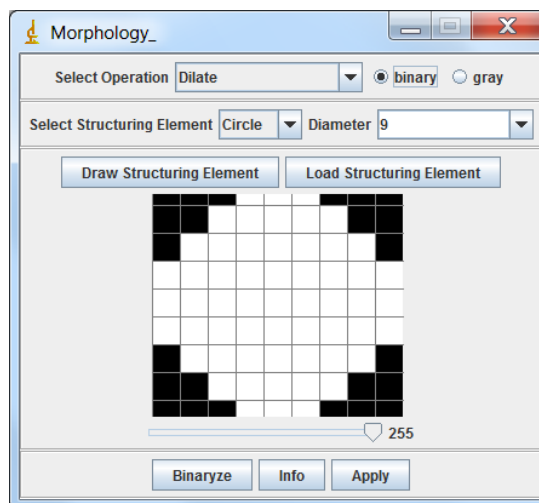
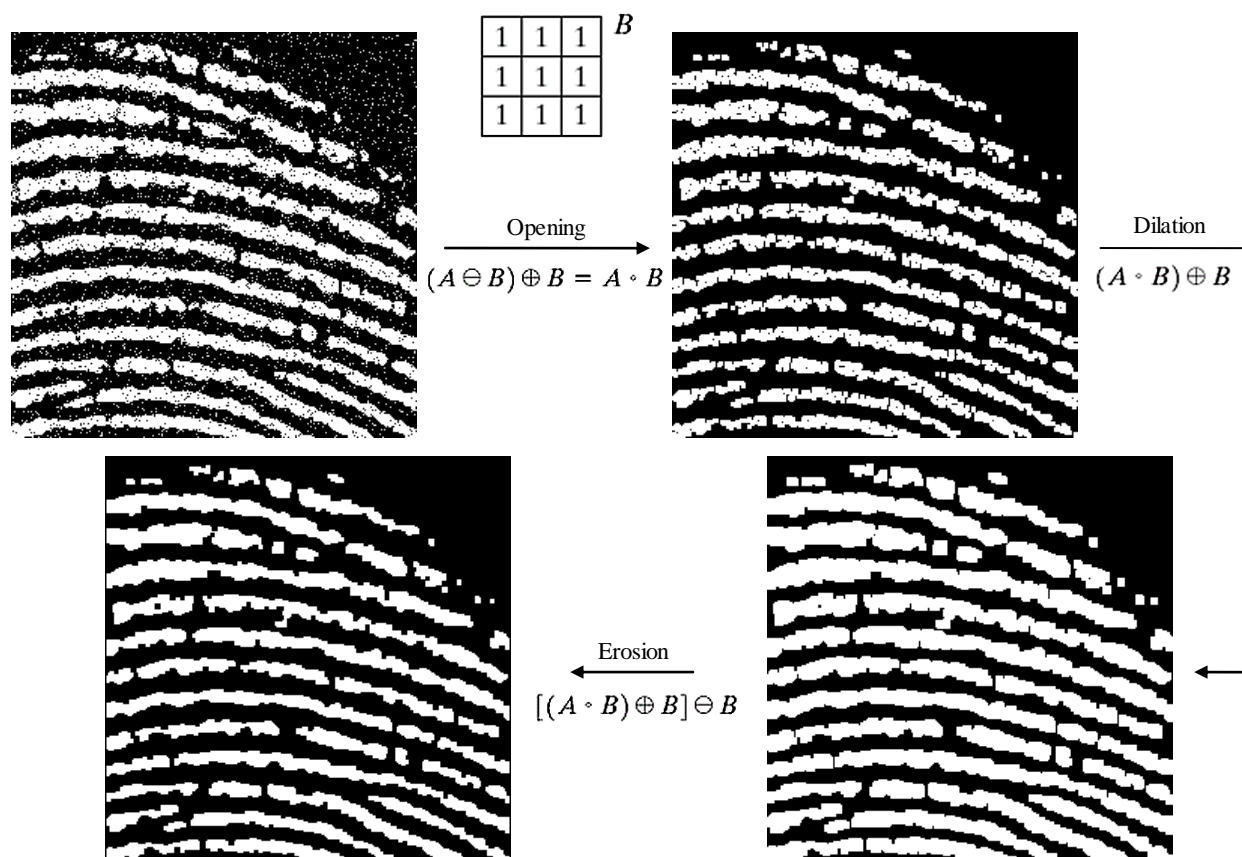


5 – Morphological operations on binary images

- Erosion, dilation, opening, closing: The four basic morphological operations together with some others can be implemented on a binary image via *Process<Binary*. However, these commands do not provide a user with a possibility to vary main parameters of the structuring element (SE) – size and shape. This is made possible using e.g. a plugin located at: <http://svg.dmi.unict.it/iplab/imagej/Plugins/Morphological%20Operators/Morphological%20Operators/MorphologicalOperatorsPlugin.htm> .
- Open image **Logos.tif** and invert it by typing **SHIFT + i** (= *Edit<Invert*), since the plugin – unlike the ImageJ's native *Process<Binary* command – works with white objects on a black background. The window that appears after selecting *Plugins<Morphology* has several options to choose from:
 - Operation type: dilation, erosion, closing, opening, boundary extraction, smoothing, hit-or-miss transform, gradient
 - Image type: binary or grayscale
 - SE shape: circle, square, cross, vertical- or horizontal line
 - SE size: 1 or more
 - Drawing or loading SE from a disk
 - Choosing SE gray level (0 – 255) when operating on grayscale images
 - Binarizing a grayscale image when working with it.
- Dilate/erode the image using various SE shapes and/or sizes.
- Morphological opening is defined as erosion followed by dilation. Prove this using a cross SE of a diameter value 3 and:
 - First, perform the image opening and
 - Second, erode the original (i.e. inverted **Logos.tif**) image and then dilate it using the same SE. Compare both images.
- Perform morphological closing. As you can see, while opening smoothes the objects (letters) and removes isolated pixels, closing smoothes the objects and fills in small holes.
- Basic morphological operations can be implemented to remove small structures – e.g. salt and pepper noise pixels – from an image without affecting larger structures. Open image **Fingerprint.tif** (top left) and perform morphological opening with a 3-by-3 square SE. The resulting image (top right) is indeed free from the background white noise pixels, but black noise pixels on the fingerprint ridges still exist. To remove these, first dilate the resulting image and finally erode the dilated image using for both operations the same 3-by-3 square SE. Note that by applying the latter two operations we actually performed morphological closing.





- Boundary extraction (Outline): Open image **Logos.tif**, invert it (**SHIFT + i**) and extract boundaries with a 3-by-3 square SE. The algorithm generates a one pixel wide outline of objects. Compare the resulting image with the one obtained using:
 - 3-by-3 horizontal line SE
 - Sobel edge detector (perform edge detection with *Process<Find Edges* on the original, non-inverted image).
- Region filling: Apply *Process<Binary<Fill Holes* on **Logos.tif** image. The algorithm fills white “holes” found in black objects, i.e., turns to black all white pixels that are completely (in a 4-connectivity sense) enclosed by black pixels – inspect closely e.g. the Reader’s Digest logo.
- Skeletonization: Command *Process<Binary<Skeletonize* repeatedly removes pixels from the edges of objects in a binary image until they are reduced to single pixel wide skeletons. See how the algorithm works on **Logos.tif** image.
- Connected components labeling: Open image **Blobs.tif** and binarize it (*Process<Binary<Make Binary*) then apply commands *Analyze<Set Measurements* and *Analyze<Analyze Particles*.