Lecture 3. Filters and Convolutions

Linear systems

Juan Carlos Niebles and Jiajun Wu

CS131 Computer Vision: Foundations and Applications

Applications of Linear systems and Filters

De-noising

Salt and pepper noise

Super-resolution

In-painting

Bertamio et al

Systems and Filters

Filtering:

- Forming a new image whose pixel values are transformed from original pixel values

Goals:

- Extract useful information from images, or transform images to modify/enhance image properties
    - Features (edges, corners, blobs…)
    - Super-resolution; in-painting; de-noising

Images as functions

- Images are usually **digital** (**discrete**):
  - **Sample** the 2D space on a regular grid

- Represented as a matrix of integer values

pixel

m

n

62	79	23	119	120	105	4	0
10	10	9	62	12	78	34	0
10	58	197	46	46	0	0	48
176	135	5	188	191	68	0	49
2	1	1	29	26	37	0	77
0	89	144	147	187	102	62	208
255	252	0	166	123	62	0	31
166	63	127	17	1	0	99	30

Images as functions

Cartesian coordinates

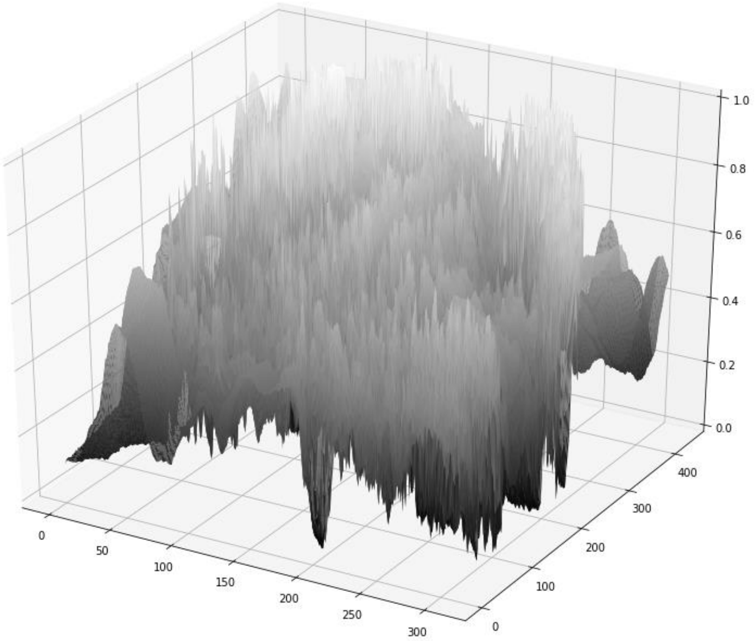
$$
f[n,m] = \begin{bmatrix}
\ddots & & \vdots & & \\
& f[-1,-1] & f[-1,0] & f[-1,1] & \\
\ldots & f[0,-1] & \underline{f[0,0]} & f[0,1] & \ldots \\
& f[1,-1] & f[1,0] & f[1,1] & \\
& & \vdots & & \ddots
\end{bmatrix}
$$

Notation for discrete functions

Images as functions

- **An Image** as a function f from R^2 to R^M:
  - $f\,[x, y]$ gives the **intensity** at position $[x, y]$
  - Defined over a rectangle, with a finite range:

$$f\colon \underbrace{[a, b] \times [c, d\,]}_{\text{Domain support}} \rightarrow \underbrace{[0,255]}_{\text{range}}$$



Images as functions

- **An Image** as a function f from R^2 to R^M:
  - $f\,[x, y]$ gives the **intensity** at position $[x, y]$
  - Defined over a rectangle, with a finite range:

$$f\colon \underbrace{[a, b] \times [c, d\,]}_{\text{Domain support}} \rightarrow \underbrace{[0,255]}_{\text{range}}$$

- A color image: $f(x, y) = \begin{bmatrix} r(x, y) \\ g(x, y) \\ b(x, y) \end{bmatrix}$

System and Filters

- We define a system as a unit that converts an input function f[n,m] into an output function g[n,m], where [n, m] are the independent variables.
  - In the case of images, [n, m] represents the **spatial position in the image.**

$$f[n,m] \rightarrow \boxed{\text{System } \mathcal{S}} \rightarrow g[n,m]$$

System and Filters

- S is the **system operator,** defined as a mapping/assignment that transforms the input f into the output g.

$$f[n,m] \rightarrow \boxed{\text{System } \mathcal{S}} \rightarrow g[n,m]$$

$$g = \mathcal{S}[f], \quad g[n,m] = \mathcal{S}\{f[n,m]\}$$

$$f[n,m] \xrightarrow{\mathcal{S}} g[n,m]$$

Filter example #1: Moving Average

2D moving average over a 3×3 neighborhood window

Original image

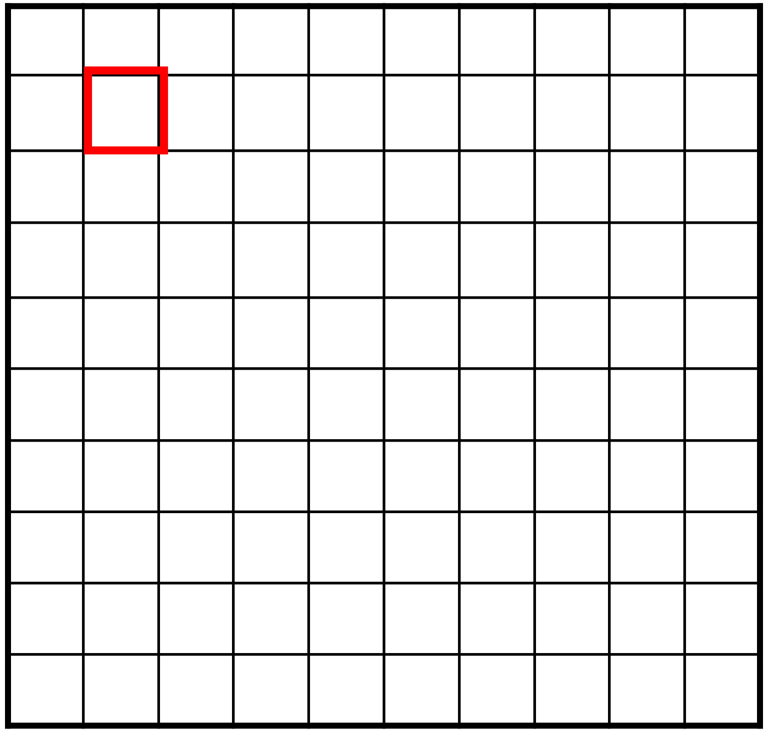
Smoothed image

Filter example #1: Moving Average

$\mathrm{f}[n, m]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$\mathrm{g}[n, m]$



Courtesy of S. Seitz

Filter example #1: Moving Average

$\text{f}[n,m]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$\text{g}[n,m]$

	0	10							

Filter example #1: Moving Average

$\mathrm{f}[n, m]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$\mathrm{g}[n, m]$

	0	10	20						

Filter example #1: Moving Average

$f[n,m]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$g[n,m]$

	0	10	20	30					

Filter example #1: Moving Average

$f[n, m]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$g[n, m]$

	0	10	20	30	30				

Filter example #1: Moving Average

$\mathrm{f}[n, m]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$\mathrm{g}[n, m]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

Filter example #1: Moving Average

2D moving average over a 3 × 3 window of neighborhood

Sum over rows

Sum over columns

$$g[n,m] = \frac{1}{9} \boxed{\sum_{k=n-1}^{n+1}} \boxed{\sum_{l=m-1}^{m+1}} f[k,l]$$

$$= \frac{1}{9} \sum_{k=-1}^{1} \sum_{l=-1}^{1} f[n-k, m-l]$$

Filter "kernel", "mask"

h

$$\frac{1}{9}$$

1	1	1
1	1	1
1	1	1

Filter example #1: Moving Average

Filter
"kernel"
"mask"

h

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

$\mathrm{f}[n, m]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$\mathrm{g}[n, m]$

	0	10	20	30	30				

Filter example #1: Moving Average

2D moving average over a 3 × 3 window of neighborhood

Sum over rows

Sum over columns

$$g[n,m] = \frac{1}{9} \boxed{\sum_{k=n-1}^{n+1}} \boxed{\sum_{l=m-1}^{m+1}} f[k,l]$$

$$= \frac{1}{9} \sum_{k=-1}^{1} \sum_{l=-1}^{1} f[n-k, m-l]$$

Filter "kernel", "mask"

h

$$\frac{1}{9}$$

1	1	1
1	1	1
1	1	1

Filter example #1: Moving Average

2D moving average over a 3 × 3 window of neighborhood

Sum over rows

Sum over columns

$$g[n,m] = \frac{1}{9} \boxed{\sum_{k=n-1}^{n+1}} \boxed{\sum_{l=m-1}^{m+1}} f[k,l]$$

$$= \frac{1}{9} \sum_{k=-1}^{1} \sum_{l=-1}^{1} f[n-k, m-l]$$

Filter "kernel", "mask"

h

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Filter example #1: Moving Average

In summary:

- This filter “transforms” each pixel value into the average value of its neighborhood.

- Achieve smoothing effect (remove sharp features)

$$h[\times,\times]$$

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Filter example #1: Moving Average

2D moving average over a 3×3 neighborhood window

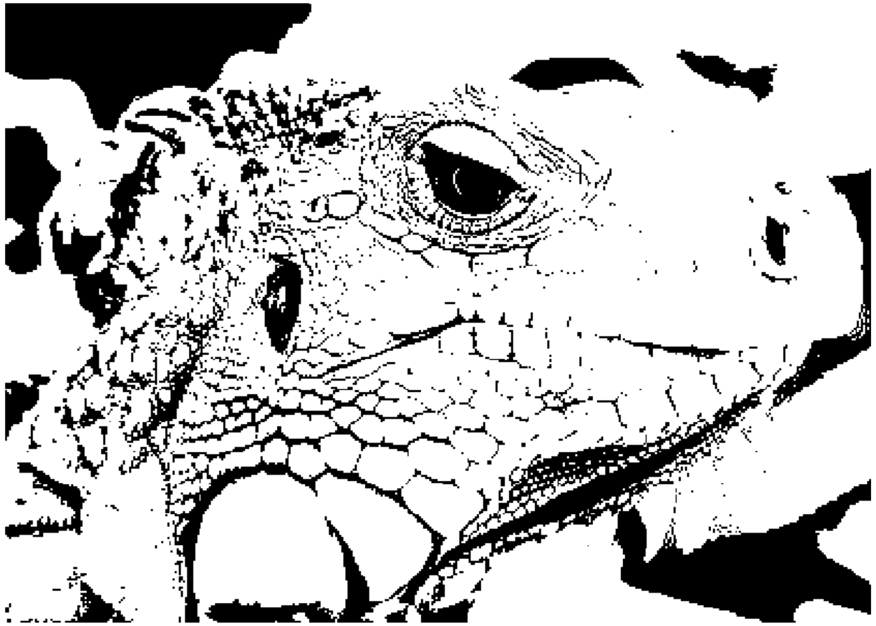
Original image

Smoothed image

Filter example #2: Image Segmentation

- Image segmentation based on a simple threshold:

$$g[n,m] = \begin{cases} 255, & f[n,m] > 100 \\ 0, & \text{otherwise.} \end{cases}$$



Systems can be classified based on their properties

- Amplitude properties:
  - Additivity
  - Homogeneity
  - Superposition
  - Stability
  - Invertibility
- Spatial properties
  - Causality
  - Shift invariance
  - Memory

Shift-Invariant (SI) Systems

- If $f[n,m] \xrightarrow{\mathcal{S}} g[n,m]$

then, S is Shift Invariant if

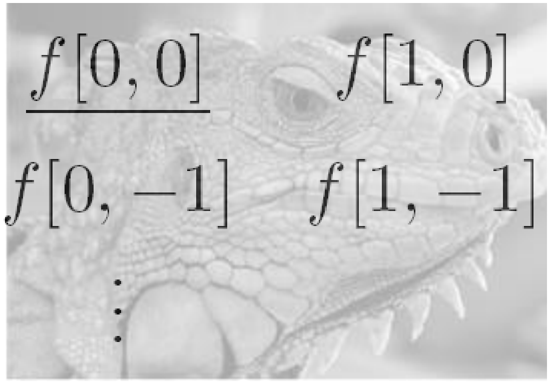
$$f[n-n_0, m-m_0] \xrightarrow{\mathcal{S}} g[n-n_0, m-m_0]$$

for every input image f[n,m] and shifts n_0, m_{0}.

What does shifting an image look like?

Cartesian coordinates

$$f[n,m] = \begin{bmatrix} \ddots & & \vdots & & \\ & f[-1,1] & f[0,1] & f[1,1] & \\ \dots & f[-1,0] & \underline{f[0,0]} & f[1,0] & \dots \\ & f[-1,-1] & f[0,-1] & f[1,-1] & \\ & & \vdots & & \ddots \end{bmatrix}$$



Is the moving average system shift invariant?

$f[n,m]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$g[n,m]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

Is the moving average system shift invariant?

- Let's start with passing f through our system

$$f[n,m] \xrightarrow{S} g[n,m]$$

$$g[n,m] = \frac{1}{9} \sum_{k=n-1}^{n+1} \sum_{l=m-1}^{m+1} f[k,l] = \frac{1}{9} \sum_{k=-1}^{1} \sum_{l=-1}^{1} f[n-k, m-l]$$

- Now, let's see what we get when we pass in a shifted version of the input f:

$$f[n-n_0, m-m_0] \xrightarrow{S} \frac{1}{9} \sum_{k=-1}^{1} \sum_{l=-1}^{1} f[(n-n_0)-k, (m-m_0)-l]$$

$$= g[n-n_0, m-m_0]$$

Yes!

Systems can be classified based on their <u>properties</u>

- Amplitude properties:
  - Additivity
  - Homogeneity
  - Superposition
  - Stability
  - Invertibility
- Spatial properties
  - Causality
  - Shift invariance
  - Memory

Linear Systems (filters)

$$f[n,m] \rightarrow \boxed{\text{System } \mathcal{S}} \rightarrow g[n,m]$$

- Linear filtering:
  - Form a new image whose pixels are a weighted sum of original pixel values
  - Use the same set of weights at each point
- **S** is a linear system (function) if and only if it *satisfies*

$$S\{\alpha f_1[n,m] + \beta f_2[n,m]\} = \alpha S\{f_1[n,m]\} + \beta S\{f_2[n,m]\}$$

superposition property

Linear Systems (filters)

$$f[n,m] \rightarrow \boxed{\text{System } \mathcal{S}} \rightarrow g[n,m]$$

- Is the moving average a linear system?

Yes!

- Is thresholding a linear system?

We could have $f_1[n,m] + f_2[n,m] > T$, when $f_1[n,m] < T$ and $f_2[n,m] < T$

No!

Systems can be classified based on their properties

- Amplitude properties:
  - Additivity
  - Homogeneity
  - Superposition
  - Stability
  - Invertibility
- Spatial properties
  - Causality
  - Shift invariance
  - Memory

Linear Shift Invariant (LSI) systems

An LSI system satisfies two properties:

- Superposition property

$$S\{\alpha f_1[n,m] + \beta f_2[n,m]\} = \alpha S\{f_1[n,m]\} + \beta S\{f_2[n,m]\}$$

- Shift invariance

$$f[n-n_0, m-m_0] \xrightarrow{\mathcal{S}} g[n-n_0, m-m_0]$$

Is the moving average filter linear and shift invariant?

Linear Shift Invariant (LSI) systems

- An LSI system is completely specified by its impulse response.

$$\boxed{\delta_2[n,m]} \xrightarrow{S} \boxed{h[n,m]}$$

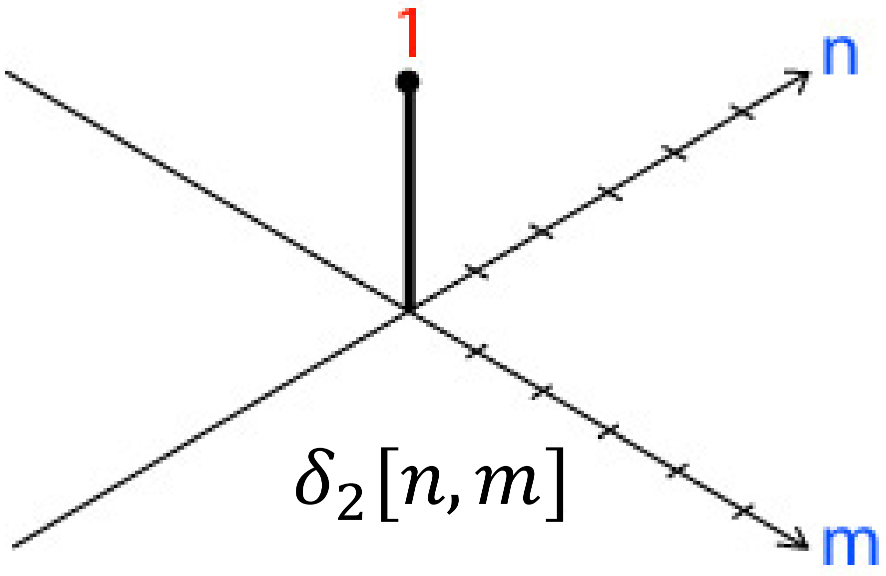
Pass in an impulse function **Record its response**

- By passing an impulse function into an LSI system, we get it's impulse response.

2D impulse function $\delta_2[n,m]$

- 1 at [0,0].
- 0 everywhere else

	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
	0	0	0	1	0	0	0	
	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	



Impulse response to the moving average filter

- Recall the expression for our 3x3 moving average filter:

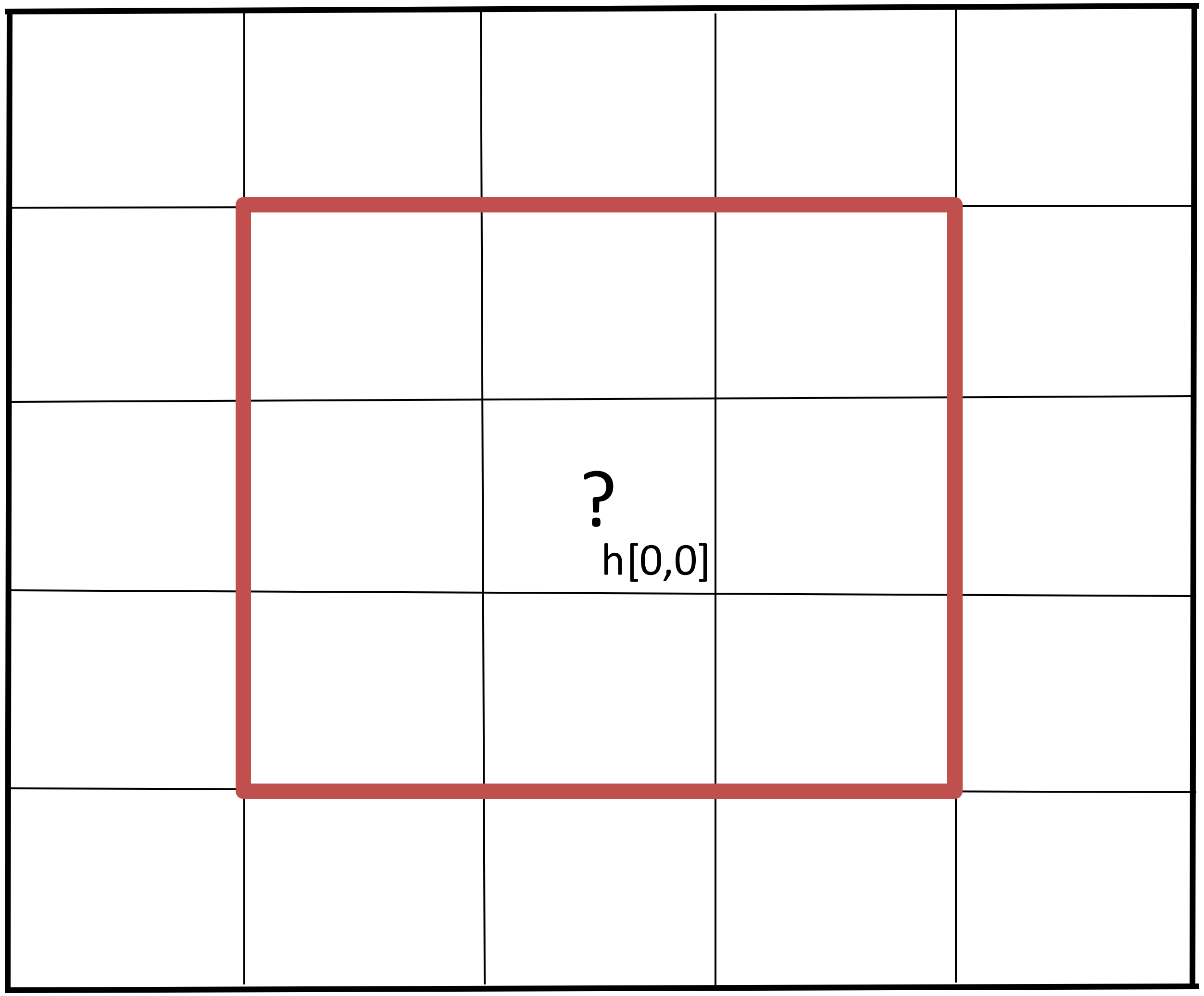
$$f[n,m] \xrightarrow{S} \frac{1}{9} \sum_{k=-1}^{1} \sum_{l=-1}^{1} f[n-k, m-l]$$

- We can use it to obtain an expression for the impulse response

$$\delta_2[n,m] \xrightarrow{S} h[n,m]$$

$$= \frac{1}{9} \sum_{k=-1}^{1} \sum_{l=-1}^{1} \delta_2[n-k, m-l]$$

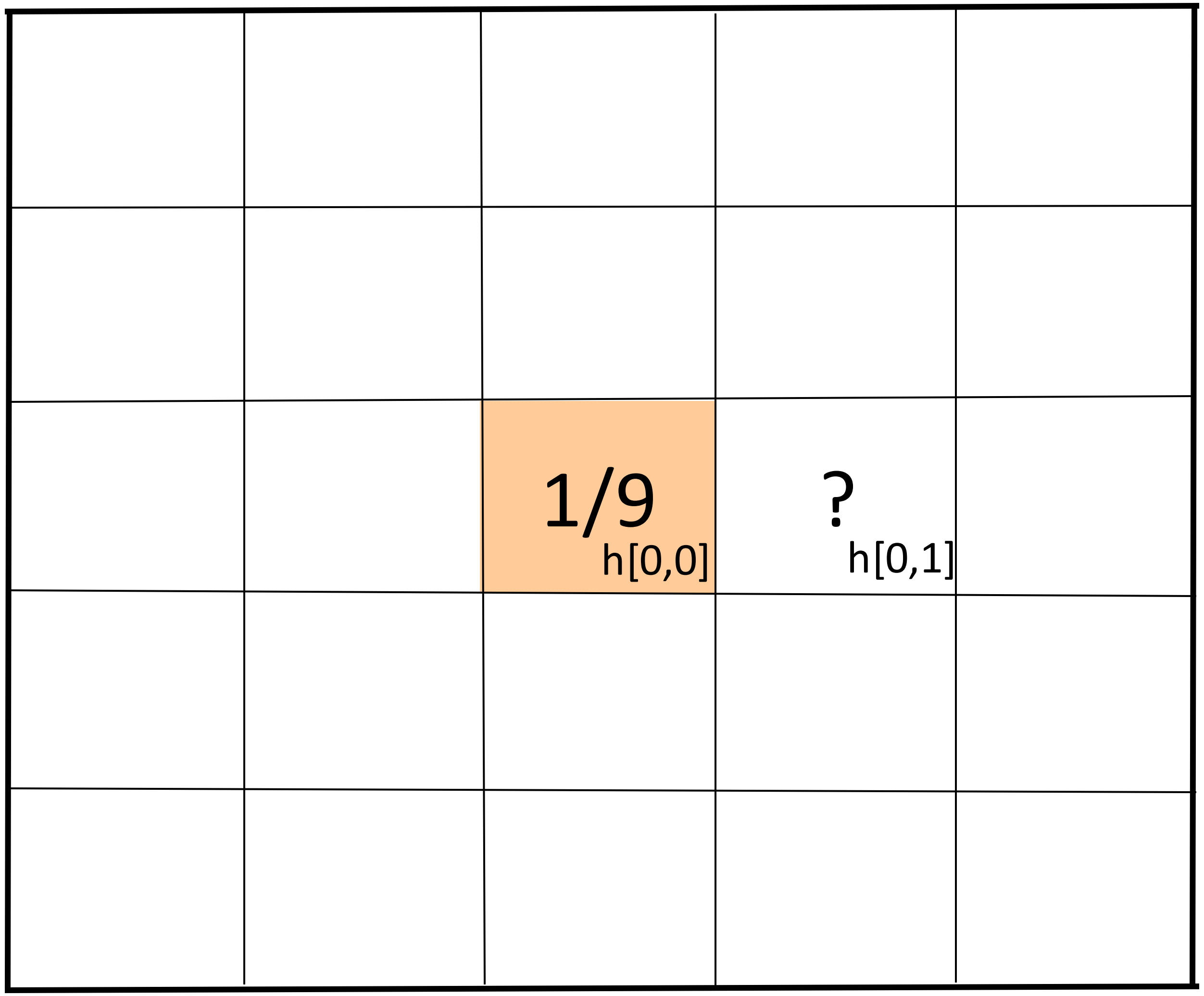
Impulse response to the moving average filter



$$\delta_2[n,m] \xrightarrow{S} h[n,m]$$

$$= \frac{1}{9}\sum_{k=-1}^{1}\sum_{l=-1}^{1}\delta_2[n-k,m-l]$$

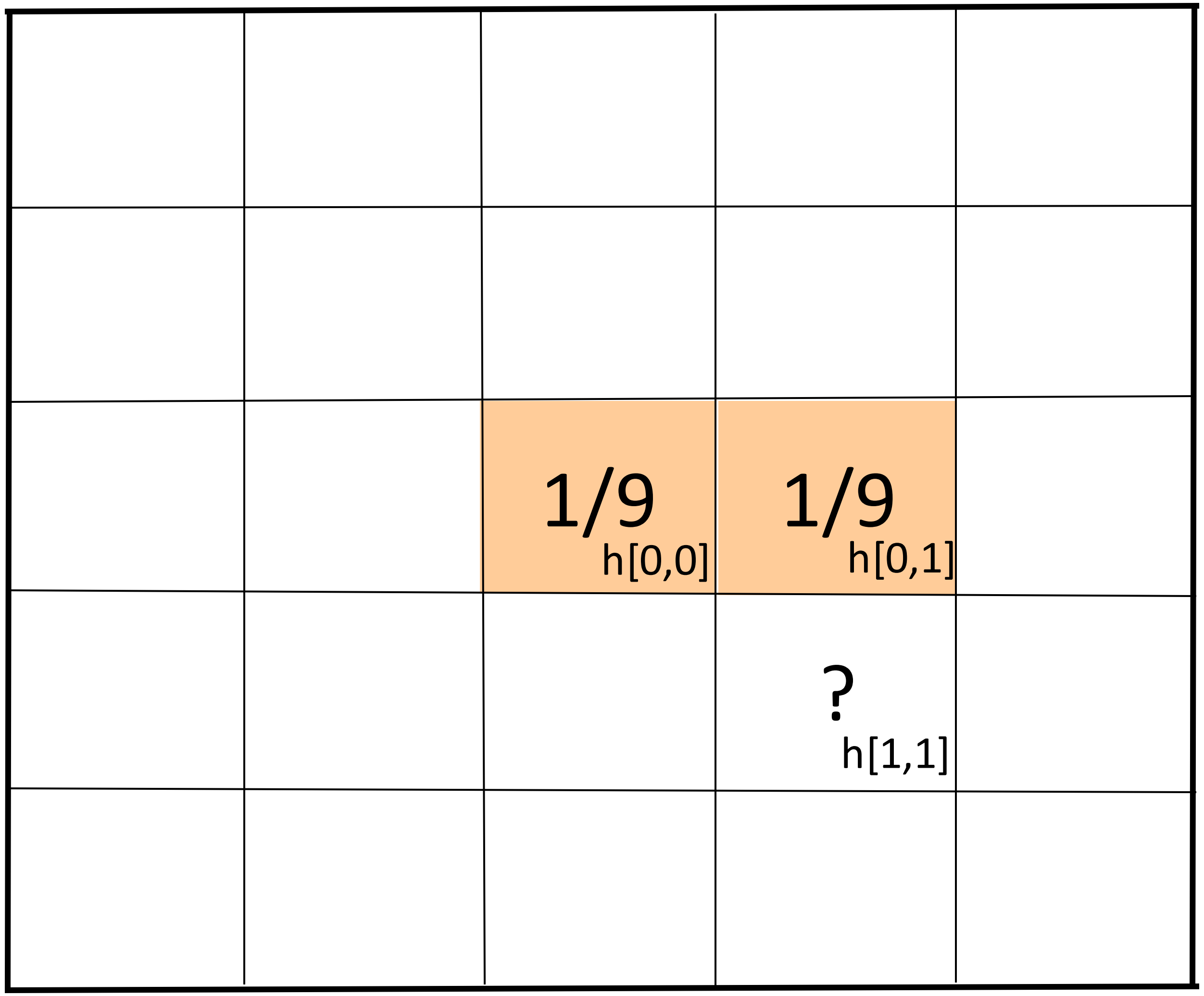
Impulse response to the moving average filter



$$\delta_2[n,m] \xrightarrow{S} h[n,m]$$

$$= \frac{1}{9} \sum_{k=-1}^{1} \sum_{l=-1}^{1} \delta_2[n-k, m-l]$$

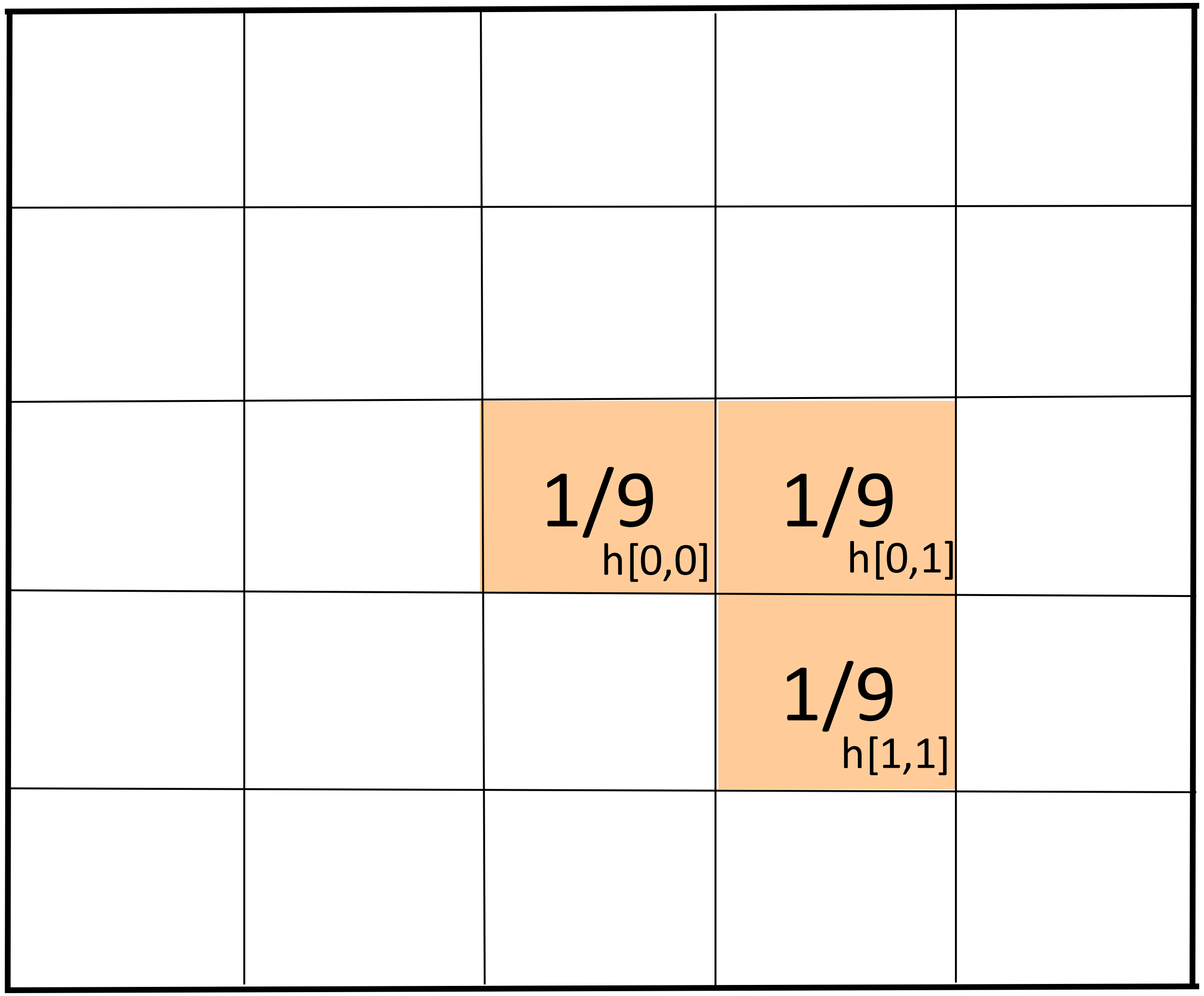
Impulse response to the moving average filter



$$\delta_2[n,m] \xrightarrow{S} h[n,m]$$

$$= \frac{1}{9} \sum_{k=-1}^{1} \sum_{l=-1}^{1} \delta_2[n-k, m-l]$$

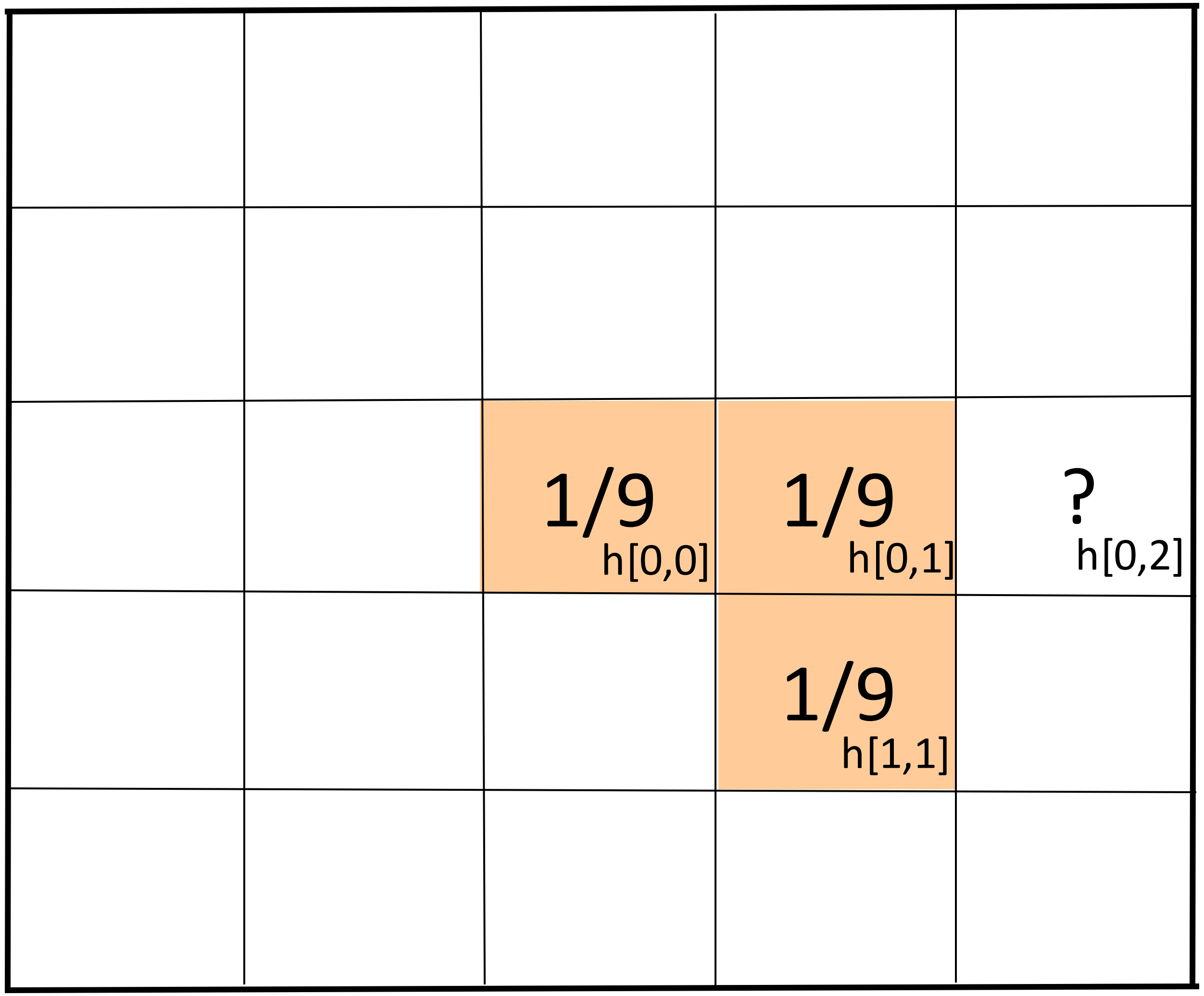
Impulse response to the moving average filter



$$\delta_2[n,m] \xrightarrow{S} h[n,m]$$

$$= \frac{1}{9} \sum_{k=-1}^{1} \sum_{l=-1}^{1} \delta_2[n-k, m-l]$$

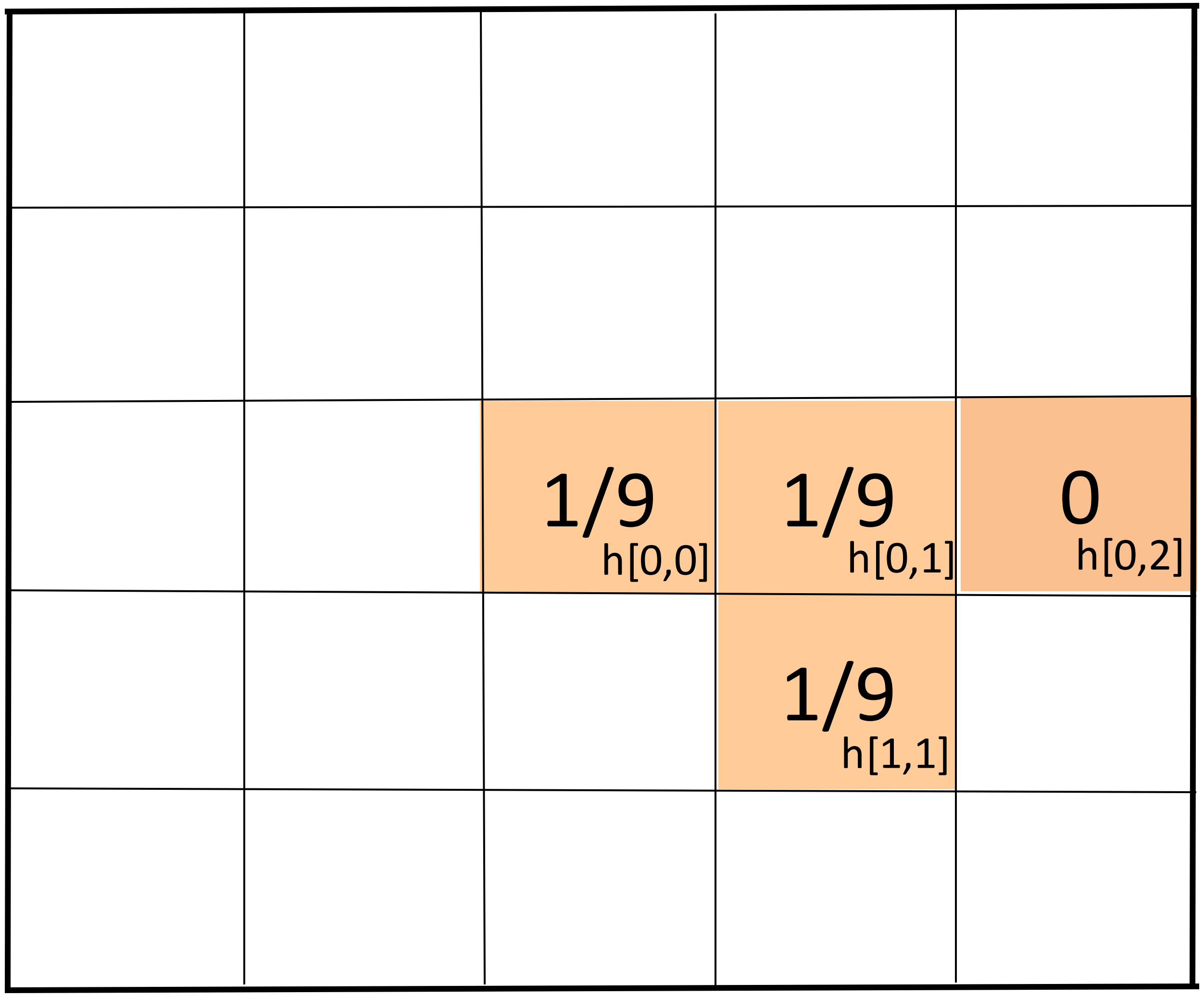
Impulse response to the moving average filter



$$\delta_2[n,m] \xrightarrow{S} h[n,m]$$

$$= \frac{1}{9} \sum_{k=-1}^{1} \sum_{l=-1}^{1} \delta_2[n-k, m-l]$$

Impulse response to the moving average filter



$$\delta_2[n,m] \xrightarrow{S} h[n,m]$$

$$= \frac{1}{9}\sum_{k=-1}^{1}\sum_{l=-1}^{1}\delta_2[n-k,m-l]$$

Impulse response to the moving average filter

0	0	0	0	0
0	1/9 h[-1,-1]	1/9	1/9	0
0	1/9	1/9 h[0,0]	1/9 h[0,1]	0 h[0,2]
0	1/9	1/9	1/9 h[1,1]	0
0	0	0	0	0

$$\delta_2[n,m] \xrightarrow{S} h[n,m]$$

$$= \frac{1}{9}\sum_{k=-1}^{1}\sum_{l=-1}^{1}\delta_2[n-k,m-l]$$

Impulse response of the 3 by 3 moving average filter

$$h[n,m] = \frac{1}{9}\sum_{k=-1}^{1}\sum_{l=-1}^{1}\delta_2[n-k,m-l]$$

$$= \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

h

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Linear Shift Invariant (LSI) systems

- An LSI system is completely specified by its impulse response.

$$\boxed{\delta_2[n,m]} \xrightarrow{S} \boxed{h[n,m]}$$

Pass in an impulse function　　Record its response

- By passing an impulse function into an LSI system, we get it's impulse response.

Linear Shift Invariant (LSI) systems

- An LSI system is completely specified by its impulse response.
  - For any input f, we can compute the output g in terms of the impulse response h.

$$f[n,m] \xrightarrow{S} g[n,m]$$

  - In the following, we'll derive an expression for g in terms of h.
  - We know the LSI system satisfies the superposition property and the shift-invariance property. We also know h:

$$\delta_2[n,m] \xrightarrow{S} h[n,m]$$

3 properties we need:

- We know what happens when we send a delta function through an LSI system:

$$\delta_2[n,m] \;\longrightarrow\; \boxed{\text{System } \mathcal{S}} \;\longrightarrow\; h[n,m]$$

- We also know that LSI systems shift the output if the input is shifted:

$$\delta_2[n-k,m-l] \;\longrightarrow\; \boxed{\text{System } \mathcal{S}} \;\longrightarrow\; h[n-k,m-l]$$

- Finally, the superposition principle:

$$S\{\alpha f_1[n,m] + \beta f_2[n,m]\} = \alpha S\{f_1[n,m]\} + \beta S\{f_2[n,m]\}$$

Key idea: write down f as a sum of impulses

Let’s say our input f is a 3x3 image:

f[0,0]	f[0,1]	f[1,1]
f[1,0]	f[1,1]	f[1,2]
f[2,0]	f[2,1]	f[2,2]

$$= \begin{bmatrix} f[0,0] & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & f[0,1] & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \ldots + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & f[2,2] \end{bmatrix}$$

$$= f[0,0]* \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + f[0,1]* \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \ldots + f[2,2]* \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= f[0,0] \cdot \delta_2[n,m] + f[0,1] \cdot \delta_2[n,m-1] + \ldots + f[2,2] \cdot \delta[n-2,n-2]$$

Key idea: write down f as a sum of impulses

- More generally:

$$f[n,m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k,l] \cdot \delta_2[n-k, m-l]$$

- We can now use superposition to see what the output g is:

$$f[n,m] \xrightarrow{S} g[n,m]$$

$$f[n,m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k,l] \cdot \delta_2[n-k, m-l]$$

$$\xrightarrow{S} \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k,l] \cdot S\{\delta_2[n-k, m-l]\}$$

Key idea: write down f as a sum of impulses

- We have:

$$f[n,m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k,l] \cdot \delta_2[n-k, m-l]$$

$$\xrightarrow{S} \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k,l] \cdot S\{\delta_2[n-k, m-l]\}$$

- Recall, by the shift invariance property that:

$$S\{\delta_2[n-k, m-l]\} = h[n-k, m-l]$$

Key idea: write down f as a sum of impulses

- We have:

$$f[n,m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k,l] \cdot \delta_2[n-k, m-l]$$

$$\xrightarrow{S} \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k,l] \cdot S\{\delta_2[n-k, m-l]\}$$

- Which means,

$$f[n,m] \xrightarrow{S} \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k,l] \cdot h[n-k, m-l]$$

Linear Shift Invariant (LSI) systems

- An LSI system is completely specified by its impulse response.
  - For any input f, we can compute the output g in terms of the impulse response h.

$$f[n,m] \xrightarrow{S} g[n,m]$$

$$f[n,m] \xrightarrow{S} \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k,l] \cdot h[n-k, m-l]$$

Discrete Convolution

$$\boxed{f[n,m] * h[n,m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k,l] \cdot h[n-k, m-l]}$$

Linear Shift Invariant (LSI) systems

- An LSI system is completely specified by its impulse response.

$$f[n,m] \xrightarrow{S} g[n,m]$$

$$g[n,m] = f[n,m] * h[n,m]$$

$$f[n,m] * h[n,m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k,l] \cdot h[n-k, m-l]$$

Summary

- Images as functions
- Systems and Filters
- LSI systems and convolution