ECE 253a	Digital Image Processing	Handout #3	Pamela Cosman	9/29/08

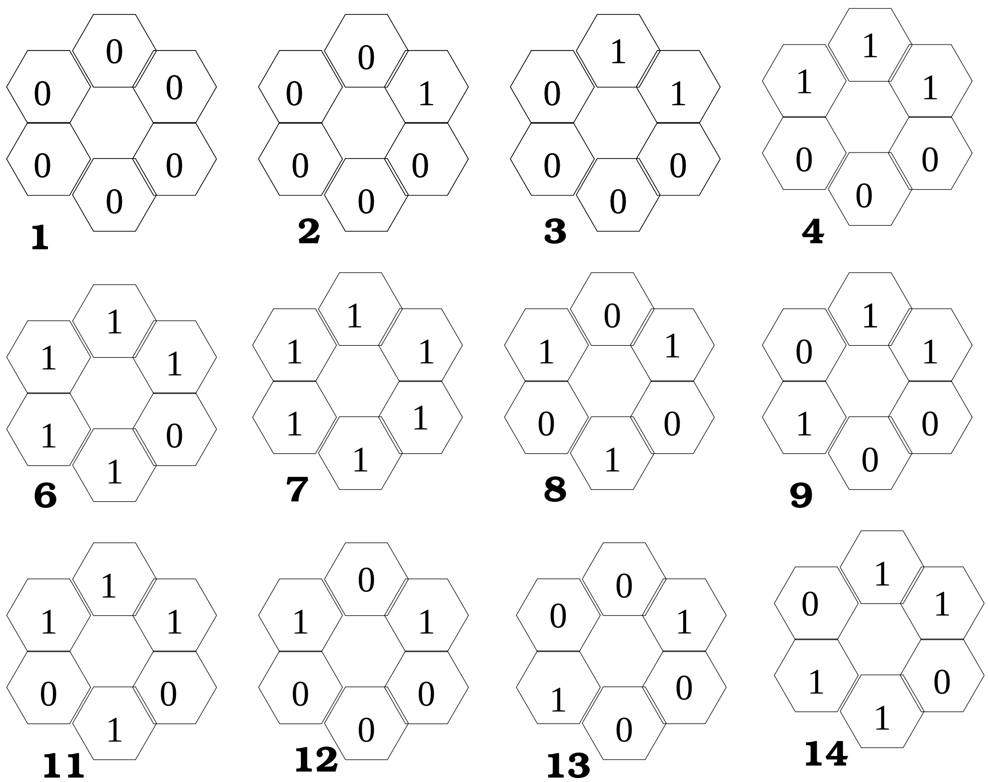
HOMEWORK 1

Due Friday, October 10, by start of class

For all the problems involving Matlab, please turn in your edited sequence of Matlab commands, and the text of any functions that you write, as well as the descriptive prose of your results.

1. **Morphological operators on a hexagonal lattice:**

   If we ignore rotations, then there are 14 possible surrounds of a cell on a hexagonal lattice, as follows:



We can uniquely define the operation of an iterative modification scheme by specifying:

- The set S of surrounds (for which $a_{ij} = 1$)
- The Boolean function $L(a, b)$ (where $c_{ij} = L(a_{ij}, b_{ij})$)
- The number of iterations, n
- The number of subfields, f, into which the image tesselation is divided

We use ab to denote a *and* b, $a+b$ to denote a *or* b, and $\bar{a}$ to denote *not* a.

Consider the following iterative modification schemes:

(1) $S=\{1\}$ $L(a,b)=b$ $n=1$ $f=1$
(2) $S=\{\}$ $L(a,b)=ab$ $n=1$ $f=1$
(3) $S=\{1\}$ $L(a,b)=ab$ $n=1$ $f=1$
(4) $S=\{2\}$ $L(a,b)=b\bar{a}$ $n=1$ $f=1$
(5) $S=\{7\}$ $L(a,b)=ab$ $n=1$ $f=1$
(6) $S=\{4\}$ $L(a,b)=ab$ $n=1$ $f=1$
(7) $S=\{3,4\}$ $L(a,b)=\bar{a}b$ $n=\infty$ $f=3$
(8) $S=\{1,2,3\}$ $L(a,b)=\bar{a}b$ $n=\infty$ $f=1$
(9) $S=\{5,6,7\}$ $L(a,b)=a+\bar{a}b$ $n=\infty$ $f=3$
(10) $S=\{8\}$ $L(a,b)=ab$ $n=1$ $f=1$
(11) $S=\{1\}$ $L(a,b)=ab$ $n=1$ $f=1$ followed by $S=\{7\}$ $L(a,b)=a\bar{b}$ $n=1$ $f=1$
(12) $S=\{7\}$ $L(a,b)=a\bar{b}$ $n=1$ $f=1$ followed by $S=\{1\}$ $L(a,b)=ab$ $n=1$ $f=1$
(13) $S=\{1,7\}$ $L(a,b)=ab$ $n=1$ $f=3$
(14) $S=\{1\}$ $L(a,b)=ab$ $n=5$ $f=1$
(15) $S=\{7\}$ $L(a,b)=ab$ $n=5$ $f=1$

Note that when we include a surround in the set S, we also include all rotated versions of that surround. For each of these schemes, find the description in words below that best fits the action it performs. Some descriptions may be used more than once, others not at all. For each answer, include some explanation.

(A) Fills in small cavities
(B) Cleans up (removes) isolated cells that are one
(C) Keeps only isolated cells that are one
(D) Removes edges of blobs, keeping the interior
(E) Removes 5 layers of edge pixels from blobs (for example, any blob whose maximum dimension is 10 pixels or less will be erased)
(F) Removes interiors of blobs, keeping only the edges
(G) Cuts off all appendages (that is, thin lines) and removes tiny blobs
(H) Mark all places where 3 thin lines come together
(I) Resets all cells to zero
(J) Marks isolated pixels of either color
(K) Removes edges of blobs, keeping only the interior, but making sure not to completely erase any blob
(L) Removes ends of lines
(M) Reproduces the input image without change (identity operator)
(N) Skeletonizes until lines are only one picture cell wide
(O) Complements all picture cells in the image
(P) Marks all corners, flushes the rest
(Q) Marks any isolated zero cell

2. **Binarization by thresholding:**

   On the web site, you will find an image named worm3.tiff. You can read it into matlab, and look at it, using the commands `c = imread('worm3.tiff','tif');` and `imshow(c)`. This is a grayscale image of a microscopic nematode (worm), known as C. elegans, and an egg. You can binarize the image by thresholding it at, say, 230, as follows: `binc = (c > 230);` and look at it: `imshow(binc)`

   Note that, with some versions of Matlab, when you manipulate images with the binary Matlab operators (such as bwmorph or applylut), Matlab puts the result into the 'uint8' format, which is not accepted as an input to some other operators (such as the `.*` command for element-by-element multiplication). So, you may need to use the "double" command to convert the variable format.

   As you can see, the binary version is not a perfectly clean representation of the worm's body. What happens if you try a slightly higher threshold? A slightly lower threshold?

3. **Dilating and eroding:**

   Let's stick with the threshold at 230. Suppose we want to clean up this binary version of the image, so that we get a perfect worm body all filled in, and one egg, and no extra pixels. We can accomplish this using 8-neighbor dilation and erosion. In Matlab, the easiest way to do this is with "bwmorph" using the "dilate" and "erode" options. Is it better to dilate first or erode first? What sequence of erosions and dilations do you need to get a "perfect result"?

4. **Skeletons:**

   We would like to find out how long this worm is. Use the bwmorph command with the "thin" option to generate a skeleton of the worm. Remember that Matlab expects object pixels to be logical value 1, so if your image is called "im" you may need to apply the skeletonization operator to the inverse of im (that is ~im) rather than im because the object of interest is the worm, not the background. Use the sum and bwlabel commands, or some other commands of your choice, to count the number of pixels in the skeleton, which tells us the length of the worm. Report the worm length and also include your skeleton picture.

5. **Spur removal, the easy way:**

   Now consider the image worm2.tiff, which also shows a nematode, but this one has an attached egg. Using the same threshold for binarization as before, and using dilation and erosion operators, get the skeleton for this worm object. Include your skeleton picture in your homework. Because of the attached egg, the skeleton has a "spur" which we would like to remove.

   Use the bwmorph command with the spur option to remove the spur. What is the smallest number of spur removal iterations that you need to do in order to remove the spur entirely?

   (Of course, now that the spur has been removed, the actual skeleton is too short. To get the skeleton to be the right length, we would, in principle, need to find the endpoints and dilate them, using the original (unpruned) skeleton as delimiter. You don't actually need to do this.)

6. **Effect of the spur:**

   Now that you have the skeleton with spur removed, does the skeleton look "correct" to you? That is, suppose the egg were perfectly erased from the original grayscale image, and suppose we then obtained a skeleton from that image. Would that skeleton be the same as this one that you obtained after spur removal? Explain. (No actual implementation for this part.)

7. **Sensitivity to the number of iterations:**

   What would have happened if you had chosen a much larger number of iterations, such as 40, for the spur removals, and then found the end points and tried to re-grow? What does this say about the sensitivity of the algorithm to this parameter? (No actual implementation for this part.)