Note: Images used in this tutorial belong to the Image Processing Toolbox.

1. (a) Binarize the rice.tif image by simply thresholding it at a suitable level,
\[
\text{>> rice} = \text{imread('rice.tif');}
\]
\[
\text{>> r} = \text{rice} > 110;
\]

Use Matlab to perform morphological **boundary extraction** on the binary image of rice.tif to obtain the
i. internal boundary, \(A - (A \Theta B)\)
ii. external boundary, \((A \oplus B) - A\)

(b) What is the difference between the internal boundary and external boundary?

2. (a) Write a Matlab function for morphological **region filling** (or **hole filling**). Assume that the regions to be filled are bounded by an 8-connected boundary.

Given a pixel \(p\) within the region/hole, we wish to fill up the entire region/hole. To do this, we start with \(p\) an dilate as many times as necessary with the structuring element, each time taking an intersection with \(A^c\) before iterating. The set expression is given by:
\[
X_k = (X_{k-1} \oplus B) \cap A^c \quad k = 1,2,3,\ldots
\]

where \(B\) is a symmetric SE. The algorithm terminates at iteration step \(k\) if \(X_k = X_{k-1}\). Finally, take the set union \(X_k \cup A\) to include the boundaries.

*Hint #1: The function should take in the binary image, starting position for filling, and structuring element, while it should output the filled image.*

*Hint #2: Region/hole filling algorithms should always use a cross structuring element, to avoid crossing the boundary.*

(b) Use the boundary-extracted rice.tif image (obtained from Q1) to test your region filling algorithm.

3. (a) Write a Matlab function to extract **connected components**. The function is almost exactly the same as that for region/hole filling in Q2, except for connected components we take the intersection with image itself (as contrary to region filling which takes an intersection with the complement of the image).
Hint: Use a square structuring element to test the connected components up to its boundaries.

Experiment with this function using the binarized “rice” image (from Q1).

Try with large size SEs (11x11 and above). What do you observe?

4. **Skeletonization**

The skeleton of a binary object is a collection of lines and curves that encapsulate the size and shape of the object. There are many different methods of defining a skeleton, and also many possible different skeletons. We will obtain a skeleton of a given object by simply using Lantuéjoul’s method – a morphological method.

\[
S(A) = \bigcup_{k=0}^{K} S_k(a)
\]

where

\[
S_k(a) = (A \Theta k B) - [(A \Theta k B) \circ B]
\]

A sequence of \( k \) erosions and openings using structuring element \( B \) is repeated until \( (A \Theta k B) \circ B \) is empty (contain all zero elements). The skeleton is then obtained by taking the unions of all the set differences.

Write a function to perform skeletonization using Lantuéjoul’s method. Then, experiment with some images to test your function.

**Homework**

1. **Morphological Noise Removal**

Noise removal can be done with morphological operations, especially in removing impulse/shot noise.

Firstly, corrupt the image circles.tif with 10% impulse/shot noise.

\[
\begin{align*}
&> c = imread('circles.tif'); \\
&> x = rand(size(c)); \\
&> d1 = find(x<=0.05); \\
&> d2 = find(x>=0.95); \\
&> c(d1) = 0; \\
&> c(d2) = 1; \\
&> imshow(c)
\end{align*}
\]
In impulse/shot noise, some of the black pixels are white, some of the white pixels are black. Start of with erosion, which will remove the single black pixels, but will enlarge the holes. We can fill the holes by dilating twice. After which, we need to reduce them back to their correct size – thus, perform a final erosion.

\[((A \ominus B) \oplus B) \ominus B = (\neg A \circ B) \bullet B\]

We end up with an opening, followed by a closing! This technique is sometimes called **morphological filtering**. Try to view each step of the process to examine what happens to the image.