

# **DIGITAL IMAGE PROCESSING, LECTURE 7**

## **ORDER-STATISTIC (NONLINEAR) FILTERS**

Order-statistic filters are nonlinear spatial filters whose response is based on ordering (ranking) the pixels contained in the region encompassed by the filter. Smoothing is achieved by replacing the value of the center pixel with the value determined by the ranking result. The best-known filter in this category is the median filter, which, as its name implies, replaces the value of the center pixel by the median of the intensity values in the neighborhood of that pixel (the value of the center pixel is included in computing the median). Median filters provide excellent noise reduction capabilities for certain types of random noise, with considerably less blurring than linear smoothing filters of similar size. Median filters are particularly effective in the presence of impulse noise (sometimes called salt-and-pepper noise, when it manifests itself as white and black dots superimposed on an image).

The median,  $\xi$ , of a set of values is such that half the values in the set are less than or equal to  $\xi$  and half are greater than or equal to  $\xi$ . In order to perform median filtering at a point in an image, we first sort the values of the pixels in the neighborhood, determine their median, and assign that value to the pixel in the filtered image corresponding to the center of the neighborhood. For example, in a  $3 \times 3$  neighborhood the median is the 5th largest value, in a  $5 \times 5$  neighborhood it is the 13th largest value, and so on. When several values in a neighborhood are the same, all equal values are grouped. For example, suppose that a  $3 \times 3$  neighborhood has values (10, 20, 20, 20, 15, 20, 20, 25, 100). These values are sorted as (10, 15, 20, 20, 20, 20, 20, 25, 100), which results in a median of 20. Thus, the principal function of median filters is to force points to be more like their neighbors. Isolated clusters of pixels that are light or dark with respect to their neighbors, and whose area is less than  $m^2/2$  (one-half the filter area), are forced by an  $m \times m$  median filter to have the value of the median intensity of the pixels in the neighborhood.

## **SHARPENING (HIGHPASS) SPATIAL FILTERS**

Sharpening highlights transitions in intensity. Uses of image sharpening range from electronic printing and medical imaging to industrial inspection and autonomous guidance in military systems. In previous lecture, we saw that image blurring could be accomplished in the spatial domain by pixel averaging (smoothing) in a neighborhood. Because averaging is analogous to integration, it is logical to conclude that sharpening can be accomplished by spatial differentiation. In fact, this is the case, and the following discussion deals with various ways of defining and implementing operators for sharpening by digital differentiation. The strength of the response of a derivative operator is proportional to the magnitude of the intensity discontinuity at the point at which the operator is applied. Thus, image differentiation enhances edges and other discontinuities (such as noise) and de-emphasizes areas with slowly varying intensities. As noted in the previous lecture, smoothing is often referred to as lowpass filtering, a term borrowed from frequency domain processing. In a similar manner, sharpening is often referred to as highpass filtering. In this case, high frequencies (which are responsible for fine details) are passed, while low frequencies are attenuated or rejected.

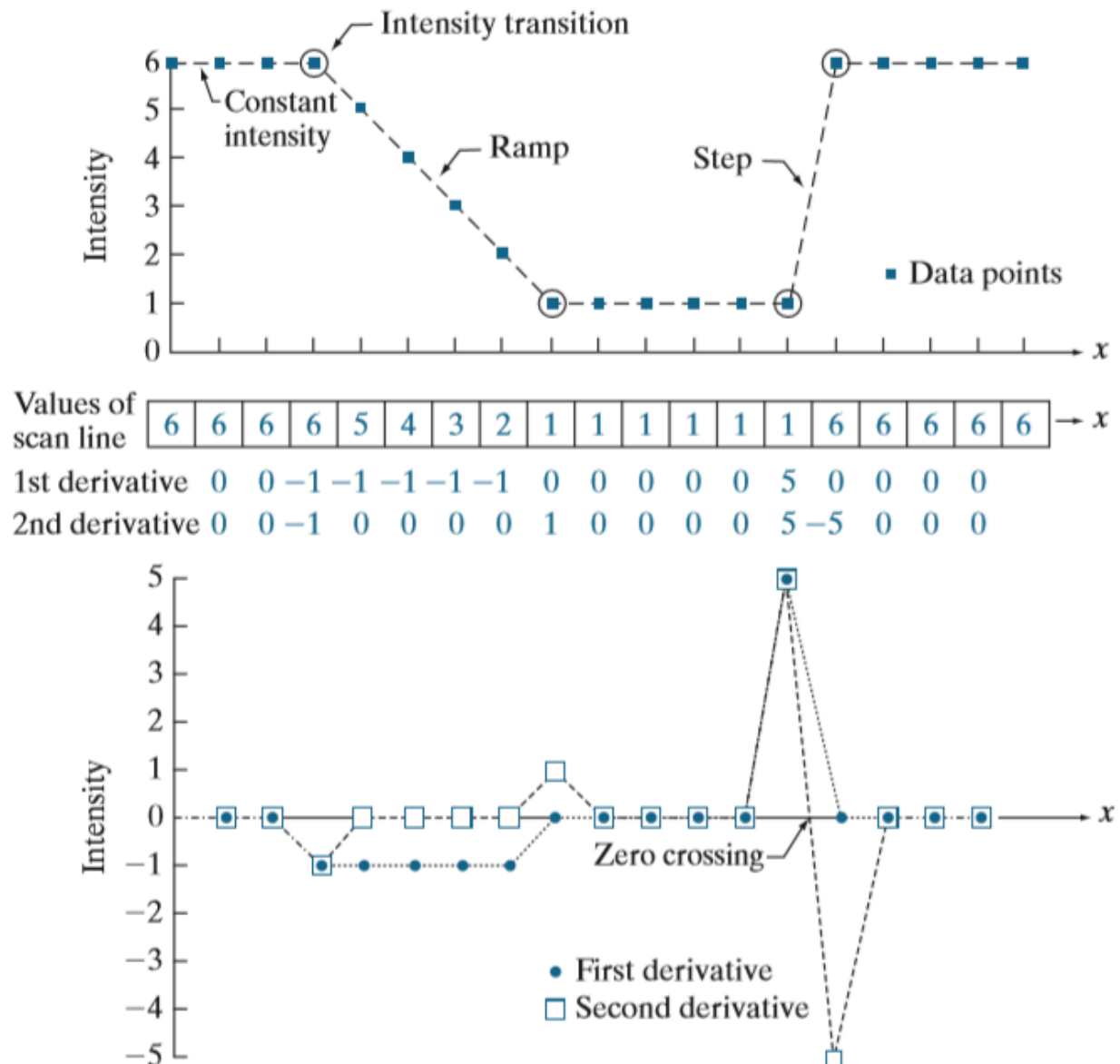


Figure 1: (a) A section of a horizontal scan line from an image, showing ramp and step edges, as well as constant segments. (b) Values of the scan line and its derivatives. (c) Plot of the derivatives, showing a zero crossing. In (a) and (c) points were joined by dashed lines as a visual aid.

Edges in digital images often are ramp-like transitions in intensity, in which case the first derivative of the image would result in thick edges because the derivative is non-zero along a ramp. On the other hand, the second derivative would produce a double edge one pixel thick, separated by zeros. From this, we conclude that the second derivative enhances fine detail much better than the first derivative, a property ideally suited for sharpening images. Also, second derivatives require fewer operations to implement than first derivatives, so our initial attention is on the former.

## USING THE SECOND DERIVATIVE FOR IMAGE SHARPENING—THE LAPLACIAN

In this section we discuss the implementation of 2-D, second-order derivatives and their use for image sharpening. The approach consists of defining a discrete formulation of the second-order derivative and then constructing a filter kernel based on that formulation.

It can be shown (Rosenfeld and Kak [1982]) that the simplest isotropic derivative operator (kernel) is the *Laplacian*, which, for a function (image)  $f(x, y)$  of two variables, is defined as

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

and, similarly, in the y-direction, we have

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

It follows from the preceding three equations that the discrete Laplacian of two variables is

$$\nabla^2 f(x, y) = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y) \quad (1)$$

0	1	0	1	1	1	0	-1	0	-1	-1	-1
1	-4	1	1	-8	1	-1	4	-1	-1	8	-1
0	1	0	1	1	1	0	-1	0	-1	-1	-1

Figure 2: (a) Laplacian kernel used to implement Eq. (1). (b) Kernel used to implement an extension of this equation that includes the diagonal terms. (c) and (d) Two other Laplacian kernels.

Because the Laplacian is a derivative operator, it highlights sharp intensity transitions in an image and de-emphasizes regions of slowly varying intensities. This will tend to produce images that have grayish edge lines and other discontinuities, all superimposed on a dark, featureless background. Background features can be “recovered” while still preserving the sharpening effect of the Laplacian by adding the Laplacian image to the original. As noted in the previous paragraph, it is important to keep in mind which definition of the Laplacian is used. If the definition used has a negative center coefficient, then we subtract the Laplacian image from the

original to obtain a sharpened result. Thus, the basic way in which we use the Laplacian for image sharpening is

$$g(x, y) = f(x, y) + c[\nabla^2 f(x, y)] \quad (2)$$

where  $f(x, y)$  and  $g(x, y)$  are the input and sharpened images, respectively. We let  $c = -1$  if the Laplacian kernels in Fig. 2(a) or (b) is used, and  $c = 1$  if either of the other two kernels is used.

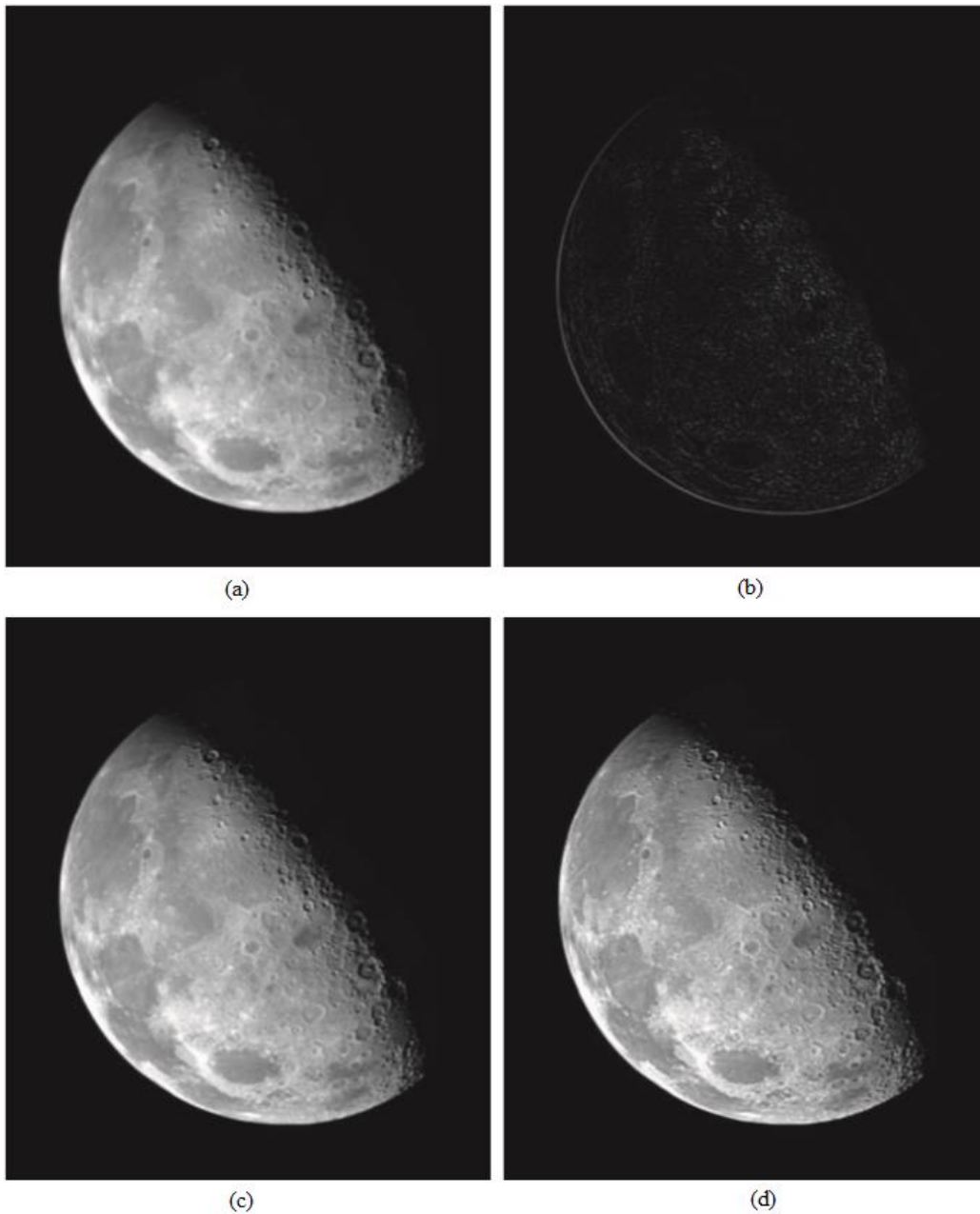


Figure 3: (a) Blurred image of the North Pole of the moon. (b) Laplacian image obtained using the kernel in Fig. 2(a). (c) Image sharpened using Eq. (3-54) with  $c = -1$ . (d) Image sharpened using the same procedure, but with the kernel in Fig. 3.45(b).

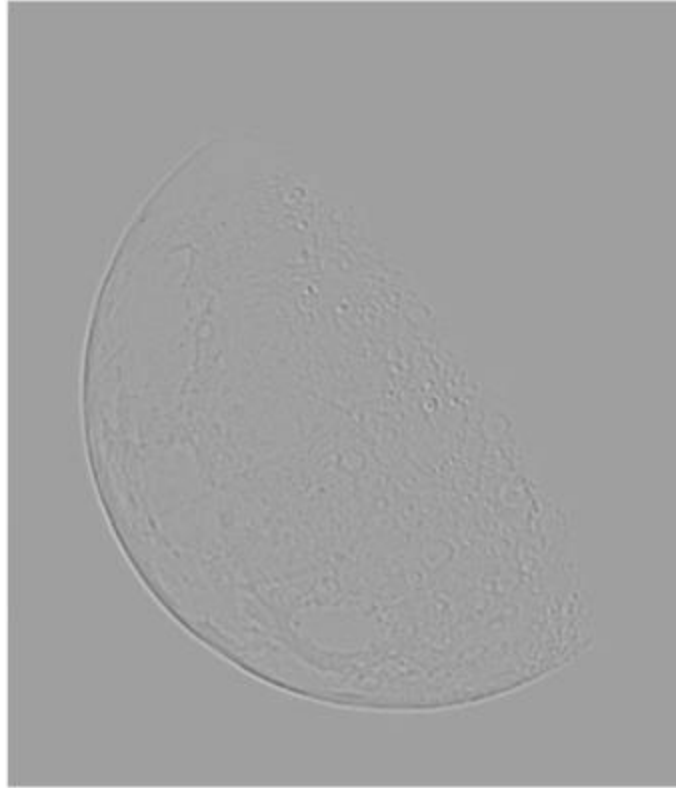


Figure 4: The Laplacian image from Fig. 3(b), scaled to the full  $[0, 255]$  range of intensity values. Black pixels correspond to the most negative value in the unscaled Laplacian image, grays are intermediate values, and white pixels corresponds to the highest positive value.

## UNSHARP MASKING AND HIGHBOOST FILTERING

Subtracting an unsharp (smoothed) version of an image from the original image is process that has been used since the 1930s by the printing and publishing industry to sharpen images. This process, called unsharp masking, consists of the following steps:

1. Blur the original image.
2. Subtract the blurred image from the original (the resulting difference is called the *mask*.)
3. Add the mask to the original.

Letting  $f(x, y)$  denote the blurred image, the mask in equation form is given by:

$$g_{mask}(x, y) = f(x, y) - \bar{f}(x, y)$$

Then we add a weighted portion of the mask back to the original image:

$$g(x, y) = f(x, y) + k g_{mask}(x, y) \quad (3)$$

where we included a weight,  $k(k \geq 0)$  for generality. When  $k = 1$  we have unsharp masking, as defined above. When  $k > 1$ , the process is referred to as highboost filtering. Choosing  $k < 1$  reduces the contribution of the unsharp mask.

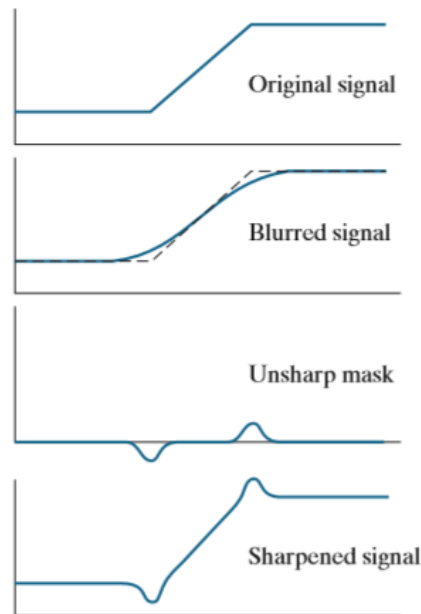


Figure 5: 1-D illustration of the mechanics of unsharp masking. (a) Original signal. (b) Blurred signal with original shown dashed for reference. (c) Unsharp mask. (d) Sharpened signal, obtained by adding (c) to (a).



Figure 6: (a) Original image of size  $600 \times 259$  pixels. (b) Image blurred using a  $31 \times 31$  Gaussian lowpass filter with  $\sigma = 5$ . (c) Mask. (d) Result of unsharp masking using Eq. (3) with  $k = 1$ . (e) Result of highboost filtering with  $k = 45$ .

## References and further reading:

Digital Image Processing, 4th edition, Gonzalez, Rafael and Woods, Richard, 2018