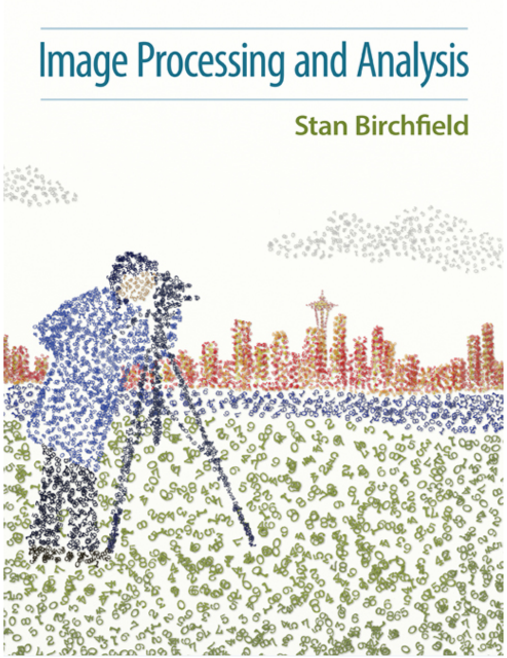
University
of Stavanger

Prof. Kjersti Engan

ELE510 Image processing and computer vision

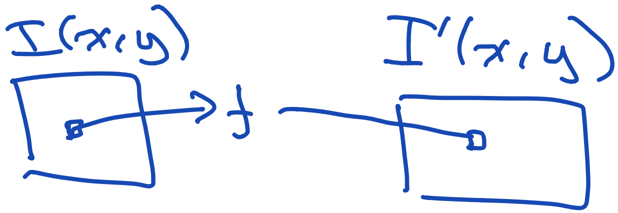
Spatial-Domain Filtering, (chap 5.1-5.4 Birchfield) 2020



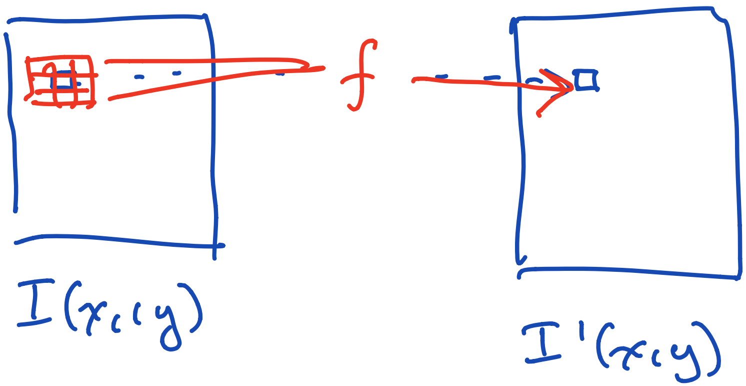
Parts in course presentations is material from Cengage learning. It can be used for teaching, and it can be share it with students on access controlled web-sites (Canvas) for use in THIS course. **But it should not be copied or distributed further in any way** (by you or anybody).

© 2018 Cengage Learning®. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Spatial Domain Filtering

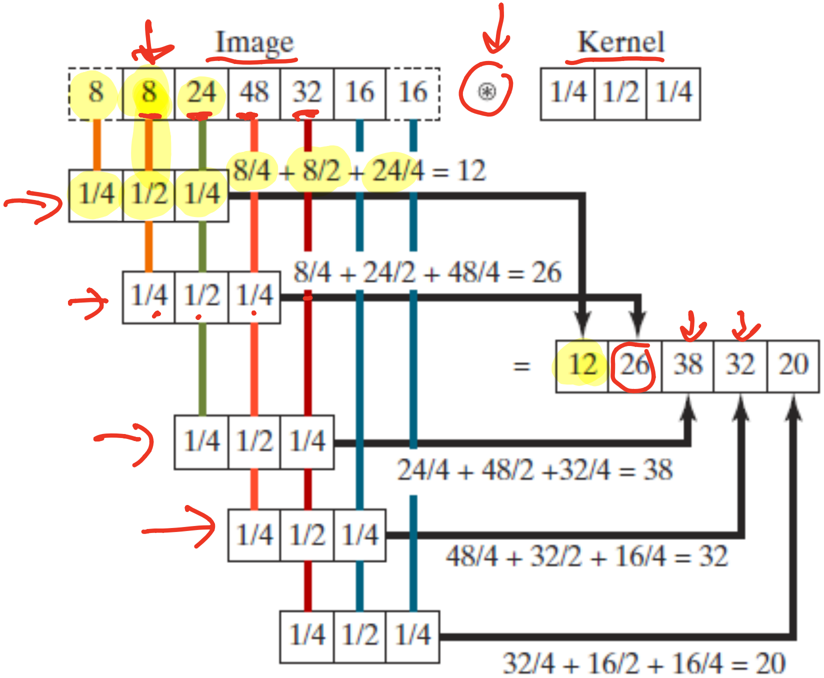


- Remember for point transformations, output value at a position (x,y) is dependent on the input value at (x,y) (Examples: thresholding, histogram operations) : $I'(x, y) = f(I(x, y))$
- For spatial domain filtering, output pixel at position (x,y) depends on input value at (x,y) AND neighbouring pixels. This can be defined by convolution.



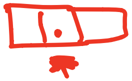
Example of 1-D convolution

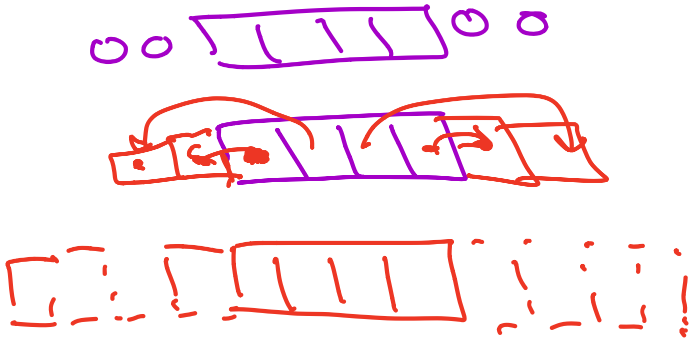
filter

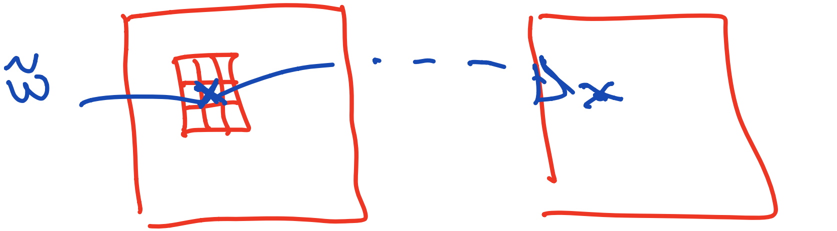


Border/edge strategy

- How to deal with the edges of the signal/image?
- Zero-padding
- Mirror
- Repeat
- truncate







(5.1) Convolution

- The discrete convolution of a 1D signal *f* with a kernel *g* is defined as:

$$f'(x) = f(x) \circledast g(x) \equiv \sum_{i=-\infty}^{\infty} f(x-i)g(i)$$
$$= \sum_{i=-\tilde{w}}^{w-\tilde{w}-1} f(x-i)g(i)$$

- *w* is the width of the kernel
- The **origin** $\boxed{\tilde{w}}$ of the kernel indicates the location where the result is stored, often the index nearest the center.
- Usually *w* is odd and $\tilde{w}$ is in the middle of the kernel.

Convolution or correlation?

Convolution is closely related to **cross-correlation**, which is defined as:

$$\tilde{w} \equiv \lfloor \tfrac{1}{2}(w - 1) \rfloor$$

$$f'_{corr}(x) = f(x) \check{\circledast} g(x) \equiv \sum_{i=-\infty}^{\infty} f^*(x+i)g(i) = \sum_{i=-\tilde{w}}^{w-\tilde{w}-1} f^*(x+i)g(i)$$

If f(x) is real, the only difference is that convolution flips the kernel.

If kernel is symmetric and f(x) is real -> No difference

(Some filtering functions might be implemented with correlation)

Kernels – some intuition

- Smoothing kernels: averaging, noise reduction, low pass filtering:

$$\boxed{\sum_i g(i) = 1}$$

usually symmetric

Differentiating kernels: Extract boundaries, edgess, high pass filters

$$\boxed{\sum_i g(i) = 0}$$

usually symmetric OR antisymmetric

Convolution as matrix multiplication

- Convolution can be written as a matrix – vector multiplication

$$\underbrace{\begin{bmatrix}12\\26\\38\\32\\20\end{bmatrix}}_{f'} = \underbrace{\frac{1}{4}\begin{bmatrix}1 & 2 & 1 & 0 & 0 & 0 & 0\\0 & 1 & 2 & 1 & 0 & 0 & 0\\0 & 0 & 1 & 2 & 1 & 0 & 0\\0 & 0 & 0 & 1 & 2 & 1 & 0\\0 & 0 & 0 & 0 & 1 & 2 & 1\end{bmatrix}}_{\substack{\textit{convolution matrix}\\ \mathbf{G}}} \underbrace{\begin{bmatrix}8\\8\\24\\48\\32\\16\\16\end{bmatrix}}_{f}$$

Image transformations, linear systems

$$f(x, y) \longrightarrow \boxed{\text{Operator, O}} \longrightarrow g(x, y)$$

Linear operator:

$$\boxed{O[af_1 + bf_2] = aO[f_1] + bO[f_2]}$$

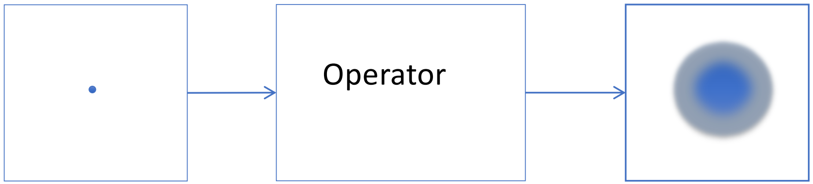
A system is called **shift-invariant** if a shift in the input causes a shift in the output by the same amount.

$$f'(x - x_0) = \mathcal{L}(f(x - x_0))$$

Linear shift-invariant systems: systems that are particularly important due to their convenient mathematical properties. Often called LTI system (linear time-invariant systems) .

If a system is not linear, then it is said to be **nonlinear**.

Point spread function



Point source, $\delta(\alpha\text{-}x,\beta\text{-}y)$

Point spread function (PSF) of the operator.

The PSF *defines* a linear operator

Equivalent to impulse response in signal processing

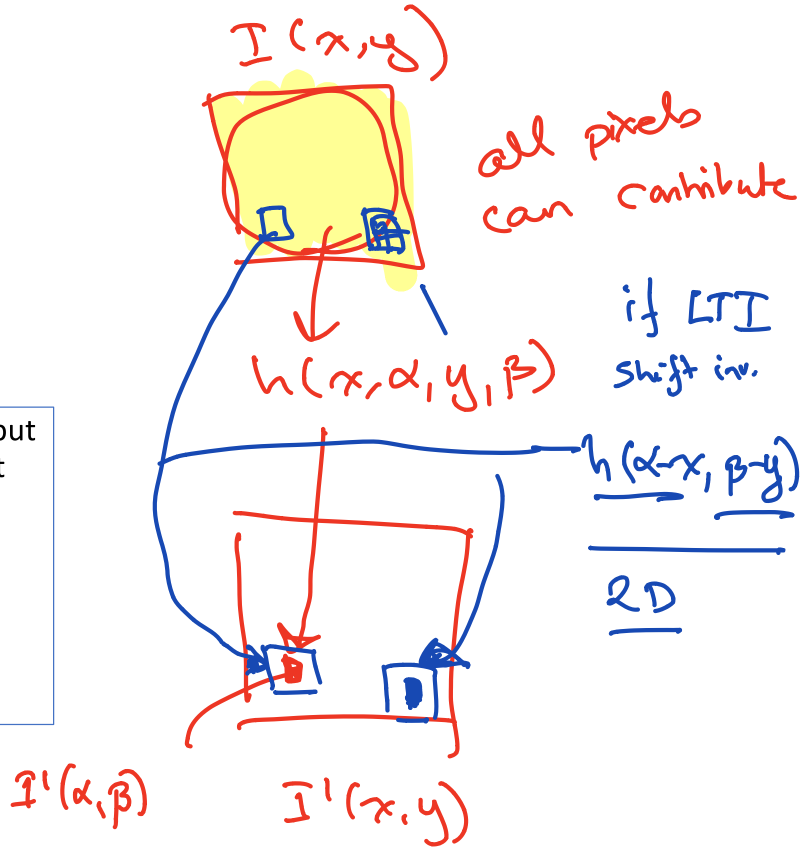
Point spread function

$$f(x,y) \longrightarrow \boxed{\overset{4D}{h(x,\alpha,y,\beta)}} \longrightarrow g(\alpha,\beta)$$

4

The point spread function express how much the input value at position (x,y) influences the output value at position (α,β)

$$g(\alpha,\beta) = \sum_{x=1}^{N}\sum_{y=1}^{M} f(x,y)h(x,\alpha,y,\beta)$$



Linear systems/filters

- Linear systems / filters are completely defined by the impuls response / point spread function.

$$\delta(x) = \begin{cases} 1 & x=0 \\ 0 & \text{otherwise} \end{cases}$$

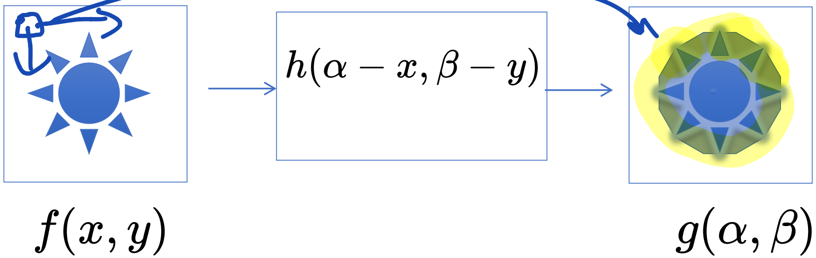
$$\delta(x) \longrightarrow \boxed{} \longrightarrow h(x)$$

$$h(x) = \delta(x) \circledast g(x) = \sum_{i=-\infty}^{\infty} \delta(x-i)\, g(x) = g(x)$$

- FIR: Finite Impulse Response. A FIR filter is a linear system or filter where h(x) is finite in duration. Can be implemented with a kernel and convolution.
- IIR: Infinite Impulse Response. Typically implemented in frequency domain or as a reqursive system.

Point spread function (PSF) and convolution

Shift invariant point spread function : LTI system

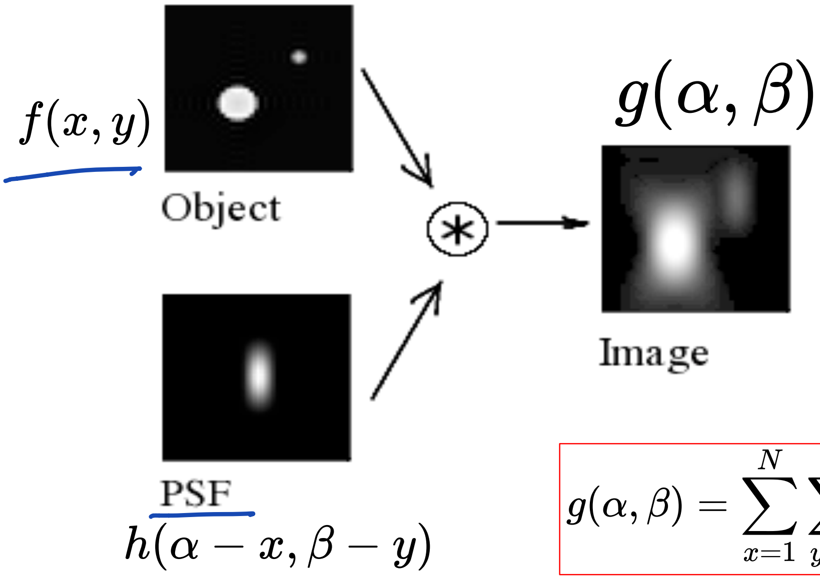


To use the convolution formula:
System/filter needs to be linear
AND shift(time)-invariant.

$$g(\alpha, \beta) = \sum_{x=1}^{N} \sum_{y=1}^{N} f(x, y) h(\alpha - x, \beta - y)$$

Convolution is commutative:

$$g(\alpha, \beta) = \sum_{x=1}^{N} \sum_{y=1}^{N} h(x, y) f(\alpha - x, \beta - y)$$



$$\boxed{g(\alpha,\beta)=\sum_{x=1}^{N}\sum_{y=1}^{N} f(x,y)h(\alpha-x,\beta-y)}$$

Convolution as Fourier multiplication

- Convolution in the spatial domain is equivalent to multiplication in the frequency domain.

$$\boxed{f'(x)} = f(x) \circledast g(x) = \underbrace{\sum_{i=-\infty}^{\infty} f(x-i)g(i)}$$

$$\boxed{f'(x)} = f(x) \circledast g(x) = \mathcal{F}^{-1}\{\mathcal{F}\{f(x)\} \cdot \mathcal{F}\{g(x)\}\}$$

Linear filters

$$x(n) \longrightarrow \boxed{h(n)} \longrightarrow y(n)$$

$$y(n) = \sum_k h(k)x(n-k)$$

$$F(y(n)) = Y(w) = H(w)X(w)$$

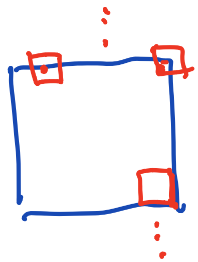
- Linear filters : convolution in time (space) domain is mulitplicatioin in freq.domain. (1D and 2D)

2D convolution

We are interested in real filters i.e. Point spread function h(x,y) should be real, not complex.

Shift invariant filters often represented with *a mask / kernel* h(x,y) with finite support.

$$g(\alpha,\beta) = \sum_{x=1}^{N}\sum_{y=1}^{N} h(x,y) f(\alpha - x, \beta - y)$$



$$g(\alpha,\beta) = \sum_{x=-\frac{L-1}{2}}^{\frac{L-1}{2}} \sum_{y=-\frac{L-1}{2}}^{\frac{L-1}{2}} h(x,y) f(\alpha - x, \beta - y)$$

finite support

$h:\ L \times L \quad f: N \times N$

$g:\ (N+L-1) \times (N+L-1)$ $\quad g(\alpha,\beta)$

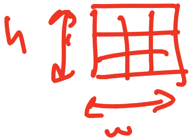
Often the output image g is truncated to the same size as input image f. ”Boundary problem”

2D convolution

2D Convolution: used to perform filtering on a 2D image, books notation.

$$I'(x,y) = I(x,y) \circledast G(x,y) = \sum_{i=0}^{w-1}\sum_{j=0}^{h-1} I(x + \tilde{w} - i, y + \tilde{h} - j)G(i,j)$$

where *w* and *h* are the width and height of the kernel.



$\tilde{h} = \lfloor \frac{1}{2}(h-1) \rfloor$ $\tilde{w} = \lfloor \frac{1}{2}(w-1) \rfloor$

In image processing we usually use FIR filters with relatively small filter kernels (can use convolution implementation, can use LTI system theory)

OR nonlinear filters.