Answers to questions in Lab 3: Image segmentation

Name: *Javier López Iniesta Díaz del Campo* Program: *TSCRM*
Name: *Mathias Näreaho* Program: *TMAIM*

1 K-means clustering

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

In the course, we have seen different methods to initialize the clusters. First, we have tried to initialize the clusters by picking randomly K colours. However, this method has a problem that these colours could be not representative to the pixel colours of the image. Therefore, we have initialized the clusters by taking the colours of random pixels of the image. This method is neither perfect, because the same colour could be picked several times. Nevertheless, achieve very good results and it is easy to implement.

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?

As we can see in the Table 1 the number of iterations L depends on the number of K and the type of image. If we increment the number of iterations, then the algorithm will create larger super pixels. Thus, larger k means longer time to reach convergence.

K	`orange.jpg`	`tiger1.jpg`	`tiger2.jpg`	`tiger3.jpg`
2	5	11	8	13
4	15	78	45	32
6	21	65	36	33
8	66	75	72	42

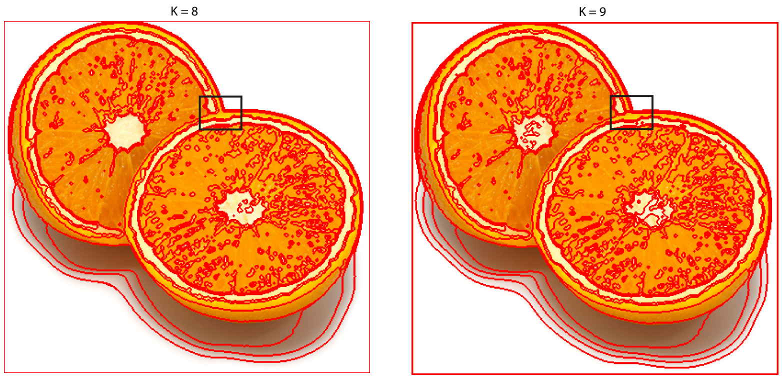
Table 1: Number of iterations needed to reach the convergence in different images.

The criterion chosen to determine convergence has been whether the center pixels of the clusters are equal to those of the previous iteration.

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.

Separating the 2 halves of the orange is not an easy task. With a small K we can separate the inner part of the orange. However to get no super pixel from the peels of both oranges, we need a minimum K equal to 9. However, this value of K depends on the compression that we made.

In the Figure 1 we can see the cluster overlay on the original image for different values of K.



(a) K=8 (b) K=9

Figure 1: Cluster overlay on the original image for K equal to 8 and 9.

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

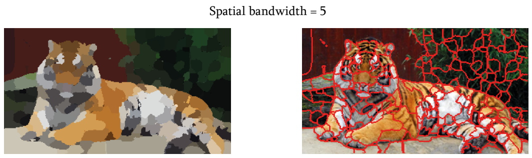
The tiger images have more features and many more colors. Therefore it is more difficult to get suitable super pixels. Therefore, we should use a larger number of clusters (K) and also increase the number of iterations (L) in order to achieve the convergence.

---

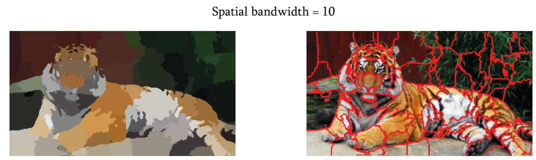
2 Mean-shift segmentation

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.

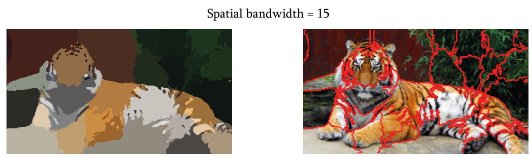
As we can see in the Figure 2 a higher spatial bandwidth will increase the size of the color blobs, making it less likely that individual pixels will be assigned closer to their true color. A lower spatial bandwidth will have the opposite effect.



(a) Spatial bandwidth = 5



(b) Spatial bandwidth = 10



(c) Spatial bandwidth = 15

Figure 2: Different values of the spatial bandwidth for a fixed value of the color bandwidth of 5.

On the other hand, the color bandwidth will effect the smoothing of the image, a large color bandwidth can cause adjacent different colors to mesh into one mixed color (Figure 3).

(a) Color bandwidth = 2

(b) Color bandwidth = 5

(c) Color bandwidth = 10

Figure 3: Different values of the color bandwidth for a fixed value of the spatial bandwidth (10).

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

Both types of segmentation determine superpixels based on the colour. The difference is that the mean-shift also takes into account the position of the pixels (x, y) to find those superpixels. Whereas, the K-means segmentation only focuses on the colour. Another difference is that mean-shift segmentation can not output a predefined number of segments. While, K-means can only generate K clusters, which were defined by the user at the beginning. Moreover, as we have more parameters mean-shift is computationally slower than K-means. Finally, mean-shift is more robust to outliers.

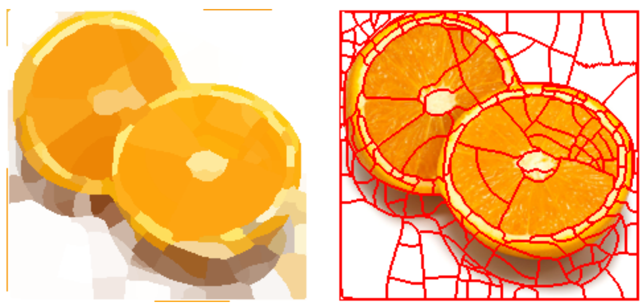
---

3 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 using the parameters you prefer for that image.

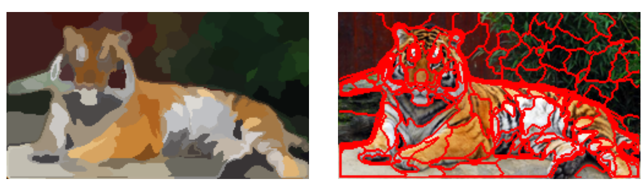
The ideal parameter settings vary depending on the images, because of the colors in the images. In the second tiger image, all of the colors are quite dark, and the cut values are therefore lower. The other two tiger images are brighter, and the differences in color are more obvious. This means that we don't need to do as much recursive splitting of the segments to get a good segmentation.

- min_area: Depends on the features and the complexity of the feature's structure.
- ncut_thresh: Depends on the colour diversity of the image. This affects how similar two areas can be and still be cut into several pieces
- max_depth: This is the amount of times we will recursively cut a segment. Useful when there are a lot of fine details (such as the true tiger images).



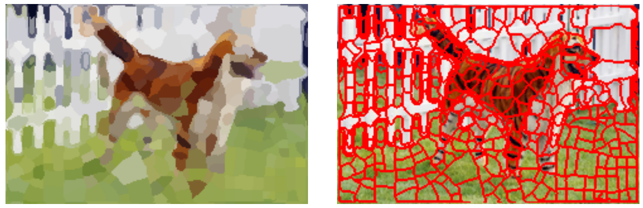
(a) Normalized cut segmented (b) Overlay bound

Figure 4: Norm cut on `orange.jpg` (colour_bandwidth=20, radius=3, ncuts_thresh=0.4, min_area=50, max_depth=8).



(a) Normalized cut segmented (b) Overlay bound

Figure 5: Norm cut on `tiger1.jpg` (colour_bandwidth=10, radius=6, ncuts_thresh=0.6, min_area=20, max_depth=8).



(a) Normalized cut segmented (b) Overlay bound

Figure 6: Norm cut on `tiger3.jpg` (colour_bandwidth=30, radius=6, ncuts_thresh=0.5, min_area=15, max_depth=10).

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

The parameter most effective for reducing the subdivision is `max_depth` because this is directly regulating the number of times we recursively split the segments.

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

If one of the subsets of vertices is a lot smaller than the other, the assoc of that subset will be very small, and the the term with that assoc in the denominator will become a lot bigger, leading to a larger N_{cut}. Thus, the algorithm attempts to make the subsets reasonably equal in size. In practice, many of the cuts are made in such a way that the Cut value is quite low, e.g. white snow on one side, dark trees on the other side. This can make the cut worth it, even if one subset is very small.

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

As shown in Figure 7, we managed to increase *radius*. If we increase the radius, it will result in an increase of the number of neighbour pixels when we compute the affinity matrix. Thus, it will decrease the number of segments in the image.

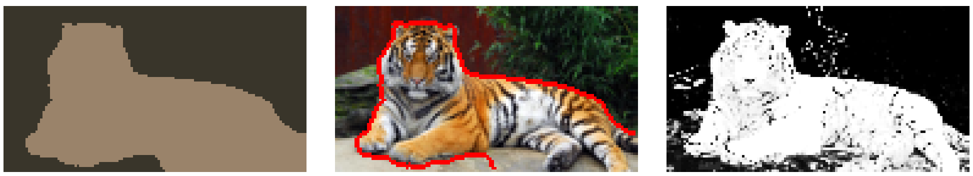
(a) Radius = 2. (b) Radius = 5. (c) Radius = 10.

Figure 7: Norm cut on `orange.jpg` with different values of the radius.

4 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.

Alpha determines the cost of cutting when two pixels have very similar color. High alpha means high cost of cutting between two similar pixels. Sigma determines how fast the cost decays when the similarity decreases.

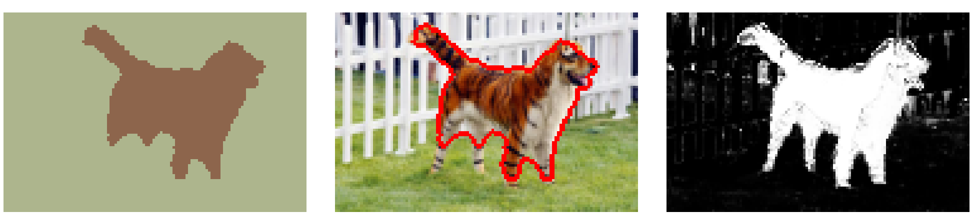


(a) Segmentation tiger 1 (b) Overlay bounds (c) Prior foreground probs

Figure 8: Graph cut on `tiger3.jpg` (alpha = 20, sigma = 15).

(a) Segmentation tiger 2 (b) Overlay bounds (c) Prior foreground probs

Figure 9: Graph cut on `tiger2.jpg` (alpha = 5, sigma = 30).



(a) Segmentation tiger 3 (b) Overlay bounds (c) Prior foreground probs

Figure 10: Graph cut on `tiger3.jpg` (alpha = 15, sigma = 10).

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

The value of K can be lowered to 3, but fewer will not give much worse results. K is the number of gaussian components that we are using in the model.

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!

It can certainly be worth the effort for many types of image where there is a object that constitutes the foregrond. It would be less useful for an image with a lot of clutter in the foreground, like an image of a desk, or an image of a forest. In these cases it is often harder to draw a meaningful rectangle for segmentation.

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!!

The main similarity between all the segmentation methods is that all rely on the colour of the pixels in order to segment out similar pixels and form clusters. Other similarity between Normalized cut and energy-based segmentation with Graph Cuts is that they treat each image as a graph. Then, they construct the weights based on the similarities between the pixels.

Regarding the differences, K-means differs from Mean-shift method since it does not take into consideration spatial information while Mean-shift does. Moreover, Mean-shift is computationally slower than K-means.

On the other hand, normalized cut differs from energy-based segmentation with Graph Cuts because they also need prior information of the pixels. In addition, Graph Cuts tries to find out the foreground and background in the image. Finally, Normalized cut tries to segment the images into equally large parts while the others not.