$ the Gaussian (white) noise.

How can we reduce the noise to try to recover the ideal image?

1.1.1 Moving Average Filter

This filter replaces each pixel with an average of all the values in its neighborhood. The filter assumes that

- pixels are similar to their neighbors.
- noise process is independent from pixel to pixel.

One can notice the effect of the moving average filter on a 1D signal in Figure 1

1.1.2 Weighted Moving Average Filter

The principle is the same, but we can add weights to the moving average filter:

- Uniform weights.
- Non-uniform weights.

The idea to change the weights in the filter lays the basics for a bigger idea. This operation is called **convolution**

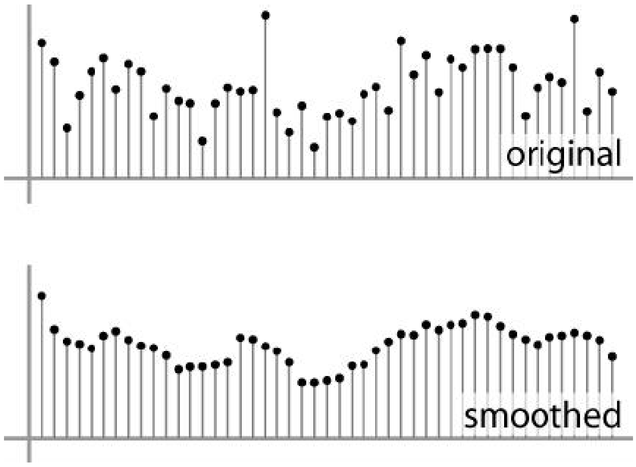

Figure 1: Moving Average Filter in 1D .

Convolution

You are given two signals (sequences). One of the sequences is flipped before sliding over the other. The notation representing convolution is $f * g$. In 1D one has:

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[n-m]. \tag{1.2}$$

In 2D one has:

$$\begin{aligned} G[i,j] &= H * F \\ &= \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i-u,j-v]. \end{aligned} \tag{1.3}$$

Convolution is *linear*, *associative* and *commutative.* In other words, convolution can be described as replacing each pixel with a linear combination of its neighbors. The filter H is also called kernel or mask. Difference between convolution and correlation: correlation is a metric for similarity between two different signals. Convolution applies one signal to the other.

1.1.3 Gaussian Filter

What if we want the **closest** pixels to have a higher influence on the output? One can use the filter function

$$h(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}. \tag{1.4}$$

Many parameters are contained here: which matter?

- The size of the kernel. The Gaussian has generally infinite support but discrete filters use finite kernels.

- The **variance** σ^2 of gaussian: determines extent of smoothing (larger variance, larger smoothing).

Boundary Issues

Edges of the image represent a problematic part of the filtering procedure. In fact, the filter window falls off the edge of the image. We need to pad the image borders with either one of the following methods:

- Zero padding, i.e. black filling.
- Wrap around
- Copy edge.
- Reflect across edge.

1.1.4 Median Filter

Linear smoothing filters do not alleviate salt and pepper noise. The median filter is a non-linear filter and removes spikes. This is good for impulses and salt and pepper noise. The filter computes the median value and replaces the high value with that. This has two major consequences:

- Advantage: it preserves sharp transitions.
- Disadvantage: it removes small brightness variations.

1.2 High-pass Filtering (Edge Detection)

The ultimate goal of edge detection, is to have an idealized line drawing. In general, the edge contours in the image correspond to important scene contours. But what are edges? Edges are nothing else than sharp intensity changes. Images can be expressed as 2-dimensional functions $f(x, y)$. Edges correspond to the **extrema of the derivative** of the intensity function, i.e. represent rapid intensity changes.

1.2.1 Differentiation and Convolution

For discrete data, it holds:

$$\frac{\partial f(x,y)}{\partial x} \approx \frac{f(x+1,y)-f(x,y)}{1}. \tag{1.5}$$

The above calculation can be implemented as a filter. In fact, partial derivatives of an image are represented with $\begin{pmatrix} -1 & 1 \end{pmatrix}$ in x and as $\begin{pmatrix} -1 \\ 1 \end{pmatrix}$ in y. Other finite differences methods are the

- **Prewitt Filter**, represented with

$$G_x = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}, \qquad G_y = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \tag{1.6}$$

- **Sobel Filter**, represented as

$$G_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, \qquad G_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \tag{1.7}$$

The **Gradient** of an image if given by

$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \end{pmatrix}. \tag{1.8}$$

The **gradient direction** (orienation of the edge normal) is given as

$$\Theta = \tan^{-1}\left(\frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}}\right). \tag{1.9}$$

The **edge strength** is given by the gradient magnitude

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial y}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}. \tag{1.10}$$

1.2.2 Handling Noise

If we differentiate a noisy signal, we get infinite many peaks and the signal is practically useless. Solutions are:

- First **smooth** the signal (with a convolution with the introduced gaussian filter h), then differentiate.
- Combining the two, use the convolution property $\frac{\partial}{\partial x}(h * f) = (\frac{\partial}{\partial x}h) * f$.

1.2.3 Laplacian of a Gaussian

Consider the laplacian of gaussian

$$\frac{\partial^2}{\partial x^2}(h * f). \tag{1.11}$$

This special function can be used to find edges in the figure. Where is an edge? This can be found at the zero-crossing.

1.2.4 The Canny Edge-Detection Algorithm

This algorithm has been implemented in 1986 and is composed of the following steps:

- We compute the gradient of the smoothed image in both directions. (convolve the image with x and y derivatives of Gaussian filters)
- We discard pixels whose gradient magnitude is below a certain threshold.
- **Non-maximal suppression**: we identify local maxima along the gradient direction. High intensity means high probability of the presence of an edge: this is not enough. Only local maxima can be considered as part of an edge. A local maxima can be found where the gradient derivative is 0.
  - Compare the edge strength of the current pixel with the edge strength of the pixel in the positive and negative gradient directions.
  - If the edge strength of the current pixel is the largest compared to the other pixels in the mask with the same direction (i.e., the pixel that is pointing in the y-direction will be compared to the pixel above and below it in the vertical axis), the value will be preserved. Otherwise, the value will be suppressed.

1.2.5 Summary

- Smoothing filters:
  - Have positive values.
  - Sum to 1 → preserve brightness of constant regions.
  - Remove high frequency components.
- Derivative Filters
  - Have opposite signs, used to get high response in regions of high constrast.
  - Sum up to 0 → no response in constant regions.
  - Highlight high frequency components.

1.3 Understanding Check

Are you able to:

- *Explain the differences between convolution and correlation*?
- *Explain the differences between a box filter and a Gaussian filter*?
- *Explain why should one increase the size of the kernel of a Gaussian filter if σ is large (i.e. close to the size of the filter kernel*?
- *Explain when would we need a median filter*?
- *Explain how to handle boundary issues?*
- *Explain the working principle of edge detection with a 1D signal?*
- *Explain how noise does affect this prodcedure*?
- *Explain the differential property of convolution*?
- *Show how to compute the first derivative of an image intensity function along x and y?*
- *Explain why the Laplacian of Gaussian operator is useful?*
- *List the properties of smoothing and derivative filters?*
- *Illustrate the Canny edge detection algorithm*?
- *Explain what is non-maxima suppression and how is it implemented*?