Digital Image Processing

Non-linear image processing: mathematical morphology

Erchan Aptoula

Spring 2016-2017

Contents

- Lattice based mathematical morphology
- Grayscale operators
- Advanced morphological tools
- A glimpse of some other non-linear filters

Having met linear image processing, we now turn towards its non-linear counterpart, and specifically to an important representative: **mathematical morphology!**

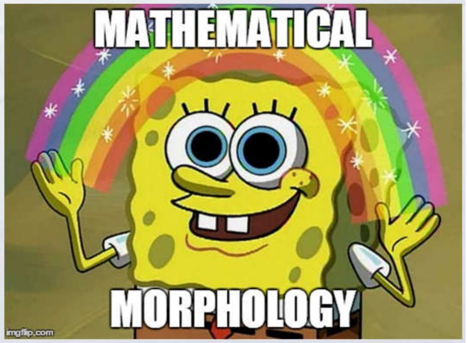
	Image Analysis Approaches	
Linear	**Linear**	**Statistical**
Non linear	**Morphological**	**Syntactic**

MM has tools for

- Enhancement
- Segmentation
- Edge detection
- Template matching
- Compression
- Shape analysis
- Texture analysis
- Space-time filtering and more

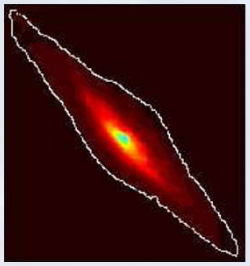
That can be used with

- Binary images
- Grayscale images
- Color images
- Vector fields
- Hyperspectral images
- Tensor images
- Video
- 3D images, and more.

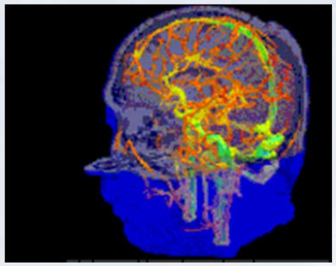


Non-linear image processing: mathematical morphology

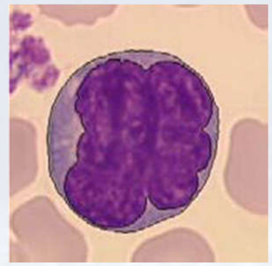
Leads to complete applications in



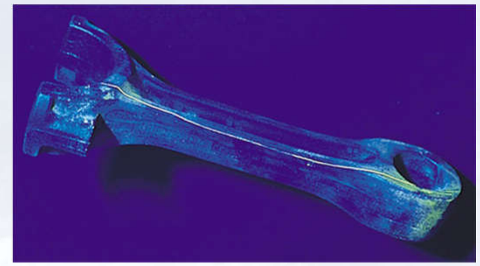
Astronomy



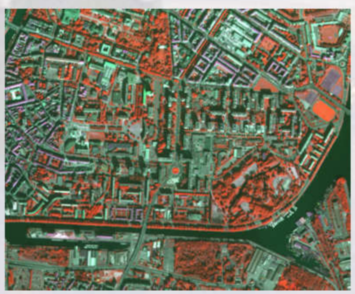
Medicine



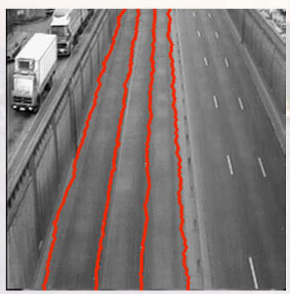
Microscopy



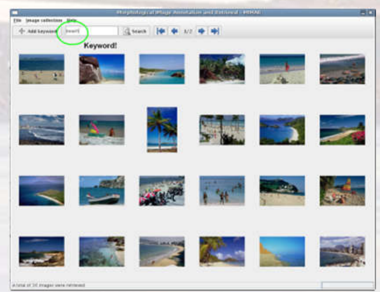
Industrial inspection



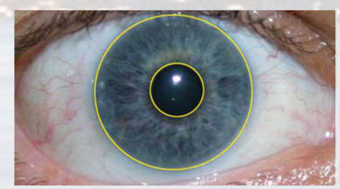
Remote sensing



Robot vision



Content based Retrieval



Biometry

and more

Why learn more about mathematical morphology?

To the non-morphological image processing community, MM is mostly known for its binary version as a post-processing option. **Wrong!**

Mathematical morphology

- is a complete image analysis image framework
- possesses a rigorous mathematical foundation
- offers unique tools for studying geometric structures
- constitutes a complementary approach to linear and statistical image analysis

The basic structure of…	
…Linear Image Processing..	**…Mathematical Morphology..**
is a **vector space**, i.e. a set of vectors V and a set of scalars K such that: 1. K is a field and V is a commutative group 2. There exists a multiplicative law between K and V	is a **complete lattice**, i.e. a non-empty set $\mathcal{L}$ such that: 1. $\mathcal{L}$ is governed by a partial ordering, i.e. a reflexive, transitive and anti-symmetric binary relation **2.** **Every non-empty subset** of $\mathcal{L}$ has a greatest lower bound (**infimum**) and a least upper bound (**supremum**)

Basic operations

Linear Image Processing	Mathematical Morphology
Since the basic laws of a vector space are **addition** and **scalar product**, the basic operations are those that distribute them: $\Psi\left(\sum \lambda_i f_i\right) = \sum \lambda_i \Psi(f_i)$ The resulting operator is **convolution**.	Since a complete lattice is based on a partial ordering relation (and its **sup** $\vee$ and **inf** $\wedge$) then the basic operations are those that distribute them: $\Psi(\vee f_i) = \vee \Psi(f_i)$: **dilation** $\Psi(\wedge f_i) = \wedge \Psi(f_i)$: **erosion**

An **ordering relation** has a central role in MM: a reflexive, transitive and anti-symmetric binary relation.

A binary relation $\mathcal{R}$ on a set $\mathcal{S}$ is called:

- **Reflexive**, if $x\,\mathcal{R}\,x, \forall\, x \in \mathcal{S}$
- **Anti-symmetric**, if $x\,\mathcal{R}\,y$ and $y\,\mathcal{R}\,x \Rightarrow x = y, \;\forall\, x, y \in \mathcal{S}$
- **Transitive**, if $x\,\mathcal{R}\,\mathrm{y}$ and $y\,\mathcal{R}\,\mathrm{w} \Rightarrow x\,\mathcal{R}\,\mathrm{w}, \;\forall\, x, y, w \in \mathcal{S}$
- **Total**, if $x\,\mathcal{R}\,\mathrm{y}$ or $y\,\mathcal{R}\,x, \;\forall\, x, y \in \mathcal{S}$, otherwise **partial**

But we saw binary mathematical morphology, there wasn't an order there…

… or was there ? ? ?

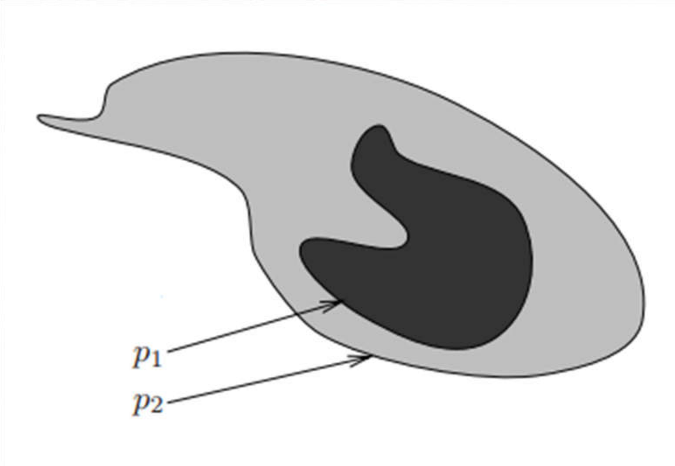
A digital image in this context is modeled as a function $f: \mathcal{E} \rightarrow \mathcal{T}$, where the domain or pixel space $\mathcal{E}$ is taken as the d-dimensional discrete space $\mathbb{Z}^d$ and $\mathcal{T}$ is a complete lattice. Moreover:

$$f, g: \mathcal{E} \rightarrow \mathcal{T}, \qquad f < g \;\Leftrightarrow\; \forall\, p \in \mathcal{E}, f(p) < g(p)$$

where “<” refers to the ordering relation of $\mathcal{T}$.

With binary 2D images: $f: \mathbb{Z}^2 \rightarrow \{0,1\}$ and the ordering relation is “⊂” inclusion, while the minimum and maximum are defined respectively through “∩” and “∪”. So **there IS** a complete lattice structure underlying binary mathematical morphology.

$p_1 \subset p_2 \Rightarrow p_1 < p_2$
$\vee\{p_1, p_2\} = \cup\,\{p_1, p_2\}$
$\wedge\{p_1, p_2\} = \cap\,\{p_1, p_2\}$



Is it a partial ordering or a total ordering?

Non-linear image processing: mathematical morphology

Now, let’s check 2D discrete grayscale images, $f: \mathbb{Z}^2 \rightarrow \mathbb{Z}$

$$\forall\, p, q \in \mathbb{Z},\ \ p \leq q \text{ or } p \geq q$$

$$f \leq g \ \Leftrightarrow\ \forall\, p, f(p) \leq g(p)$$
$$(max\{f, g\})(x) = max\{f(x), g(x)\}$$
$$(min\{f, g\})(x) = min\{f(x), g(x)\}$$

And that is sufficient for using mathematical morphology with any discrete grayscale image.

Is the scalar ordering relation partial or total?

And how do you take the complement of a grayscale image?
If $f: \mathbb{Z}^2 \rightarrow \{0,1,2, \dots, L-1\}$ then the complement is $f^c(p) = L - 1 - f(p)$

And how do we process grayscale images?

Once again with **structuring elements** (SE) that are represented this time by functions.

If zero valued: $H: \mathbb{Z}^2 \to 0, (H \subset \mathbb{Z}^2)$ they are called **flat SEs.**
Or they can also be **non-flat SEs**, $h: \mathbb{Z}^2 \to \mathbb{Z}$.
Both contain an arbitrary origin point, usually their center.

We will limit our scope to flat SEs.
Reflection (or transposition) of a SE: $\breve{H} = \{(-s, -t) | (s, t) \in H\}$

The filters are applied in the usual way, by hovering the SE over the image domain and calculating the interaction of the SE with the underlying image values.

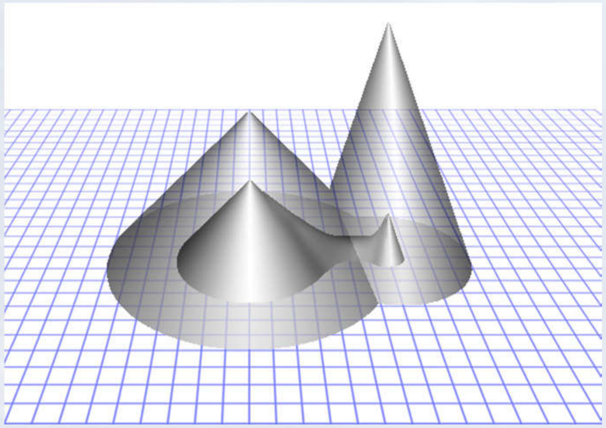
Grayscale dilation with a flat SE: $$\delta_H(f)(x,y) = \max_{(s,t)\,\in\, H}\{f(x-s,y-t)\}$$

Grayscale erosion with a flat SE: $$\varepsilon_H(f)(x,y) = \min_{(s,t)\,\in H}\{f(x-s,y-t)\}$$

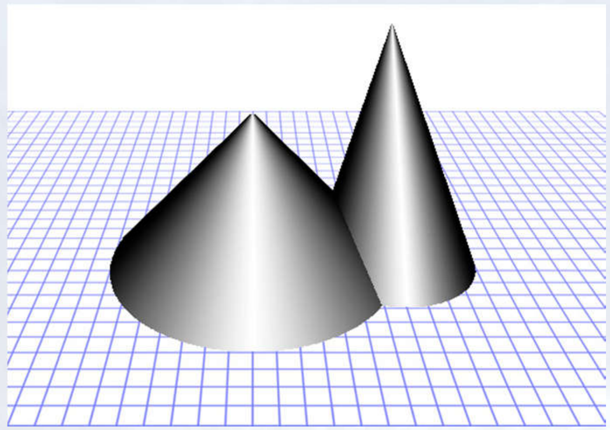
Parallelism with linear image processing	
sum	max or min
product	sum
convolution	erosion and dilation
$\sum_{(s,t)} f(x-s,y-t)h(s,t)$	$\max_{(s,t)}\{f(x-s,y-t)+h(s,t)\}$

We are still dealing with convolutions, just with different laws.

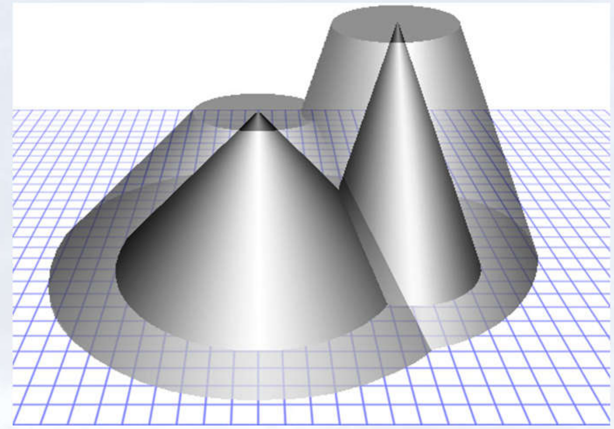
Non-linear image processing: mathematical morphology



Erosion



Original



Dilation

Properties

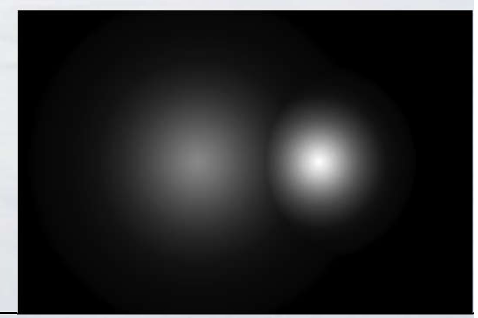
Duality: $\delta(f) = [\varepsilon(f^c)]^c$ and $\varepsilon(f) = [\delta(f^c)]^c$

Anti-extensivity $\varepsilon(f) \leq f \leq \delta(f)$ and extensivity

Increasingness: $f \leq g \Rightarrow \delta(f) \leq \delta(g)$

$f \leq g \Rightarrow \varepsilon(f) \leq \varepsilon(g)$

Adjunction: $f \leq \varepsilon(g) \Leftrightarrow \delta(f) \leq g$



Dilation enlarges structures brighter than their surrounding (i.e shrinks darker ones).

Erosion enlarges structures darker than their surrounding (i.e. shrinks brighter ones).

Erosion

Original

Dilation

It’s all about order, and relative intensity!

Implementation of dilations and erosions

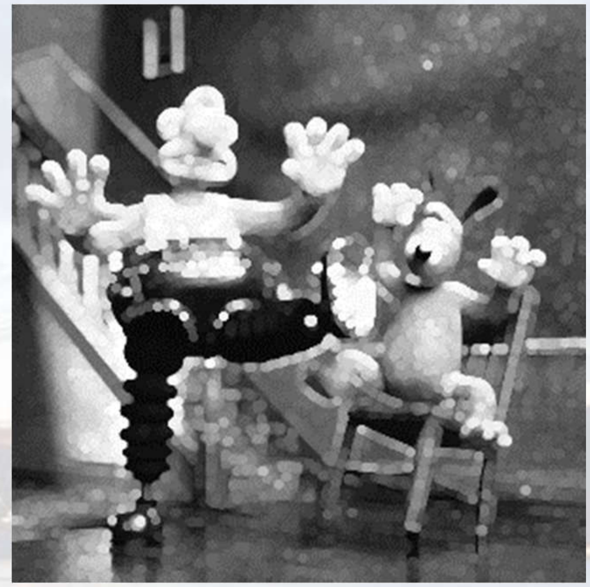
A direct implementation with a SE of size N, will require $N-1$ min/max comparisons **per pixel** of the input image. Fortunately there are many ways to accelerate them.

- Decompose SEs when possible; e.g. square into horizontal & vertical lines.
- At every shift of your SE, you can keep track of which pixels leave and which enter the field of computation and update the min and max regardless of SE size.
- Binary dilations and erosions can be accelerated drastically thanks to bit-wise logical operators.

Non-linear image processing: mathematical morphology

Erosion

Original



Dilation

Erosion and dilation constitute the elementary blocks of more advanced operators.

Grayscale morphological gradient

$g_i(f) = f - \varepsilon(f)$

$g(f) = \delta(f) - \varepsilon(f)$

$g_e(f) = \delta(f) - f$

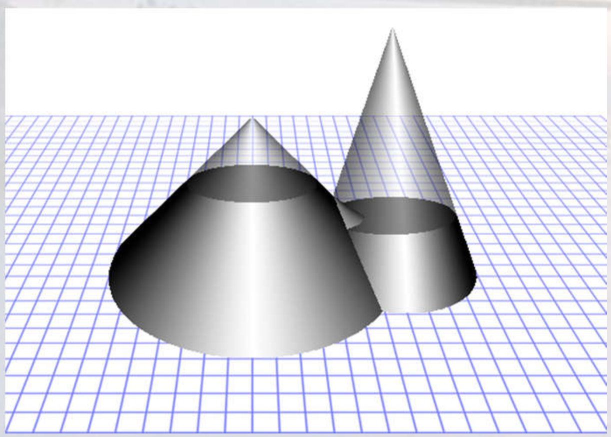
Non-linear image processing: mathematical morphology

Opening and closing

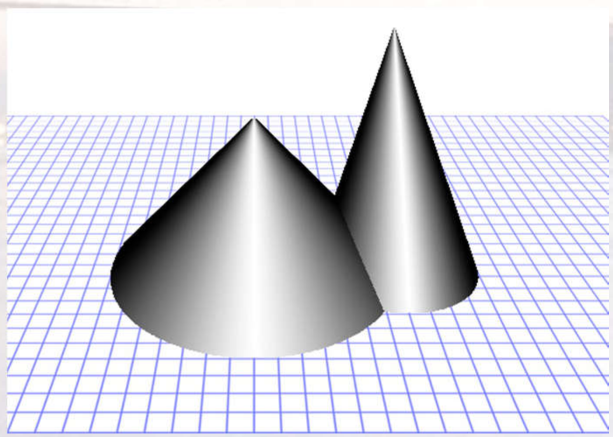
$$\gamma_B(f) = \delta_{\breve{B}}(\varepsilon_B(f))$$
$$\varphi_B(f) = \varepsilon_{\breve{B}}(\delta_B(f))$$

Opening removes from the image regions that are brighter than their surrounding and that fit into the SE; cuts off peaks.

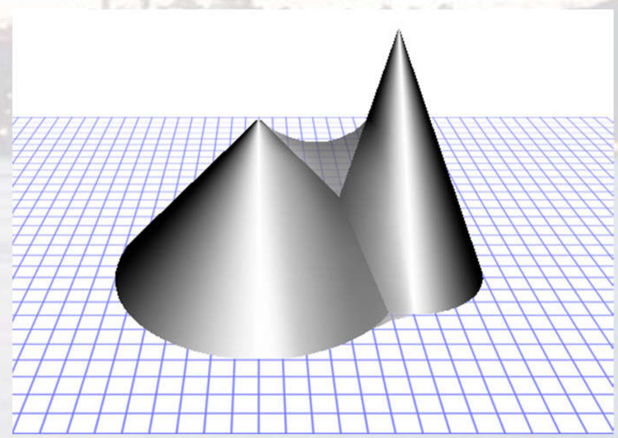
Closing fills in the image regions that are darker than their surrounding and that fit into the SE; fills lakes.



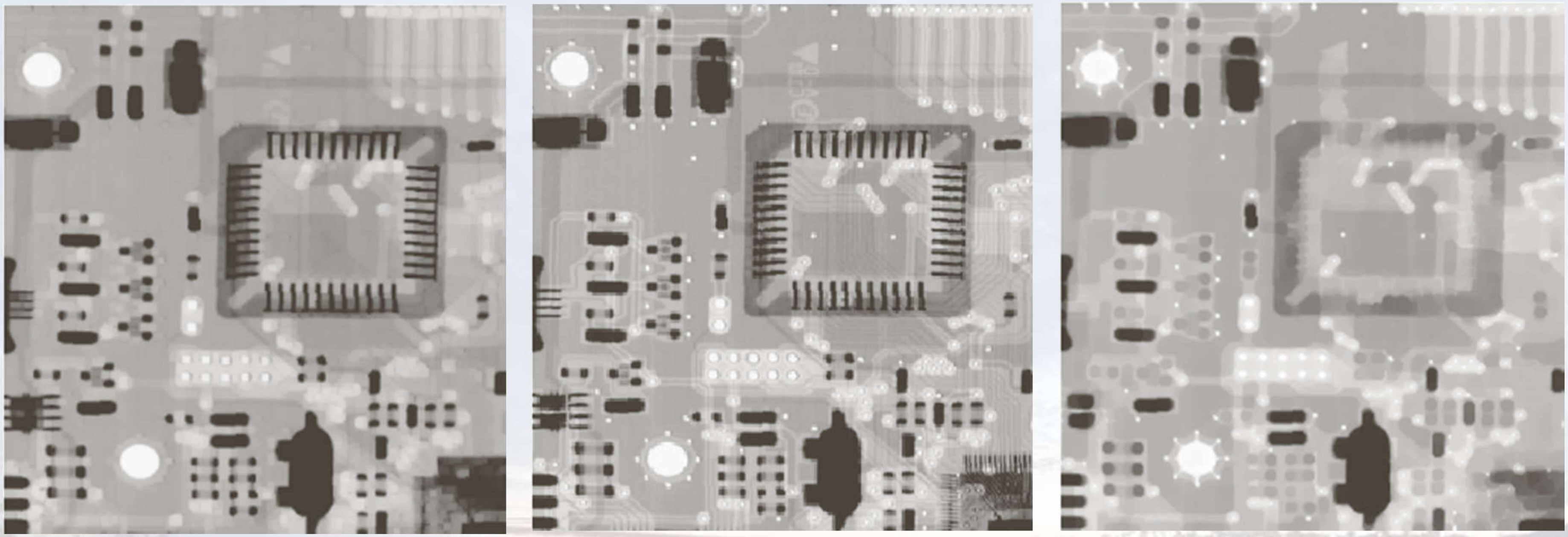
Opening



Original



Closing



opening original closing

Properties

Anti-extensivity: $\varepsilon(f) \leq \gamma(f) \leq f$

Extensivity: $f \leq \varphi(f) \leq \delta(f)$

Idempotence: $\gamma(f) = \gamma(\gamma(f))$ and $\varphi(f) = \varphi(\varphi(f))$

Increasingness: $f \leq g \Rightarrow \gamma(f) \leq \gamma(g)$

$f \leq g \Rightarrow \varphi(f) \leq \varphi(g)$

Non-linear image processing: mathematical morphology

A **linear filter** is (usually) a linear and shift invariant operator.

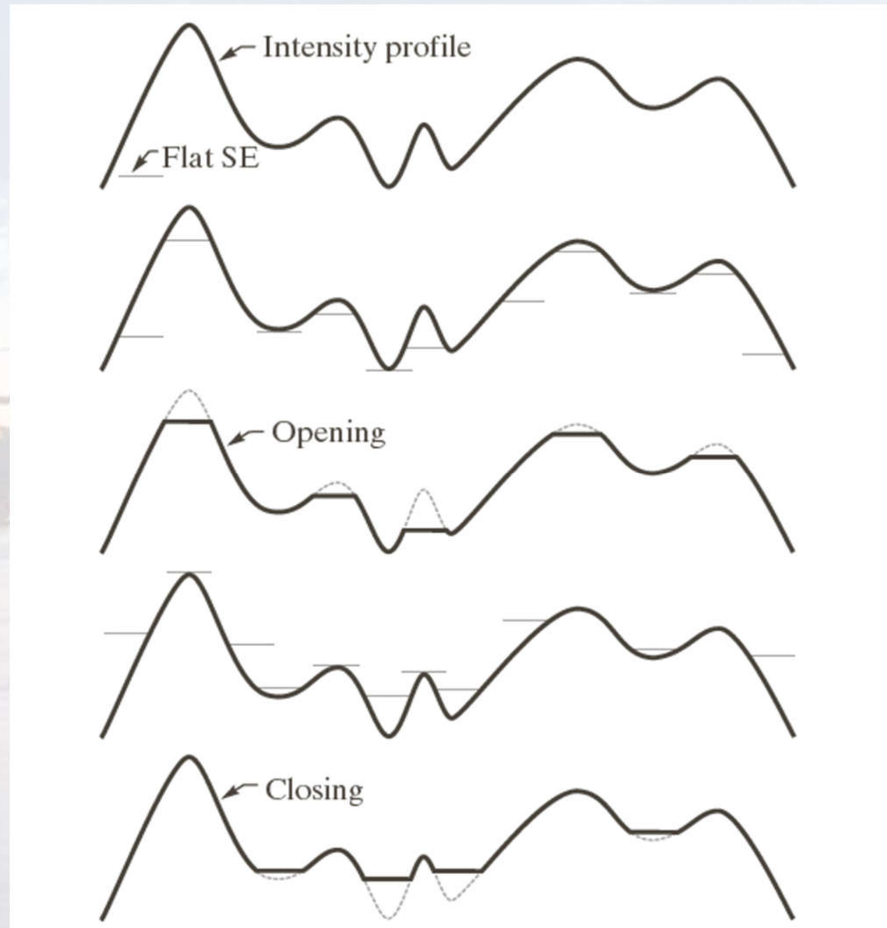
A **morphological filter** is an idempotent and increasing transformation of a complete lattice into itself. Opening and closing are such filters.

$$f: \mathcal{E} \rightarrow \mathcal{T}$$
$$\Psi: \mathcal{T}^{\mathcal{E}} \rightarrow \mathcal{T}^{\mathcal{E}}$$

$$\Psi(f) = \Psi(\Psi(f))$$
$$f \leq g \Rightarrow \Psi(f) \leq \Psi(g)$$

a
b
c
d
e

FIGURE 9.36 Opening and closing in one dimension. (a) Original 1-D signal. (b) Flat structuring element pushed up underneath the signal. (c) Opening. (d) Flat structuring element pushed down along the top of the signal. (e) Closing.



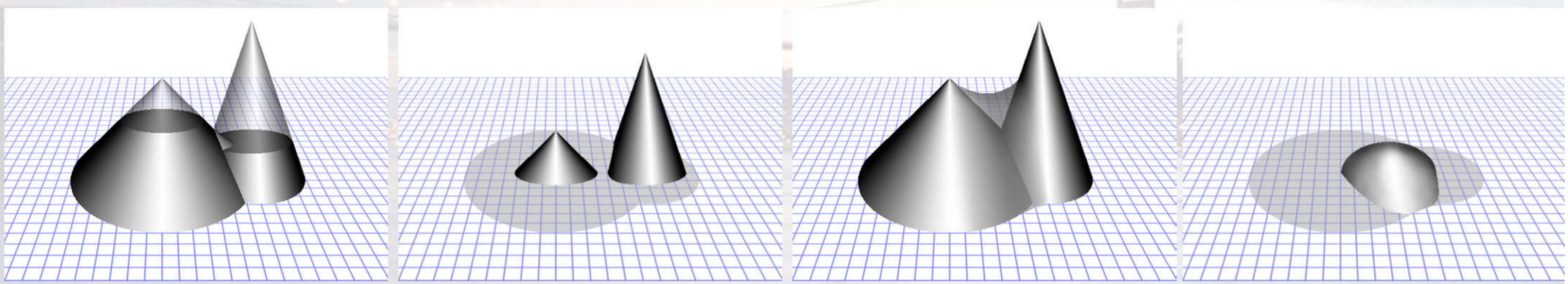
Non-linear image processing: mathematical morphology

TopHat and BotHat

$$TH(f) = f - \gamma(f)$$
$$BH(f) = \varphi(f) - f$$

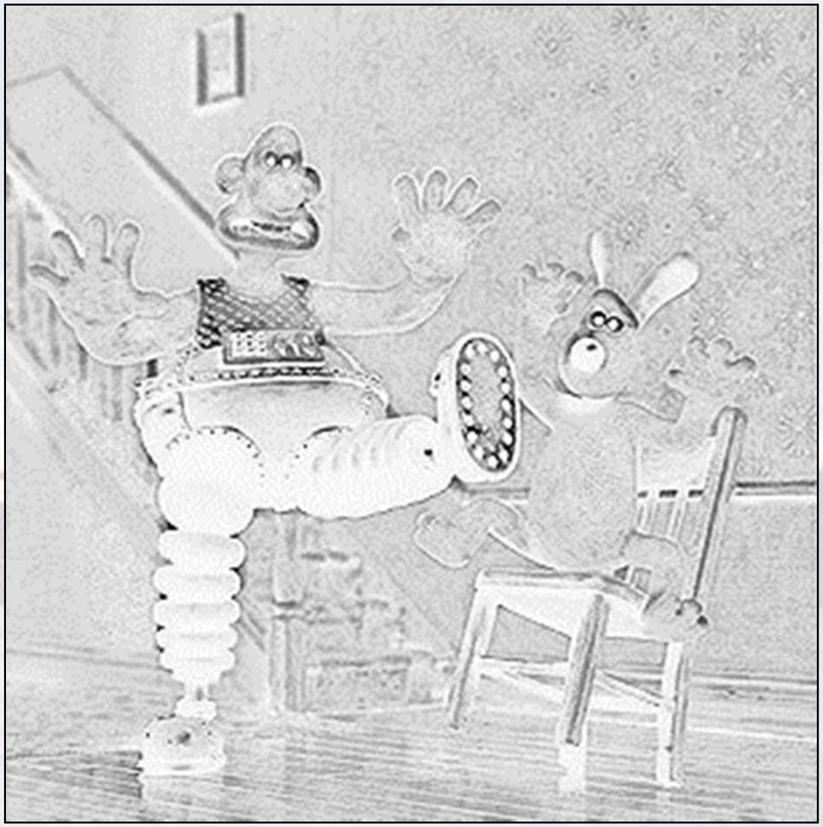
TopHat outputs the regions of the image removed by an opening.

BotHat outputs the regions of the image filled by a closing.



Opening TopHat Closing BotHat

TopHat



The output is inverted

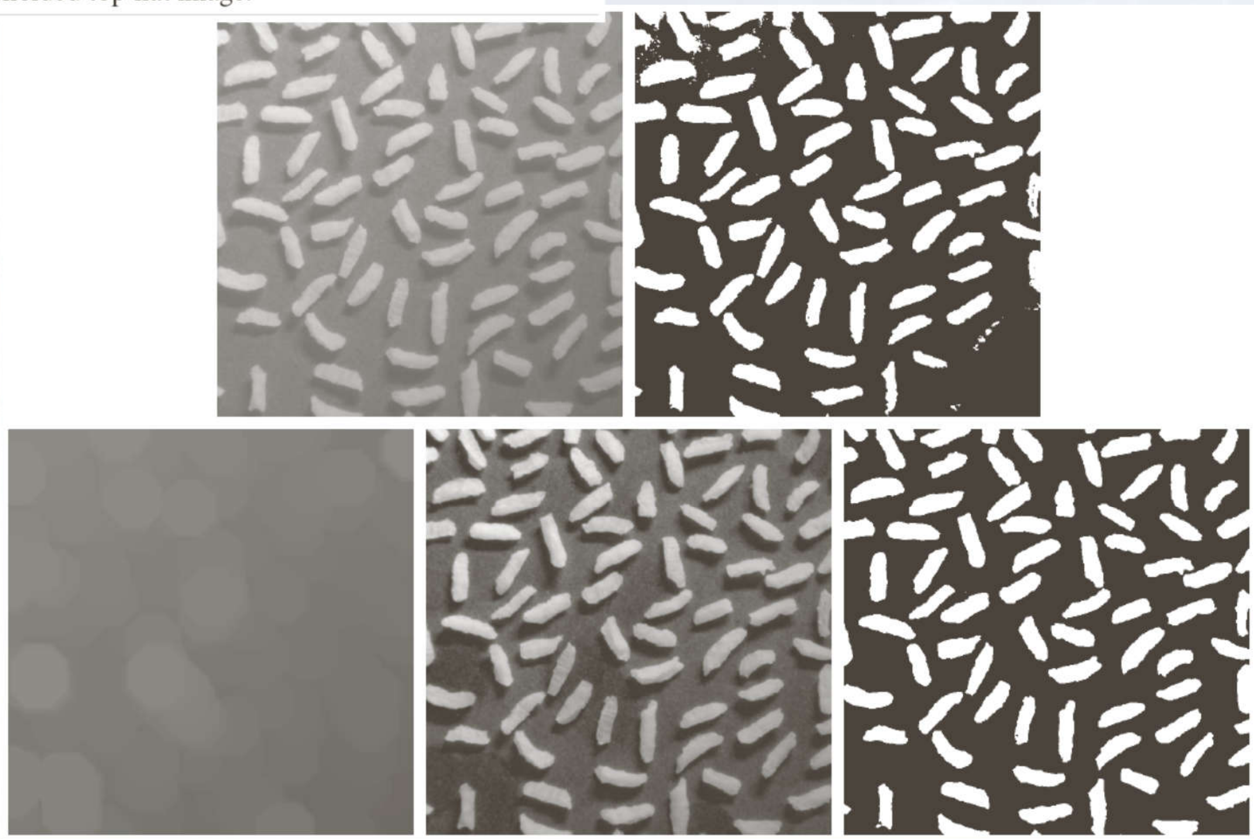
BotHat

The output is inverted

Shading correction with TopHat (it removes regions independently of their absolute intensity)

a b
c d e

FIGURE 9.40 Using the top-hat transformation for *shading correction*. (a) Original image of size 600 × 600 pixels. (b) Thresholded image. (c) Image opened using a disk SE of radius 40. (d) Top-hat transformation (the image minus its opening). (e) Thresholded top-hat image.

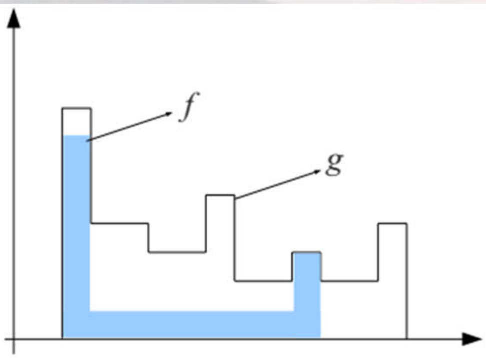


Geodesic operators are a class of morphological operators. They admit **two images** f and g as input. The transformation is applied on the **f**, while forcing it to remain below or above the **marker g**.

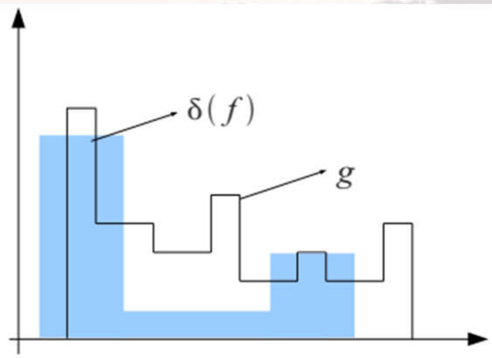
Geodesic dilation: $\delta^{(1)}(f,g) = \delta(f) \wedge g, f \leq g$

Geodesic erosion: $\varepsilon^{(1)}(f,g) = \varepsilon(f) \vee g, g \leq f$

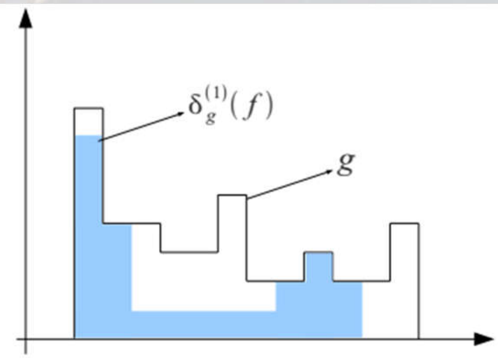
They can be repeated: $\delta^{(n)}(f,g) = \delta^{(1)}(\delta^{(n-1)}(f,g))$



Original 1D input

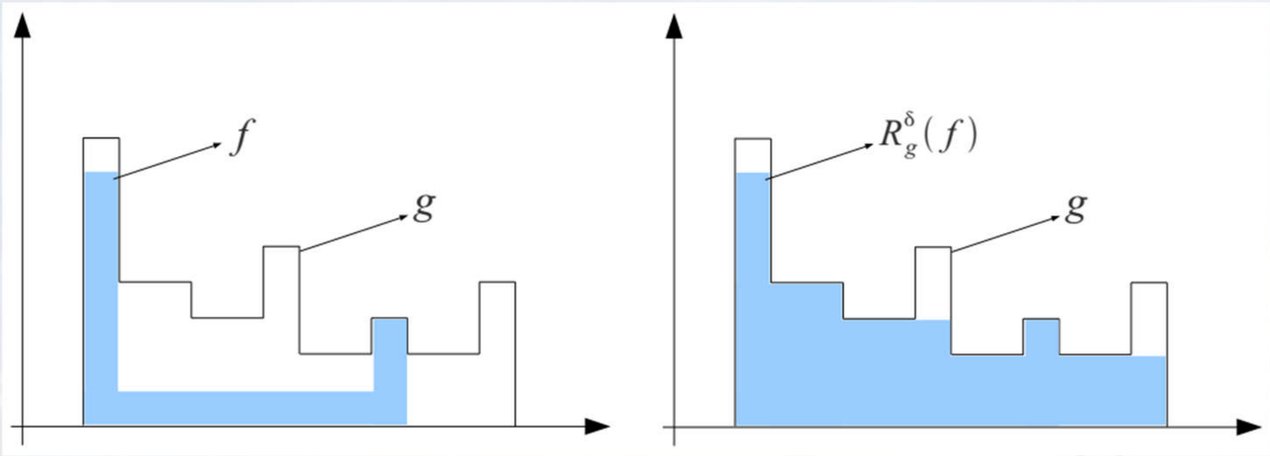


Dilation



Geodesic dilation

Geodesic tools lead to reconstruction based operators when applied **until stability**.



Reconstruction by dilation

$R^{\delta}(f,g) = \delta^{(i)}(f,g), i$ such that $\delta^{(i)}(f,g) = \delta^{(i+1)}(f,g)$

Reconstruction by erosion

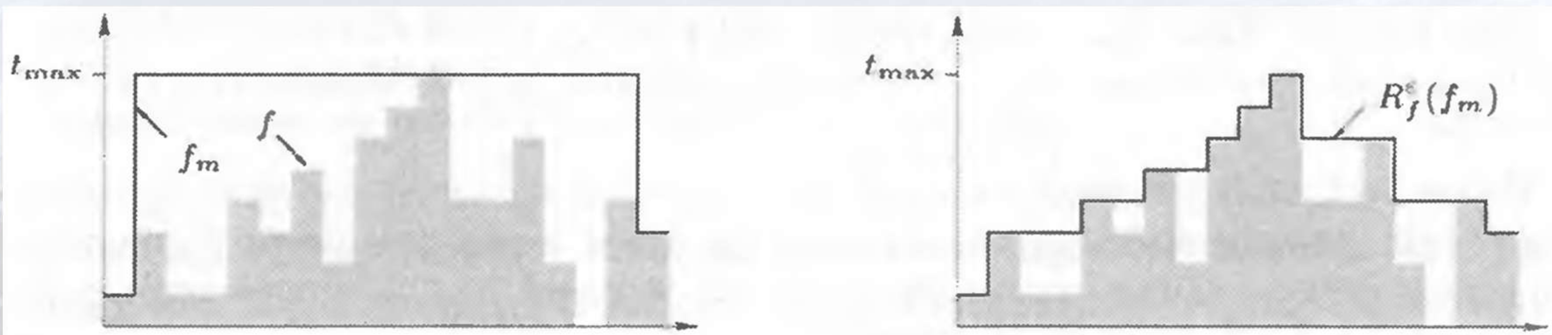
$R^{\varepsilon}(f,g) = \varepsilon^{(i)}(f,g), i$ such that $\varepsilon^{(i)}(f,g) = \varepsilon^{(i+1)}(f,g)$

Useful for connected component extraction, regional extrema extraction, etc.

Filling holes

$$FILL(f) = R^{\varepsilon}(f', f)$$

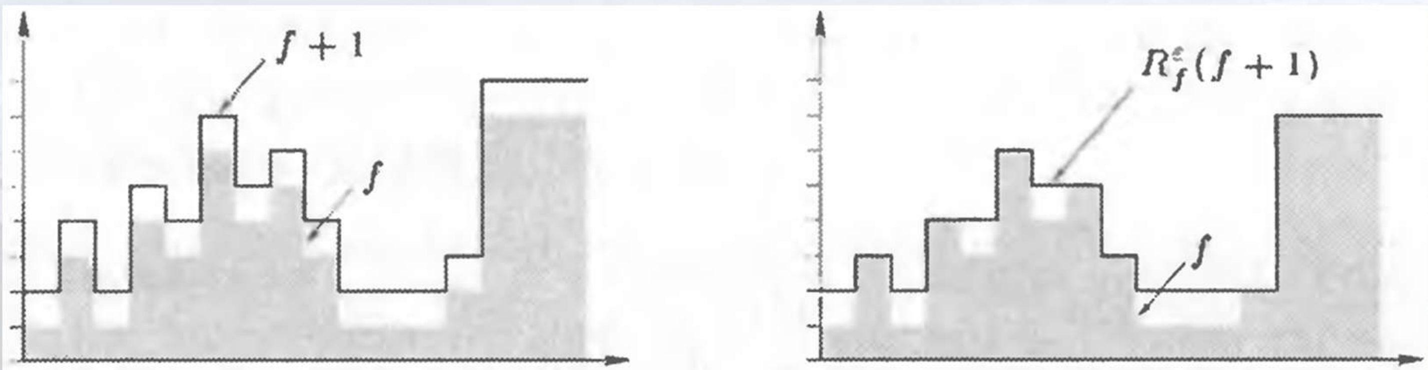
$$f'(p) = \begin{cases} f(p), & \text{if lies on the border of the definition domain of } f \\ t_{max}, & \text{otherwise} \end{cases}$$



Adapted from P. Soille

A **regional minimum** of an image f at elevation t is a connected component with the value t whose external boundary pixels have a value strictly greater than t

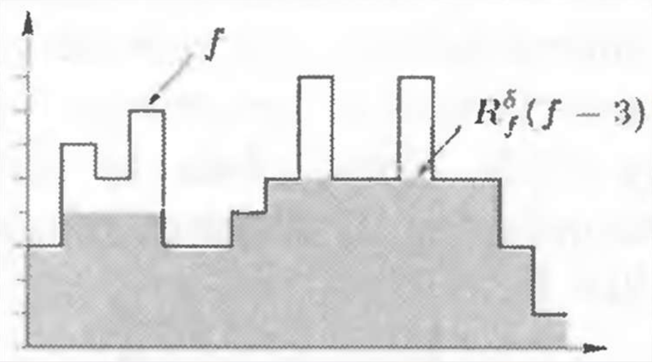
$$RMIN(f) = R^{\varepsilon}(f + 1, f) - f$$
$$RMAX(f) = f + 1 - R^{\delta}(f, f + 1)$$



Adapted from P. Soille

The **h-maxima** transformation suppresses all maxima whose depth is lower or equal to a predefined threshold h

$$HMAX_h(f) = R^{\delta}(f - h, f)$$
$$HMIN_h(f) = R^{\varepsilon}(f + h, f)$$



Adapted from P. Soille

Opening by reconstruction

$$\gamma_R(f) = R^{\delta}(\varepsilon(f), f)$$

Closing by reconstruction

$$\varphi_R(f) = R^{\varepsilon}(\delta(f), f)$$

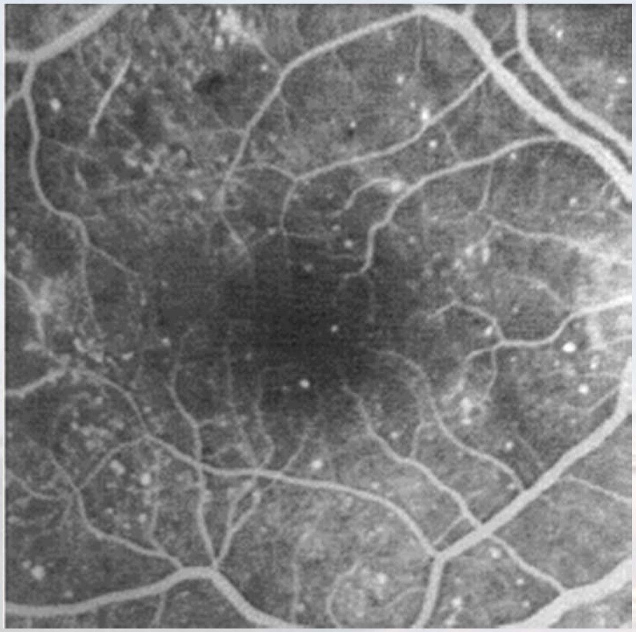
Original

Opening of size 11

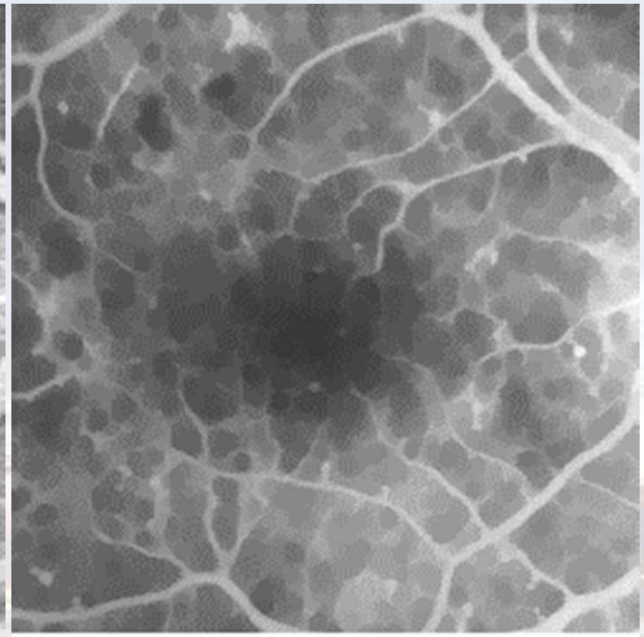
Opening by rec. of size 11

They remove/fill regions smaller than the SE, but the rest is preserved **perfectly**.

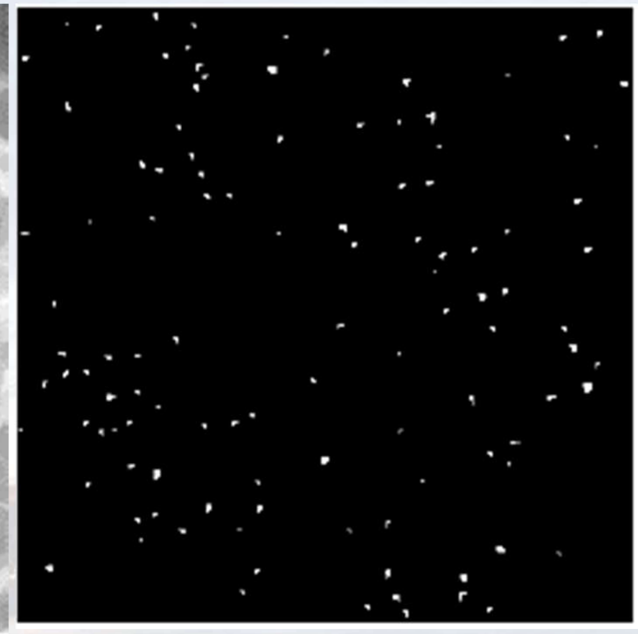
Retina examination: extraction of aneurysms (small isolated peaks)



Original

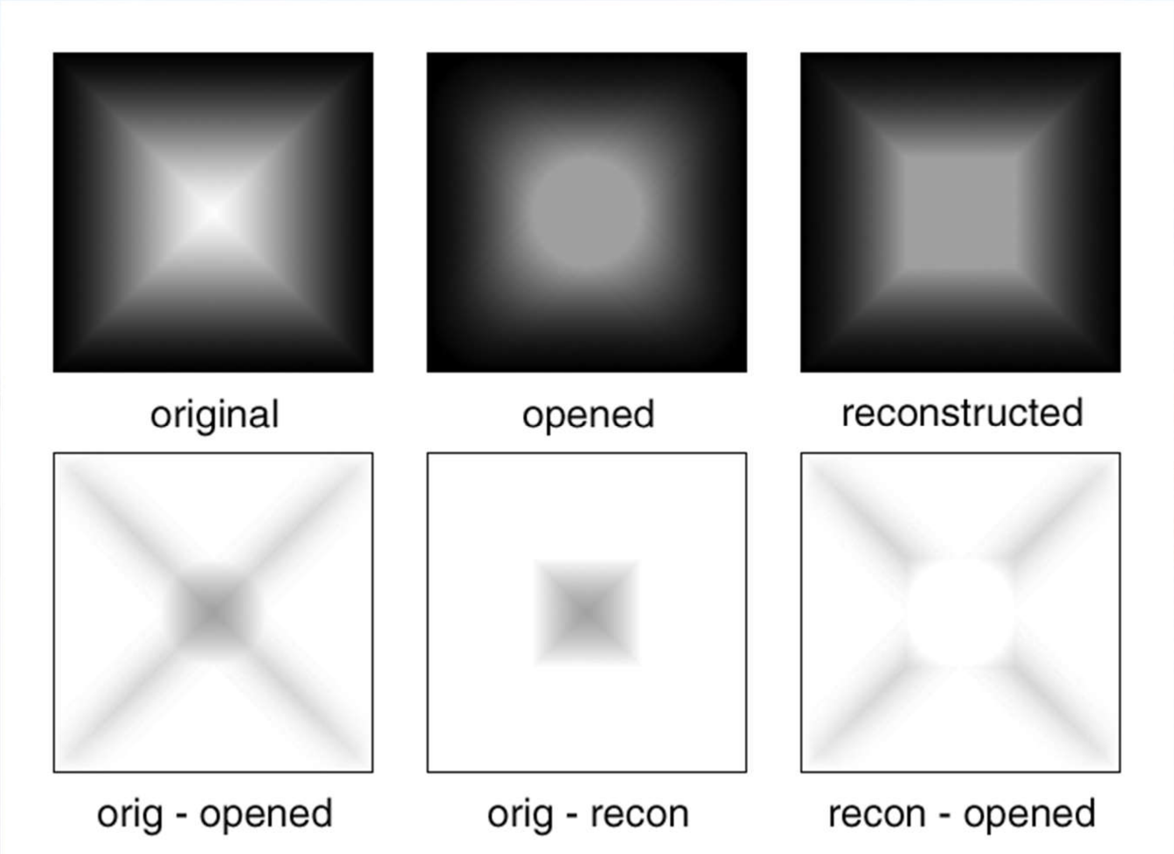


Opening by reco.



Their difference

Grayscale reconstruction



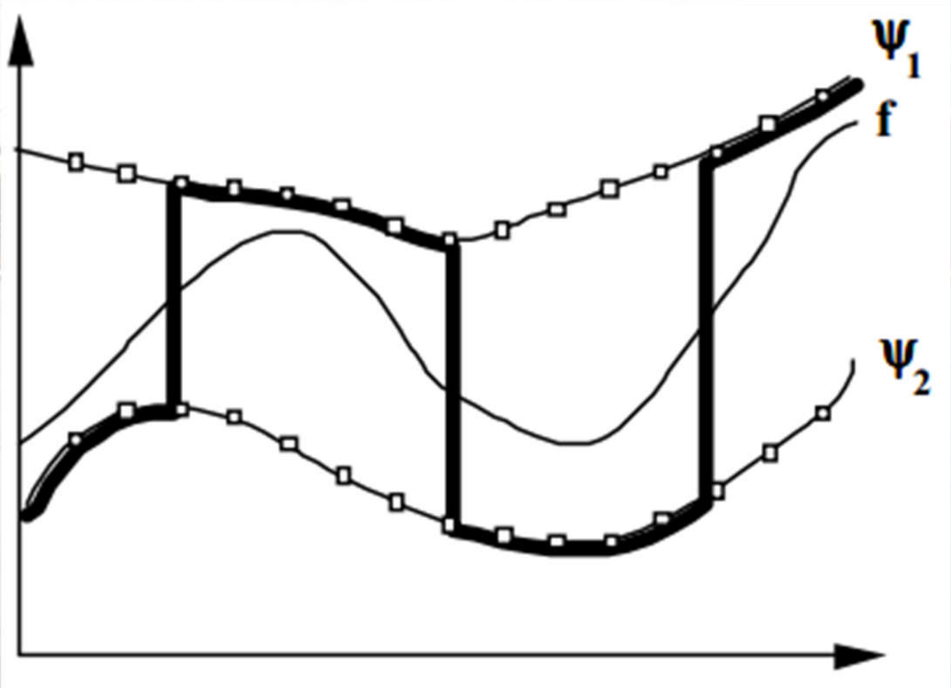
Grayscale opening by reconstruction: smoothing of unwanted bright features

Toggle mapping: contrast enhancement, sharpening.

Given an anti-extensive transformation Ψ_1 (e.g. an erosion), and an extensive transformation Ψ_2 (e.g. a dilation) of an image f, then the toggle map is defined as:

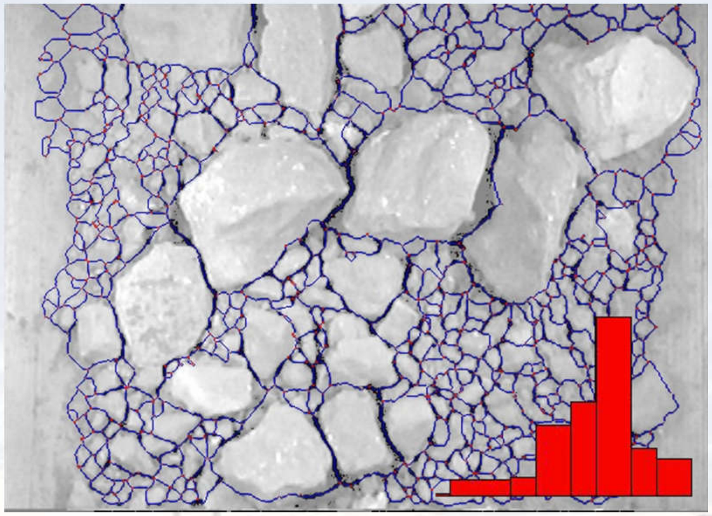
$$f_{TM} = \begin{cases} \Psi_1(f), \text{if } f - \Psi_1(f) < \Psi_2(f) - f \\ \Psi_2(f), \text{otherwise} \end{cases}$$

In other words, the output for each pixel is the closest to the original of the two transformations.



Toggle mapping example

Granulometry is about determining the size of the distribution of particles in an image.



It is very helpful for describing image content.

But normally you'd need to detect the particles first.

In reality, you just sieve the material through progressively larger sieves.

and record the amount of material passing through. Can we do that digitally??

Armed with MM, yes we can! Axioms:

- A sieve of size λ should remove image volume, i.e. should be anti-extensive:

$$\Psi_\lambda(f) \leq f$$

- If another image greater, then its sieve output should be greater as well, for the same sieve; i.e. increasingness:

$$f \leq g \Rightarrow \Psi_\lambda(f) \leq \Psi_\lambda(g)$$

- If we increase the sieve size, it should remove more:

$$\lambda \geq \mu \Rightarrow \Psi_\lambda(f) \leq \Psi_\mu(f)$$

- The effect of the largest sieve supersedes all smaller ones:

$$\Psi_\lambda\left(\Psi_\mu(f)\right) = \Psi_\mu(\Psi_\lambda(f)) = \Psi_{\max\{\lambda,\mu\}}(f)$$

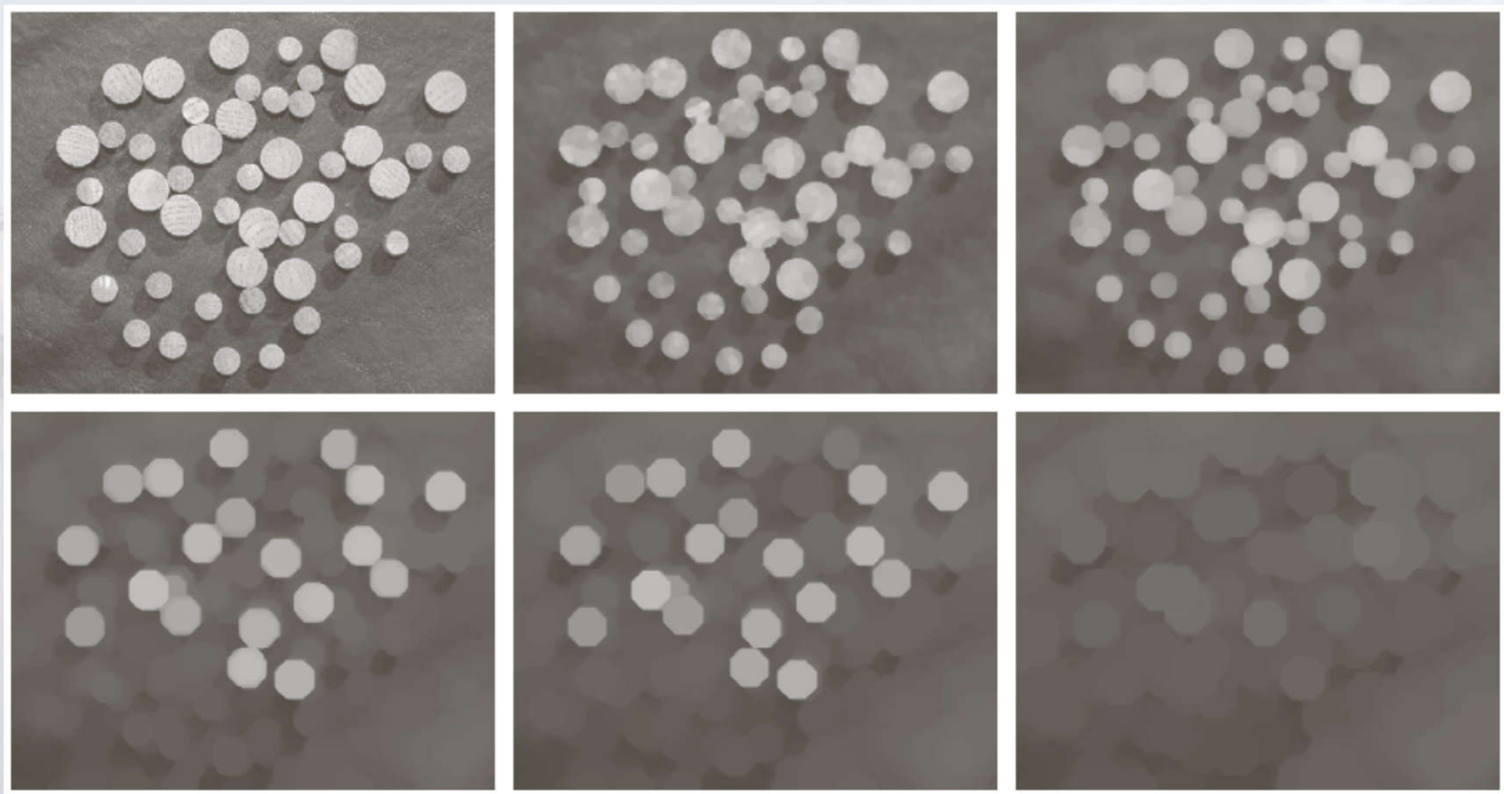
All verified by openings!

Non-linear image processing: mathematical morphology

A granulometry is a family of openings $\{\gamma_{\lambda_i}\}_{1\leq\lambda_i\leq n}$ using SEs of progressively larger sizes λ_i.

a b c
d e f

FIGURE 9.41 (a) 531 × 675 image of wood dowels. (b) Smoothed image. (c)–(f) Openings of (b) with disks of radii equal to 10, 20, 25, and 30 pixels, respectively. (Original image courtesy of Dr. Steve Eddins, The MathWorks, Inc.)



The discrete derivative of a granulometry is called a **pattern spectrum**.

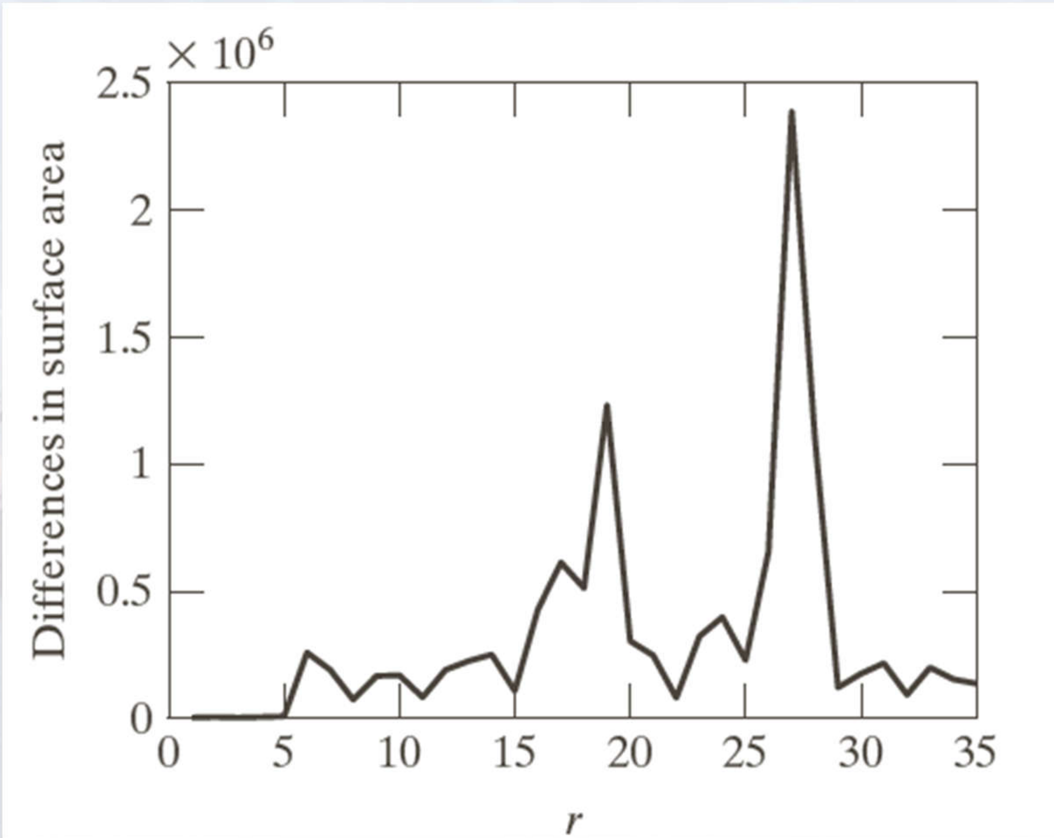


FIGURE 9.42 Differences in surface area as a function of SE disk radius, *r*. The two peaks are indicative of two dominant particle sizes in the image.

Very useful in automated inspection applications.

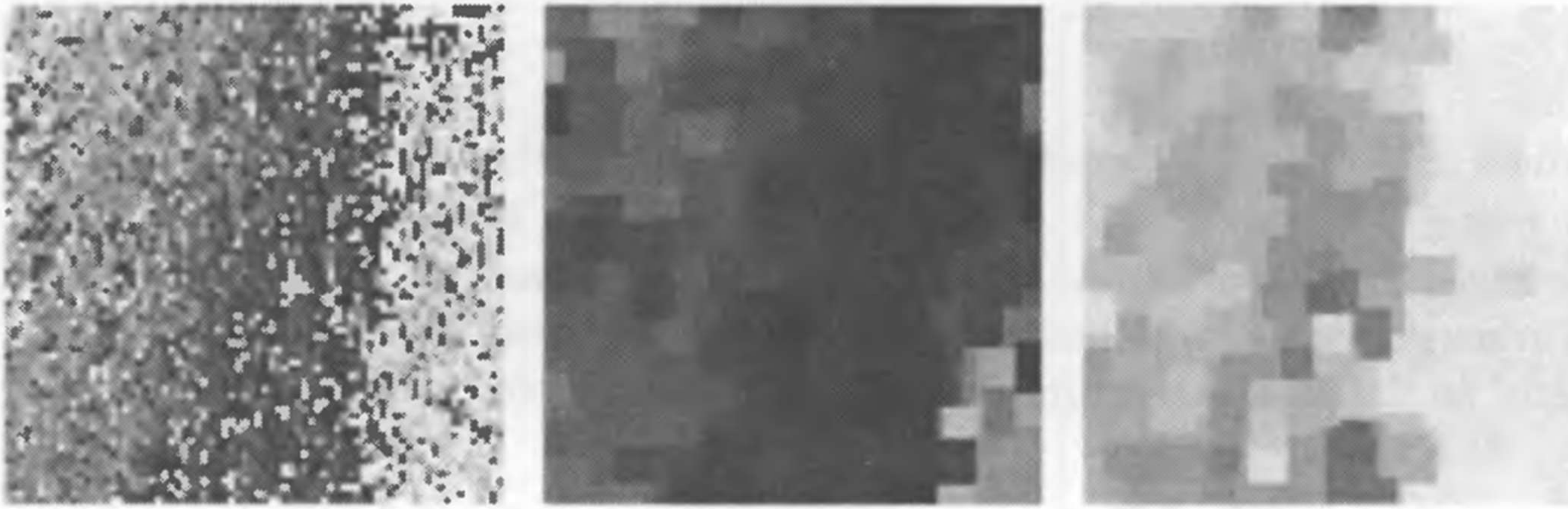
Alternating Sequential Filters (ASF): a composition of increasingly more powerful openings and closings; very effective against **severe noise**.

Let $\mathrm{H} = \{H_i\}_{0 \le i \le n}$ be a family of symmetrical SEs such that:

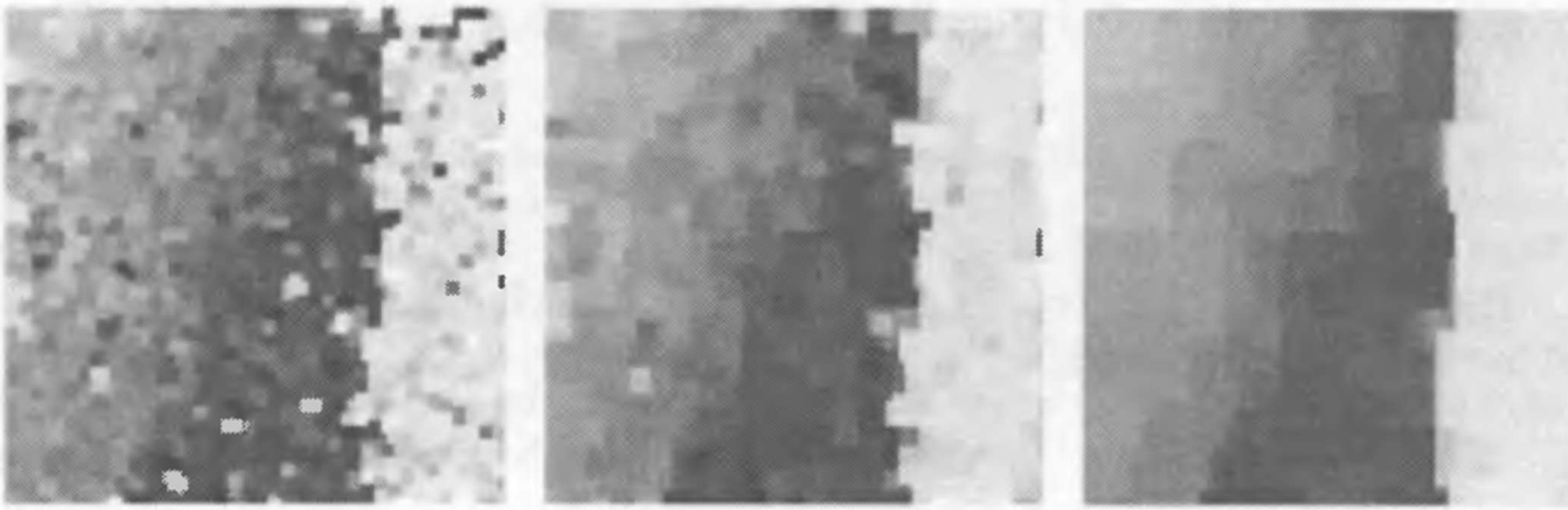
$$H_i \le H_j, \qquad \forall\, i,j \quad 0 \le i \le j \le n$$

The following are ASFs:

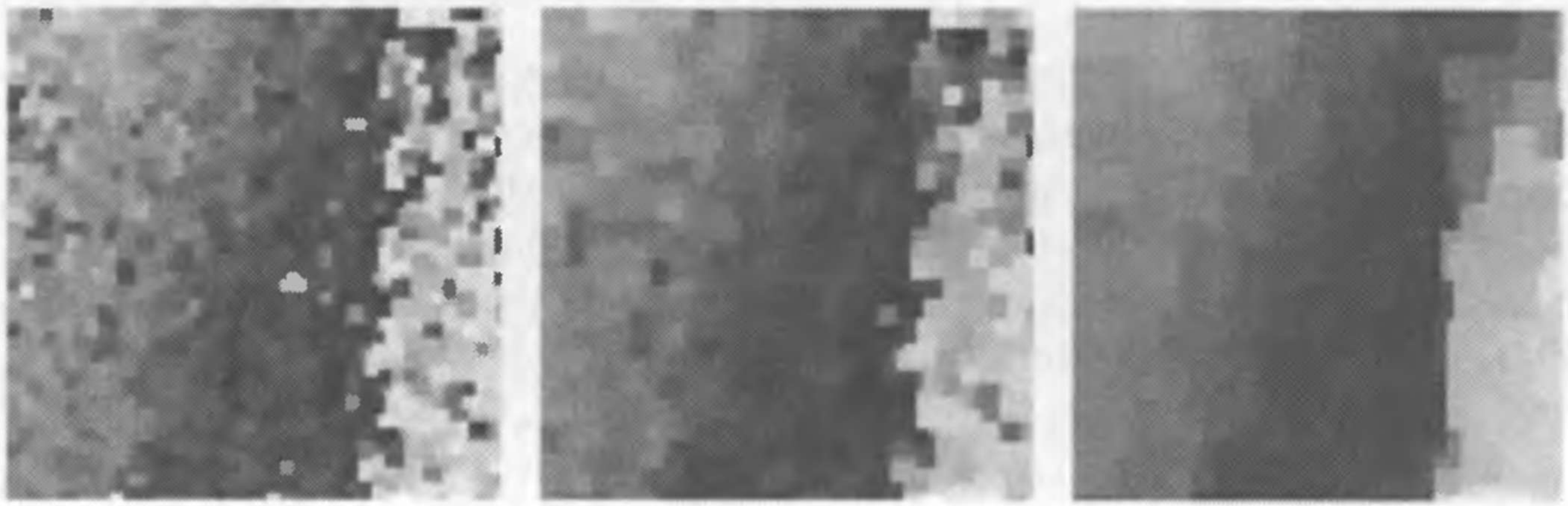
$$N_{\mathrm{H}} = \phi_n \gamma_n \phi_{n-1} \gamma_{n-1} \cdots \phi_0 \gamma_0$$
$$M_{\mathrm{H}} = \gamma_n \phi_n \gamma_{n-1} \phi_{n-1} \cdots \gamma_0 \phi_0$$
$$S_{\mathrm{H}} = \gamma_n \phi_n \gamma_n \gamma_{n-1} \phi_{n-1} \gamma_{n-1} \cdots \gamma_0 \phi_0 \gamma_0$$
$$R_{\mathrm{H}} = \phi_n \gamma_n \phi_n \phi_{n-1} \gamma_{n-1} \phi_{n-1} \cdots \phi_0 \gamma_0 \phi_0$$



(a) Noisy interferogram f. (b) Open-close filter with a 5×5 square. (c) Close-open filter with a 5×5 square.

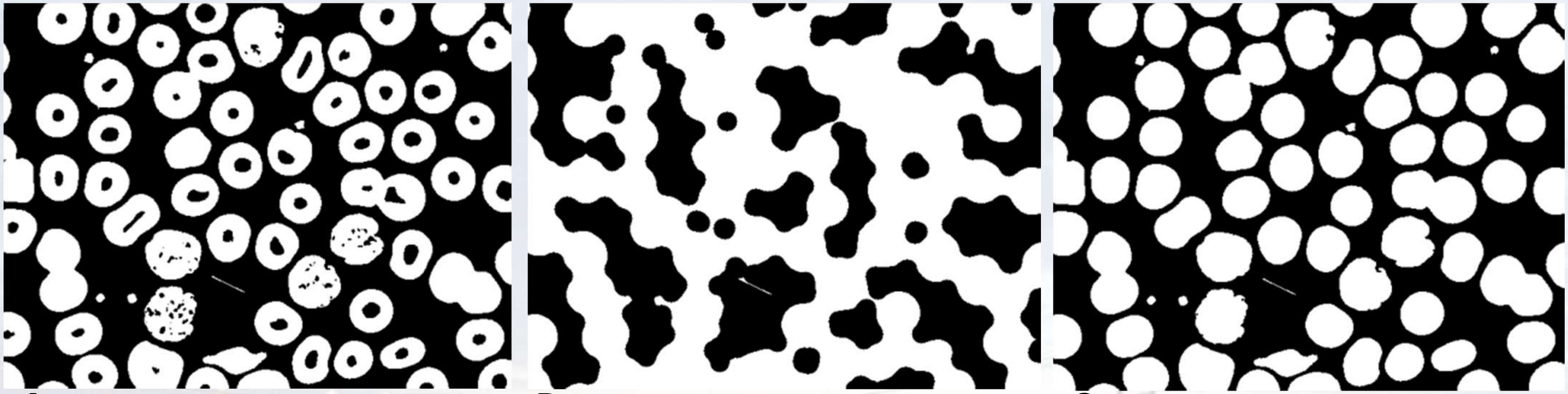


(d) $M_1(f) = \gamma_1[\phi_1(f)]$. (e) $M_2(f) = \gamma_2[\phi_2(\gamma_1[\phi_1(f)])]$. (f) $M_3(f)$.



(g) $N_1(f) = \phi_1[\gamma_1(f)]$. (h) $N_2(f) = \phi_2[\gamma_2(\phi_1[\gamma_1(f)])]$. (i) $N_3(f)$.

Connected operators



A B C

Example adapted from Jean Cousty.

Fill in the holes of (A)

A: input image of cells

B: closing of (A), holes are filled but the contours have moved

C: closing by reconstruction of (A), holes are filled and contours are stable.

In the grayscale case, a connected component is a collection of pixels with the same value that are connected to each other w.r.t either 4- or 8-pixel connectivity.

0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	0	0	0	0	0
0	1	0	0	0	0	0	2	2	0
0	0	0	0	1	1	0	0	2	0
0	0	1	1	1	1	1	0	0	0
0	0	1	2	2	1	0	0	1	0
0	0	1	2	2	1	0	1	1	0
0	0	1	2	2	1	0	1	1	0
0	0	1	1	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0

A **connected operator/filter** is an operator that

- Merges disjoint sets
- Assigns new gray level/color to them

Consequently they are strictly **edge-preserving**; i.e. they neither create new edges, nor move existing ones.

Connected operators

- operate on the level of CCs instead of pixels, leading to higher-level tools.
- are independent of the underlying connectivity relation; hence highly flexible.
- cannot remove noise along edges (can be solved with second order connectivity)
- cannot separate objects linked by noise (same as above)

Examples: **reconstruction based operators**, **levelings** (a self-dual (i.e. same output from both f and f^c) version of reconstruction based filters), **attribute filters**.

The leveling $\lambda(f|g)$ of f from marker g:

$$\big(\lambda(f|g)\big)(x) = \begin{cases} \delta^{(i)}(f,g)(x), \text{if } f(x) \leq g(x) \\ \big(\delta^{(i)}(f^c,g^c)\big)^c(x), \text{otherwise} \end{cases}$$

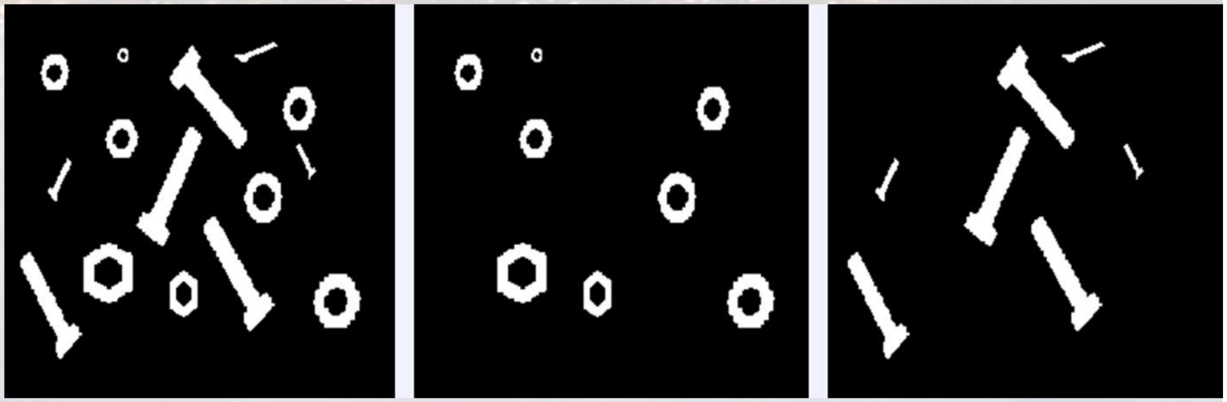
Adapted from M. Wilkinson

Attribute filters are powerful filters that remove connected components from an image not verifying an arbitrary (geometric, spectral, etc) criterion. Examples:

- "Does the CC have an area of at least 300 pixels?"
- "Does the CC have a circularity of at most 0.3?"
- "Does the CC have a standard deviation of 0.25?"…

The flexibility in defining the criterion leads to powerful filters that are no longer limited by fixed SE choices.

They are known since 1996 (Breen and Jones) and their efficient implementation relies on **tree representations**.



Adapted from Urbach and Wilkinson

More formally, let $T: \mathcal{P}(E) \rightarrow \{true, false\}$ be an increasing criterion;
i.e. $C \subseteq D \Rightarrow T(C) \subseteq T(D)$

A binary **trivial opening** $\Gamma_T: \mathcal{P}(E) \rightarrow \mathcal{P}(E)$ using T is defined as:

$$\Gamma_T(C) = \begin{cases} C, \text{if } T(C) \\ \emptyset, \text{otherwise} \end{cases}$$

The criterion can typically look like $T(C) = (\mu(C) \geq \lambda)$, where λ is the attribute threshold, and μ an increasing scalar attribute; i.e. $C \subseteq D \Rightarrow \mu(C) \leq \mu(D)$

A binary attribute opening Γ^T is:

$$\Gamma^T(X) = \cup_{x \in X} \Gamma_T(\Gamma_x(X))$$

where $\Gamma_x(X)$ is the CC of X containing x.

Adapted from M. Wilkinson

Non-linear image processing: mathematical morphology

X $\quad$ $T = A(C) \geq 11^2$ $\quad$ $T = I(C) \geq 11^4/6$

We obtain an **area opening** by using the criterion area A and a **moment of inertia opening** by using the moment of inertia I as attribute.

Adapted from M. Wilkinson

If the criterion is not increasing, instead of trivial opening and attribute opening we call the operators respectively **trivial thinning** and **attribute thinning**. Their extensive dual versions (attribute closing & attribute thickening) are obtained by operating on the complement of the input, and complementing the output.

Generalizing attribute filters to grayscale images is possible through *threshold decomposition*. The threshold set X_h of level h of the image f:

$$X_h(f) = \{x \in E | f(x) \geq h\}$$

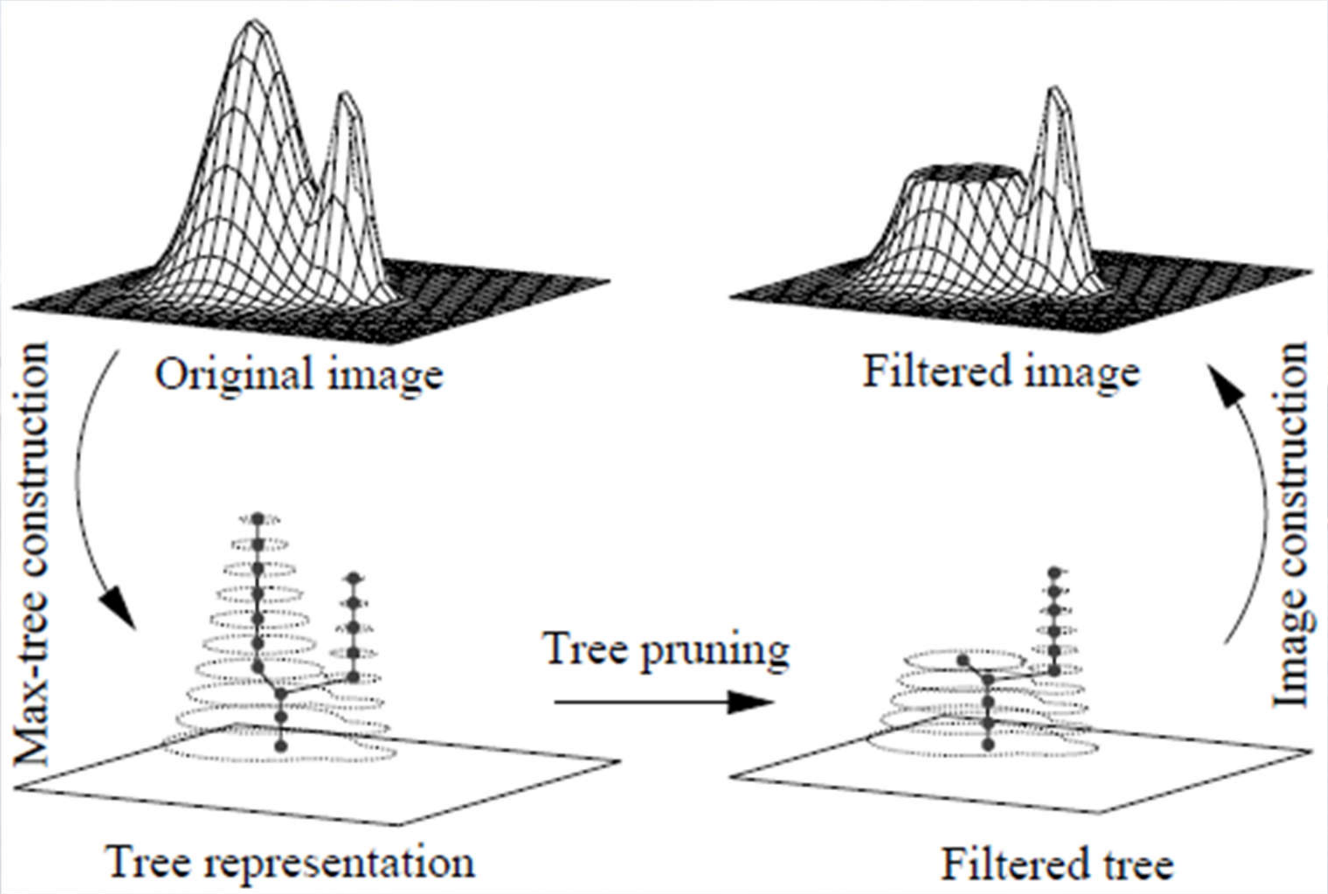
Thus the grayscale attribute opening γ^T becomes:

$$(\gamma^T(f))(x) = \sup\{h \leq f(x) | x \in \Gamma^T(X_h(f))\}$$

Their dual is defined through complementation.

…there is just one problem: **threshold decomposition is slow**.

Idea: convert the image into a tree representation encoding the CC information, and if you need to remove a CC, just prune the tree. When you are done, return to the regular grid based representation.

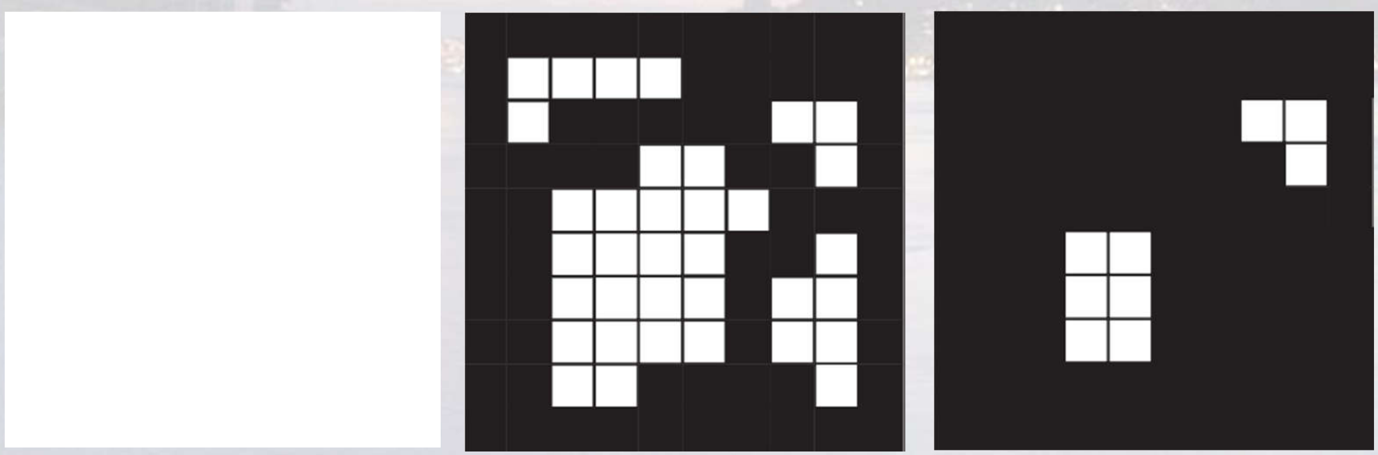


Adapted from Salembier and Garrido

One efficient implementation is the max-tree based one (Salembier *et al.*, 1998); flexible and $O(N)$.

It creates a tree where every node represents a **peak component**; i.e. a connected component of X_h

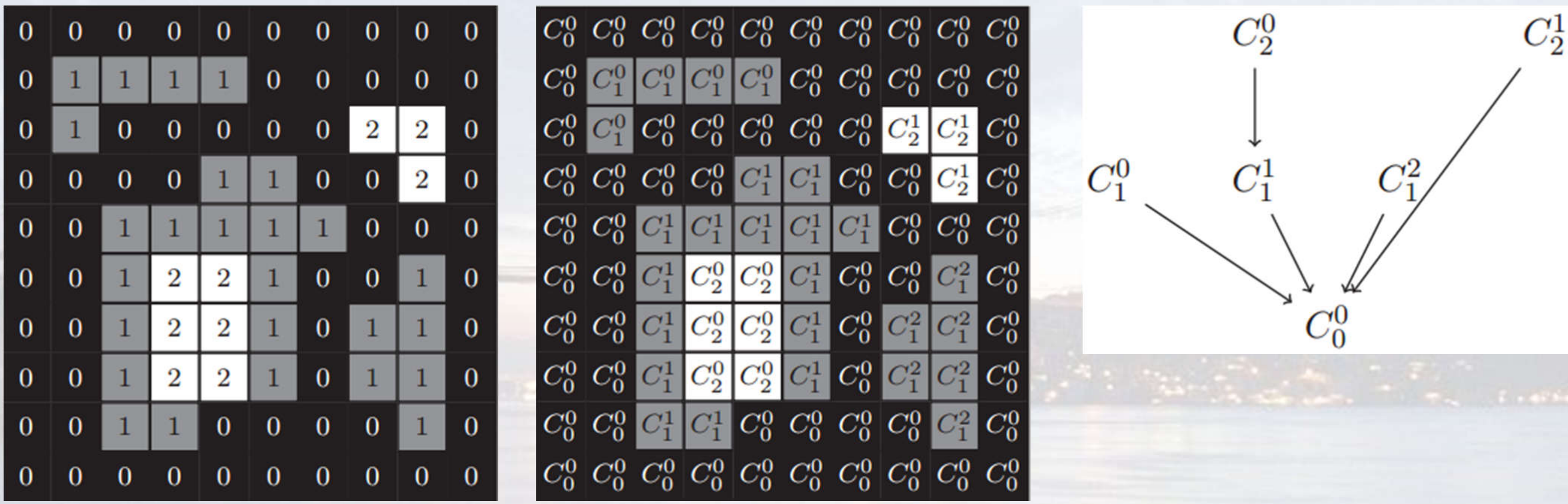
$h = 0$ $h = 1$ $h = 2$



One efficient implementation is the max-tree based one (Salembier *et al.*, 1998); flexible and $O(N)$.

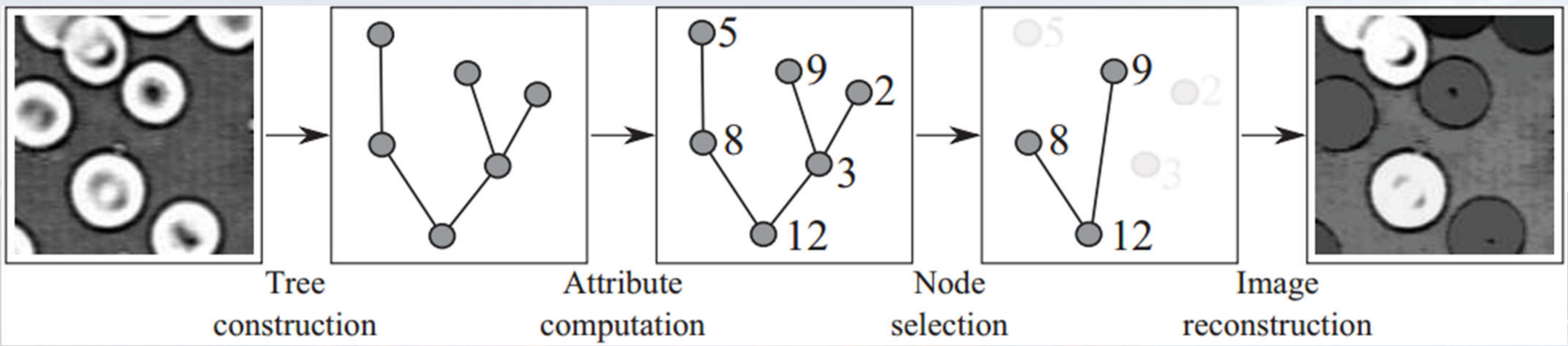
It creates a tree where every node represents a **peak component**; i.e. a connected component of X_h

The max tree can be stored for successive filterings.



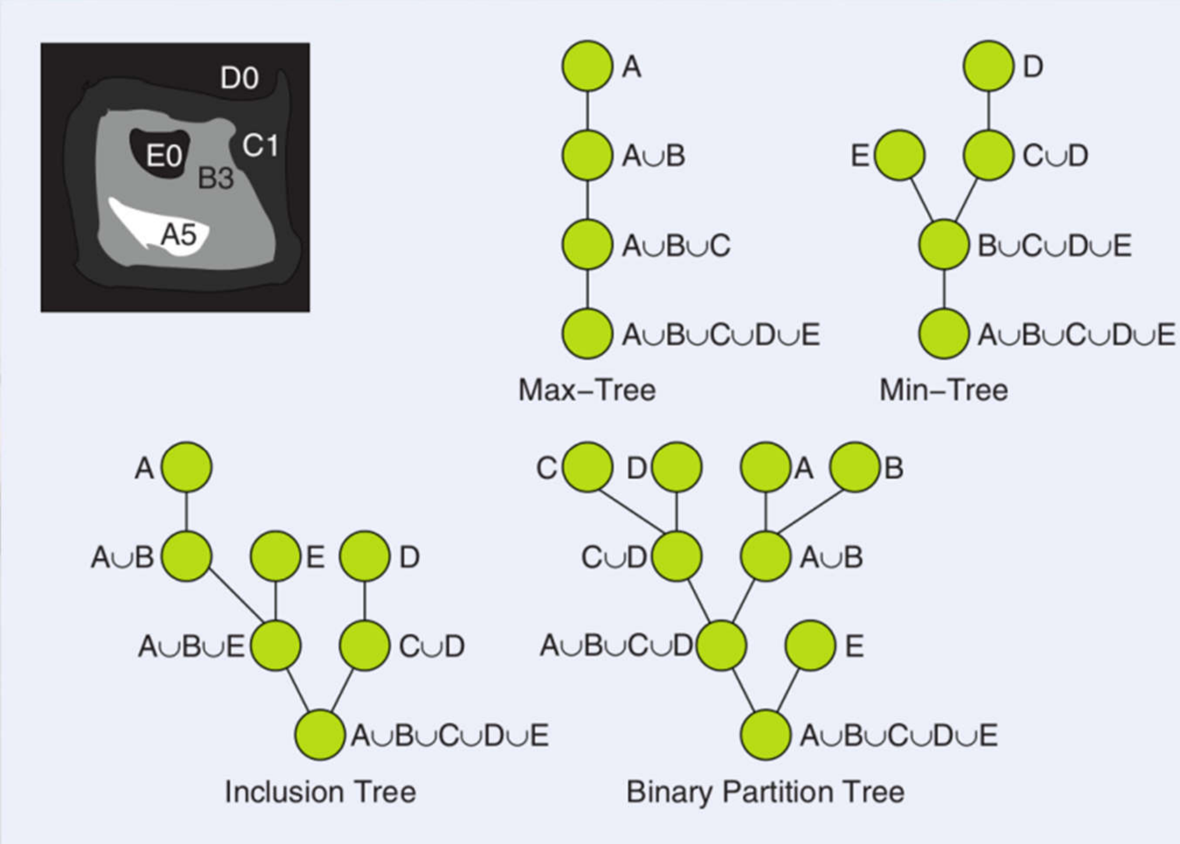
Adapted from M. Dalla Mura et al.

Non-linear image processing: mathematical morphology



Adapted from Perret and Collet

There are many types of trees, such as **inclusion trees** (max/min, tree of shapes) and **partition trees** (alpha trees, omega trees, binary partition trees), and many different strategies for pruning/filtering them (max/min/direct/subtractive, etc).



[FIG8] **Tree representations. The root is at the bottom.**

Adapted from Salembier and Wilkinson

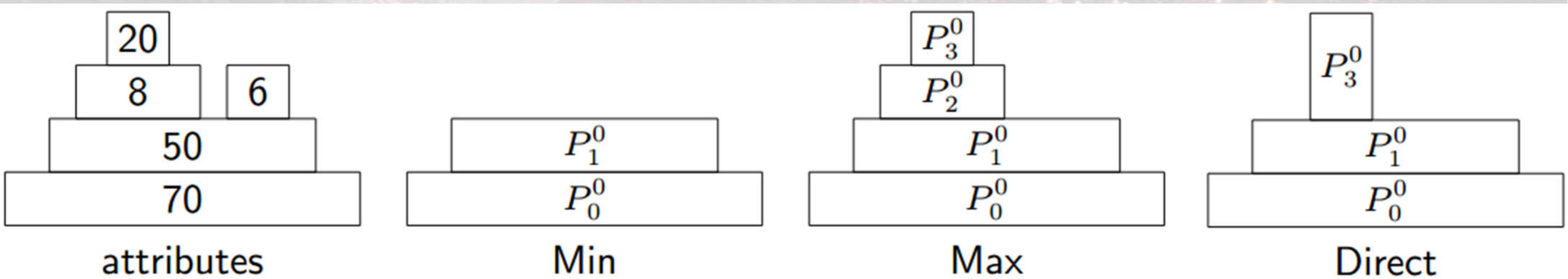
Direct filtering: a node C is removed if the criterion is not satisfied. The pixels stored in C but not in any of its descendants take the gray level of the first preserved ancestor starting from C to the root.

Max filtering: a node C is removed along with its descendants if the criterion is not satisfied for it and all of its descendants.

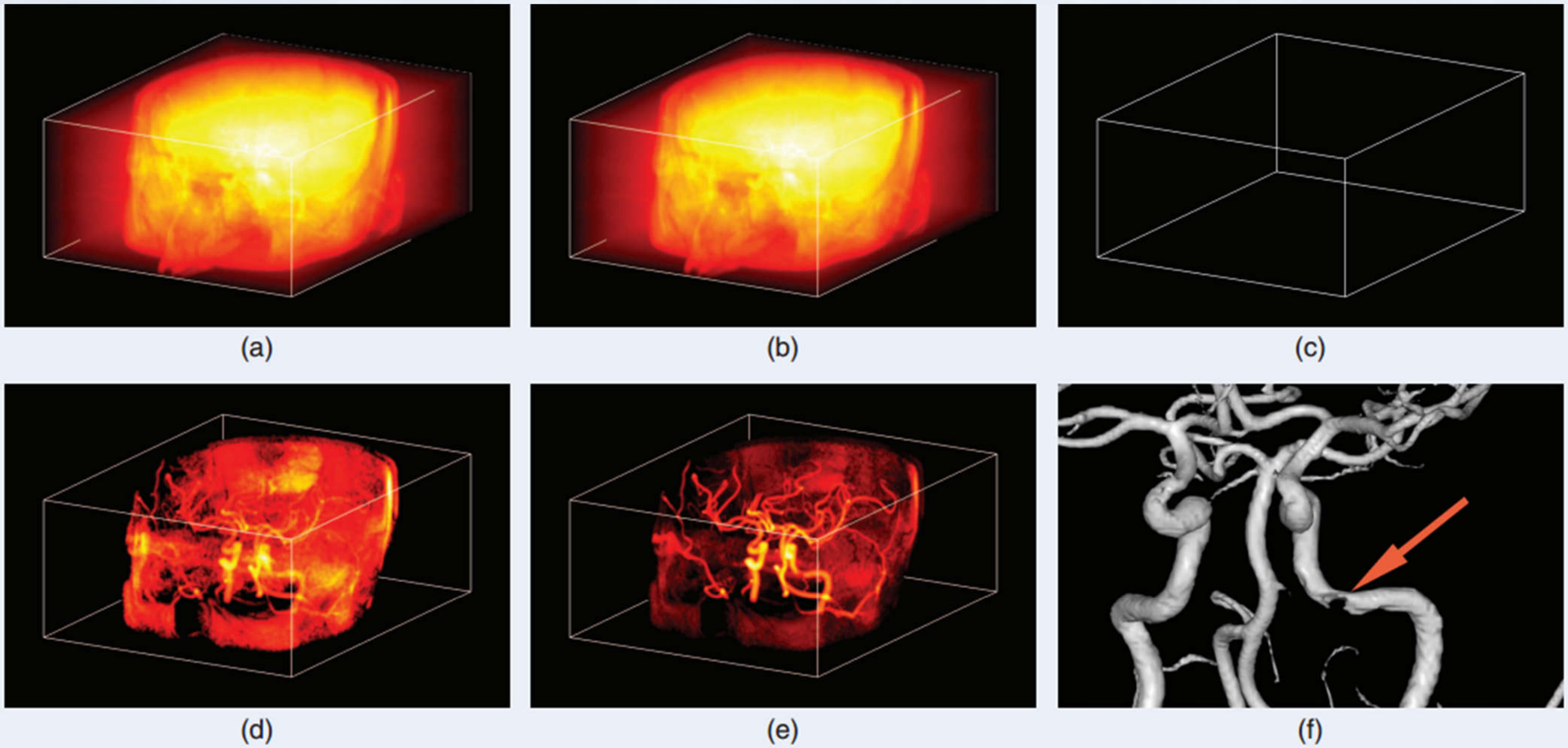
Min filtering: a node C is removed along with its descendants if the criterion is not satisfied for it or one of its ancestors.

and more..

$\lambda = 10$



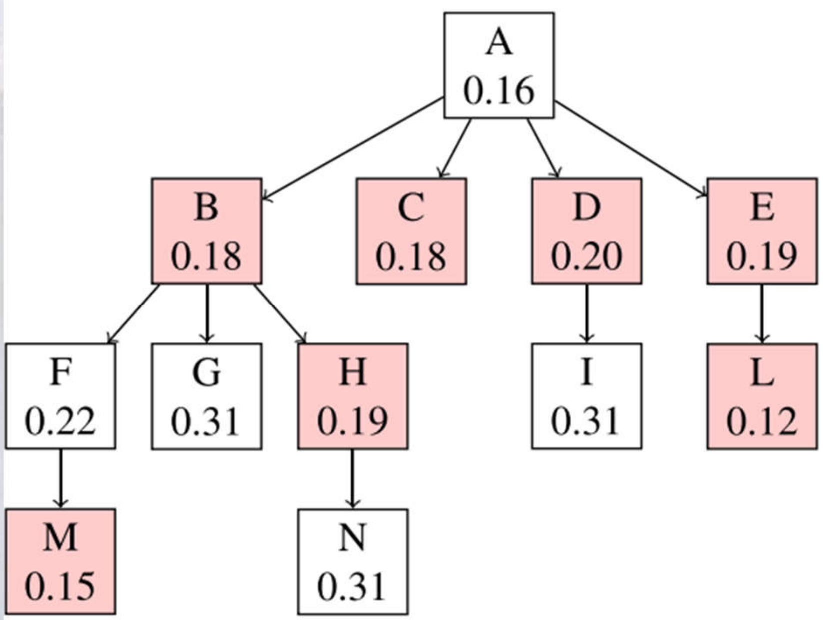
Adapted from M. Wilkinson

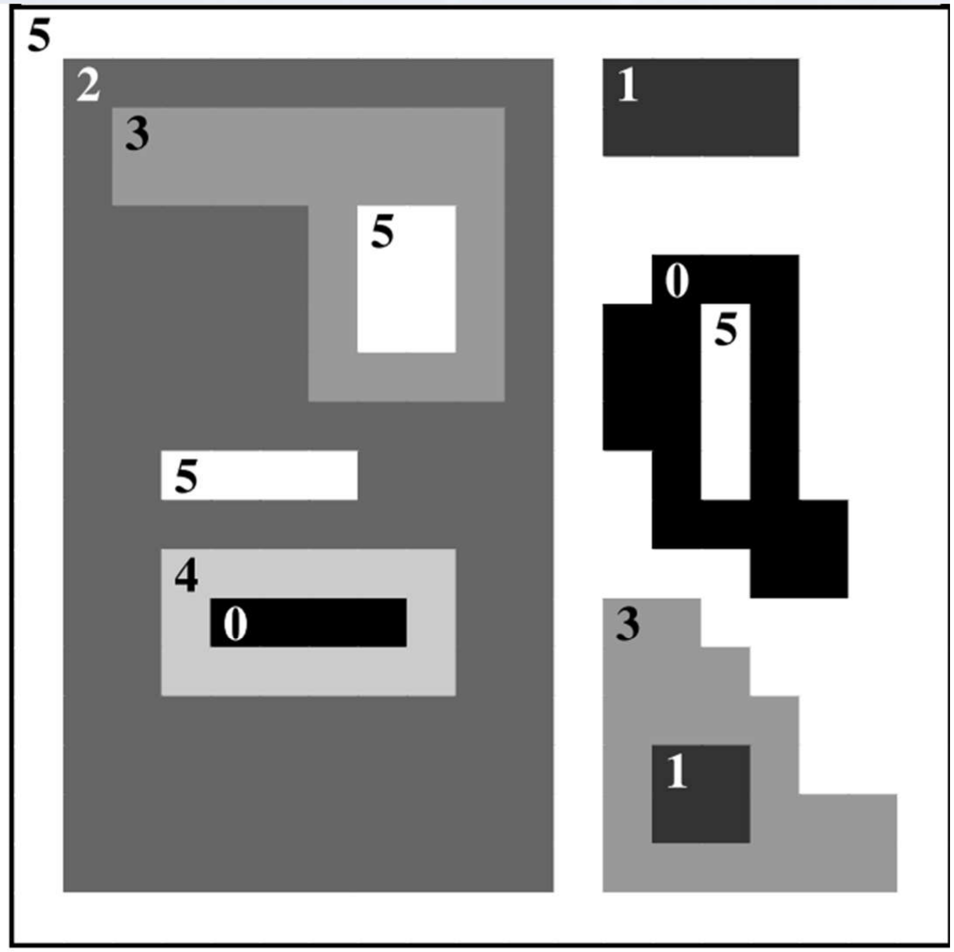


[FIG12] An example of 3-D attribute filtering: (a) X-ray rendering of magnetic resonance angiogram; (b)–(e) result of attribute filter using different filtering rules (b) max; (c) min; (d) direct; (e) subtractive; and (f) detail of iso-surface rendering, showing blood clot-like structure. Images generated using the mtdemo program (http://www.cs.rug.nl/~michael/MTdemo/).

Non-linear image processing: mathematical morphology

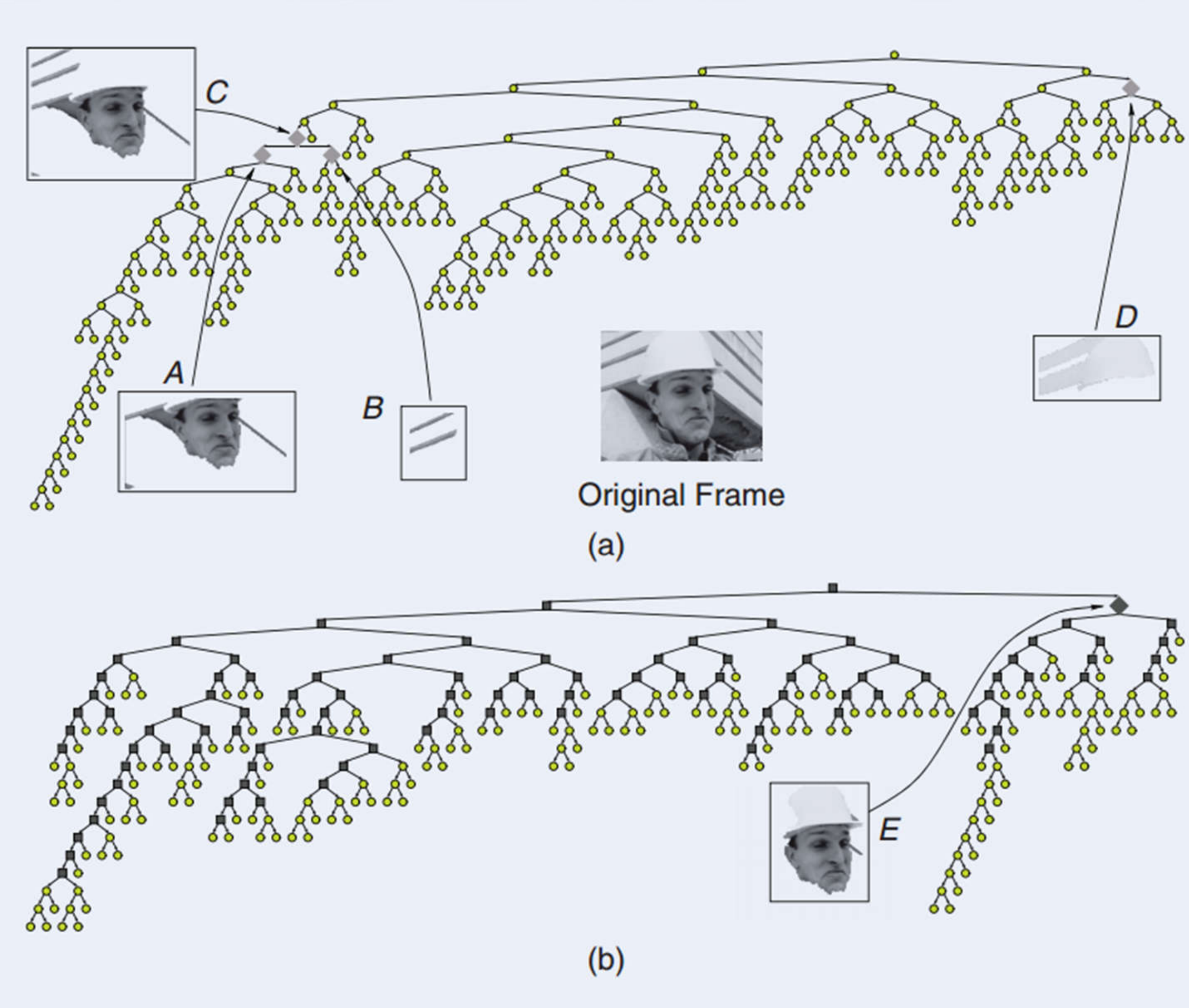
The **tree of shapes** is a **self dual** tree; i.e. you get the same tree from the image and from its complement. A contrast-invariant tree!





Adapted from Cavallaro et al.

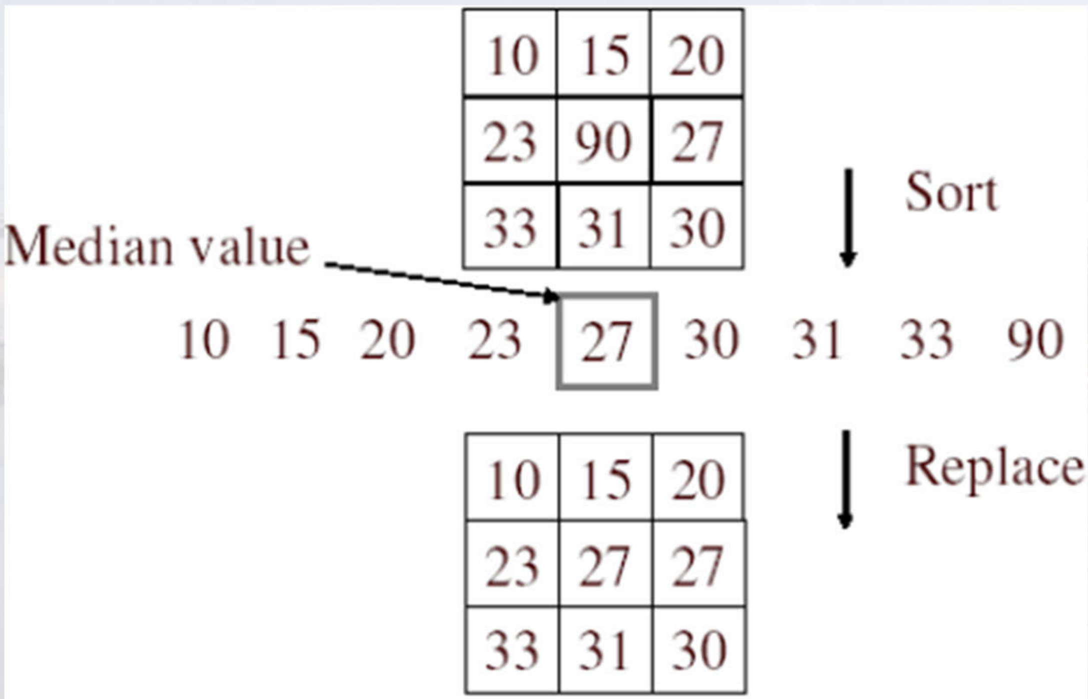
The **binary partition tree** is constructed by merging similar regions.



[FIG9] **Examples of creation of BPT: (a) color homogeneity criterion and (b) color and motion homogeneity criteria.**

Rank filters (or **order-statistic filters**) are **non-linear** filters operating in the spatial domain. Their output is based on the ranking (or ordering) of the pixel values under the filter.

Their most famous representative is the **median filter**.

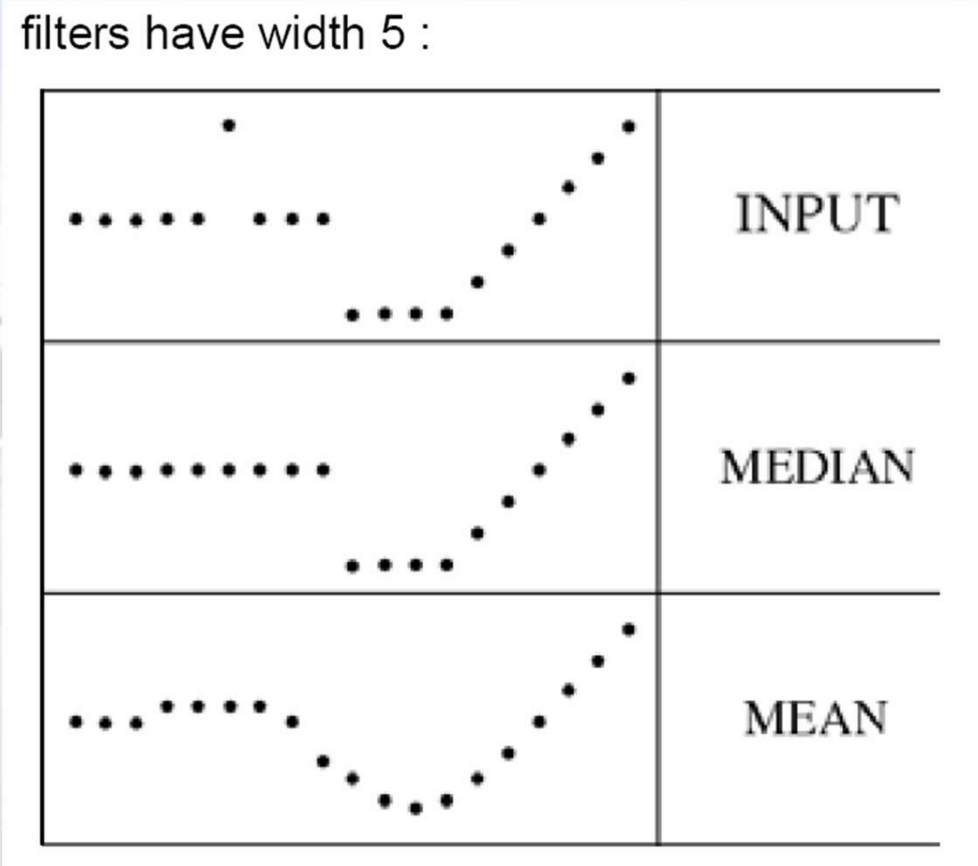


Can you prove that it's not linear?

Properties of the median filter

- Better than mean filter at preserving edges
- Since it selects one the input values as output, it does not create new and potentially unrealistic values.
- Relatively expensive since it

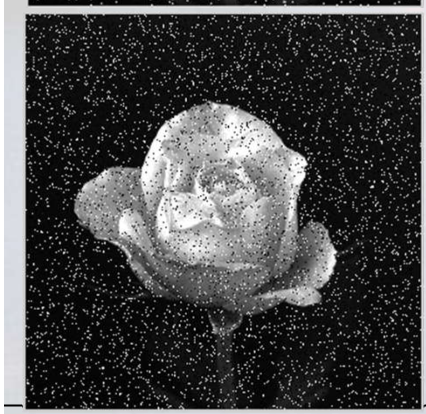
requires sorting



Non-linear image processing: mathematical morphology


Noise density 0.01


Noise density 0.05




Noise density 0.09

Non-linear image processing: mathematical morphology

Gaussian filtering with progressively greater σ

p = 10%

3x3 window

5x5 window

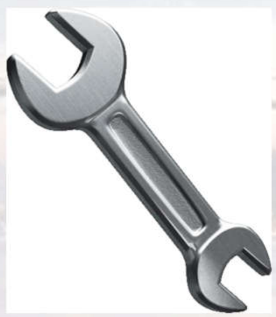
7x7 window

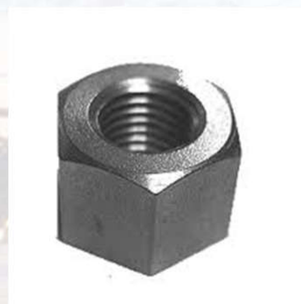
Non-linear image processing: mathematical morphology

Gaussian filter

Gaussian noise

Non-linear filtering (median)





Salt-pepper / impulse noise

Frequency techniques

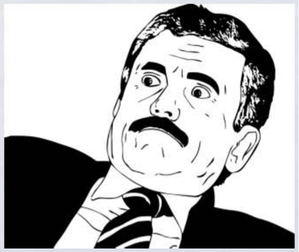
Periodic noise

The **alpha-trimmed mean filter** is a hybrid of linear and non-linear filtering.

You first sort the pixel values under the window of size N,
then you discard a certain number (alpha) of them from each end,
and the output is the mean of the remaining ones.

If alpha = 0 it becomes identical to the mean filter
If alpha = (N x N - 1) / 2 it becomes the median filter.

It can (somewhat) handle both impulse and Gaussian noise.



Adaptive filters are a particular class of sophisticated filters that adapt themselves to the data handed to them; there are both linear and non-linear examples.

The **bilateral filter** (late 1990s) is a classic example.

- Every pixel value will be replaced with a weighted average of its neighbors.
- The weight decreases the farther away you move from the mask center
- The weight also decreases in proportion to pixel intensity difference.

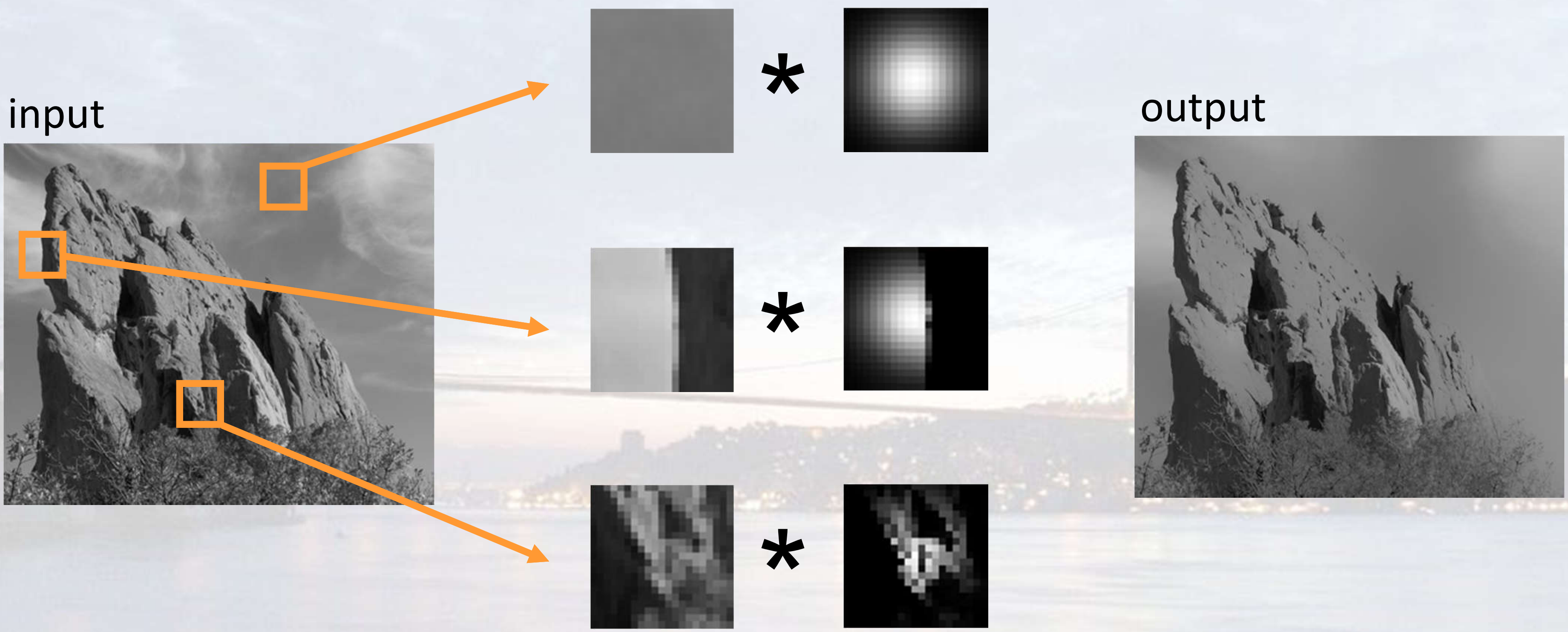
With a Gaussian function:

$$w(i,j,k,l) = e^{\left(-\frac{(i-k)^2+(j-l)^2}{2\sigma_d^2} - \frac{\|I(i,j)-I(k,l)\|^2}{2\sigma_r^2}\right)}$$

The bilateral filter is expensive to calculate by brute force, but there are efficient algorithms now

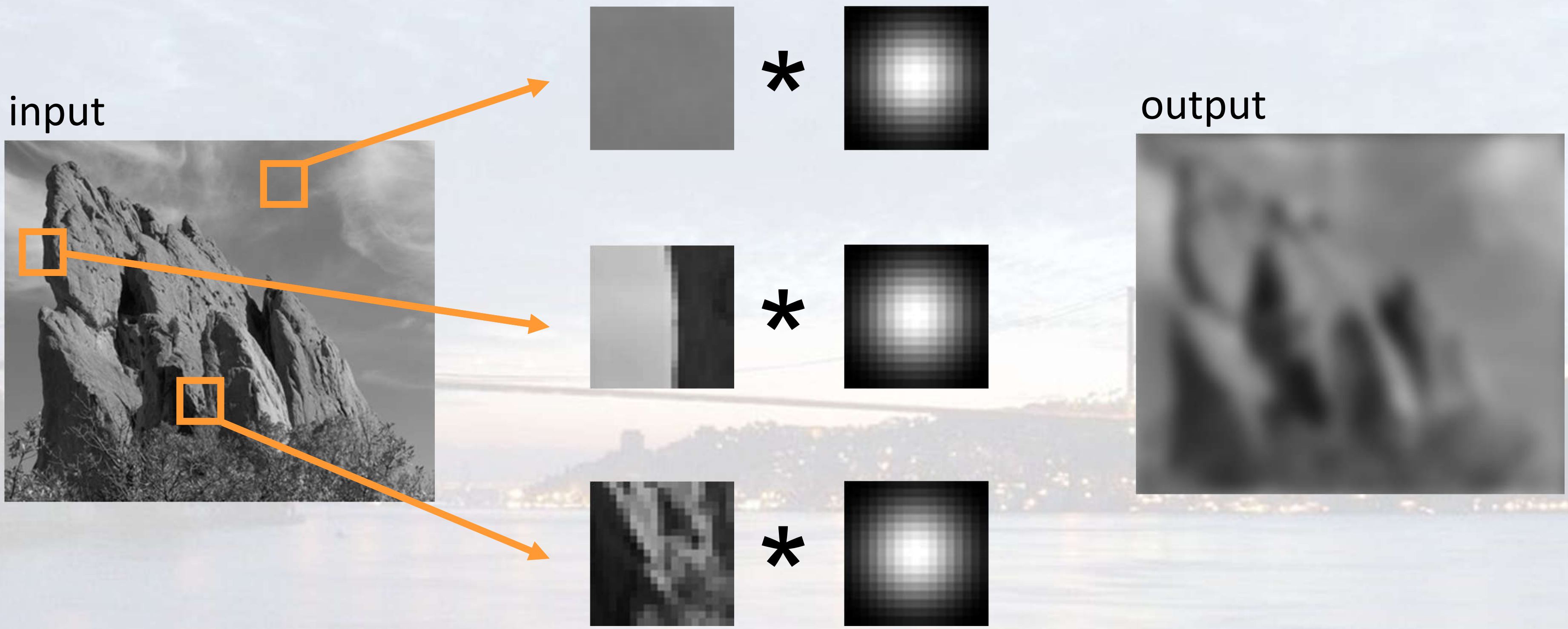
Non-linear image processing: mathematical morphology

Bilateral filter



Adapted from Selim Aksoy and Sylvain Paris

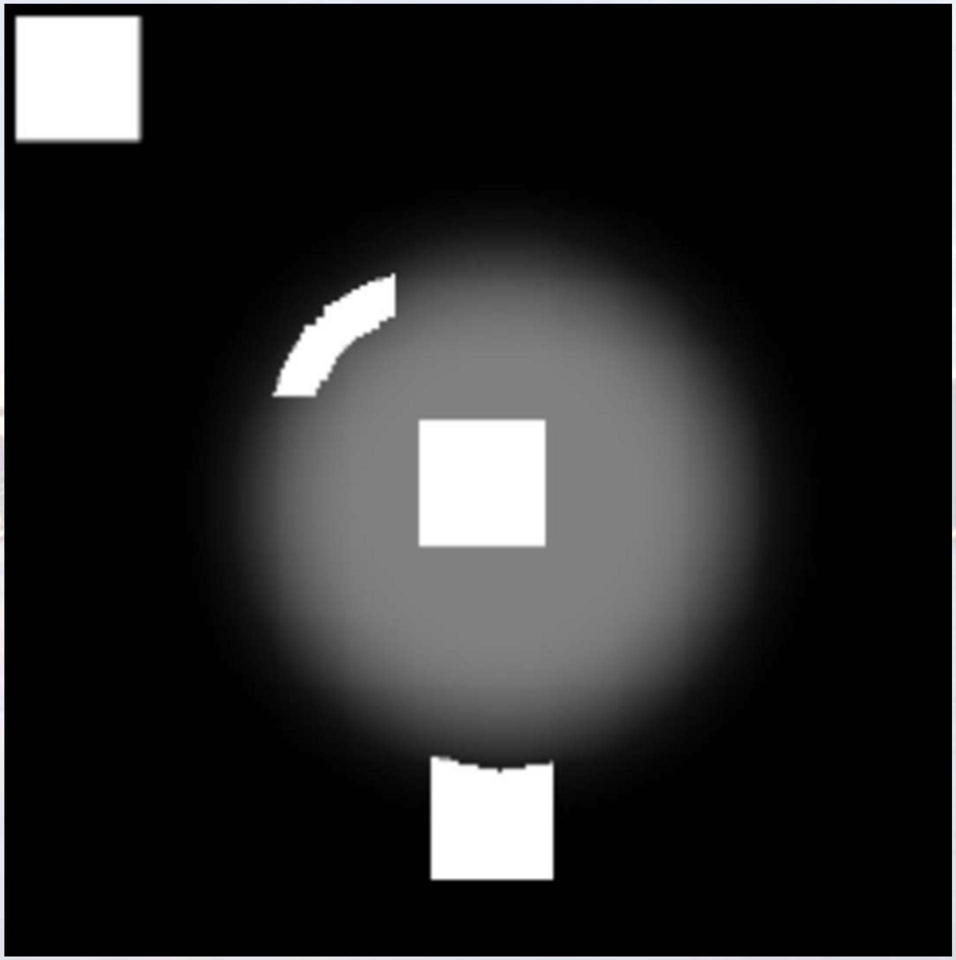
Gaussian filter



Adapted from Selim Aksoy and Sylvain Paris

Mathematical morphology has spatially variant filters as well, they are commonly known as **morphological amoebas**

Their principle is to adapt the structuring element's shape to the gradient of the input.



Example of morphological amoebas

The non-linear image processing world is vast, this was just an introduction. There are still many more concepts out there:

- Path filtering
- Anisotropic diffusion, etc.