Answers to questions in Lab 3: Image segmentation

Name: Ramadhani Pamapta Putra (rpputra@kth.se) Program: TMLEM

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:
We can randomly generate K amount of integer numbers from 0 – 255 to produce the centroids. And then we let each of pixels from the image to be assigned (by changing the RGB value) to the closest centroids.
This should be a good method because this is computationally less demanding.

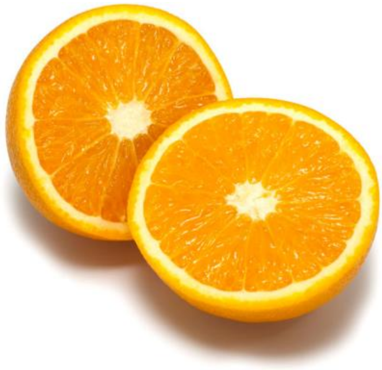
Question 2: How many iterations L do you typically need to reach convergence, that is the point where no additional iterations will affect the end results?

Answers:
With K = 8;

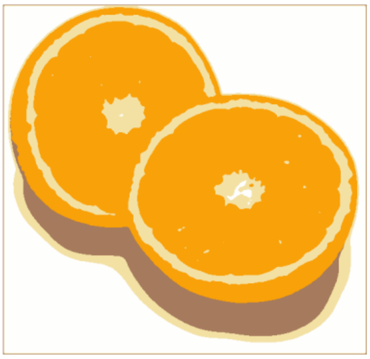
- orange.jpg requires 36 iterations
- Tiger1.jpg requires 23 iterations
- Tiger2.jpg requires 17 iterations
- Tiger3.jpg requires 32 iterations

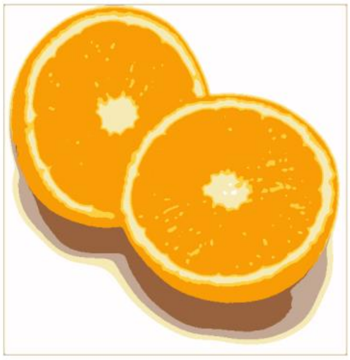
Question 3: What is the minimum value for K that you can use and still get no superpixel that covers parts from both halves of the orange? Illustrate with a figure.

Answers:



Original image





Segmented image, K = 5 Segmented image, K = 6

We can see that at K = the two halves of orange starts to segmented from each other, where the boundary is clear enough.

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

Answers:
Since the tiger image is more diverse in terms of color spectrum and levels, we need to increase the number of K to accommodate the spectrum and level diversity. Consequently, the iterations of L need to be increased as well. However, L value is not as crucial as K value.

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.

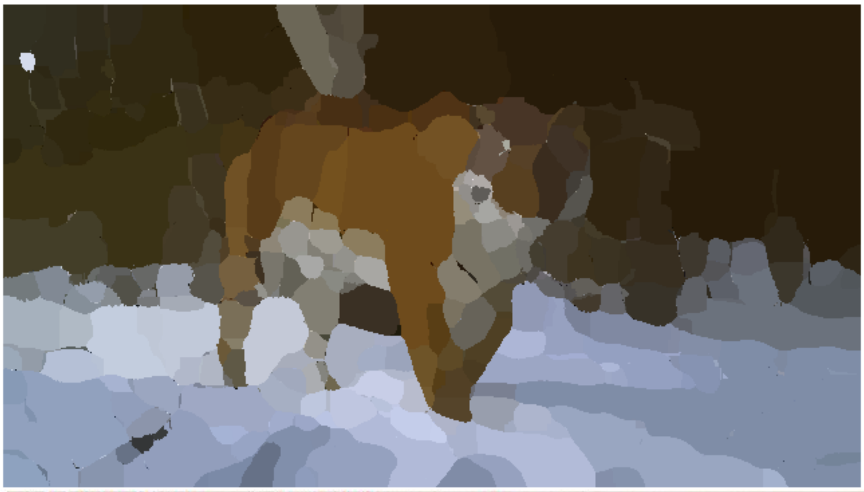
Answers:

Using larger number for bandwidths (especially for spatial bandwidth) will likely to make the segmentation patches to be larger and eventually undersegmentation, where two or more objects are segmented under same segmentation.

For different images, I'd prefer to change the color bandwidth rather than spatial bandwidth because color bandwidth affect color spectrum that will be able to represent image color diversity in different images. However, in terms of segmentation, the spatial bandwidth is more important to prevent undersegmentation.

Using ‘tiger1.jpg’ picture; the largest configuration is: spatial bandwidth of 5 and color bandwidth of 5.

Using ‘tiger2.jpg’ picture; the viable largest configuration number is: spatial bandwidth of 17 and color bandwidth of 5.



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

Answers:

Similarities:

- K-means and mean shift segmentation are having similarities in general ability to perform segmentation in an image to distinguish objects by creating patches/clusters that constitute each individual object in the image.

Differences :

- K-means segmentation is computationally lighter than mean-shift segmentation.
- K-means segmentation risks oversegmentation, however this is good when we want to find superpixels.
- K-means segmentation is prone to outlier pixels because it adapts to the means of pixel color intensities. Mean-shift segmentation is more robust.
- K-means segmentation does not involve pixel spatial location properties into account.

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:

Parameters :

```
colour_bandwidth = 30.0;
radius = 2;
ncuts_thresh = 0.1
min_area = 200;
max_depth = 10;
```

Parameters:

```
colour_bandwidth = 40.0;
radius = 2;
ncuts_thresh = 0.25;
min_area = 150;
max_depth = 10;
```

Parameters:

```
colour_bandwidth = 40.0;
radius = 2;
ncuts_thresh = 0.7;
min_area = 50;
max_depth = 10;
```

Ideal parameters are different for each images. This is due to the pixel color difference/similarities between foreground and background objects. Also, the image size would affect parameters such as *min_area*.. In case of diverse and complex image, *min_area* should be lower and *ncut_thresh* should be higher to avoid undersegmentation.

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

Answers:
Minimum area and cutting thresholds are the parameters that ensure how subdivisions are reduced in the image.

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

Answers:

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

Based on the Normalized Cut cost function above, the *assoc (A,V)* and *assoc (B,V)* should be as similar as possible in order to have minimized value of Ncut cost function. Therefore the size of two adjacent cluster (A and B) needs to be similar. However, in practice this does not always happen.

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

Answers:

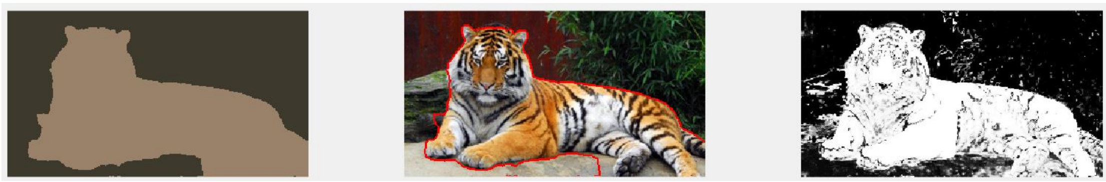
When the value of radius is increased step by step, the segmentation is affected by reduced number of clusters while retaining the edges between different objects (foreground vs background). In other words, the clusters inside a same object are reduced while keeping the inter-object edges. However increasing radius is computationally heavy and there are cases of miscoloring in segmented images.

Radius = 5

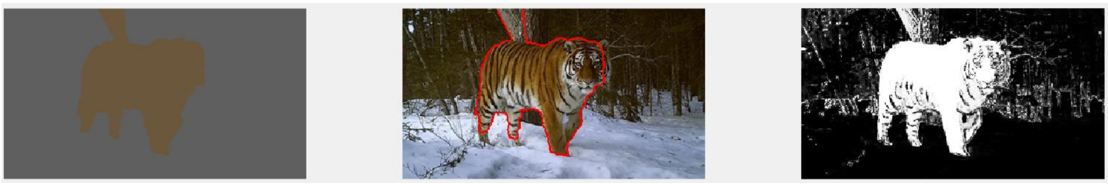
Radius = 8

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:



Best parameter settings for tiger1.jpg ; with alpha = 17, sigma = 10,
foreground area= [120, 64, 300, 300]



Best parameter settings for tiger2.jpg ; with alpha = 15, sigma = 12,
foreground area=[199, 90, 355, 199]

Best parameter settings for tiger3.jpg ; with alpha = 13, sigma = 8,
foreground area=[250, 112, 436, 228]

Each image requires unique combination of alpha and sigma (and of course foreground area for training) to ensure acceptable segmentation.

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

Answers:
For tiger1.jpg , it is K=3

For tiger2.jpg , it is K=2
For tiger3.jpg , it is K=2

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:
Since the input gives 'hint' to the segmentation algorithm to define the foreground pixels, the effort is absolutely worth it. Compared with other segmentation algorithm, the graphcut segmentation performs much better in distinguishing between foreground and background if given a proper input(most pixels inside the rectangle is the foreground).

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:

All these segmentation methods give labels to the pixels that are having similar intensity level; then segment and group those pixels into multiple clusters (and sub-clusters) based on the intensities.

Differences :

K-means = There is no spatial information needed in the segmentation process (inter-pixel differences is irrelevant). Cannot predict how many segmentations will be produced. Produces many-subclusters.
Mean-shift = Spatial information is taken into account in the segmentation process. Cannot predict how many segmentations will be produced. Produces many-subclusters
.
Normalized cut = It uses graph-based segmentation where pixels are reformed as graphs with corresponding weighted edges that represents difference/similarities between neighboring pixels. Produces subclusters.
Graph cut = Similar with normalized cut in the graph segmentation nature. However, it needs user input to define the foreground; and the segmentation is based on the inter-pixel differences/similarities(weighted edges) related to the predefined foreground. Generally produces little to no subcluster on the foreground.
