Image processing

Advanced smoothing filters and morphological operators

Image Feature Extraction Techniques

Piotr Fulmański

FACULTY OF MATHEMATICS
AND COMPUTER SCIENCE
University of Lodz

Advanced smoothing filters

Nonlocal means

Definition

You can think about the *nonlocal means* operator as an advanced, more sophisticated version of the averaging operator mentioned earlier.

The basic function of the operator is to assign a point a value that is the mean of a different areas which are closest to the mean at the value of the point, rather than the mean at that point.

Difficult to understand?

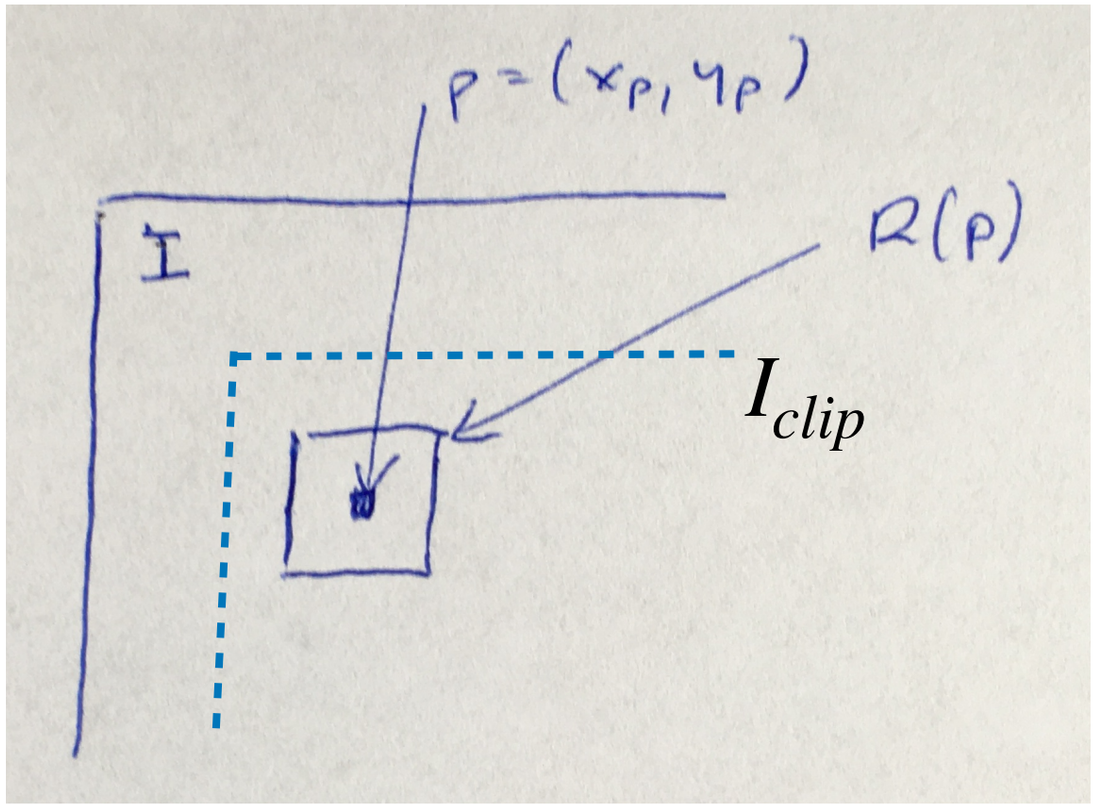
Let's explain it step by step.

Nonlocal means

Step 1

Let $p = (x_p, y_p)$ is a point of intensity v_p from input image I.

Define a smal region surrounding a given point $p = (x_p, y_p) \in I_{clip} \subset I$ and denote it as $R(p)$. I_{clip} is an image I without some internal border such that for every $p \in I_{clip}$ region $R(p) \in I$.



I will also call this region the same as I called so far: a *window* or a *template*. From practical point of view it is required that region has a rectangular shape and its size n is an od number.

Being more precise, I_{clip} is smaller because $R_{big}(p)$ area should be enclosed in I. I will discuss this in next few slides.

Nonlocal means

Step 2

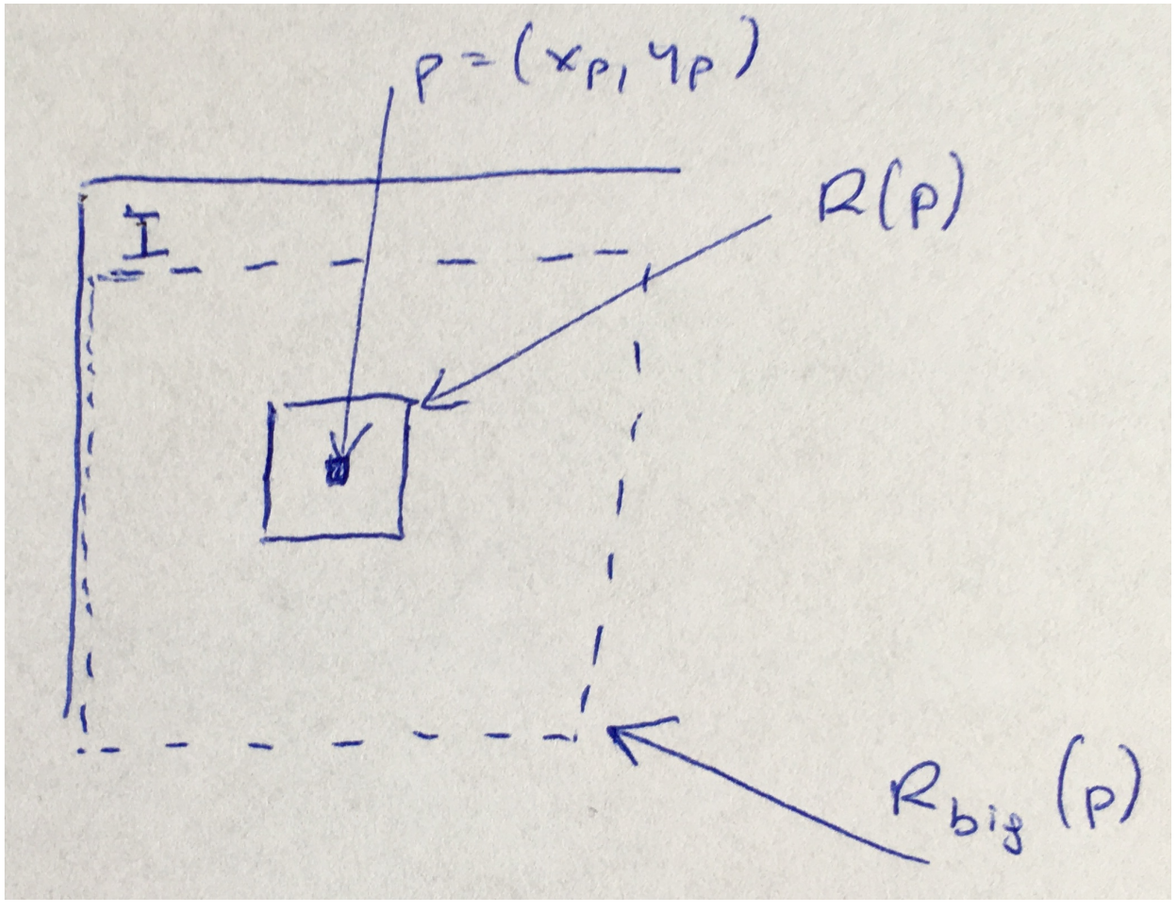
Having $R(p)$ defined, you can compute mean (average) at point p taking into account pixels from region $R(p)$ surrounding this point:

$$a(p) = \frac{1}{n^2} \sum_{p' \in R(p)} v_{p'}.$$

Nonlocal means

Step 3

Define a big region surrounding a given point $p = (x_p, y_p) \in I$ and denote it as $R_{big}(p)$.



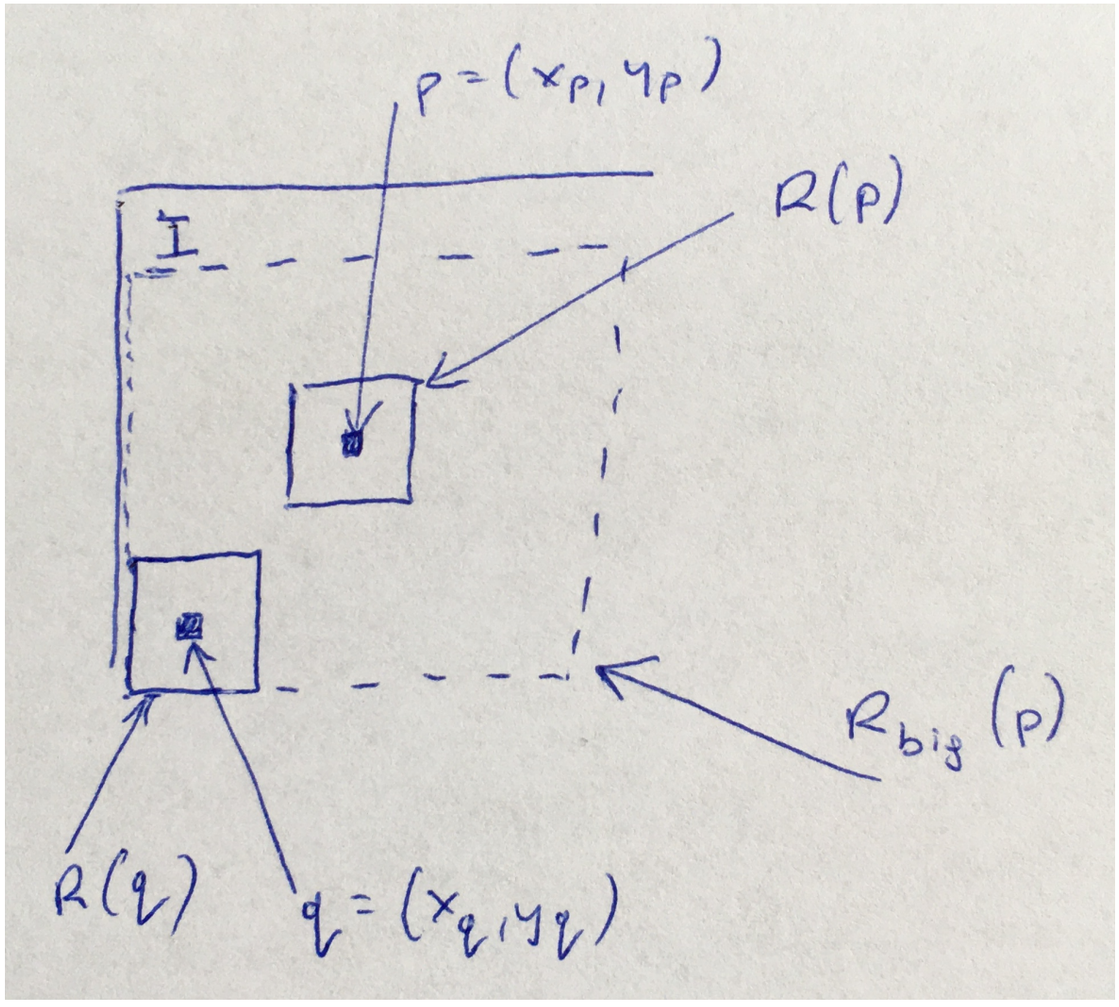
This region has also a rectangular shape, its size m is an od number and $m > n$.

Nonlocal means

Step 4

Having $R_{big}(p)$ defined, you can compute mean (average) at any point $q \in R_{big}(p)$ such that region $R(q)$ surrounding point q is completely enclosed in $R_{big}(p)$ taking into account pixels from $R(q)$:

$$a(q) = \frac{1}{n^2} \sum_{q' \in R(q) \subseteq R_{big}(p)} v_{q'}$$



Nonlocal means

Step 5

Now you can compute how the average $a(p)$ at point p is similar to the average $a(q)$ at point q:

$$f(p,q) = \frac{1}{2\pi\sigma^2} e^{-\frac{(a(p)-a(q))^2}{2\sigma^2}}.$$

This would be your *weighting function*.

The weight will be maximum when $a(p) = a(q)$ and much less when the two averages are very different. The product $a(q)f(p,q)$ is close to $a(p)$ when the mean of the point of interest p is the same as the mean of the region at point q.

Nonlocal means

Step 6

Computed accumulated weights:

$$F(p) = \sum_{q \in R_{big}(p)} f(p, q)$$

Nonlocal means

Step 7

Compute final value v_{p_o} of the output pixel p_o corresponding to pixel p:

$$v_{p_o} = \frac{1}{F(p)} \sum_{q \in R_{big}(p)} q \cdot f(p, q)$$

Nonlocal means

Final words

The parameters that you must choose are:

- n -- the window size of the averaging operator; this defines the size of R,
- m -- the size of the search region; this defines the size of R_{big},
- σ -- the standard deviation.

To have a guarantee that $R_{big}(p)$ is completely enclosed in I, you have to properly defined I_{clip}, for example as:

$$I_{clip} = \{p = (x_p, y_p) \in I : b \leq x_p \leq x_{max} - b, b \leq y_p \leq y_{max} - b\},$$

where:

- $b = floor(m/2)$,
- x_{max} is the greatest x coordinate,
- y_{max} is the greatest y coordinate.

Nonlocal means

Examples

[tutu examples]

Nonlocal means

Conclusions

Nonlocal means operator compared with Gaussian averaging is much more computational expensive but offers less loss of details.

Bilateral filter

Definition

As you know, Gaussian averaging operator mentioned in previous part, blurs every part of an image regardless of the image content: whether some part contains large flat part or sharp details, everything is blurred the same way. In consequence you loose details.

Idea behind bilateral filter is to use two factors influencing the final form of the resulting image:

- one responsible for "traditional" blur effect (this could be, and very of the it is, Gaussian averaging),

- second for limiting blur strength across boundaries of different area by decreasing the filter weight when the intensity difference is too large.

You can think of it as a weighted Gaussian averaging.

Bilateral filter

Definition

Let $p = (x_p, y_p)$ is a point from input image I. Define a smal region surrounding a given point $p = (x_p, y_p) \in I_{clip} \subset I$ and denote it as $R(p)$. I_{clip} is an image I without some internal border such that for every $p \in I_{clip}$ region $R(p) \in I$.

You can define a function:

$$b(p, q) = g(p, q) \cdot w(p, q),$$

where:

- $q \in R(p)$,
- $g(p, q)$ is the spatial (or domain) kernel for smoothing differences in coordinates (this function can be a Gaussian function); this function measures distance,
- $w(p, q)$ is the range kernel for smoothing differences in intensities (this function can also be a Gaussian function; this function detects boundaries.

Very general form of this filter is given below:

$$v_{p_o} = \frac{1}{\sum_{q \in R(p)} b(p, q)} \sum_{q \in R(p)} v_q \cdot b(p, q).$$

Bilateral filter

Definition

When both spatial and range kernels is a Gaussian kernel, $b(p, q)$ is given by the following formula:

$$b(p, q) = g(p, q) \cdot w(p, q)$$

$$= \frac{1}{2\pi\sigma^2} e^{-\frac{(x_p - x_q)^2 - (y_p - y_q)^2}{2\sigma^2}} \cdot \frac{1}{2\pi\sigma^2} e^{-\frac{(v_p - v_q)^2}{2\sigma^2}}$$

where:

- $(x_p - x_q)^2 + (y_p - y_q)^2$ is an Euclidean distance between point p and q,
- $(v_p - v_q)^2$ is a difference in intensity between point p and q.

Bilateral filter

Examples

[tutu examples]

Bilateral filter

Conclusions

Bilateral filter, as all other filters can be use iteratively.

Usually a single iteration produces a much cleaner image than the original, and is probably sufficient for most image processing needs.

Multiple iterations have the effect of flattening the colors in an image considerably, but without blurring edges.

The resulting image of cartoon-like appearance has a much smaller color map. All shadows and edges are preserved.

Morphological operators

Mathematical morphology

Idea

Mathematical morphology is a theory and technique for the analysis and processing of geometrical structures, based on set theory, lattice theory, topology, and random functions.

In mathematical morphology, you process images according to shape, by treating both as sets of points. In this way, morphological operators define local transformations that change pixel values that are represented as sets. The ways pixel values are changed is formalised by the definition of the hit or miss transformation.

The basic idea in morphology is to *probe* an image with a simple, pre-defined shape, drawing conclusions on how this shape fits or misses the shapes in the image. This simple "probe" is called the *structuring element* (instead of coefficient matrix, weight matrix or window as you have seen in previous chapters).

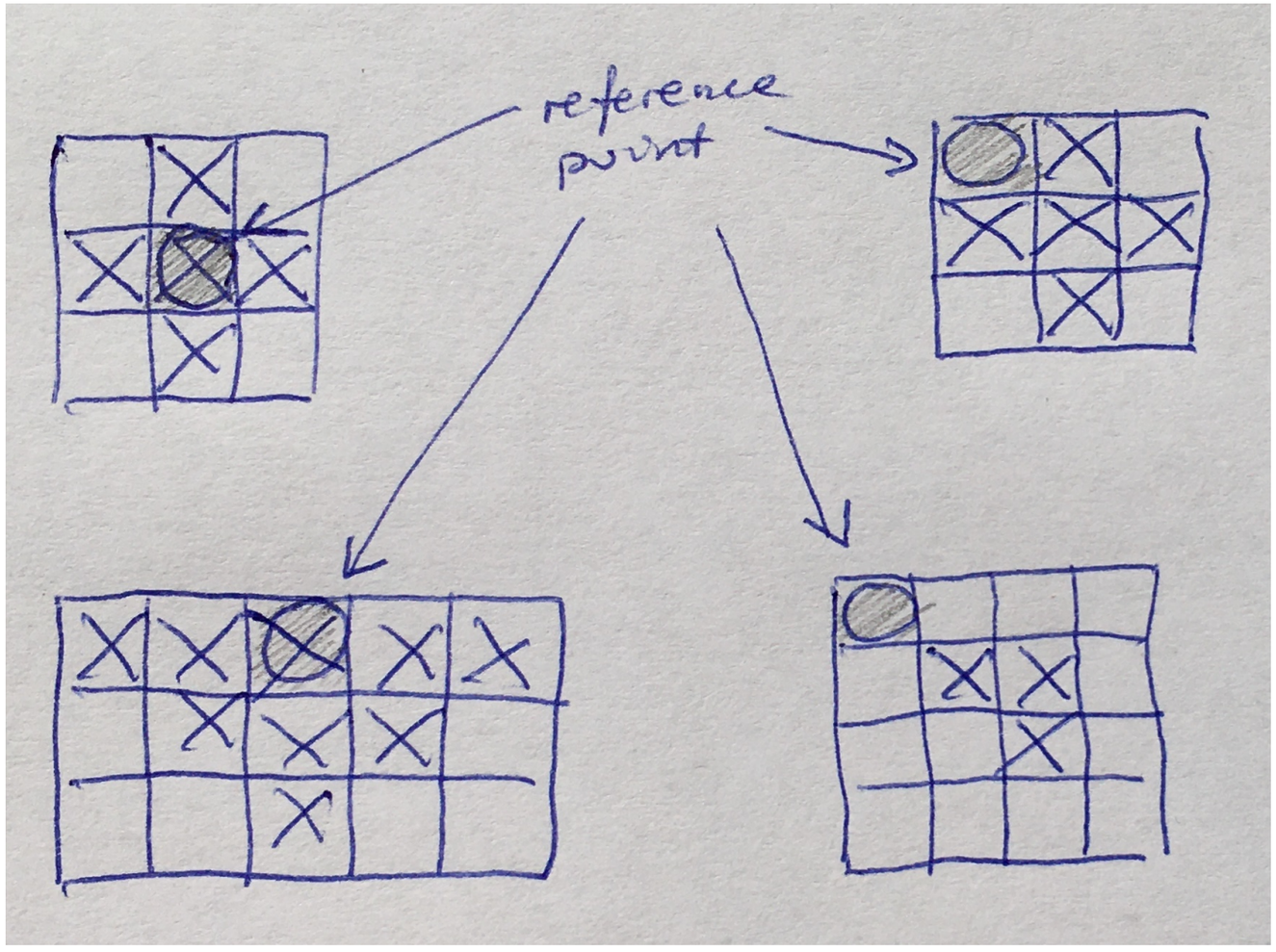
Mathematical morphology

Idea

To simplify consideration I will be talking about *binary morphology* which means morphology applied to binary images and binary structuring element.

In this case both image and structuring element contain only value 0 and 1, where 1 represents pixel in an image and 0 represents image background. To define structuring element, you have to specify:

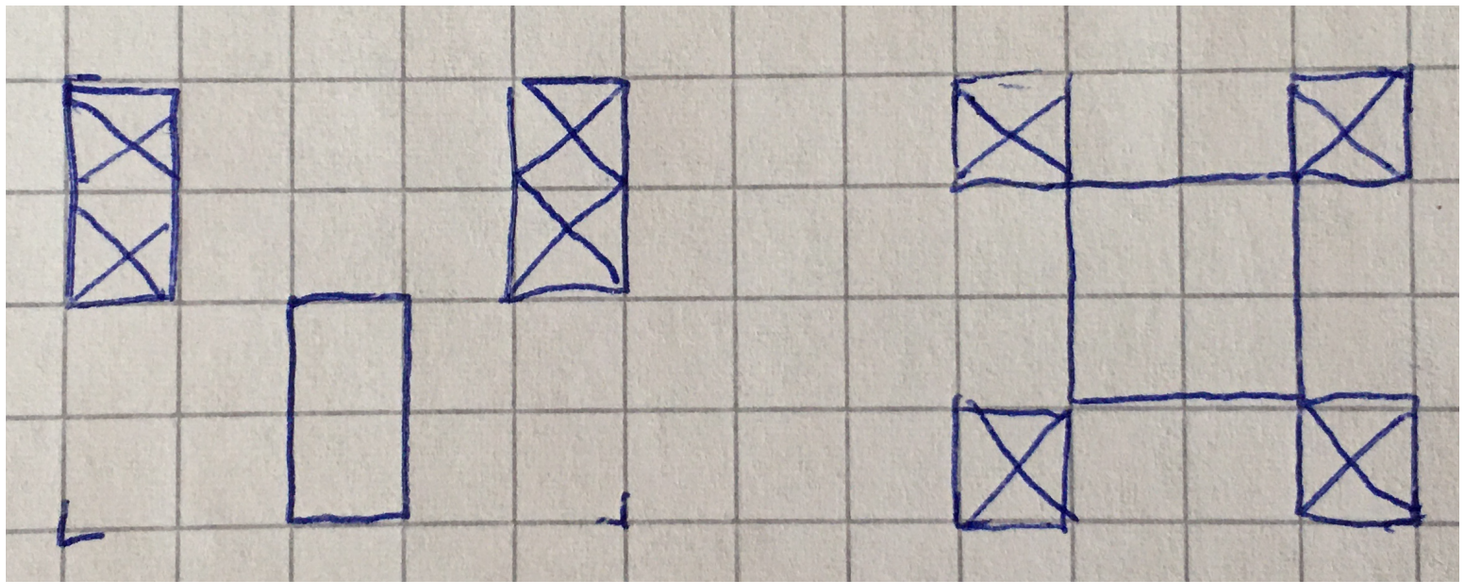
- its size (dimensions);
- structuring element B which is represented by two parts:
  - B_1 -- set of pixels of value 1,
  - B_2 -- set of pixels of value 0;
- reference point used during translation.



Mathematical morphology

Idea

Please keep in mind that B doesn't have to be rectangular. Following structuring elements are perfectly fine:



Mathematical morphology

Idea

Below I give an example of a binary image and a structuring element.

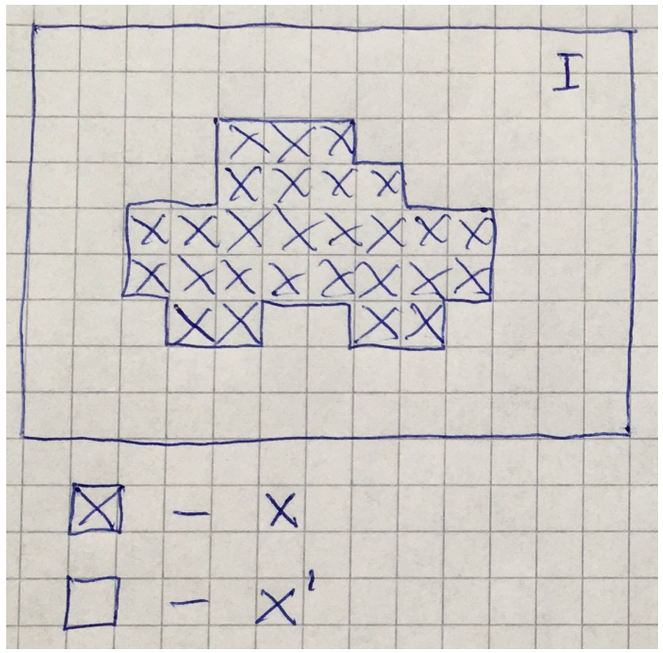
Pixels from image I are divided into those belonging to X (corresponding to image values 1) and those belonging to its complement X' (corresponding to background values 0) :

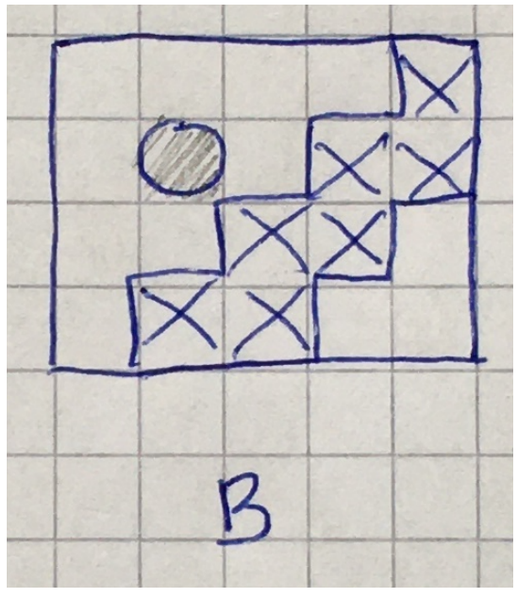
$I = X \cup X'$ and $X \cap X' = \emptyset$.

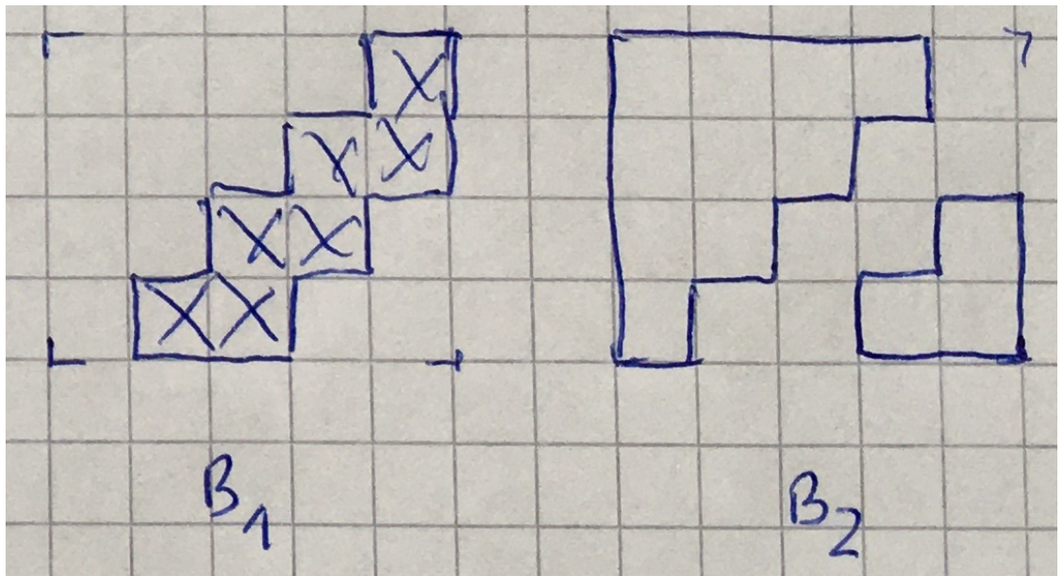
In this case I use term *image* in two different meanings: first means image as a whole set of pixels, second means part of image different than background.

There is also an example of structural element B and its decomposition into the two sets B_1 and B_2.

Each subset of B will be used to analyse the set X and its complement.







Mathematical morphology

Idea

The hit or miss transformation is defined as the point operator:

$$I \otimes B = \{x \in I : B_1(x) \subseteq X \wedge B_2(x) \subseteq X'\}.$$

In this equation, x represents element of I. The operation of B_1 on X is a *hit*; the operation of B_2 on X' is a *miss*. The index x in the structural element indicates that it is moved to the position of the element x in such a way that the reference point coincides with point x. That is quite similar to other group operators you have seen so far -- B defines a specific window that is moved through the image.

This transformation defines a process that moves the structural element B to be placed at each pixel in the image, and it performs a pixel-by-pixel comparison against the template B. If the value of the image is the same as that of the structuring element, then the image's pixel forms part of the resulting set $I \otimes B$.

The simplest form of morphological operators is defined when either B_1 or B_2 is empty.

Mathematical morphology

Operators: erosion (reduction)

When B_2 is empty the hit or miss transformation defines an *erosion* (reduction):

$$I \ominus B = \{x \in I : B_1(x) \subset X\}.$$

According to this formula pixel x belongs to the eroded set if each point of the element B_1 translated to x is on X. Since all the points in B_1 need to be in X, this operator removes the pixels at the borders of objects in the set X. Thus, it actually erodes or shrinks the set.

One of the most common applications of this is to remove noise in thresholded images.

Mathematical morphology

Operators: erosion (reduction)

[tutu examples]

Mathematical morphology

Operators: dilation (increase)

When B_1 is empty the hit or miss transformation defines an *dilation* (increase):

$$I \oplus B = \{x \in I : B_2(x) \subset X'\}.$$

According to this formula pixel x belongs to the dilated set if each point of the element B_2 translated to x is on X'.

Since all the points in B_2 need to be in X', this operator removes the pixels at the borders of the complementary set X'. Thus, it actually erodes or shrinks the the complementary set and when the it is eroded, the set X is dilated.

Mathematical morphology

Operators: dilation (increase)

[tutu examples]

Mathematical morphology

Note

- Neither dilation nor erosion specifies a required shape for the structuring element.

- By sequences of erosions and dilations other operators can be easily defined:

  - the *opening operator* is defined by an erosion followed by a dilation:

    $$I \circ B = (I \ominus B) \oplus B$$

  - the *closing operator* is defined by a dilation followed of an erosion:

    $$I \bullet B = (I \oplus B) \ominus B.$$

- Closing and opening operators are generally used as filters that remove dots characteristic of pepper noise and to smooth the surface of shapes in images. These operators are generally applied in succession, and the number of times they are applied depends on the structural element size and image structure.

- In addition to filtering, morphological operators can also be used to develop other image processing techniques. For example edges can be detected by subtracting the original image and the one obtained by an erosion or dilation.

Mathematical morphology

Gray level morphology

The *cross-section representation* uses multiple thresholds to obtain a pile of binary images. Thus, morphological operator can be applied to collection of binary images formed at each threshold level.

Mathematical morphology

Gray level morphology

The *umbra approach* considers all points (x, v) such that:

- $v < f(x)$,
- $f(x)$ is a gray level at point x,
- $x \in I$
- and $v \in [0,255]$.

Other words, in case of 2D images umbra $U(I)$ is a 3D object and it consist of all points located under the surface F given by:

$$F(I) = \{(x, z) : x \in I, z = f(x)\}.$$

In consequence:

$$U(I) = \{(x, z) : x \in I, z < f(x)\}.$$

To umbra you apply three-dimensional structuring element in a similar way as you do with to two-dimensional case. In some sense you can imagine that you play three-dimensional tetris.

Bibliography

Bibliography