Spatial Processing

Computational Photography (CSCI 3240U)

Faisal Z. Qureshi
http://vclab.science.ontariotechu.ca
Ontario Tech University

Special thanks to Ioannis Gkioulekas

- Many of the slides are taken with his permission from the computational photography course that he has developed at CMU

Story thus far

- Digital cameras
  - Imaging pipeline
- Image formation
  - Pinholes, lenses, aperture, etc.
- Pixel-wise operations
  - Histogram equalization

Image Enhancement

- Make an image more suitable for a particular application than the original image
- Types of techniques
  - Point processing

E.g., Human perception

on point processes

Image Enhancement

- Make an image more suitable for a particular application than the original image
- Types of techniques
  - Point processing
  - Spatial processing
  - Frequency domain processing

E.g., Human perception

Image Enhancement

- Make an image more suitable for a particular application than the original image
- Types of techniques
  - Point processing
  - Spatial processing (pixel neighbourhoods)
  - Frequency domain processing

E.g., Human perception

Image Enhancement

- Make an image more suitable for a particular application than the original image
- Types of techniques
  - Point processing
  - Spatial processing (pixel neighbourhoods) ⟵ Today's Focus
  - Frequency domain processing

Spatial Processing

- Input image: $f(x, y)$
- Output image: $g(x, y)$
- T is an operator on f or a set of f
  - T is defined over some neighbourhood N of (x, y)
  - T can operate over a set of images

Spatial Filtering

- Two main types
  - Linear filtering
  - Non-linear filtering
- Linear filters
  - Remove, isolate, modify frequencies in the image
  - Foundation based upon the convolution theorem
- Non-linear filters
  - Based upon image statistics

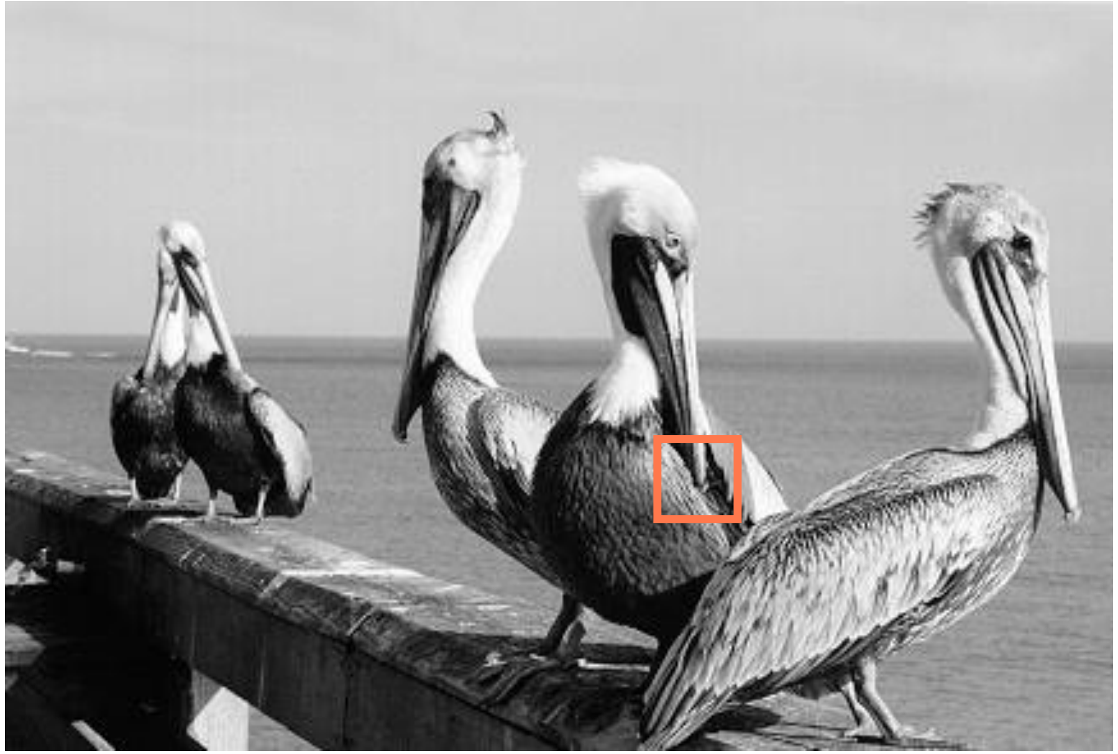
An Example of Spatial Filtering

$f(x, y)$

$g(x, y)$

5 x 5 neighbourhood

An Example of Spatial Filtering



$f(x, y)$

$g(x, y)$

5 x 5 neighbourhood

Linear Filtering in 1D

Signal

$f =$

1	**2**	**4**	**1**	**0**	**9**	**1**	**3**	**5**	**2**

Filter

$h =$

1	**0**	**-1**

Linear Filtering in 1D

1	2	4	1	0	9	1	3	5	2

1	0	-1

Linear Filtering in 1D

1	**2**	**4**	**1**	**0**	**9**	**1**	**3**	**5**	**2**

1	**0**	**-1**

(1)(1)+(2)(0)+(4)(-1)=-3

Linear Filtering in 1D

1	**2**	**4**	**1**	**0**	**9**	**1**	**3**	**5**	**2**

1	**0**	**-1**

$(1)(1)+(2)(0)+(4)(-1)=-3$

↓

	-3								

Linear Filtering in 1D

1	2	4	1	0	9	1	3	5	2

1	0	-1

$$(1)(1)+(2)(0)+(4)(-1)=-3$$

Dot-product

	-3								

Linear Filtering in 1D

1	**2**	**4**	**1**	**0**	**9**	**1**	**3**	**5**	**2**

1	**0**	**-1**

	-3	**?**							

Linear Filtering in 1D

1	**2**	**4**	**1**	**0**	**9**	**1**	**3**	**5**	**2**

1	**0**	**-1**

	-3	**1**							

Linear Filtering in 1D

1	2	4	1	0	9	1	3	5	2

1	0	-1

	-3	1	?						

Linear Filtering in 1D

1	**2**	**4**	**1**	**0**	**9**	**1**	**3**	**5**	**2**

1	**0**	**-1**

↓

	-3	**1**	**4**						

Linear Filtering in 1D

1	**2**	**4**	**1**	**0**	**9**	**1**	**3**	**5**	**2**

1	**0**	**-1**

	-3	**1**	**4**	**?**					

Linear Filtering in 1D

1	**2**	**4**	**1**	**0**	**9**	**1**	**3**	**5**	**2**

1	**0**	**-1**

	-3	**1**	**4**	**-8**					

Linear Filtering in 1D

1	**2**	**4**	**1**	**0**	**9**	**1**	**3**	**5**	**2**

1	**0**	**-1**

	-3	**1**	**4**	**-8**	**?**				

Linear Filtering in 1D

1	2	4	1	0	9	1	3	5	2

1	0	-1

	-3	1	4	-8	-1				

Linear Filtering in 1D

1	**2**	**4**	**1**	**0**	**9**	**1**	**3**	**5**	**2**

1	**0**	**-1**

	-3	**1**	**4**	**-8**	**-1**	**?**			

Linear Filtering in 1D

1	**2**	**4**	**1**	**0**	**9**	**1**	**3**	**5**	**2**

1	**0**	**-1**

	-3	**1**	**4**	**-8**	**-1**	**6**			

Linear Filtering in 1D

1	2	4	1	0	9	1	3	5	2

						1	0	-1	

	-3	1	4	-8	-1	6	?		

Linear Filtering in 1D

1	**2**	**4**	**1**	**0**	**9**	**1**	**3**	**5**	**2**

1	**0**	**-1**

	-3	**1**	**4**	**-8**	**-1**	**6**	**-4**		

Linear Filtering in 1D

1	**2**	**4**	**1**	**0**	**9**	**1**	**3**	**5**	**2**

1	**0**	**-1**

Result

	-3	**1**	**4**	**-8**	**-1**	**6**	**-4**	**?**	

Linear Filtering in 1D

1	2	4	1	0	9	1	3	5	2

1	0	-1

Result

	-3	1	4	-8	-1	6	-4	1	

Linear Filtering in 1D

1	**2**	**4**	**1**	**0**	**9**	**1**	**3**	**5**	**2**

1	**0**	**-1**

half-width = 1

Filter width = 2 x (half width) + 1

= 3

Result

	-3	**1**	**4**	**-8**	**-1**	**6**	**-4**	**1**	

Cross-correlation: $CC(i)$

$f =$

1	3	4	1	10	3	0	1

$h =$

1	0	-1

$$CC(i) = \sum_{k \in [-w,w]} \mathbf{f}(i+k)\mathbf{h}(k)$$

Half-width w

Convolution $f * h$

$f =$

1	**3**	**4**	**1**	**10**	**3**	**0**	**1**

$h =$

1	**0**	**-1**

$$(\mathbf{f} * \mathbf{k})_i = \sum_{k \in [-w, w]} \mathbf{f}(i-k)\mathbf{h}(k)$$

Convolution $f * h$

$f =$

1	3	4	1	10	3	0	1

$h =$

1	0	-1

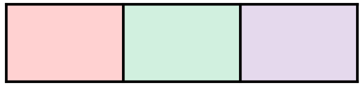
$$(\mathbf{f} * \mathbf{k})_i = \sum_{k \in [-w,w]} \mathbf{f}(i-k)\mathbf{h}(k)$$

Filter is flipped

Linear Filtering in 1D

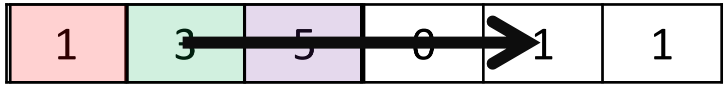
Signal					
1	3	5	0	1	1

Kernel/Filter



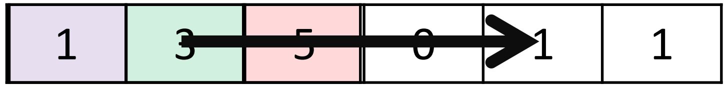
Cross-correlation

$$CC(i) = \sum_{k \in [-w,w]} \mathbf{f}(i+k)\mathbf{h}(k)$$



Convolution

$$(\mathbf{f} * \mathbf{k})_i = \sum_{k \in [-w,w]} \mathbf{f}(i-k)\mathbf{h}(k)$$



Filter half-width

$f =$

1	**3**	**4**	**1**	**10**	**3**	**0**	**1**

$h =$

What is the half-width of this filter h?

Filter half-width

$f =$

1	**3**	**4**	**1**	**10**	**3**	**0**	**1**

$h =$

Filter width = 2 x (half width) + 1

7 = 2 x (3) + 1

What is the half-width of this filter h? (Answer is 3)

Sometimes it is called a 7-tap filter

Linear Filtering in 1 d

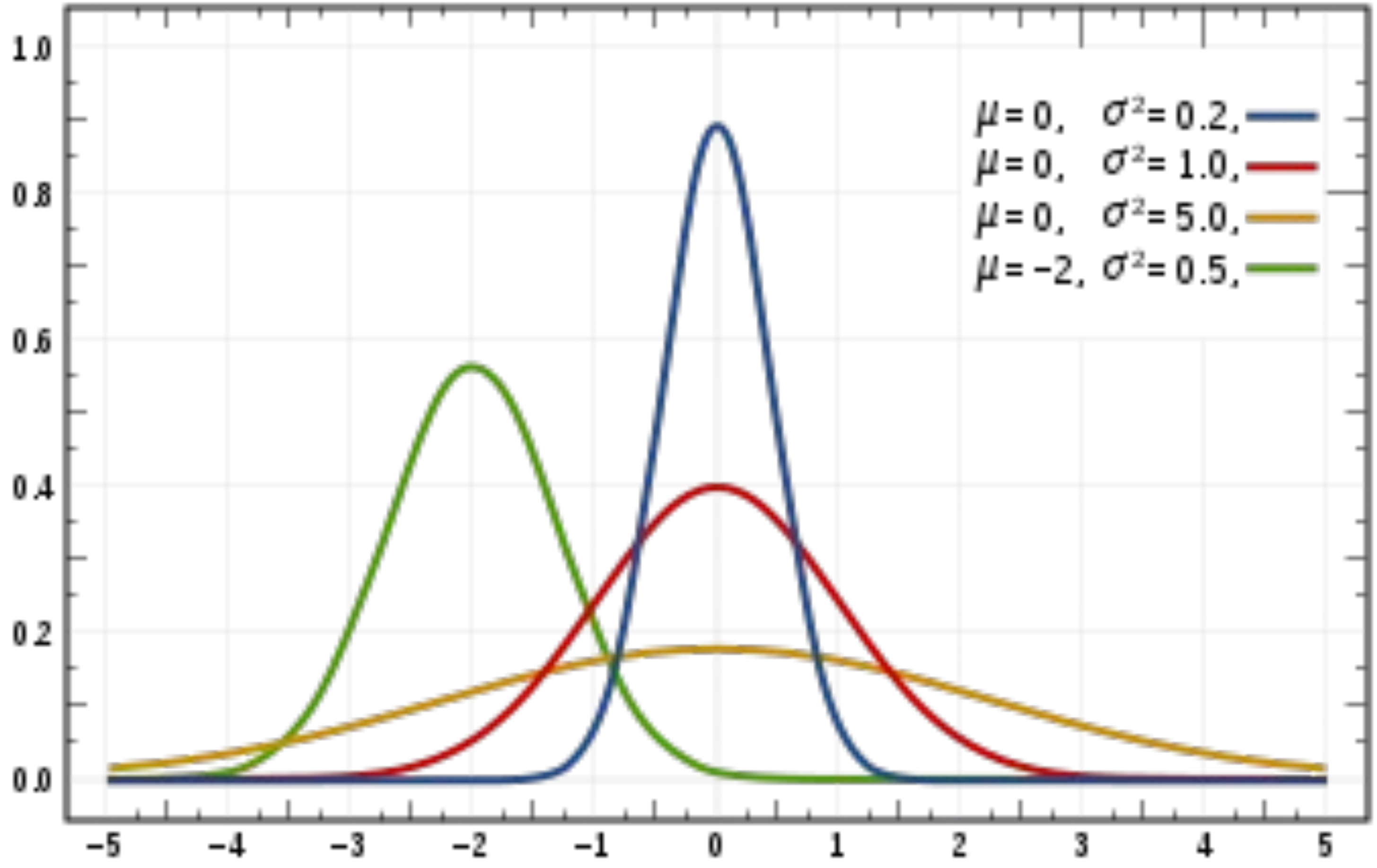
- Signal: f
- Kernel (sometimes called mask or filter): h
- Half-width of kernel: w

Cross-correlation $$CC(i) = \sum_{k \in [-w,w]} \mathbf{f}(i+k)\mathbf{h}(k)$$

Convolution $$(\mathbf{f} * \mathbf{k})_i = \sum_{k \in [-w,w]} \mathbf{f}(i-k)\mathbf{h}(k)$$

Gaussian in 1D

$$G(x;\mu,\sigma^2)$$



$$x$$

From Wikipedia

Gaussian in 1D

mean

$$G(x;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}}\exp^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

variance

standard deviation

Mean (μ) and and variance (σ^2)

Given data points $\boldsymbol{x} = \{x_1, x_2, \cdots, x_N\}$

Mean

$$\mu = E[\boldsymbol{x}] = \frac{1}{N}\sum_{i=1}^{N} x_i$$

Variance

$$\sigma^2 = E\left[(\boldsymbol{x} - \boldsymbol{\mu})^2\right] = \frac{1}{N}\sum_{i=1}^{N} (x_i - \mu)^2$$

Setup a 5-tap Gaussian filter

$$G(x;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}}\exp^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Setup a 5-tap Gaussian filter

What information is missing?

$$G(x; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Setup a 5-tap Gaussian filter

$$G(x;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}}\exp^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

What information is missing?

$$\sigma = 2$$

Setup a 5-tap Gaussian filter

$$G(x;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

What information is missing?

$\sigma = 2$

$\mu = 0$

$x = -2$	$x = -1$	$x = 0$	$x = 1$	$x = 2$

Setup a 5-tap Gaussian filter

$$G(x;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}}\exp^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

What information is missing?

$\sigma = 2$

$\mu = 0$

$\frac{(-2-0)^2}{(2)(2)^2}$	$\frac{(-1-0)^2}{(2)(2)^2}$	$\frac{(0-0)^2}{(2)(2)^2}$	$\frac{(1-0)^2}{(2)(2)^2}$	$\frac{(2-0)^2}{(2)(2)^2}$
$x=-2$	$x=-1$	$x=0$	$x=1$	$x=2$

Setup a 5-tap Gaussian filter

$$G(x; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

What information is missing?

$\sigma = 2$

$\mu = 0$

$x = -2$	$x = -1$	$x = 0$	$x = 1$	$x = 2$
$-\frac{(-2-0)^2}{(2)(2)^2} = -\frac{4}{8}$	$-\frac{(-1-0)^2}{(2)(2)^2} = -\frac{1}{8}$	$-\frac{(0-0)^2}{(2)(2)^2} = -\frac{0}{8}$	$-\frac{(1-0)^2}{(2)(2)^2} = -\frac{1}{8}$	$-\frac{(2-0)^2}{(2)(2)^2} = -\frac{4}{8}$

Setup a 5-tap Gaussian filter

$$G(x;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}}\exp^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

What information is missing?

$\sigma = 2$

$\mu = 0$

$\exp(-0.5)$	$\exp(-0.125)$	$\exp(0)$	$\exp(-0.125)$	$\exp(-0.5)$
$x = -2$	$x = -1$	$x = 0$	$x = 1$	$x = 2$

Setup a 5-tap Gaussian filter

$$G(x;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}}\exp^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

What information is missing?

$\sigma = 2$

$\mu = 0$

$\exp(-0.5)$ ≈ 0.607	$\exp(-0.125)$ ≈ 0.882	$\exp(0)$ $= 1$	$\exp(-0.125)$ ≈ 0.882	$\exp(-0.5)$ ≈ 0.607
$x = -2$	$x = -1$	$x = 0$	$x = 1$	$x = 2$

Setup a 5-tap Gaussian filter

$$G(x; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

What information is missing?

$\sigma = 2$

$\mu = 0$

$x = -2$	$x = -1$	$x = 0$	$x = 1$	$x = 2$
$\exp(-0.5)$ ≈ 0.607	$\exp(-0.125)$ ≈ 0.882	$\exp(0)$ $= 1$	$\exp(-0.125)$ ≈ 0.882	$\exp(-0.5)$ ≈ 0.607
?	?	?	?	?

Fit a Gaussian to the following data

1	**2**	**-1**	**0**	**4**	**5**	**3**	**6**	**1**	**2**

on linear filtering

Linear Filtering in 2D

1	3	5	0	1	1
2	1	-1	3	0	4
0	1	2	2	3	4
1	3	4	9	10	3
3	7	8	2	1	4
1	5	6	4	2	1

Image

Kernel

Linear Filtering in 2D

CC (Image , Kernel) =

1	3	5	0	1	1
2	1	-1	3	0	4
0	1	2	2	3	4
1	3	4	9	10	3
3	7	8	2	1	4
1	5	6	4	2	1

Image

Kernel

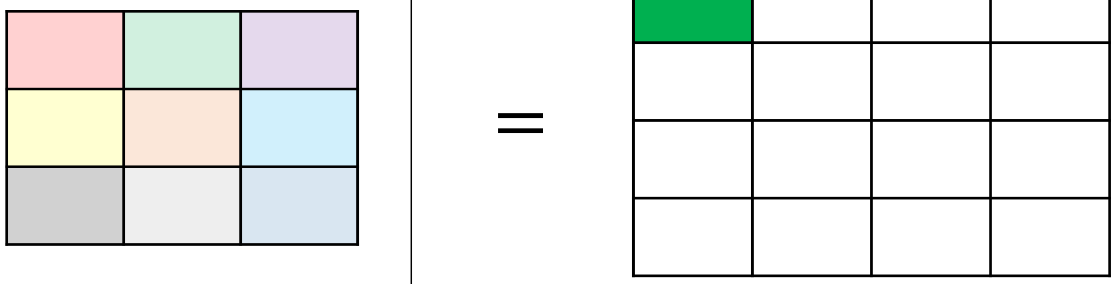
?			

Linear Filtering in 2D

CC

1	3	5	0	1	1
2	1	-1	3	0	4
0	1	2	2	3	4
1	3	4	9	10	3
3	7	8	2	1	4
1	5	6	4	2	1

Image



Kernel

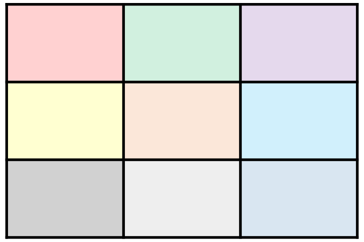
Linear Filtering in 2D

CC (

1	3	5	0	1	1
2	1	-1	3	0	4
0	1	2	2	3	4
1	3	4	9	10	3
3	7	8	2	1	4
1	5	6	4	2	1

Image

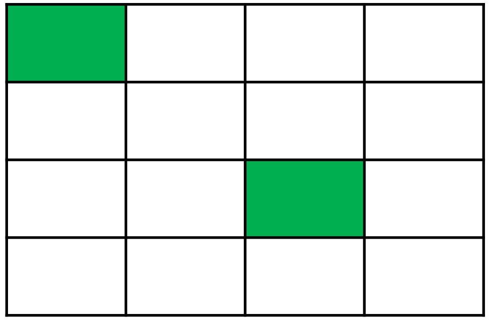
,



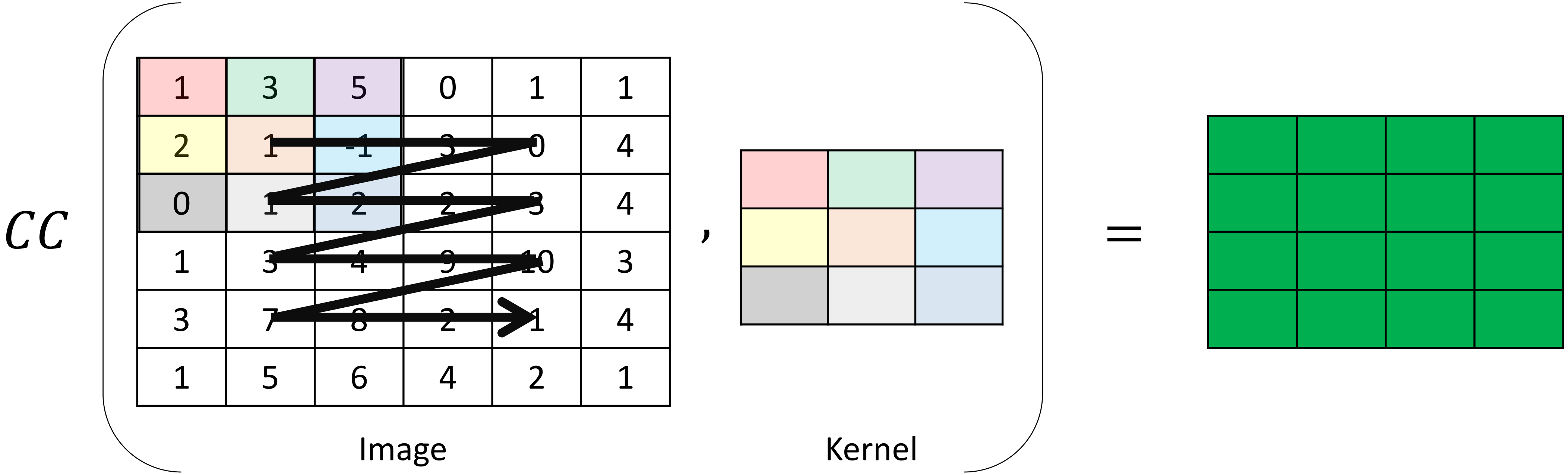
Kernel

)

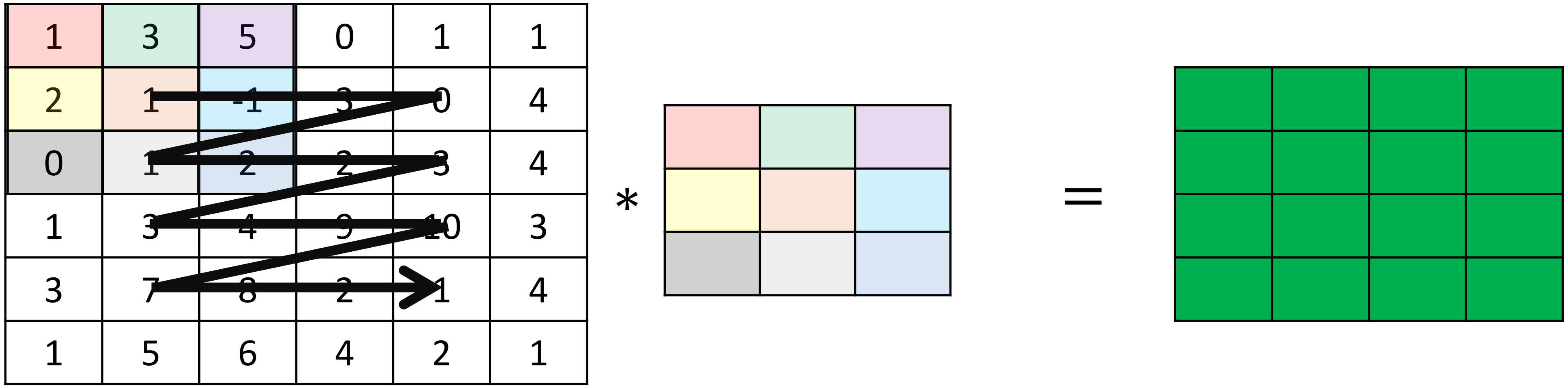
=



Linear Filtering in 2D



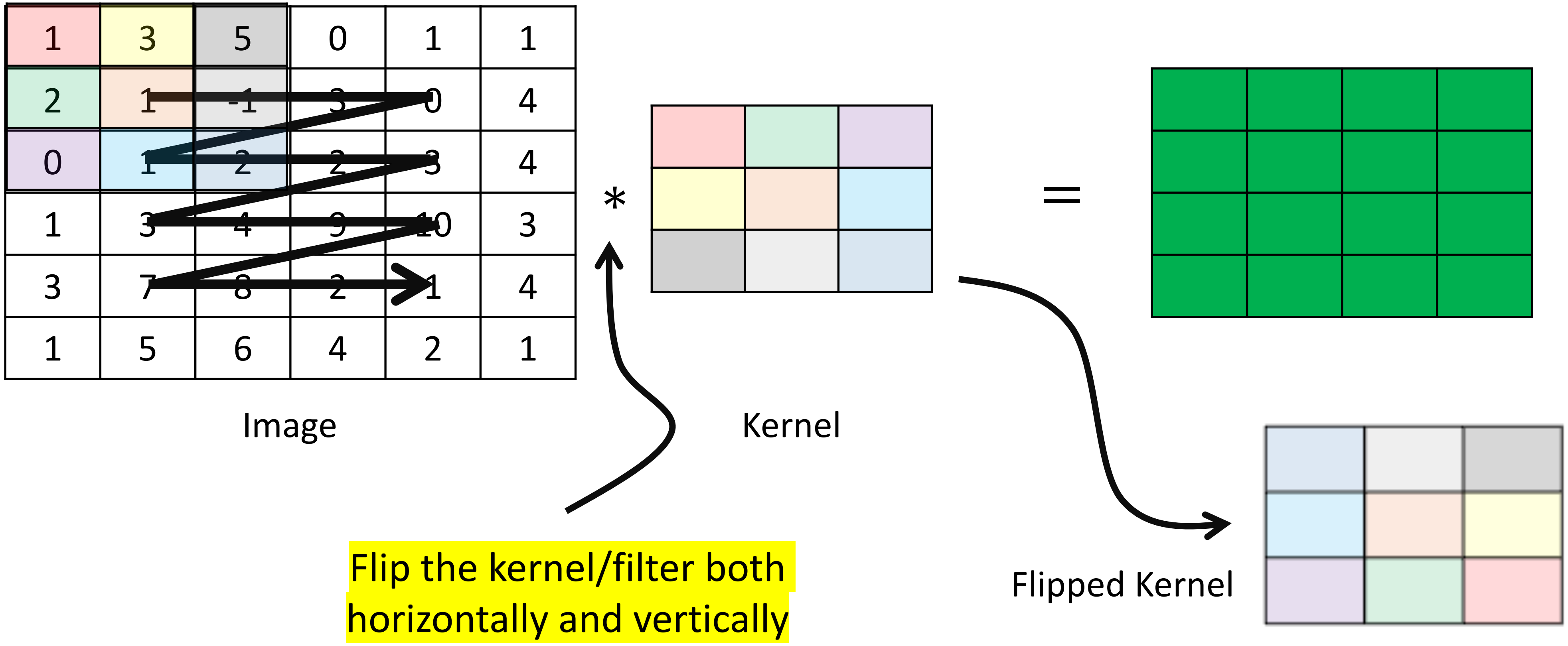
Linear Filtering in 2D



Image

Kernel

Linear Filtering in 2D



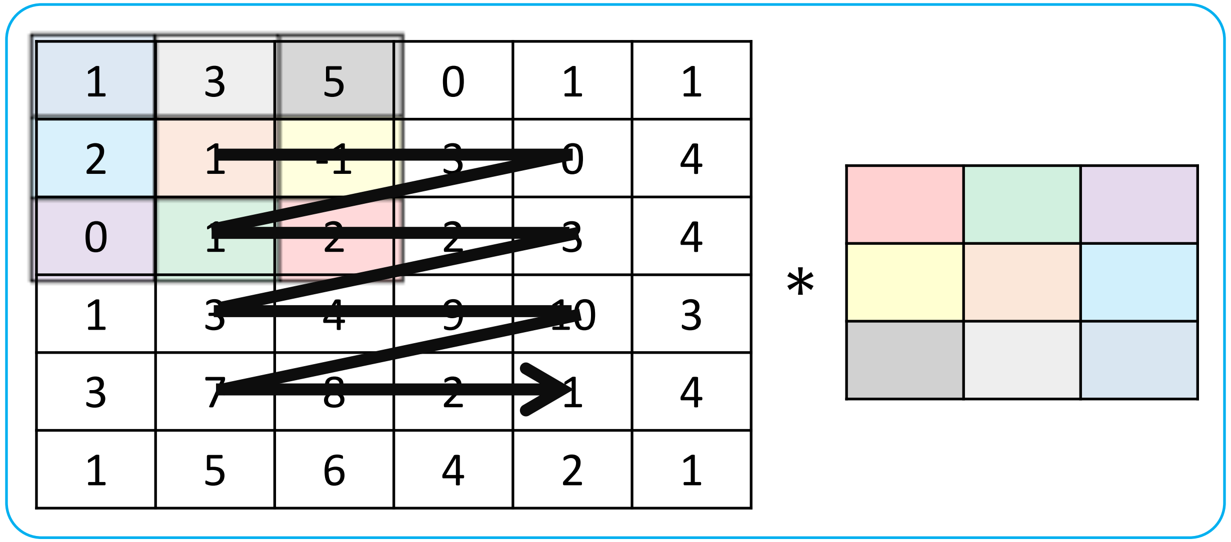
Linear Filtering in 2D

Cross-correlation

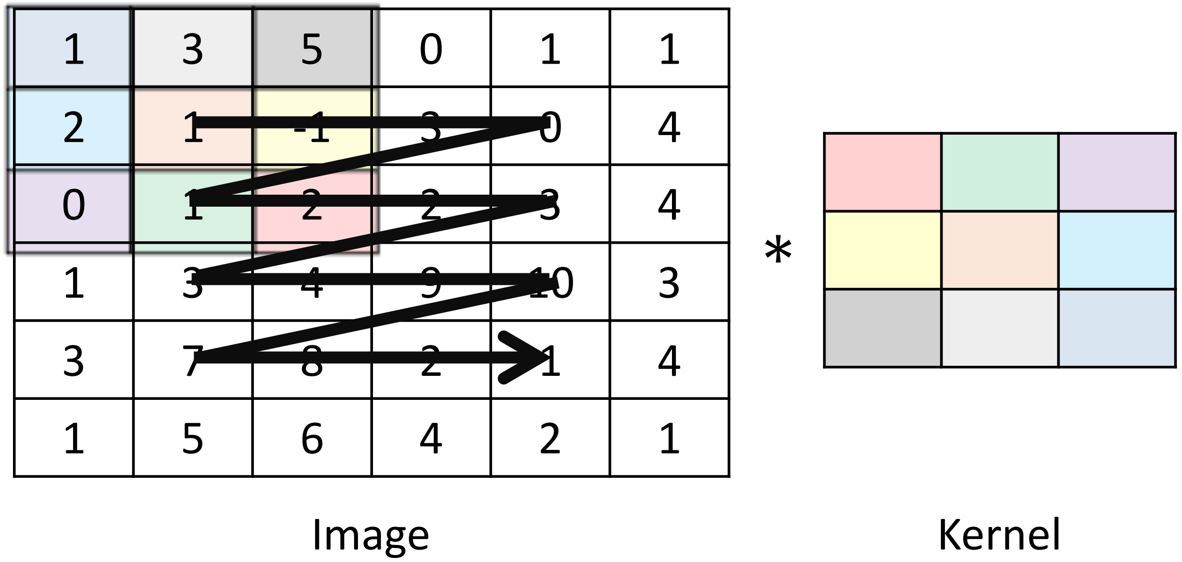
$$CC(i,j) = \sum_{\substack{k \in [-w,w] \\ l \in [-h,h]}} \mathbf{f}(i+k, j+l)\mathbf{h}(k,l)$$

Convolution

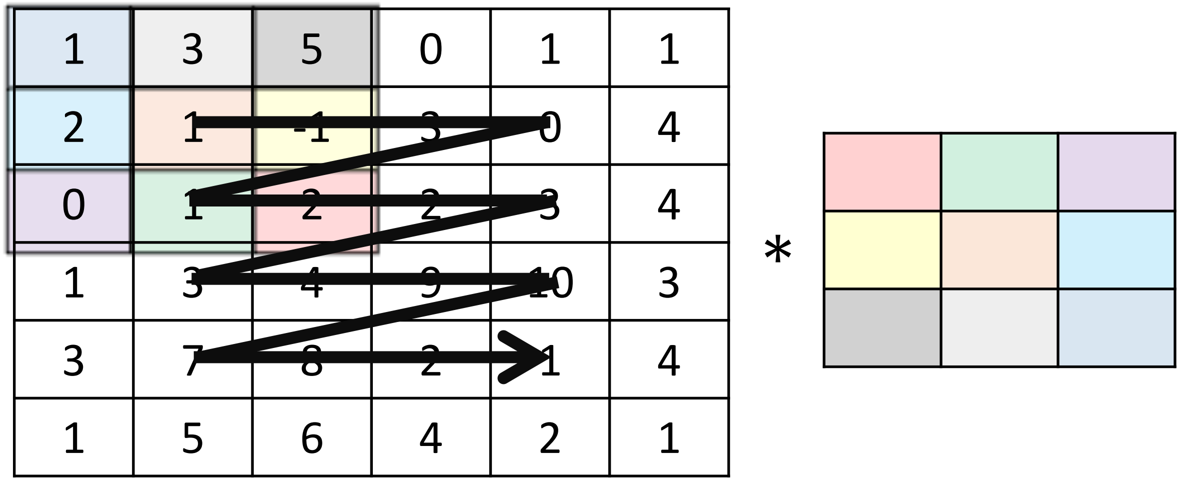
$$(\mathbf{f} * \mathbf{k})_{i,j} == \sum_{\substack{k \in [-w,w] \\ l \in [-h,h]}} \mathbf{f}(i-k, j-l)\mathbf{h}(k,l)$$



Number of multiplications and additions



Number of multiplications and additions



Image

Kernel

#locations = (4)(4)

#multiplications at each location = 9

#additions at each location = 8

#total = (9)(4)(4) multiplications

(8)(4)(4) additions

Multivariate Guassian (in k-dimensions)

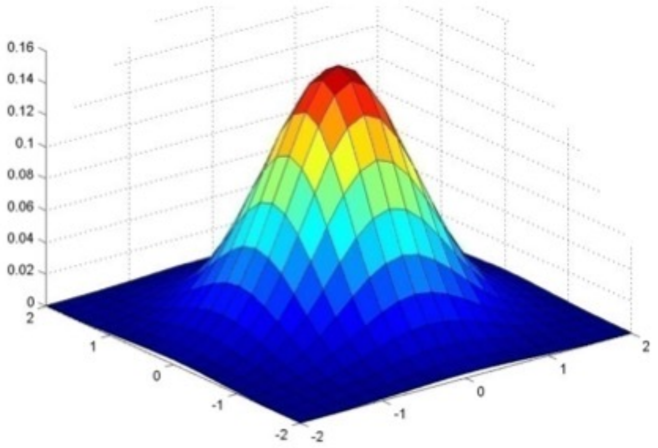
$$G(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma) = \frac{1}{\sqrt{(2\pi)^k \det(\Sigma)}} exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu})\right)$$

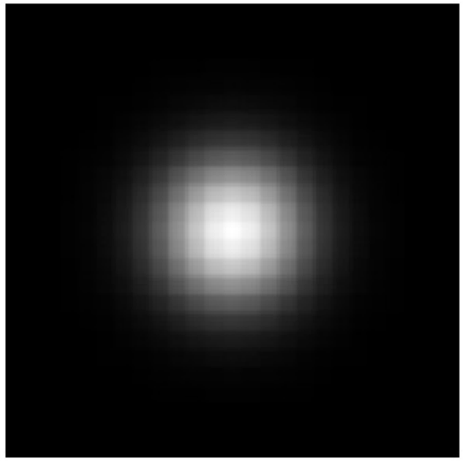
where

$\boldsymbol{x} \in R^k$

$\boldsymbol{\mu} \in R^k$

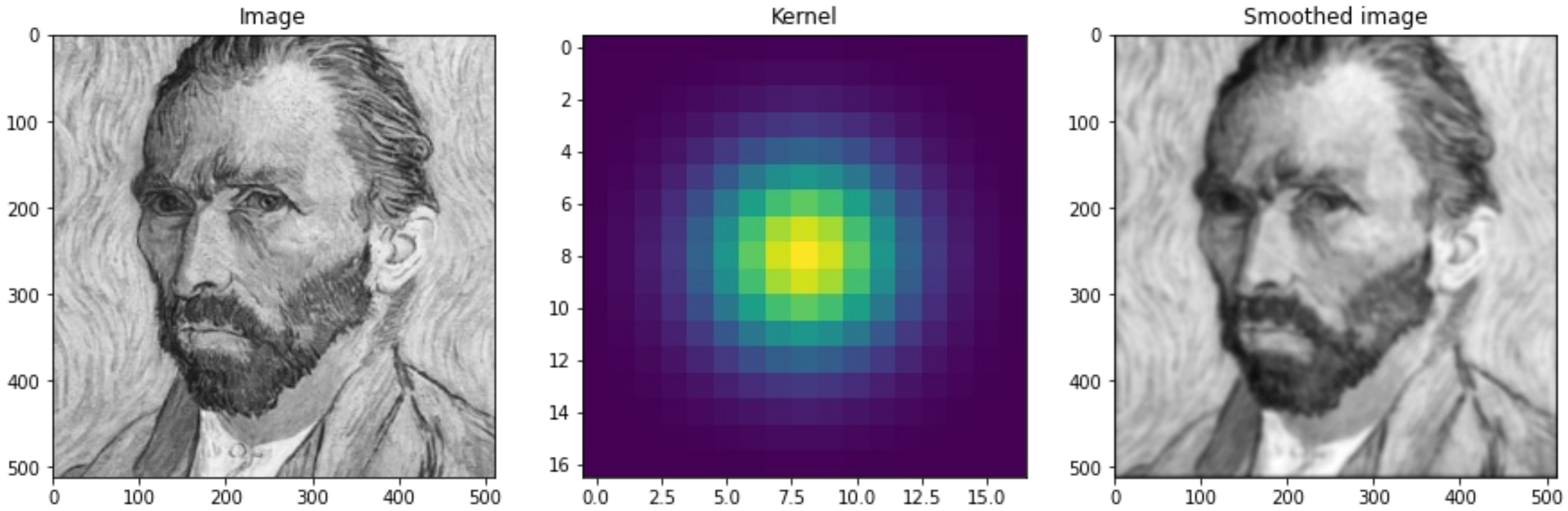
$\Sigma \in R^{k \times k}$





Gaussian in 2D

Gaussian Blurring

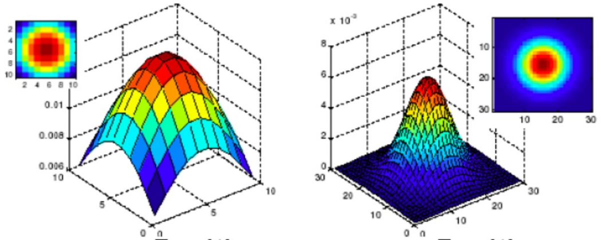


Gaussian Blurring

- We often use the following approximation of a Gaussian function

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

- Gaussian functions have infinite support, but discrete Gaussian kernels are finite

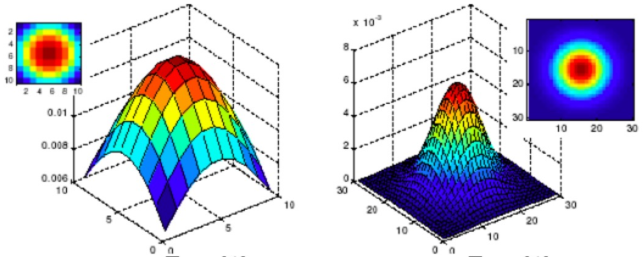


σ = 5 with 10 x 10 kernel

σ = 5 with 30 x 30 kernel

Gaussian Blurring

- Variance controls how broad or peaky the filter is



σ = 5 with 10 x 10 kernel

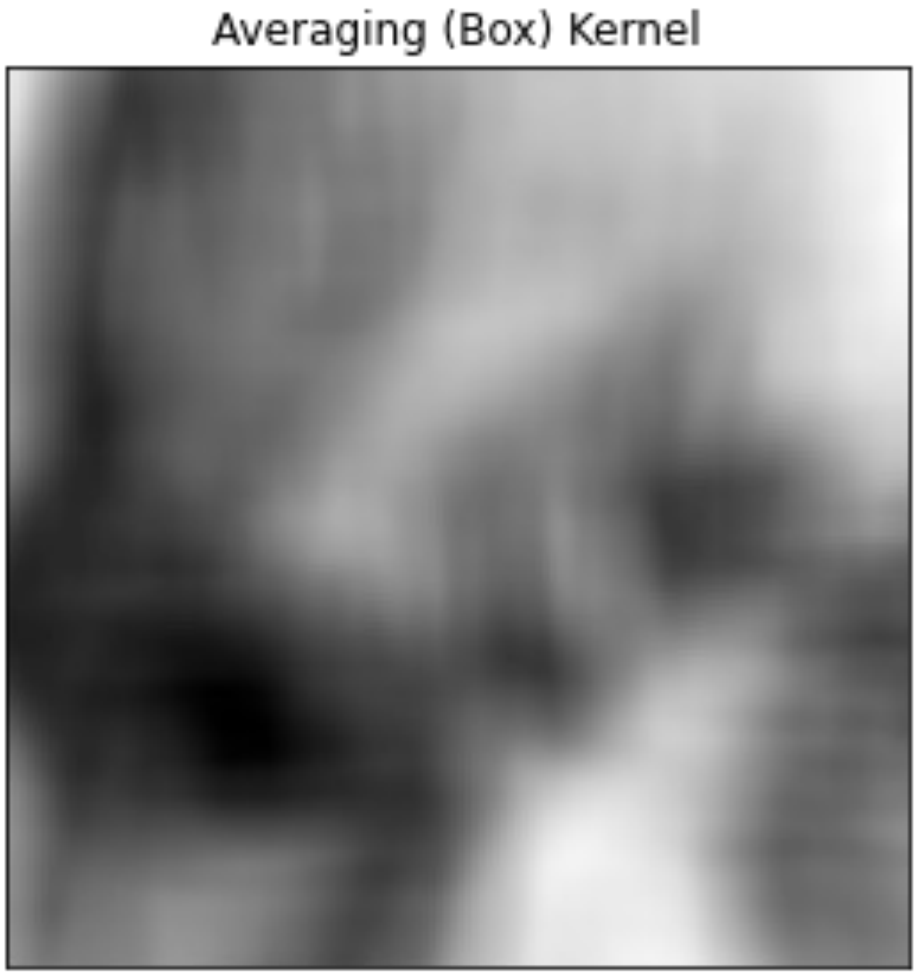
σ = 5 with 30 x 30 kernel

- Removes high-frequency components from the image
  - Blurs the image
  - Acts as a low-pass filter

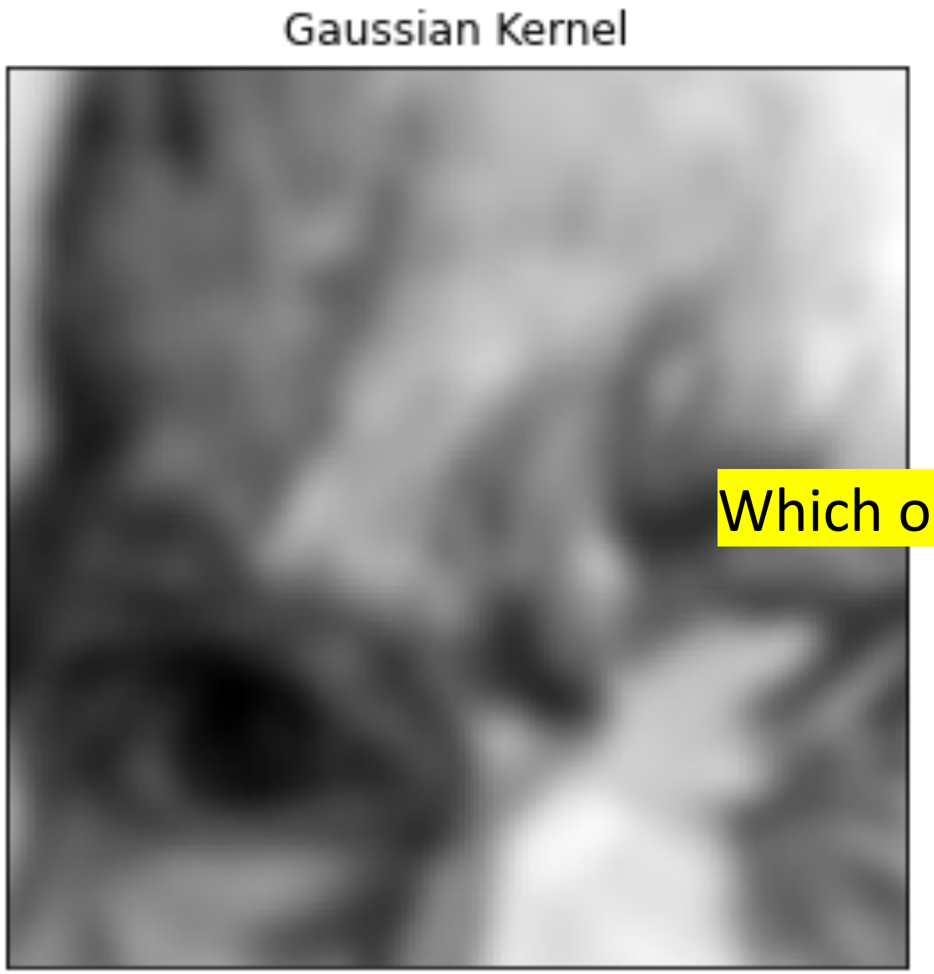
Gaussian Blurring

- Convolving twice with Gaussian kernel of width σ^2 is the same as convolving once with kernel of width $\sigma\sqrt{2}$
- Applying a Gaussian filter with variance σ_1^2, followed by applying a Gaussian filter with variance σ_2^2 is the same as applying once with Gaussian filter with variance $\sqrt{\sigma_1^2 + \sigma_2^2}$
- All values are positive
- Values sum to 1?
  - Why is this relevant?
- This size of the filter, plus its variance, determines the extent of smoothing

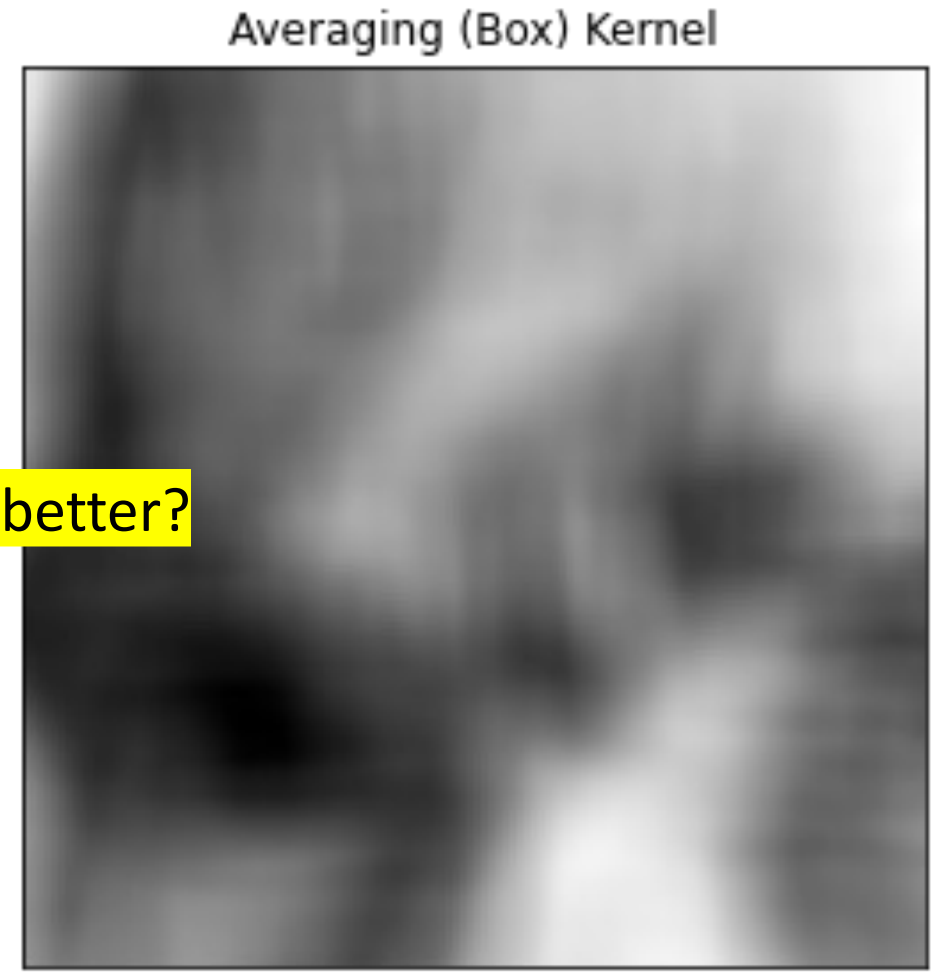
Gaussian Blurring vs. Average (Box) Filtering



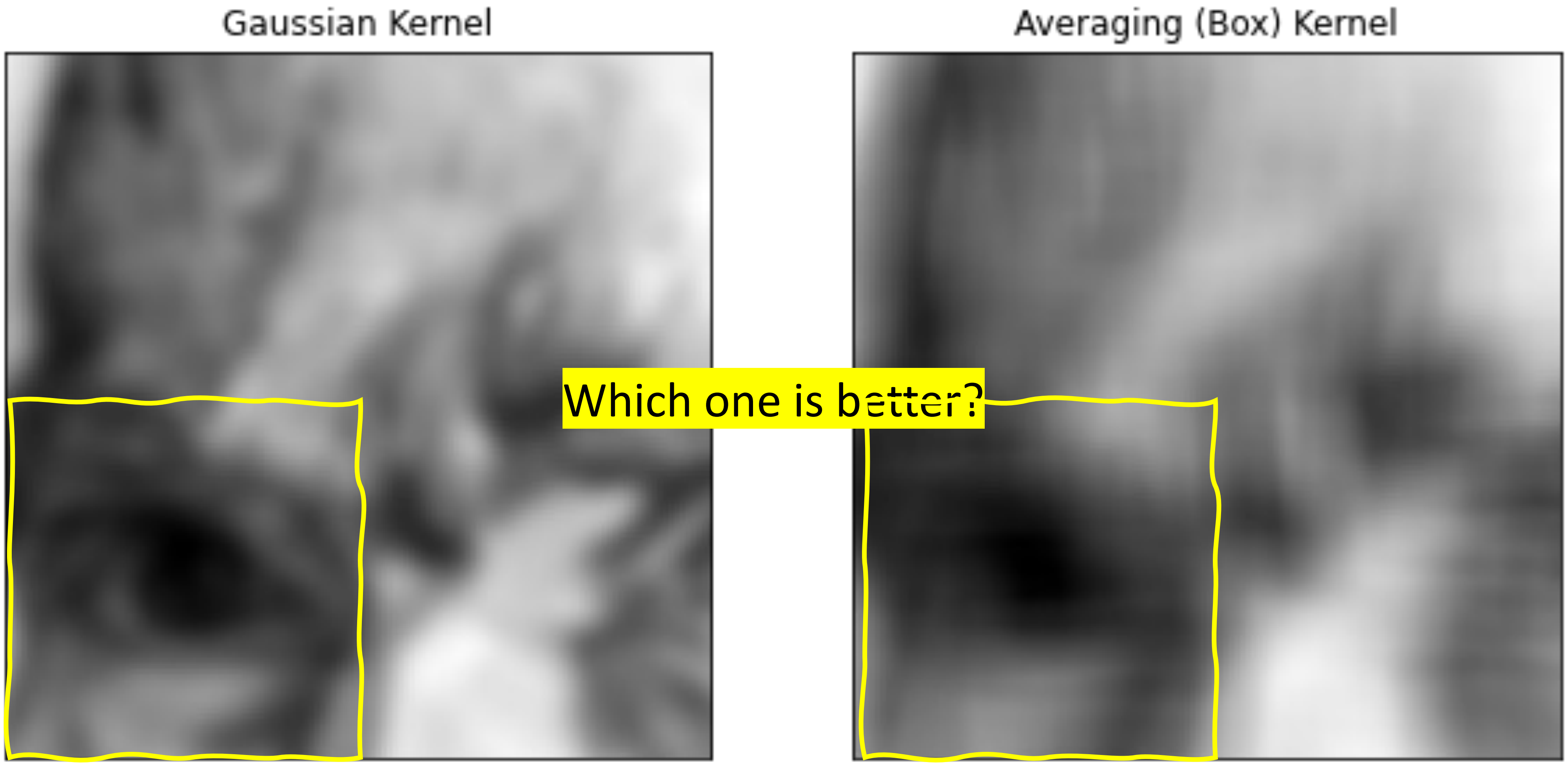
Gaussian Blurring vs. Average (Box) Filtering



Which one is better?



Gaussian Blurring vs. Average (Box) Filtering



Which one is better?

Separability

- An n-dimensional filter that can be expressed as an outer-product of n 1-dimensional filters is called a separable filter

Outer-Product and Inner-Product

$a = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ and $b = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

Outer-Product and Inner-Product

$a = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ and $b = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

Inner-Product = $a^T b = (1)(1) + (2)(0) = 1$

Outer-Product and Inner-Product

$a = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ and $b = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

Inner-Product = $a^T b = (1)(1) + (2)(0) = 1$

Outer-Product = $ab^T = \begin{bmatrix} 1 & 0 \\ 2 & 0 \end{bmatrix}$

Outer-Product and Inner-Product

$a = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$ and $b = \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix}$

Exercise

Separability

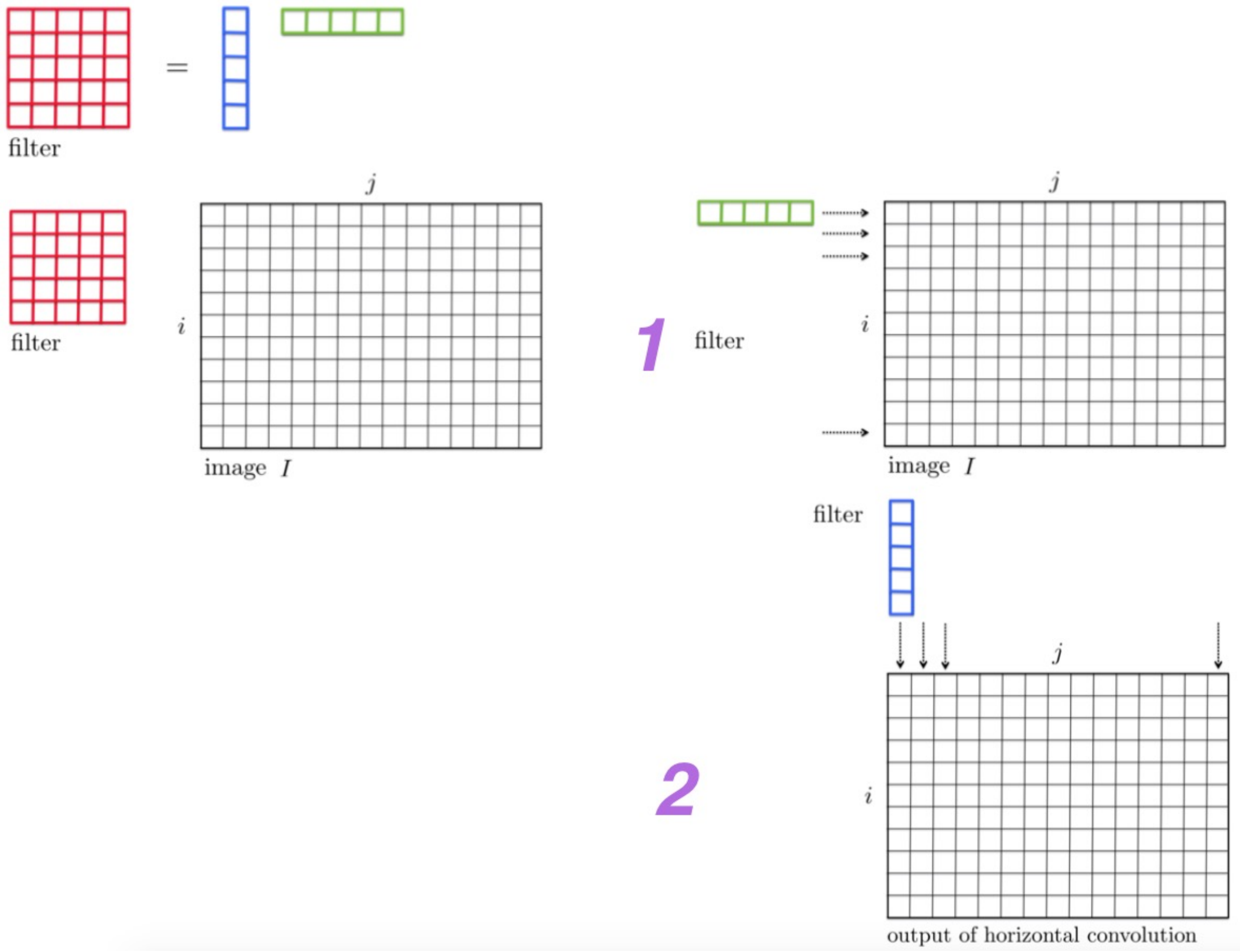
- An n-dimensional filter that can be expressed as an outer-product of n 1-dimensional filters is called a separable filter

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

Convolution with Separable Filters in 2D

- [Step 1] Perform row-wise convolution with horizontal filter
- [Step 2] Perform column-wise convolution with the results obtained in step 1 with vertical filter

Convolution with Separable Filters in 2D



Convolution with Separable Filters in 2 d

Signal $\begin{bmatrix} 1 & 0 & -2 & 2 \\ 2 & -1 & 6 & 1 \\ 3 & 0 & 1 & 3 \end{bmatrix}$

Filter/Kernel $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$

Convolution with Separable Filters in 2 d

Signal $\begin{bmatrix} 1 & 0 & -2 & 2 \\ 2 & -1 & 6 & 1 \\ 3 & 0 & 1 & 3 \end{bmatrix}$ Filter/Kernel $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$

Say we want to compute the response at location $(1,1)$, highlighted above.

Convolution with Separable Filters in 2 d

Signal $\begin{bmatrix} 1 & 0 & -2 & 2 \\ 2 & -1 & 6 & 1 \\ 3 & 0 & 1 & 3 \end{bmatrix}$ Filter/Kernel $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$

Say we want to compute the response at location $(1,1)$, highlighted above.

Using 2D convolution (without exploiting separability)

$$(1)(1) + (0)(2) + (-2)(1) + (2)(2) + (-1)(4) + (6)(2) + (3)(1) + (0)(2) + (1)(1)$$
$$= 1 + 0 - 2 + 4 - 4 + 12 + 3 + 0 + 1$$
$$= 15$$

A

Convolution with Separable Filters in 2 d

Signal $\begin{bmatrix} 1 & 0 & -2 & 2 \\ 2 & -1 & 6 & 1 \\ 3 & 0 & 1 & 3 \end{bmatrix}$ Filter/Kernel $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$

Say we want to compute the response at location $(1,1)$, highlighted above.

Using 2D convolution (exploiting separability)

Step 1: use horizontal filter

$$\begin{bmatrix} (1)(1) + (0)(2) + (-2)(1) \\ (2)(1) + (-1)(2) + (6)(1) \\ (3)(1) + (0)(2) + (1)(1) \end{bmatrix}$$

$$= \begin{bmatrix} -1 \\ 6 \\ 4 \end{bmatrix}$$

Convolution with Separable Filters in 2 d

Signal $\begin{bmatrix} 1 & 0 & -2 & 2 \\ 2 & -1 & 6 & 1 \\ 3 & 0 & 1 & 3 \end{bmatrix}$ Filter/Kernel $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$

Say we want to compute the response at location $(1,1)$, highlighted above.

Using 2D convolution (exploiting separability)

Step 1: use horizontal filter to

$$\begin{bmatrix} (1)(1) + (0)(2) + (-2)(1) \\ (2)(1) + (-1)(2) + (6)(1) \\ (3)(1) + (0)(2) + (1)(1) \end{bmatrix}$$

$$= \begin{bmatrix} -1 \\ 6 \\ 4 \end{bmatrix}$$

Step 2: use vertical filter

$$(-1)(1) + (6)(2) + (4)(1)$$

$$= 15$$

B

Check that this is the same value as in

A

Computational Considerations

- For non-separable filters

$$O(w_k \times h_k \times w \times h)$$

- For separable filters

$$O(w_k \times w \times h) + O(w_h \times w \times h)$$

Signal $\begin{bmatrix} 1 & 0 & -2 & 2 \\ 2 & -1 & 6 & 1 \\ 3 & 0 & 1 & 3 \end{bmatrix}$

Filter/Kernel $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$

Computational Considerations

- For non-separable filters

$$O(w_k \times h_k \times w \times h)$$

- For separable filters

$$O(w_k \times w \times h) + O(w_h \times w \times h)$$

Signal $\begin{bmatrix} 1 & 0 & -2 & 2 \\ 2 & -1 & 6 & 1 \\ 3 & 0 & 1 & 3 \end{bmatrix}$

Filter/Kernel $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$

Where possible exploit separability to speed up convolutions

Gaussian filter is separable

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$
$$= \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{x^2}{2\sigma^2}\right)\cdot\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{y^2}{2\sigma^2}\right)$$
$$= g_\sigma(x)\cdot g_\sigma(y)$$

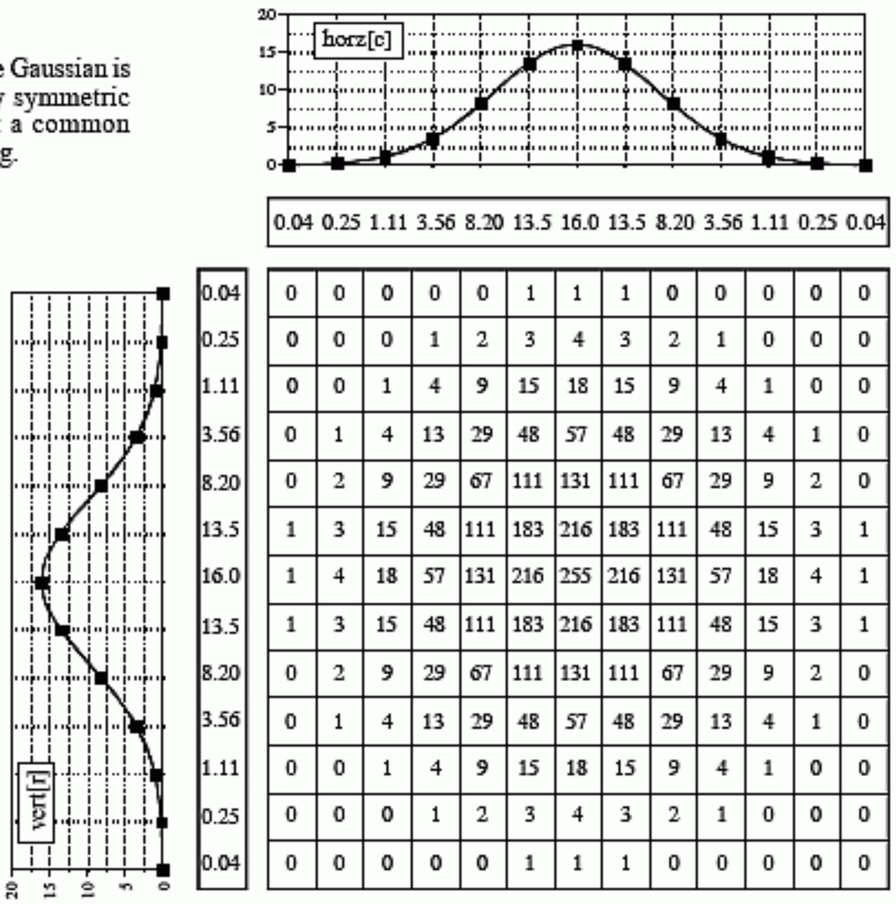


FIGURE 24-7
Separation of the Gaussian. The Gaussian is the only PSF that is circularly symmetric *and* separable. This makes it a common filter kernel in image processing.

	0.04	0.25	1.11	3.56	8.20	13.5	16.0	13.5	8.20	3.56	1.11	0.25	0.04
0.04	0	0	0	0	0	1	1	1	0	0	0	0	0
0.25	0	0	0	1	2	3	4	3	2	1	0	0	0
1.11	0	0	1	4	9	15	18	15	9	4	1	0	0
3.56	0	1	4	13	29	48	57	48	29	13	4	1	0
8.20	0	2	9	29	67	111	131	111	67	29	9	2	0
13.5	1	3	15	48	111	183	216	183	111	48	15	3	1
16.0	1	4	18	57	131	216	255	216	131	57	18	4	1
13.5	1	3	15	48	111	183	216	183	111	48	15	3	1
8.20	0	2	9	29	67	111	131	111	67	29	9	2	0
3.56	0	1	4	13	29	48	57	48	29	13	4	1	0
1.11	0	0	1	4	9	15	18	15	9	4	1	0	0
0.25	0	0	0	1	2	3	4	3	2	1	0	0	0
0.04	0	0	0	0	0	1	1	1	0	0	0	0	0

The Scientist and Engineer's Guide to Digital Signal Processing
By Steven W. Smith, Ph.D.

Singular value decomposition

Factor a matrix M as follows: $M = U\Sigma V^T = \sum_i \sigma_{ii} \boldsymbol{u}_i \boldsymbol{v}_i^T$

$$U = [\boldsymbol{u}_1 \quad \cdots \quad \boldsymbol{u}_m] \qquad \Sigma = \begin{bmatrix} \sigma_{11} & \\ & \ddots \end{bmatrix} \qquad V^T = \begin{bmatrix} \boldsymbol{v}_1^T \\ \vdots \\ \boldsymbol{v}_n^T \end{bmatrix}$$

$M \in R^{m \times n}$

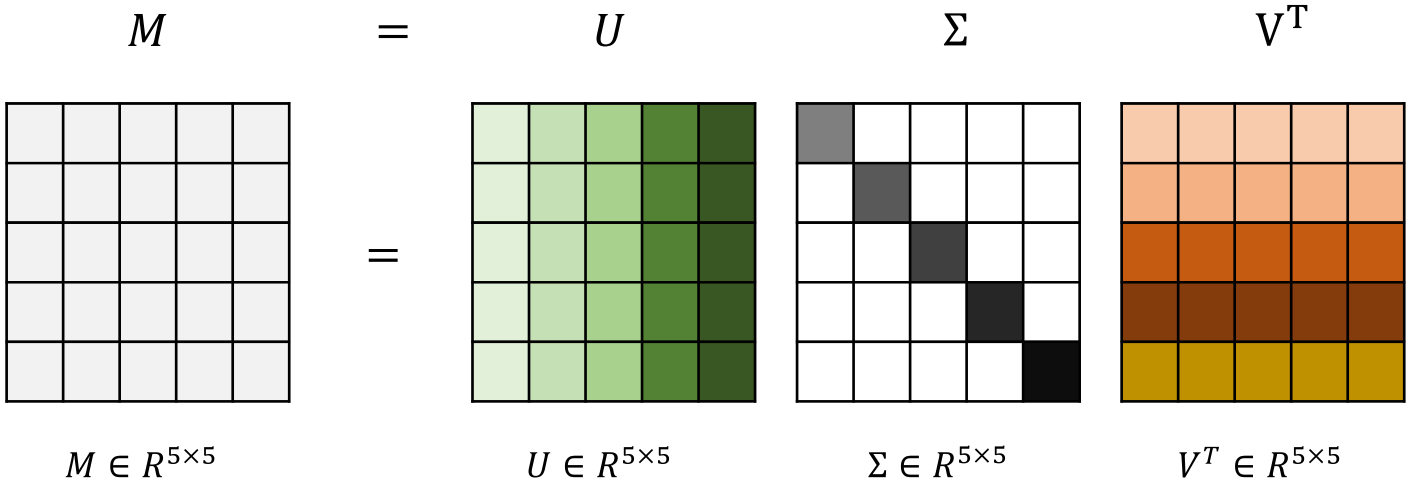
$U \in R^{m \times m}$

$\Sigma \in R^{m \times n}$ is a rectangular diagonal matrix. σ_{ii} contains the singular values

$V^T \in R^{n \times n}$

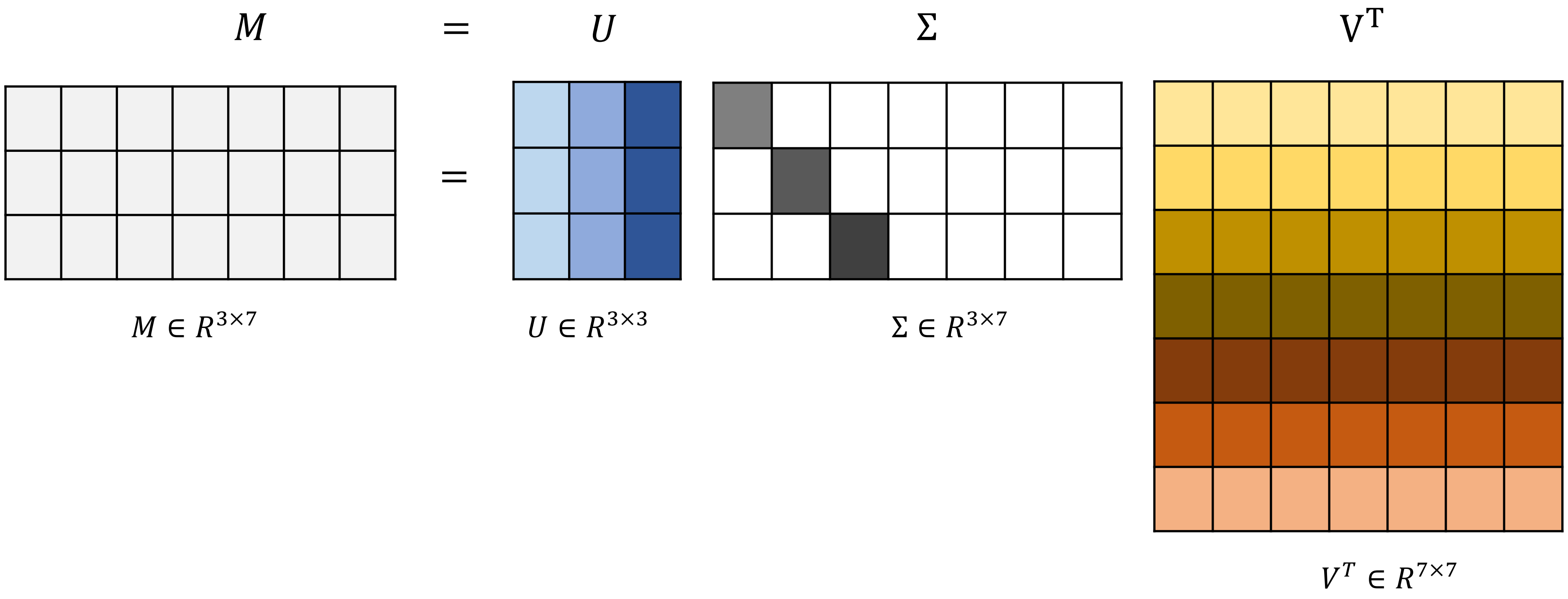
Singular value decomposition

Matrix M is square



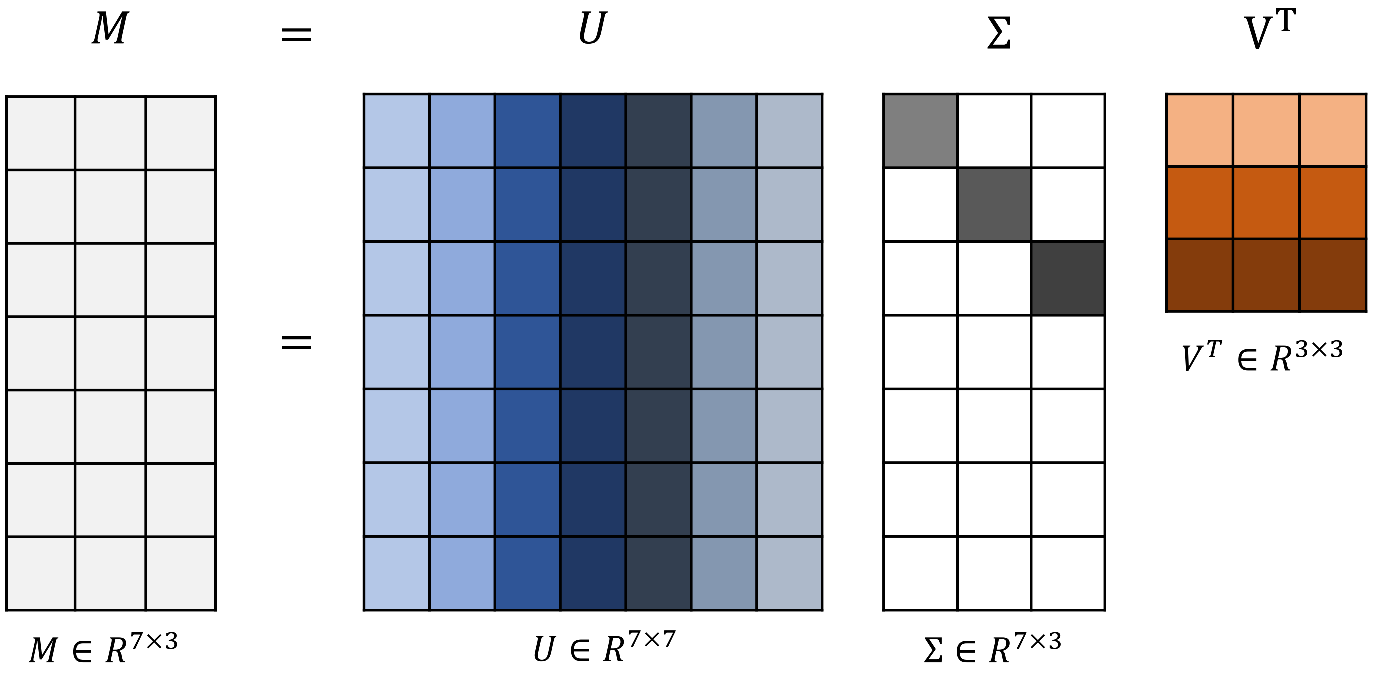
Singular value decomposition

Matrix M is wide



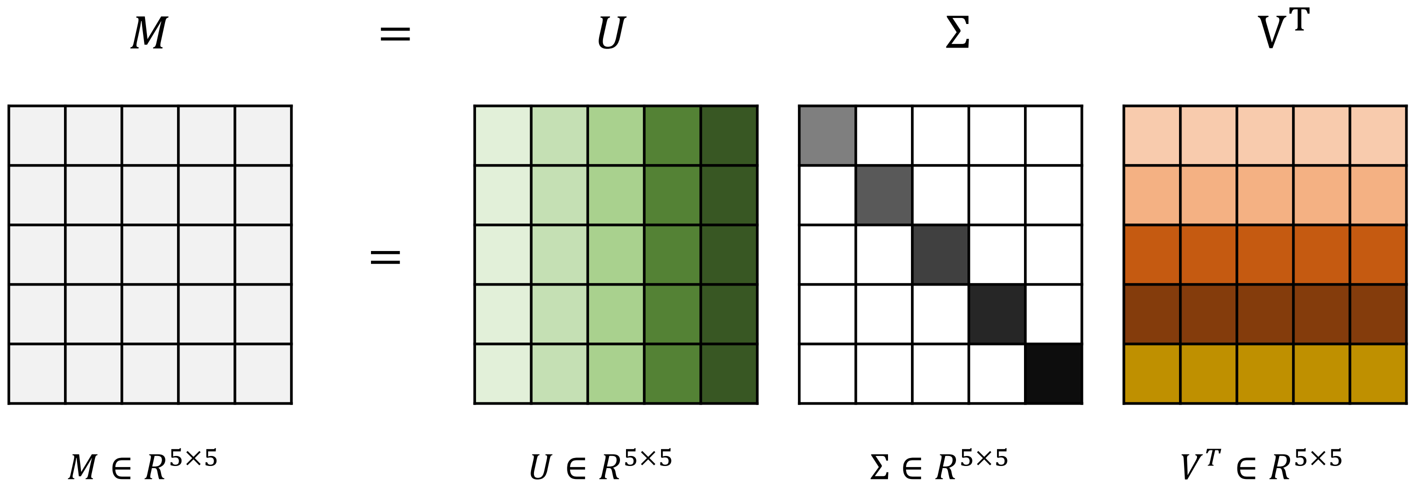
Singular value decomposition

Matrix M is tall



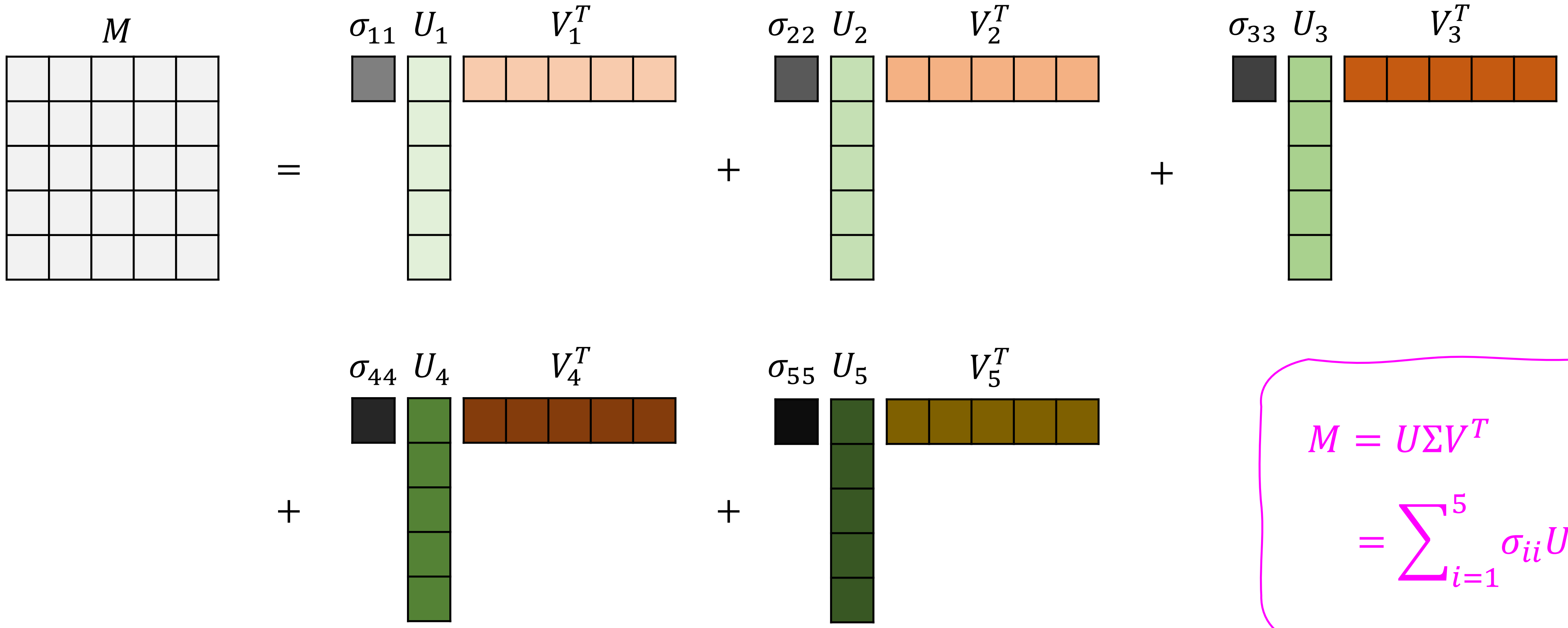
Singular value decomposition

Expressed as a sum of scaled outer-products between columns of U and rows of V^T



Singular value decomposition

Expressed as a sum of scaled outer-products between columns of U and rows of V^T



$$M = U\Sigma V^T$$

$$= \sum_{i=1}^{5} \sigma_{ii} U_i V_i^T$$

How to find if a 2D filter is separable?

- Use Singular Value Decomposition (SVD)
  - If only one singular value is non-zero then the 2D filter is separable

- [Step 1] Compute SVD and check if only one singular value is non-zero

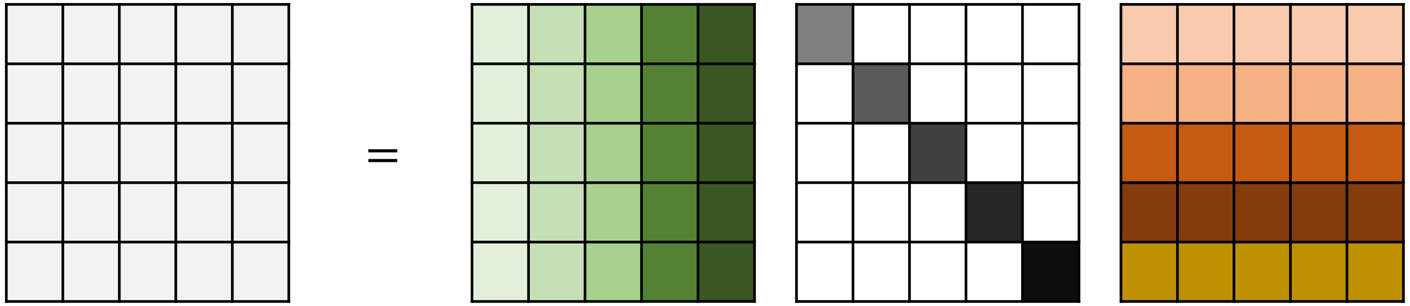
$$\mathbf{F} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{T}} = \sum_i \sigma_{ii}\, \boldsymbol{u}_i \boldsymbol{v}_i^T$$

where $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_{ii})$

Singular value decomposition

What if only σ_{11} is non-zero?

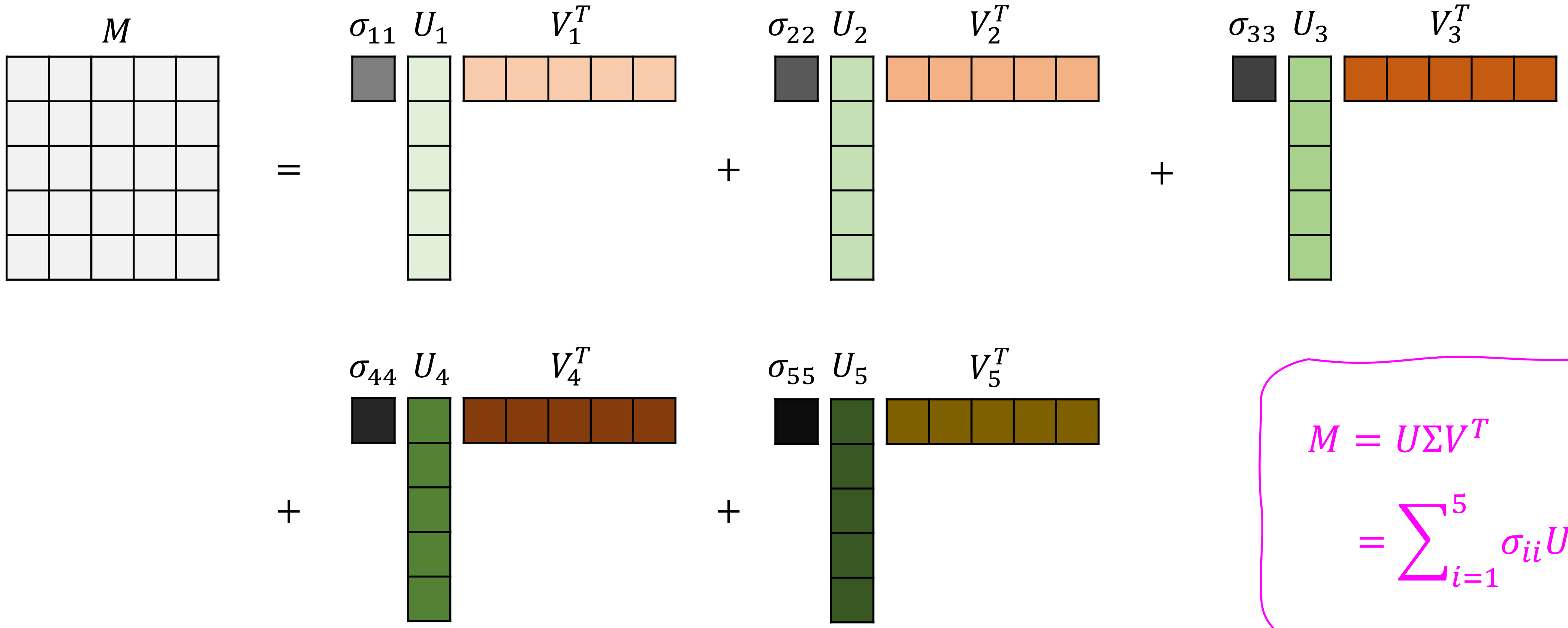
$M \quad = \quad U \quad \Sigma \quad \mathrm{V}^{\mathrm{T}}$



$M \in R^{5\times5}$ $U \in R^{5\times5}$ $\Sigma \in R^{5\times5}$ $V^T \in R^{5\times5}$

Singular value decomposition

What if only σ_{11} is non-zero?

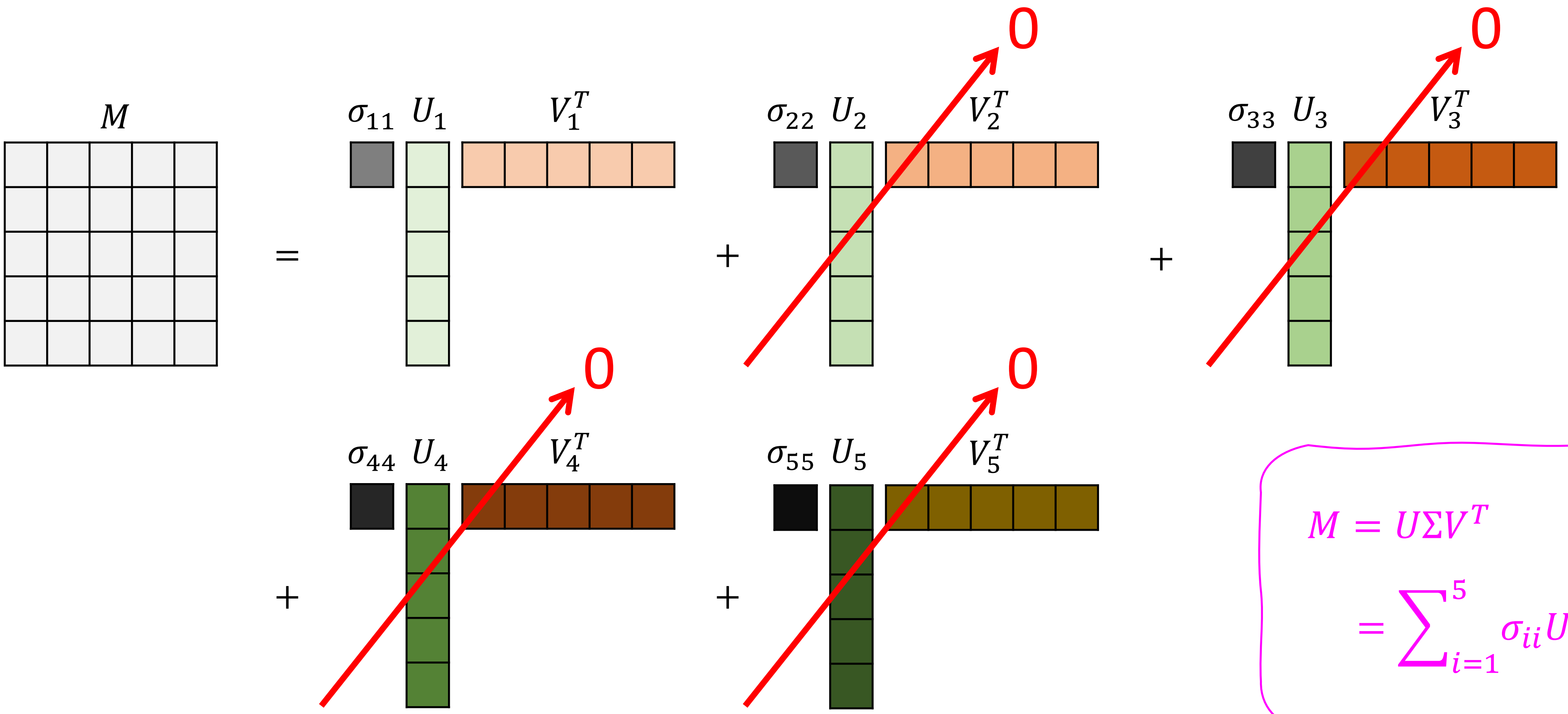


$$M = U\Sigma V^T$$

$$= \sum_{i=1}^{5} \sigma_{ii} U_i V_i^T$$

Singular value decomposition

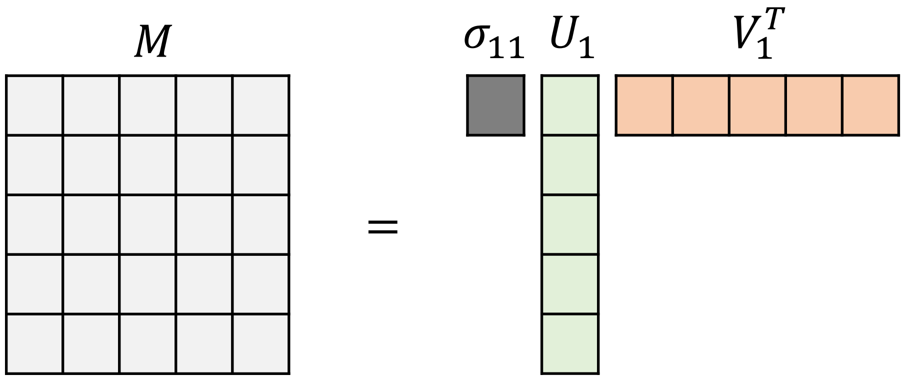
What if only σ_{11} is non-zero?



$$M = U\Sigma V^T$$
$$= \sum_{i=1}^{5} \sigma_{ii} U_i V_i^T$$

Singular value decomposition

What if only σ_{11} is non-zero?



Outer-product of $\left(\sqrt{\sigma_{11}}\right)U_1$ and $\left(\sqrt{\sigma_{11}}\right)V_1^T$

$$M = U\Sigma V^T$$
$$= \sigma_{11} U_1 V_1^T$$

How to find if a 2D filter is separable?

- Use Singular Value Decomposition (SVD)
  - If only one singular value is non-zero then the 2D filter is separable

- [Step 1] Compute SVD and check if only one singular value is non-zero

$$\mathbf{F} = \mathbf{U\Sigma V}^{\mathrm{T}} = \sum_i \sigma_i \, \boldsymbol{u}_i \boldsymbol{v}_i^T$$

where $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_i)$

- [Step 2] Vertical and horizontal filters are: $\sqrt{\sigma_1}\boldsymbol{u}_1$ and $\sqrt{\sigma_1}\boldsymbol{v}_1^T$

How to deal with missing (boundary values)

	?	?	?	
1	2	1	4	5
1	3	90	4	5
30	1	1	3	1
1	2	3	1	4
1	3	2	60	1

	0	0	0	
1	2	1	4	5
1	3	90	4	5
30	1	1	3	1
1	2	3	1	4
1	3	2	60	1

Set missing value to a particular value, say 0

How to deal with missing (boundary values)

	?	?	?	
1	2	1	4	5
1	3	90	4	5
30	1	1	3	1
1	2	3	1	4
1	3	2	60	1

	2	1	4	
1	2	1	4	5
1	3	90	4	5
30	1	1	3	1
1	2	3	1	4
1	3	2	60	1

Repeat boundary entries

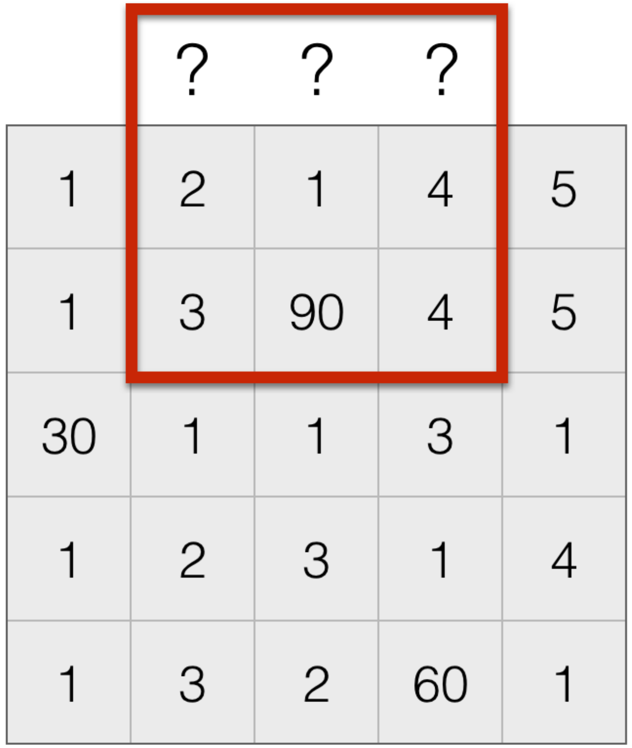
How to deal with missing (boundary values)

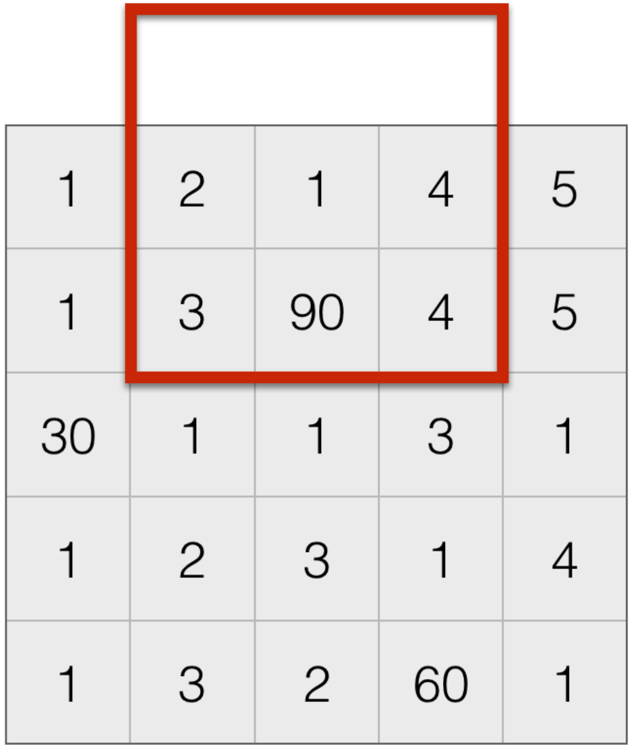
	?	?	?	
1	2	1	4	5
1	3	90	4	5
30	1	1	3	1
1	2	3	1	4
1	3	2	60	1

	3	2	60	
1	2	1	4	5
1	3	90	4	5
30	1	1	3	1
1	2	3	1	4
1	3	2	60	1

Wrap around. Useful to create an infinite domain.

How to deal with missing (boundary values)





Do nothing. Not a good choice, since the output size isn't the same as the input image, creating a host of engineering problems

Linear Filtering Properties

- Linearity

$$\texttt{filter}(\alpha_1 f_1 + \alpha_2 f_2) = \alpha_1 \texttt{filter}(f_1) + \alpha_2 \texttt{filter}(f_2)$$

- Shift-invariance

$$\texttt{filter}(\texttt{shift}(f)) = \texttt{shift}(\texttt{filter}(f))$$

- Any linear, shift-invariant filter can be represented as a convolution.

Properties of convolution

- Commulative: $a * b = b * a$
- Associative: $a * (b * c) = (a * b) * c$
- Distributes over addition: $a * (b + c) = a * b + a * c$
- Scalars factors out: $ka * b = a * kb = k(a * b)$
- Identity: $a * e = a$, where e is unit impulse

Linear filtering

- Remove, isolate, modify frequencies in the image
- Foundation based upon the convolution theorem

Recap (Linear Filtering)

- Check out Linear Filtering notes here
- Cross-correlation and convolution
  - 1D and 2D
- Gaussian blurring
- Separable filters
- Dealing with missing values
- Linearity and shift-invariance
- Properties of convolution

The story continues

- Digital cameras
    - Imaging pipeline
- Image formation
    - Pinholes, lenses, aperture, etc.
- Pixel-wise operations
    - Histogram equalization
- Spatial Processing (Linear filtering)
    - Cross-correlation and convolution
    - Separable filters
    - Dealing with missing values
    - Linearity and shift-invariance
    - Properties of convolution

The story continues

- Digital cameras
  - Imaging pipeline
- Image formation
  - Pinholes, lenses, aperture, etc.
- Pixel-wise operations
  - Histogram equalization
- Spatial Processing (Linear filtering)
  - Cross-correlation and convolution
  - Separable filters
  - Dealing with missing values
  - Linearity and shift-invariance
  - Properties of convolution

} Image enhancement

The story continues

- Digital cameras
  - Imaging pipeline
- Image formation
  - Pinholes, lenses, aperture, etc.
- Pixel-wise operations
  - Histogram equalization
- Spatial Processing (Linear filtering)
  - Cross-correlation and convolution
  - Separable filters
  - Dealing with missing values
  - Linearity and shift-invariance
  - Properties of convolution

Image enhancement

What about non-linear filtering?

Local Image Patches

- We have considered pixels completely independently of each other, except in the case of linear filtering
- In reality, photos have a lot of structure

Local Image Patches

- We have considered pixels completely independently of each other, except in the case of linear filtering
- In reality, photos have a lot of structure

Can be analyzed locally (e.g., small groups of neighbouring pixels) or globally (e.g., the entire image)

Local Image Patches

- There are many different types of patches in an image
  - Edges
  - Corners
  - Texture
  - Common surfaces
  - Perceptually significant

Local Image Patches

- There are many different types of patches in an image
  - Edges
  - Corners
  - Texture
  - Common surfaces
  - Perceptually significant

What makes a good patch?
When a group of neighbouring pixels is considered a patch?

Local Image Patches

- There are many different types of patches in an image
  - Edges
  - Corners
  - Texture
  - Common surfaces
  - Perceptually significant

What makes a good patch?
When a group of neighbouring pixels is considered a patch?

There isn't a good answer to these questions. The notion of a patch is *relative* and even a single pixel can be considered a 1x1 patch.

Local Image Patches

- There are many different types of patches in an image
  - Edges
  - Corners
  - Texture
  - Common surfaces
  - Perceptually significant

We will develop a mathematical description of patches, starting with small 3x3 patches and making our way to the entire image

What makes a good patch?
When a group of neighbouring pixels is considered a patch?

There isn't a good answer to these questions. The notion of a patch is *relative* and even a single pixel can be considered a 1x1 patch.

Local Image Patches: Why Do we Care?

- Recognition
- Inspection
- Video-based Tracking
- Special effects

Summary

- Spatial processing
  - Linear filtering
    - Check out Linear Filtering notes here
    - Cross-correlation and convolution
      - 1D and 2D
    - Gaussian blurring
    - Separable filters
    - Dealing with missing values
    - Linearity and shift-invariance
    - Properties of convolution
- Image patches