Spatial Processes – Part 3

Computational Photography (CSCI 3240U)

Faisal Z. Qureshi

http://vclab.science.ontariotechu.ca

OntarioTech 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

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 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

Linear Filtering in 1D

Signal

1	3	5	0	1	1

$w=1$

Kernel/Filter

+1	0	-1

not symmetric

Cross-correlation

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

1	3	5	0	1	1

Convolution

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

1	3	5	0	1	1

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

$$
CC\left(\underbrace{\begin{bmatrix} 1 & 3 & 5 & 0 & 1 & 1 \\ 2 & 1 & -1 & 3 & 0 & 4 \\ 0 & 1 & 2 & 2 & 3 & 4 \\ 1 & 3 & 4 & 9 & 10 & 3 \\ 3 & 7 & 8 & 2 & 1 & 4 \\ 1 & 5 & 6 & 4 & 2 & 1 \end{bmatrix}}_{\text{Image}} , \underbrace{\begin{bmatrix} \; & \; & \; \\ \; & \; & \; \\ \; & \; & \; \end{bmatrix}}_{\text{Kernel}} \right) = \begin{bmatrix} \; & \; & \; & \; \\ \; & \; & \; & \; \\ \; & \; & \; & \; \\ \; & \; & \; & \; \end{bmatrix}
$$

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

) =

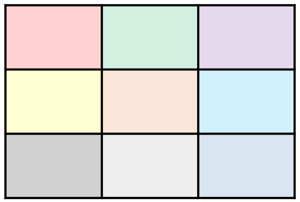
Linear Filtering in 2D

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

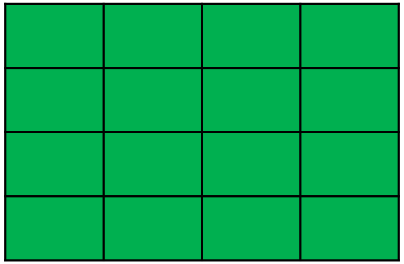
Image

,



Kernel

=



Linear Filtering in 2D

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

=

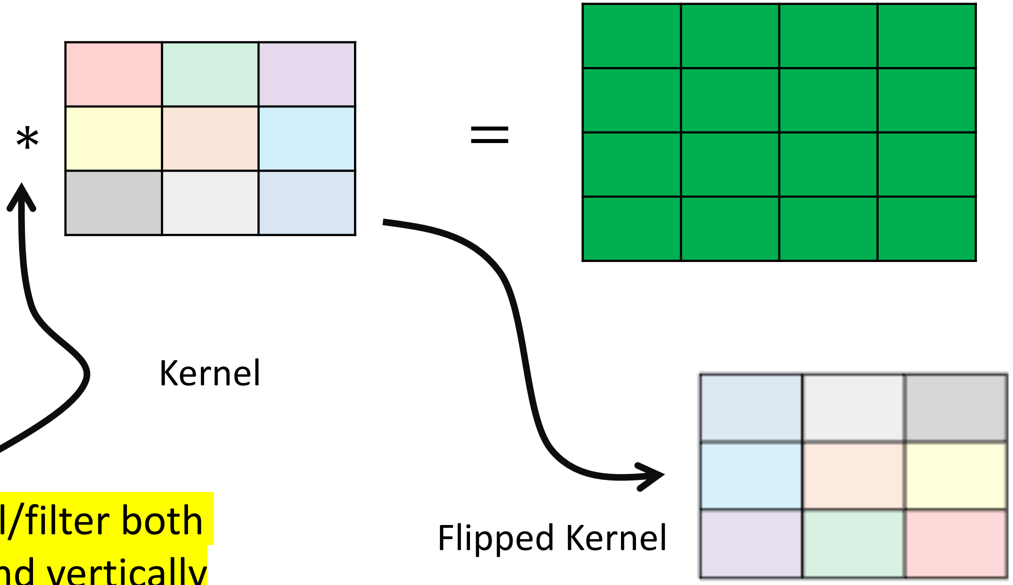
?			

Linear Filtering in 2D

ERROR IN SPATIAL-PROCESSING-2

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

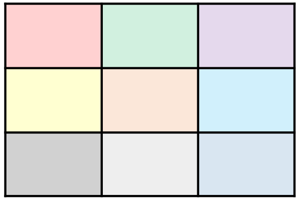
Image



Flip the kernel/filter both horizontally and vertically

Linear Filtering in 2D

Kernel/filter



Cross-correlation

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

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

Convolution

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

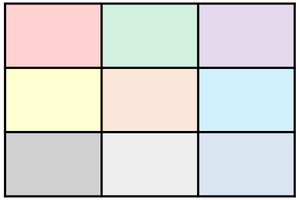
1	3	5	0	1	1
2	1	-1	3	0	4
0	1	2	2	3	4
1	3	4	9	10	3
3	7	8	2	1	4
1	5	6	4	2	1

Linear Filtering in 2D:
Number of multiplications and additions

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

*



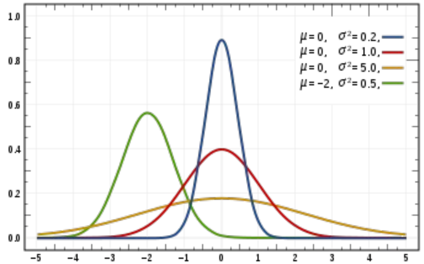
Kernel

#locations = (4)(4)

At each location:
#multiplications = (3)(3)
#additions = (3)(3)-1

Total = 16 x (9 MUL + 8 ADD)

Gaussian in 1D



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

0

Here μ and σ refer to the mean and standard deviation of this Gaussian.

$e^{-\frac{(-2)^2}{2\sigma^2}}$ | $e^{-\frac{(-1)^2}{2\sigma^2}}$ | e^0

Aside: Mean and Standard Deviation

Given n data points $\{x_1, x_2, \cdots, x_n\}$

$x = -2$

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

$$\sigma^2 = \mathrm{E}[(x-\mu)^2] = \frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2$$

Multivariate Guassian (in k-dimensions)

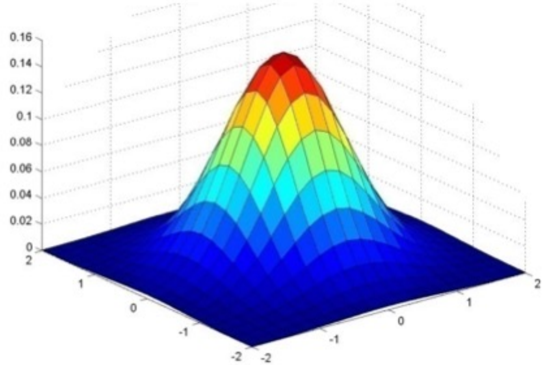
$$G(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma) = \frac{1}{\sqrt{(2\pi)^k \det(\Sigma)}} exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu})\right)$$

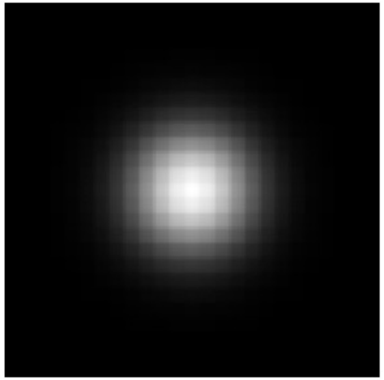
where

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

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

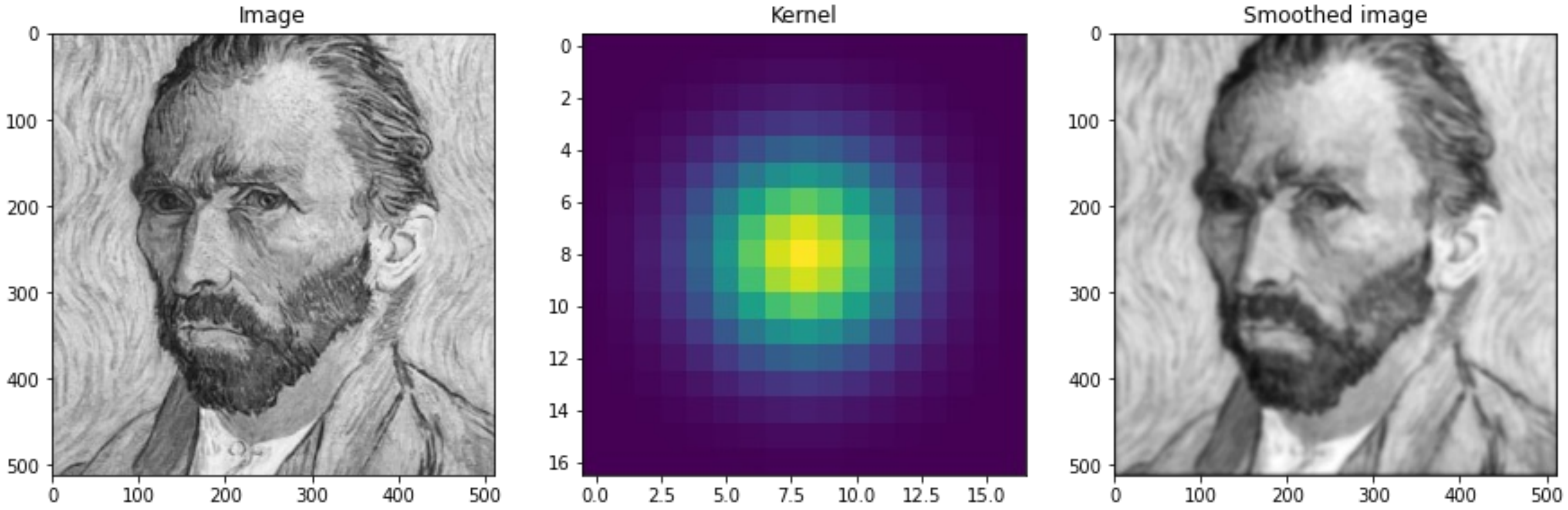
$$\Sigma \in R^{k \times k}$$





Gaussian in 2D

Gaussian Blurring

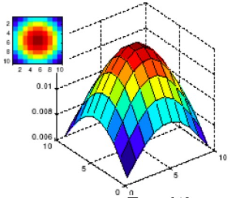


Gaussian Blurring

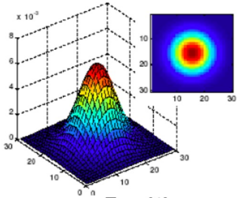
- We often use the following approximation of a Gaussian function

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

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



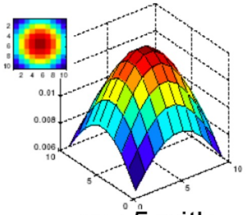
σ = 5 with 10 x 10 kernel



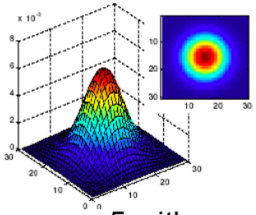
σ = 5 with 30 x 30 kernel

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

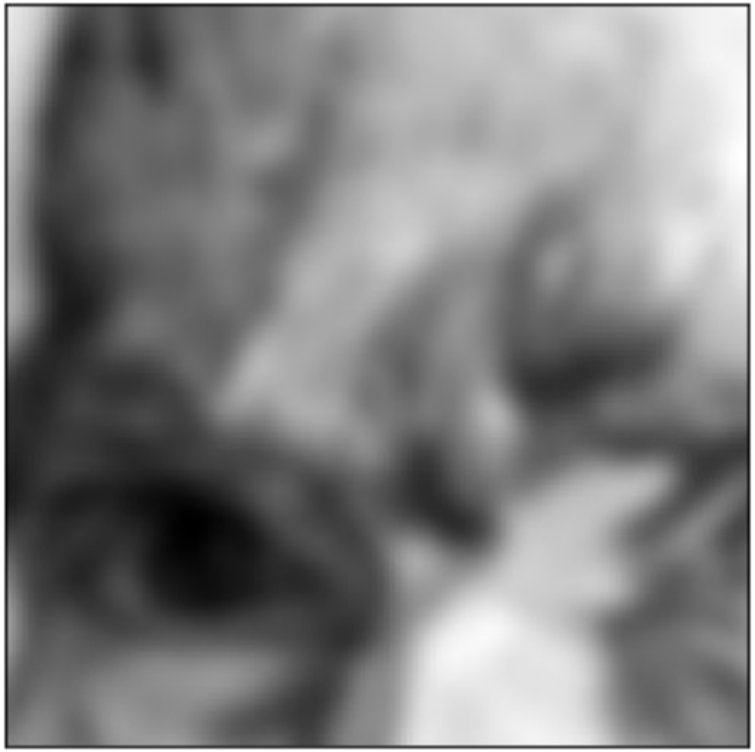
- Removes high-frequency components from the image
  - Blurs the image
  - Acts as a low-pass filter

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

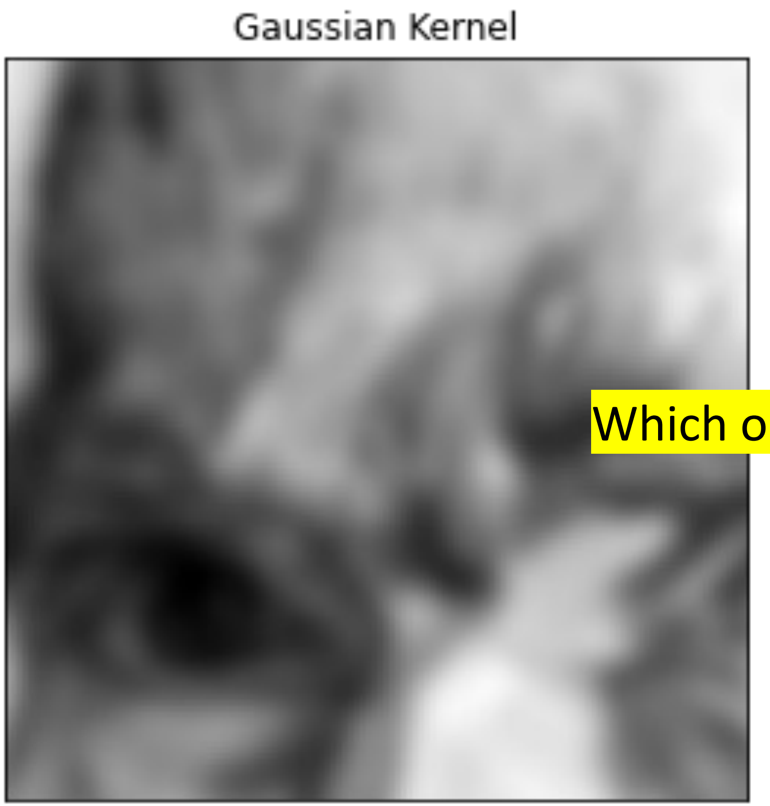
Gaussian Blurring vs. Average (Box) Filtering

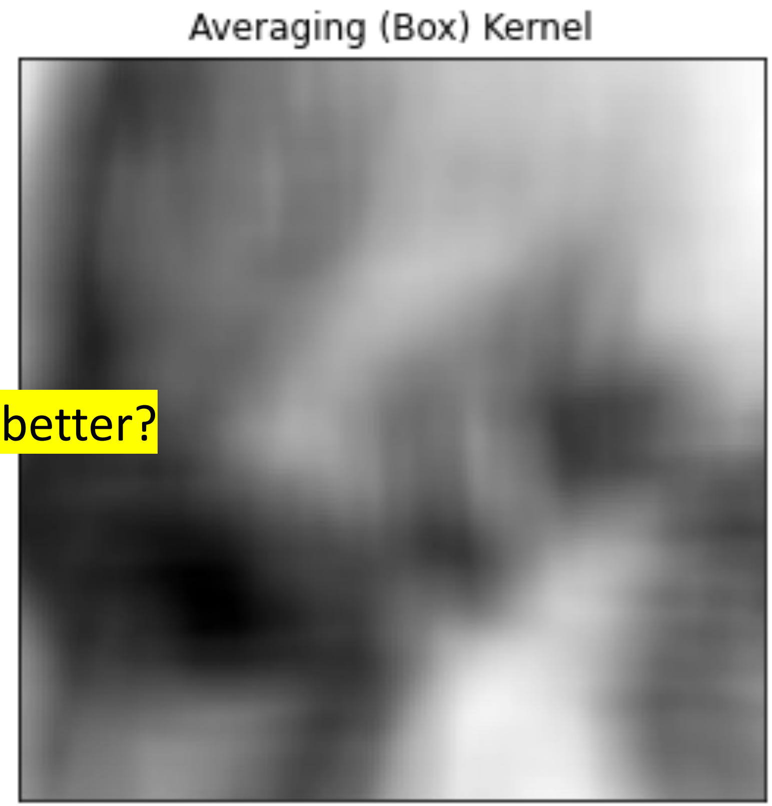
Gaussian Kernel



Averaging (Box) Kernel

Gaussian Blurring vs. Average (Box) Filtering



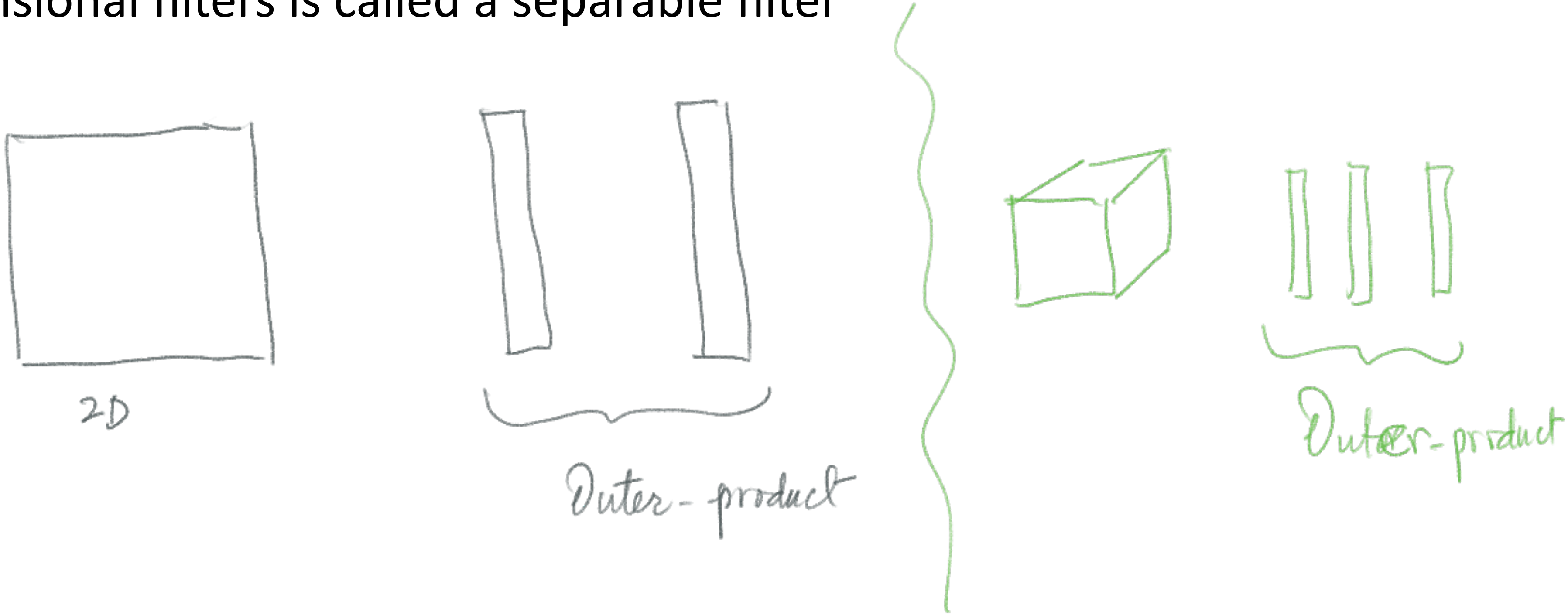


Which one is better?

MIDTERM 1

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

$a^T = [1,2]$ and $b^T = [1,0]$

Inner-product.

$$\underset{\in \mathbb{R}^{1\times 2}}{\begin{bmatrix} 1 & 2 \end{bmatrix}} \underset{\in \mathbb{R}^{2\times 1}}{\begin{bmatrix} 1 \\ 0 \end{bmatrix}} = (1)(1) + (2)(0) = 1$$

$$a^T b = 1$$

DOT-PRODUCT

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

Outer-product

$$a b^T = \underset{\in \mathbb{R}^{2\times 1}}{\begin{bmatrix} 1 \\ 2 \end{bmatrix}} \underset{\in \mathbb{R}^{1\times 2}}{\begin{bmatrix} 1 & 0 \end{bmatrix}} = \underset{\in \mathbb{R}^{2\times 2}}{\begin{bmatrix} (1)(1) & (1)(0) \\ (2)(1) & (2)(0) \end{bmatrix}}$$

$$= \begin{bmatrix} 1 & 0 \\ 2 & 0 \end{bmatrix}$$

Outer-Product and Inner-Product

$a^T = [1,2,1]$ and $b^T = [1,1,3]$

$$a^T b = \underset{\in \mathbb{R}^{1\times 3}}{\begin{bmatrix} 1 & 2 & 1 \end{bmatrix}} \underset{\in \mathbb{R}^{3\times 1}}{\begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix}} = (1)(1) + (2)(1) + (1)(3) = 1+2+3 = 6$$

$$a b^T = \underset{\in \mathbb{R}^{3\times 1}}{\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}} \underset{\in \mathbb{R}^{1\times 3}}{\begin{bmatrix} 1 & 1 & 3 \end{bmatrix}} = \underset{\in \mathbb{R}^{3\times 3}}{\begin{bmatrix} (1)(1) & (1)(1) & (1)(3) \\ (2)(1) & (2)(1) & (2)(3) \\ (1)(1) & (1)(1) & (1)(3) \end{bmatrix}} = \begin{bmatrix} 1 & 1 & 3 \\ 2 & 2 & 6 \\ 1 & 1 & 3 \end{bmatrix}$$

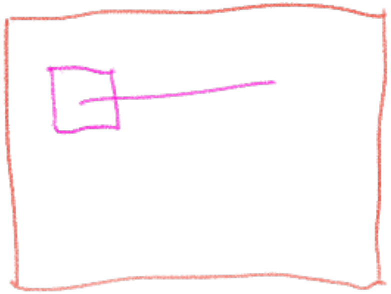
Separability

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

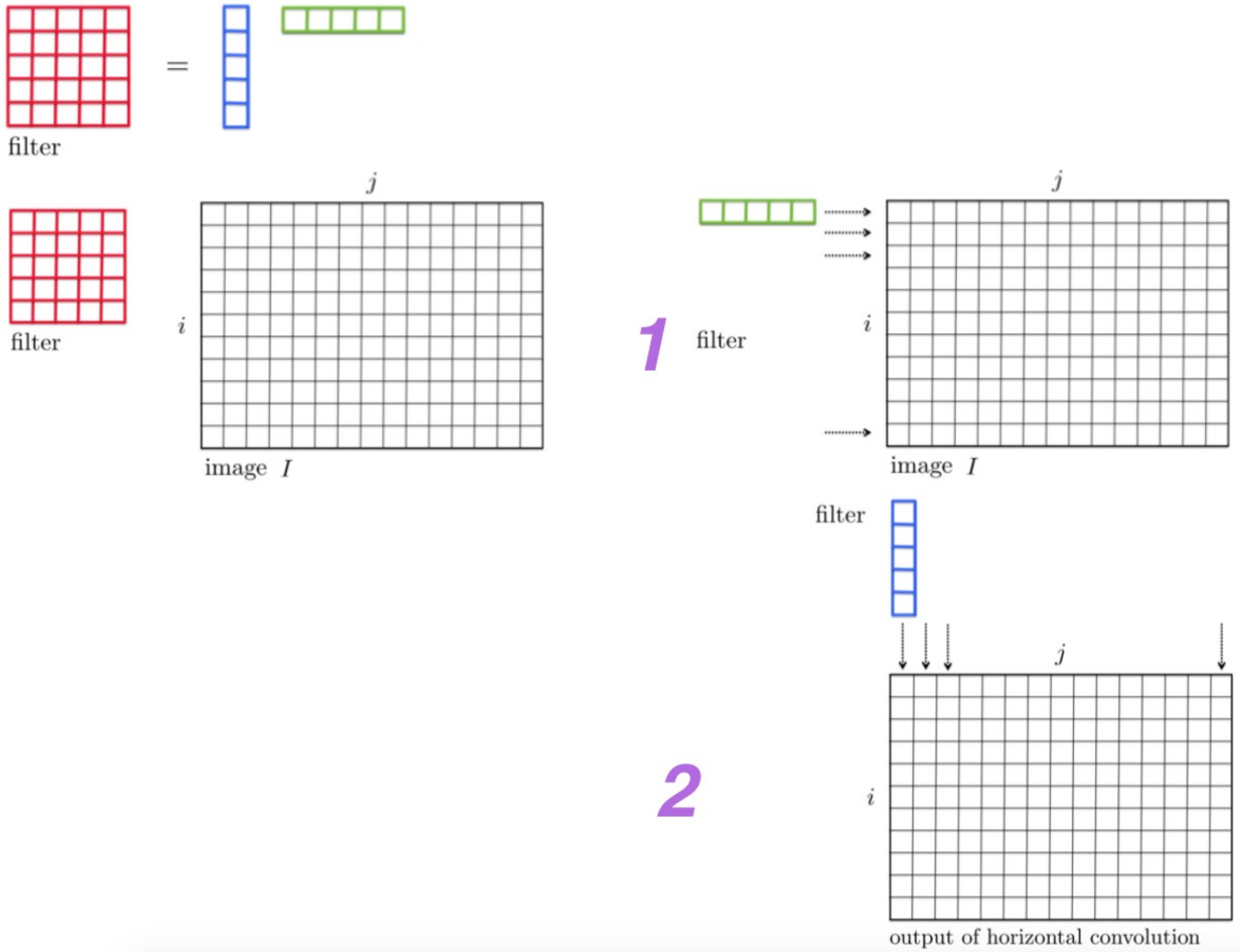
$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & \overset{1}{\cancel{3}} \end{bmatrix}$$

Convolution with Separable Filters in 2D

- [Step 1] Perform row-wise convolution with horizontal filter
- [Step 2] Perform column-wise convolution 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 $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$

$(1)(1)+(0)(2)+(-2)(1)+(2)(2)+(-1)(4)+(6)(2)+(3)(1)+(0)(2)+(1)(1)$

$= 1+0-2+\cancel{4}-\cancel{4}+12+3+0+1$

$= \cancel{1}-\cancel{2}+12+3+\cancel{1}$

$= 15$

① $(1)(1)+(2)(0)+(1)(-2) = 1-2 = -1$

② $\cancel{(1)(2)+(2)(-1)}+(6)(1) = 6$

③ $(1)(3)+(2)(0)+(1)(1) = 4$

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

$= -1+12+4$

$= \boxed{15}$

$\begin{bmatrix} \cdot & -1 & \cdot & \cdot \\ \cdot & 6 & \cdot & \cdot \\ \cdot & 4 & \cdot & \cdot \end{bmatrix}$

Computational Considerations

- For non-separable filters

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

60 60 1000 1000

- For separable filters

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

60 x 1000 x 1000 60 x 1000 x 1000

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

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

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 $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$

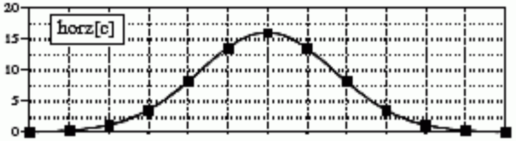
Where possible exploit separability to speed up convolutions ✓

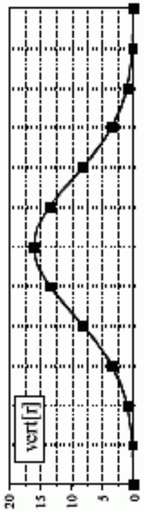
Gaussian filter is separable

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

Take home exercise
confirm that this matches
our defn. of multivariate Gaussian.

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.

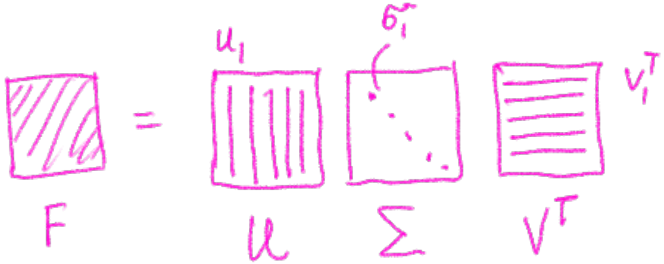
How to find if a 2D filter is separable?

- 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

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

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



$$= \underline{\underline{\sigma_1 u_1 v_1^T}} + \overset{=0}{\sigma_2} u_2 v_2^T + \cdots + \overset{=0}{\sigma_d} u_d v_d^T$$

≈ 0

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

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

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)

	?	?	?	
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

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

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

Summary

- Linear filtering
- Separable filters
- Dealing with missing values
- Linearity and shift-invariance
- Properties of convolution

Check out Linear Filtering notes here.