Answers to questions in Lab 3: Image segmentation

Name: _____Chia-Hsuan Chou_____ Program: _____SCR_____

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!

2.K-mean clustering

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

Answers:

I initialize the clustering process by setting the two of the clusters as [225 , 255 , 225] and [0 , 0 , 0] in the beginning while randomizing the other clustering. The reason why I set the clustering as such extreme RGB values is that k-means will gradually converge into the actual colors, if I initialize the two clustering in an extreme value, then the cluster can search for larger range of colors and finally converge into the correct ones. Although this might need more iterate times, but I think this is a good way to guarantee the colors found.

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:

The following table shows the L needed to converge with different k in different images. In this case, I assign a threshold and let it equals to 0.01, when the difference of old centers and new centers are all less than this threshold, then stop the iteration.

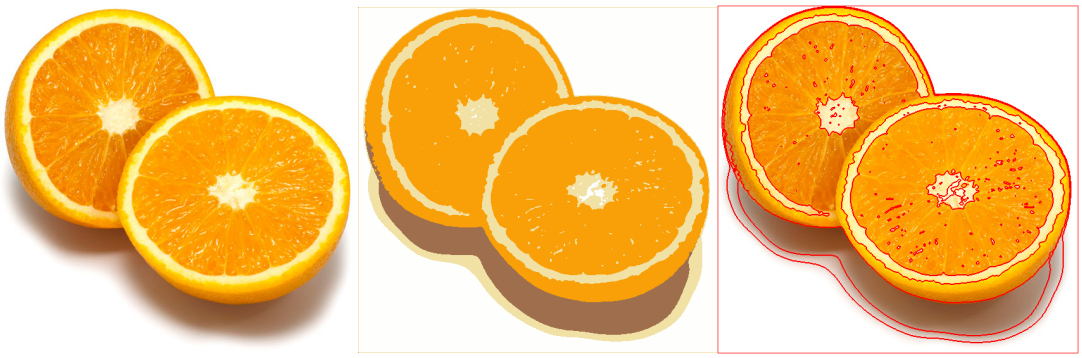
(with scale_factor = 1.0 and image_sigma = 1.0)

K	Orange.jpg	Tiger1.jpg	Tiger2.jpg	Tiger3.jpg
2	7	9	10	13
4	5	38	43	32
6	13	28	26	30
8	11	38	38	35
10	15	55	49	78

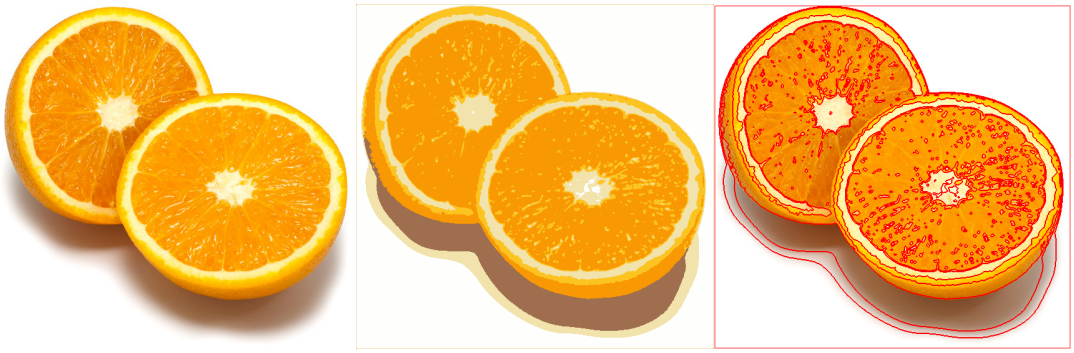
Question 3: What is the minimum value for K that you can use and still get no superpixel that covers parts from both halves of the orange? Illustrate with a figure.

Answers:

The minimum value for K that I can use and still get no superpixel covers part from both halves of the orange is 8, the following figures show the difference between k=8 and k=9:



with k = 8



with k = 9

From the figures above, we can see that when k = 8, there is no boundary between two halves of oranges, which means both haves of orange are in the same superpixel. however, when k = 9, there is a lighter orange on the boundary of the front orange, therefore the minimum k is 9.

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

Answers:

Since the images of tigers are more complicated than the orange image, we need to increase the iteration times L to get more suitable cluster centers. Also, the variety colors of tigers are more than oranges, we need to increase to amount of cluster centers as well. What's more, there are more noise in the images of tigers than in the images of oranges, the *image_sigma* should increase as well so that we can get the suitable superpixels for the tigers images.

---

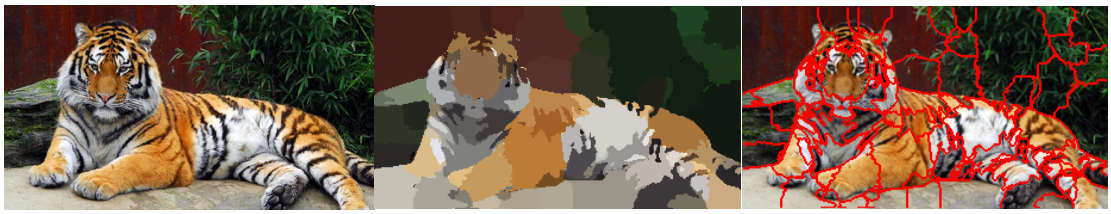
3. Mean - Shift Segmentation

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:

If we increase the bandwidths, the details on the image will disappear, and only color blocks on the image. On the other hand, if we decrease the bandwidths, there will be more details on the images. However, if we decrease the bandwidths too much, then only some small details will be left while other color blocks merge into one segmentation. Therefore we know that the bandwidths controls how large the mode will be, if we have a small bandwidths, then we will have more modes and segmentations, thus the details comes out, and vice versa.

Following images shows the example '*tiger1.jpg*' with different value of bandwidths:



spatial bandwidths: 10.0 ; color bandwidths : 5.0 (default)

spatial bandwidths: 20.0 ; color bandwidths : 10.0

spatial bandwidths: 30.0 ; color bandwidths : 20.0

spatial bandwidths: 5.0; color bandwidths: 2.0

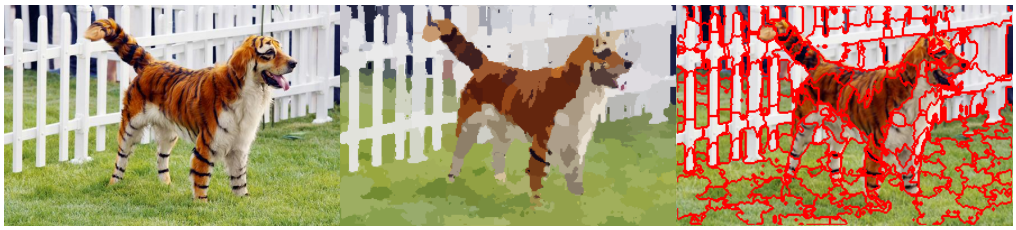
spatial bandwidths: 2.0; color bandwidths : 2.0

For me, the best result of implementation should be that we can separate the background and the foreground (object) and the superpixels will not cover more than one object in the scene. In the images above, we found that the best parameters for the image'*tiger1.jpg*' is that spatial bandwidth is 5.0 and color bandwidth is 2.0. The following figures show the other example images with the best parameters for that in mean-shift segmentation:

1. Tiger2.jpg

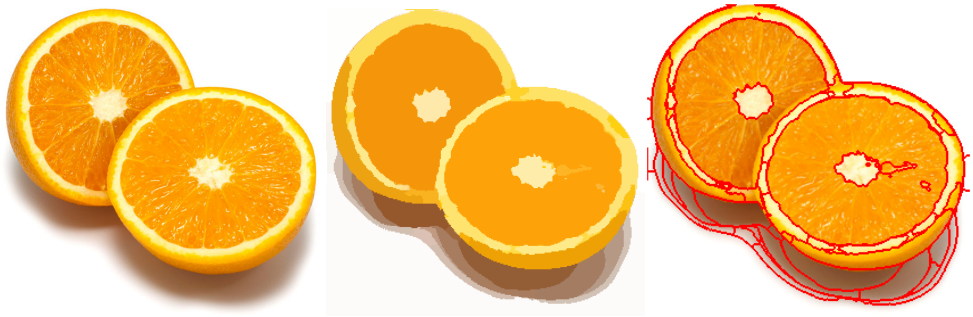
spatial bandwidths: 6.0; color bandwidths : 1.0

2. Tiger3.jpg



spatial bandwidths: 6.0; color bandwidths : 2.0

3. Orange.jpg



spatial bandwidths: 6.0; color bandwidths: 3.0

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

Answers:

Compare to k-mean segmentation, mean-shift segmentation is much more time-comsuming. What's more, Mean-shift segmentation considers not only the color but also the

position x and y, therefore, when implement mean-shift segmentation, some small details such as vesicles and flavedo will hardly show up in the segmentation.

Also, we cannot predefine how many clusters and modes we would like to the mean-shift segmentation, while in k-mean segmentation we can define it. As for the starting point, k-mean randomizes the clusters centers or initialize by ourselves while in mean-shift segmentation, all the pixels are the starting points.

---

4. Normalized Cut

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 parameters setting does vary depending on the images. For example, the parameter *color_bandwidth* calculates the similarity between the pixels, if we have larger *color_bandwidth*, then the algorithm will tend to cut the image into square instead of tracking the edge. If we have smaller *color_bandwidth*, then the segmentation can show more details on an object. Therefore, if the color difference is small between background and foreground ,for example, image tiger2, then we have better take a smaller color_bandwidth.

The parameter *min_area* might vary between different images. *Min_area* controls the minimum size of the segmentation, thus if we have an image with larger size of features and easier complexity of the feature, we should set a larger *min_area*.

Parameter *Ncut_thresh* controls the maximum allowed value for Ncut(A,B) for a cut to take place, if we have a smaller *Ncut_thresh,* we will have large area of each segmentation ,with smaller color diversity and smaller number of subdivisions.

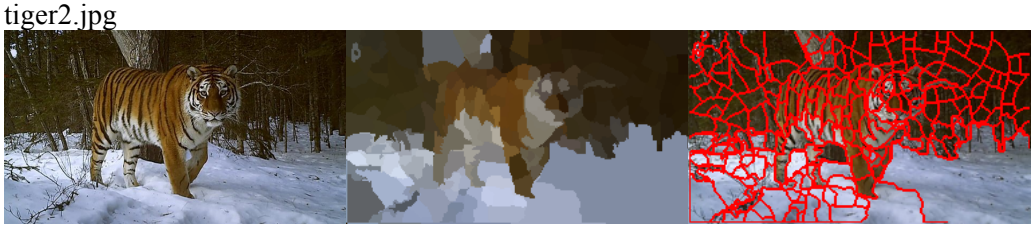
1. orange.jpg:

min_area = 40; color_bandwidth = 35; radius = 5 ; ncuts_thresh = 0.2

2. tiger1.jpg

min_area = 20; color_bandwidth = 60; radius = 8; ncuts_thresh = 0.3

3. tiger2.jpg



min_area = 60; color_bandwidth = 15; radius = 3; ncuts_thresh = 0.2

4. tiger3.jpg

min_area = 60; color_bandwidth = 60; radius = 6; ncuts_thresh = 0.4

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

Answers:

The parameters *min_area* and *ncuts_thresh* are most effective for reducing the subdivision and still result in a satisfactory segmentation. In general, a bigger "*min_area*" is and a smaller "*ncuts_thresh*" can reduce the subdivision and get a good result. Also the max_depth can be effective to reduce the subdivision, but

---

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

Answers:

The definition of $Ncut(A,B)$ is :

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

where $cut(A,B)$ is the sum of all edges that connect to two vertices A and B respectively and $assoc(A,V)$ is the sum of all edges that connect to any vertex in A. Therefore, we know that

$$assoc(V,V) = assoc(A,V) + assoc(B,V) - cut(A,B)$$

$Ncut(A,B)$ can be rewritten as :

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

and we can replace $assoc(A,V)$ with $assoc(V,V) - assoc(B,V) + cut(A,B)$ and get :

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

To find the minimum value of $Ncut(A,B)$, we take derivative of it :

$$\frac{dNcut(A,B)}{dassoc(B,V)} = \frac{-cut(A,B)\times\big(assoc(V,V) + cut(A,B)\big)\times\big(assoc(V,V) - 2assoc(B,V) + cut(A,B)\big)}{(assoc(V,V)\times assoc(B,V) - assoc(B,V)^2 + cut(A,B)\times assoc(B,V))^2} = 0$$

We get:

$$assoc(V,V) - 2assoc(B,V) + cut(A,B) = 0$$

and

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

we know that $assoc(V,V) = assoc(A,V) + assoc(B,V) - cut(A,B)$
therefore we get $assoc(A,V)$ by replace $assoc(B,V)$:

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

Theoretically the normalized cut segmentation prefers cuts of approximately equal size, however practically the normalized cut doesn't always cut the equal sizes as we can see from the result images in question 10.

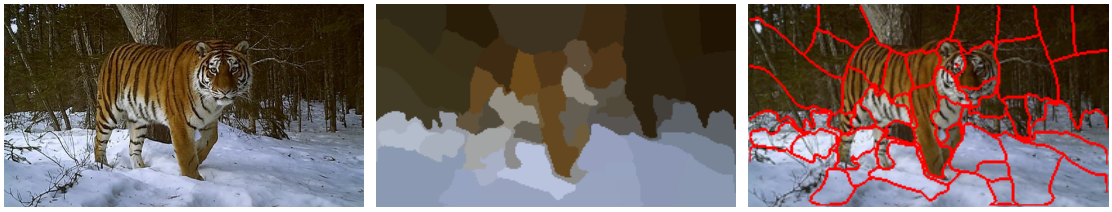
Question 10: Did you manage to increase *radius* and how did it affect the results?

Answers:

When increasing the radius, the size of each segmentation will also increase as well, the following figures show that under different radius with the image '*tiger2.jpg*', how do the segmentation change while other parameters remain default:

radius = 3

radius = 5



radius = 10

radius = 15

---

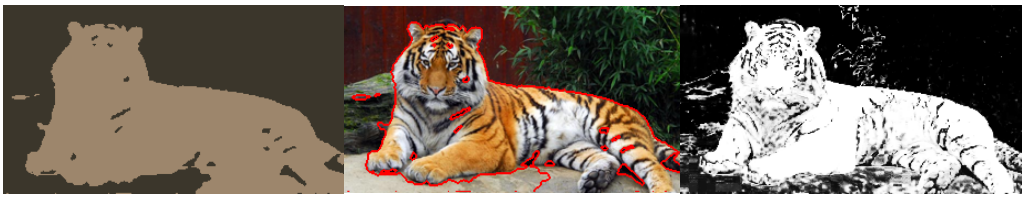
5. Segmentation using graph cuts

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:

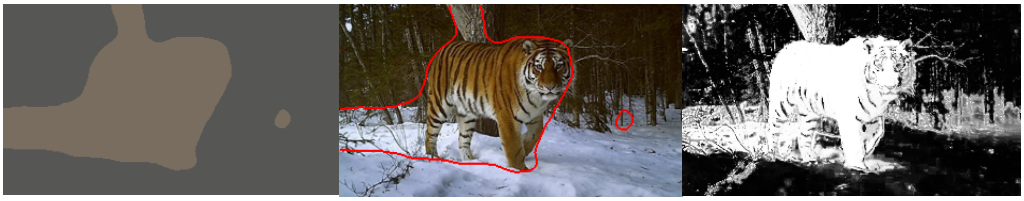
The following images show the best alpha and sigma in different images:

1. tiger1.jpg



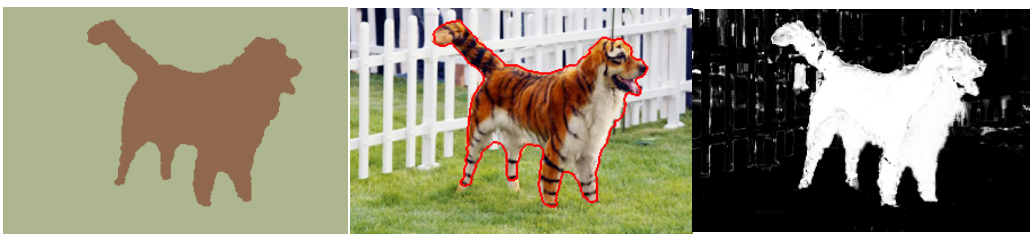
k = 16; alpha = 8; sigma = 10

2. tiger2.jpg



k = 16; alpha = 32; sigma = 40

3. tiger3.jpg



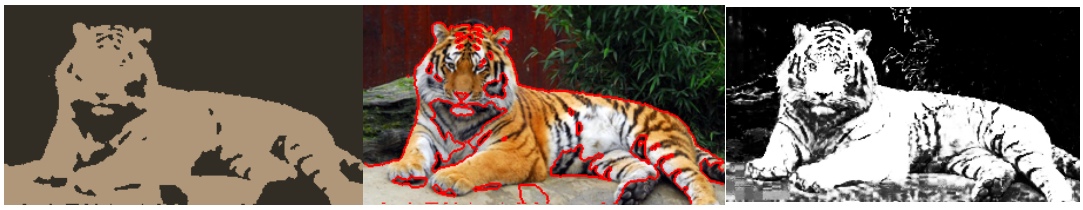
k = 16; alpha = 15; sigma = 18

The above images can prove that the parameter alpha and sigma do vary a lot between different images. We can find that if the image is much more complex, or the brief pixels colors in foreground and background are similar, then we need to increase both sigma and alpha.

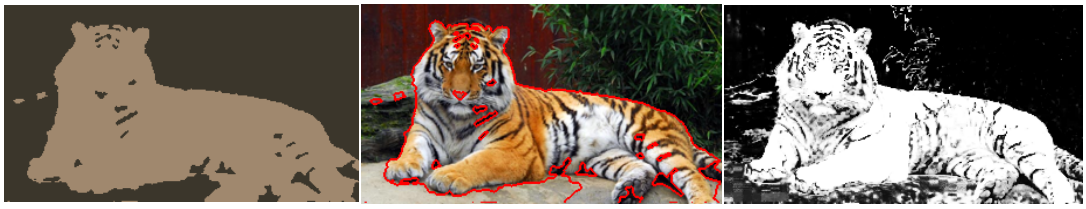
Question 12: How much can you lower K until the results get considerably worse?

Answers:

For *tiger1.jpg*, when k = 5, the results get considerably worse :

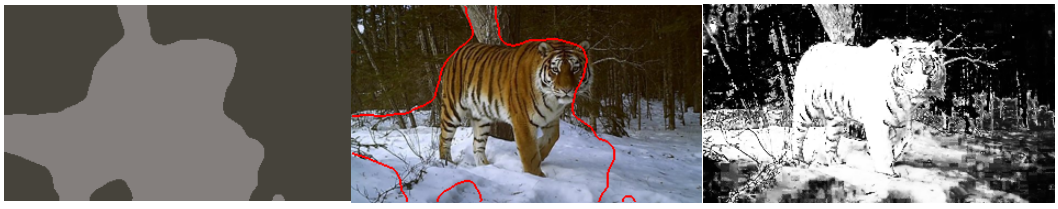


k = 5

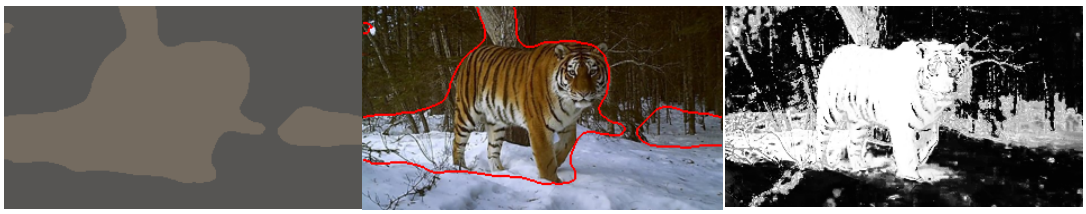


k = 6

For *tiger2.jpg,* when k = 3, the results get considerably worse:



k = 3



k = 4

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:

If most of the foreground is in the rectangle, then the system will have better results on segmentation. This is because graph cut segmentation cut the graph with probability, if most

of foreground is in this rectangle, then the system can give a higher probability on the pixels in foreground to be foreground in the first iteration. If there are many pixels which are in the background in this area, then the pixels in foreground will get relatively low probability to be background and the pixels in background might get higher probability to be foreground as well, then we will get a worse result.

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:

K-mean, mean-shift and normalized cut are methods of image segmentation which divided image in regions with pixels of similar qualities, while energy-based segmentation with graph cuts is the method of divide images into foreground and background.

Also, normalized-cut and graph cuts consider pixels as nodes, and give similarity as edge to find the best cut, the difference between these two methods is that graph cut needs probability to be foreground or background about the pixels to get a better result.

K-mean segmentation does not consider pixels' position when clustering, while mean-shift considers it. Also, we can define how many modes in K-means and initialize them. in the same time mean-shift segmentation cannot predefine how many modes, what's more, mean-shift segmentation takes every pixel as starting point so the mean-shift take more time than k-means.
