EPFL

Sensing and spatial modeling for earth observation

Extracting relevant features

Devis TUIA

EPFL, spring semester 2025

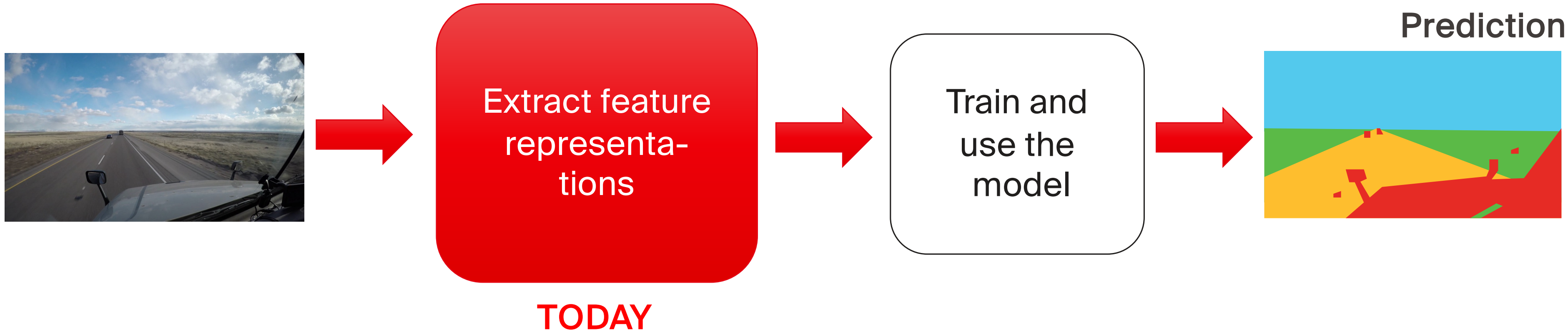
École polytechnique fédérale de Lausanne

We made it, we have a DEM!

- Now we have the input to predict environmental variables!
- Here we work with a DEM, but We could also use orthoimages, satellite images, etc. (but that's in another course ENV-540)

We made it, we have a DEM!

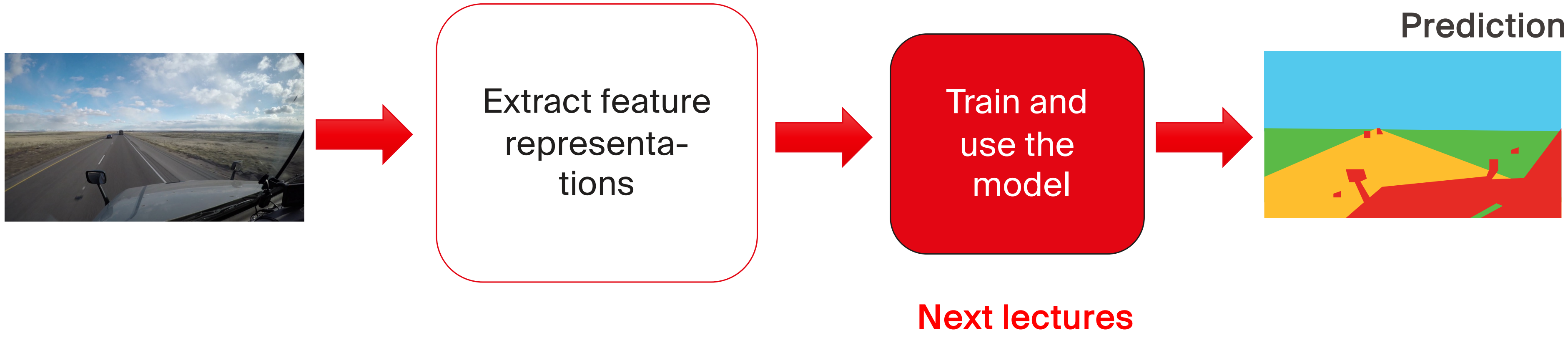
- Now we have the input to predict environmental variables!
- We could also use orthoimages, satellite images, etc.
- The structure to follow would look like:



EPFL

We made it, we have a DEM!

- Now we have the input to predict environmental factors!
- We could also use orthoimages, satellite images, etc.
- The structure to follow would look like:



Features

Why do we need features

Good properties for features

Types of features

EPFL

What are features?

- Features: new variables issued from the data that are more expressive to solve the problem

Visual properties that help the model
Make the right decision

Extract feature representa-tions

Train and use the model

Prediction

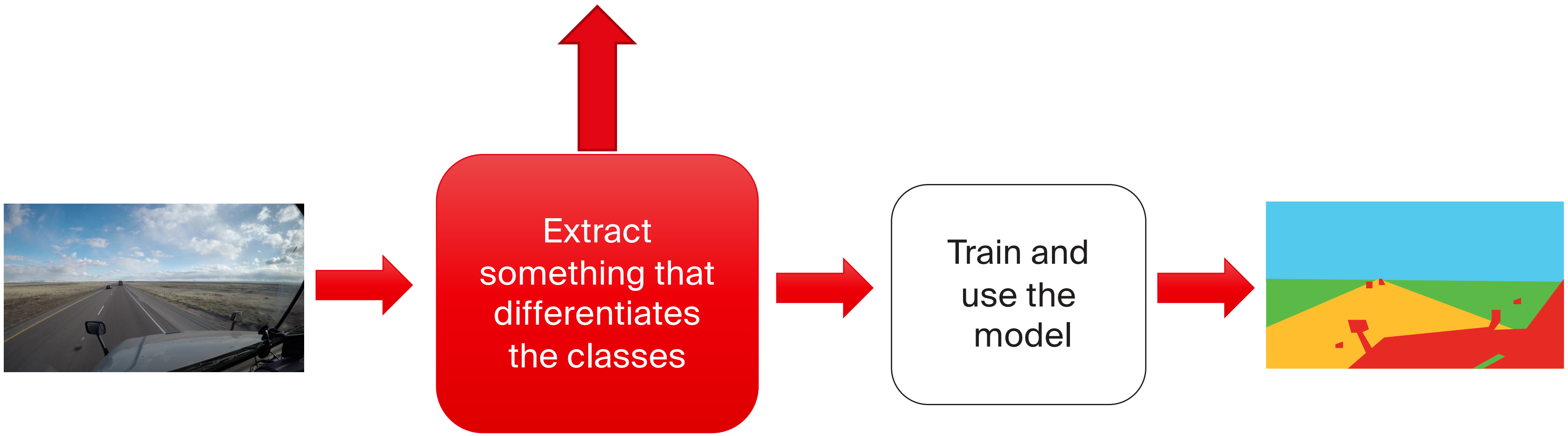
EPFL

“Features” sound like the descriptors of week 2...

- We can use the two terms interchangeably.
- For clarity, here I will use <u>features</u> for descriptors that are dense (= values for every pixel)
- This is in contrast to descriptors as those seen in the keypoints course (e.g. SIFT), where the description was computed only at the keypoint location

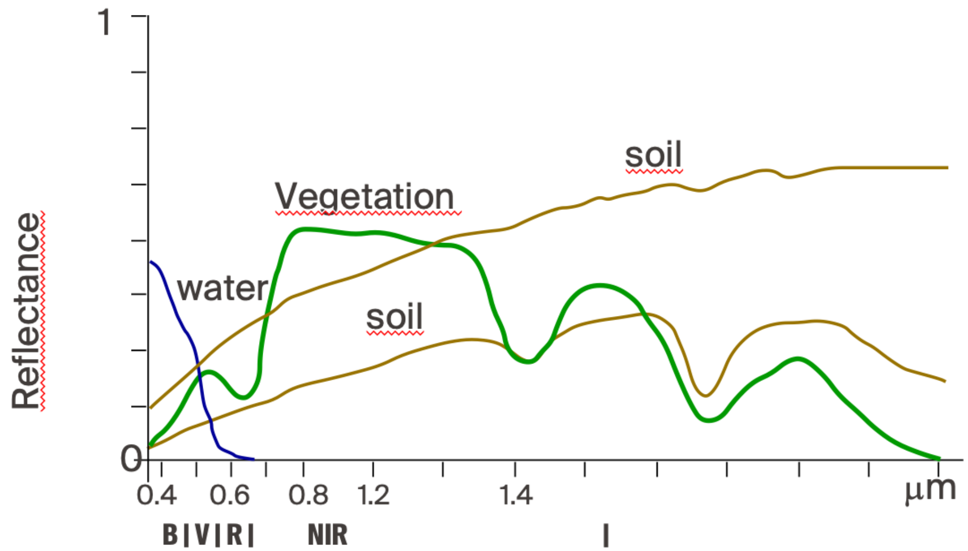
Examples of features

- Vegetation → features related to reflectivity of vegetation → vegetation indices from NIR bands
- Urban → features relative to the shape of objects → spatial context in visible bands
- Clouds → features relative to thermal reflectivity → TIR bands



“Good” vs “bad” features

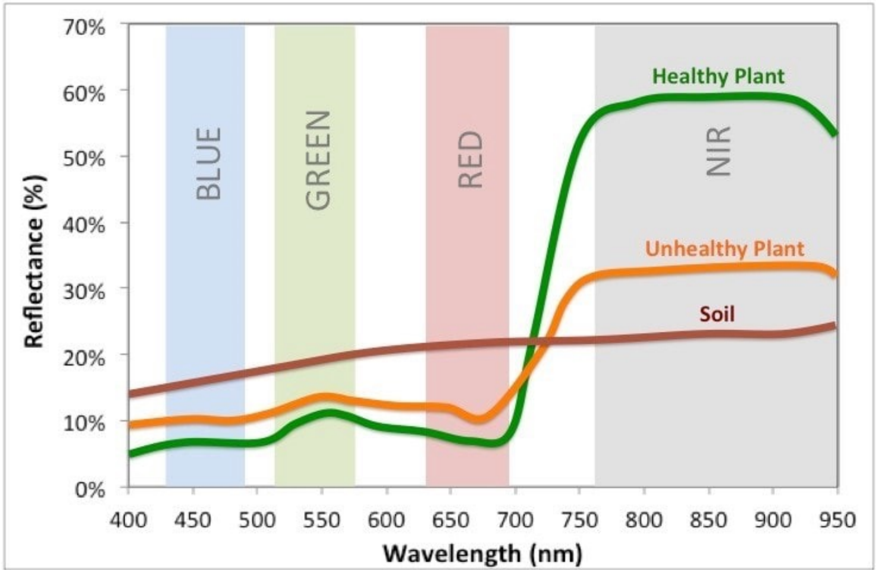
- In spectral images (as those we use in remote sensing) each surface is characterised by a **spectral signature**
- A sensor samples the true (continuous) signature according to its resolution
- E.g. 3 bands = 3 values.



“Good” vs “bad” features

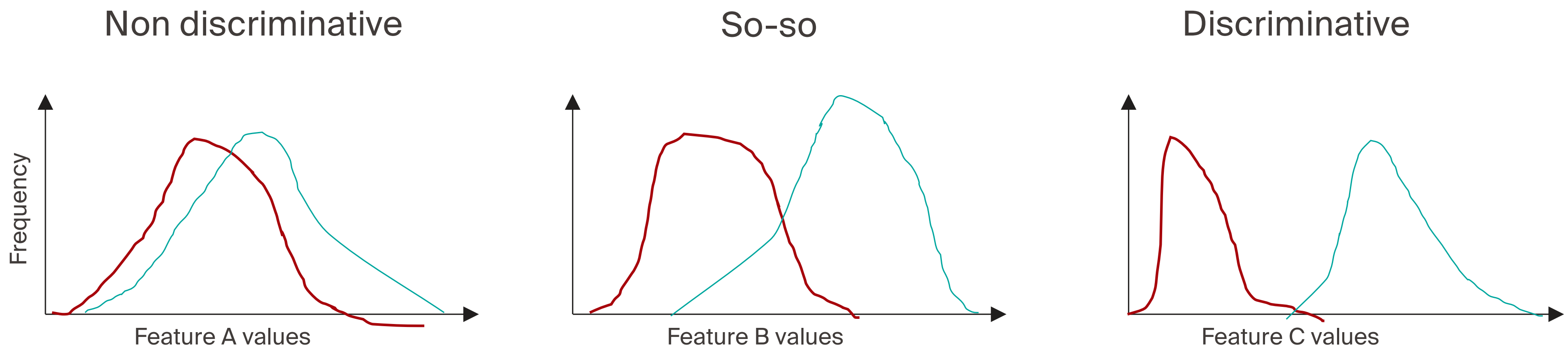
- In the same way, we can extract features that tailored to the problem at hand.
- A classical example is the NDVI
- By comparing infrared to red light, can highlight healthy vegetated surfaces
- So
  - it’s a **good** feature to detect vegetation
  - It’s a **bad** feature to detect cars

$$NDVI = \frac{x_i^{(NIR)} - x_i^{(R)}}{x_i^{(NIR)} + x_i^{(R)}}$$



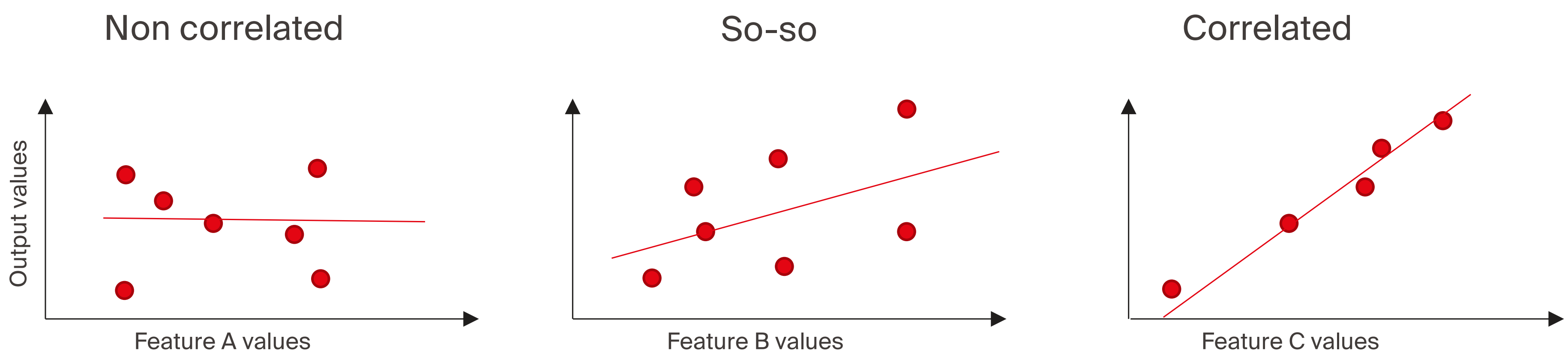
Classification: the aim is being discriminative

- Good features separate examples of a class from examples of the other
- Below: the histogram of three features for two classes (in red and green)



Regression: the aim is being correlated to the output

- Good features can lead to good fits in regression problems
- Below the scatterplots of three features for predicting a continuous output value
- A good feature correlates with the variable being predicted (output)



How many types of features are around?

- As many as you can imagine.
- They all are signal modifications, channels combinations, etc.
- In neural networks, you learn them from data (so no feature engineering) → Join ENV-540 if you want to know more.

A (very rough) taxonomy of features

- Spectral: band combinations at different wavelenghts (not covered today, see ENV-140/540).
- Spatial: accounting for the spatial context around the pixel you are looking at
  - Based on convolution windows (e.g. low/high pass filters).
  - Extracted using some machine learning pipeline (e.g. BoVW, ENV540).
- In the next part we focus on spatial features, based on convolutions, to be extracted from the DEM.

Convolution-based features

Low- and high-pass filters (a reminder)

Sobel filters

Why spatial filters?

- Spatial filters tell us about image context and improve discrimination

Why spatial filters?

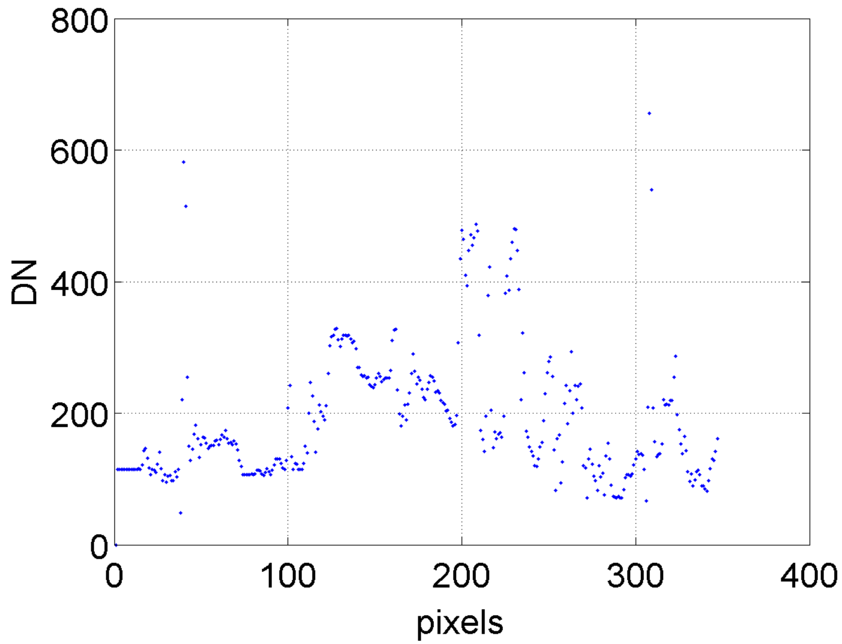
- Spatial filters tell us about image context and improve discrimination

Local enhancement

- One way to make images aware of context is to augment the input space with information about the surroundings.
- Augment = add new variables.
- We can create new features summarizing something about the local context (remember SIFT descriptions?).
- Each feature is a new variable telling us something about
  - Color distributions
  - Edges
  - Direction of spatial structures

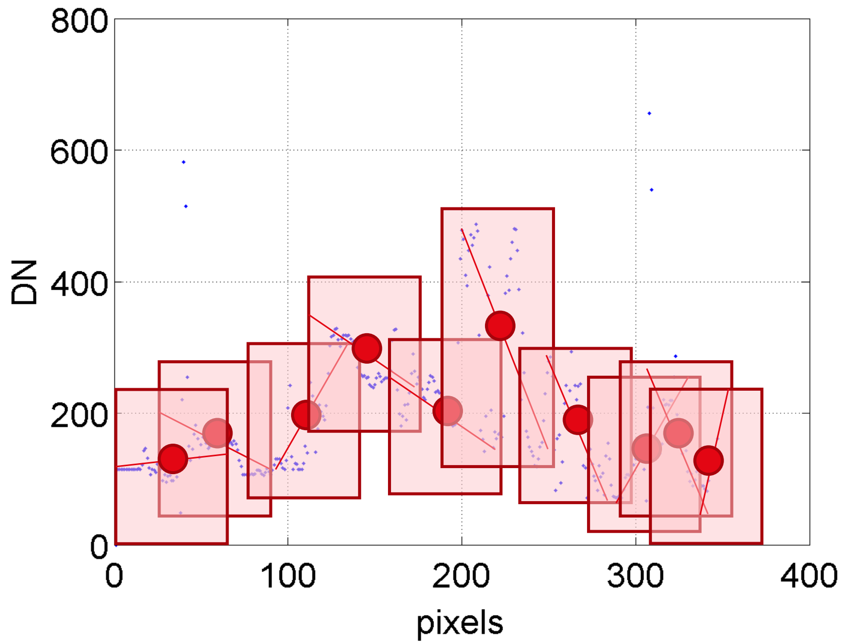
Local convolutions (1D)

- In 1 D, we can see each row (resp. column) of the image as a series of discrete values
- We can use a convolution window to summarize nearby values



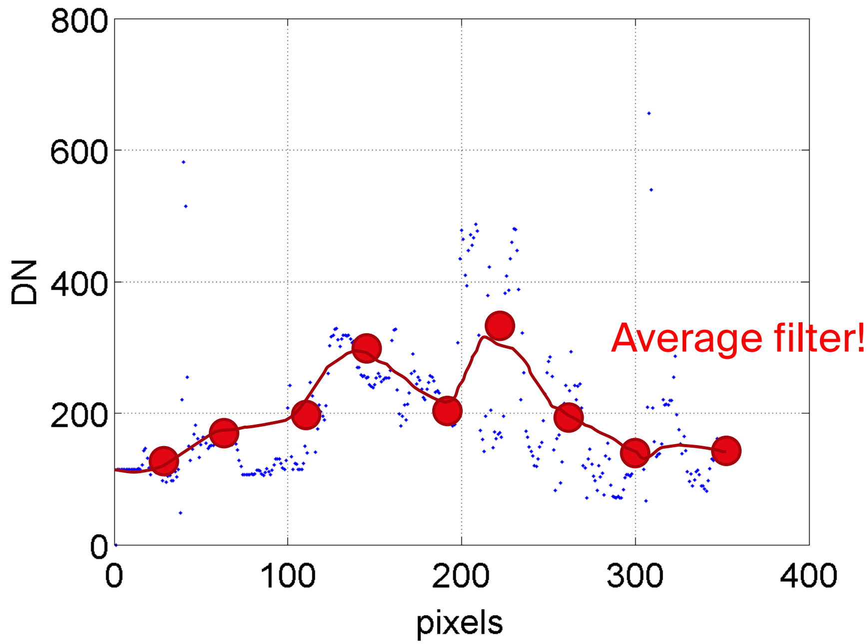
Local convolutions (1D)

- In 1 D, we can see each row (resp. column) of the image as a series of discrete values
- We can use a convolution window to summarize nearby values



Local convolutions (1D)

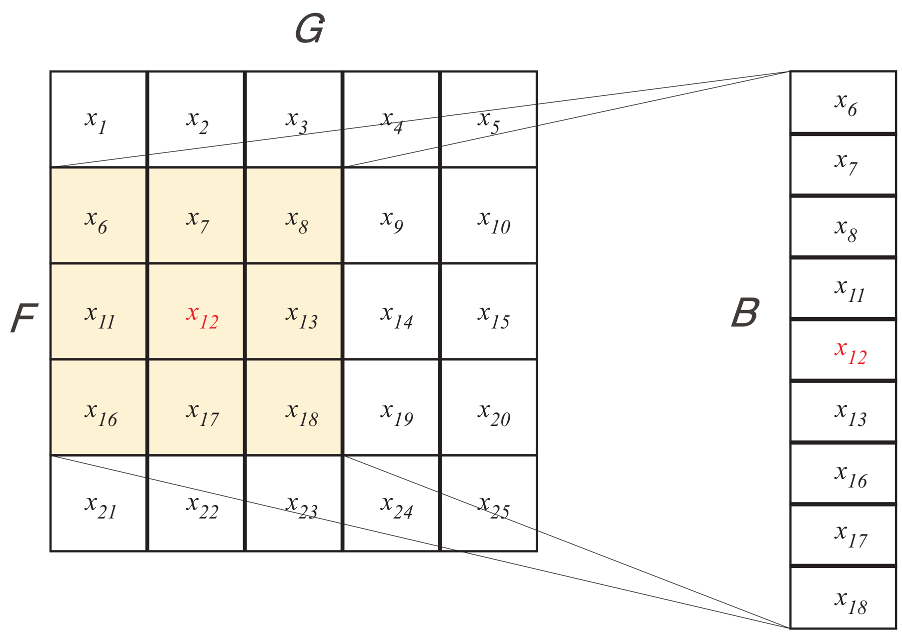
- In 1 D, we can see each row (resp. column) of the image as a series of discrete values
- We can use a convolution window to summarize nearby values



Convolution filters in 2D

D. Tuia

- We want to filter pixel x_{12} of the image G
- We use a 3 x 3 convolution filter F
- The pixels considered by the convolution are in the vector B



Convolution operation

- A dot product is applied between the moving window B and the filter F
- For a filter of size with C coefficients and sum of coefficients S
  (ex: for F of the previous slide, C = 9)

$$x_{12}^{F} = \frac{1}{S}\left(F^{T} * B\right) = \frac{1}{S}\sum_{i=1}^{C} F_i B_i$$

Low pass filter: average

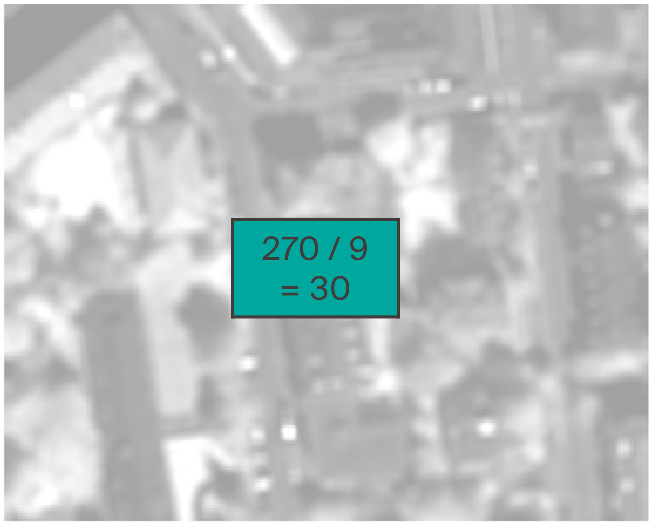
- A low pass filter smooths the image (the resulting feature is the average of all the points in the window)

$$* \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} =$$

Low pass filter: average

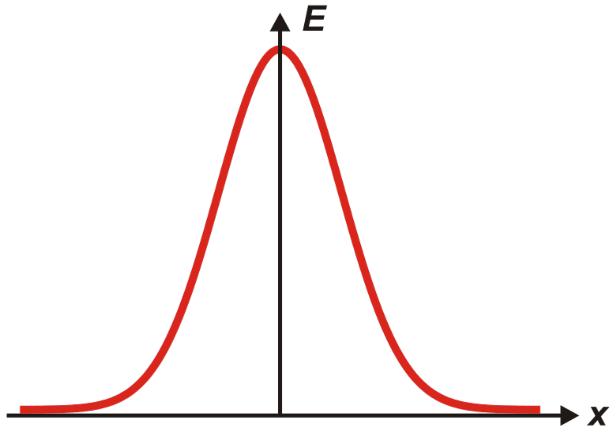
- A low pass filter smooths the image (the resulting feature is the average of all the points in the window)

$$* \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} =$$



Low pass filter: Gaussian

- With the average filter, the smoothing is often too strong
- A Gaussian filter weights the coefficients with respect to distance



Low pass filter: Gaussian

$$* \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} =$$

Comparison

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

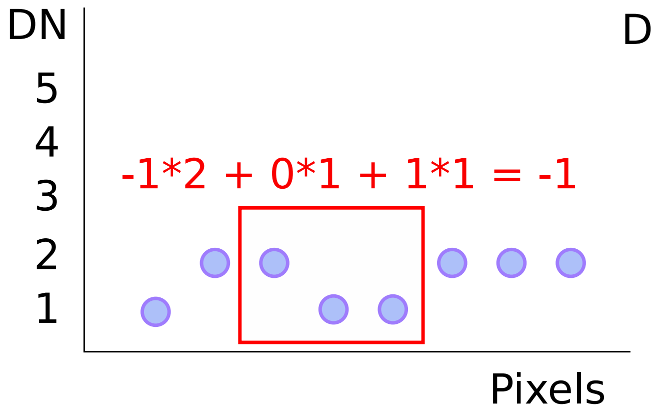
What about edges? Sobel filter

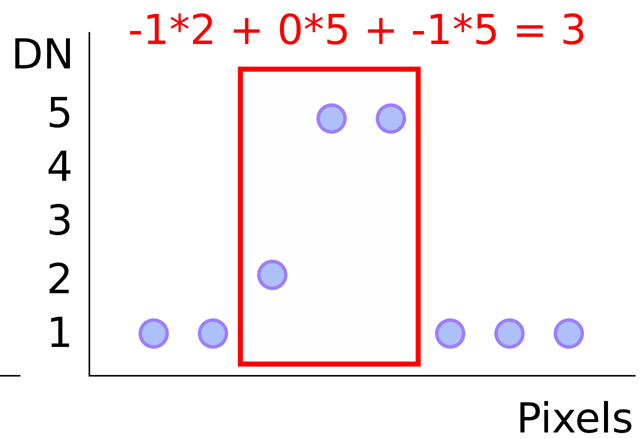
- Sobel filters are a family of directional high pass filters
- They work on image gradients
- The filters are usually convolutions in 2D, e.g. $\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$
- They are computed in different directions, then results averaged.

Recap: Filters based on gradients in 1D

$$\begin{pmatrix} -1 & 0 & 1 \end{pmatrix}$$

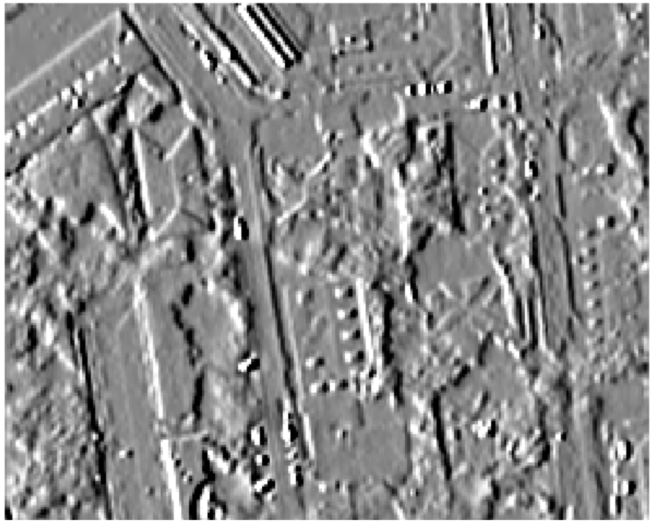
- It is a differentiation operator
- Gives the direction of the largest possible increase in intensity
- Shows how smoothly the image changes at that point
  - If an edge, the value is large
  - If not, the value is small





Sobel filter: horizontal derivative = vertical structures

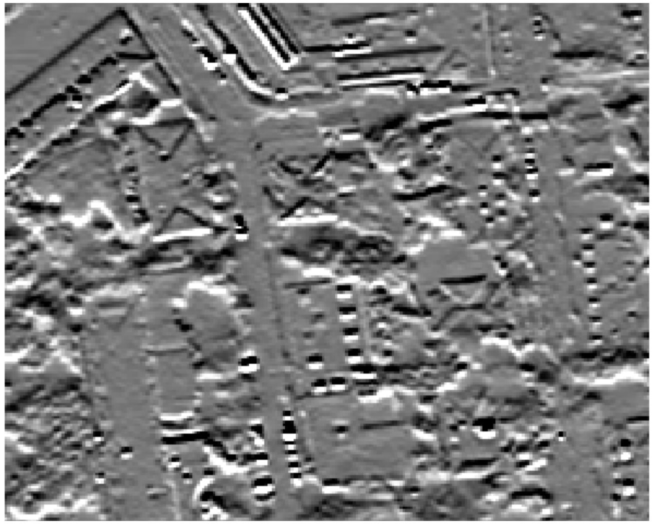
$$* \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} =$$



I_V

Sobel filter: vertical derivative = horizontal structures

$$* \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} =$$



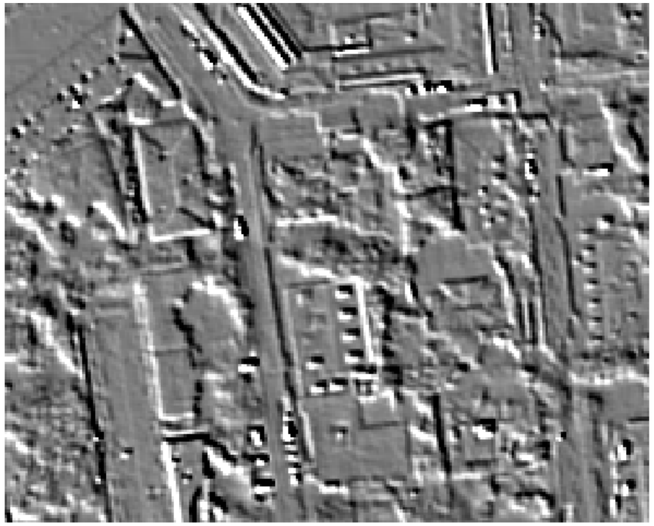
I_H

Sobel filter: first diagonal component

$$* \begin{pmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{pmatrix} =$$

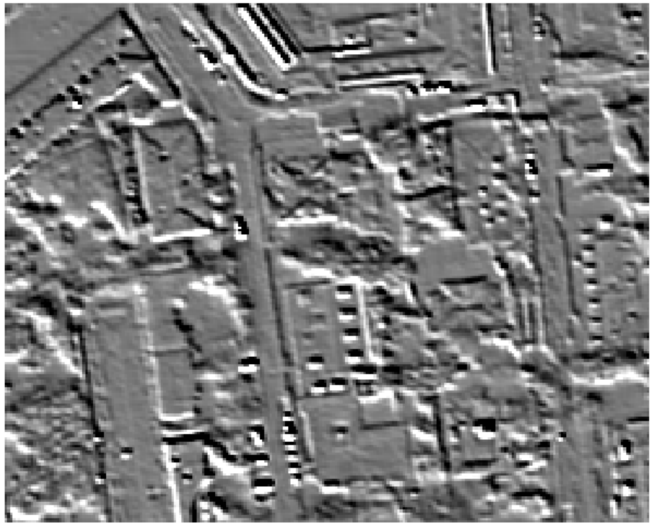
Sobel filter: second diagonal component

$$* \begin{pmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{pmatrix} =$$



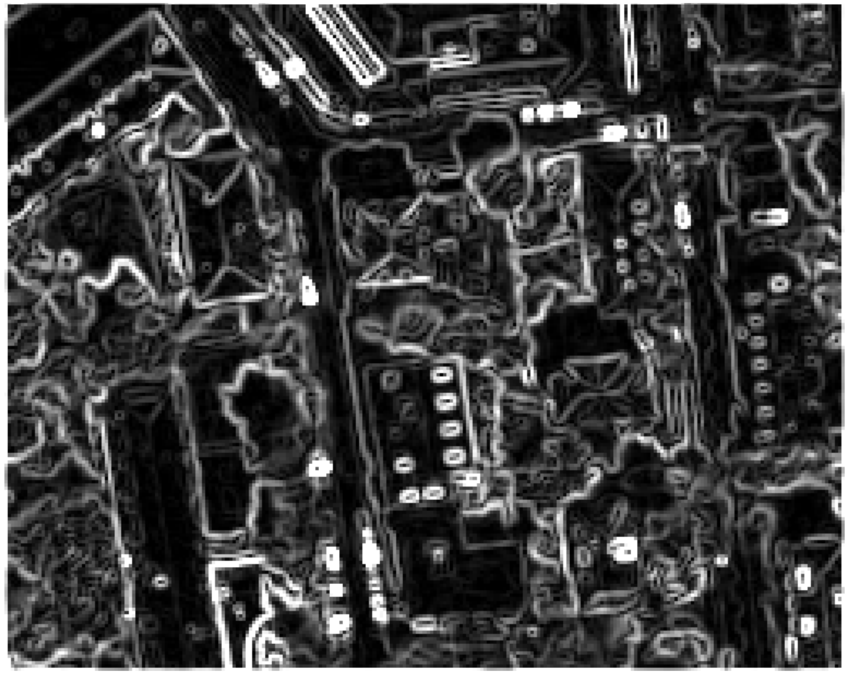
Sobel filter: isotrope filter

$$*(S_h + S_v + S_{d1} + S_{d2}) =$$



Sobel filter

$$\sqrt{I_H^2 + I_V^2}$$



Convolution filters for DEMs

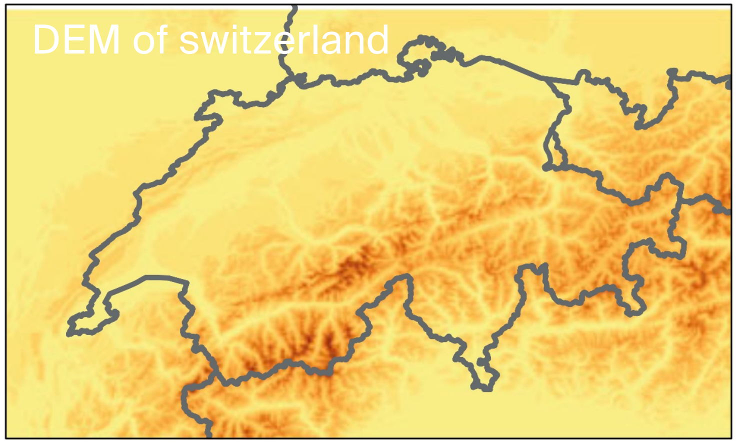
DoG

Directional derivatives

Slope

Convolution filters for DEMs

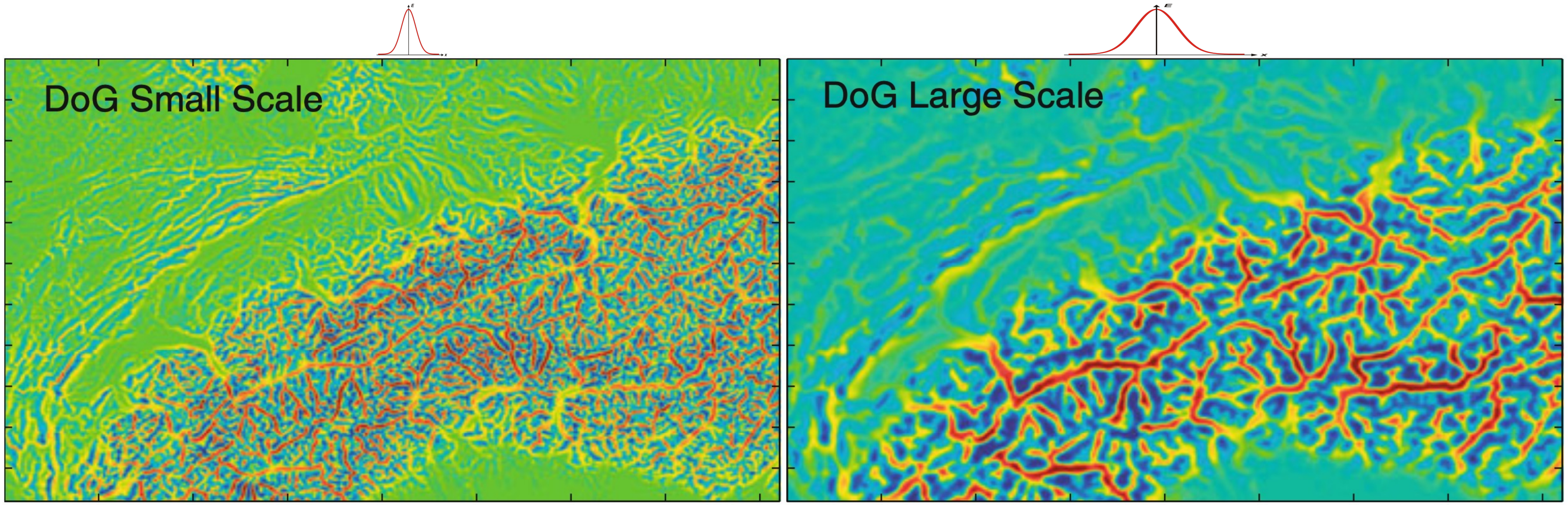
- Since we work on digital elevation models here, there are some favorites that can be computed on DEMs
- They use the ingredients we saw before



From: L. Foresti, D. Tuia, M. Kanevski, and A. Pozdnoukhov. Learning wind fields with multiple kernels. *Stoch. Env. Res. Risk. Ass.*, 25(1):51–66, 2011.

1. Difference of Gaussians

- We calculate two Gaussian filters = are two blurred DEMs
- Blurred at two different scales (two different Gaussian σ values)



From: L. Foresti, D. Tuia, M. Kanevski, and A. Pozdnoukhov. Learning wind fields with multiple kernels. *Stoch. Env. Res. Risk. Ass.*, 25(1):51–66, 2011.

1. Difference of Gaussians

- We calculate two Gaussian filters = two blurred DEMs
- Blurred at two different scales (two different Gaussian σ values)
- Then we substract them
- Depending on the σ values, different details will appear.

From: L. Foresti, D. Tuia, M. Kanevski, and A. Pozdnoukhov. Learning wind fields with multiple kernels. *Stoch. Env. Res. Risk. Ass.*, 25(1):51–66, 2011.

1. Difference of Gaussians

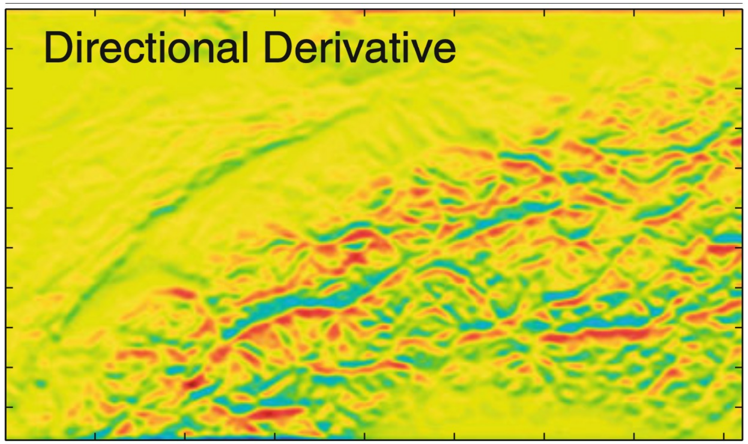
- We calculate two Gaussian filters = two blurred DEMs
- Blurred at two different scales (two different Gaussian σ values)
- Then we substract them
- Depending on the σ values, different details will appear.

- Yes, it is the same DoG seen for the SIFT detector (see course 2).

EPFL

2. Directional derivatives

- DD show the main gradients of the image in a specific direction
- It is basically the horizontal or vertical component of the Sobel filter
- E.g. here the horizontal derivative:



From: L. Foresti, D. Tuia, M. Kanevski, and A. Pozdnoukhov. Learning wind fields with multiple kernels. *Stoch. Env. Res. Risk. Ass.*, 25(1):51–66, 2011.

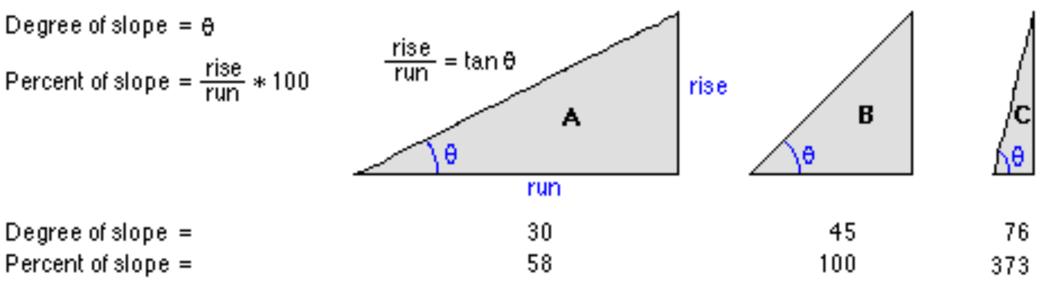
3. Slope

- Slope is formally described by a plane at a tangent to a point on a surface
- Slope has two components:
  - Gradient: the maximum rate of change of the elevation of the plane the angle that the plane makes with a horizontal surface. Often referred to as slope.
  - Aspect: the direction of the plane with respect to some arbitrary zero (usually north)

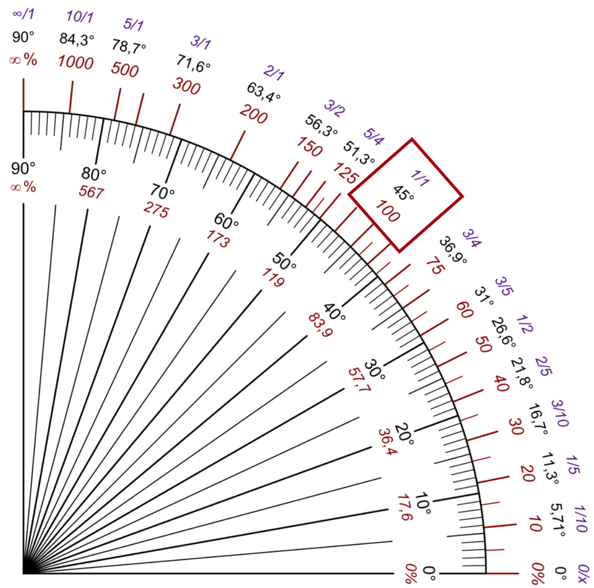
http://www.geo.uzh.ch/microsite/geo372/PDF/week4_geo372_terrain.pdf

3. Slope: gradient

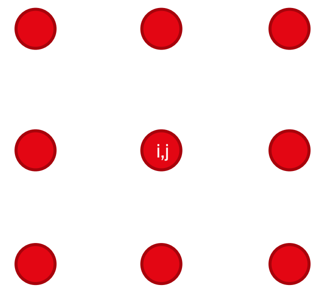
- Gradients can be calculated both in degrees (angle) or percent (rise vs run).



Comparing values for slope in degrees versus percent



3. Slope: steepest drop method

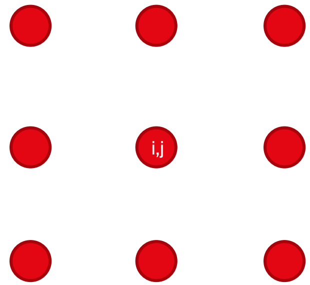


- There are many methods to calculate gradients
  - Steepest drop: use a focal function for max drop

$$gradient = \max_{a,b \in [-1,0,1]} \phi_i \frac{z_{i,j} - z_{i+a,j+b}}{\lambda}$$

where $\phi_i = \begin{cases} 1 & for N, S, W, E \\ \frac{1}{\sqrt{2}} & for NE, SE, NW, SW \end{cases}$ and λ is the resolution.

3. Slope: steepest drop method

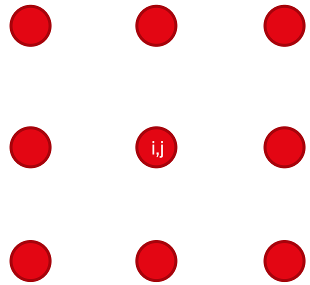


- There are many methods to calculate gradients
  - Steepest drop: use a focal function for max drop

- Aspect is the direction of steepest drop
- Pros: simple
- Cons: max 8 possible aspects

EPFL

3. Slope : finite differencing method

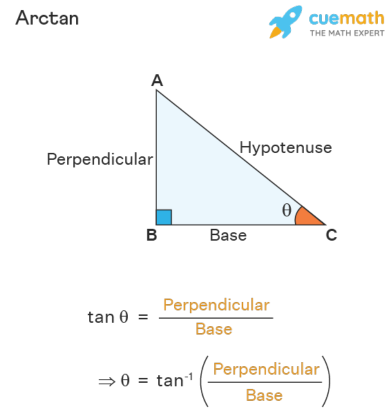


- There are many methods to calculate gradients
  - Steepest drop: use a focal function for max drop
  - Finite differencing

$$\text{gradient} = tan^{-1}\sqrt{\left(\frac{\partial z}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial y}\right)^2}$$

This gradient is (perp./base) in the horizontal

This gradient is (perp./base) in the vertical



http://www.geo.uzh.ch/microsite/geo372/PDF/week4_geo372_terrain.pdf

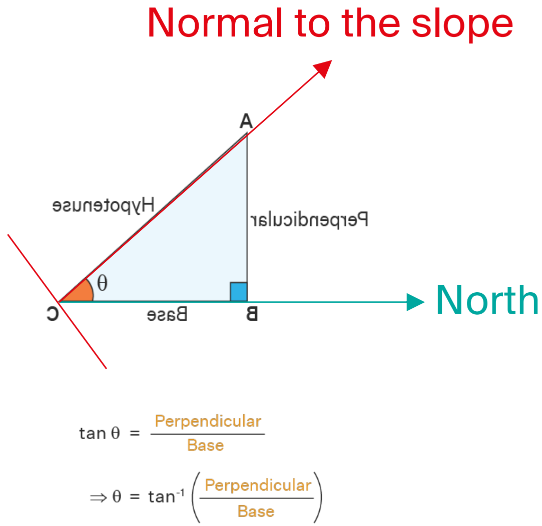
EPFL

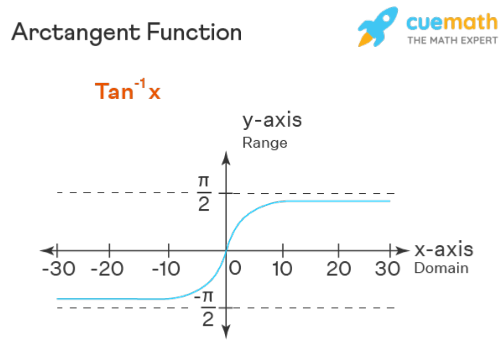
3. Slope : finite differencing method

- For the aspect, we use: $\text{aspect} = -tan^{-1}\left[\frac{\left(\frac{\partial z}{\partial y}\right)}{\left(\frac{\partial z}{\partial x}\right)}\right]$

all aspects are in the range [-π/2, π/2]
[-90, 90]

- This is ambiguous for 2D coordinates!

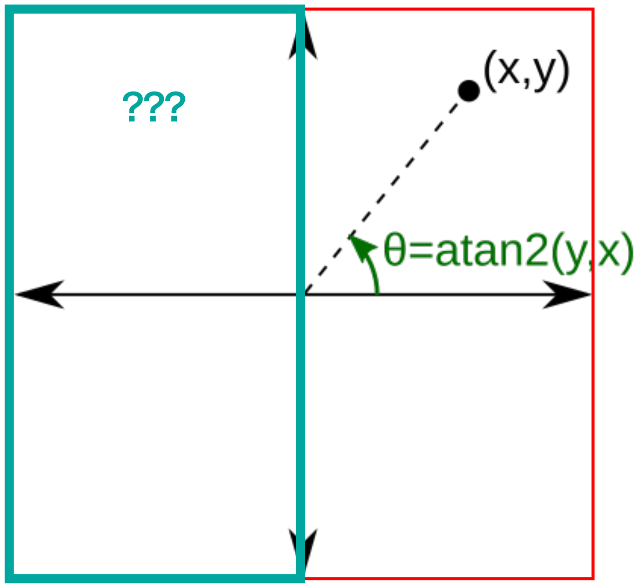




http://www.geo.uzh.ch/microsite/geo372/PDF/week4_geo372_terrain.pdf

3. Solving the ambiguity of aspect

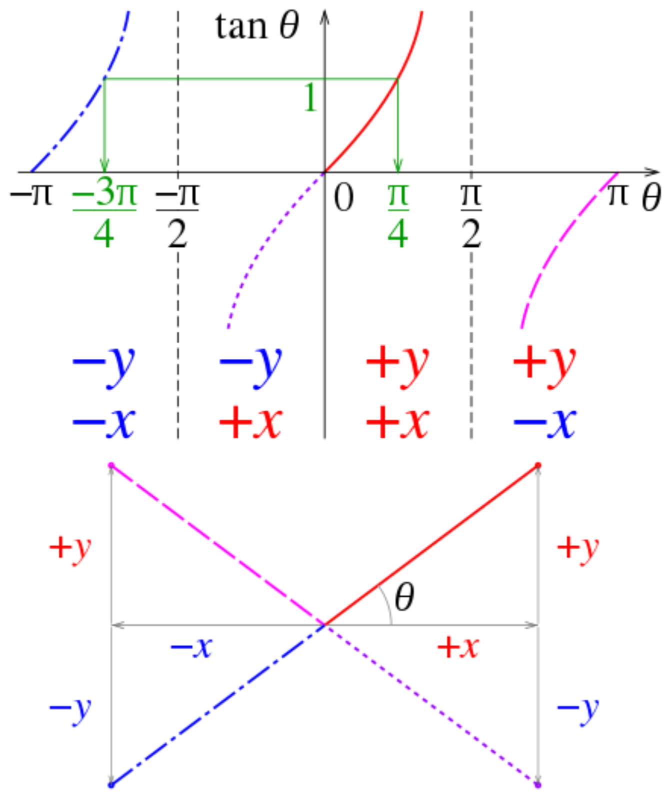
- If calculating the angle between the x axis and a vector in the 2D plane to a point (x,y), we cannot reach the left quadrant
- This is because y/x and –y/-x give the same result, so we don't know in which quadrant we will be



Range of validity of the $\tan^{-1}(y/x)$ fct!

3. Solving the ambiguity of aspect

- If calculating the angle between the x-axis and a vector in the 2D plane to a point (x,y), we cannot reach the left quadrant
- This is because y/x and –y/-x give the same result, so we don't know in which quadrant we will be
- To solve this abiguity, use the arctan2 fct, which works anywhere in the Cartesian plane
- It uses the sign of x and y to locate the right quadrant

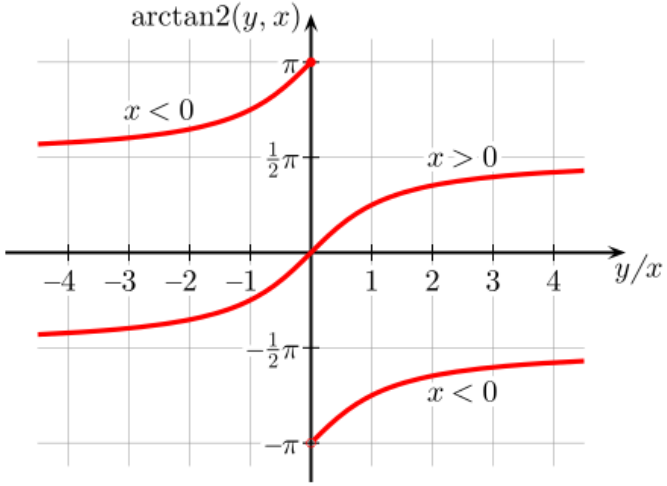


https://en.wikipedia.org/wiki/Atan2

3. Solving the ambiguity of aspect

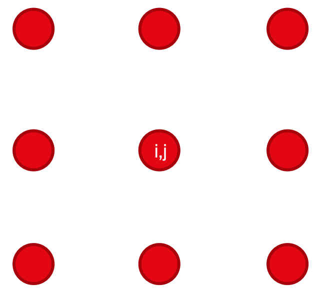
- To have aspects in [0,360], use the arctan2 function:

$$\text{aspect} = -tan^{-1}\left[\left(\frac{\partial z}{\partial y}\right), \left(\frac{\partial z}{\partial x}\right)\right]$$

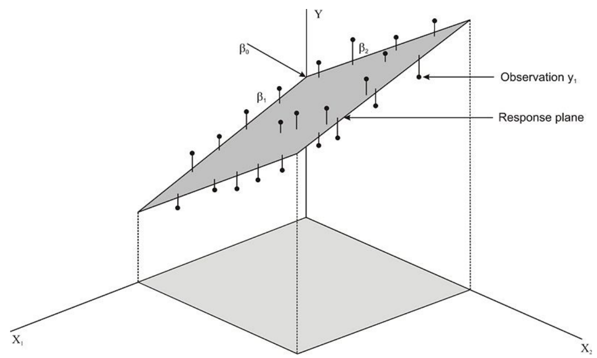


https://geo.libretexts.org/Courses/University_of_California_Davis/GEL_056%3A_Introduction_to_Geophysics/Geophysics_is_everywhere_in_geology.../zz%3A_Back_Matter/Arctan_vs_Arctan2

3. Slope: quadratic fit method

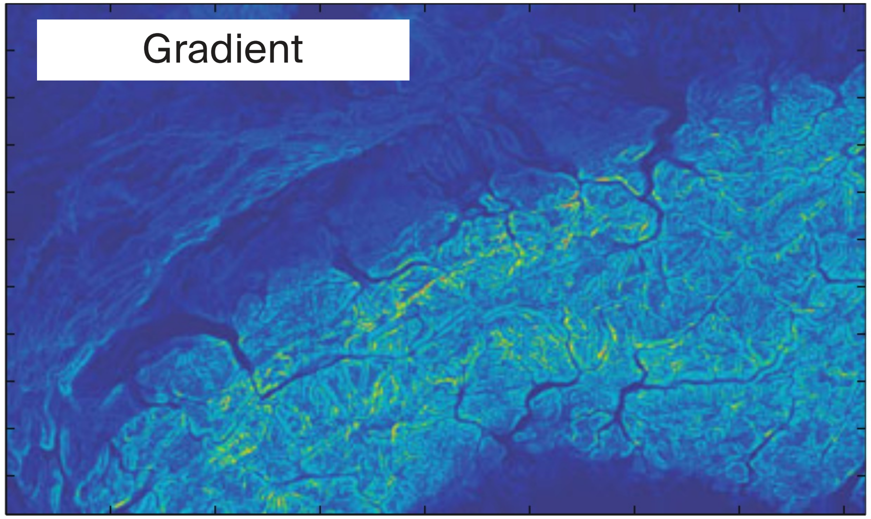


- There are many methods to calculate gradients
  - Steepest drop: use a focal function for max drop
  - Finite differencing
  - Fit a quadratic surface to the points, minimizing errors

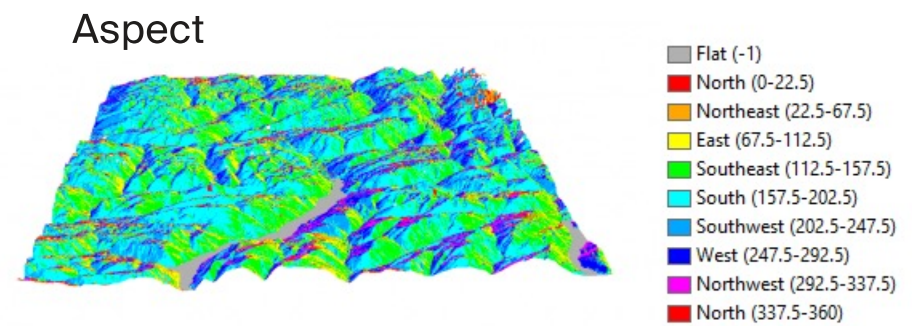


3. Slope

- Exemples of gradient and aspect features



From: L. Foresti, D. Tuia, M. Kanevski, and A. Pozdnoukhov. Learning wind fields with multiple kernels. *Stoch. Env. Res. Risk. Ass.*, 25(1):51–66, 2011.



https://gisgeography.com/aspect-map/

Features specific to images

Going further

There are a lot of local descriptors for remote sensing, here a selection:

- **Gray-level co-occurrence matrix (GLCM)-based** (notebook here, paper using them here)
- **Mathematical morphology-based** (https://hal.inria.fr/hal-00737075/document)
- Attribute profiles (https://ieeexplore.ieee.org/document/5482208)
- Dense SIFT (you calculate the SIFT descriptor at each pixel location in a sliding window fashon)
- Features going beyond local patterns: local-to-global features describing recurrent patterns in the image: Bag of Visual Words (ENV-540)

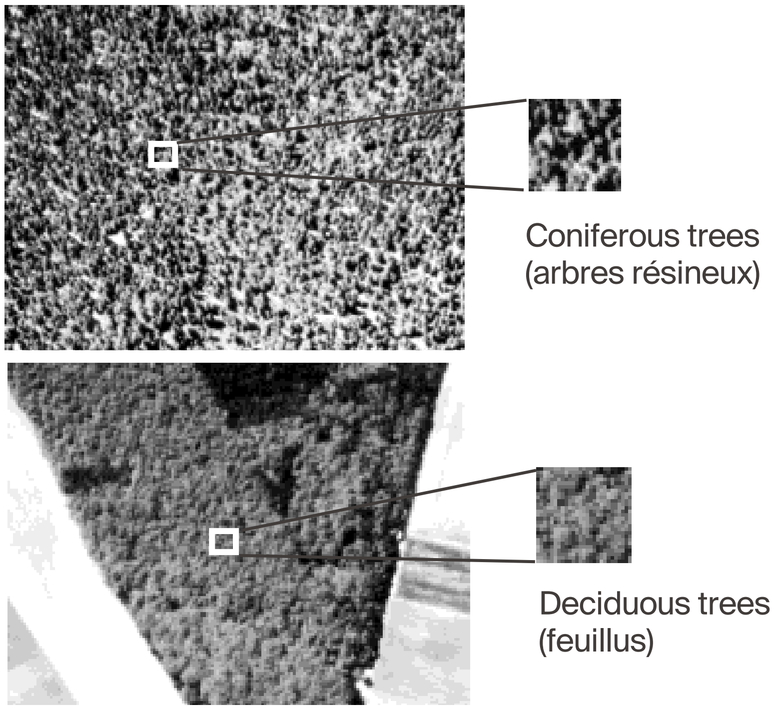
Going contextual

- In this last part, we see two family of contextual filters
  - Texture filters: compute 1st and 2nd order textural indices in local neighborhoods
  - Morphological filters: consider images as a terrain and work in filling “valleys” or erode “slopes”

Texture

- It can be seen as the visual aspect of the images, when taken locally
- They differentiate homogeneous areas from inhomogeneous areas
- They are invariant in translation and rotation (it's a property that holds for a type of region, come back to this later...)

An example of texture



Source: Caloz, 2001

- Here we have two types of tree coverages
- Visually it is easy to differentiate them
- How to incorporate this knowledge into relevant indices?

1st order texture: occurrence

- 1st order textures are local indices computed in moving windows
- Local statistics used
  - Mean
  - Variance
  - Range
  - …
- They are also called occurrence indices

Occurrence indices

$$V \in \left\{ \left[i - \frac{m}{2}; i + \frac{m}{2} \right], \left[j - \frac{m}{2}; j + \frac{m}{2} \right] \right\}$$

Is a local neighborhood of size m x m

- Mean

$$\mu_{ij} = \frac{1}{PP} \sum_{m \in V} \sum_{n \in V} x_{mn}$$

Local average. Same as low-pass filter

- Variance

$$\text{var}_{ij} = \frac{1}{PP} \sum_{m \in V} \sum_{n \in V} \left[x_{mn} - \mu_{ij} \right]^2$$

Fluctuation around the mean in the local window

Occurrence indices

Mean, 3x3

Mean, 5x5

Variance, 3x3

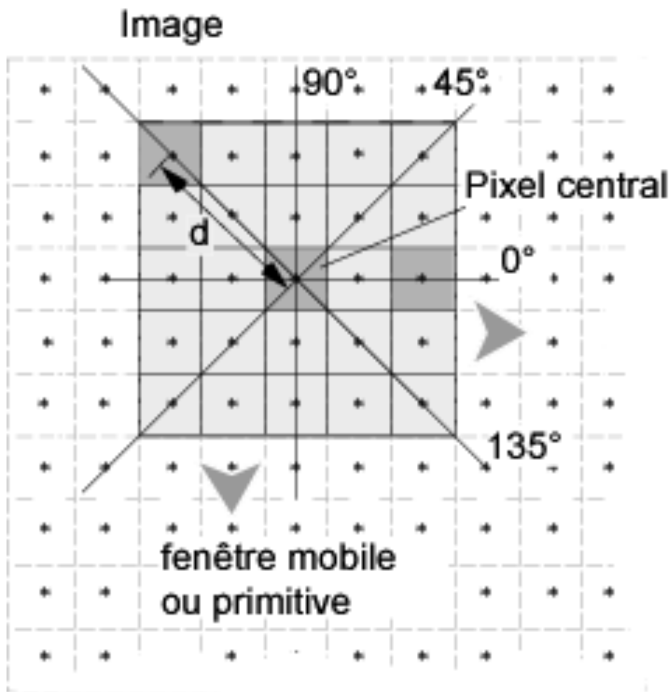
Variance, 5x5

2nd order indices: co-occurrence

- They consider spatial information between pixels
- Textural relations are made according to distance and orientation
- Basically searches autocorrelation between graylevels
- To compute them, we use the grey level co-occurrence matrix (GLCM, Haralick, 1973)

Co-occurrence matrix

- We take a mobile window and count the co-occurrences of the same DN for a given
- Distance d
- Direction q
- Each pixel has its own matrice!!!



Source: Caloz, 2001

Toy example

34	35	36	35	36
36	36	35	34	35
36	35	35	35	34
34	35	35	35	34
36	34	35	34	36

P(d=1, θ=0) = vertical

	34	35	36
34	2	5	4
35	5	10	4
36	4	4	2

P(d=1, θ=45) = 45º diagonal

	34	35	36
34	2	4	1
35	4	12	2
36	1	2	4

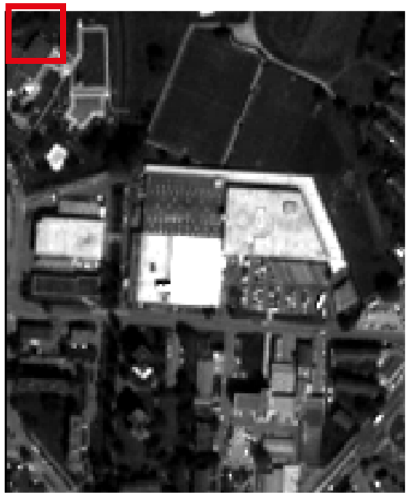
P(d=1, θ=90) = horizontal

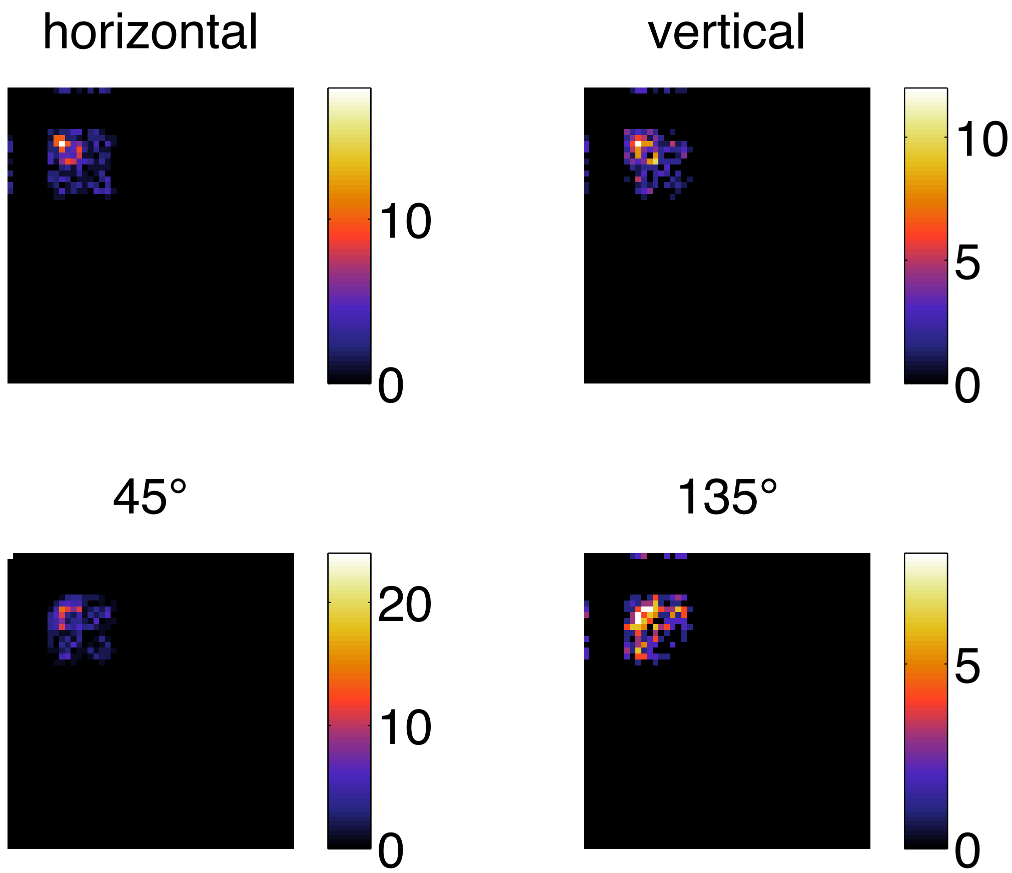
	34	35	36
34	0	8	2
35	8	8	5
36	2	5	2

P(d=1, θ=135) = 135º diagonal

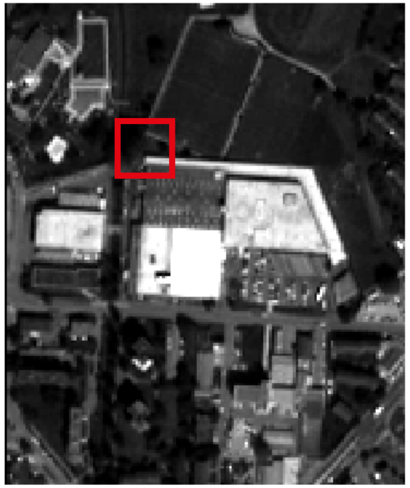
	34	35	36
34	4	2	2
35	2	12	4
36	2	4	0

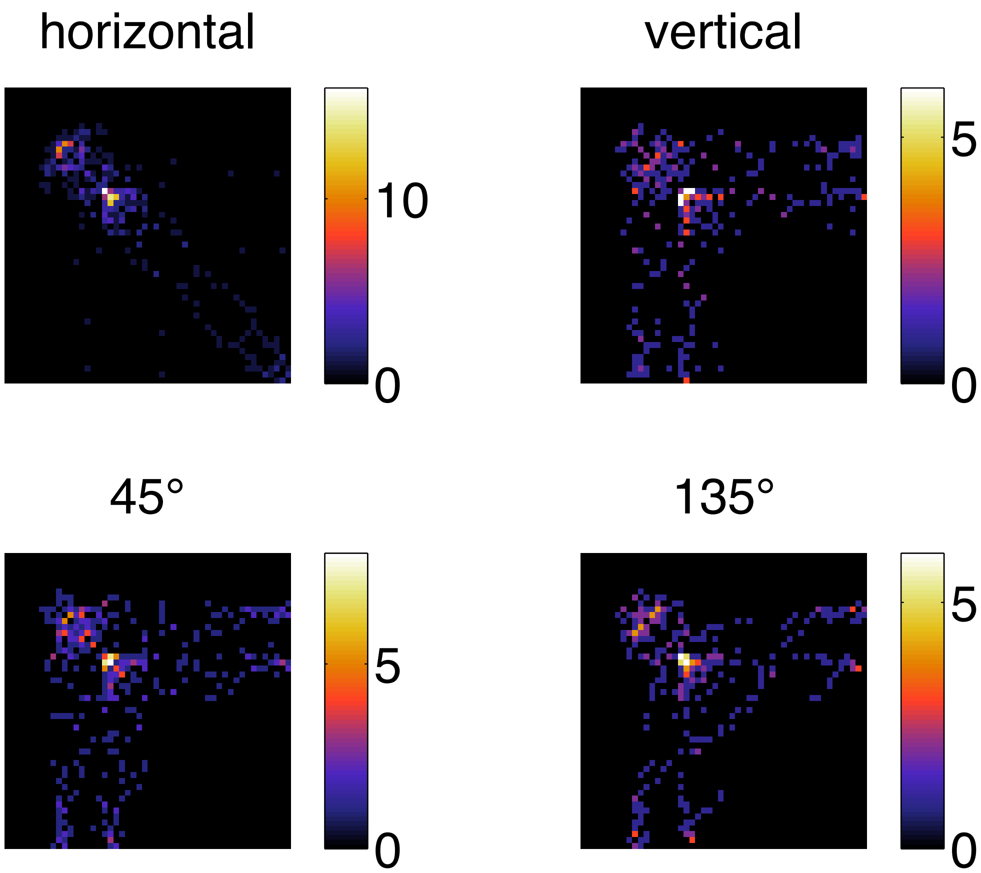
Real example (d=3)





Real example (d=3)





Co-occurrence indices

- Once we have the GLCM, we can compute the co-occurrence indices

- Entropy

$$H_{ij} = -\sum_{m=1}^{Nbits}\sum_{n=1}^{Nbits} P_{d,\theta}(m,n)\log\left(P_{d,\theta}(m,n)\right)$$

Low when few values in the window are Present among Nbits

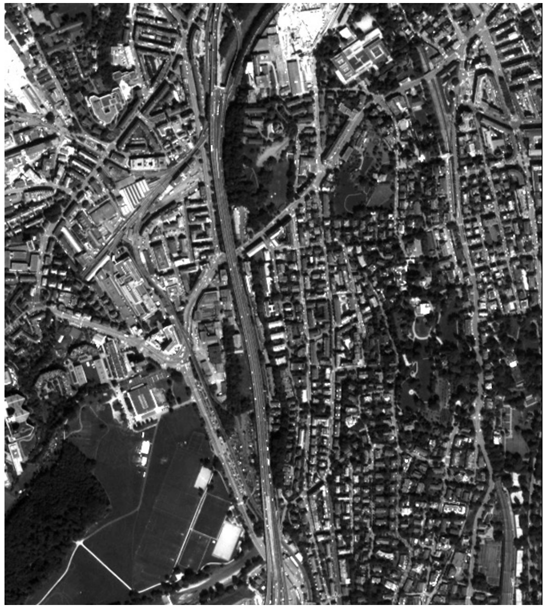
- Contrast

$$C_{ij} = \sum_{m=1}^{Nbits}\sum_{n=1}^{Nbits} (m-n)^2 P_{d,\theta}(m,n)$$

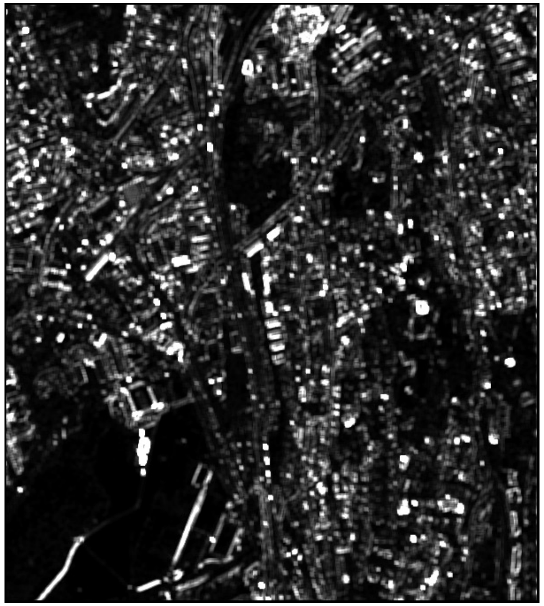
Enhances strong jumps in DNs

(Nbits is the number of rows of the GLCM)

Real examples



Image



Contrast, 5x5,d=3, average angles

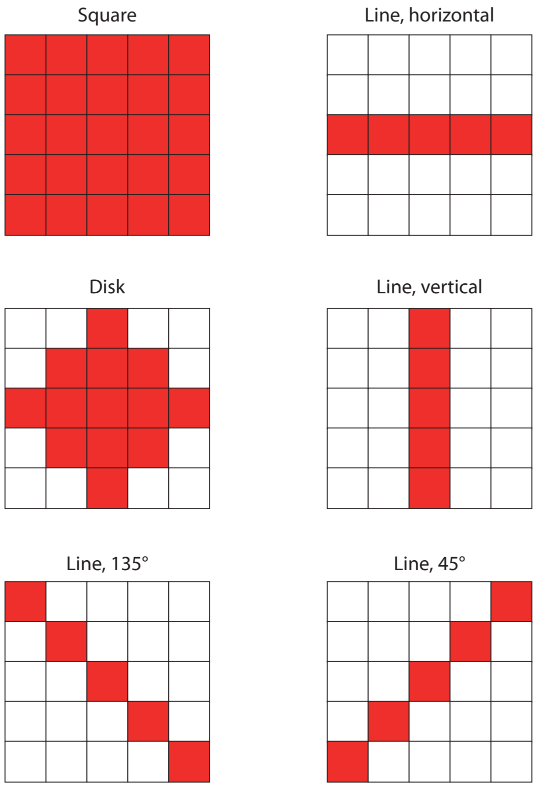
Entropy, 5x5,d=3, average angles

Morphology

- Filters based studying local intensity properties of the image
- They are based on three ingredients: the image, the structuring element and the operator.
- Most used morphology operators return filtered images enhancing elements that are darker or brighter than their surroundings

Morphology: structuring element

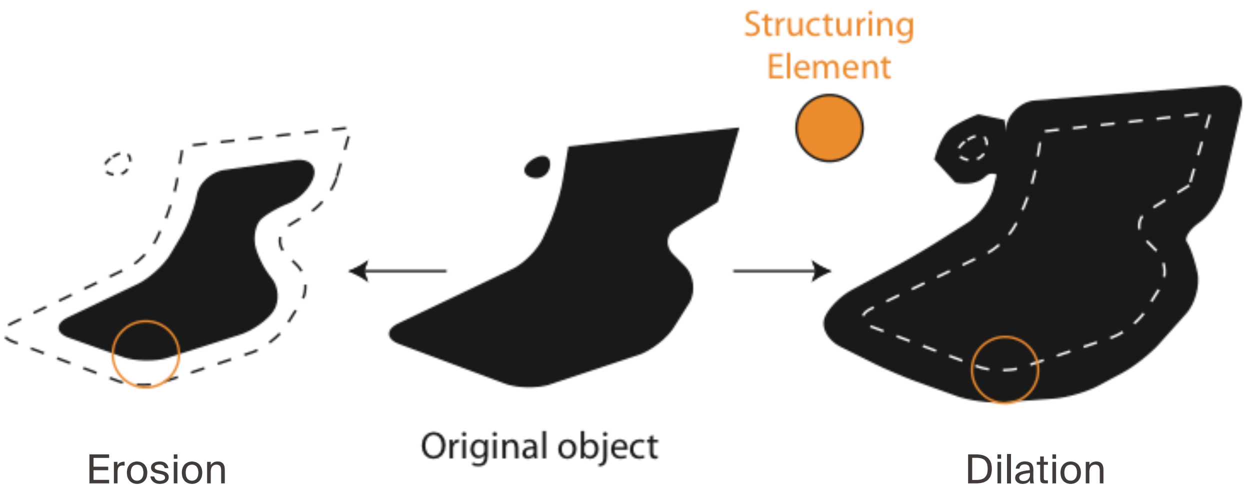
- It is the convolution window
- Contrarily to texture or convolution, the structuring element (the moving window) can take any shape
- Ex → (red = 1; white = 0)



EPFL

Morphology: operators

- Two base operators: erosion and dilation



Erosion and dilation

- Erosion is given by the intersection between the image patch and the structuring element

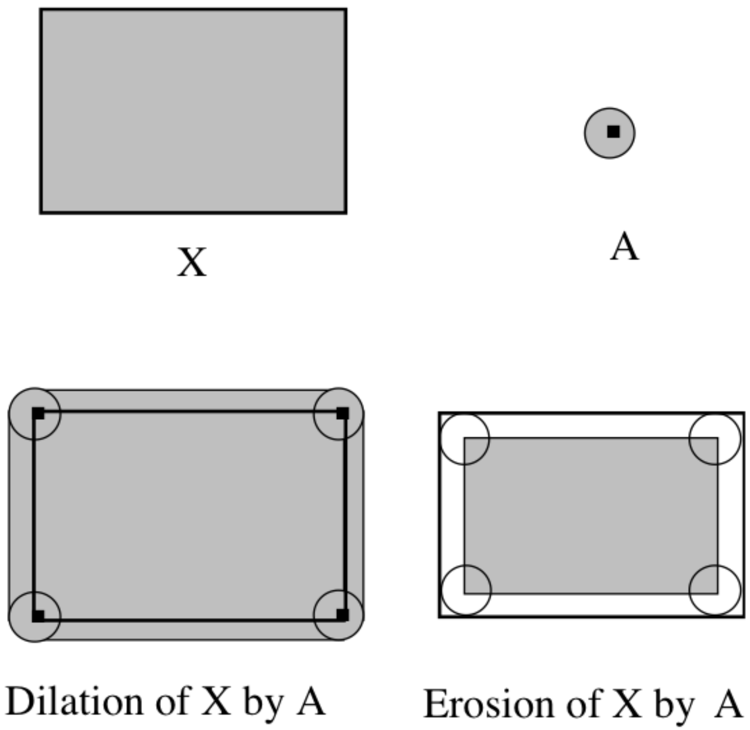
$$\varepsilon_B(X) = X \ominus B = \bigcap_{b \in B} X_{-b}$$

- Dilation is given by the union between the image patch and the structuring element

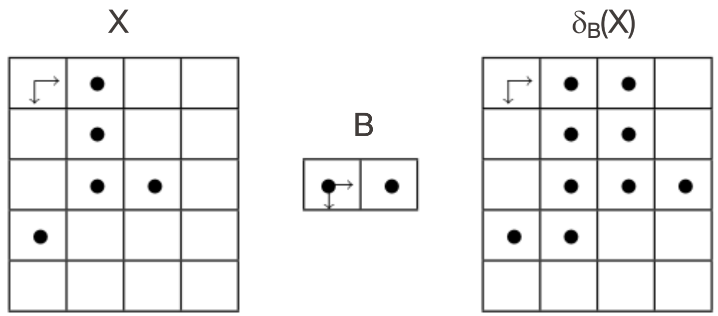
$$\delta_B(X) = X \oplus B = \bigcup_{b \in B} X_b$$

Source: Wilkinson course, Uni Groeningen 2010

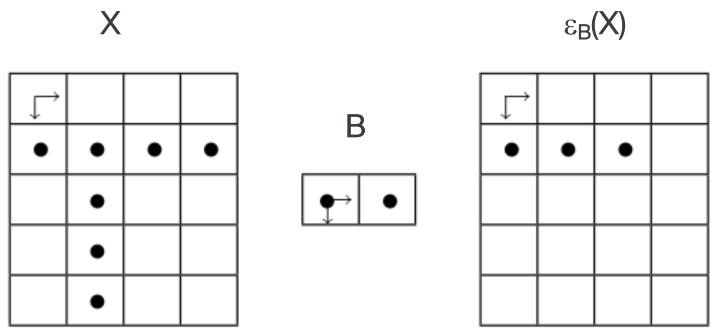
Erosion and dilation



Source: Wilkinson course, Uni Groeningen 2010



Dilation of X by a horizontal SE



Erosion of X by a horizontal SE

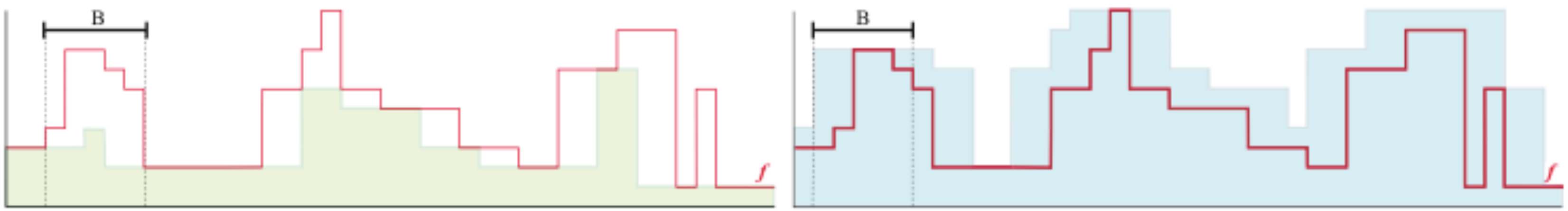
Morphology: operators for grayscale

- Two base operators: erosion and dilation

$$\varepsilon_{SE}(x) = \min_{i \in SE} x_i$$

$$\delta_{SE}(x) = \max_{i \in SE} x_i$$

- Erosion is the minimum in SE (dark elements)
- Dilation is the maximum in SE (bright elements)



(a) (b)

Morphology: operators

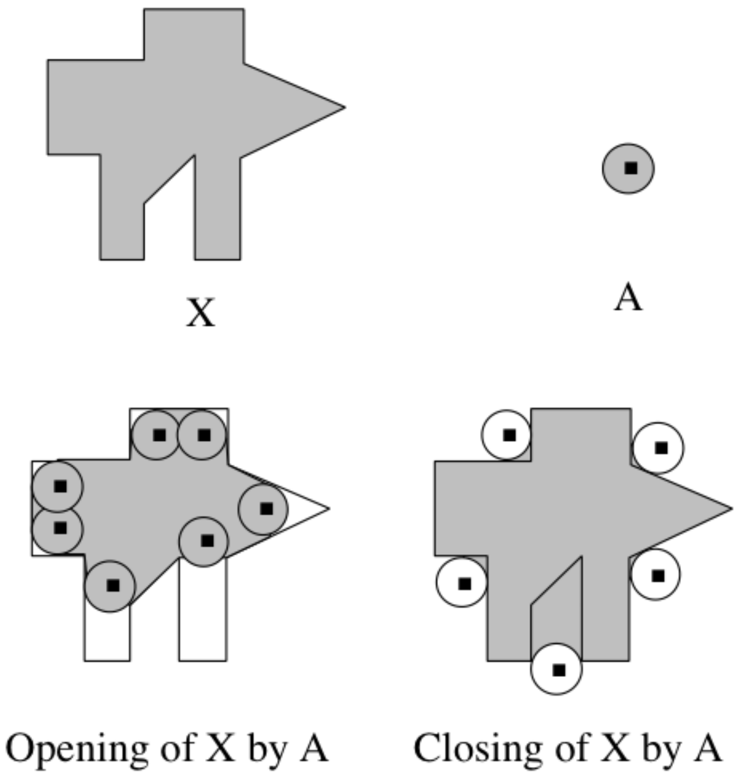
- Opening is an erosion followed by a dilation
  filters structures brighter than the surroundings

$$op_{SE}(I) = \delta_{SE}[\varepsilon_{SE}(I)]$$

- Closing is a dilation followed by an erosion
  filters structures darker than the surroundings

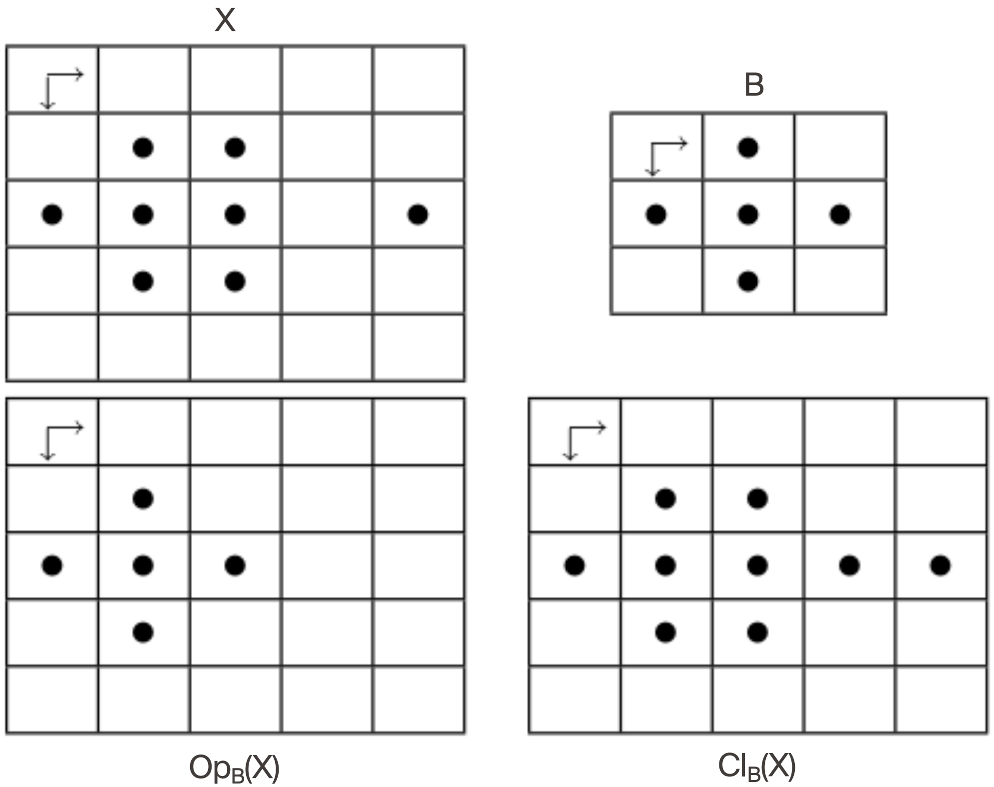
$$cl_{SE}(I) = \varepsilon_{SE}[\delta_{SE}(I)]$$

Structural opening and closing



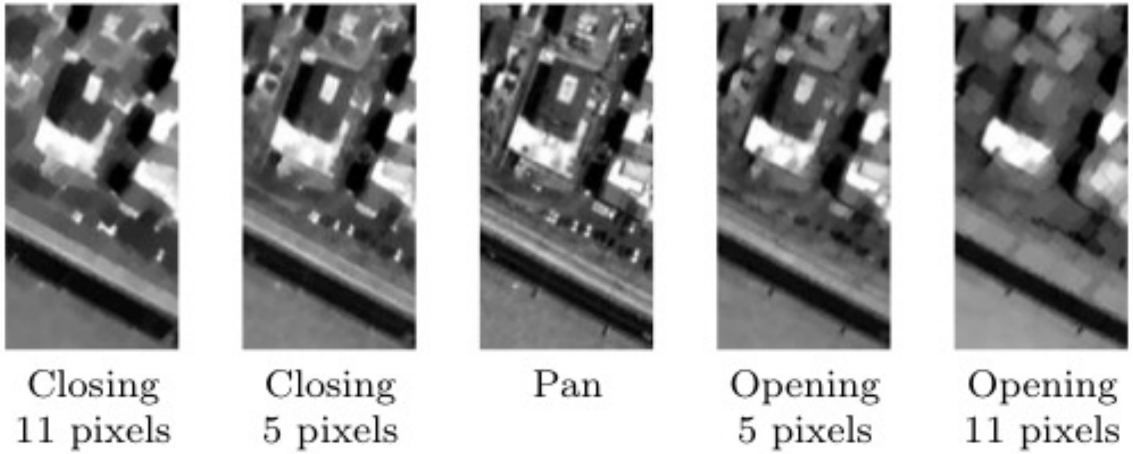
Source: Wilkinson course, Uni Groeningen 2010

Structural opening and closing

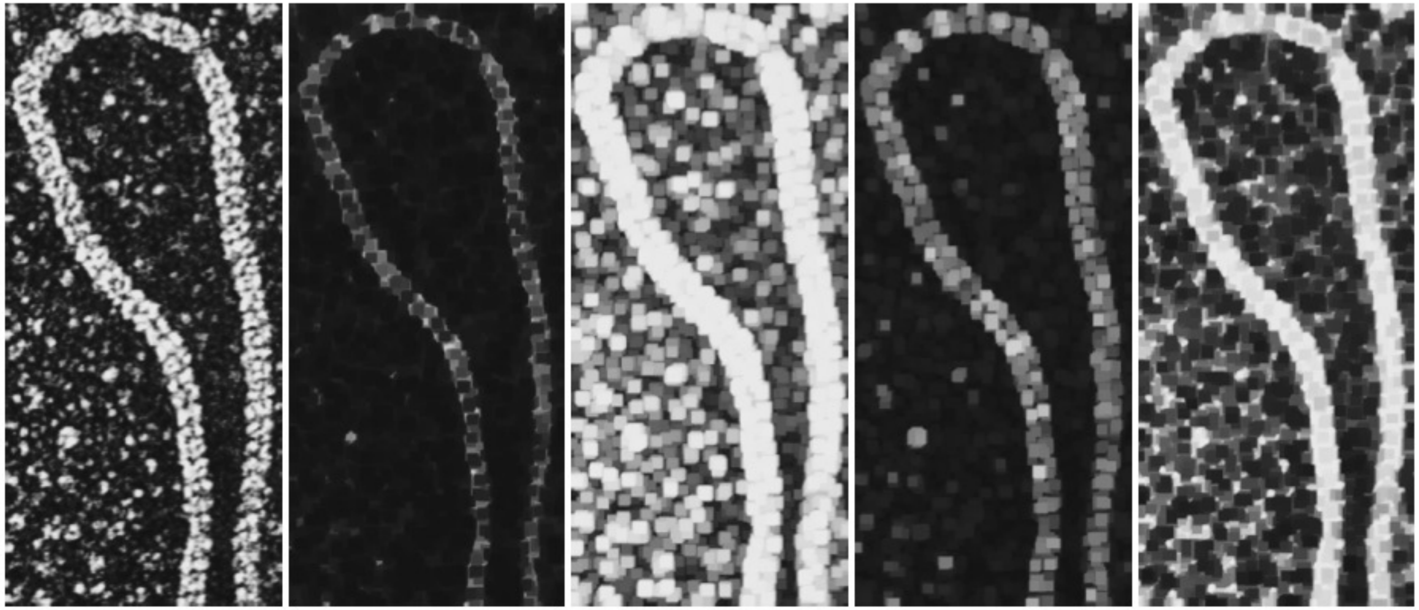


Source: Wilkinson course, Uni Groeningen 2010

Morphology: grayscale opening and closing



Morphology: grayscale opening and closing



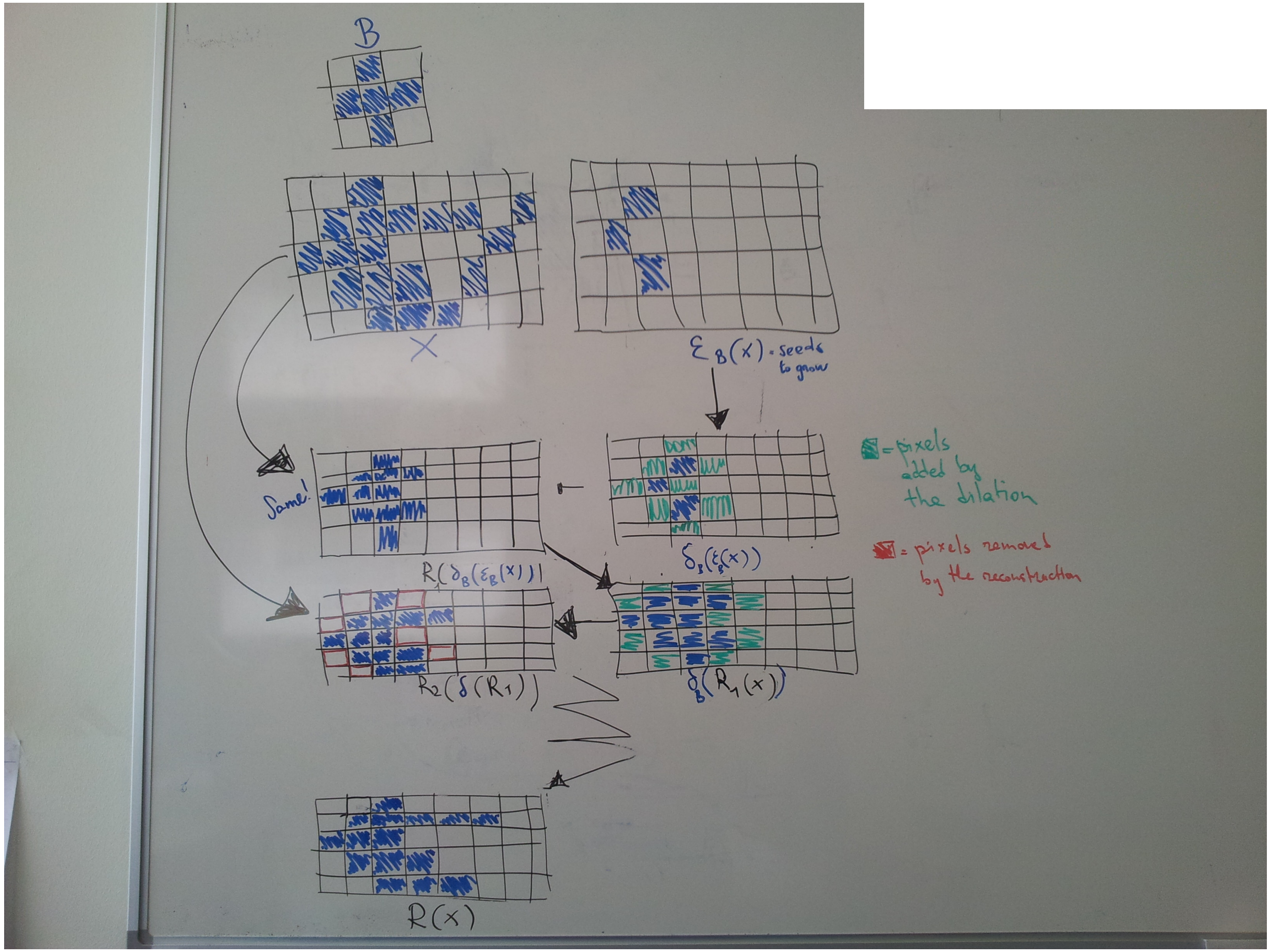
(a) f (b) $\varepsilon_9 f$ (c) $\delta_9 f$ (d) $\delta_9\varepsilon_9 f$ (e) $\varepsilon_9\delta_9 f$

Source: Wilkinson course, Uni Groeningen 2010

Morphology: operators

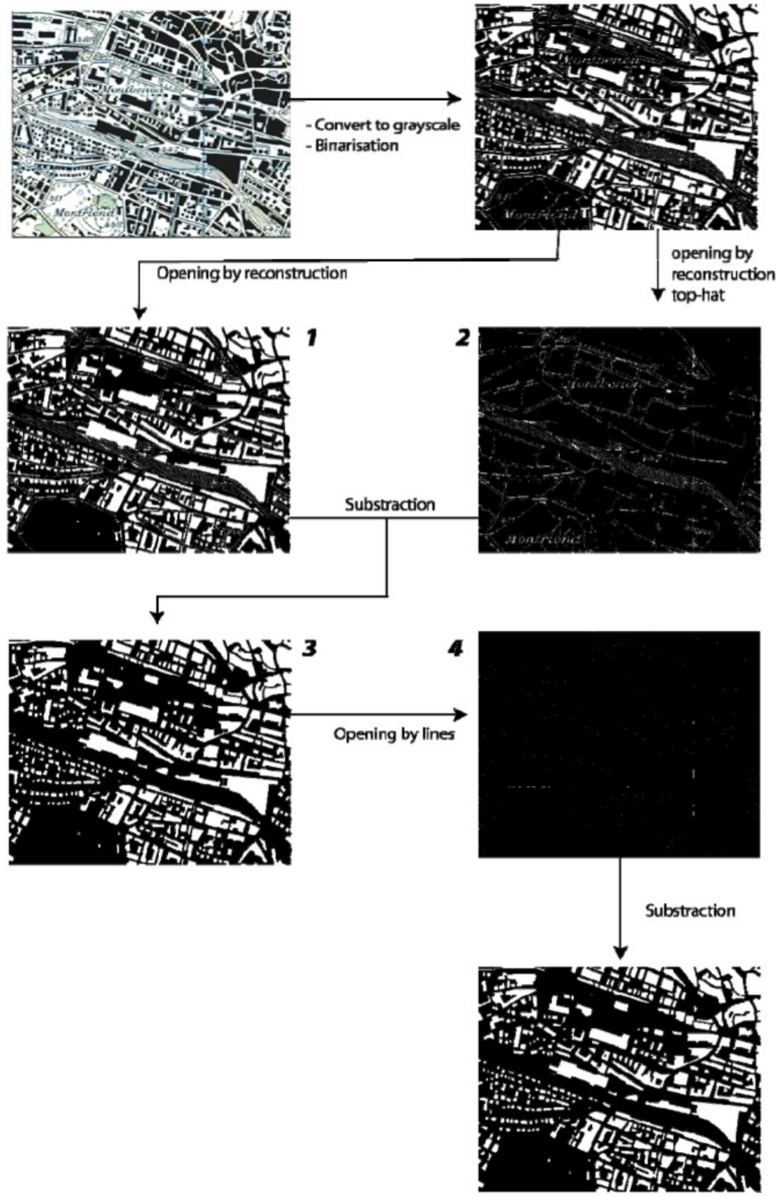
- Opening and closing do not respect border of objects
- Especially in VHR imagery, we want to keep this geometrical information
- We can use reconstruction filtering

- Iterative procedure
- (ex: opening by reconstruction) For each dilation
  - Take the minimum between dilation and original image
  - Continue until no changes are observed


B
X
$\varepsilon_B(x)$ = seeds to grow
Same!
$R_1(\delta_B(\varepsilon_B(x)))$
$\delta_B(\varepsilon_B(x))$
= pixels added by the dilation
= pixels removed by the reconstruction
$R_2(\delta(R_1))$
$\delta_B(R_1(x))$
R(x)

Reconstruction filters

- This may seem a bit artificial, but can be very useful in practical scenarios

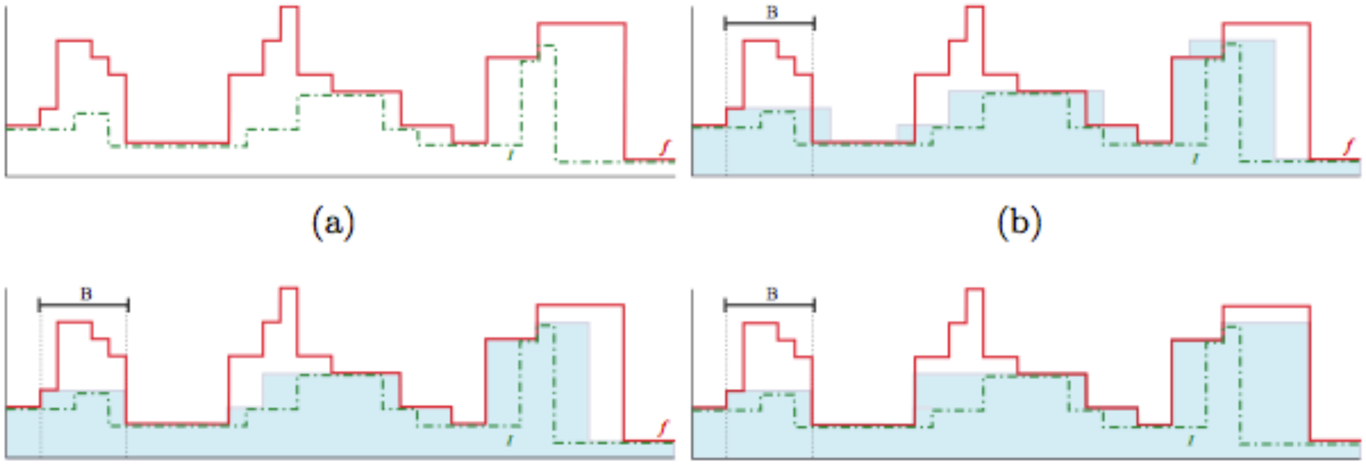


Source: Tuia and Kaiser, ECTQG 2008

Final result

Source: Tuia and Kaiser, ECTQG 2008

Reconstruction filters

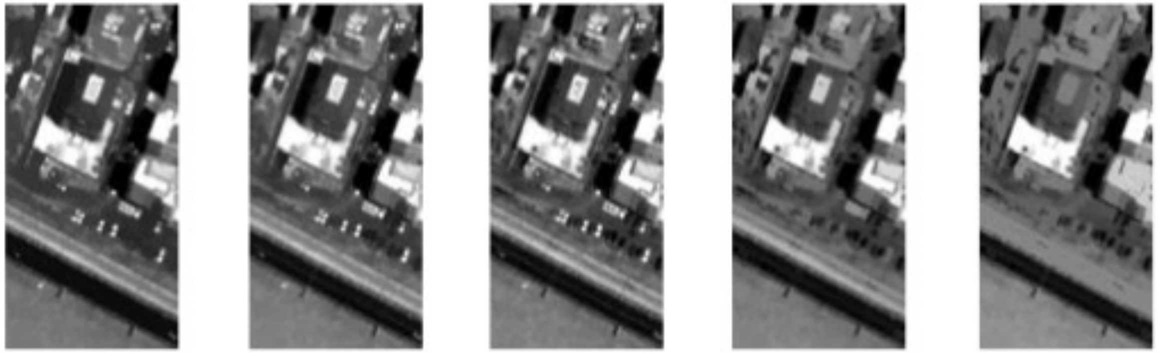


(a) (b) (c) (d)

Figure C.5: Opening by reconstruction: (a) line of a grayscale image f and erosion marker I, (b) dilation by a structuring element B, (c) first geodesic dilation by B and (d) second (and last) geodesic dilation by B.

Source: D. Tuia, PhD thesis

Filters by reconstruction



Clos. Rec. 11 pixels | Clos. Rec. 5 pixels | Pan | Open Rec. 5 pixels | Open Rec. 11 pixels

In summary

- Many features can be extracted from images

- From the DEM, all kind of topographic information

- If taking the DEM as an image, oyu can extract many informations about texture
  - features based on occurrence and co-occurrence
  - features bysed on mathematical morphology