Image Derivatives

Introduction

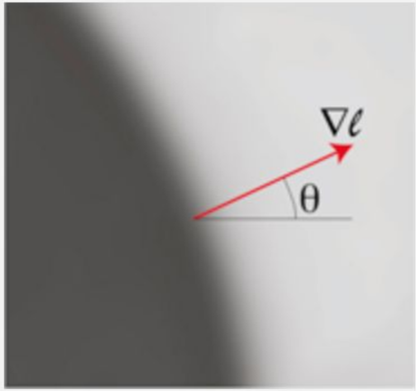
1. “Computing image derivatives is an essential operation for extracting useful information from images.”
2.

Derivative - discrete approximation

1.

$$\nabla \ell = \left(\frac{\partial \ell}{\partial x}, \frac{\partial \ell}{\partial y} \right)$$

Gradient of an image at one location:



$$\frac{\partial \ell}{\partial x} \simeq \ell(x, y) - \ell(x-1, y) \qquad (2.5)$$

$$\frac{\partial \ell}{\partial y} \simeq \ell(x, y) - \ell(x, y-1) \qquad (2.6)$$

Derivative - discrete approximation

1. “The derivative operator is linear and translation invariant. Therefore, it can be written as a convolution.”

$$\ell(x, y) \longrightarrow \boxed{\left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)} \longrightarrow \left(\ell_x(x, y), \ell_y(x, y)\right)$$

Figure 18.1: Computing image derivatives along x- and y-dimensions.

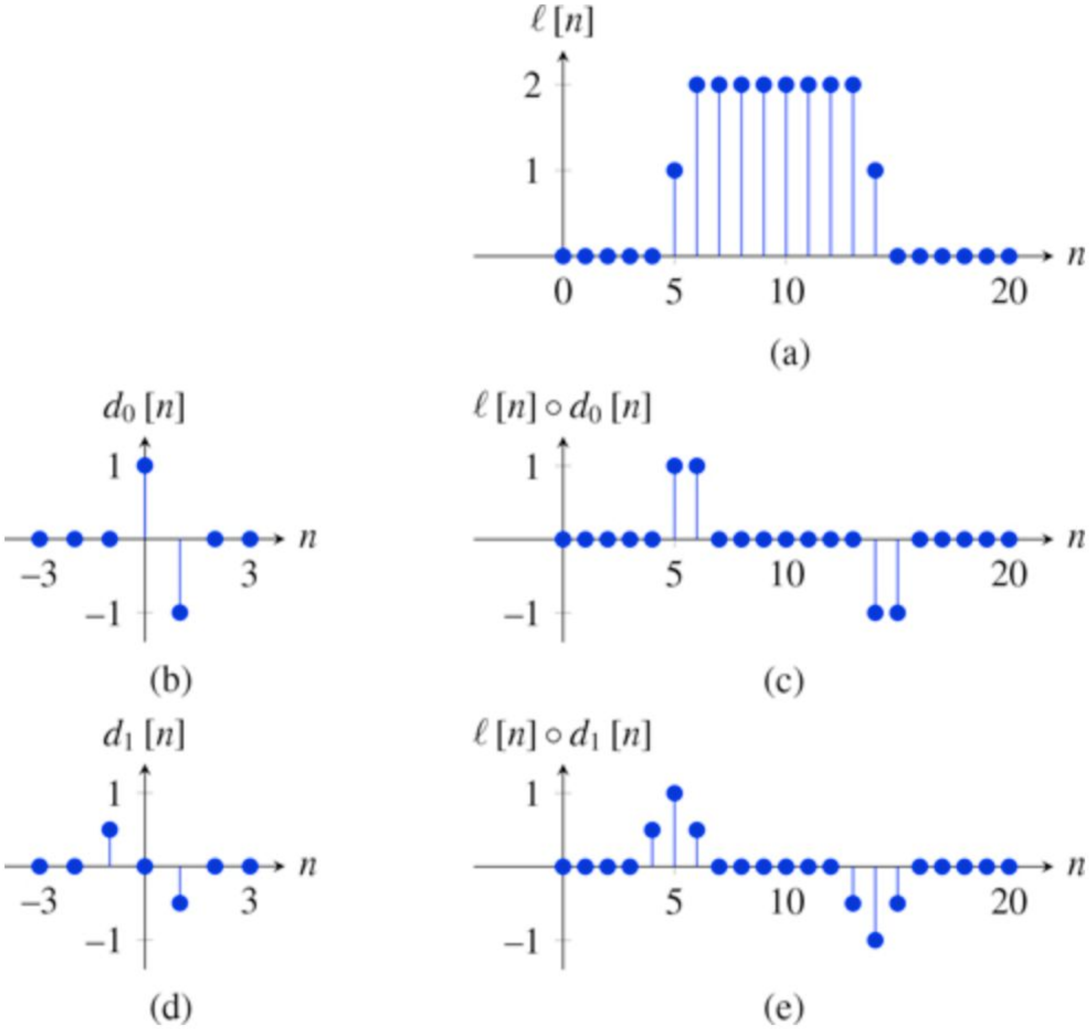
Derivative - discrete approximation

1. 1 D derivative operators
2. $d_0 = [1, -1]$

$$\ell \circ d_0 = \ell[n] - \ell[n-1]$$

3. "$d1 = [1, 0, -1]/2$"

$$\ell \circ d_1 = \frac{\ell[n+1] - \ell[n-1]}{2}$$


ℓ [n]
2
1
0
5
10
20
n
(a)
d0 [n]
1
−3
3
−1
n
(b)
ℓ [n] ∘ d0 [n]
1
5
10
20
−1
n
(c)
d1 [n]
1
−3
3
−1
n
(d)
ℓ [n] ∘ d1 [n]
1
5
10
20
−1
n
(e)

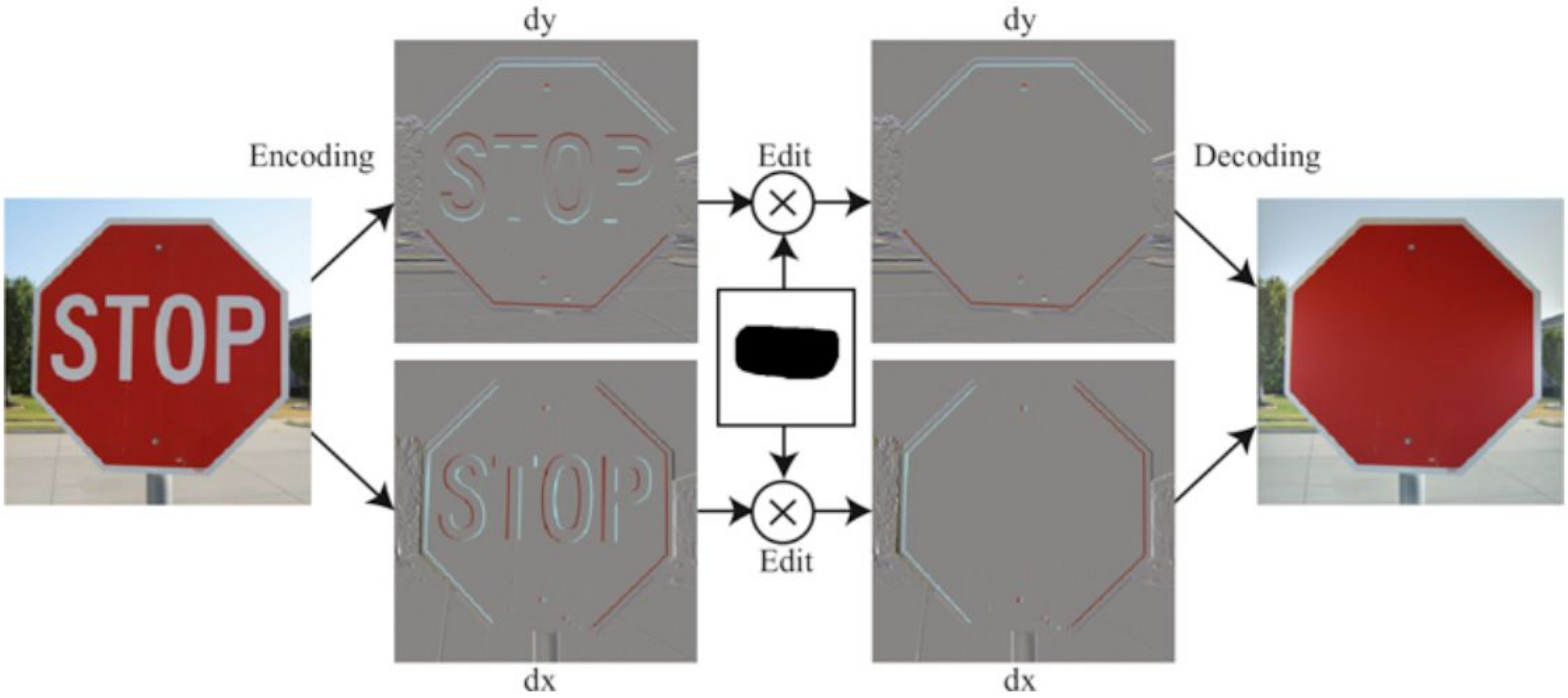


Figure 18.5: Image inpainting: Using image derivatives we delete the word "stop" by setting to zero the gradients indicated by the mask. The resulting decoded image propagates the red color inside the region that contained the word.

Gaussian Derivatives

1. “derivatives are sensitive to noise”

2.

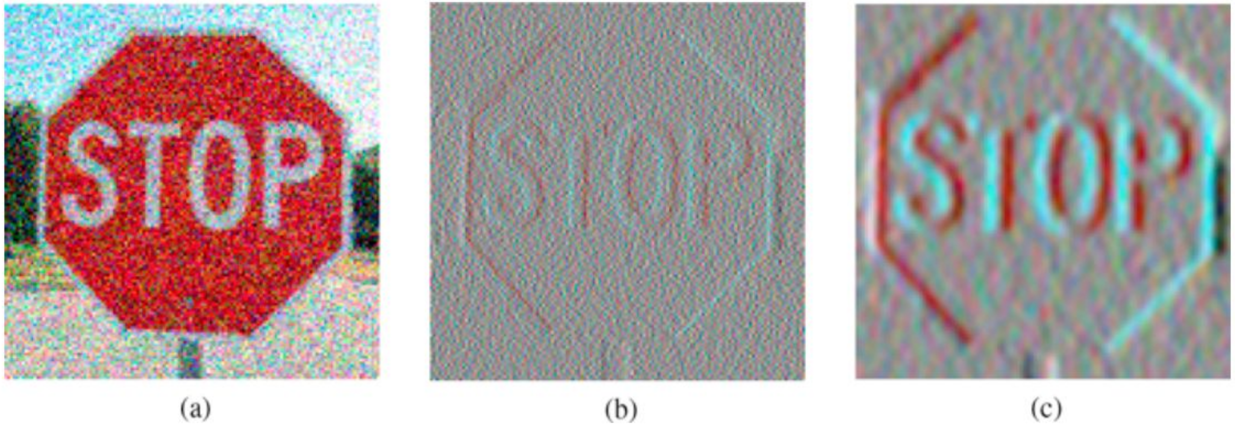


Figure 18.6: Derivatives of a noisy image. (a) Input image with noise, (b) its x-derivative obtained by convolving with a kernel [1, −1], and (c) its x-derivative obtained using a gaussian derivative kernel.

Gaussian derivatives discrete approximations

d_0						1		–1					
d_1					1		0		–1				
d_2				1		1		–1		–1			
d_3			1		2		0		–2		–1		
d_4		1		3		2		–2		–3		–1	
d_5	1		4		5		0		–5		–4		–1

Figure 18.13: Derivative of binomial coefficients resulting from the convolution $b_n \circ [1, -1]$. The filters, d_0 and d_1, are the ones we have studied in the previous section.

Sobel Operator

In two dimensions, we can use separable filters and build a partial derivative as

$$Sobel_x = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} \circ \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \tag{18.21}$$

$$Sobel_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{18.22}$$

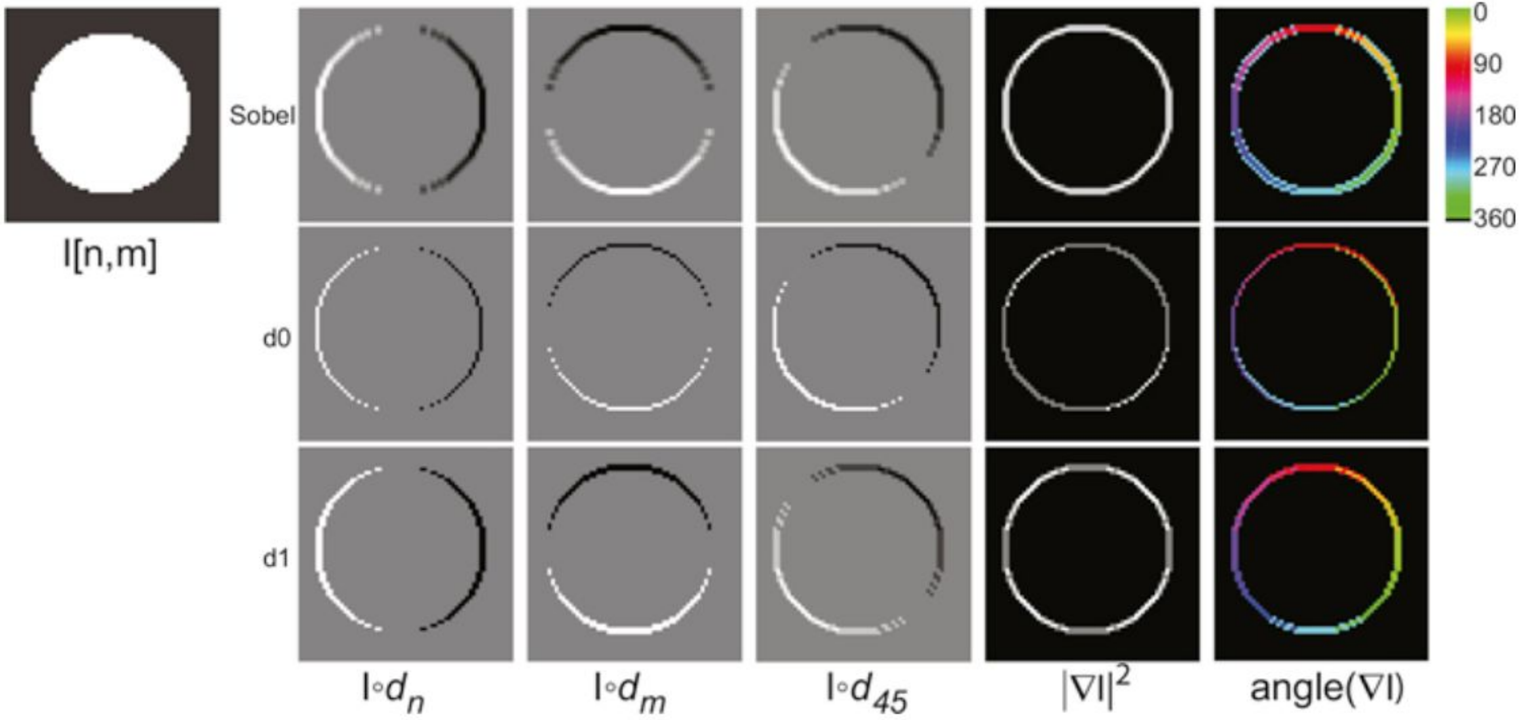


Figure 18.15: Derivatives of a circle along the directions n, m, and 45 degrees. The angle is shown only where the magnitude is > 0. The derivative output along 45 degrees is obtained as a linear combination of the derivatives outputs along n and m. Check the differences among the different kernels. The Sobel operator gives the most rotationally invariant gradient magnitude, but it is blurrier.

Laplacian Filter

1. The Laplacian operator is defined as the sum of the second order partial derivatives of a function

$$\nabla^2 \ell = \frac{\partial^2 \ell}{\partial x^2} + \frac{\partial^2 \ell}{\partial y^2} \qquad (18.31)$$

2. Laplacian is more sensitive to noise than the first order derivative. Therefore, in the presence of noise, when computing the Laplacian operator on an image $\ell(x, y)$, it is useful to smooth the output with a Gaussian kernel, $g(x, y)$

Laplacian Filter

1. In one dimension, the Laplacian can be approximated by [1, –2, 1],
   a. which is the result of the convolution of two 2-tap discrete approximations of the derivative [1, –1] ◦ [1, –1].
2. In two dimensions, the most popular approximation is the five-point formula, which consists in convolving the image with the kernel:
   a. Sum separable of 1D and its transpose

$$\nabla_5^2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

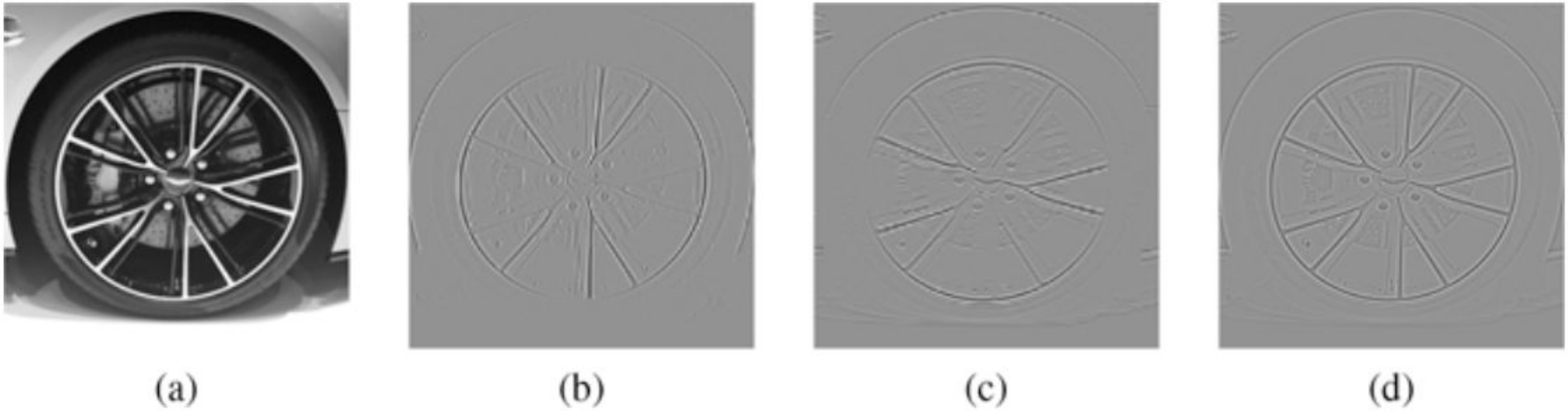


Figure 18.18: (a) Input image. (b) Second order derivative along x. (c) Second-order derivative along y. (d) The sum of (b) + (c), which gives the Laplacian.

Laplacian Filter

1. It is rotationally invariant. It is a linear operator that responds equally to edges in any orientation.
2. It measures curvature. If the image contains a linear trend the derivative will be non-zero despite having no boundaries, while the Laplacian will be zero.
3. Gradient-based detectors are good at detecting any change in intensity, including linear trends.
4. The Laplacian is better at detecting abrupt changes, which are more indicative of **object boundaries**.

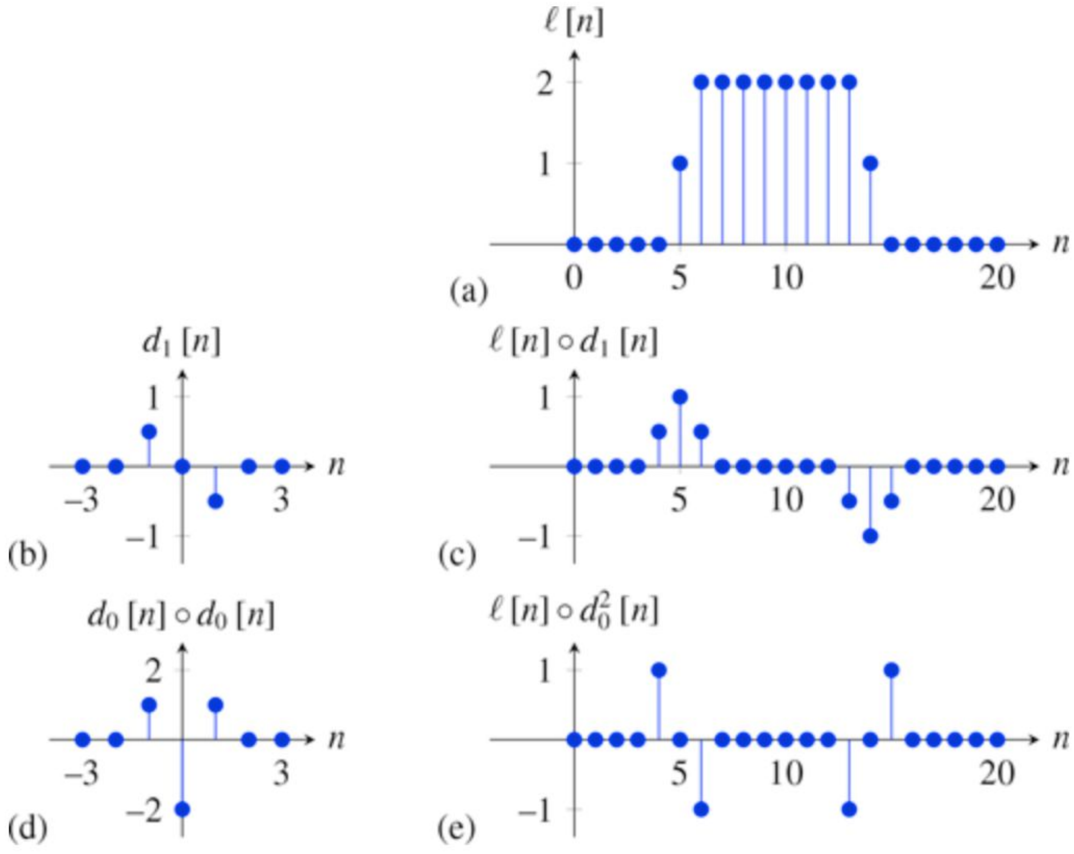


Figure 18.19: Comparison between the output of a first-order derivative and the Laplacian of 1D signal. (a) Input signal. (b) Kernel d_1. (c) Output of the derivative, that is, convolution of (a) and (b). (d) Discrete approximation of the Laplacian. (e) Output of convolving the signal (a) with the Laplacian kernel (d).

Laplacian Filter

1. Laplacian is not used now a days for edge detection
2. It is used to build Image Pyramids (multiscale representations)
3. To detect points of interest in images
   a. It is the basic operator used to detect keypoints to compute SIFT descriptors

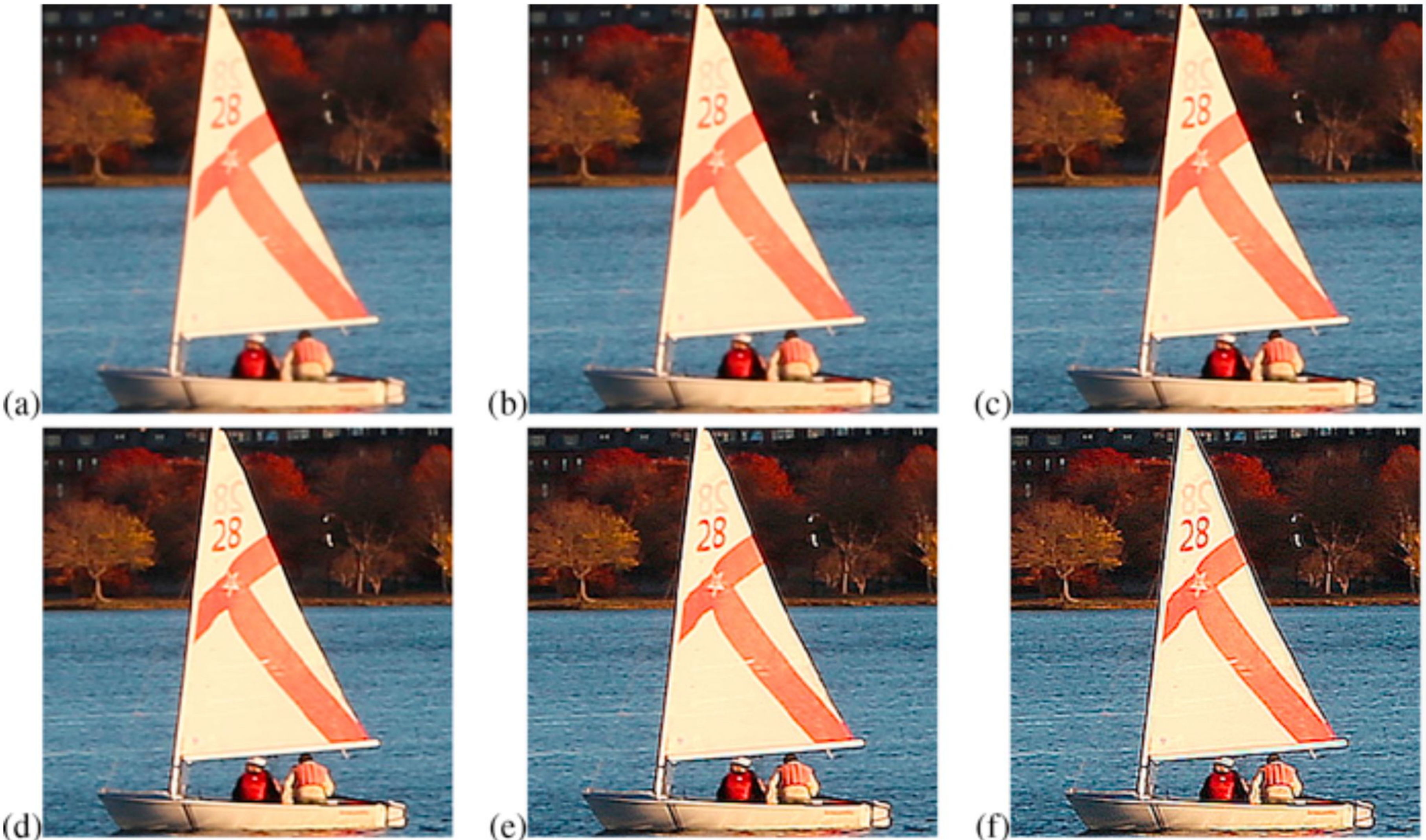
Sharpening Filter

1. "The goal of a sharpening filter is to transform an image so that it appears sharper (i.e., it contains more fine details)"
2. "This can be achieved by amplifying the amplitude of the high-spatial frequency content of the image."
3. "we start with twice the original image (sharp plus blurred parts), then subtract away the blurred components of the image"

$$\text{sharpening filter} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{18.36}$$

Sharpening Filter

1. We can apply this filter successively in order to further enhance the image details.
2. If too much sharpening is applied we might end up enhancing noise and introducing image artifacts.
3.


(a)
(b)
(c)
(d)
(e)
(f)

Retinex

1. “How do you tell gray from white?”
2. “If two patches receive the same amount of light, then we will perceive as being darker the patch that reflects less light.”
3. “If we increase the amount of light projected on top of the dark patch, what will be seen now?”

$$\ell(x, y) = r(x, y) \times l(x, y) \tag{18.37}$$

4. “what happens if we see two patches of unknown reflectance, and each is illuminated with two different light sources of unknown identity?”

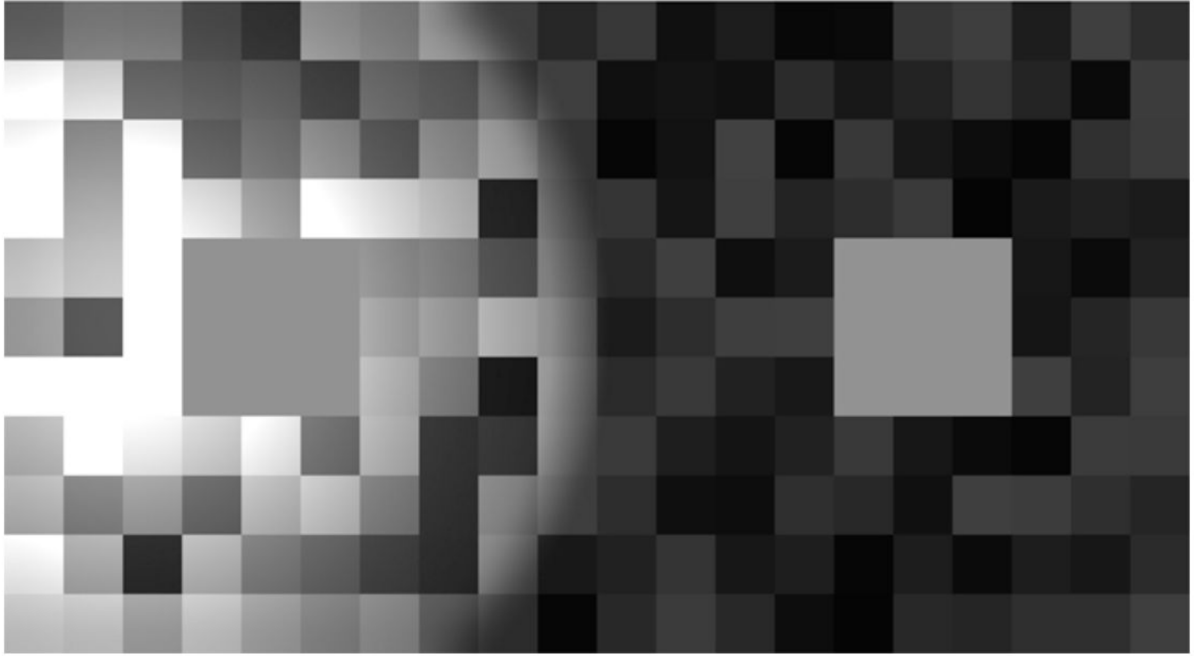


Figure 18.24: Simultaneous contrast illusion. What happens if we see two patches of unknown reflectance, and each is illuminated with two different light sources of unknown identity?

1. we discount (at least partially) the effects of the illumination, l(x, y)

Retinex

1. The difference between two adjacent pixels values inside one of the patches of uniform reflectance will be very small even if illumination is nonuniform.
2. If the two adjacent pixels are in the boundary between two patches of different reflectances, then the intensities of these two pixels will be very different

$$\log \ell(x, y) = \log r(x, y) + \log l(x, y) \tag{18.38}$$

$$\frac{\partial \log \ell(x, y)}{\partial x} = \frac{\partial \log r(x, y)}{\partial x} + \frac{\partial \log l(x, y)}{\partial_x} \tag{18.39}$$

$$\frac{\partial \log r(x, y)}{\partial x} = \begin{cases} \frac{\partial \log \ell(x,y)}{\partial x} & \text{if } \left|\frac{\partial \log \ell(x,y)}{\partial x}\right| > T \\ 0 & \text{otherwise} \end{cases} \tag{18.40}$$

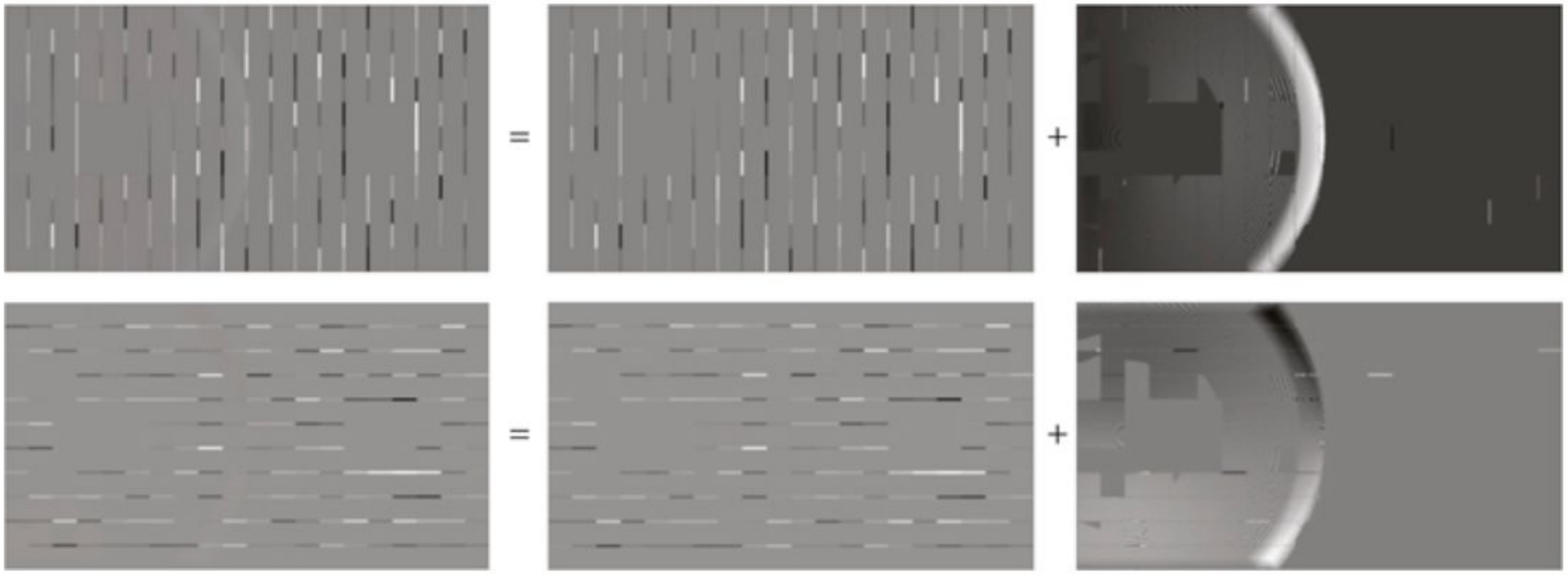


Figure 18.25: Derivatives classified into reflectance or luminance components.

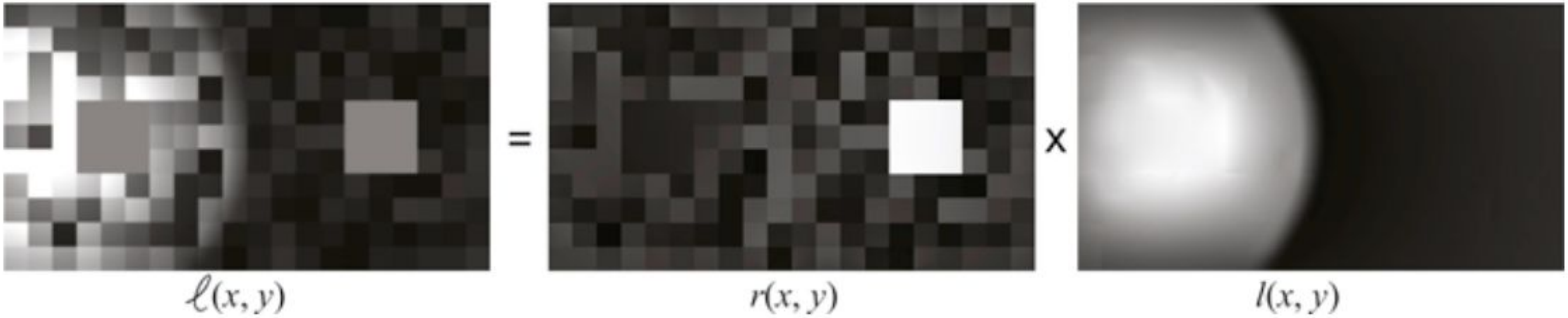


Figure 18.26: Recovered components. The estimated reflectance, $r(x, y)$, is close to what we perceive. It seems that what we perceive contains part of $l(x, y)$.

The underlying assumption is that the illumination image, l(x, y), varies smoothly and the reflectance image, r(x, y), is composed of uniform regions separated by sharp boundaries.

References

1. Foundations of Computer Vision - Chapter 18