Answers to questions in Lab 3: Image segmentation

Name: Timotheos Souroulla   Program: Systems, Control and Robotics

Instructions: Complete the lab according to the instructions in the notes and respond to the questions stated below. Keep the answers short and focus on what is essential. Illustrate with figures only when explicitly requested.
Good luck!

Question 1: How did you initialize the clustering process and why do you believe this was a good method of doing it?

Answers:

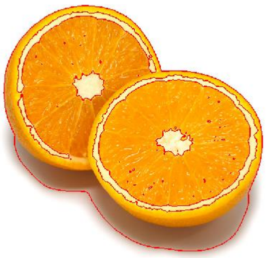
I initialized the cluster centroids by picking random pixels of the image and assigning to the centroids the color values of these pixels. I believe that this is a good method of initializing the centroids because we ensure that the specific color belongs to at least one pixel of the image.
If we initialize the cluster centroid with random color value, then the algorithm may not have optimal results because this random color may not be included in the image.
The best method to initialize the cluster centroids is to select the most common color of the image. Consequently, the algorithm will converge with less repetitions than the other methods.

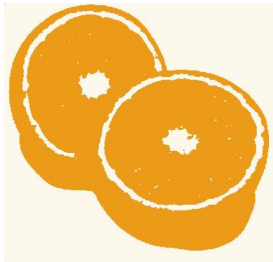
Question 2: How many iterations L do you typically need to reach convergence, that is the point where no additional iterations will affect the end results?

Answers:

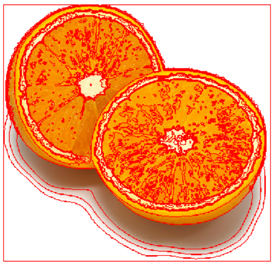
Depends on the number of clusters K. For a low value of K, a low value of L is needed and a high value when number of clusters is bigger.
For two clusters ($K = 2$), 5 iterations are enough to have almost optimal results, but for 10 clusters ($K = 10$), 15 iterations seem to be enough.
It is however, difficult to determine any exact number of iterations needed since we initialize the clusters randomly, which leads to different number of iterations needed until the convergence of the algorithm.
The number of iterations to reach convergence also depends on the Gaussian blur, since a blurrier image means less colors are presented. So, as a result, less iterations will be needed to reach convergence in comparison with a non-blurred image.

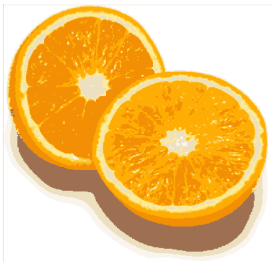
K = 2





K = 15

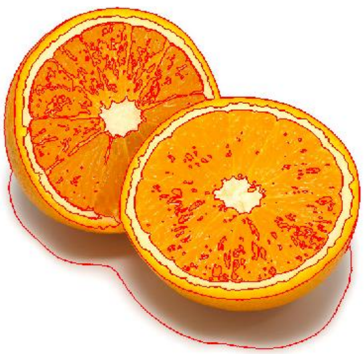


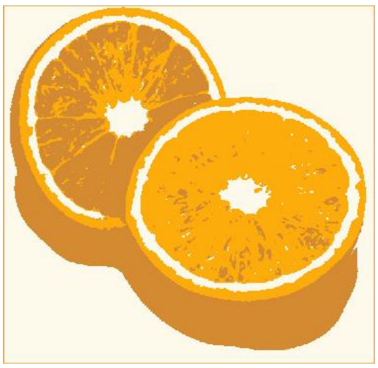


Question 3: What is the minimum value for K that you can use and still get no superpixel that covers parts from both halves of the orange? Illustrate with a figure.

Answers:

Minimum value for K that no super pixel covers part of both halves of the orange is 3. With 3 clusters, the image will be split to background and the 2 orange halves. The shade of the orange will be assigned to one of the halves cluster as seen from the figure below.





Question 4: What needs to be changed in the parameters to get suitable superpixels for the tiger images as well?

Answers:

Tiger1:

Number of Clusters: 4

For the tiger1 image, we need 4 clusters minimum in order to distinguish the tiger from the background. 1 cluster will be assigned to “rock” pixels, 1 to tree pixels, 1 to the walls and the last one to the tiger. If we increase the image_sigma factor, we will get better results.

Tiger2:

The minimum number of clusters required in order to distinguish the tiger from its background is again 4. The image may be more complicated than the previous image, but the minimum number of clusters required is the same. One cluster is assigned to snow, one the dark trees, one to bright green trees and the last one to the tiger. Since we need more detail to split the picture into clusters, the scale_factor should be increased and the image_sigma should remain the same.

Number of Clusters: 4

Tiger3:

Minimum number of clusters required for this image is 3. One for the white fence, one for the grass and one for the tiger. Some parts of the tiger are assigned to the other clusters, but we can still distinguish the tiger from the background items. We could get better results if we increase the scale_factor but increasing the image_sigma will not have positive effect on the results.

Number of Clusters: 3

Question 5: How do the results change depending on the bandwidths? What settings did you prefer for the different images? Illustrate with an example image with the parameter that you think are suitable for that image.

Answers:

<u>The spatial bandwidth:</u>

A high spatial bandwidth leads to larger unicolor areas. As we increase the spatial bandwidth, the number of modes will decrease. This is happening because more pixels are included in the mean calculation, since our interest region is larger. Since number of modes become larger when spatial bandwidth increases, we can say that the bandwidth affects the density function in our five-dimensional space.

If we have a small bandwidth, then more pointy peaks and more accurate assignment of pixels to modes.
If we have a larger bandwidth, then flattened Gaussian, which means neighboring pixels will be blended together, and also it is making it easier for assignment of pixels to modes that are more spread-out.

<u>The color bandwidth:</u>

A high color bandwidth means the image will be smoothened to a higher degree. The bandwidth determines the radius for the color space.

Low Spatial-Bandwidth:

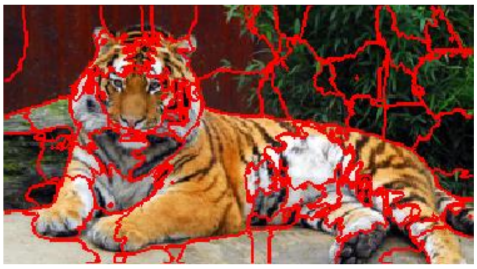
Spatial Bandwidth: 5.000000

High Spatial-Bandwidth:

Spatial Bandwidth: 15.000000

Low Color-Bandwidth:

Color Bandwidth: 5.000000



High Color-Bandwidth:

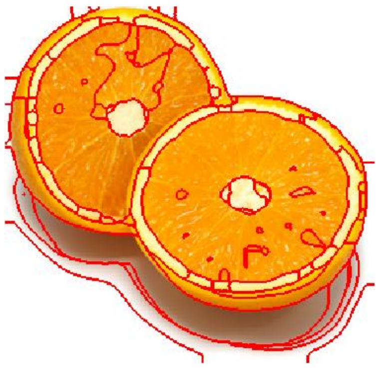
Color Bandwidth: 15.000000

<u>Preferable values for each image:</u>

Oranges:

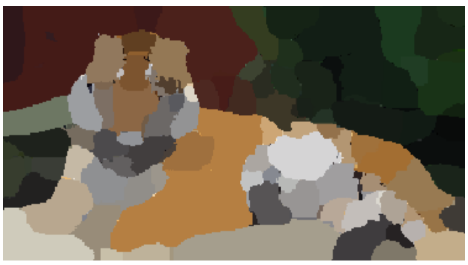
Color Bandwidth: 6.000000

Spatial Bandwidth: 4.000000



Tiger1:

Color Bandwidth: 10.000000



Spatial Bandwidth: 7.000000

Tiger2:

Color Bandwidth: 8.000000

Spatial Bandwidth: 8.000000

Tiger3:

Color Bandwidth: 5.000000

Spatial Bandwidth: 9.000000

Question 6: What kind of similarities and differences do you see between K-means and mean-shift segmentation?

Answers:

<u>Differences:</u>

- K-means considers only color information ([r g b]) for each pixel, while mean-shift takes into account position and color information ([r g b x y]).
- For K-means algorithm, a pre-specified number of clusters is required, whereas the mean-shift algorithm will find several modes but needs pre-specified bandwidth.
- K-means has a high sensitivity to outliers, whereas mean-shift is less affected by them.

<u>Similarities:</u>

- Both methods are iterative. K-means will update its cluster center, according to the mean color while mean-shift will update its position, in respect to the maximum local density location.
- Both methods are used for segmenting images, treating color and pixels as samples from a probability distribution.

Question 7: Does the ideal parameter setting vary depending on the images? If you look at the images, can you see a reason why the ideal settings might differ? Illustrate with an example image with the parameters you prefer for that image.

Answers:

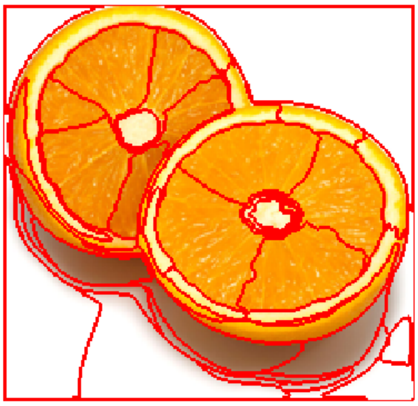
The ideal parameter settings differ for different images, since each image consists different color patterns and regions, so the parameters must be set differently for each image. In addition, each image has different complexity and number of different colors, so different parameters should be used.

Example:

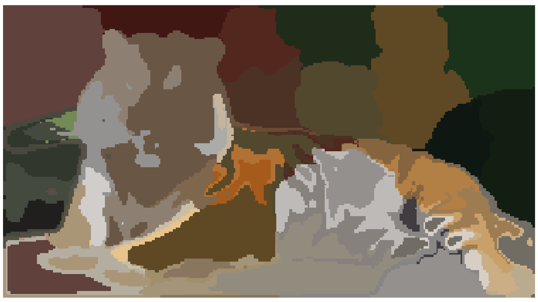
Parameters used:

- colour_bandwidth = 20
- radius = 15
- ncuts_thresh = 0.5
- min_area = 60
- max_depth = 6

<u>Orange.jpg:</u>



<u>Tiger1.jpg:</u>

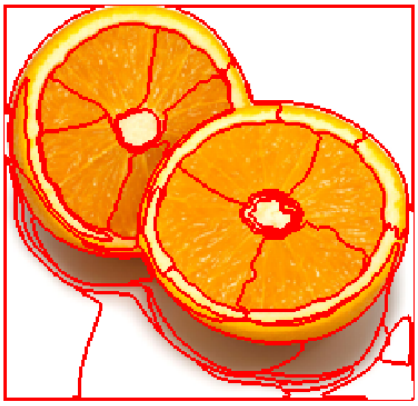


It can be clearly seen that these two image results, when using the same parameters for normalized cuts method, do not have the same accuracy. In the tiger example, some regions are not perfectly segmented, but in the orange image, the results are very good. So, as a conclusion, for each image different parameter values must be used in order to get optimal results.

Best results:

Orange.jpg:

- colour_bandwidth = 20
- radius = 15
- ncuts_thresh = 0.5
- min_area = 60
- max_depth = 6




Tiger1.jpg:

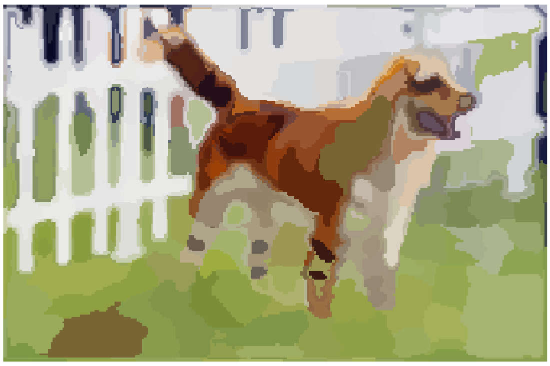
- colour_bandwidth = 20
- radius = 15
- ncuts_thresh = 0.5
- min_area = 60
- max_depth = 10


Tiger2.jpg:

- colour_bandwidth = 15
- radius = 10
- ncuts_thresh = 0.5
- min_area = 18
- max_depth = 6

Tiger3.jpg:

- colour_bandwidth = 18
- radius = 10
- ncuts_thresh = 0.5
- min_area = 10
- max_depth = 10



Question 8: Which parameter(s) was most effective for reducing the subdivision and still result in a satisfactory segmentation?

Answers:

The most effective parameters used in order to reduce the subdivision and still get satisfactory results of segmentation are the max_depth, ncut_thresh and the min_area. The other parameters, colour_bandwidth and radius, also had effects on the resulted image, but these three parameters were the most effective.

Question 9: Why does Normalized Cut prefer cuts of approximately equal size? Does this happen in practice?

Answers:

This could be easily explained by using some calculations and some mathematical equations as shown below:

$$Ncut = \frac{\mathrm{cut(A, B)}}{\mathrm{assoc(A, V)}} + \frac{\mathrm{cut(A, B)}}{\mathrm{assoc(B, V)}} \quad (1)$$

Since both assoc(A, V) and assoc(B,V) include those edges whose one vertex is from the opposite subset of vertices, the following equation should hold:

$$assoc(V) = assoc(A, V) + assoc(B, V) - cut(A, b) \quad (2)$$

By combining (1) and (2) we get:

$$Ncut = \frac{\mathrm{cut(A, B)}}{\mathrm{assoc(A, V)}} + \frac{\mathrm{cut(A, B)}}{\mathrm{assoc(V)} - assoc(A, V) + cut(A, B)} \quad (3)$$

Normalized cuts algorithm aims to minimize Ncut(A, B), so we set the derivative of Ncut(A,B) equal to 0.

$$\frac{\mathrm{dNcut(A, B)}}{\mathrm{d(assoc(A, V))}} = 0 \quad (4)$$

By evaluating (4) using (3) we get the following equation:

$$\frac{\mathrm{cut(A, B)} * \left(assoc(V) + cut(A, B)\right) * \left(-2assoc(A, V) + assoc(V) + cut(A, B)\right)}{assoc^2(A, V)(assoc(V) - assoc(A, V) + cut(A, B))^2} = 0 \quad (5)$$

Therefore:

$$assoc(A, V) = \frac{assoc(V) + cut(A, B)}{2} = \frac{assoc(A, V) + assoc(B, V)}{2} \quad (6)$$

Which means:

$$assoc(A, V) = assoc(B, V) \quad (7)$$

So, the Normalized Cut prefers cuts of approximately equal size theoretically, but this does not always happen in practice because of the complexity (Np hard) problem.

Question 10: Did you manage to increase *radius* and how did it affect the results?

Answers:

Yes, I managed to increase the radius in my implementations.

Increasing the radius parameter will influence the resulted image. The resulted image will include more neighborhood pixels into computation, so, larger super-pixels will be included in the results. This will lead to a decrease on the number of segments, but the color approximation will not be so accurate anymore.

Example:

Tiger3.jpg:

Radius = 3

Radius = 13

Question 11: Does the ideal choice of *alpha* and *sigma* vary a lot between different images? Illustrate with an example image with the parameters you prefer.

Answers:

Parameter alpha represents the maximum cost of an edge, while the parameter sigma represents the speed of the decays for decreasing similarities between neighboring pixels.

If we increase the value of alpha, then the maximum cost of each edge will be increased, which leads to cutting across similar pixels or smooth surfaces becoming more difficult.

If we decrease sigma, then the costs between the similarity between two neighboring pixels will decrease, and consequently, it will be easier to cut along strong edges of the image.

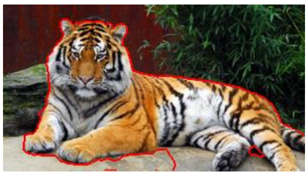
Taking these statements into account, we can easily say that between different images the ideal values for alpha and sigma may vary a lot.

<u>Examples:</u>

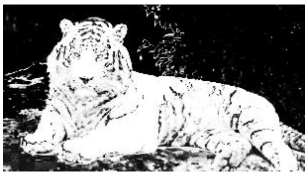
Tiger1.jpg: alpha = 15.0 sigma = 10.0

Resulting Segmentation

Overlay bounds



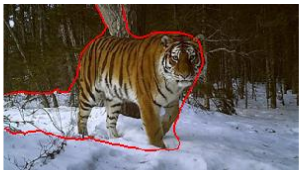
Prior foreground Probabilities



Tiger2.jpg: alpha = 20.0 sigma = 20.0

Resulting Segmentation

Overlay bounds



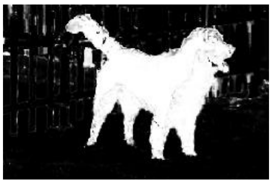
Prior foreground Probabilities

Tiger3.jpg: alpha = 17.0 sigma = 10.0

Resulting Segmentation

Overlay bounds

Prior foreground Probabilities



Question 12: How much can you lower K until the results get considerably worse?

Answers:

A value of K = 3 is the lowest value until the results get considerably worse.

Question 13: Unlike the earlier method Graph Cut segmentation relies on some input from a user for defining a rectangle. Is the benefit you get of this worth the effort? Motivate!

Answers:

Yes, this effort is worth it, since the segmentation is much better when using this method instead of the graph cut method. But this benefit depends on the image. If, for example the image includes a clear object surrounded by some background elements, then the accuracy of this algorithm will be higher than the graph cut algorithm. On the other hand though, if the image is more complex and includes more that one specific objects, then defining a rectangle will not be beneficial for the result. Another example that defining a rectangle will not be a good idea is when an image includes foreground elements all across the image, so the defined rectangle should be the same size as the whole image.

Question 14: What are the key differences and similarities between the segmentation methods (K-means, Mean-shift, Normalized Cut and energy-based segmentation with Graph Cuts) in this lab? Think carefully!!

Answers:

Similarities:

- These methods try to label similar points and group them to form different clusters
- All the methods use clustering to group data
- Mean-shift and Graph Cut both use the Gaussian distribution to model data
- Both of Normalized Cut and Graph Cut are graph-based

Differences:

- Some methods require prior knowledge from the user (K-means: number of clusters, Graph Cuts: define a rectangle), while some others do not require prior knowledge (Mean-Shift and Normalized Cuts)
- Mean-Shift takes into account spatial and color information of each pixel, while K-means only takes into account color information
- Normalized cuts, Graph cuts and Mean-Shift algorithms measure the similarity between neighboring pixels, while K-means does not use this similarity