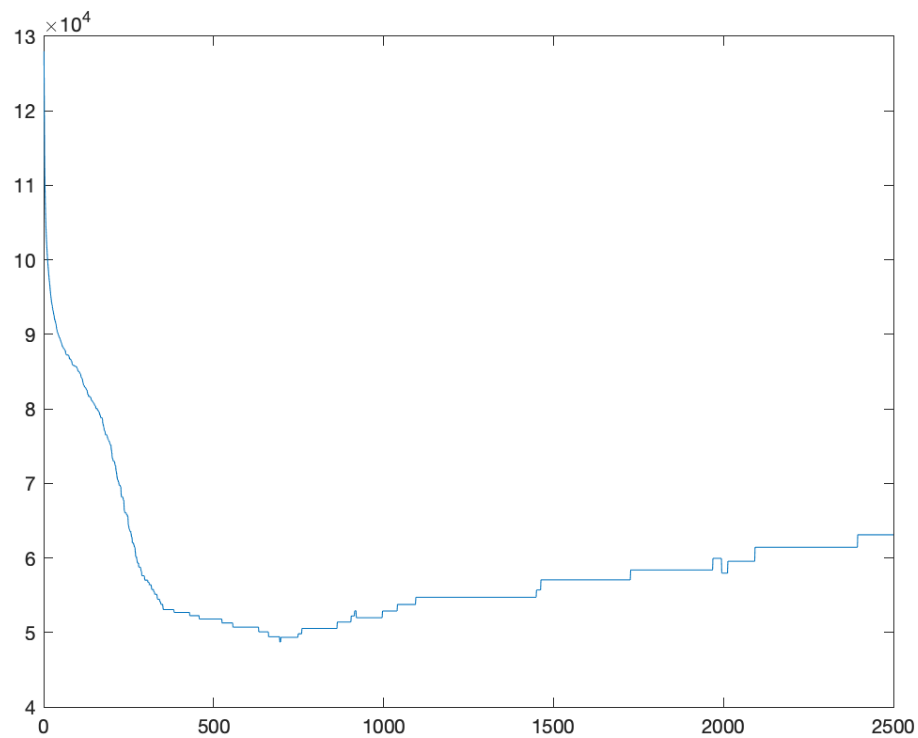
ECE 253 (Fall 2022), Homework 1 Solutions

1) **Cleaning by cluster removal:**

(a) The plot can be generated as follows:

```
x=1:2500;
y=x;
for j = 1:2500
cleanIm= 1−bwareaopen(1−badIm,j);
y(j)=sum(sum(xor(cleanIm,idealIm)));
end
plot(x,y)
```



(b) The curve initially goes down monotonically because `bwareaopen` at first is behaving exactly the way we would want it to. It removes the small clusters of black pixels that are the ”noise”. Eventually though, the value of P gets large enough that the letters in the words ”Morphological Image Processing” are getting removed along with the noise. In going from a particular value of P to $P+1$, it is possible that the curve goes down (because a noise cluster gets removed) or goes up (because a good letter gets removed).

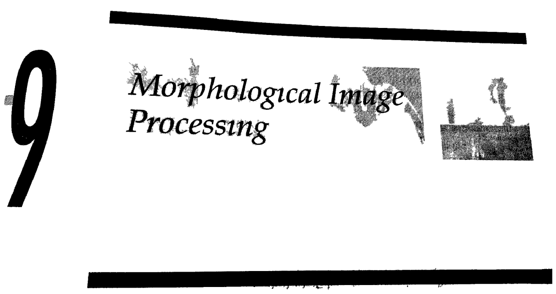
(c) We use `min(y)` to find the minimum value of y is 48,730 pixels. And the x values which correspond to that are found by:

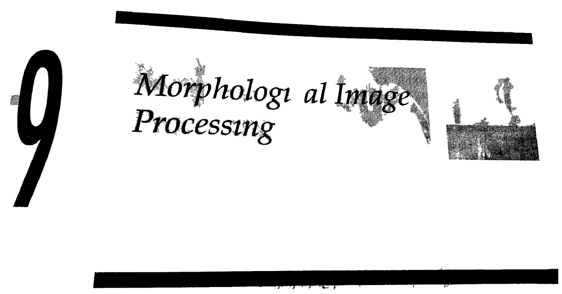
```
find(y == min(y))
```

which tells us that x=695, 696, and 697 all produce that minimum value.

Common mistake: Some students reported only one x value that corresponds to the minimum value of y. Here, the x value is not the number of iterations, but the size of the pixel cluster that we want to remove. Therefore, all three x values have to be reported.

As seen below left, at the minimum, the cluster removal process has removed lots of noise but also the dots above the letter 'i' in Morphological and in Processing. Removing the dots in each letter 'i' is not desirable but they are small so they don't affect the bad pixel count very much. However, just after the minimum (shown below right) when x=698 the cluster removal process removes the letter 'c' in Morphological. Removing that component which should not get removed has a large effect on the count of bad pixels.



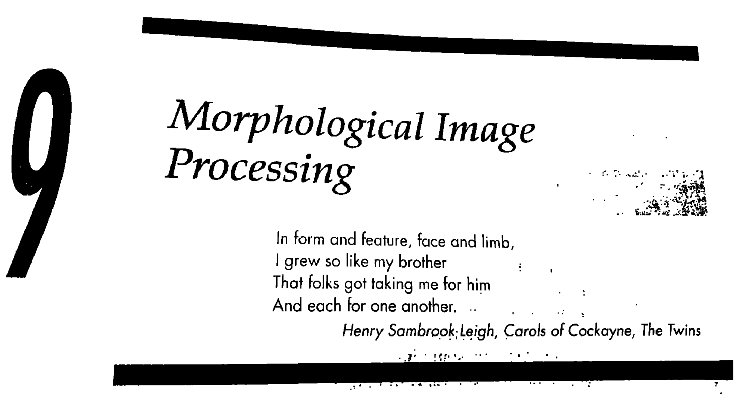


2) **Erosion and dilation:**

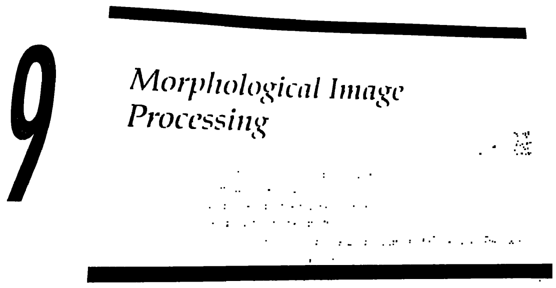
   a) Right away with simple 8-neighbor dilation and erosion, we do better than the best of the cluster removal.

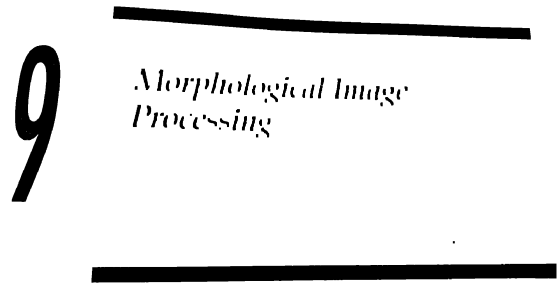
```
d3 = imdilate(badIm,ones(3));
e3 = imerode(d3,ones(3));
sum(sum(xor(e3,idealIm)))
```

   This yields a bad pixel count of 33,638. The resulting image is substantially different as well:



   Doing this closing operation using ones(5) yields an image e5 (below left) with a bad pixel count of 7,399, so that is a huge improvement. But if we keep going, doing the closing operation with ones(7), then the bad pixel count is 9505, so things got a bit worse. That image is called e7, and is shown below right.





We can see that e7 has the advantage of less noise, whereas e5 does a better job of retaining the pixels in the letters.

Common mistake: Some students did only dilation or erosion. Doing only one of them without the other will result in a great amount of bad pixels. In the closing operation, dilation adds an additional layer of pixels to the white foreground and also fills small holes. Erosion is required after the dilation to erode the unwanted additional layer away to preserve the original shape.

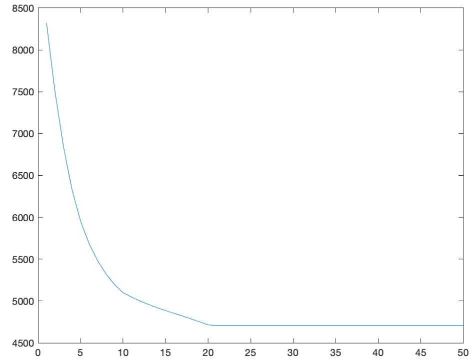
b) The geodesic reconstruction starts with e7, and then successively erodes it, taking the intersection with e5 at each step. Technically what we should be doing here is not erosion applied to the image shown, but rather dilation applied to the inverted image. But erosion of one color is generally the same as dilation of the other color (although one may encounter unwanted effects at boundaries).

```
x1 = 1:50;
y1 = x1;
temp3 = e7;
for j = 1:50,
new3 = imerode(temp3,ones(3));
temp3 = new3 | e5;
y1(j) = sum(sum(xor(temp3,idealIm)));
end
plot(x1,y1)
```

The minimum value of y1 is 4708, so we are closer to `idealIm` than before.

Common mistake 1: Using idealIm as the mask. The ideal image is provided as a reference for us to check the cleaning result. It shouldn't be used for cleaning.

Common mistake 2: Not inverting e7 when applying dilation. When applying dilation or erosion, think about what the pixels that we want to grow back. If we want to grow back the 0-valud pixels, we have to either apply erosion on the original image or dilation on the



inverted image.

Common mistake 3: Not replacing the marker image with the geodesic dilation output. The output from current iteration should be the marker image for the next iteration.

c) There are many things that can improve the bad pixel count for this particular image. For example, we can do the geodesic reconstruction a single time with a large structuring element, rather than doing it repeatedly with a tiny structuring element.
`sum(sum(xor((imerode(e7,ones(49))|e5),idealIm)))`
produced a bad pixel count of 4307.

3) **Skeletonization**

a) The skeletonization is just one line:
`skelIm = bwmorph(1-idealIm, 'skel','inf');`
which we can show like this using `imshow(1-skelIm)`

There are lots of little spurs in the skeleton because of corners and other features.

b) In class, a method was discussed to remove spurs that was roughly like this: For each connected component within the skeleton image, use the thinning operator for a certain number of times, N, then find the endpoints of what remains, and dilate it back, taking the intersection with the original skeleton at the end.

That general approach might have trouble with this image, because the big components (the long lines at the top and bottom, as well as the number 9 on the left) are much larger than the letters that make up the words ”Morphological Image Processing”. So if you take a value of N that is large enough to remove the spurs for the big number 9, it will also wipe out some of the small letters. One possibility is to make the value of N relate to the size of the component, so bigger components can remove longer spurs.

4) **Pencil and Paper Problem: Erosion and Dilation**

Note: This problem was badly worded, because the problem statement should not have said that S_2 has zeros inside the ring; it should have just said that there's a ring of object pixels (1-valued). This problem is not a hit-or-miss transform, trying to match *both* zeros and ones. Most people got the problem right, just ignoring the zeros. Because the problem statement was badly worded about S_2, we did not take off points for incorrect statements about S_2.

The easiest way to solve this problem is to eliminate possible answers. For simplicity, let's think of the little squares as 3x3. Let's start with choice (a), and consider what happens with an isolated white square located in the outer black region.

The operation $(A \ominus S_1)$ will erode the white square to a single point at its center. In the meantime, A^c in that region has a white ring surrounding a 3x3 section of black pixels, so S_2 finds a hit there, and that center point is part of the erosion $A^c \ominus S_2$. But in the end, there is an erosion by S_1 and so the center point goes away. So we can eliminate choice (a).

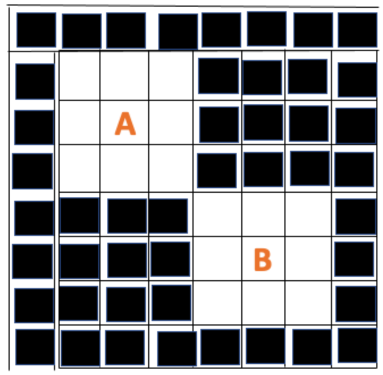
Next let's consider choice (d) because the beginning part is identical. $(A \ominus S_1)$ erodes the white square to a single point at its center. Likewise, that center point is part of the erosion $A^c \ominus S_2$, so it's part of the intersection. The final step is to take the intersection with A, so the center point remains, but the rest of the little white square is gone, so choice (d) doesn't work either.

Still just considering the isolated white square, choice (b) seems like a possibility, because the final step, dilation by S_2, means the isolated white square regrows. Choice (c) also doesn't get ruled out by considering an isolated white square.

So we can consider instead what happens with two white squares that touch. We can see in the picture that they disappear. So let's see between options (b) and (c) if one of them makes it disappear.

The operation $A \ominus S_1$ will retain only pixels A and B out of the touching white squares:

Meanwhile the operation $A^c \ominus S_2$ will erode that to nothing. So the composite operation $[(A \ominus S_1) \cap (A^c \ominus S_2)] \oplus S_1$ will have nothing there, which is correct.



In contrast, for choice (c) the operation $A \oplus S_1$ will grow the touching white squares, and $A^c \oplus S_2$ will also grow into the region, so that the final intersection with set A will not wipe out the touching white squares.

So by process of elimination, the answer is (b).

5) **Pencil and Paper Problem: Skeletons**

From the left hand side, the barbell and the line segment and the rectangle with rounded ends all skeletonize down to the line segment on the right.

- For the rectangle with rounded ends, you can picture a ball of exactly the right size to fit into the rounded ends, and then slide that ball across to the other side.
- For the line segment on the length, the definition allows balls to have radius zero, so again you can think of the ball sliding along the line segment.
- For the barbell, the center part would accommodate a small ball sliding along, and the larger elements at the ends would accommodate a larger ball, and you can picture a series of intermediate balls (getting smaller from that largest one) that successively poke farther into the central rod portion.

The disk skeletonizes down to a single point; there is only one maximal ball needed to encompass the whole thing.

The square skeletonizes down to the X-shape. You can picture a single large ball in the center (that just fills the square) and successively smaller balls that march towards the four corners.

The three touching disks has no match on the right. The correct skeleton would have three isolated points, one corresponding to each of the three disks. The equilateral triangle is not the correct skeleton because it is not possible to form the edges without defining some maximal balls that would be contained by the largest maximal ball for each disk. The skeleton does not include the isolated points where each disk touch each other either, because those zero-radius balls are also a member of the largest maximal ball for each disk.

The rectangle skeletonizes down to the sideways stick figure– we can picture this case as similar to the square.

The ring has a skeleton which is a circle; one can picture a ball of the right diameter sliding around the centerline of the ring.