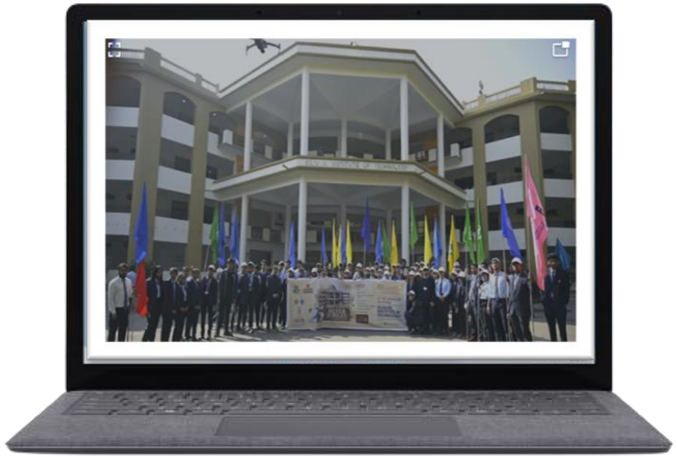


Image Sharpening: High Pass Filtering

Image Sharpening

- 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.
- 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.

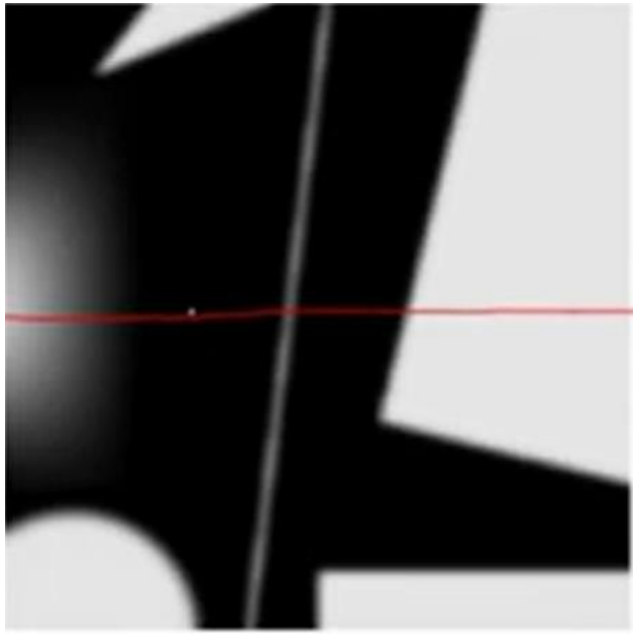
Image Sharpening

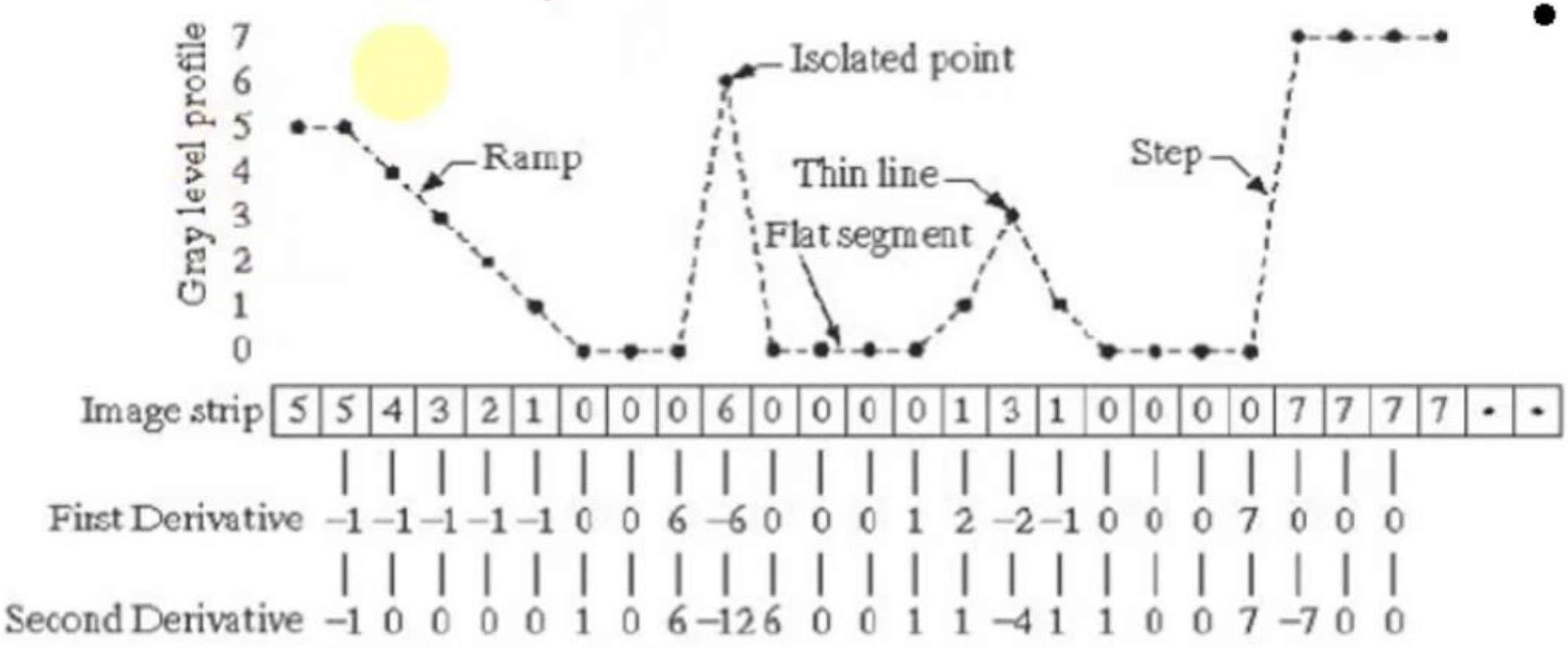
- a derivative operator is proportional to the magnitude of the intensity discontinuity at the point at which the operator is applied.
- image differentiation enhances edges and other discontinuities (such as noise) and de-emphasizes areas with slowly varying intensities.
- smoothing is often referred to as lowpass filtering, a term borrowed from frequency domain processing.
- sharpening is often referred to as highpass filtering.
- high frequencies (which are responsible for fine details) are passed, while low frequencies are attenuated or rejected.

Foundation of Highpass Filtering

- Derivatives of a digital function are defined in terms of differences. We use definition of a first derivative as:
  1. Must be zero in areas of constant intensity.
  2. Must be nonzero at the onset of an intensity step or ramp.
  3. Must be nonzero along intensity ramps.
- Definition of a second derivative
  1. Must be zero in areas of constant intensity.
  2. Must be nonzero at the onset and end of an intensity step or ramp.
  3. Must be zero along intensity ramps.

- A basic definition of the first-order derivative of a one-dimensional function f (x) is the difference
- $\frac{dy}{dx} = f(x+1) - f(x)$
- Define 2nd order derivative as difference
- $\frac{d^2y}{dx^2} = f(x+1) + f(x-1) - 2f(x)$




Gray level profile
7
6
5
4
3
2
1
0
Isolated point
Ramp
Thin line
Flat segment
Step
Image strip 5 5 4 3 2 1 0 0 0 6 0 0 0 0 1 3 1 0 0 0 0 7 7 7 7 • •
First Derivative −1 −1 −1 −1 −1 0 0 6 −6 0 0 0 1 2 −2 −1 0 0 0 7 0 0 0
Second Derivative −1 0 0 0 0 1 0 6 −12 6 0 0 1 1 −4 1 1 0 0 7 −7 0 0

- 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 nonzero along a ramp.
- On the other hand, the second derivative would produce a double edge one pixel thick, separated by zeros.
- The second derivative enhances fine detail much better than the first derivative, a property ideally suited for sharpening images

The Laplacian

- 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
- $\nabla^2 f = \frac{d^2 f}{dx^2} + \frac{d^2 f}{dy^2}$
- $\frac{d^2 f}{dx^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$
- $\frac{d^2 f}{dy^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$
- $\nabla^2 f = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$

The Laplacian

- the coefficients of each kernel sum to zero.
- Convolution based filtering implements a sum of products, so when a derivative kernel encompasses a constant region in a image, the result of convolution in that location must be zero.

(a)

0	1	0
1	−4	1
0	1	0

(b)

1	1	1
1	−8	1
1	1	1

(c)

0	−1	0
−1	4	−1
0	−1	0

(d)

−1	−1	−1
−1	8	−1
−1	−1	−1

a b c d

FIGURE 3.45 (a) Laplacian kernel used to implement Eq. (3-53). (b) Kernel used to implement an extension of this equation that includes the diagonal terms. (c) and (d) Two other Laplacian kernels.

The Laplacian

- Constant areas in images filtered with these kernels would be constant also in the filtered image.
- When convolving an image with a kernel whose coefficients sum to zero, it turns out that the pixels of the filtered image will sum to zero also.
- This implies that images filtered with such kernels will have negative values, and sometimes will require additional processing to obtain suitable visual results.
- $g(x,y) = f(x,y) + c[\nabla^2 f], let\ c = -1$

0	1	0
1	-4	1
0	1	1

20	20	20	20	20
20	5	20	20	20
20	20	20	20	20
20	20	20	5	20
20	20	20	20	20

Unsharp Masking And Highboost Filtering

- Subtracting an unsharp (smoothed) version of an image from the original image is used 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 $\bar{f}(x, y)$ denote the blurred image, the mask in equation form is given by: $g_{mask} = f(x, y) - \bar{f}(x, y)$
- Then we add a weighted portion of the mask back to the original image: $g(x, y) = f(x, y) + k g_{mask}$

Using First-order Derivatives For Image Sharpening—the Gradient

- The gradient of an image f at coordinates (x, y) is defined as the two dimensional column vector $\nabla f = grad(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{df}{dx} \\ \frac{df}{dy} \end{bmatrix}$
- The magnitude (length) of vector f, denoted as $M(x, y)$ (the vector norm notation $||\nabla f||$ is also used frequently), where
- $M(x, y) \Vdash \big|\big|\nabla f\big|\big| = mag(\nabla f) = \sqrt{g_x^2 + g_y^2}$
- Another implementation is $M(x, y) \approx |g_x| + |g_y|$

Using First-order Derivatives For Image Sharpening—the Gradient

- the simplest approximations to a first-order derivative that satisfy the conditions stated are $g_x = z_9 - z_5$ and $g_y = z_8 - z_6$.
- we compute the gradient image as $M(x, y) = [(z_9 - z_5)^2 + (z_8 - z_6)]\hat{}(1/2)$
- or $M(x, y) = [|(z_9 - z_5)| \quad + |(z_8 - z_6)|$
- Approximations to g_x and g_y using a 3×3 neighborhood centered on z_5 are as follows:
- $g_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$
- $g_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$

Sobel Operator

- The coefficients in all the kernels sum to zero, so they would give a response of zero in areas of constant intensity, as expected of a derivative operator.
- This kernels are Sobel operators.

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0
0	1

0	−1
1	0

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1