Lab4: Mathematical morphology

Xabier Morales and Pritam Mishra

Course 2021-2022

Deadline June 3rd 23:59

1 Mathematical morphology

1.1 Non-morphology: linear filters

Before moving on to the morphology framework, some linear filters must be recalled, where an image is convolved with some kernel. It is useful because morphology works in a similar way, with the so-called *structuring elements* are used to erode or dilate an image.

To run linear filters in Matlab (and Octave), a kernel must be defined before calling `conv2` function. Try the following kernels, which quite clearly show what can be done with linear filters. For example, the modulus of the gradient and Laplacian detect edges, the "high-pass" makes an image sharper, etc..

```
matlab〉 Ksmooth = [1 1 1; 1 1 1; 1 1 1]
matlab〉 Kdx = [0 0 0; 0 −1 1; 0 0 0]
matlab〉 Kdy = [0 0 0; 0 −1 0; 0 1 0]
matlab〉 Kdxsobel = [−1 0 1; −2 0 2; −1 0 1]
matlab〉 Kdysobel = [−1 −2 −1; 0 0 0; 1 2 1]
matlab〉 Klaplacian = [0 −1 0; −1 4 −1; 0 −1 0]
matlab〉 Khipass = [0 −1 0; −1 5 −1; 0 −1 0]
```

Try them, for example, with image `lena.png`.

```
matlab〉 x = im2double(imread('lena.png'));
matlab〉 imshow(conv2(x,Ksmooth/9.0));
matlab〉 imshow(conv2(x,Klaplacian));
matlab〉 imshow(conv2(x,Khipass));
matlab〉 xdx = conv2(x,Kdxsobel);
matlab〉 xdy = conv2(x,Kdysobel);
matlab〉 xgrad = sqrt(xdx.^2 + xdy.^2);
matlab〉 imshow(normalize(xgrad));
matlab〉 ...
```

1.2 Morphological filters. Theory

The following table summarizes morphological operations from the theory class and provides their definitions in a unified notation. Let B be a structuring element. We will define the morphological operations on sets and on functions in a similar way, according to the following table, where u represents either a set (binary image) or a function (grayscale image).

Name	notacion	Octave
Dilation	$u \oplus B$	`imdilate`, `dilate`
Erosion	$u \ominus B$	`imerode`, `erode`
Opening	$u \circ B := (u \ominus B) \oplus B$	`imopen`
Closing	$u \bullet B := (u \oplus B) \ominus B$	`imclose`
Advanced morphological gradient	$\nabla_B^+(u) := u \oplus B - u$	
Delayed morphological gradient	$\nabla_B^-(u) := u - u \ominus B$	
Central morphological gradient	$\nabla_B^\pm(u) := u \oplus B - u \ominus B$	
Top hat	$\rho_B(u) := u - u \circ B$	
Bottom hat	$\rho_B^*(u) := u \bullet B - u$	
Iterative erosion	$u \ominus nB := u \ominus B \ominus \cdots \ominus B$	
Skeleton	$S(u) := \bigcup_{n=0}^\infty [u \ominus nB \backslash (u \ominus nB \circ B)]$	

Note that this table contains new operations that were not explained in the theory class. All of them are defined based on the erosion and dilation, which are the basic morphological operations along with thresholding.

1.3 Morphological filters. Practice

The most important morphological filter is the binarization or *thresholding*. In Matlab it might be implemented by inequality operators. From a given image `x` with gray values between 0 and 1, the following program creates a binary image containing the level set of level 0.5. In the theory notation, we have that $y = X_{0.5}x$:

```
matlab⟩ y = x > 0.5;
```

In other words, an image with high values in every pixel position where the original image `x` has values greater then the thresholding level (level 0.5).

Pay attention, that the thresholding operation does not need a structuring element. All other morphological operators require a structuring element. Here are some possible structuring elements:

```
matlab⟩ SEsquare = [1 1 1; 1 1 1; 1 1 1]
matlab⟩ SEcross = [0 1 0; 1 1 1; 0 1 0]
matlab⟩ SEhoriz = [1 1 1 1 1 1 1 1 1 1]
matlab⟩ SEvert = [1;1;1;1;1;1;1;1;1;1]
matlab⟩ SEsquare7 = ones(7,7);
matlab⟩ ...
```

Try the morphological filters with these structuring elements on the image `lena.png`, or any other image of your choice.

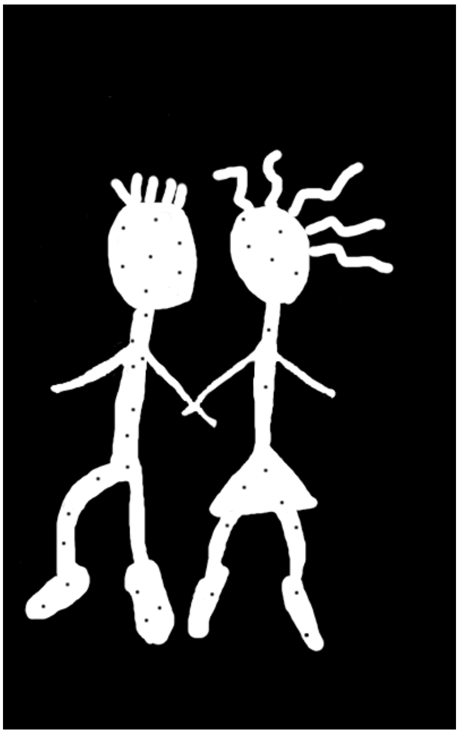
```
matlab⟩ x = im2double(imread('image.png'));
matlab⟩ se = SEsquare;
matlab⟩ imshow(imdilate(x,se));
matlab⟩ imshow(imerode(x,se));
matlab⟩ imshow(imdilate(imerode(x,se),se));
matlab⟩ imshow(imerode(imdilate(x,se),se));
matlab⟩ imshow(imdilate(x,se)-x);
matlab⟩ imshow(x-imerode(x,se));
```

```
matlab⟩ imshow(imdilate(x,se)-imerode(x,se));
matlab⟩ imshow(x-imdilate(imerode(x,se),se));
matlab⟩ imshow(imerode(imdilate(x,se),se)-x);
matlab⟩ se = ···
```

Note that in the last seven examples respectively an opening, a closing, three morphological gradients, a top-hat and bottom-hat of the image x are calculated. Try all these filters with different structuring elements, until you get the understanding of the effect of each of them.

Exercise 1.

As seen in the following Figure, Evarist and Ermessenda just started dating.

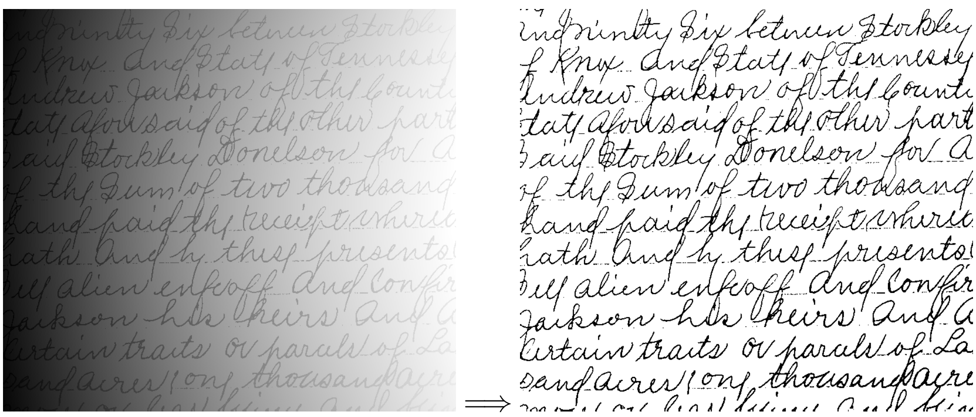


Perform the following tasks:

- Due to some random reasons, you do not want to see them holding hands. Apply morphological filters to avoid it.
- Belive it or not, someone hated even more the situation and shot them. Fortunately, this is just a picture, but both guys have holes in their bodies; use morphology to heal them.
- Get the boundaries of the holding-hands couple.

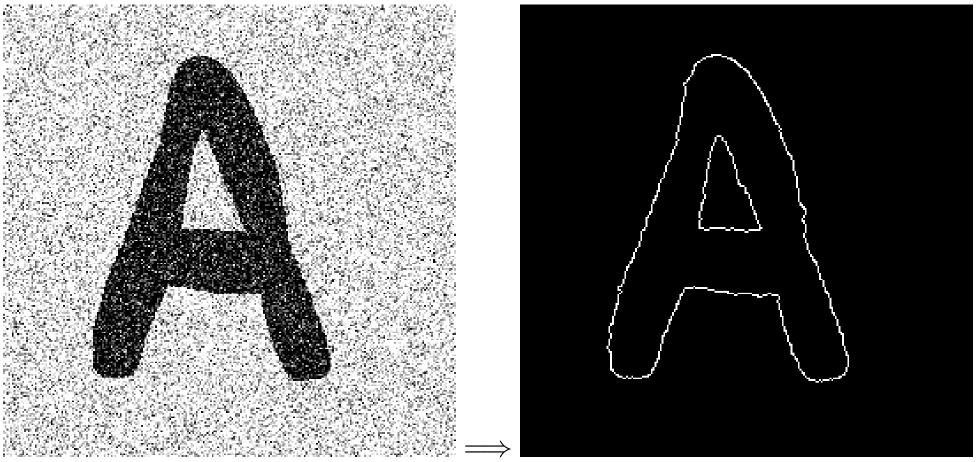
Exercise 2.

Find a proper segmentation of the image `letters.png` separating the letters from the background.



Exercise 3.

Draw three letters in a blank image with a any kind of Drawing software (Gimp or Paint work fine); introduce noise to this image with the *imnoise* built-in Matlab function. The goal of this task is to obtain, once again, the silhouette of all three letters.



Exercise 4. (Optional)

Provide examples of morphological gradients, top-hat and bottom-hat operations in both binary and grayscale images. Explain their main properties and differences between each other.