IIST, Thiruvananthapuram | AVD624 - Computer Vision

Lecture Notes: Week - 3

08/02/2022 - 11/02/2022

Lecturer: Dr. Deepak Mishra | *Summarised by: Vatsalya Gupta*

This is a weekly summary of the lectures given by *Dr. Deepak Mishra*, Prof. and Head of Department of Avionics, for the course on *Computer Vision* conducted during the even semester 2021-22.

1 Image Processing

An image is a collection of pixels of varying intensity. It is a 2D function in the domain [x y] and the range depending upon the bit depth. We can do two basic types of image transformations:

- Filtering: change in pixel values, i.e. range of image function, $G(x) = h\{F(x)\}$
- Warping: change in pixel locations, i.e. domain of image function, $G(x) = F(h\{x\})$

2 Image Filtering

Image filtering can be of two types - point operation and neighbourhood operation. Point operation is the one where the output of the transformation function corresponds to only one pixel of the input. Whereas in neighbourhood operation, the output depends upon certain neighbourhood of pixels.

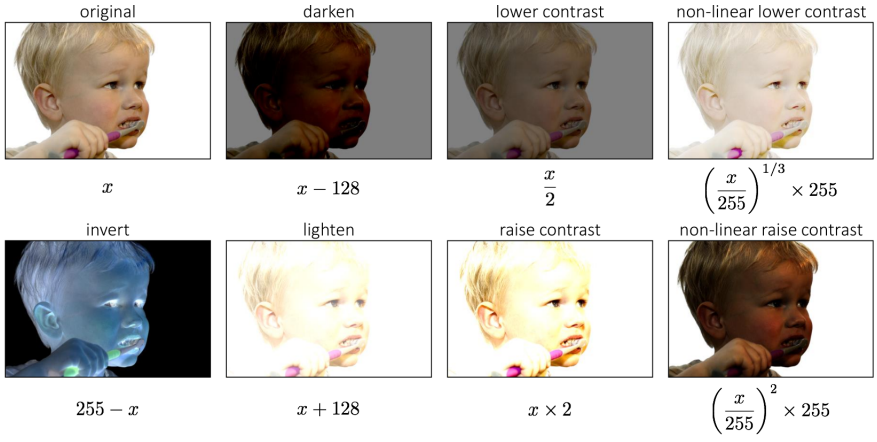


Figure 1: Examples of Point Processing

2.1 Linear Shift-Invariant Image Filtering

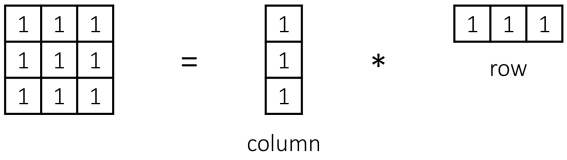
An LSI system has two essential properties - linearity and shift invariance, i.e. $f(ax+by) = af(x)+bf(y)$ and $f(x(t+\tau)) = y(t+\tau)$. In LSI filtering, we choose a filter kernel and shift it across all pixel locations and replace each pixel by a linear combination of its neighbours. For example, box filter, Gaussian filter etc.

2.2 Convolution

The convolution operation can be defined for 1D continuous signals as $(f * g)(x) = \int_{-\infty}^{\infty} f(y)g(x-y)\,dy$, and for 2D discrete signals as $(f * g)(x,y) = \sum_{i,j=-\infty}^{\infty} f(i,j)I(x-i,y-j)\,dy$. An analogous expression for discrete 2D correlation can also be defined $(f * g)(x,y) = \sum_{i,j=-\infty}^{\infty} f(i,j)I(x+i,y+j)\,dy$, where f is the filter, I is the input image and $(f * g)$ is the filtered image.

2.3 Separable Filters

A 2D filter is separable if it can be written as the product of a column and a row. 2D convolution with a separable filter is equivalent to two 1D convolutions. If the image has $M \times M$ pixels and the filter kernel has size $N \times N$, then the cost of convolution with a non-separable filter will be $M^2 \times N^2$, but with a separable filter it will be $2 \times N \times M^2$. An example of this can be a box filter.



Note - There may be overflow issues while applying the kernel at the boundary of an image. To resolve such issues, we use different padding techniques such as zero pad, circular, replicate and symmetric.

3 Types of Filters

3.1 Box Filter

Box filters are non-isotropic filters, which smooth further along diagonals than along rows and columns. Weights have abrupt cutoff leading to discontinuities in the output. Repeated application of a box blur will approximate a Gaussian blur.

3.2 Gaussian Filter

Gaussian filters are linear low-pass filters with rotational symmetry. The weights give higher significance to pixels near the edge. Since large filters are implemented using small 1D filter, they are computationally efficient. An important property is that a 2D Gaussian can be represented as a product of two Gaussians, hence it is separable with a complexity $O(n^2 m)$. The kernel values are sampled from the following 2D Gaussian function,

$$f(i,j) = \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{2\sigma^2}}$$

3.3 Sobel Filter

Sobel filter can be thought of as a combination of blurring and 1D derivative filter. They can either do horizontal operation or vertical operation based upon the requirement.

Horizontal Sober filter:

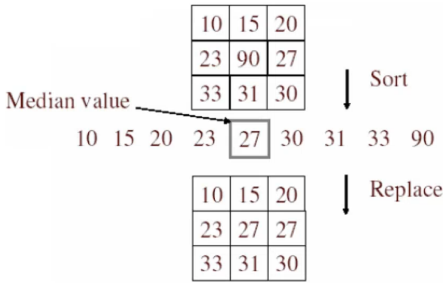
$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$$

Vertical Sobel filter:

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

3.4 Median Filter

A median filter operates over a window by selecting the median intensity in the window. This is particularly helpful in removing salt and pepper noise from the images. But, using a larger kernel size may cause patches in the output image.



3.5 Gabor Filter

Gabor filter is a linear filter used for texture analysis, i.e. it analyzes whether there is any specific frequency content in the image in specific directions in a localised region around the point or region of analysis. If we know the basis images, we can say that any image will be some kind of linear combination of the basis images, i.e. $I = \sum \alpha f_i f_j^T$. In the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave.

$$\psi(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}} cos(2\pi\mu x)$$

Note - There are also other filters possible having different kernels, such as sharpening filters, filters which shift every pixel according to a certain criteria, Scharr filter, Prewitt filter, Roberts filter etc.

4 Image Gradients

Image gradients or edges are very sharp discontinuities in intensity of neighbouring pixels. We can identify these edges by taking derivatives, since they are large at discontinuities. The method of finite differences can be used to differentiate a discrete image. Following steps can be followed for computing image gradients,

1. Select any derivative filter like Sobel, Scharr, Prewitt, Roberts etc.

2. Convolve with the image to compute derivatives.

3. Form the image gradient and compute its direction and amplitude.

$$\nabla \boldsymbol{f} = \left[\frac{\partial \boldsymbol{f}}{\partial x}, \frac{\partial \boldsymbol{f}}{\partial y}\right] \qquad \theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right) \qquad ||\nabla f|| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

gradient direction amplitude

4.1 Edge Detection

An edge in an image may be due to object boundaries, surface normal discontinuities and boundaries of material properties. Edges are of three types - step, ridge, roof. We can detect an edge using a derivative filter. But since differentiation is sensitive to high frequency noise (due to it being high pass in nature), it is critical to blur first. So, we use the derivative of Gaussian to obtain the output at the point where the derivative is sharp. Another method is to use a second derivative filter, such as Laplace filter [1 -2 1], along with Gaussian filtering. This will give us zero crossing at the edges. Zero crossings are not very convenient, but they are more accurate at localizing edges.

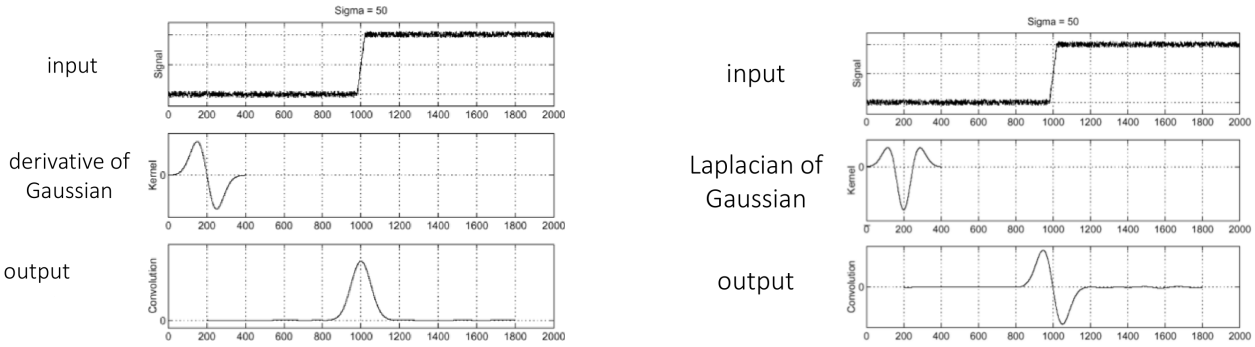


Figure 2: Edge detection - derivative of Gaussian, Laplacian of Gaussian

Note - Apart from edge detection, other applications of image gradients include finding boundaries and morphological operations, such as dilation, erosion, opening and closing.

5 Template Matching

Template matching refers to determining the relation between two images or signals, i.e. whether a certain patch or template can be found in another image or signal. This can be done using various techniques depending upon the speed and invariance - zero-mean, sum of squared differences, normalised cross-correlation. For example, if I_1 and I_2 are similar, then their KL divergence (non-negative, non-symmetric) will be zero, sum of squared differences $\sum(I_1 - I_2)^2$ will be low, and correlation will be high.

Note - Minimising the sum of squared differences of two signals, maximises their correlation and vice-versa.

■ ■ ■