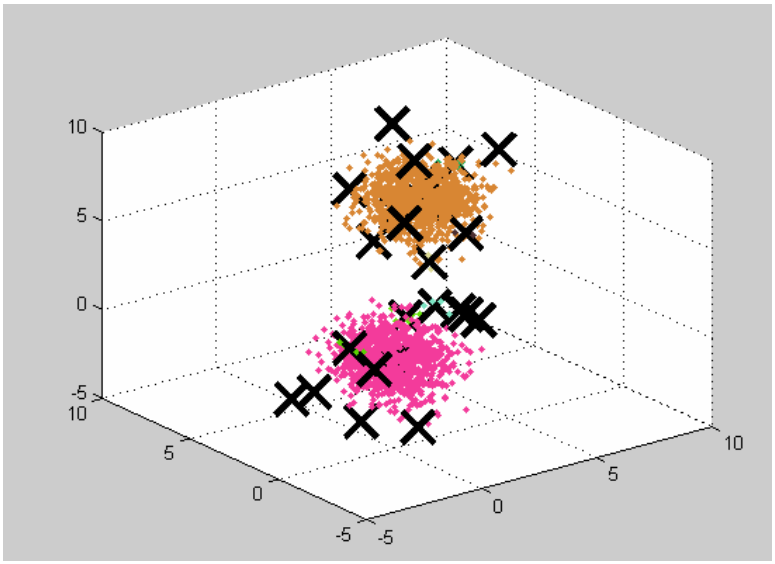
Computer Vision
Homework #2

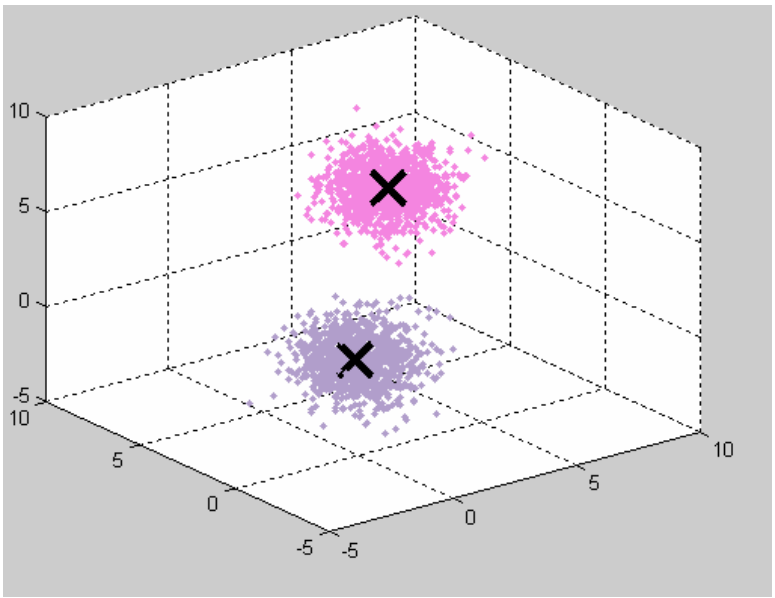
Mohamed Eldawy (eldawy@cs.wisc.edu)

Clustering 3D Points

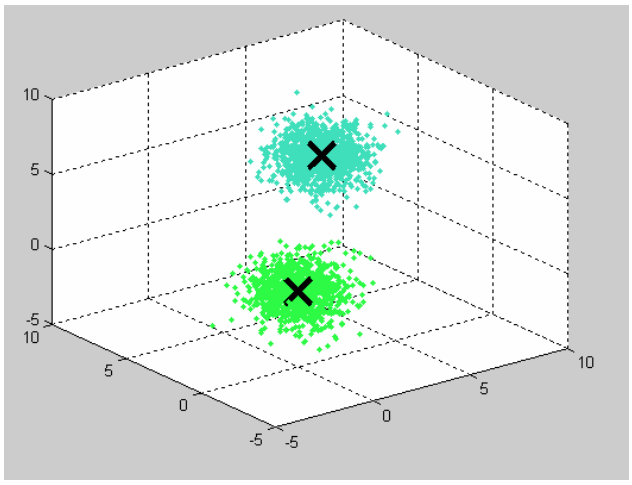
With optimizations
For r = 1



For r = 2

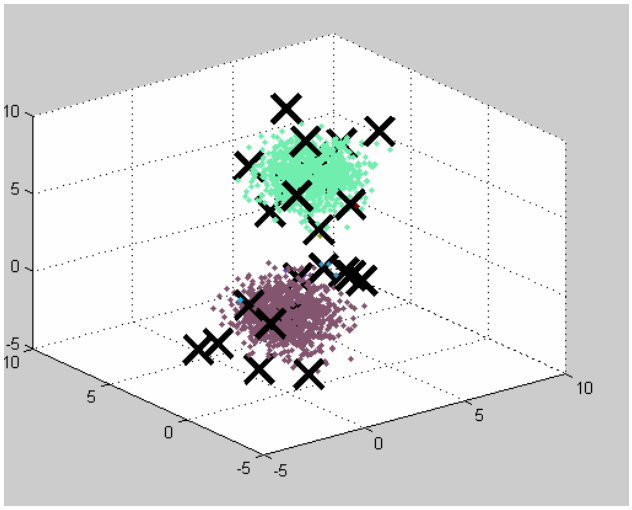


For r = 4

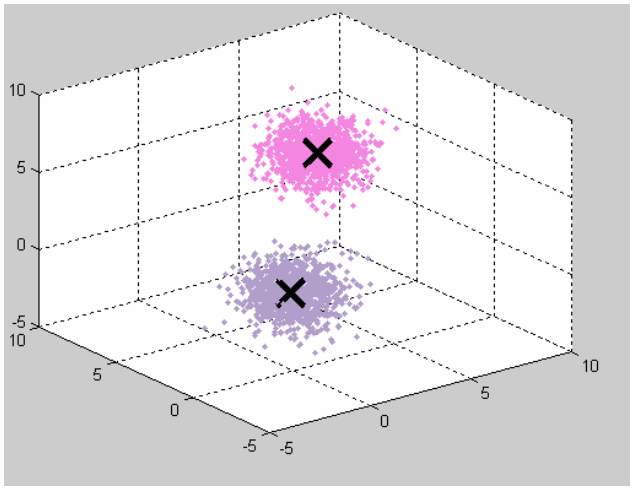


The results without optimizations

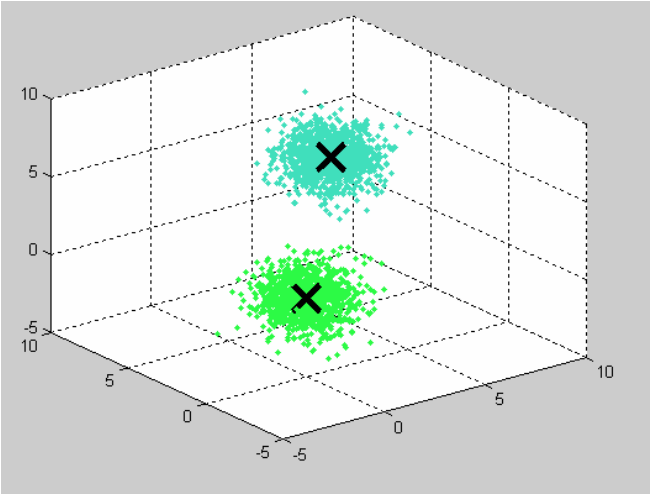
For r = 1



For r = 2



For r = 4



Mean Shift Clustering

The images I chose to work with are the following…

The 5 images represent a sunset scene, a road scene, a keyboard, a real face, and an artistic face.

(a) using only LUV color values for the feature vector

Using only LUV color values as feature vectors, and r = 5, the results are…

Number of regions in this image = 135 regions

Number of regions in this image = 138 regions

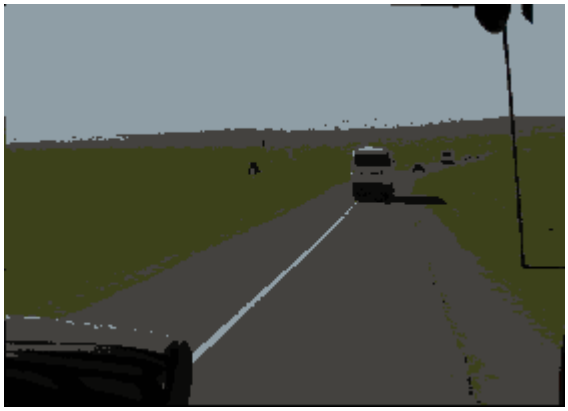
Number of regions in this image = 74 regions

Number of regions in this image = 158 regions

Number of regions in this image = 1 region (Yes! It is all white!)

Using only LUV color values as feature vectors, and r = 10, the results are…

Number of regions in this image = 10 regions



Number of regions in this image = 11 regions

Number of regions in this image = 11 regions

Number of regions in this image = 10 regions

Number of regions in this image = 1 region (Again, yes, it is all white!)

(b) Using both position values and LUV color values for the feature vector

Using both position and LUV values for the feature vector and r = 5, the results were the following

Number of regions in this image = 3859 regions

Number of regions in this image = 3956 regions

Number of regions in this image = 6343 regions

Number of regions in this image = 3908 regions

Number of regions in this image = 341 regions

Now, using both position and LUV values for the feature vector and r = 10, the results were the following

Number of regions in this image = 795 regions

Number of regions in this image = 606

Number of regions in this image = 1176 regions

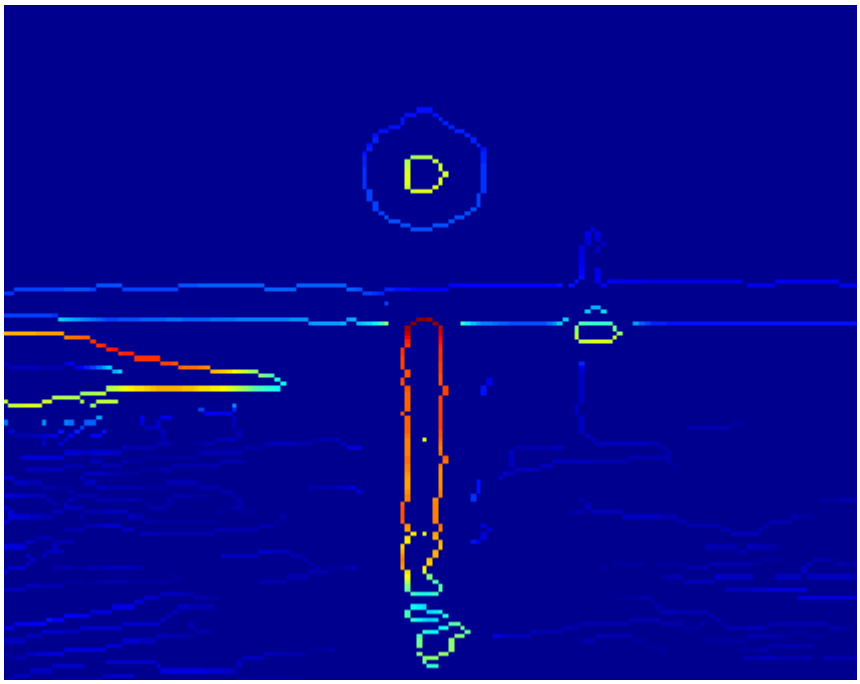
Number of regions in this image = 614 regions

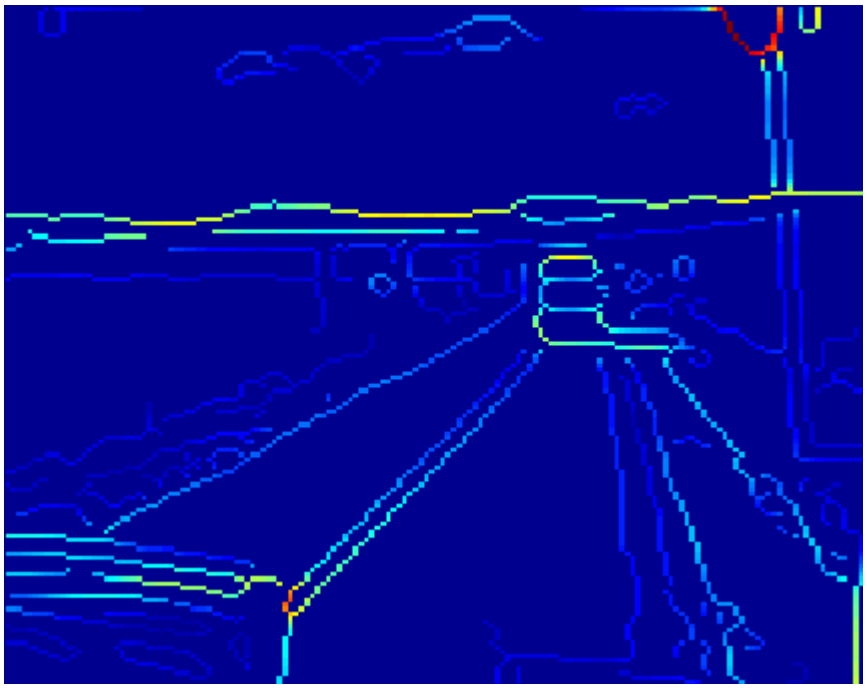
Number of regions in this image = 165 regions

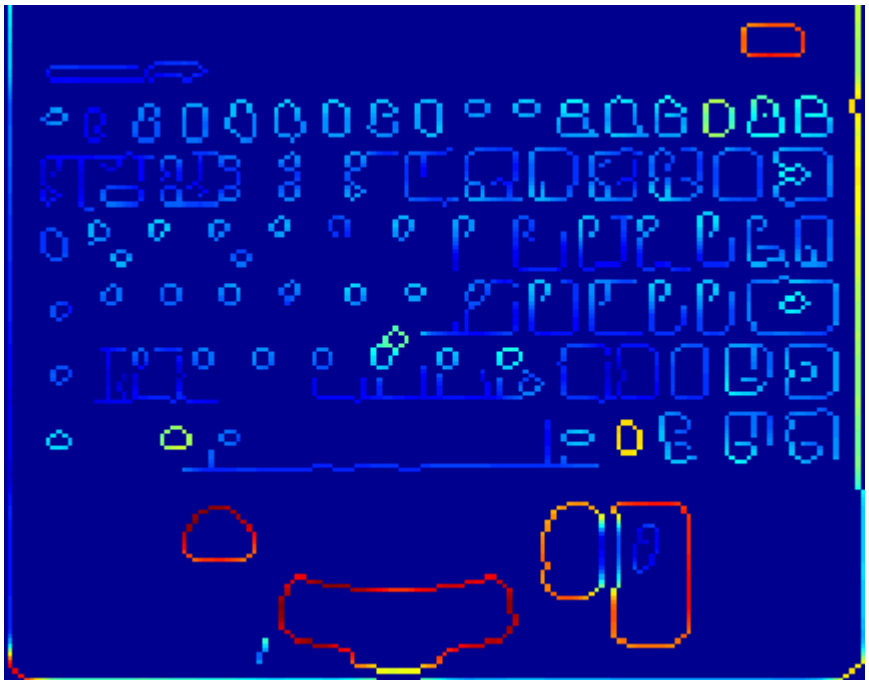
Normalized Cut Results

Below, I present the results of using the normalized cut method to segment the same 5 images. For reach image, I show the detected edges, and the resulting segments. In all cases, I used 5 segments s the number of segments for the algorithm to find. I experimented with other values, but I am only including those for lack of space.

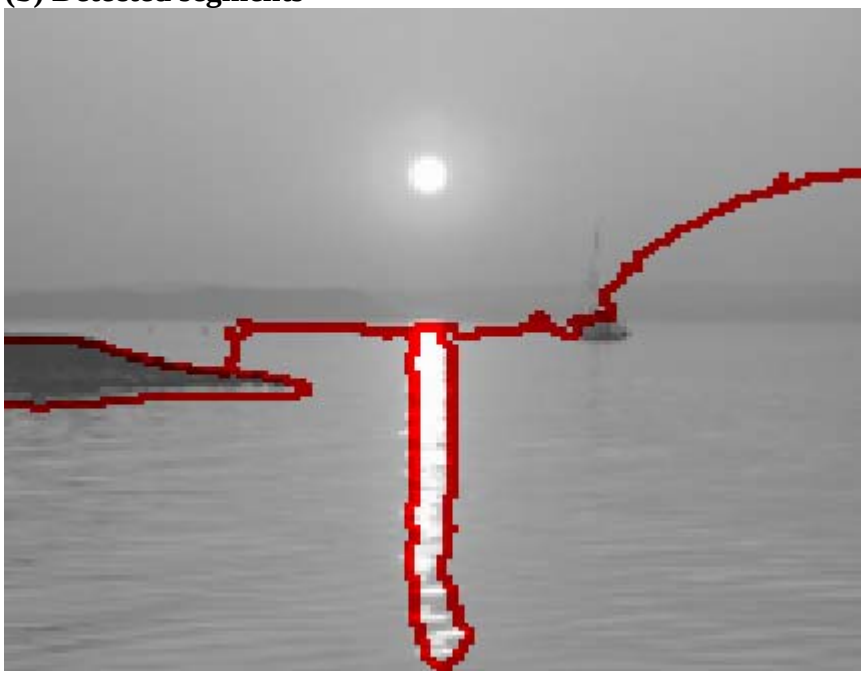
(a) Detected edges:







(b) Detected segments



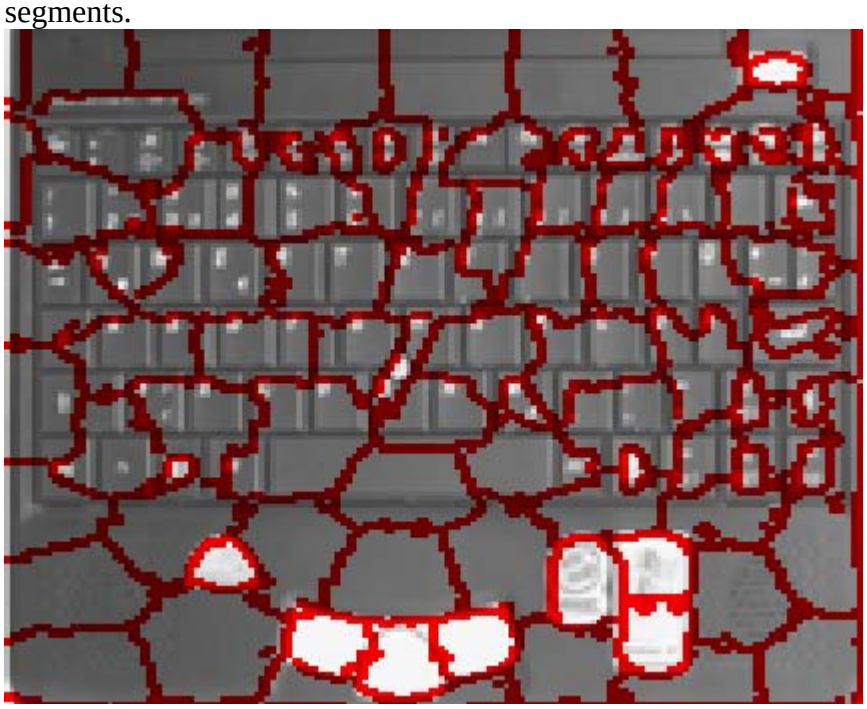
Comparison of normalized cuts to mean shift clustering

Normalized cuts algorithm produced better results than mean shift clustering with only color values as feature vector for the artistic face (Mean shift clustering failed horribly on that image, discovering only a white region!).

Mean shift clustering with positional features almost always beats normalized cuts. Even though mean shift clustering itself fails miserably sometimes (the keyboard example), normalized cuts still successfully found a way to do worse.

Both methods fail on certain types of images. The keyboard example was never segmented correctly using any method. I think normalized cuts fail way more often than mean shift clustering.

Normalized cut algorithm can produce superior results to mean shift clustering if we know the exact number of segments we are looking for. This is because, in mean shift clustering, even if we know the number of segments, we will still have to play around with r to figure out the value of r that finds the best results. This is not the case for normalized cuts.
I experimented with different numbers of segments with some of the examples above, but couldn't put my hands on a certain value that will make the results be truly correct.
For example, here are the results of the normalized cuts on the keyboard example with 120 segments.



The results are still far from being optimal. The rest pads are segmented way too often, whilst several keys on the keyboard are combined into the same region.

Answers to written questions

What effect does varying r seem to have on the resulting segmentation?
Increasing r causes larger regions to be discovered (and therefore, the algorithm will result in fewer regions overall).

What effect does varying feature vector seem to have on the resulting segmentation?
When the feature vector contains only color values, regions can be very spread out. For example, in the sunset example, the whole sea and sky were merged into one region even though there is a barrier between them in the image.
When the feature vector contains also position information, areas didn't become as huge. As a result more areas are discovered by the algorithm when the positions were added to the feature vector.

What are the advantages and disadvantages of each feature vector?

First, using only color values has the advantage that the algorithm runs faster (it has to deal with less data).
Including position values has the advantage that it produces acceptable results even when the colors are very close (like the artistic face)
Using only color values can find good results when a certain feature is occluded by another feature. For example, if we have a blue piece of paper with a ruler on top of it. Using position and color will detect each side of the paper as a separate region. Using only color will find the whole paper as a single region.

Can you suggest any extensions that might avoid many of the situations where single regions are over-segmented into multiple regions?
I think blurring the image slightly before applying the clustering algorithm should help.

The second idea is that if an area is totally enclosed in another area, we can treat the smaller area as a secondary feature. The main feature would be just the bigger area with the smaller area enclosed. This should work for cases like, for example, the keyboard example. Each letter is enclosed within the key. So, it will be treated as one unit (the key) with a secondary feature of the letter inside it. This can also happen in the sunset example (the rock on the left)

Also, after segmenting, we can try treating close enough areas as one area, and evaluate the new shapes of the regions. If the new regions (combinations of old regions) form a more uniform shape than the original regions, they can be combined. This should work in the case of the keyboard again (in some cases, each key was detected as two separate regions, one dark and one light).

The last two ideas can be combined together. For example, if two regions combined do enclose another region, we can remove the inner region and combine all three regions into one.