Image Processing and Imaging
Morphological Image Processing

Christof Kauba
Fachbereich Computerwissenschaften
Universität Salzburg

Wintersemester 2020/21

Outline

Introduction - Binary Images (1)

Binary images:

- Having only two gray levels
- Constitute an important subset of digital images
- A binary image (e.g., a silhouette or an outline) normally results from an image segmentation operation
- If the initial segmentation is not completely satisfactory $\rightarrow$
- Some form of processing done on the binary image can often improve the situation

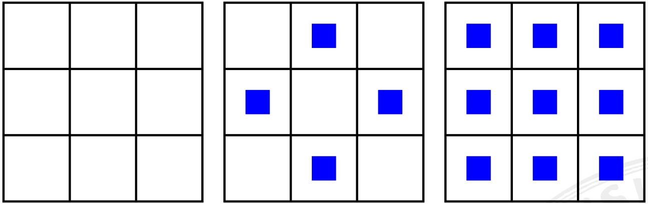
- Many of the processes discussed here can be implemented as 3×3 neighborhood operations
- Binary image: any pixel, together with its neighbors, represents nine bit of information
- There are only $2^9 = 512$ possible configurations for a 3×3 neighborhood in a binary image

- Convolution of a binary image with a 3×3 kernel generates a nine-bit (512-gray-level) image:

Introduction - Binary Images (2)

- Gray level of each pixel specifies the configuration of the 3×3 binary neighborhood centered on that point
- Neighborhood operations thus can be implemented with a 512-entry look-up table with one-bit output

16	8	4
32	1	2
64	128	256



0   128+32+8+2=170   511

- This can be used to implement a logical operation called a *hit-or-miss transformation*
- Look-up table is loaded to search for a particular pattern—e.g. all nine pixels being black
- Output is one or zero, depending on whether the neighborhood matches the mask
- If, whenever the pattern is matched (a hit), the central pixel is set to white
- Central pixel of all other configurations is left unchanged (a miss) $\rightarrow$
- Operation would reduce solid objects to their outlines by eliminating interior points

Outline

Mathematical Morphology:

- A powerful set of binary image processing operations developed from a set-theoretical approach
- Basic operations are simple
- They and their variants can be concatenated to produce much more complex effects
- Are amenable to a look-up table implementation in relatively simple hardware for fast *pipeline processing*
- While commonly used on binary images
- Can be extended to gray-scale images as well

Morphological Image Processing:

- In general case, operates by passing a *structuring element* over the image in an activity similar to convolution
- Like the convolution kernel, structuring element can be of any size, and it can contain any complement of 1's and 0's
- At each pixel position, a specified logical operation is performed between the structuring element and the underlying binary image

- Binary result of that logical operation is stored in the output image at that pixel position
- Effect created depends upen the size and content of the structuring element and upon the nature of the logical operation

- For this introduction to the subject, we concentrate on the simplest case, namely, the use of a basic 3×3 structuring element containing all 1's
- With this restriction, it is the logical operation that determines the outcome

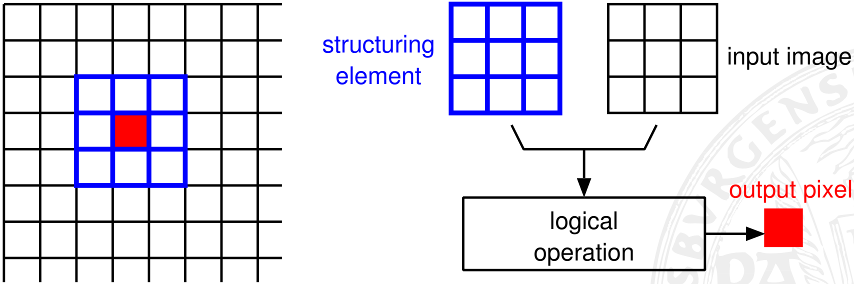


Figure: Morphological image processing

Outline

Language of Morphological Processing:

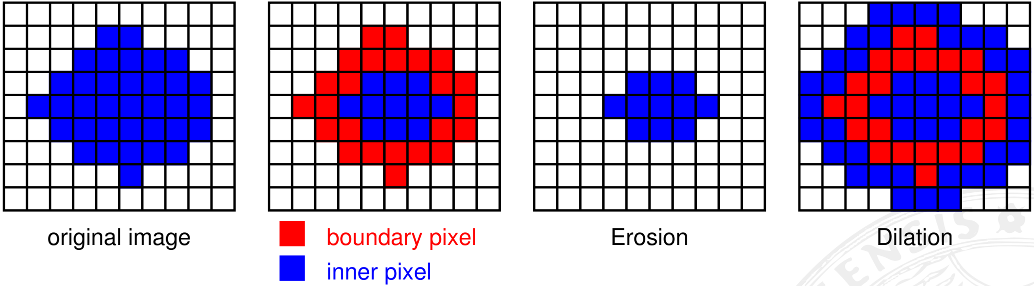
- Both the binary image, **B**, and the structuring element, **S**, are sets defined on a two-dimensional Cartesian grid,
- 1's are the elements of those sets

- We denote by $\mathbf{S}_{xy}$ the structuring element after it has been translated so that its origin is located at the point (x, y)
- Output of a morphological operation is another set
- Operation can be specified by a set-theoretical equation

Outline

Erosion and Dilation (1)

- The basic morphological operations are erosion and dilation
- By definition, a boundary point is a pixel that is located inside an object, but that has at least one neighbor outside the object



- Simple **erosion** is the process of eliminating all the boundary points from an object
- Leaving the object smaller in area by one pixel all around its perimeter
- If the object is circular, its diameter decreases by two pixels with each erosion
- If it narrows to less than three pixels thick at any point, it will become disconnected (into two objects) at that point
- Objects no more than two pixels thick in any direction are eliminated

Erosion and Dilation (2)

- Useful for removing from a segmented image objects that are too small to be of interest
- General erosion is defined by:

$$\mathbf{E} = \mathbf{B} \otimes \mathbf{S} = \{x, y | \mathbf{S}_{xy} \subseteq \mathbf{B}\} \tag{1}$$

- Binary image **E** that results from eroding **B** by **S** is the set of points (x, y) such that:
  - If **S** is translated so that its origin is located at (x, y), then it is completely contained within **B**
- With the basic 3×3 structuring element, general erosion reduces to simple erosion

Erosion and Dilation (3)

Dilation:

- Simple **dilation** is the process of incorporating into the object all the background points that touch it leaving it larger in area by that amount
- If the object is circular, its diameter increases by two pixels with each dilation
- If the two objects are seperated by less than three pixels at any point, they will become connected (merged into one object) at that point
- Dilation is useful for filling holes in segmented objects

- General dilation is defined by

$$\mathbf{D} = \mathbf{B} \oplus \mathbf{S} = \{x, y | \mathbf{S}_{xy} \cap \mathbf{B} \neq \emptyset\} \tag{2}$$

- Binary image **B** that results from dilating **B** by **S** is the set of points (x, y) such that:
  - If **S** is translated so that its origin is located at (x, y), then its intersection with **B** is not empty
- With the basic 3×3 structuring element, this reduces to simple dilation

Outline

Opening and Closing (1)

Opening:

- Erosion followed by dilation
- Effect of eliminating small thin objects, breaking objects at thin points, and generally smoothing the boundaries of larger objects without significantly changing their area
- Opening is defined by:

$$\mathbf{B} \circ \mathbf{S} = (\mathbf{B} \otimes \mathbf{S}) \oplus \mathbf{S} \tag{3}$$

Closing:

- Dilation followed by erosion
- Effect of filling small and thin holes in objects, connecting nearby objects, and generally smoothing the boundaries of objects without significantly changing their area
- Closing is defined by:

$$\mathbf{B} \bullet \mathbf{S} = (\mathbf{B} \oplus \mathbf{S}) \otimes \mathbf{S} \tag{4}$$

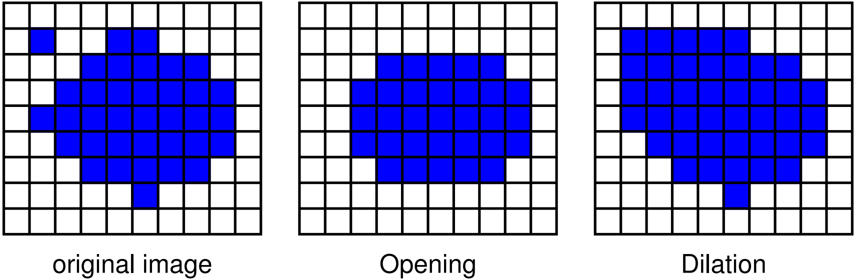


Figure: Opening and closing

When noisy images are segmented by thresholding:

- Resulting boundaries are quite ragged
- Objects have false holes
- Background is peppered with small noise objects

- Successive openings or closings can improve the situation markedly
- Sometimes several iterations of erosion, followed by the same number of dilations, produces the desired effect

Outline

Shrinking

- Erosion implemented in such a way that single-pixel objects are left intact
- Useful when the total object count must be preserved

- Can be used iteratively to develop a size distribution for a binary image containing approximately circular objects
- It is run alternately with a 3×3 operator that counts the number of single-pixel objects in the image
- With each pass, the radius is reduced by one pixel, and more of the objects shrink to single-pixel size
- Recording the count at each iteration gives the cumulative distribution of object size
- Highly noncircular objects (e.g., dumbbell-shaped objects) may break up while shrinking $\rightarrow$ this technique has its restrictions

Outline

Erosion programmed as a two-step process that will not break objects:

- First step is a normal erosion, but it is conditional:
  - Pixels are marked as candidates for removal, but are not actually eliminated
- In the second pass, those candidates that can be removed without destroying connectivity are eliminated, while those that cannot are retained
- Each pass is a 3×3 neighborhood operation that can be implemented as a table-lookup operation
- Thinning reduces a curvilinear object to a single-pixel-wide line, showing its topology graphically

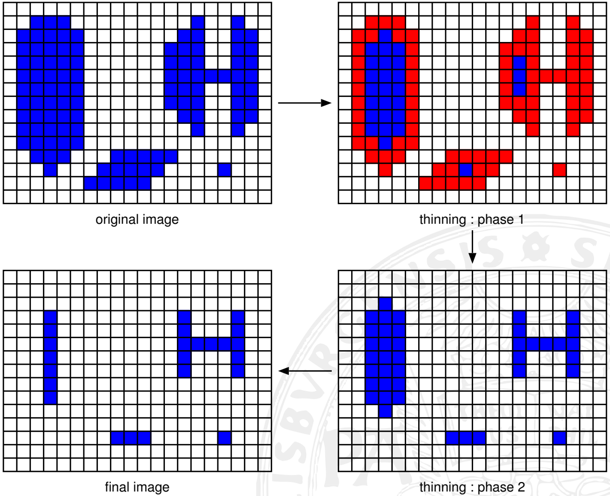


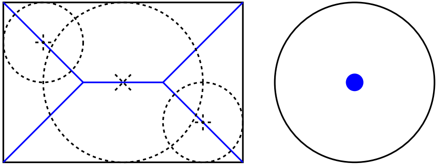
Figure: Thinning

Outline

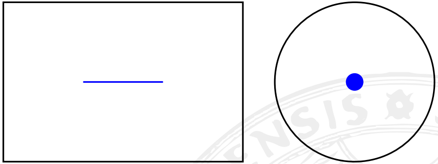
- Is an operation related to thinning
- Also known as *medial axis transform* or the *grass-fire technique*
- Medial axis is the locus of the centers of all the circles that are tangent to the boundary of the object at two or more disjoint points
- Skeletonization is seldom implemented by actually fitting circles inside the object

Conceptually, the medial axis can be thought of as being formed in the following way:

- Imagine that a patch of grass, in shape of the object, is set on fire all around the periphery at once
- As the fire progresses inward, the locus of points where advancing fire lines meet is the medial axis



skeletonization



thinning

- Skeletonization can be implemented with a two-pass conditional erosion, as with thinning
- Rule for deleting pixels, however, is slightly different

Outline

Pruning

- Thinning or skeletonization process will often leave spurs on the resulting figure
- Spurs are short branches having an endpoint located within three or so pixels of an intersection

- Spurs result from single-pixel-sized undulations in the boundary that give rise to a short branch
- Can be removed by a series of 3×3 operations that remove endpoints (thereby shortening all the branches), followed by reconstruction of the branches that still exist.

- A three-pixel spur, for example, disappears after three iterations of removing endpoints
- Not having an endpoint to grow back from, the spur is not reconstructed

Outline

Thickening

- Dilation can be implemented so as not to merge nearby objects
- This can be done in two passes, similarly to thinning
- An alternative is to complement the image and use the thinning operation on the background

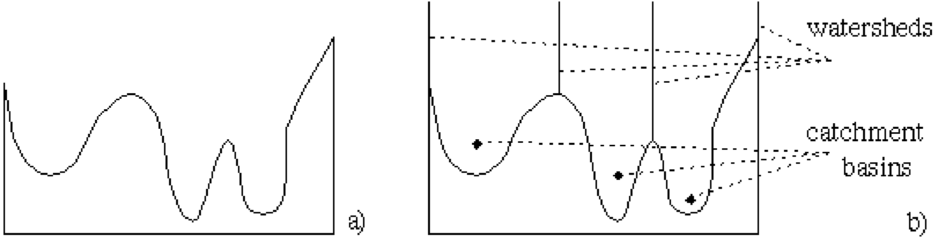
- In fact, each of the variants of erosion has a companion dilation-type operation obtained when it is run on a complemented image

- Some segmentation techniques tend to fit rather tight boundaries to objects so as to avoid erroneously merging them
- Often, the best boundary for isolating objects is too tight for subsequent measurement
- Thickening can correct this by enlarging the boundaries without merging separate objects

Outline

A good (animated) visualisation and several examples may be found at:
`http://cmm.ensmp.fr/~beucher/wtshed.html`

- Watersheds separate different (water)basins
- Transfer this notion into image processing context →
- Image is interpreted as three dimensional structure:
  - Luminance values are interpreted as height measure (elevation at the corresponding image coordinates)
  - In many cases a gradient image is used for further processing
  - Region borders (i.e. edge chains) correspond to “high valued” edge chains and inner areas with low gradient correspond to basins



- Basins are homogeneous in the sense that all pixels belonging to a catchment basin are connected to the minimum value of the basin by a path, the pixel of which are monotonically decreasing in direction to the minumum
- Basins represent the regions of the segemented image, the watersheds represent region borders

There are two diffent approaches to watershed transform:

1. Determines a "downstream" path to a minimum for each pixel; A catchment basin is defined as the set of pixels the path of which leads to the same minimum; problem is the unique determination of the path (which can be facilitated by local gradients in the contineous case).
2. Dual to the first one and uses "flooding": catchment basins are flooded from below (assume that holes are at the location of minima in the 3D surface, when submerging the surface, water enters through the holes); as soon as two catchment basins would be fused due to rising water, a dam is constructed to prevent this; value of the dam pixels is set to the maximum value of the image.

Watershed Segmentation - Flooding Approach (1)

We focus onto the second technique:

- Preprocessing: pixels are sorted according to their gray-scale, gray-scale histogram is computed, and a list of pointers is built pointing at pixels with gray-scale h
- In this manner, all pixels with a specific gray-scale can be addressed efficiently

- Assume that flooding has been propagated until gray-scale k
- Each pixel with gray-scale $\leq k$ has been uniquely assigned to a basin and carries its label
- In the next step all pixels with gray-scale $k+1$ are processed
- A pixel with this gray scale can be assigned to basin l in case at least one direct neighbour carries label l

- In order to determine membership to a basin, zones of influence are defined:
  - Zone of influence of a basin l are the positions of the not-assigned but connected pixels with gray-scale $k+1$, the distance to l of which is smaller than to any other basin.

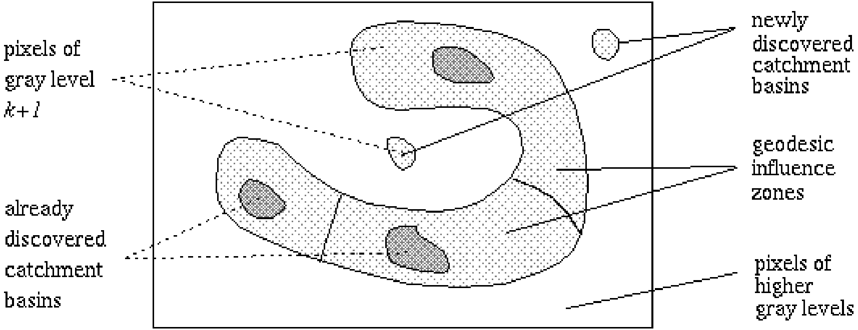


Figure: Zones of influence of basins

- All pixels with gray-scale $k+1$ belonging to the influence zone of the basin l are assigned the label l, which means that catchment basins grow
- Not-yet assigned pixels are processed successively, pixels without label correspond to new basins and get a new label
- Borders separating the catchment basins are the watersheds

Watershed Segmentation - Flooding Approach (3)

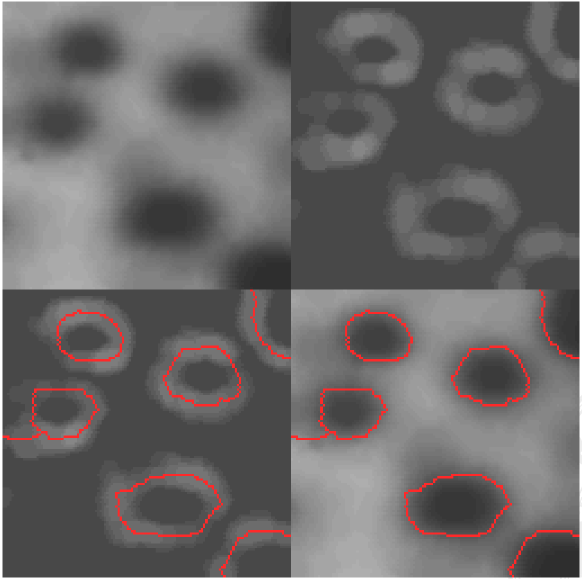
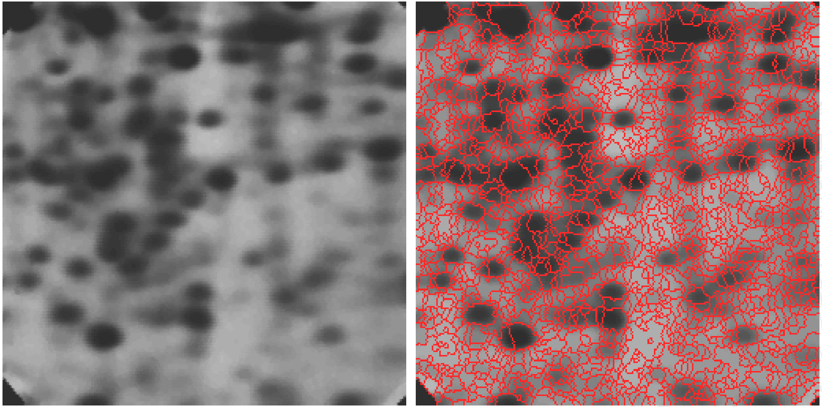


Figure: Original, gradient image, watersheds, original with watersheds

Note: We have not described how to explicitly construct dams – this will be explained subsequently using morphological operators

Oversegmentation: When applying watershed segmentation as described so far, we result in significant oversegmentation quite often (i.e. too may regions are formed), since the number of minima is simply too high (e.g. caused by noise):

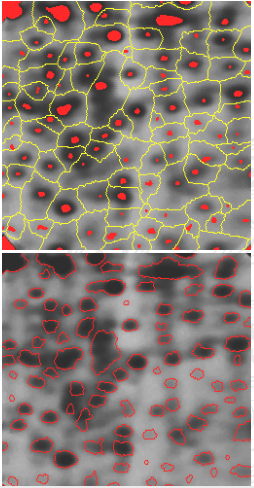


In order to limit this effect, the follwoing strategy can be applied:

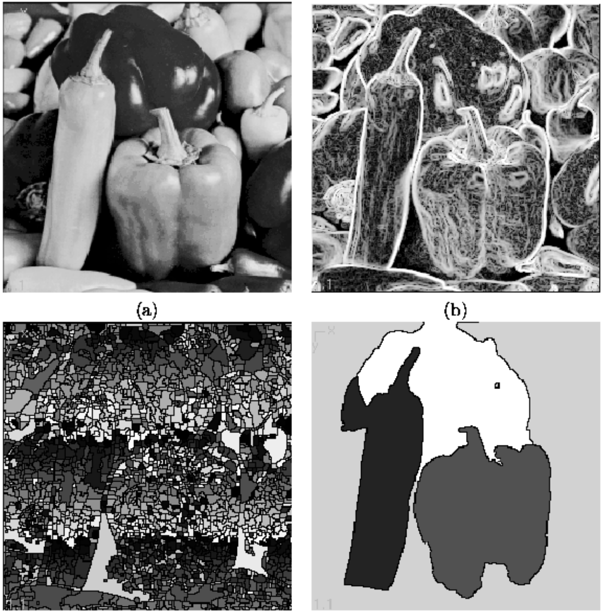
- Image smoothing (e.g. Gauss filtering with large σ)
- Marker: Only "internal markers" are accepted as local minima, i.e. contigous regions with identical gray-scale surrounded by pixels with larger value.

Markers:

- Setting internal markers and subsequently applying the watershed algorithm to the image
- Resulting watershed lines are denoted as "external markers", which partition the image into regions
- Each region contains a single object with its associated background
- Now we can apply a watershed procedure or thresholding to each region to arrive at the desired segmentation
- Additional measure: number of internal markers can be limited or a minimum size can be required



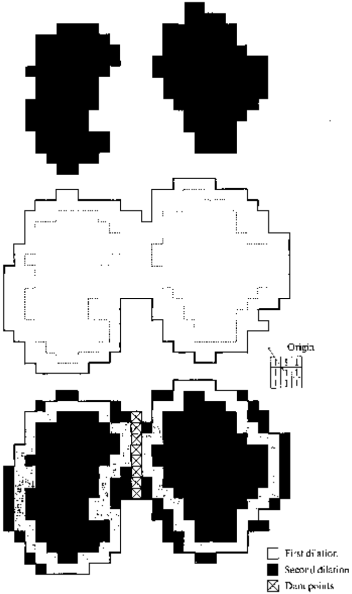
Watershed Segmentation - All Variants Example



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

Dam construction

- If requirement of an explicit dam construction process during flooding:
- Flooding step $n-1$ right before a fusion of two catchment basins is taken as the initial stage for dam construction (black regions in figure)
- Fused region after step n is denoted as q (white region in figure)
- Subsequently, dilation is applied to the two black regions using a constant 1 3 x 3 structuring element, satisfying two conditions:
  1. The center of the structuring element resides in q.
  2. Dilation is only applied without a fusion of the two regions.



Example (right):

- First dilation step: can be conducted without any problems
- The two black regions are enlarged consistently
- Second dilation run: several points do not satisfy the first condition
- That is why the perimeter curve is disconnected
- Points satisfying the first condition but not the second one are defined to be dam points
- These points are set to the maximal luminance value in order not to get over-flooded again
- Then, flodding is continued