Linear and non-linear filters

- The mean filter is a linear filter:

$$\text{mean}(af + g) = a\,\text{mean}(f) + \text{mean}(g)$$

- It is also space invariant.

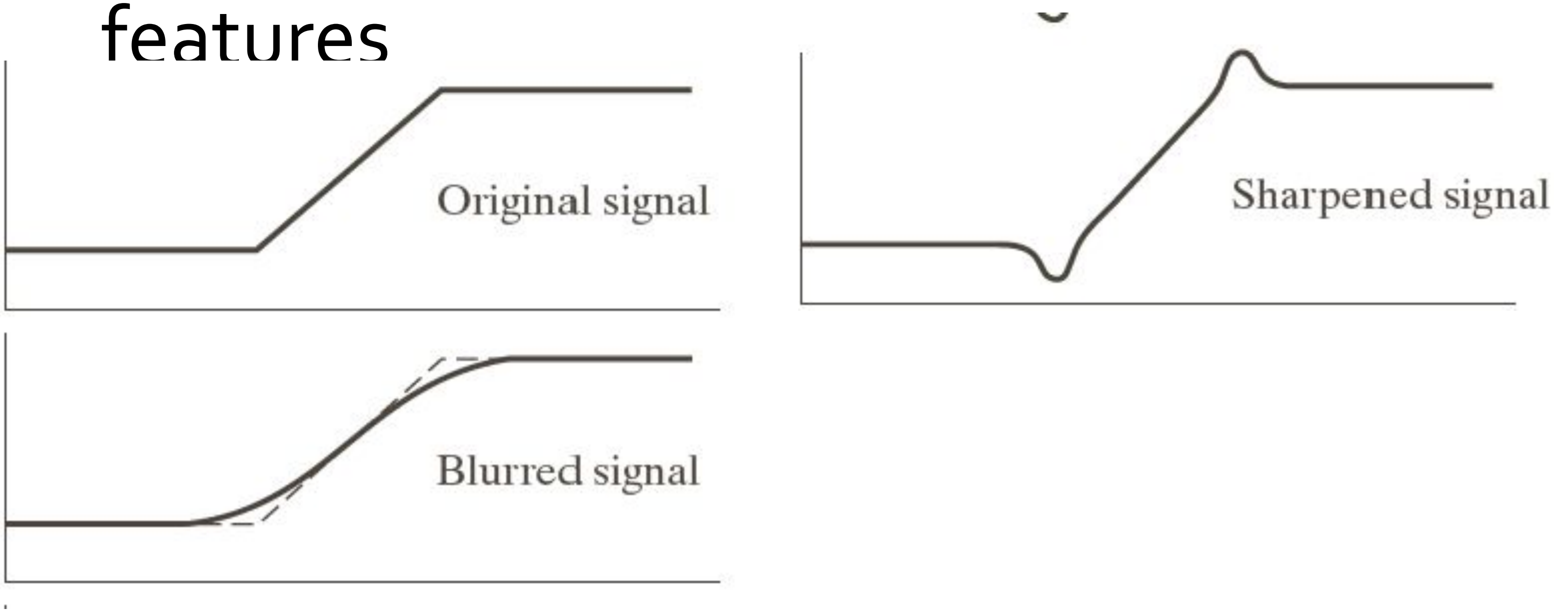
- So it can be implemented as a convolution.

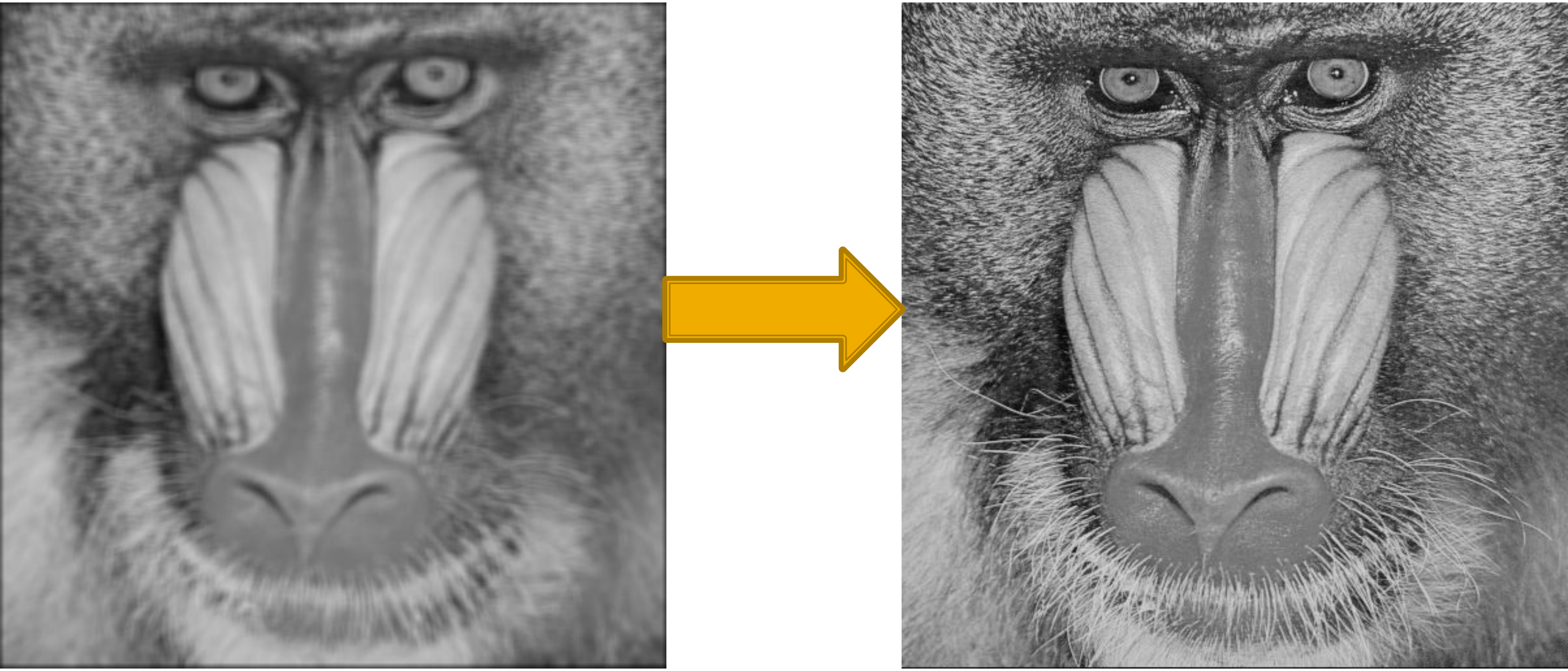
- The median filter cannot be implemented using a convolution as it is nonlinear.

$$\text{median}(f + g) \neq \text{median}(f) + \text{median}(g)$$

Sharpening filter

- Smoothing filters – local average (integration)
- Sharpening filters – local intensity derivatives (differentiation) to enhance local distinctive features





Sharpening filters

- Mean filter = local averaging filter
- Averaging causes blurring, equivalent to integration.
- Sharpening – uses intensity differentiation
- We will compute local image intensity derivatives.
- Greater the derivative magnitude = sharper change in intensity.
- Aim: to add back local derivative magnitudes to the low-contrast image!

Digital Derivative Operators (1D)

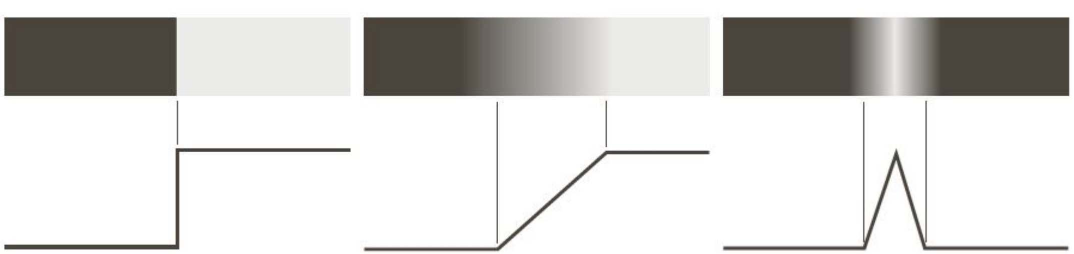
Assume 1-D image for now, i.e. our image is f(x) instead of f(x,y)

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

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

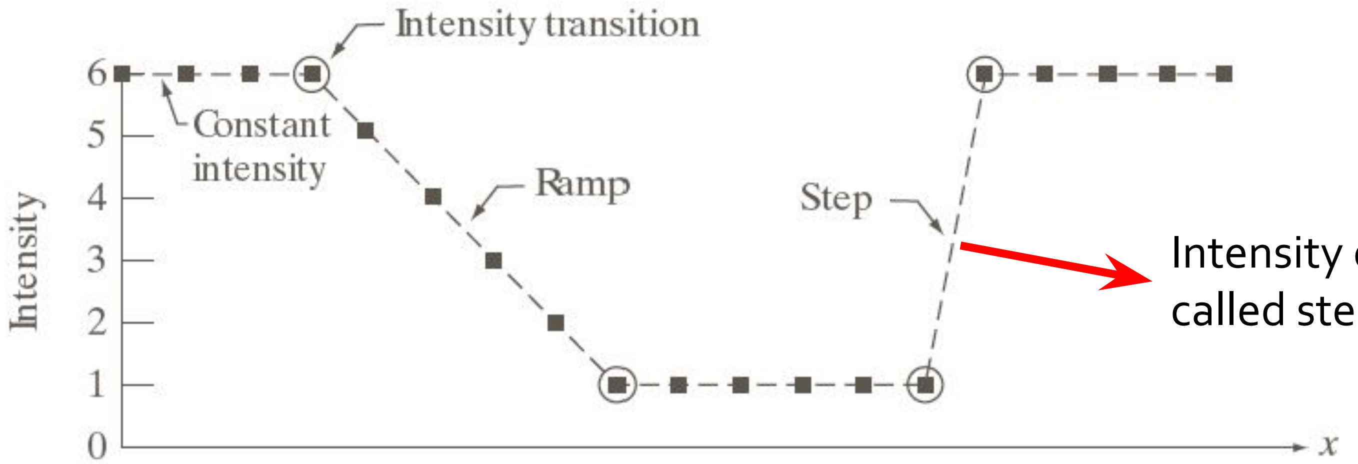
1st derivative: Zero in constant areas, non-zero at the onset of an intensity ramp or intensity step, non-zero along a ramp

2nd derivative: Zero in constant areas, zero along intensity ramps of constant slope, non-zero at the onset and end of an intensity ramp or step



a b c

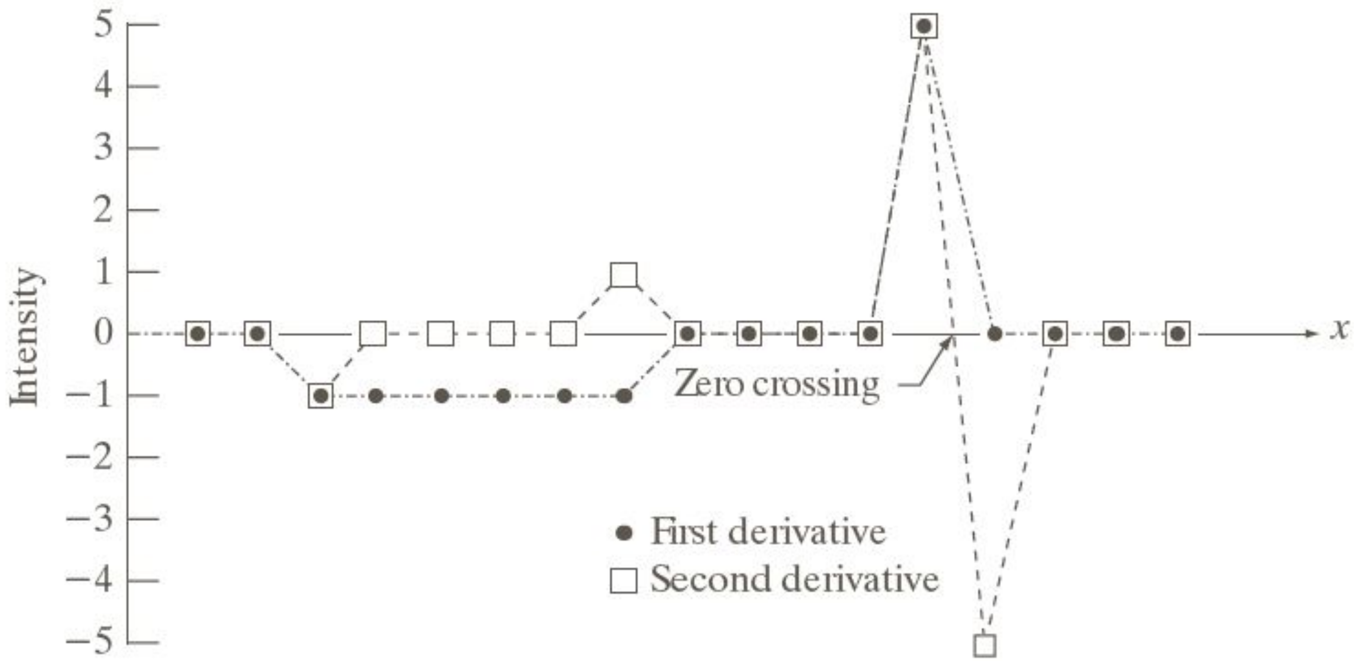
FIGURE 10.8 From left to right, models (ideal representations) of a step, a ramp, and a roof edge, and their corresponding intensity profiles.

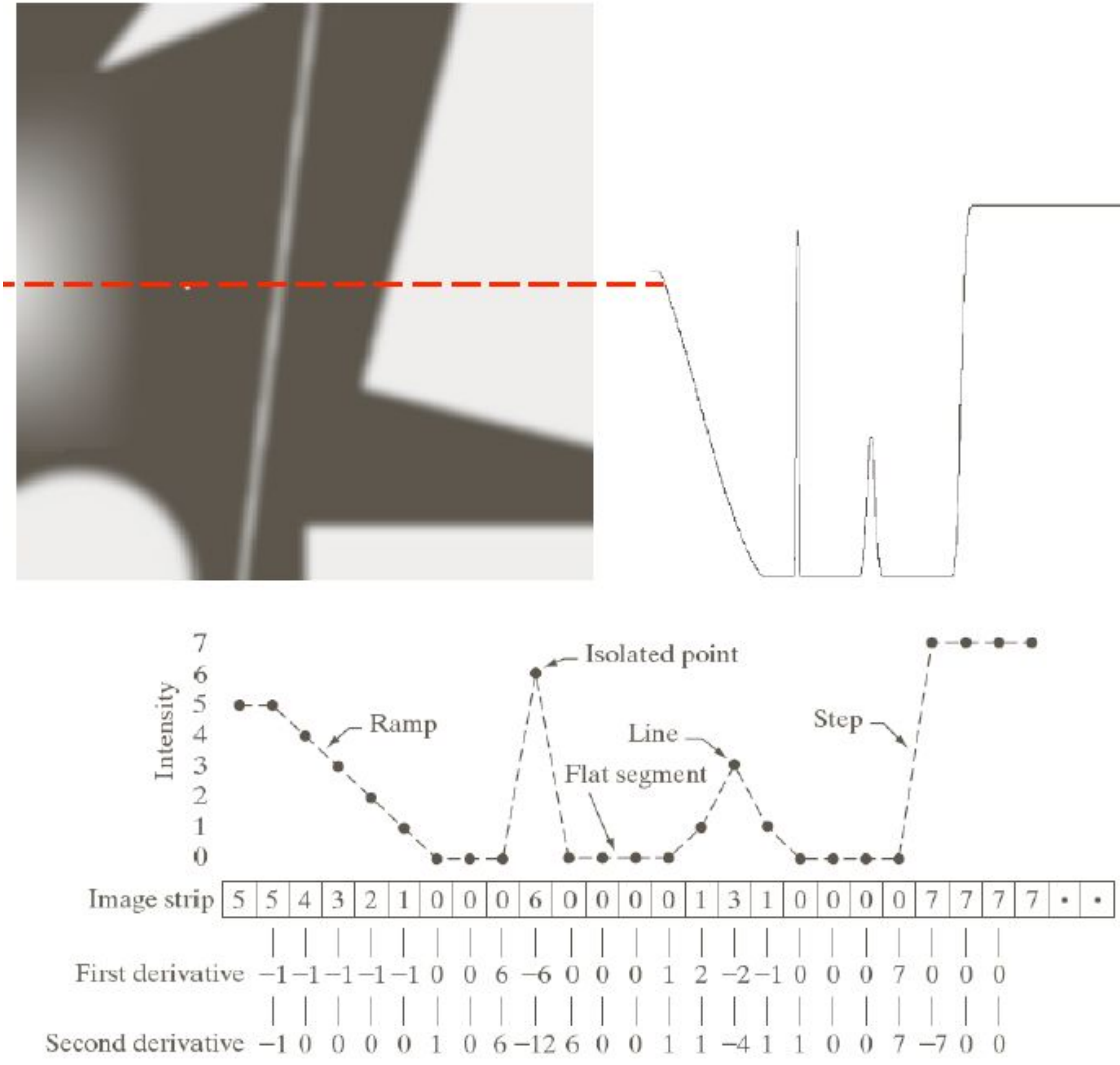


Intensity discontinuity, also called step edge

Scan line	6	6	6	6	5	4	3	2	1	1	1	1	1	1	6	6	6	6	6
1st derivative		0	0	−1	−1	−1	−1	−1	0	0	0	0	0	5	0	0	0	0	
2nd derivative		0	0	−1	0	0	0	0	1	0	0	0	0	5	−5	0	0	0	

From book by Gonzalez and Woods





From book by Gonzalez and Woods

Digital derivatives: first versus second

- Along a ramp (i.e. intensity change with constant slope), first derivative is a non-zero constant.

- Second derivative is zero along the ramp, except at the start and end!

- Second derivative is preferable for image sharpening! Many edges in images are ramp-like, in which case **first** derivative will give **thick** edges (undesirable), whereas **second** derivative gives **thin** edges two pixels wide (desirable).

- Second derivative changes sign midway at a step edge **(zero crossing property).**

Laplacian of an image

- Images are 2D – what kind of second derivative do we use? Along X or Y direction?
- We look at isotropic operators, i.e. operators whose output does not depend upon the direction of image intensity discontinuity.
- That is, we are interested in **rotationally invariant** 2nd derivative operator for images.

Laplacian of an image

- A filter is said to be rotationally invariant if rotating the image and then applying the filter to the rotated image gives the same result as applying the filter to the image and then rotating the result.

Laplacian of an image

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

Rotationally symmetric operator (in the continuous domain)

$$= f(x+1, y) + f(x-1, y) - 2f(x, y)$$
$$+ f(x, y+1) + f(x, y-1) - 2f(x, y)$$
$$= f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Laplacian operators: second operator is obtained by adding second derivatives along both the diagonals, to the first operator

Laplacian of an image

- The two masks on the previous slide are applied point-wise to the entire image.
- For image smoothing as well, we applied point-wise masks.
- But here the mask weights sum to 0. In smoothing the weights summed to 1.

Laplacian for image sharpening

- A Laplacian de-emphasizes regions with slowly varying intensities.
- Highlights intensity discontinuities in an image.
- Subtracting the Laplacian from an image yields sharpening:

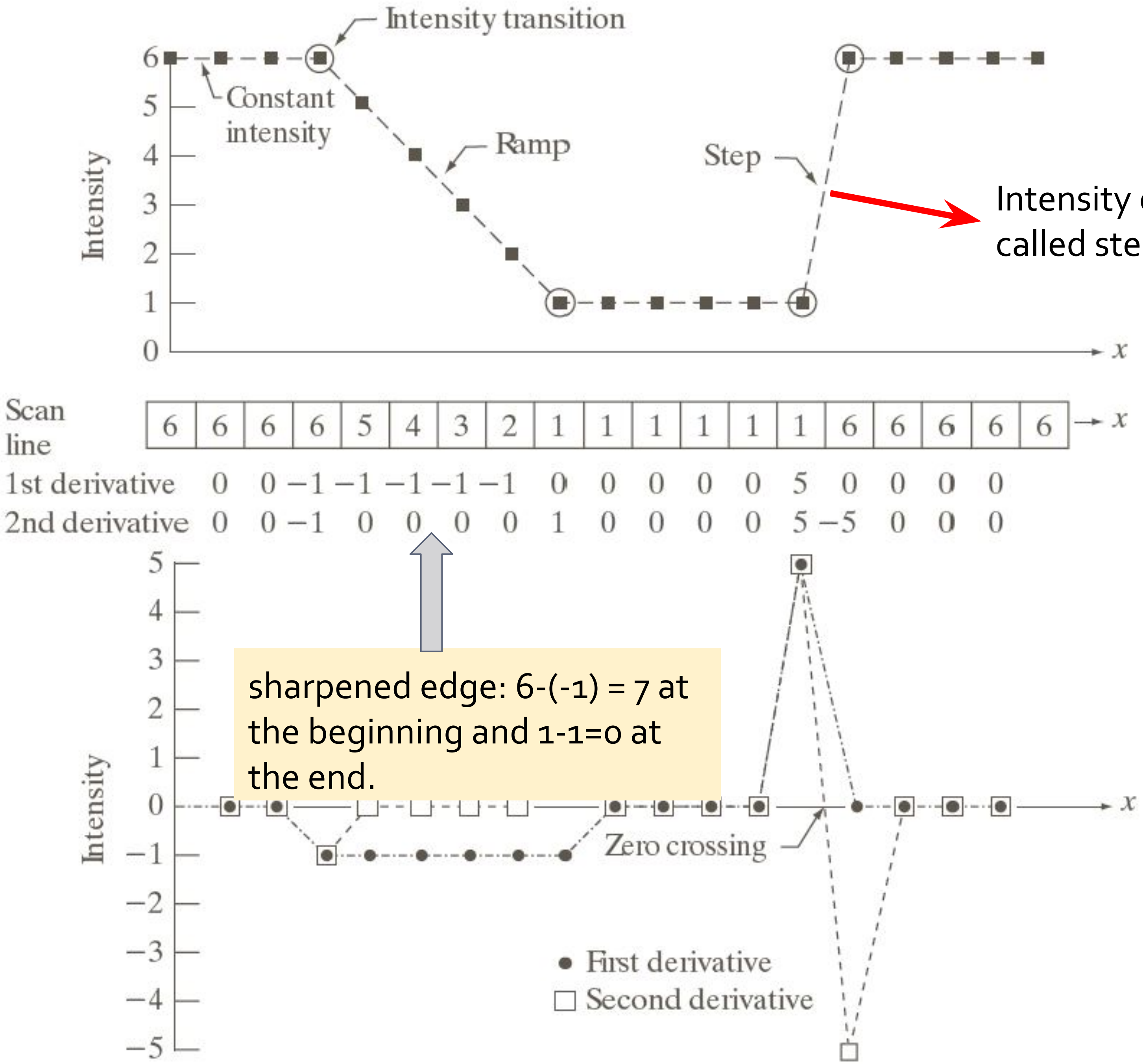
$$g(x,y) = f(x,y) - c\nabla^2 f(x,y), c > 0$$

Laplacian for image sharpening

- Subtracting the Laplacian from an image yields sharpening:

$$g(x,y) = f(x,y) - c\nabla^2 f(x,y), c > 0$$

- Why is this the case?
- Observe that the Laplacian changes sign across an edge.
- For an edge with decreasing intensity, it is negative and positive at the beginning and end of the edge respectively.
- Hence deducting it sharpens the edge. See diagram on next slide



Intensity discontinuity, also called step edge

sharpened edge: 6-(-1) = 7 at the beginning and 1-1=0 at the end.

From book by Gonzalez and Woods

Laplacian for image sharpening

- In some books, the following masks are used for the Laplacian (they are equal to the earlier masks multiplied by -1):

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}, \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

- With such masks, the sharpening filter is expressed as:

$$g(x,y) = f(x,y) + c\nabla^2 f(x,y), c > 0$$

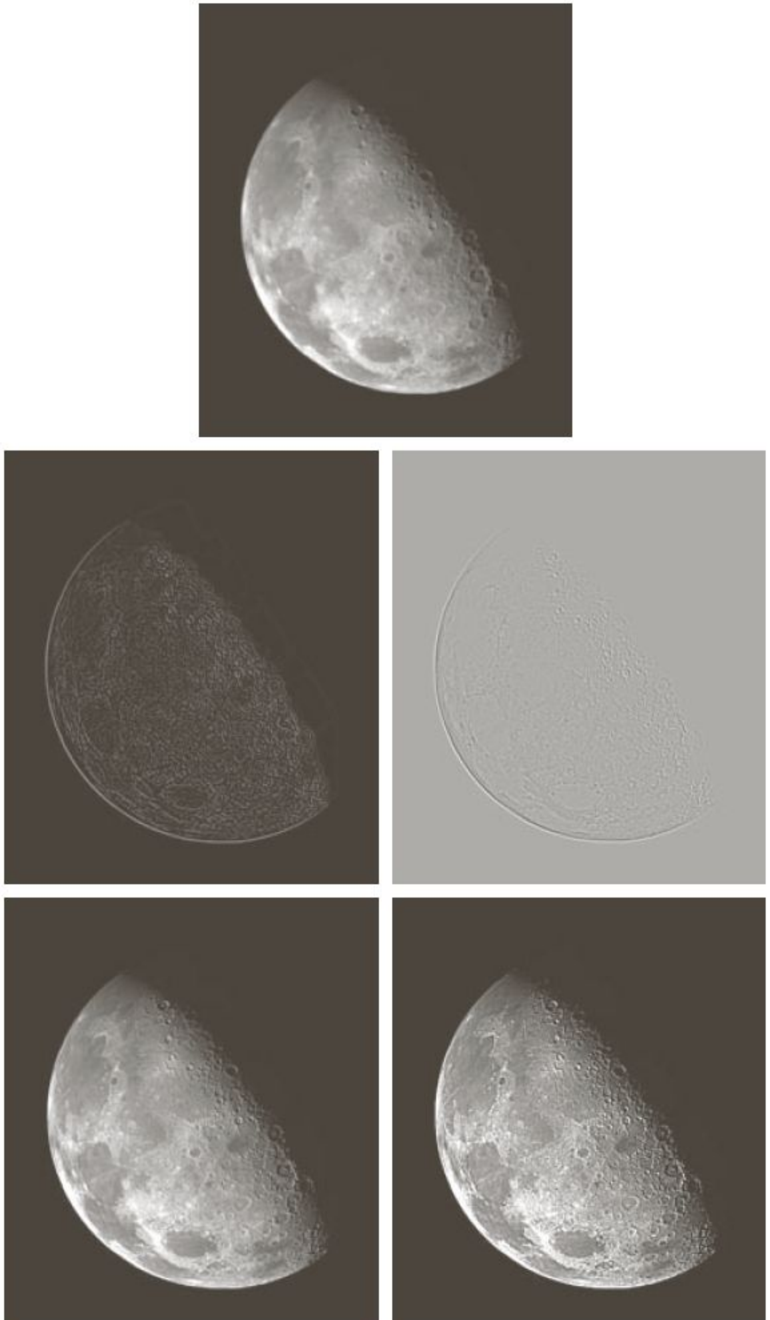


FIGURE 3.38
(a) Blurred image of the North Pole of the moon. (b) Laplacian without scaling. (c) Laplacian with scaling. (d) Image sharpened using the mask in Fig. 3.37(a). (e) Result of using the mask in Fig. 3.37(b). (Original image courtesy of NASA.)

From Book by Gonzalez and woods

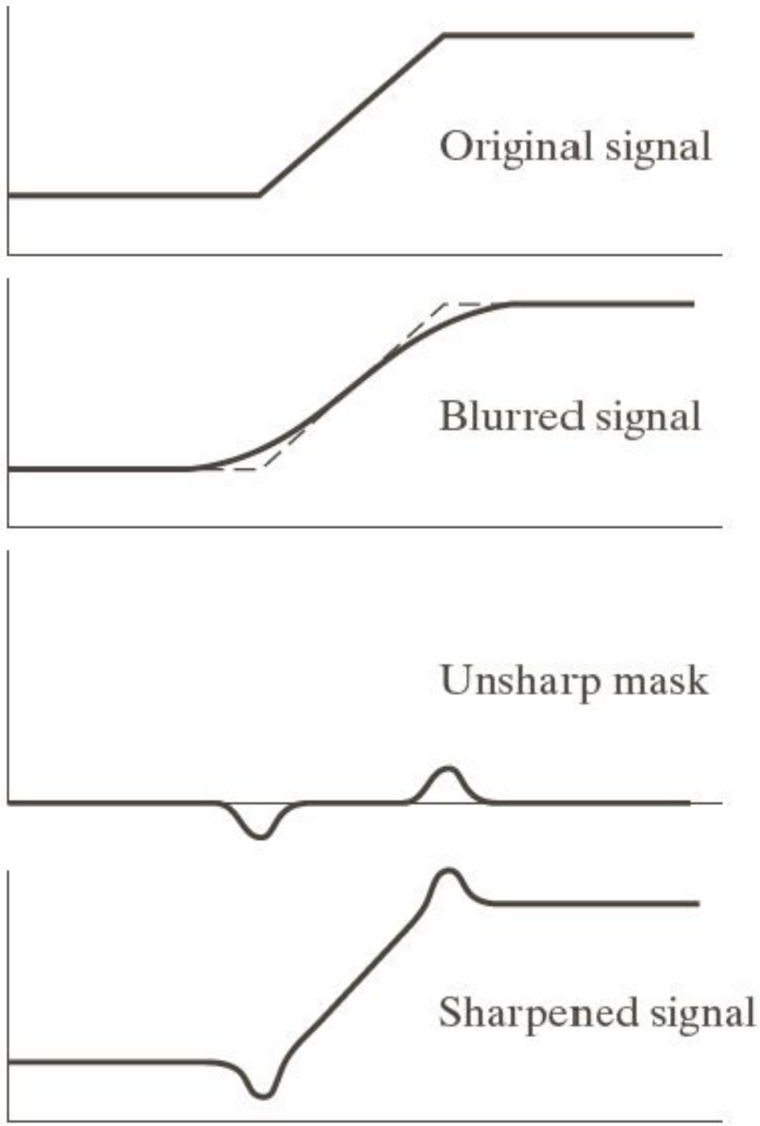
Sharpening filter: convolution mask

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} - c \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & -c & 0 \\ -c & 1+4c & -c \\ 0 & -c & 0 \end{pmatrix}$$

https://en.wikipedia.org/wiki/Unsharp_masking

Unsharp masking

- Smooth the original image f using a filter g yielding $f1 = f*g$.
- Compute the difference between original and smoothed image, i.e. $f2 = f-f*g$.
- Add it back to the original image to yield $f3 = f + k\, f2$ for some scalar k.
- You get a sharpened image.
- If $k = 1$, we call it unsharp masking (i.e. removing blurred components)
- If $k > 1$, we term it highboost filtering.



a
b
c
d

FIGURE 3.39 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).

From Book by Gonzalez and woods

a
b
c
d
e

FIGURE 3.40
(a) Original image.
(b) Result of blurring with a Gaussian filter.
(c) Unsharp mask. (d) Result of using unsharp masking.
(e) Result of using highboost filtering.

Derivative filters: dealing with noise

- Derivative filters in general exacerbate noise.
- Hence they are applied in conjunction with smoothing filters.
- For example, the Sobel filters smooth in the X and Y directions before performing differencing.

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0
0	1

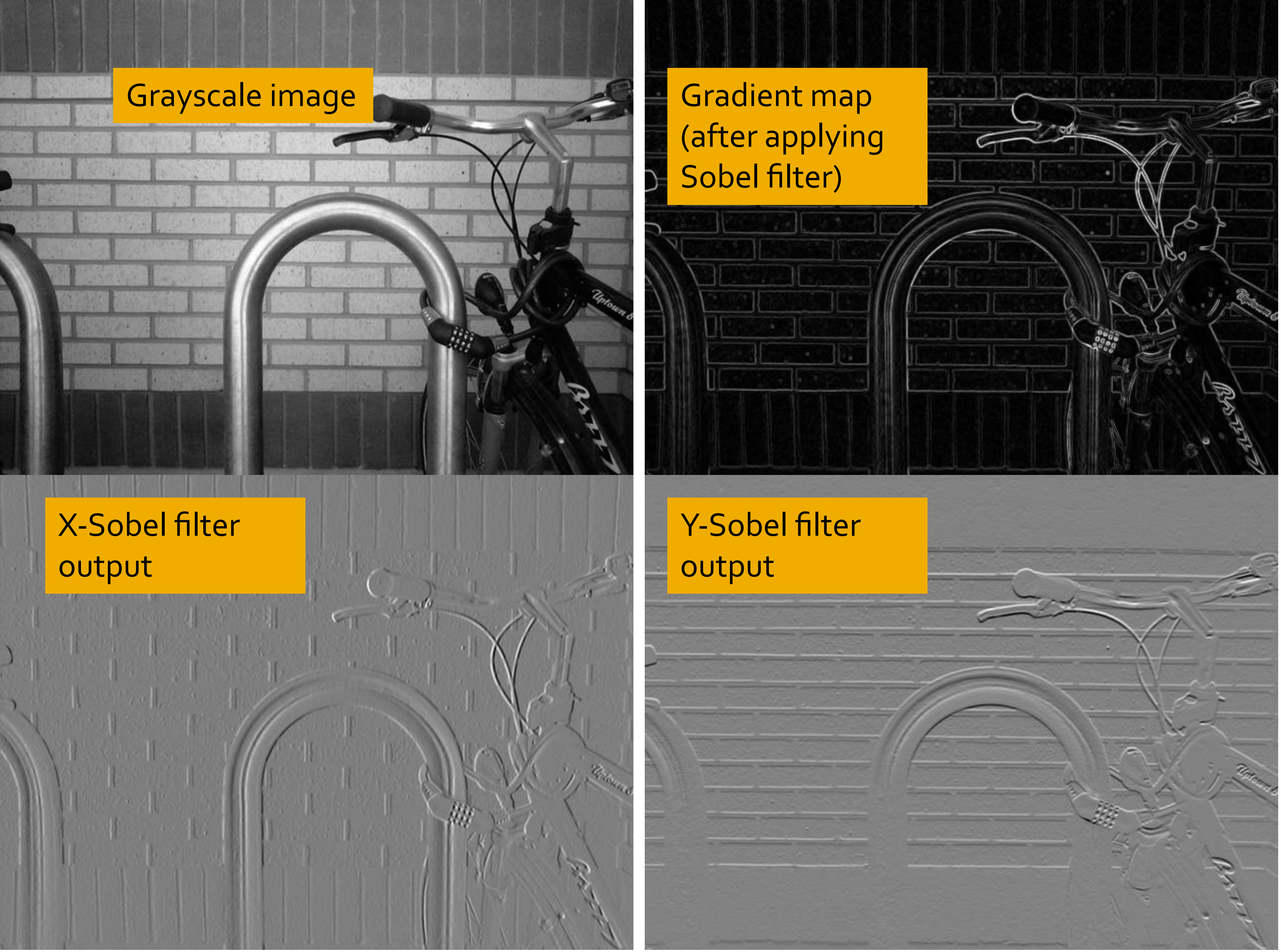
0	−1
1	0

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

a
b c
d e

FIGURE 3.41
A 3×3 region of an image (the zs are intensity values).
(b)–(c) Roberts cross gradient operators.
(d)–(e) Sobel operators. All the mask coefficients sum to zero, as expected of a derivative operator.


Grayscale image
Gradient map (after applying Sobel filter)
X-Sobel filter output
Y-Sobel filter output

Sobel filters: outer products

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

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

- Filter which can be represented in this outer-product form are called separable filters.
- Convolution between a M x N image and a K x L mask has a complexity of O(MNKL), which can be expensive.
- But with separable filters, this reduces to O(MN [L+K]) since you express the K x L mask as the outer product of K x 1 mask and a 1 X L mask. Applying the 1 X L mask requires O(MNL) operations whereas applying the K x 1 mask requires O(MNK) operations. Here, we are using the **associativity** of convolution.

Other examples of separable filters

Mean filter :

$$\frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} = \frac{1}{3}\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \frac{1}{3}\begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$$

2D Gaussian filter (weighted mean filter wit h Gaussian w eights) :

$$g(x, y) = \frac{\exp(-(x^2 + y^2)/(2\sigma^2))}{2\pi\sigma^2}$$

Outerprodu ct :

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \begin{pmatrix} v_1 & v_2 & v_3 \end{pmatrix} = \begin{pmatrix} u_1 v_1 & u_1 v_2 & u_1 v_3 \\ u_2 v_1 & u_2 v_2 & u_2 v_3 \\ u_3 v_1 & u_3 v_2 & u_3 v_3 \end{pmatrix}$$

Laplacian of Gaussians

- The image Laplacian is susceptible to noise.
- Hence the image f is first convovled with a Gaussian of std. dev. σ to smooth some of the noise, and then the Laplacian is applied, i.e. we compute: $\nabla^2 (g_\sigma * f)$.
- Equivalent to $(\nabla^2 g_\sigma) * f$, due to associative and commutative nature of convolution.

Laplacian of Gaussians

- For efficiency, the Laplacian of Gaussians mask is pre-computed.

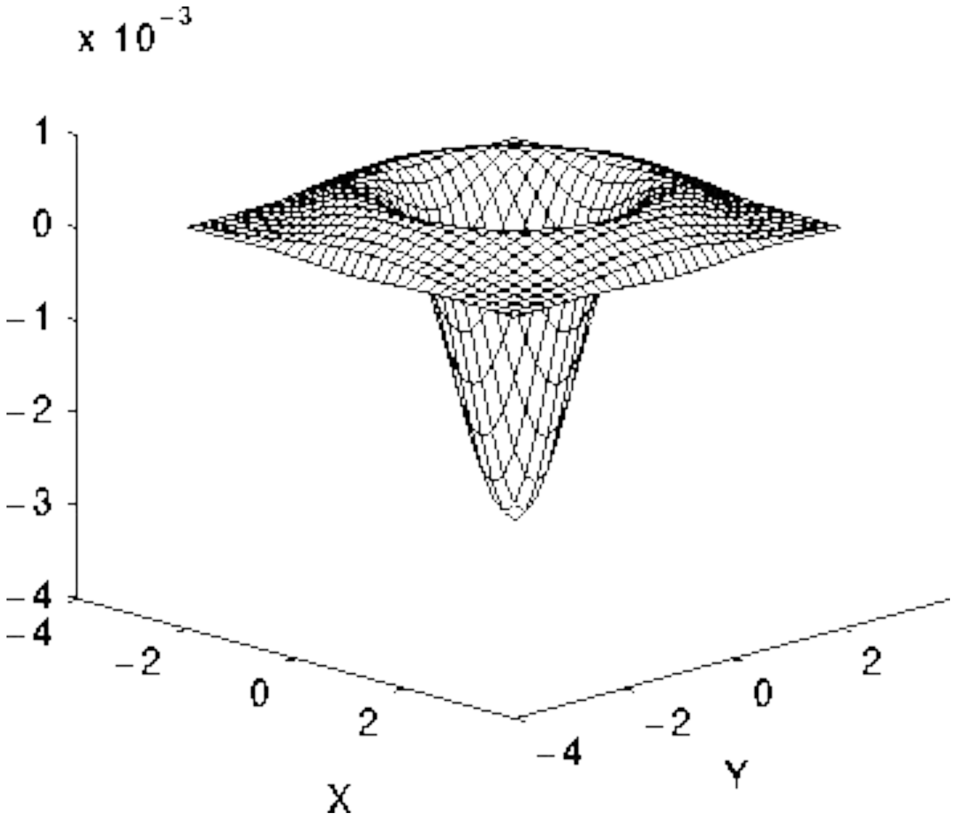
- The formula for this is:

$$g_\sigma(x, y) = \frac{\exp(-(x^2 + y^2)/(2\sigma^2))}{2\pi\sigma^2}$$

$$\frac{\partial^2 g_\sigma(x, y)}{\partial x^2} = \frac{1}{2\pi\sigma^4}\left(\frac{x^2}{\sigma^2} - 1\right)\exp(-(x^2 + y^2)/(2\sigma^2))$$

$$\frac{\partial^2 g_\sigma(x, y)}{\partial y^2} = \frac{1}{2\pi\sigma^4}\left(\frac{y^2}{\sigma^2} - 1\right)\exp(-(x^2 + y^2)/(2\sigma^2))$$

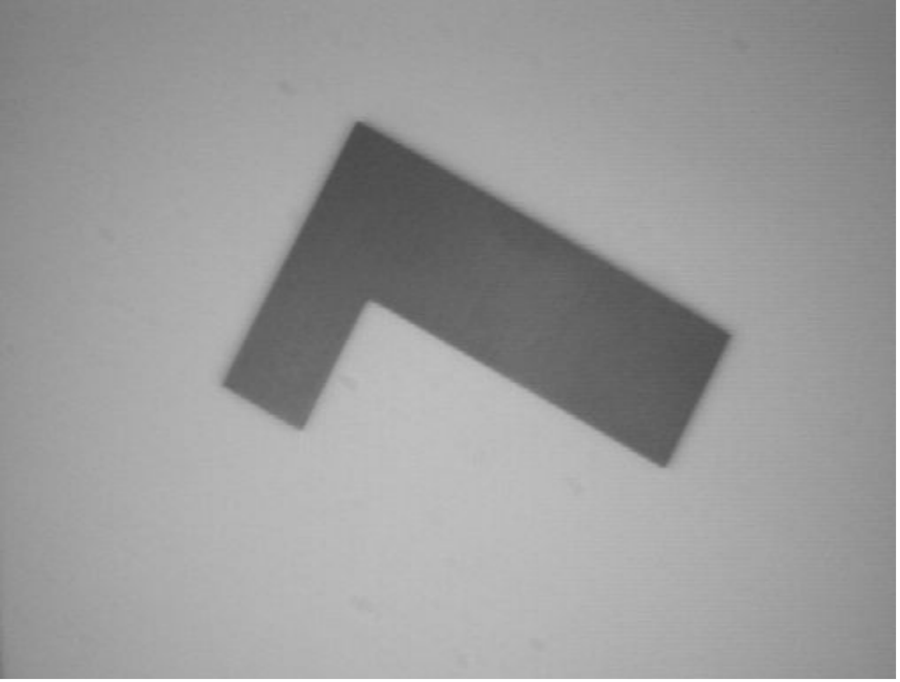
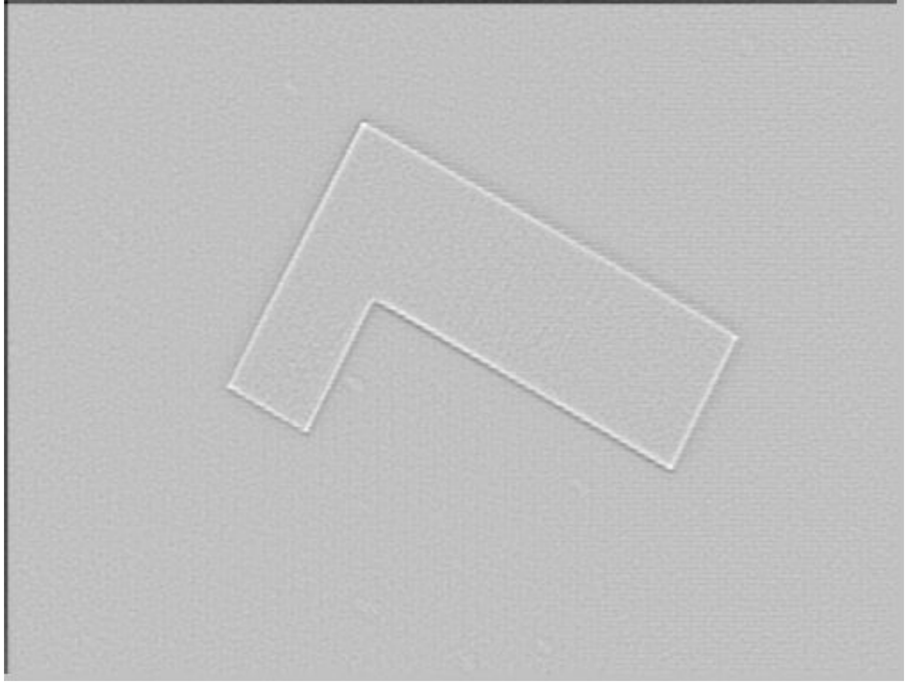
$$\therefore LOG_\sigma(x, y) = \nabla^2 g_\sigma(x, y) = \frac{\exp(-(x^2 + y^2)/(2\sigma^2))}{\pi\sigma^4}\left(\frac{x^2 + y^2}{\sigma^2} - 1\right)$$



0	1	1	2	2	2	1	1	0
1	2	4	5	5	5	4	2	1
1	4	5	3	0	3	5	4	1
2	5	3	-12	-24	-12	3	5	2
2	5	0	-24	-40	-24	0	5	2
2	5	3	-12	-24	-12	3	5	2
1	4	5	3	0	3	5	4	1
1	2	4	5	5	5	4	2	1
0	1	1	2	2	2	1	1	0

Discrete approximation with σ=1

https://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm

https://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm
LoG filter of size 7 x 7 with σ=1

Bilateral Filters

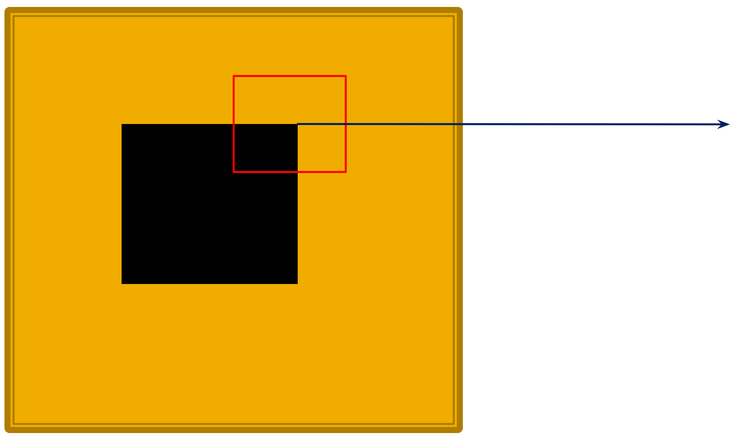
Bilateral filter versus mean/median filter

- Mean filter: does not preserve image edges
- Median filter: too may not always preserve edges
- To preserve edges well, we may need a non-linear filter
- Bilateral filter is one such

$\sigma = 4$ $\sigma = 8$ $\sigma = 16$ $\sigma = 32$

Gaussian blur

https://people.csail.mit.edu/sparis/bf_course/course_notes.pdf



The median filter will assign the median of the values in the red neighborhood to the corner pixel – and will convert the otherwise black pixel to a yellow one (as most of the pixels in the neighborhood are yellow). Thus the median filter does not always preserve edges well!

Bilateral Filter

$$BF[I]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in \mathcal{S}} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|)\, G_{\sigma_r}(I_{\mathbf{p}} - I_{\mathbf{q}})\, I_{\mathbf{q}}$$

$$W_{\mathbf{p}} = \sum_{\mathbf{q} \in \mathcal{S}} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|)\, G_{\sigma_r}(I_{\mathbf{p}} - I_{\mathbf{q}})$$

Intensity-based weights

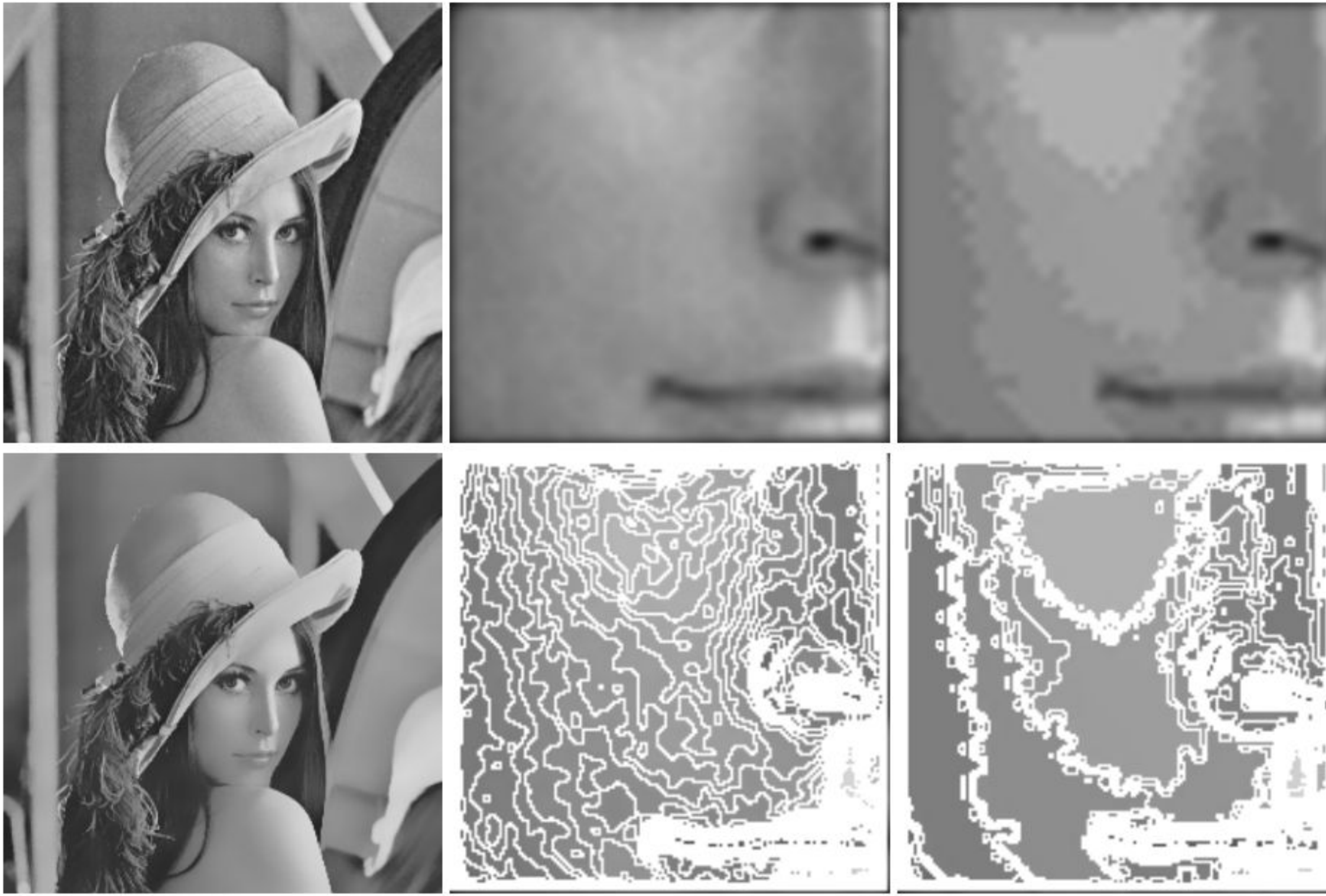
Space-based weights

Effect of parameter σ_s:

- As it increases, a larger and larger neighborhood of values around **p** =(x,y) will contribute to the averaging (more noise reduction but possible contribution from dissimilar regions)
- As it shrinks towards 0, fewer and fewer neighbors of the central pixel **p** will contribute to the averaging

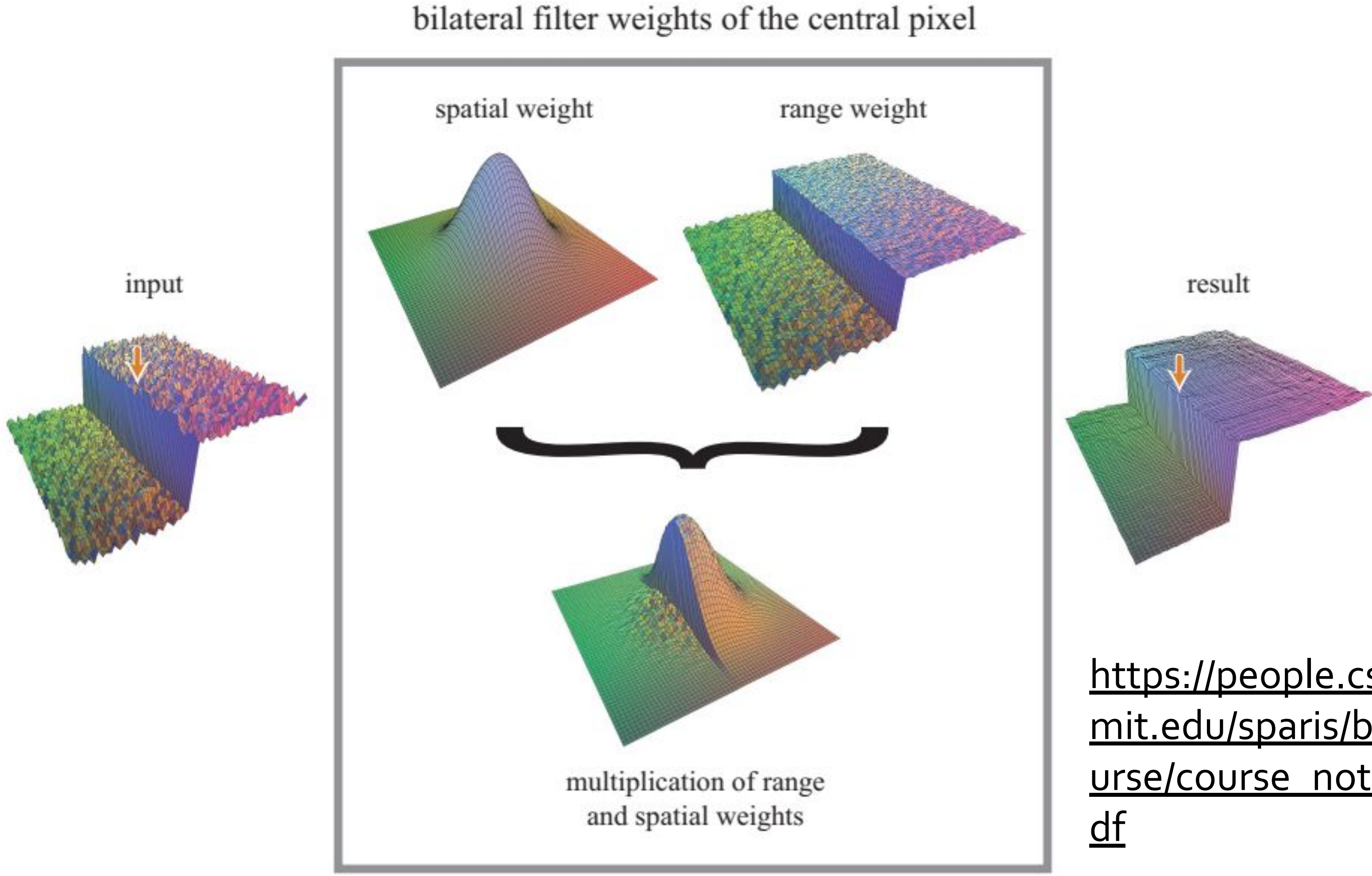
Effect of parameter σ_r:

- For moderate values, only intensities close to I(p) will affect the averaging
- For larger values, the bilateral filter will begin to resemble a Gaussian filter
- Features or edges with intensity difference less than $\boldsymbol{\sigma_r}$ will be blurred, others will be preserved



Bilateral filter result　　Zoom on original　　Zoom on result

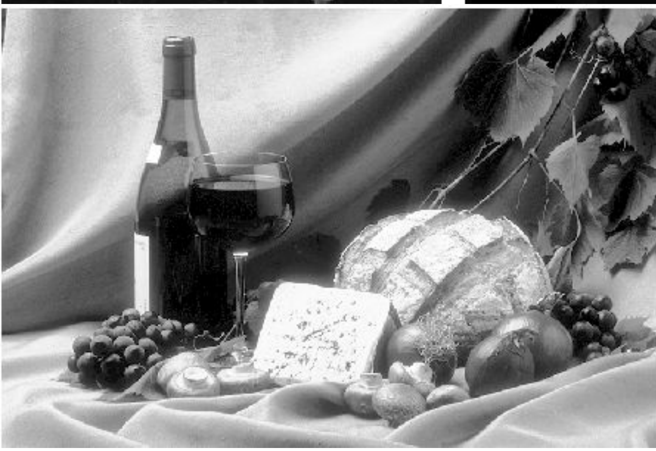
https://people.csail.mit.edu/sparis/bf_course/course_notes.pdf



https://people.csail.mit.edu/sparis/bf_course/course_notes.pdf

Figure 5: The bilateral filter smooths an input image while preserving its edges. Each pixel is replaced by a weighted average of its neighbors. Each neighbor is weighted by a spatial component that penalizes distant pixels and range component that penalizes pixels with a different intensity. The combination of both components ensures that only nearby similar pixels contribute to the final result. The weights are represented for the central pixel (under the arrow). The figure is reproduced from: Fast bilateral filtering for the display of high-dynamic-range images *Durand and Dorsey* ACM SIGGRAPH conference (c) 2002, Association for Computing Machinery, Inc. Reprinted by permission. `http://doi.acm.org/10.1145/566570.566574`


https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MANDUCHI1/Bilateral_Filtering.html





Bilateral filter implementation

- A bilateral filter cannot be implemented as a convolution for two reasons.
- The weights themselves depend on the intensity values of the original (noisy) image.
- The weights change from pixel to pixel.
- Thus the filter is neither linear nor space-invariant

Cross-bilateral filter

- Consider a noisy no-flash and a non-noisy flash image of the same scene.
- The latter has changes of color in the scene due to the flash
- Cross-bilateral filter: apply a bilateral filter to the no-flash image with intensity/range weights from the flash image

Orig. (top) Detail Transfer (bottom) Flash No-Flash Detail Transfer with Denoising

Figure 1: This candlelit setting from the wine cave of a castle is difficult to photograph due to its low light nature. A flash image captures the high-frequency texture and detail, but changes the overall scene appearance to cold and gray. The no-flash image captures the overall appearance of the warm candlelight, but is very noisy. We use the detail information from the flash image to both reduce noise in the no-flash image and sharpen its detail. Note the smooth appearance of the brown leather sofa and crisp detail of the bottles. For full-sized images, please see the supplemental DVD or the project website http://research.microsoft.com/projects/FlashNoFlash.

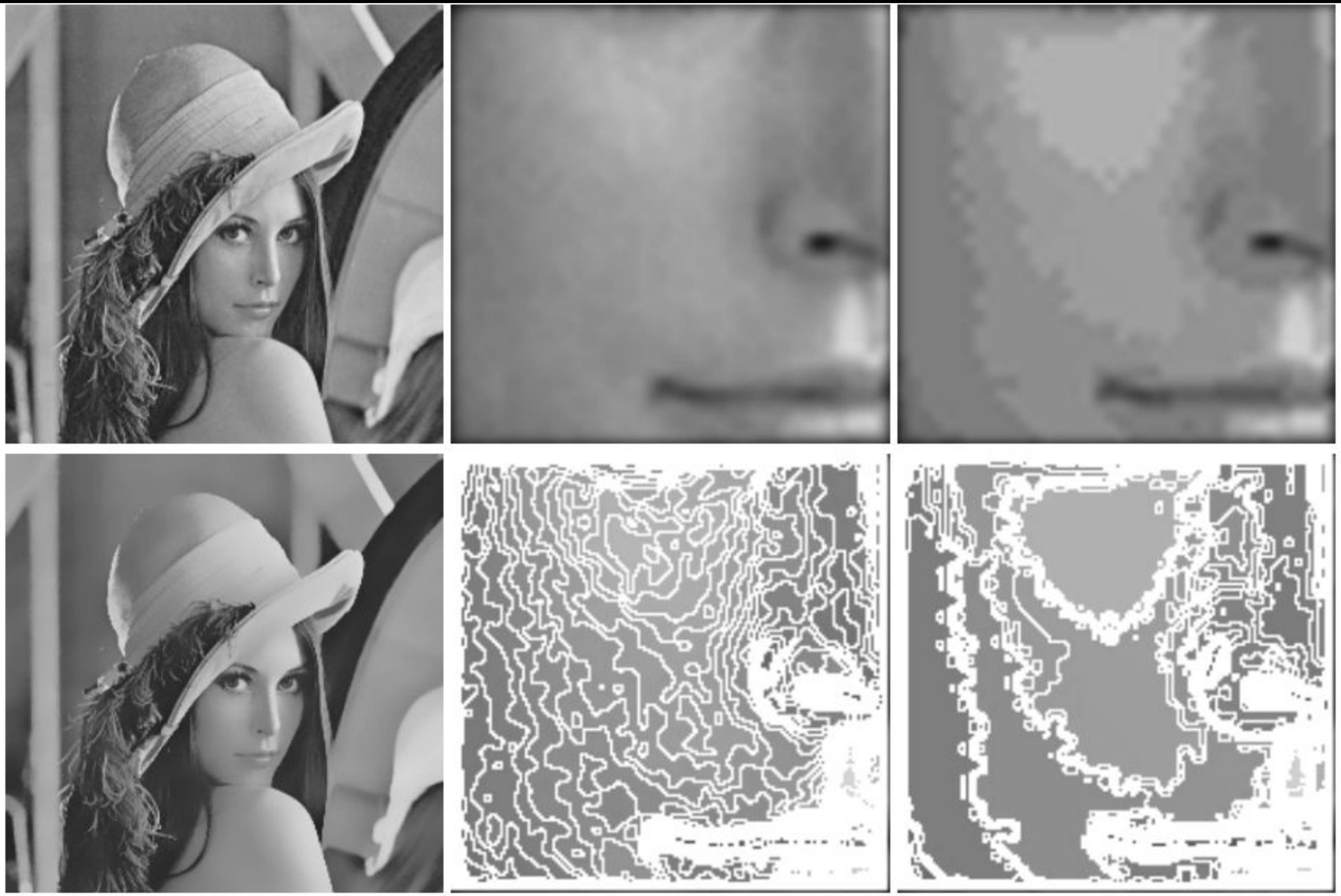
http://hhoppe.com/flash.pdf

Orig. (top) Detail Transfer (bottom) Flash No-Flash Detail Transfer with Denoising

Figure 6: An old European lamp made of hay. The flash image captures detail, but is gray and flat. The no-flash image captures the warm illumination of the lamp, but is noisy and lacks the fine detail of the hay. With detail transfer and denoising we maintain the warm appearance, as well as the sharp detail.

http://hhoppe.com/flash.pdf

Staircase effect of the bilateral filter



Bilateral filter result Zoom on original Zoom on result

https://people.csail.mit.edu/sparis/bf_course/course_notes.pdf

Staircase effect of the bilateral filter

- The bilateral filter has a tendency to create artificial edges in homogenous regions.
- Why? Look at a 1D example – next slide.
- The signal is locally concave (line joining two points is above the curve).

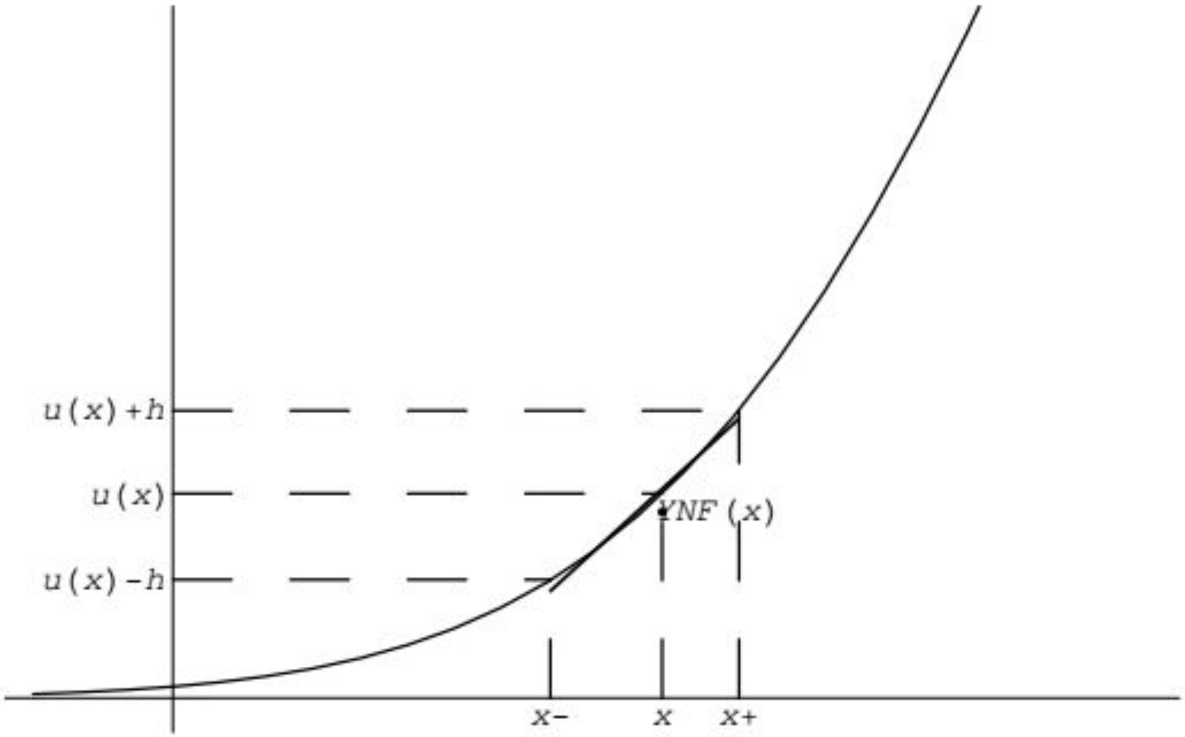


Fig. 4. Illustration of the shock effect of the YNF on a concave signal. The number of points y satisfying $u(x) - h < u(y) \leq u(x)$ is larger than the number satisfying $u(x) \leq u(y) < u(x) + h$. Thus, the average value $YNF(x)$ is smaller than $u(x)$, enhancing that part of the signal. The regression line of u inside $(x-, x+)$ better approximates the signal at x.

https://hal.archives-ouvertes.fr/hal-00271143/document

Staircase effect of the bilateral filter

- We are looking at pixels (y) whose intensities are within a range of u(x)-h to u(x)+h where h is approximately $3\sigma_r$.
- But the number of points such that u(x)-h < u(y) <= u(x) is more than the number of points for which u(x) < u(y) <= u(x)+h.
- So effectively, the filtered intensity value is less than the original intensity value u(x).
- For a locally convex signal (line joining two points is below the curve), the filtered intensity value is more than the original value u(x).
- At points where convex and concave parts meet (called inflection points), you get an artificial increase in contrast – causing the staircase effect.

Bokeh Filters

Bokeh filters

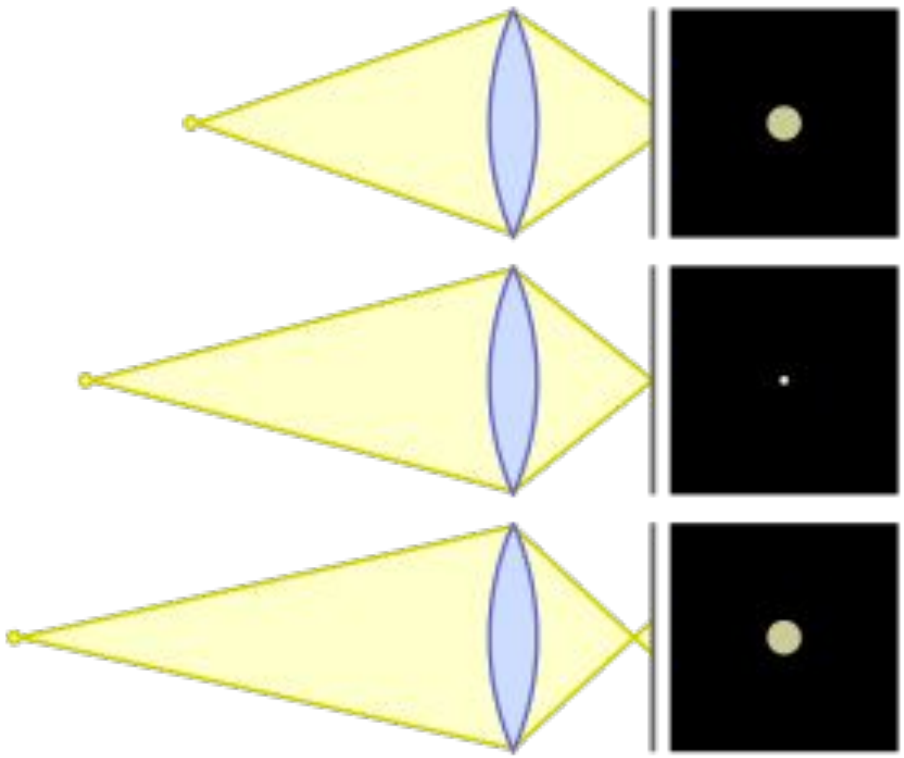
- Bokeh is an effect in photography where the main object is in focus and the rest of the scene is severely defocussed.
- This produces a pleasing aesthetic effect.

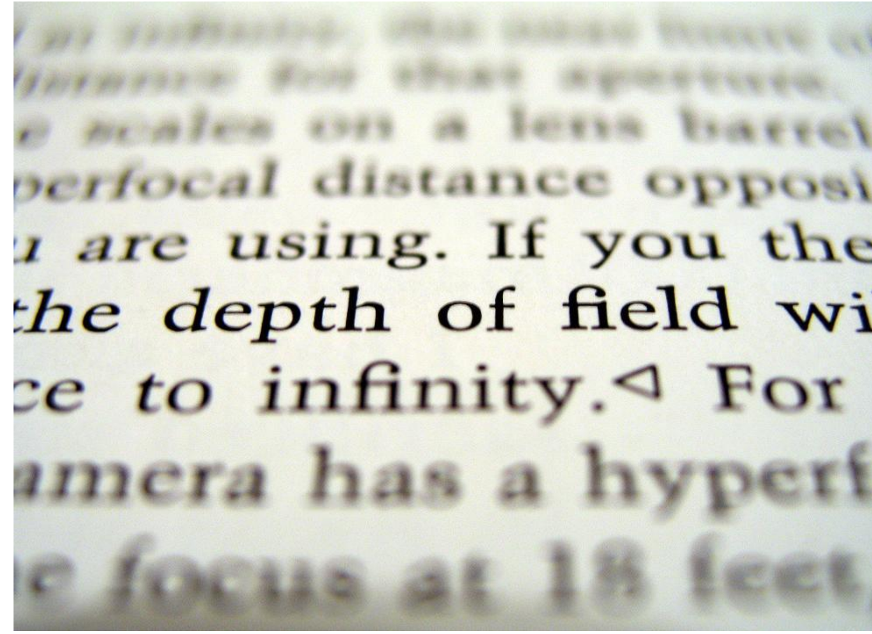
https://en.wikipedia.org/wiki/Bokeh#/media/File:Josefina_with_Bokeh.jpg

https://en.wikipedia.org/wiki/Bokeh

Bokeh

- The amount of blur is depth dependent.
- Certain areas are in focus, the rest are not and lie in a “circle of confusion”.
- The depth of field of a camera is the region where the circle of confusion has a size less than the resolution of the human eye.





Shallow depth of field

https://en.wikipedia.org/wiki/Bokeh

Bokeh

- Bokeh characteristics may be quantified by examining the image's circle of confusion. In out-of-focus areas, each point of light becomes an image of the aperture, generally a more or less round disc.
- Bokeh can be simulated by convolving the image with a kernel that corresponds to the image of an out-of-focus point source taken with a real camera.
- Unlike conventional convolution, this convolution has a kernel that depends on the distance of each image point and – at least in principle – has to include image points that are occluded by objects in the foreground.[28]
- Also, bokeh is not just any blur. To a first approximation, defocus blur is convolution by a uniform disk, a more computationally intensive operation than the "standard" Gaussian blur; the former produces sharp circles around highlights whereas the latter is a much softer effect.

From left to right: an original photo with no bokeh or blur; the same photo with synthetic bokeh effect applied to its background; the same photo with Gaussian blur applied to its background

https://en.wikipedia.org/wiki/Bokeh

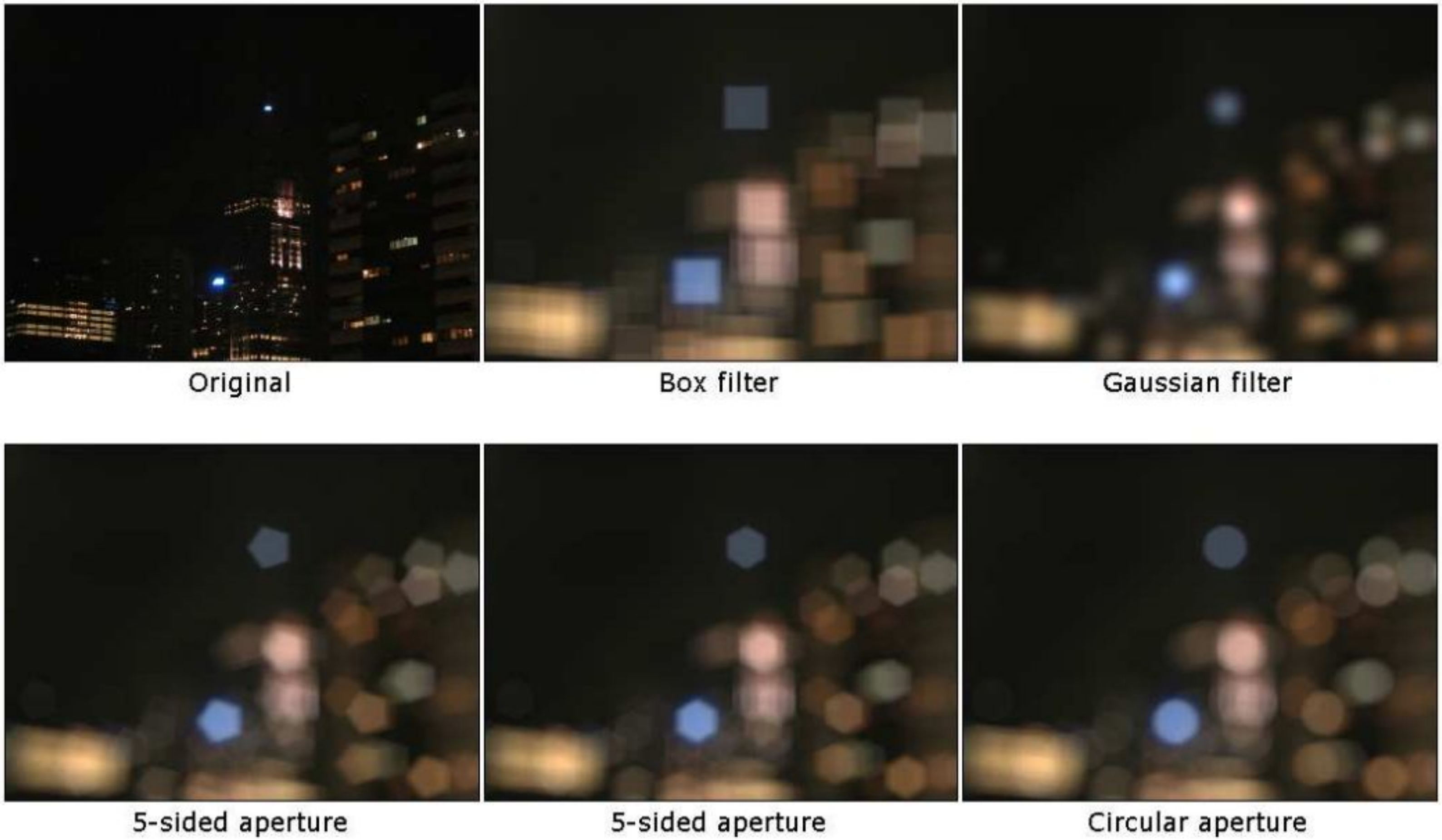


Fig. 5: Low-rank filtering results and the ground truth 2d convolution for the 8-sided polygonal aperture.