



Image processing operators (2)

Bing Gong
(巩冰)

gongbing@shnu.edu.cn

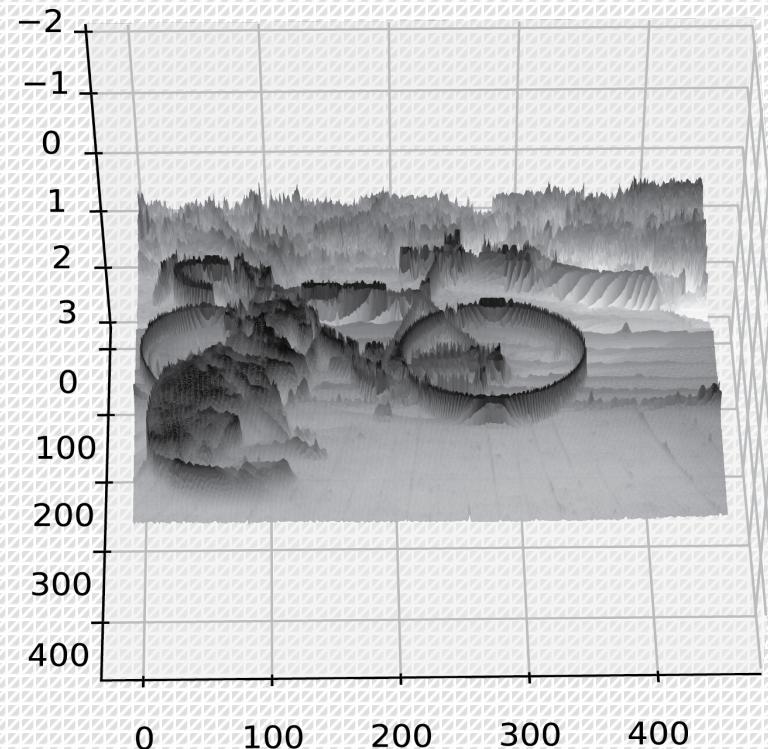
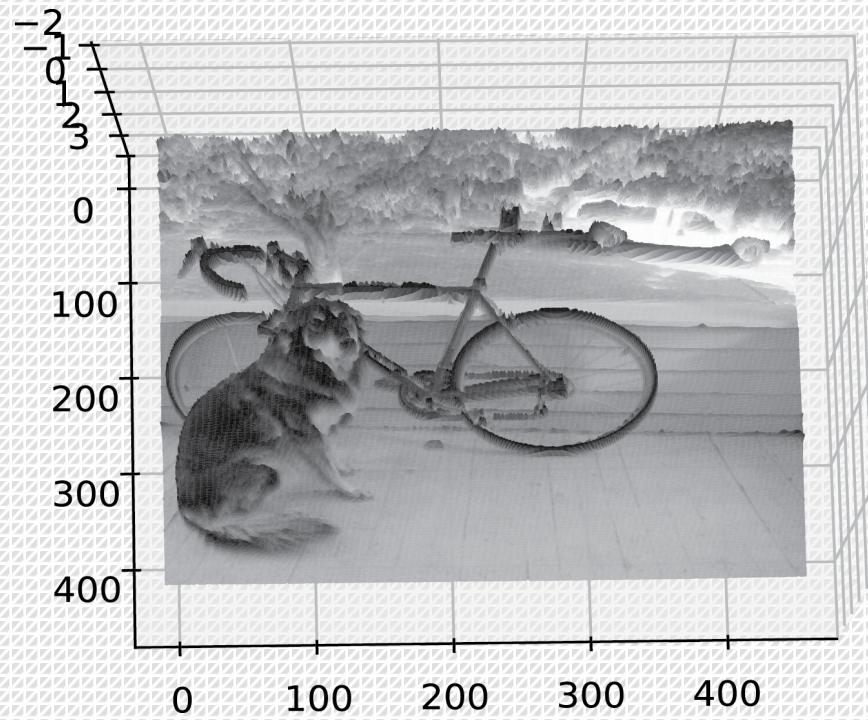


Edges

Source: <https://courses.cs.washington.edu/courses/cse576/23sp/notes.html>



- What is an “edge”?

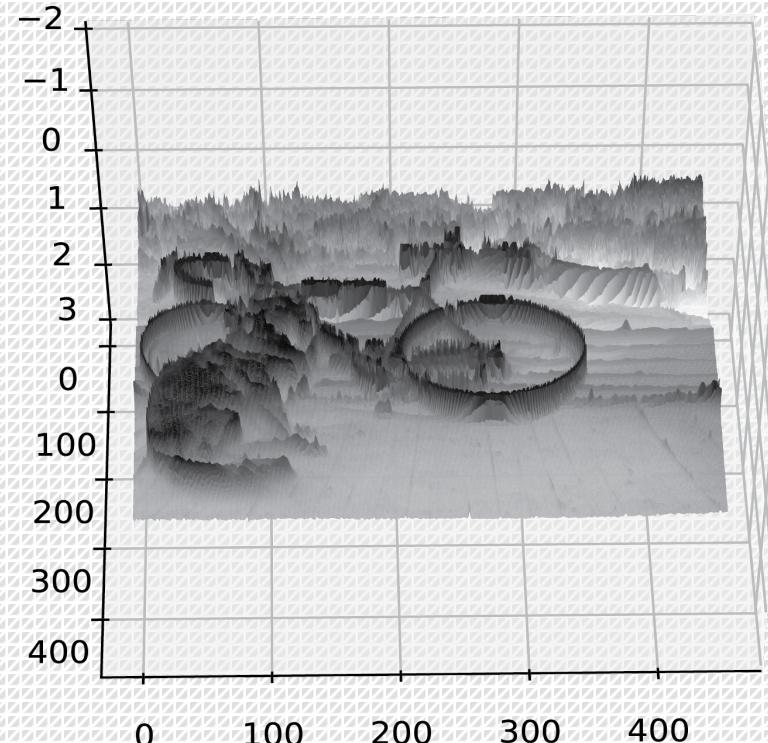
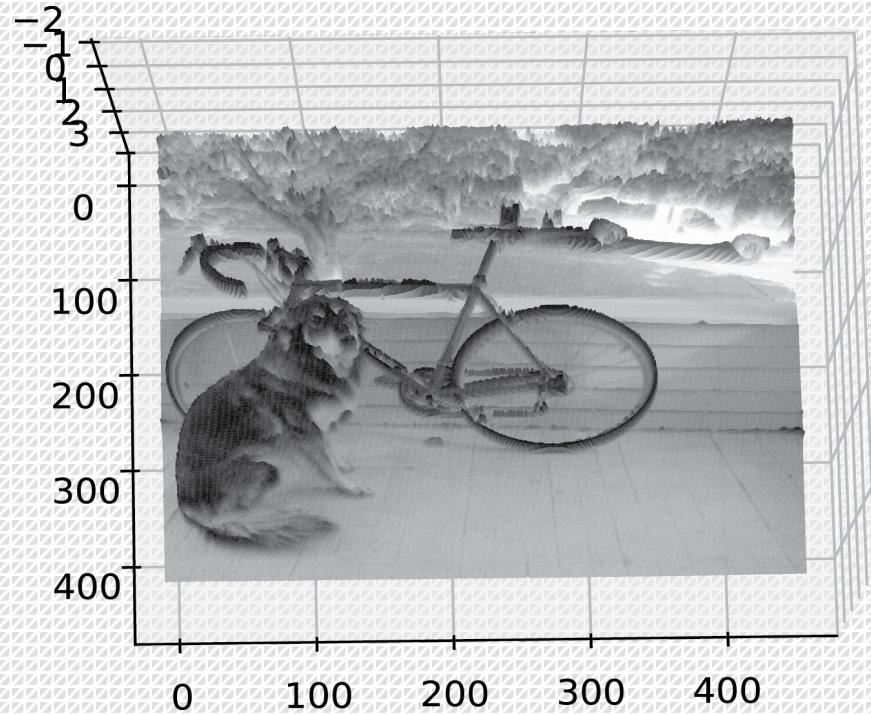


Edges

Source: <https://courses.cs.washington.edu/courses/cse576/23sp/notes.html>



- Image is a function.
- Think of the gray tones as HEIGHTS.
- Edges are rapid changes in this function

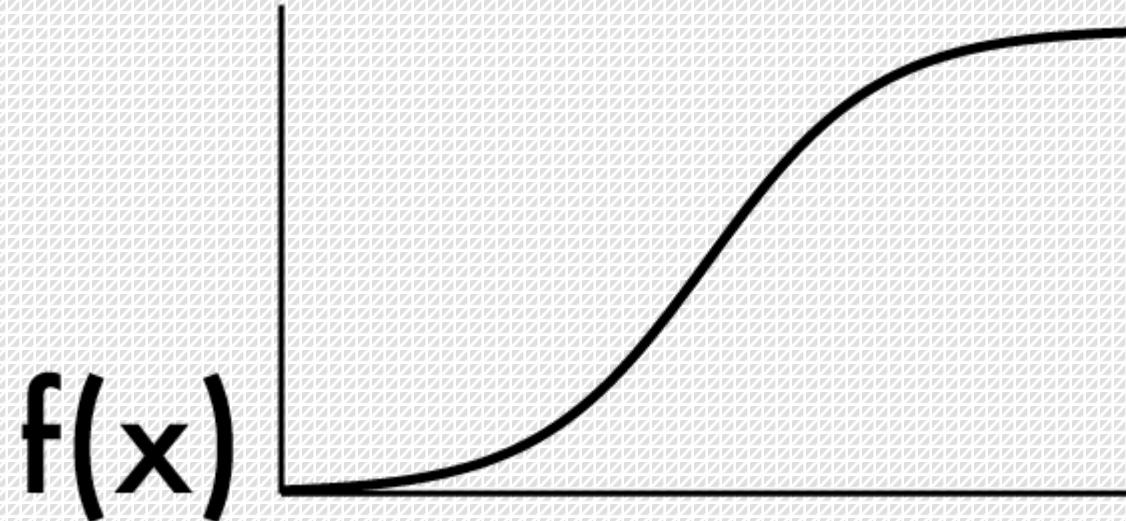


Edges

Source: <https://courses.cs.washington.edu/courses/cse576/23sp/notes.html>



- Image is a function
- Edges are rapid changes in this function



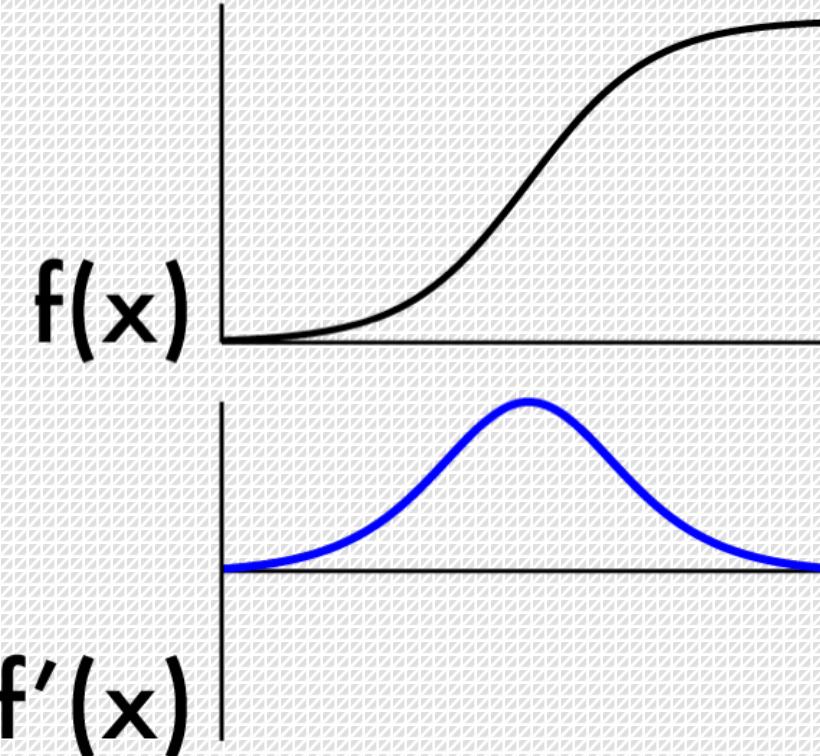


Edges

Source: <https://courses.cs.washington.edu/courses/cse576/23sp/notes.html>



- Image is a function
- Edges are rapid changes in this function



Edges

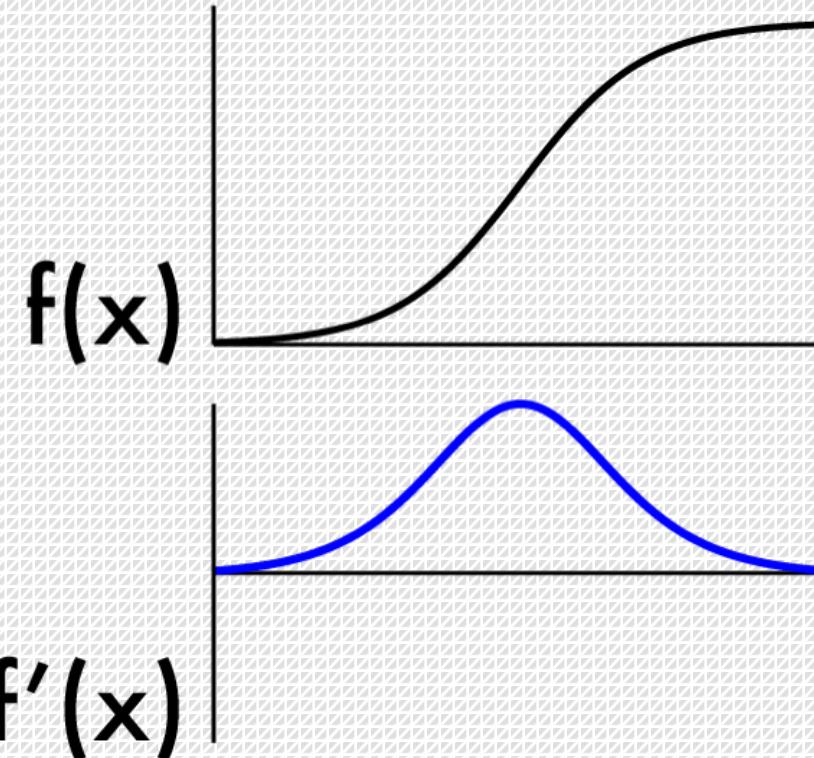
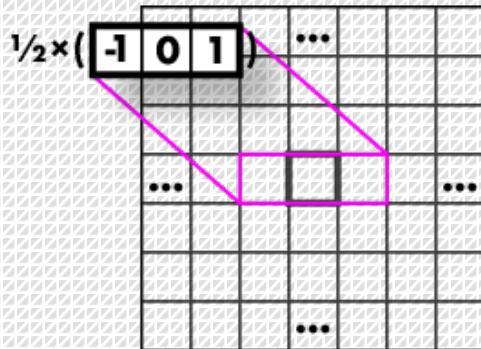
Source: <https://courses.cs.washington.edu/courses/cse576/23sp/notes.html>



- Recall:

$$f'(a) = \lim_{h \rightarrow 0} \frac{f(a+h) - f(a)}{h}.$$

- We don't have an "actual" function, must estimate
- Possibility: set $h = 2$
- What will that look like?

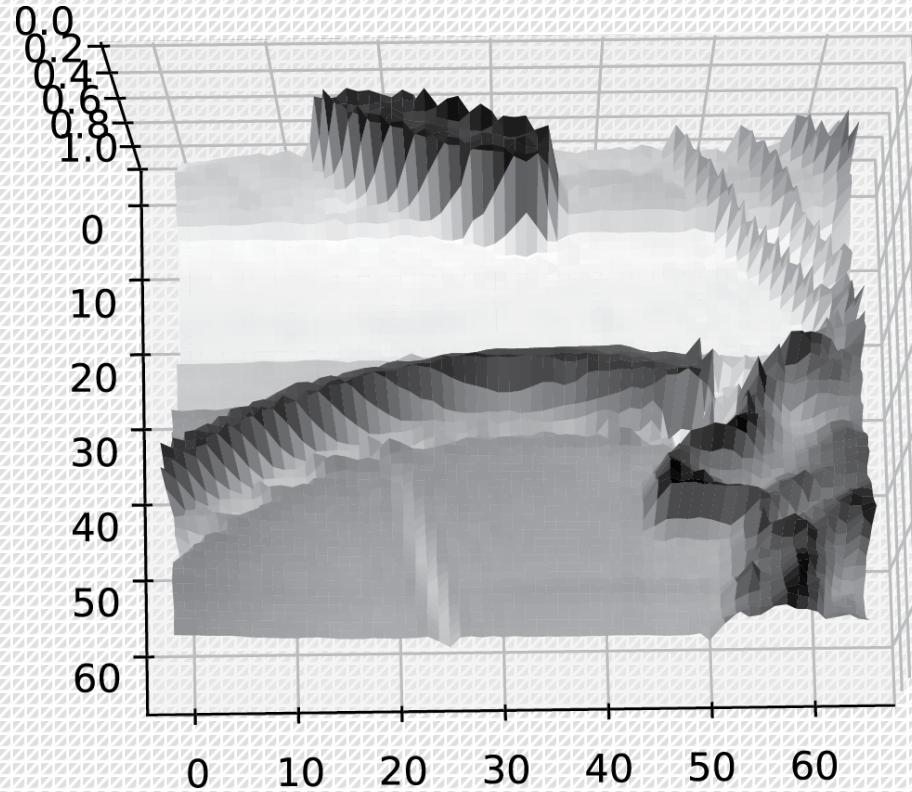
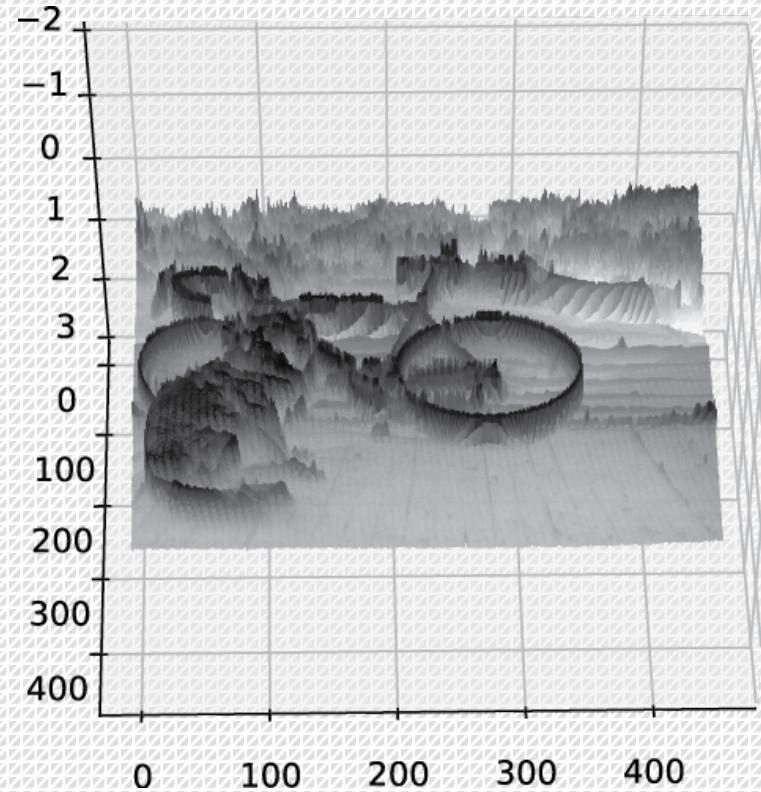


Edges

Source: <https://courses.cs.washington.edu/courses/cse576/23sp/notes.html>



- Images are noisy!

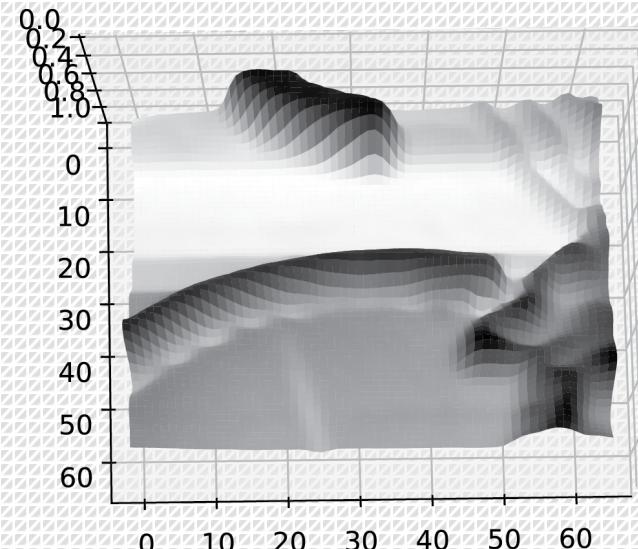
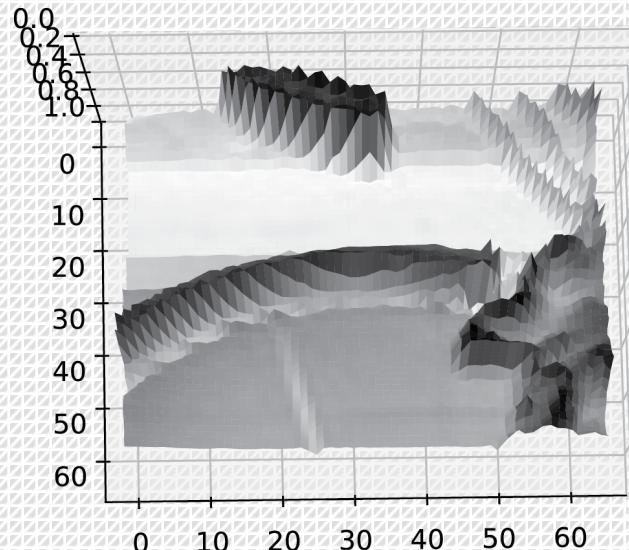
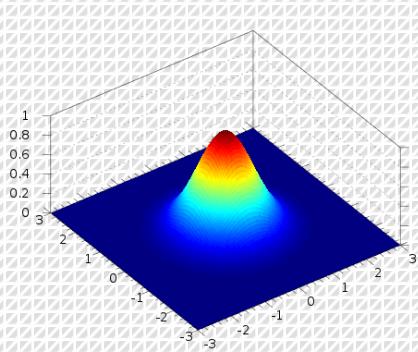


Edges

Source: <https://courses.cs.washington.edu/courses/cse576/23sp/notes.html>



- Images are noisy!



Edges

Source: <https://courses.cs.washington.edu/courses/cse576/23sp/notes.html>



- Images are noisy!

Diagram illustrating the convolution operation for edge detection:

Input image (G): A 3x3 grid with values 1, 2, 1; 2, 4, 2; 1, 2, 1. The middle value 4 is circled in red.

Kernel (K): A 1x3 filter with values -1, 0, 1. The last value 1 is highlighted in a red box.

Output: The result of the convolution is a 3x3 grid of zeros.

$$\frac{1}{2} \times (-1 \mid 0 \mid 1) * \left(\begin{array}{ccc} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{array} \right) = \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}$$

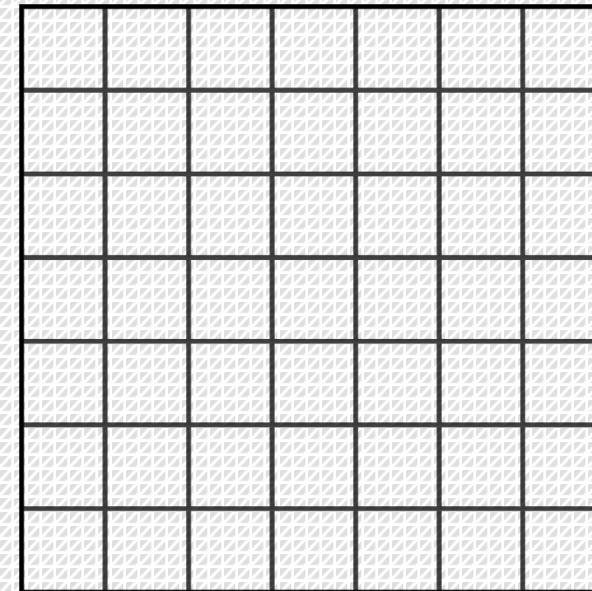
Edges

Source: <https://courses.cs.washington.edu/courses/cse576/23sp/notes.html>



- Smooth first, then derivative

$$\frac{1}{2} \times \left(\begin{bmatrix} -1 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \right) *$$



Edges

Source: <https://courses.cs.washington.edu/courses/cse576/23sp/notes.html>



- Smooth first, then derivative

$$\frac{1}{2} \times \left(\begin{bmatrix} -1 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \right)$$

The diagram illustrates the convolution operation. A 1x3 kernel $\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$ is applied to a 3x3 input matrix $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$. The result is a 1x1 output $\begin{bmatrix} 2 \end{bmatrix}$, indicated by a large arrow. A pink line connects the top-left element of the kernel to the top-left element of the input matrix, highlighting the receptive field of that output unit.

Edges

Source: <https://courses.cs.washington.edu/courses/cse576/23sp/notes.html>



- Smooth first, then derivative

$$\frac{1}{2} \times \left(\begin{bmatrix} -1 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \right)$$

$\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$ \rightarrow $\frac{1}{2} \times \begin{bmatrix} 2 & 0 \\ & \end{bmatrix}$

The diagram illustrates a convolution operation. At the top, a horizontal kernel $\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$ is shown multiplying a vertical input matrix $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$. Below this, a pink arrow points from the kernel to a 3x3 input matrix $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$, with a pink box highlighting the central 3x3 submatrix $\begin{bmatrix} 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$. A large black arrow points to the result, which is scaled by $\frac{1}{2}$ to produce the output $\begin{bmatrix} 2 & 0 \\ & \end{bmatrix}$.

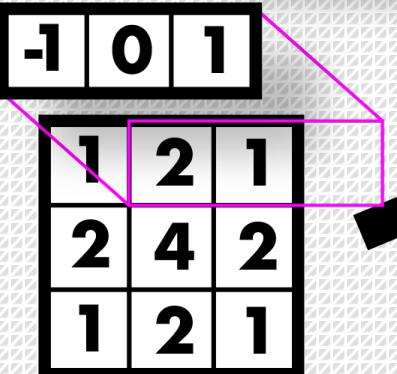
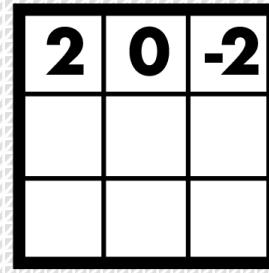
Edges

Source: <https://courses.cs.washington.edu/courses/cse576/23sp/notes.html>



- Smooth first, then derivative

$$\frac{1}{2} \times \left(\begin{bmatrix} -1 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \right)$$

$\frac{1}{2} \times$  \rightarrow $\frac{1}{2} \times$ 

A diagram illustrating a 3x3 convolution operation. A 3x3 input matrix is multiplied by a 3x3 kernel. The result is then scaled by $\frac{1}{2}$. The input matrix has values 1, 2, 1 in its first row, and 2, 4, 2, 1, 2, 1 in its second and third rows respectively. The kernel has values -1, 0, 1 in its first row. The result matrix has values 2, 0, -2 in its first row, and empty cells in its second and third rows. A pink arrow points from the input matrix to the result matrix, indicating the flow of data through the convolution process.

Edges

Source: <https://courses.cs.washington.edu/courses/cse576/23sp/notes.html>



- Smooth first, then derivative

$$\frac{1}{2} \times \left(\begin{bmatrix} -1 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \right)$$

The diagram illustrates a convolution operation. A 3x3 input matrix (bottom) is multiplied by a 3x3 kernel (top-left). The result is a 2x2 output matrix (top-right). A pink arrow points from the top-left corner of the input to the top-left corner of the output, indicating the receptive field of that output unit. The input values are 1, 2, 1; 2, 4, 2; 1, 2, 1. The kernel values are -1, 0, 1; 2, 4, 2; 1, 2, 1. The output values are 2, 0, -2; 4.

-1	0	1	2	1	1	2	1
2	4	2	2	4	2	2	4
1	2	1	1	2	1	1	2

2	0	-2		
4				

Edges

Source: <https://courses.cs.washington.edu/courses/cse576/23sp/notes.html>



- Smooth first, then derivative

$$\frac{1}{2} \times \left(\begin{bmatrix} -1 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \right)$$

The diagram illustrates a convolution operation. On the left, a 3x3 input matrix (1, 2, 1; 2, 4, 2; 1, 2, 1) is multiplied by a 1x3 filter (-1, 0, 1). The result is a 3x3 output matrix (2, 0, -2; 4, 0, -4; 2, 0, -2), which is then scaled by $\frac{1}{2}$. A large black arrow points from the input to the output. A pink line highlights the bottom-right element of the input matrix (1) and the corresponding element in the filter (-1), indicating they are multiplied together during the computation.

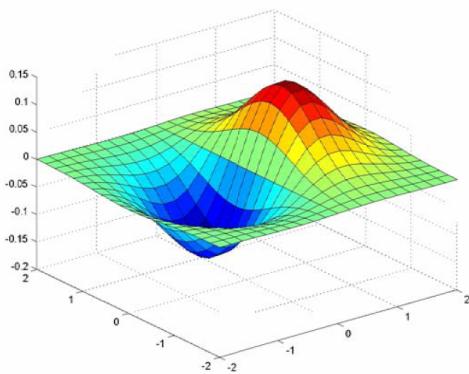
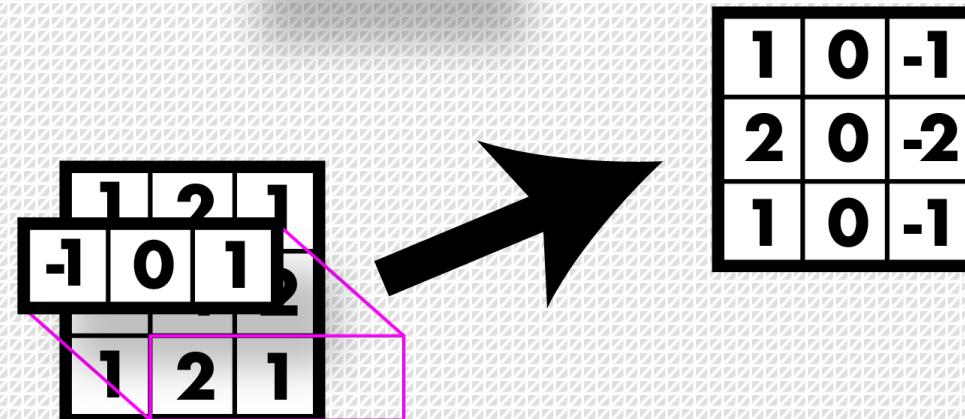
Edges

Source: <https://courses.cs.washington.edu/courses/cse576/23sp/notes.html>



- Smooth first, then derivative

$$\frac{1}{2} \times \left(\begin{bmatrix} -1 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \right)$$



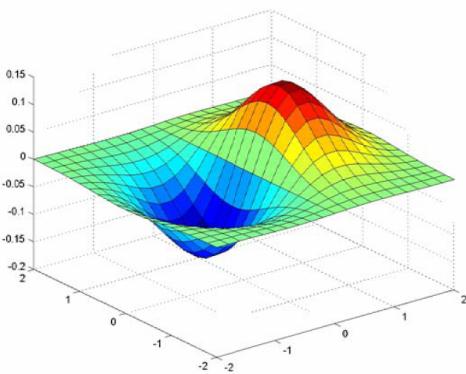
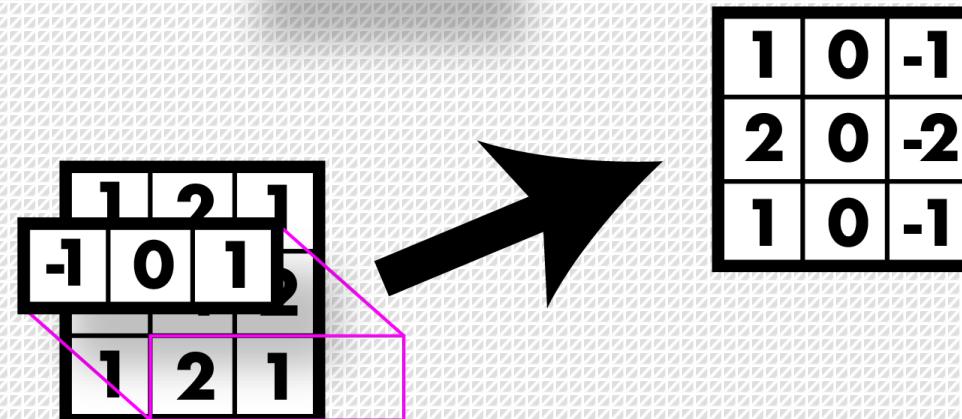
Edges

Source: <https://courses.cs.washington.edu/courses/cse576/23sp/notes.html>



- Smooth first, then derivative

$$\frac{1}{2} \times \left(\begin{bmatrix} -1 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \right)$$



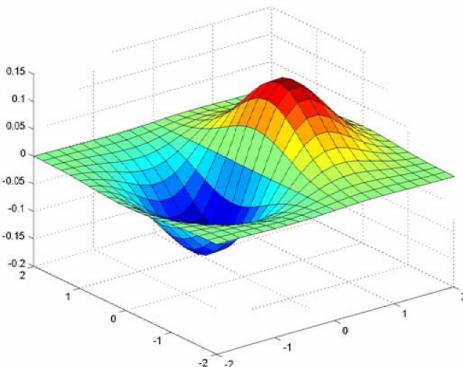
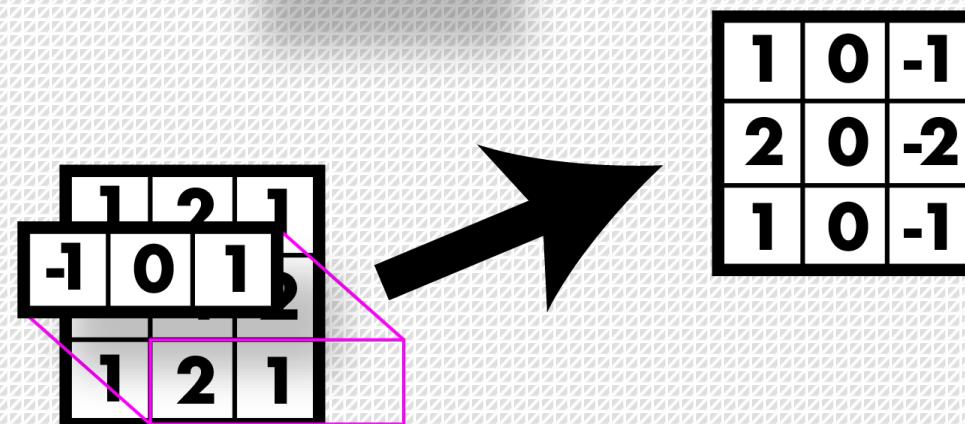
Edges

Source: <https://courses.cs.washington.edu/courses/cse576/23sp/notes.html>



- Smooth first, then derivative \rightarrow Sobel filter!

$$\frac{1}{2} \times \left(\begin{bmatrix} -1 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \right)$$



Edges

Source: <https://courses.cs.washington.edu/courses/cse576/23sp/notes.html>



- Smooth first, then derivative

$$\frac{1}{K^2} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & 1 & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$$\frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

$$\frac{1}{8} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\frac{1}{4} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

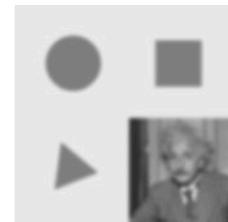
$$\frac{1}{K} \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}$$

$$\frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

$$\frac{1}{16} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

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

$$\frac{1}{2} \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$$



(a) box, $K = 5$

(b) bilinear

(c) “Gaussian”

(d) Sobel

(e) corner

Figure 3.14 Separable linear filters: For each image (a)–(e), we show the 2D filter kernel (top), the corresponding horizontal 1D kernel (middle), and the filtered image (bottom). The filtered Sobel and corner images are signed, scaled up by $2\times$ and $4\times$, respectively, and added to a gray offset before display.

Edges

Source: <https://courses.cs.washington.edu/courses/cse576/23sp/notes.html>



- But

$$\frac{1}{K^2} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$$\frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

$$\frac{1}{8} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\frac{1}{4} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

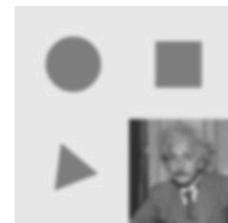
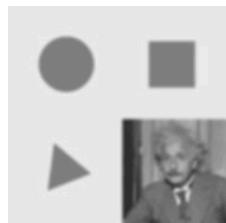
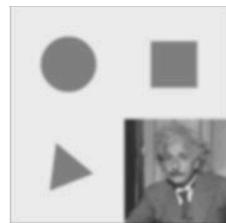
$$\frac{1}{K} \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}$$

$$\frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

$$\frac{1}{16} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

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

$$\frac{1}{2} \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$$



(a) box, $K = 5$

(b) bilinear

(c) “Gaussian”

(d) Sobel

(e) corner

Figure 3.14 Separable linear filters: For each image (a)–(e), we show the 2D filter kernel (top), the corresponding horizontal 1D kernel (middle), and the filtered image (bottom). The filtered Sobel and corner images are signed, scaled up by $2\times$ and $4\times$, respectively, and added to a gray offset before display.

Edges

Source: <https://courses.cs.washington.edu/courses/cse576/23sp/notes.html>



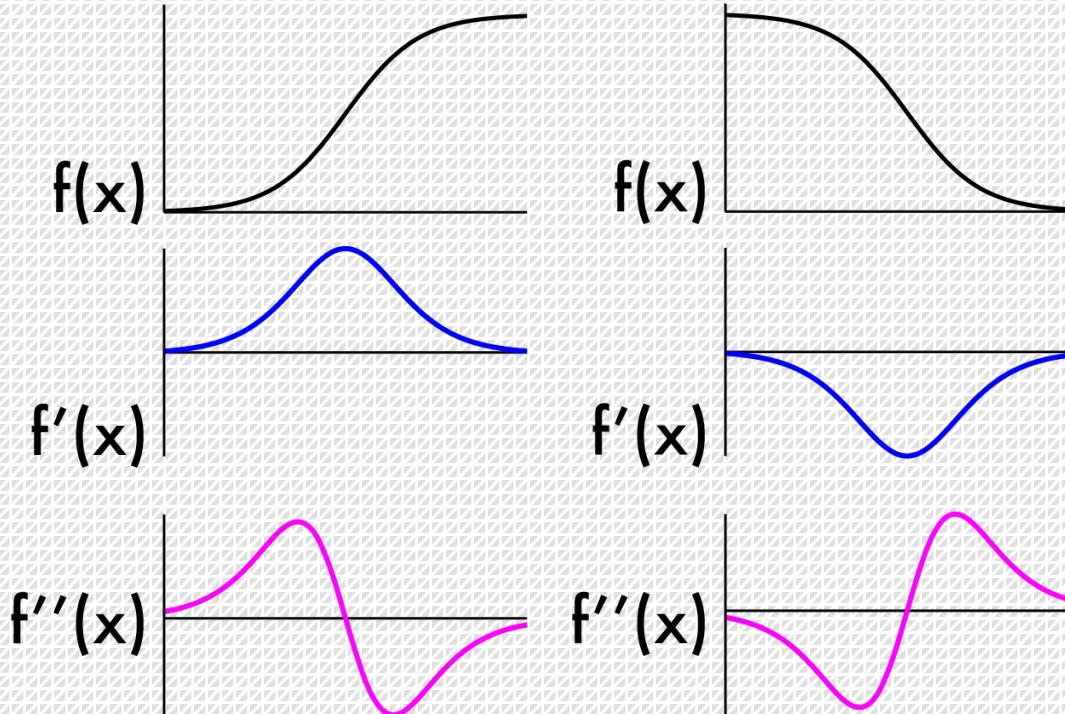
- Could take derivative
- Find high responses
- Sobel filters!
- But...
- Edges go both ways
- Want to find extrema

Edges

Source: <https://courses.cs.washington.edu/courses/cse576/23sp/notes.html>



- Could take derivative
- Find high responses
- Sobel filters!
- But...
- Edges go both ways
- Want to find extrema



Edges

Source: <https://courses.cs.washington.edu/courses/cse576/23sp/notes.html>



- Crosses zero at extrema

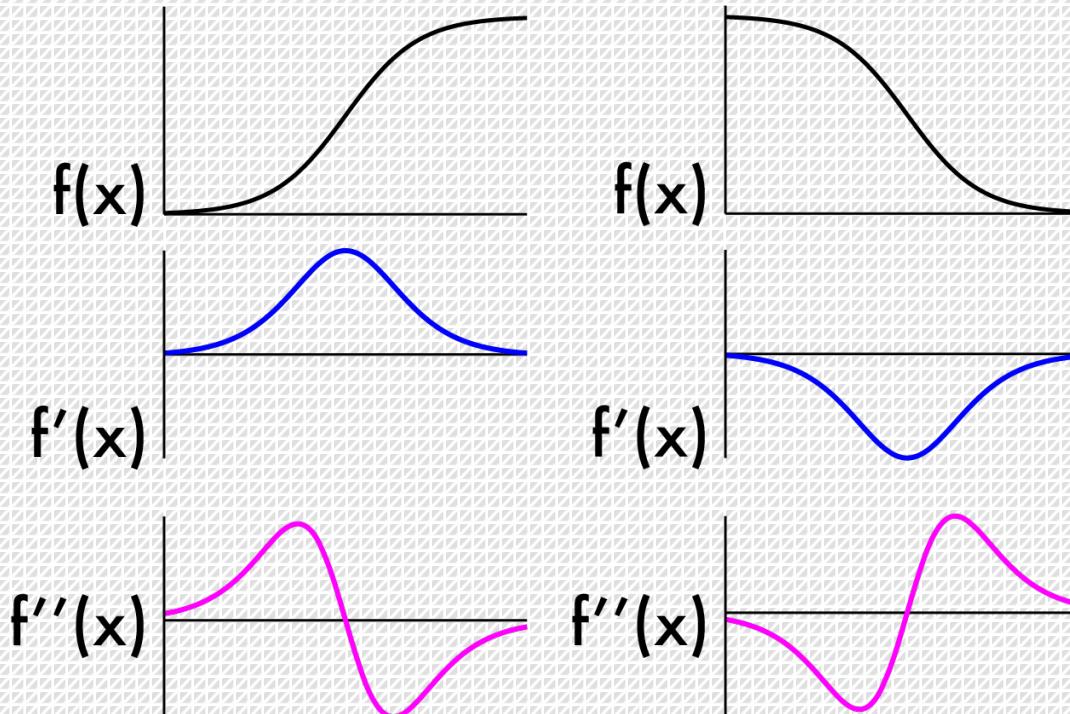
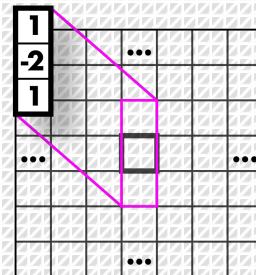
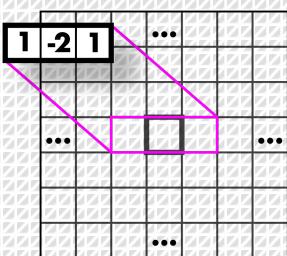
- Recall:

- $$f''(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}.$$

- Laplacian:

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

- Again, have to estimate $f''(x)$:



Edges

Source: <https://courses.cs.washington.edu/courses/cse576/23sp/notes.html>



- Laplacians

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

Edges

Source: <https://courses.cs.washington.edu/courses/cse576/23sp/notes.html>



- Laplacians

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

$$\left(\begin{array}{|c|c|c|} \hline 1 & -2 & 1 \\ \hline \end{array} \right) * \boxed{\text{grid}} + \left(\begin{array}{|c|c|c|} \hline 1 & -2 & 1 \\ \hline \end{array} \right) * \boxed{\text{grid}} = \\
 \left(\begin{array}{|c|c|c|} \hline 1 & -2 & 1 \\ \hline \end{array} \right) + \left(\begin{array}{|c|c|c|} \hline 1 & -2 & 1 \\ \hline \end{array} \right) * \boxed{\text{grid}}$$

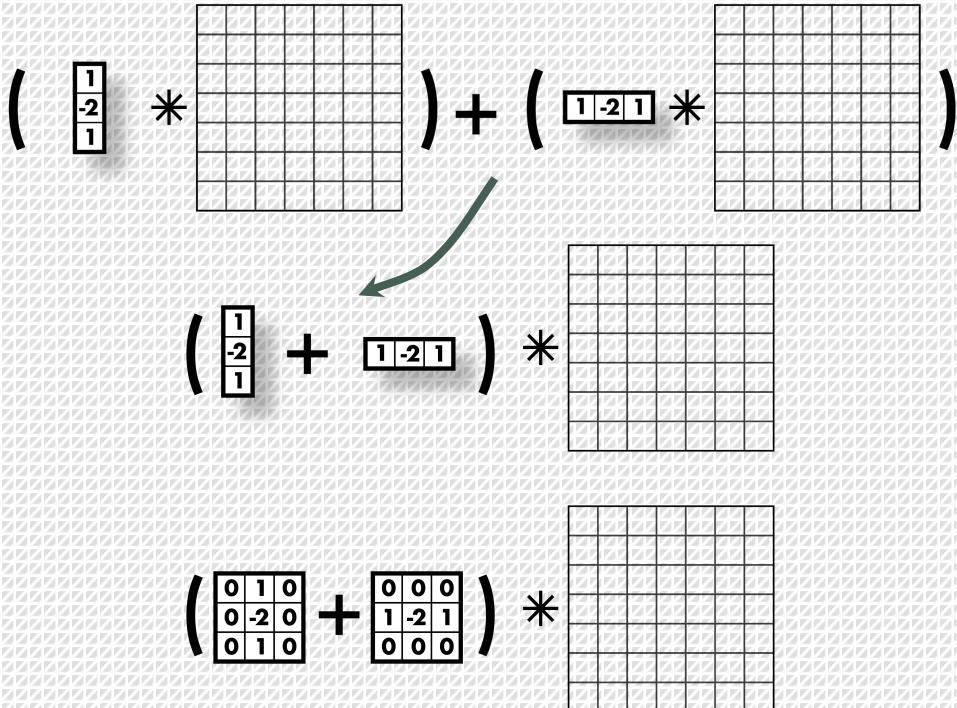
Edges

Source: <https://courses.cs.washington.edu/courses/cse576/23sp/notes.html>



- Laplacians

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



Edges

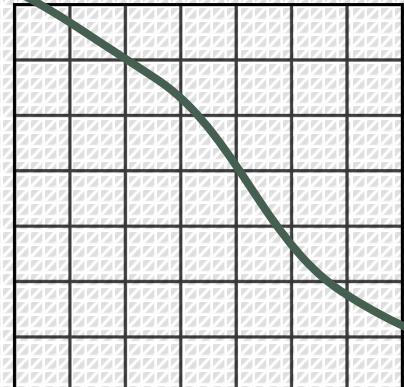
Source: <https://courses.cs.washington.edu/courses/cse576/23sp/notes.html>



- Laplacians

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

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$



$$\left(\begin{array}{|c|c|c|} \hline 1 & & \\ \hline -2 & & \\ \hline 1 & & \\ \hline \end{array} \right) * \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|} \hline \end{array} \right) + \left(\begin{array}{|c|c|c|} \hline 1 & -2 & 1 \\ \hline \end{array} \right) * \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|} \hline \end{array} \right)$$

$$\left(\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} \right) *$$

Edges

Source: <https://courses.cs.washington.edu/courses/cse576/23sp/notes.html>

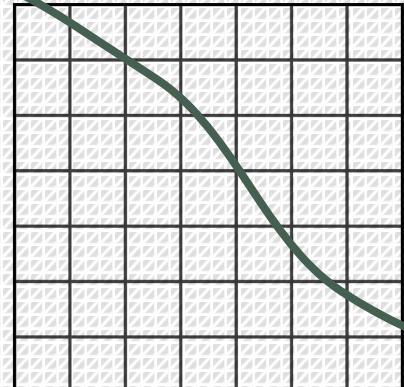


- Laplacians

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

$$\begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

*



$$\left(\begin{array}{|c|c|c|} \hline 1 & & \\ \hline 2 & & \\ \hline 1 & & \\ \hline \end{array} \right) * \boxed{\text{ }} \quad) + \left(\begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline \end{array} \right) * \boxed{\text{ }} \quad)$$

$$\left(\begin{array}{c|c} 1 & \\ -2 & \\ 1 & \end{array} + \begin{array}{c|c|c} 1 & -2 & 1 \end{array} \right) *$$

Other linear filters

- summed area tables (*integral image*)
- ... is used for face detection to compute simple multi-scale low-level features

3	2	7	2	3
1	5	1	3	4
5	1	3	5	1
4	3	2	1	6
2	4	1	4	8

(a) $S = 24$

3	5	12	14	17
4	11	19	24	31
9	17	28	38	46
13	24	37	48	62
15	30	44	59	81

(b) $s = 28$

3	5	12	14	17
4	11	19	24	31
9	17	28	38	46
13	24	37	48	62
15	30	44	59	81

(c) $S = 24$

Figure 3.17 Summed area tables: (a) original image; (b) summed area table; (c) computation of area sum. Each value in the summed area table $s(i, j)$ (red) is computed recursively from its three adjacent (blue) neighbors (3.31). Area sums S (green) are computed by combining the four values at the rectangle corners (purple) (3.32). Positive values are shown in **bold** and negative values in *italics*.



Talk from the top scientist



AI 如何拯救人性

■ Neighborhood operators

- Gaussian filter

