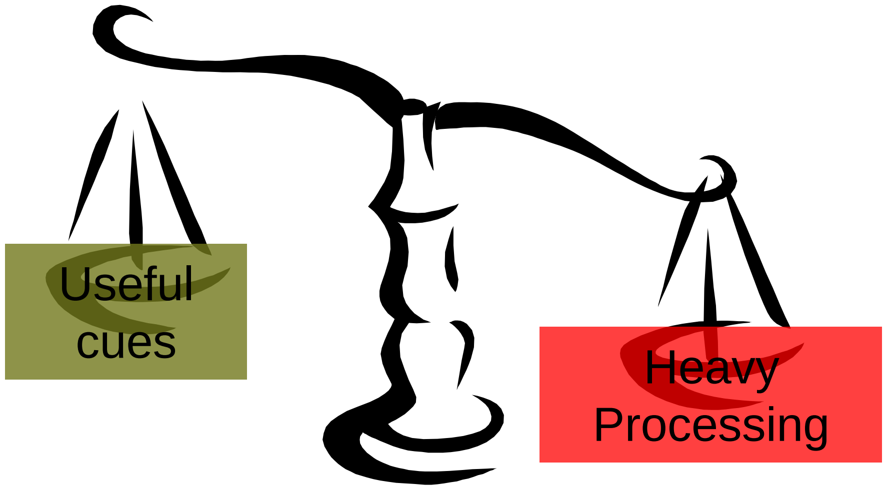
Perception | **Correlation & Convolution**
Autonomous Mobile Robots

Margarita Chli – University of Edinburgh

Paul Furgale, Marco Hutter, Martin Rufli, Davide Scaramuzza, Roland Siegwart

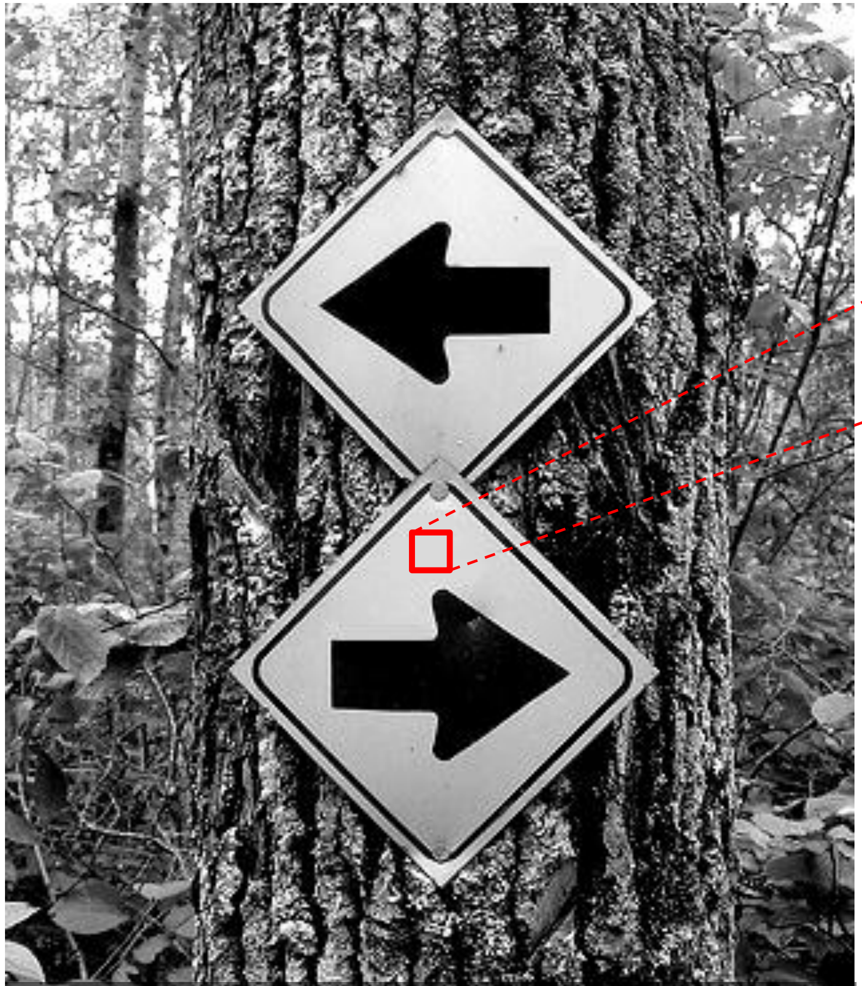
Image Saliency | data reduction

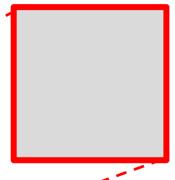
- Monochrome image ⇨ matrix of intensity values
- Typical sizes:
  - 320 x 240 (QVGA)
  - 640 x 480 (VGA)
  - 1280 x 720 (HD)
- Intensities sampled to 256 grey levels ⇨ 8 bits
- Images capture a lot of information



⇨ Reduce the amount of input data:
preserving useful info & discarding redundant info

Image Saliency | what is USEFUL, what is REDUNDANT ?





218	218	218	218	218	218	218	218	218	218	218
218	218	218	218	218	218	218	218	218	218	218
218	218	218	218	218	218	218	218	218	218	218
218	218	218	218	218	218	218	218	218	218	218
218	218	218	218	218	218	218	218	218	218	218
218	218	218	218	218	218	218	218	218	218	218
218	218	218	218	218	218	218	218	218	218	218
218	218	218	218	218	218	218	218	218	218	218
218	218	218	218	218	218	218	218	218	218	218
218	218	218	218	218	218	218	218	218	218	218
218	218	218	218	218	218	218	218	218	218	218

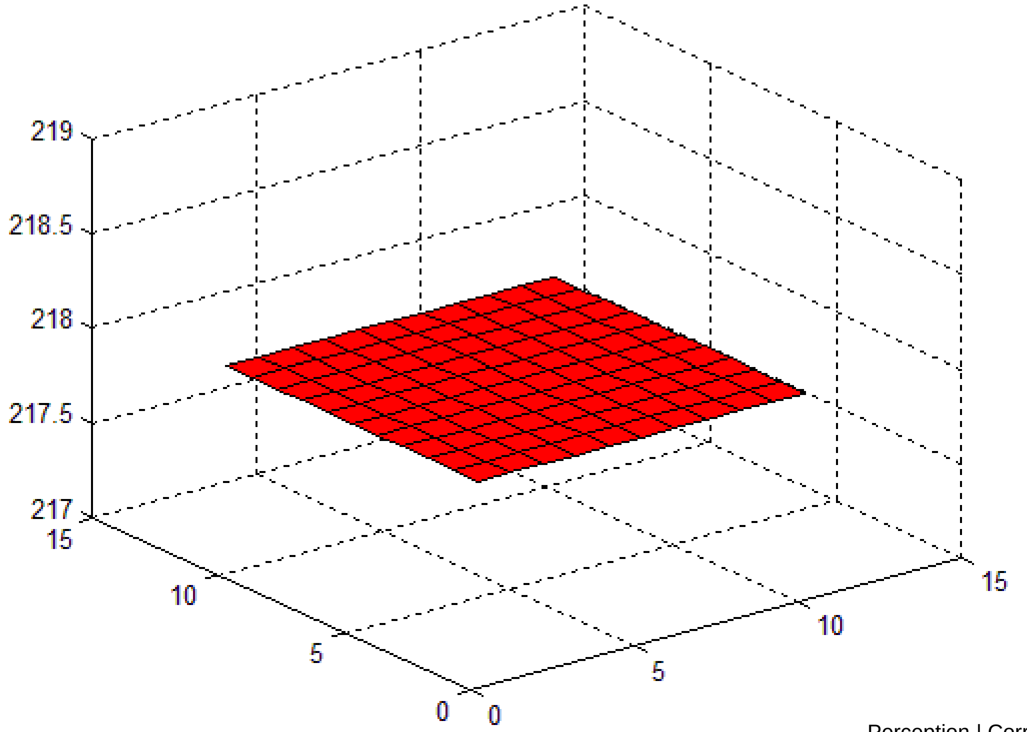


Image Saliency | what is USEFUL, what is REDUNDANT ?

‘Slightly confusing signs’ by Dan McKay, http://www.flickr.com/photos/mukluk/241256203/.
Licence at http://creativecommons.org/licenses/by/2.0

208	208	208	208	208	208	208	208	208	208	208
208	208	208	208	208	208	208	208	208	208	208
208	208	208	208	208	208	208	208	208	208	208
208	208	208	208	208	208	208	208	208	208	208
208	208	208	208	208	208	208	208	208	208	208
208	207	208	208	208	208	208	208	208	208	208
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

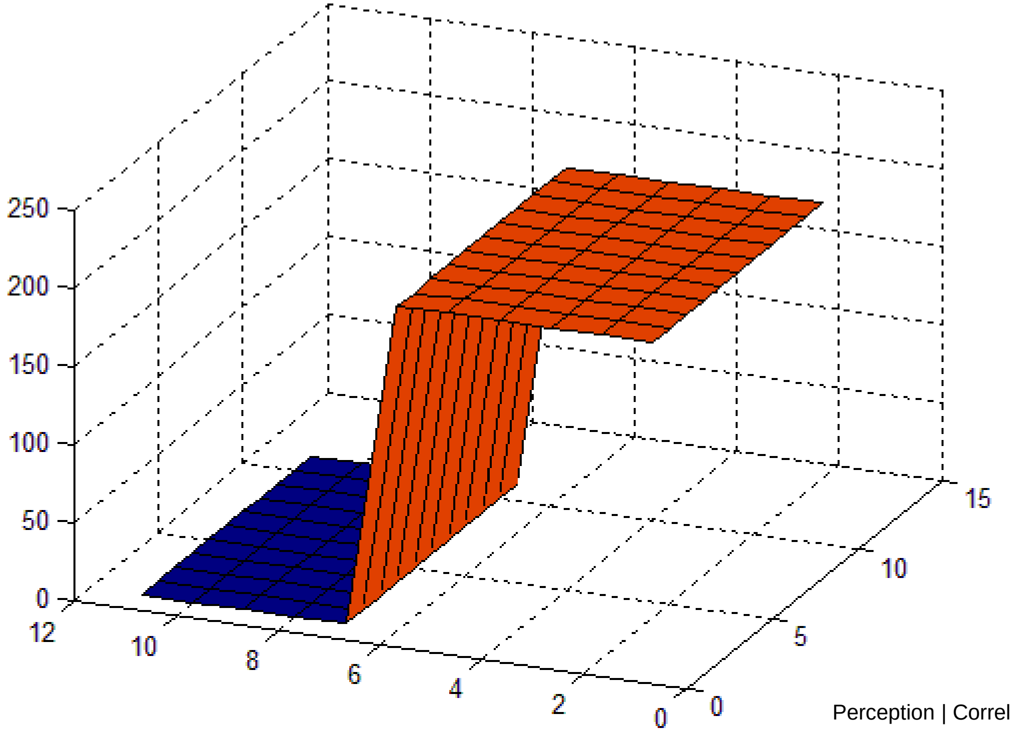
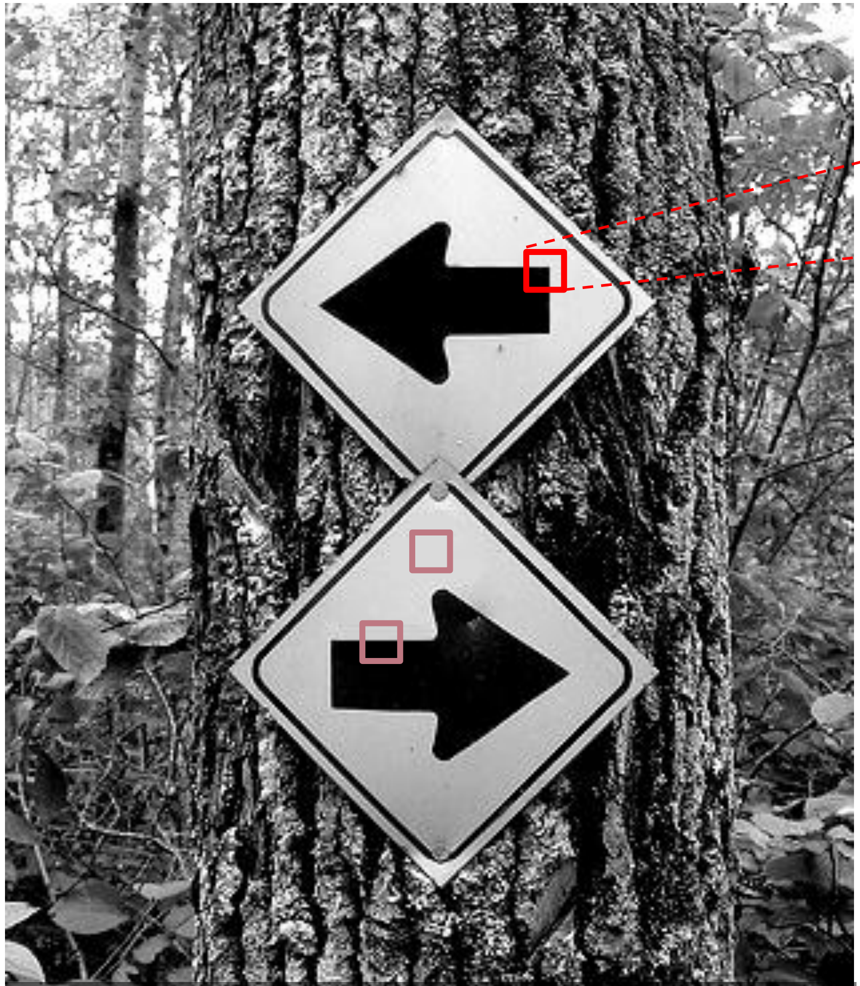


Image Saliency | what is USEFUL, what is REDUNDANT ?



‘Slightly confusing signs’ by Dan McKay, http://www.flickr.com/photos/mukluk/241256203/.
Licence at http://creativecommons.org/licenses/by/2.0

229	229	229	229	229	229	229	229	229	229	229
229	229	229	229	229	229	229	229	229	229	229
229	229	229	229	229	229	229	229	229	229	229
229	229	229	229	229	229	229	229	229	229	229
229	229	229	229	229	230	229	229	229	229	229
5	17	31	7	1	0	229	229	229	229	229
0	0	1	0	0	0	229	229	229	229	229
0	0	0	0	0	0	229	229	229	229	229
0	0	0	0	1	4	229	229	229	229	229
0	0	0	0	0	11	229	229	229	229	229
0	0	0	0	0	5	229	229	229	229	229

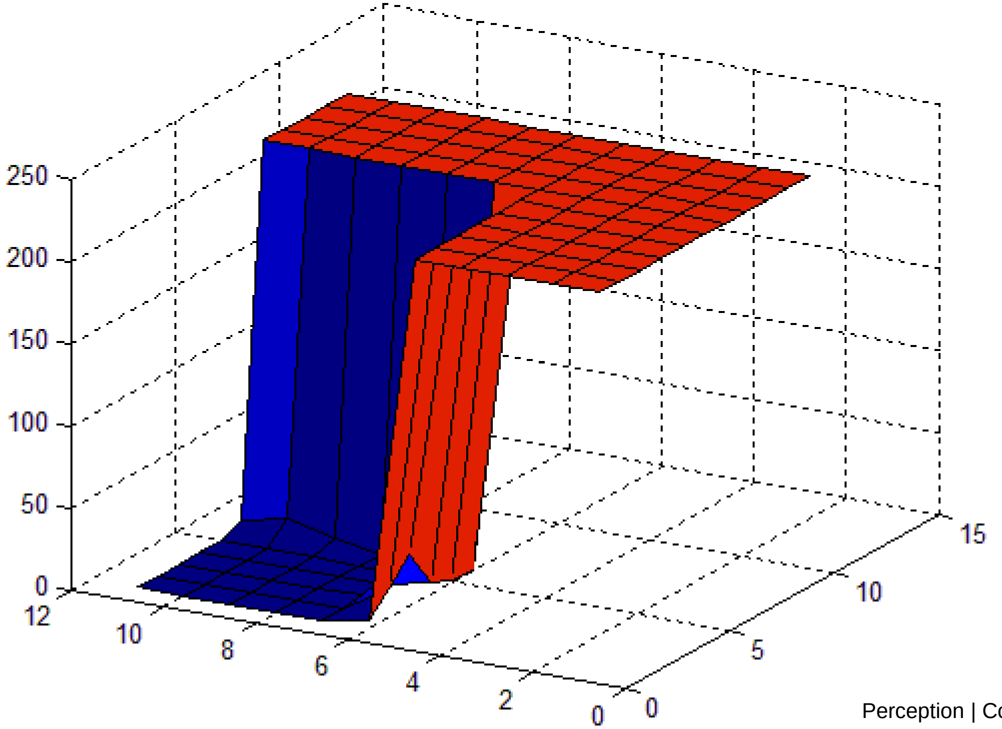


Image filtering | spatial filters

- "filtering": accept / reject certain components
- S_{xy} : neighborhood of pixels around the point (x,y) in an image I
- Spatial filtering operates on S_{xy} to generate a new value for the corresponding pixel at output image J

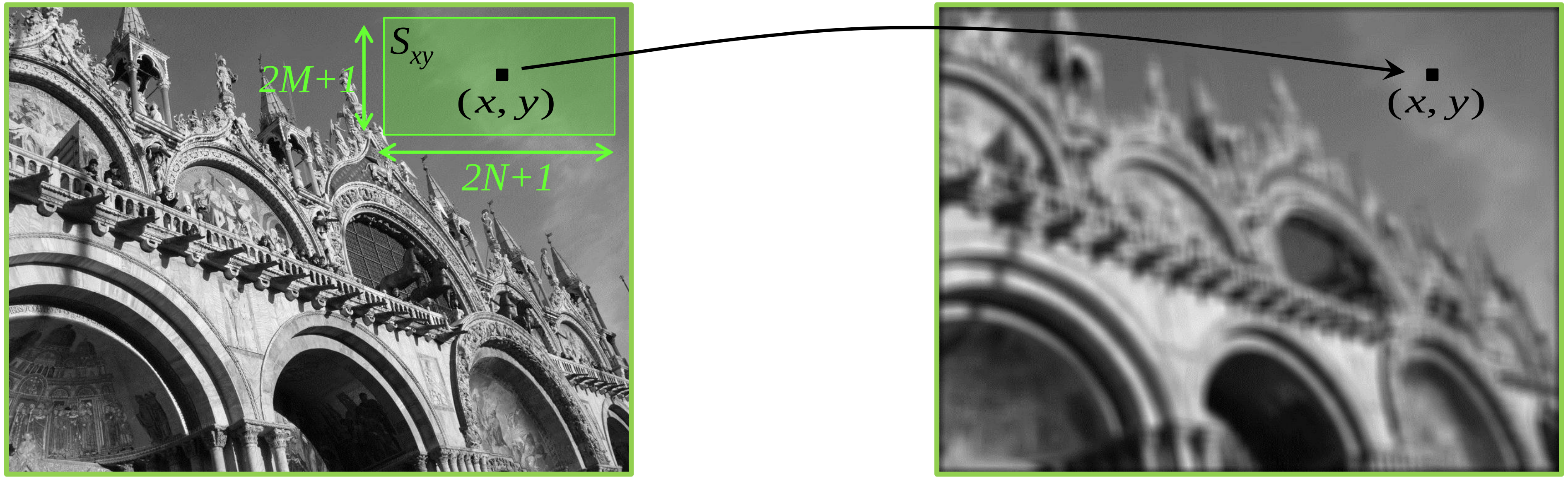


Image I

Filtered Image $J = F(I)$

- For example, an averaging filter is: $J(x,y) = \dfrac{\sum_{(r,c)\in S_{xy}} I(r,c)}{(2M+1)(2N+1)}$

Image filtering | linear, shift-invariant filters

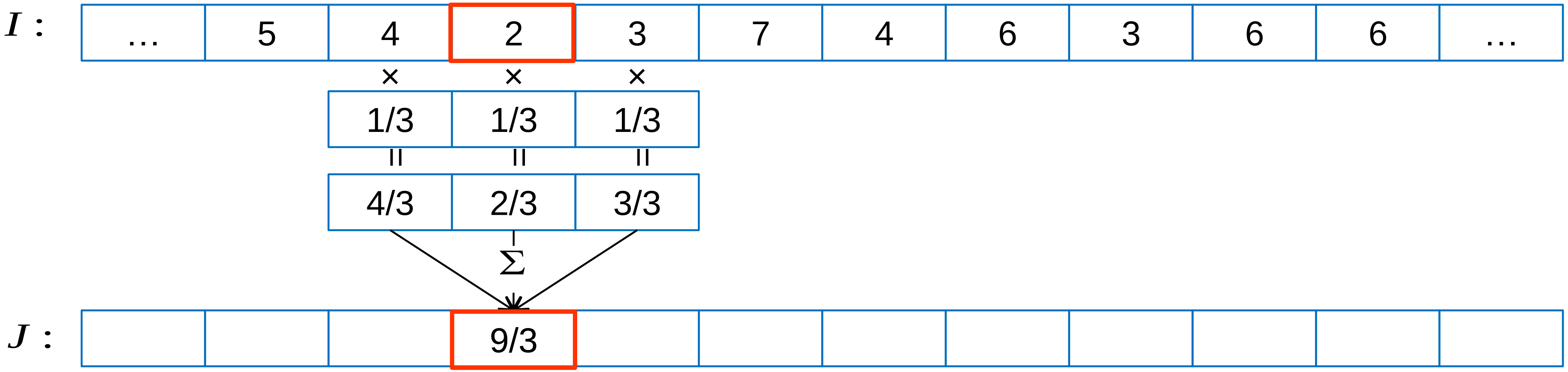
- **Linear**: every pixel is replaced by a linear combination of its neighbors

- **Shift-invariant**: the same operation is performed on every point on the image
- Basic & very useful filtering operations:
  - Correlation
  - Convolution

- Brief study of these filters in the simplest case of 1D images (i.e. a row of pixels) & their extension to 2D

Image filtering | correlation

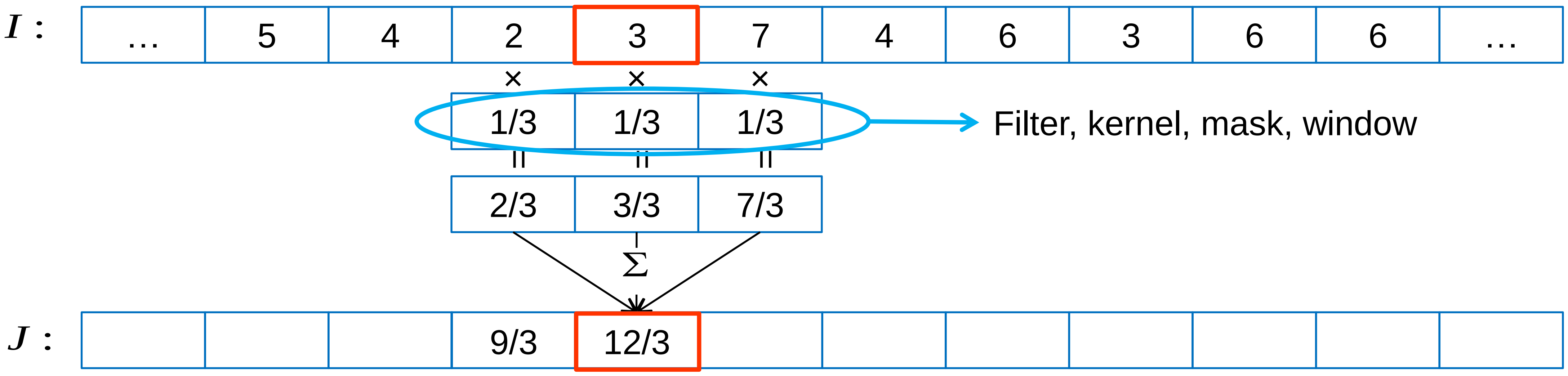
- An averaging filter



- How to handle boundaries?
  - Ignore filtered values at boundaries
  - Pad image with zeros
  - Pad image with first/last image values

Image filtering | correlation

- An averaging filter



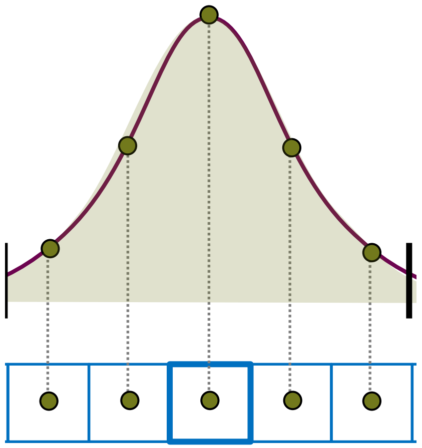
- Formally, Correlation is $J(x) = F \circ I(x) = \sum_{i=-N}^{N} F(i) I(x+i)$

- In this smoothing example $F(i) = \begin{cases} 1/3, & i \in [-1,1] \\ 0, & i \notin [-1,1] \end{cases}$

Other examples of smoothing filters?

Image filtering | constructing filter from a continuous f^n

- Common practice for image smoothing:
  use a Gaussian



Normalize filter so that values always add up to 1

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

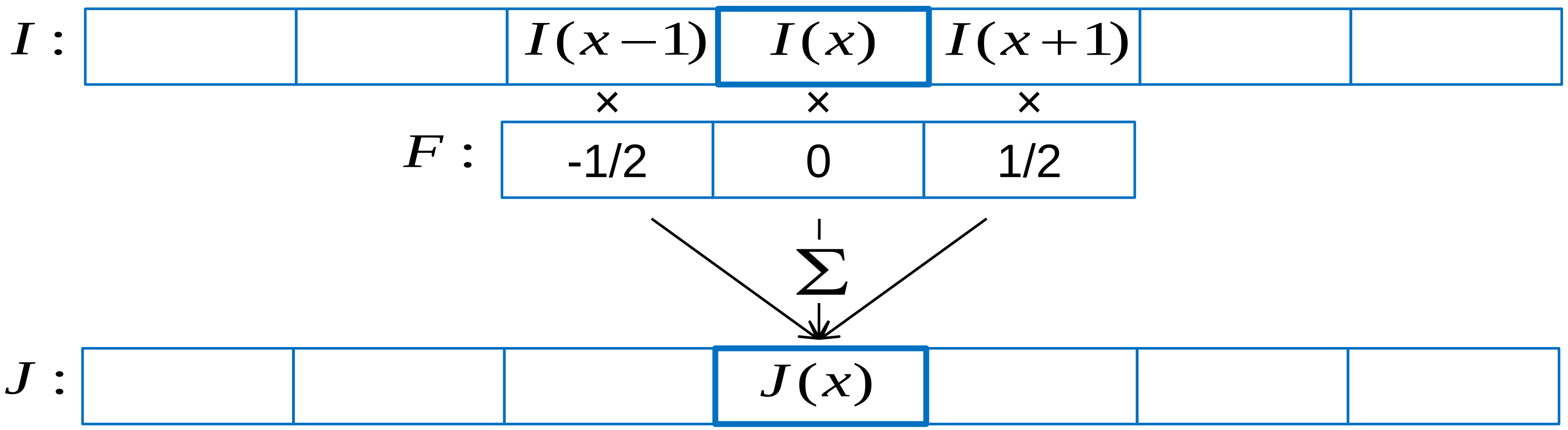
$$\mu = 0$$

σ : controls the amount of smoothing

- Near-by pixels have a bigger influence on the averaged value rather than more distant ones

Image filtering | taking derivatives with correlation

- Derivative of an image: quantifies how quickly intensities change (along the direction of the derivative)
- Approximate a derivative operator:



$$J(x) = \frac{I(x+1) - I(x-1)}{2}$$

Image filtering | matching using correlation

- Find locations in an image that are similar to a ***template***
- Filter = template

3	8	3

➪ test it against all image locations

I :

3	2	4	1	3	8	4	0	3	8	7	7

- Similarity measure: Sum of Squared Differences (**SSD**) – minimize

$$\sum_{i=-N}^{N}(F(i)-I(x+i))^2=\sum_{i=-N}^{i=N}(F(i))^2+\sum_{i=-N}^{i=N}(I(x+i))^2-2\sum_{i=-N}^{i=N}(F(i)I(x+i))$$

J :

26	37	21	50	54	1	50	65	59	16	42	17

Image filtering | matching using correlation

- Find locations in an image that are similar to a ***template***
- Filter = template

3	8	3

  ➪ test it against all image locations

I :

3	2	4	1	3	8	4	0	3	8	7	7

- Similarity measure: Sum of Squared Differences (**SSD**) – minimize

$$\sum_{i=-N}^{N}\left(F(i)-I(x+i)\right)^2=\sum_{i=-N}^{i=N}\left(F(i)\right)^2+\sum_{i=-N}^{i=N}\left(I(x+i)\right)^2-2\underbrace{\boxed{\sum_{i=-N}^{i=N}\left(F(i)I(x+i)\right)}}_{\text{Correlation}}$$

J :

26	37	21	50	54	1	50	65	59	16	42	17

- Similarity measure: Correlation? – maximize

J :

30	37	41	29	51	85	56	21	48	86	101	77

Image filtering | NCC: Normalized Cross Correlation

- Find locations in an image that are similar to a ***template***
- Filter = template

3	8	3

  ➪ test it against all image locations

I :

3	2	4	1	3	8	4	0	3	8	7	7

- Correlation value is affected by the magnitude of intensities
- Similarity measure: Normalized Cross Correlation (**NCC**) – maximize

$$\frac{\sum_{i=-N}^{i=N}\left(F(i)I(x+i)\right)}{\sqrt{\sum_{i=-N}^{i=N}\left(F(i)\right)^2}\sqrt{\sum_{i=-N}^{i=N}\left(I(x+i)\right)^2}}$$

J :

0.919	0.759	0.988	0.628	0.655	0.994	0.691	0.464	0.620	0.860	0.876	0.859

Image filtering | ZNCC: Zero-mean Normalized Cross Correlation

- Find locations in an image that are similar to a ***template***
- Filter = template

3	8	3

➪ test it against all image locations

I :

3	2	4	1	3	8	4	0	3	8	7	7

- For better invariance to intensity changes, try to eliminate all absolute effects on intensities

Similarity measure: Zero-mean Normalized Cross Correlation (**ZNCC**) – maximize

$$\frac{\sum_{i=-N}^{i=N}\left(F(i)-\mu_F\right)\left(I(x+i)-\mu_{I_x}\right)}{\sqrt{\sum_{i=-N}^{i=N}\left(F(i)-\mu_F\right)^2}\sqrt{\sum_{i=-N}^{i=N}\left(I(x+i)-\mu_{I_x}\right)^2}}$$

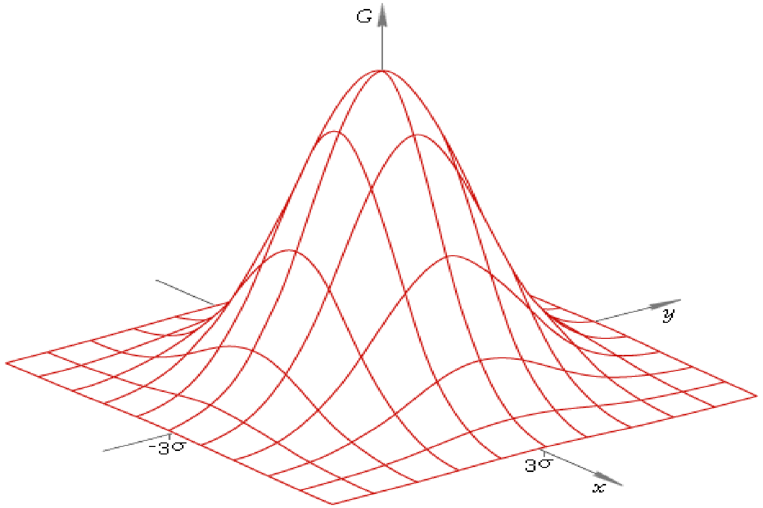
where

$$\begin{cases} \mu_F = \dfrac{\sum_{i=-N}^{N} F(i)}{2N+1} \\ \mu_{I_x} = \dfrac{\sum_{i=-N}^{N} I(x+i)}{2N+1} \end{cases}$$

Image filtering | 2D Gaussian smoothing

- Correlation in 2D: $F \circ I(x,y) = \sum_{j=-M}^{M} \sum_{i=-N}^{N} F(i,j) I(x+i, y+j)$
- A general, 2D Gaussian $G(x,y) = \frac{1}{2\pi |S|^{1/2}} e^{-\frac{1}{2}\begin{pmatrix} x \\ y \end{pmatrix} S^{-1} (x \quad y)}$
- We usually want to smooth by the same amount in both *x* and *y* directions $S = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}$
- So this simplifies to:

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} = \underbrace{\boxed{\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}}}_{G_\sigma(x)} \cdot \underbrace{\boxed{\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{y^2}{2\sigma^2}}}}_{G_\sigma(y)}$$



- A “separable” filter

Image filtering | convolution

- Convolution is ***equivalent*** to Correlation with a flipped filter before correlating
- CONVOLUTION: $J(x) = F * I(x) = \sum_{i=-N}^{N} F(i)I(x-i)$
- CORRELATION: $J(x) = F \circ I(x) = \sum_{i=-N}^{N} F(i)I(x+i)$

So if $F = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$
$F' = \begin{bmatrix} 3 & 2 & 1 \end{bmatrix}$
Then, $J(x) = J'(x)$

- Likewise, in 2D we flip the filter both horizontally & vertically

$$J(x,y) = F * I(x,y) = \sum_{j=-M}^{M} \sum_{i=-N}^{N} F(i,j)I(x-i,y-j)$$

- Key difference with correlation: **convolution is associative**: $F * (G * I) = (F * G) * I$
- Very useful!
- Example: smooth an image & take its derivative ⇨ convolve the Derivative filter with the Gaussian filter & convolve the resulting filter with the Image