Lab 3: Image segmentation

DD2423 Image analysis and Computer Vision, 2019

Federico Favia
favia@kth.se

Martin De Pellegrini
martindp@kth.se

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?*

Since the dominant colours in the image `orange.jpg` are the orange and white, to initialize the cluster in a good manner, the clusters centers are taken from the dataset pixels pseudo-randomly based on the Mersenne Twister algorithm (`rng(seed, 'twister')`) in order to be sure that these centroids are equally distributed. In this manner, more likely the centroids will start with different colours. Another method, but less useful, could be of completely randomly initialize the centroids not being sure they will fall onto some of the image pixels.

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?*

The number of iterations depends on the value of K and the type of image. The analysis made on the image `orage.jpg` have shown that with a low number of iterations pixels are grouped in many different small regions, leading in a over segmentation. Setting an higher number of iterations instead, the algorithm create larger super pixels. However, iterating too much do not introduce any improvement. The maximum number of iteration for convergence found in this experience is `L=12`. To improve the algorithm, a good practice could be to stop the iterations when the maximum value difference of cluster centers on the current and previous time step is below a certain threshold.

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.*

As we can see from Fig. 1, K = 5 is the minimum value to get no super pixel that covers parts from both halves of the orange, apart for a small connection on the upper intersection of the orange skin (that we can neglect). There is a clear boundary between the two halves of the orange when `K = 5`, while that boundary disappears when `K = 4`.

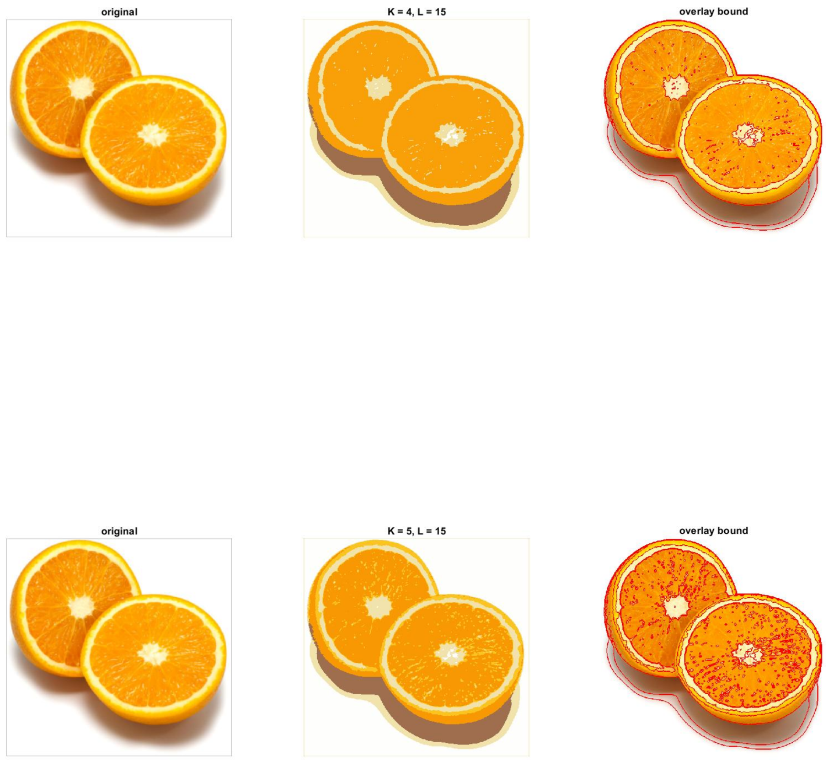


Figure 1: Results of K-means on orange.jpg with K = 4 and K = 5.

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

As we can observe from Fig. 2, because the images regarding tiger are more diverse in colours than the orange colour, we need to increase the cluster number K. As K increases, more iterations will be needed to obtain convergence. So, the number of iterations L should be increased as well.

Figure 2: Results of K-means on tiger1.jpg with K = 5 and K = 7.

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 from Fig. 3 and 4, the value of spatial bandwidth controls the radius of the region of interest. The larger the spatial bandwidth is, the larger the region of interest becomes, which means more pixels are involved in computing the mean and fewer modes will be generated. On the contrary, decreasing the spatial bandwidth will increase the number of modes and segments generated. As the colour bandwidth increases, the image gets smoothed more and the colour approximation becomes better. On the contrary, a low bandwidth will lead to more colour differences in the segmented image.
We tried with the image `tiger1.jpg` and since it is richer in colours details, a lower `colour_bandwidth` is needed to segment better the image. Instead the `spatial_bandwidth` needs to be fine tuned carefully in order to avoid over-segmentation.

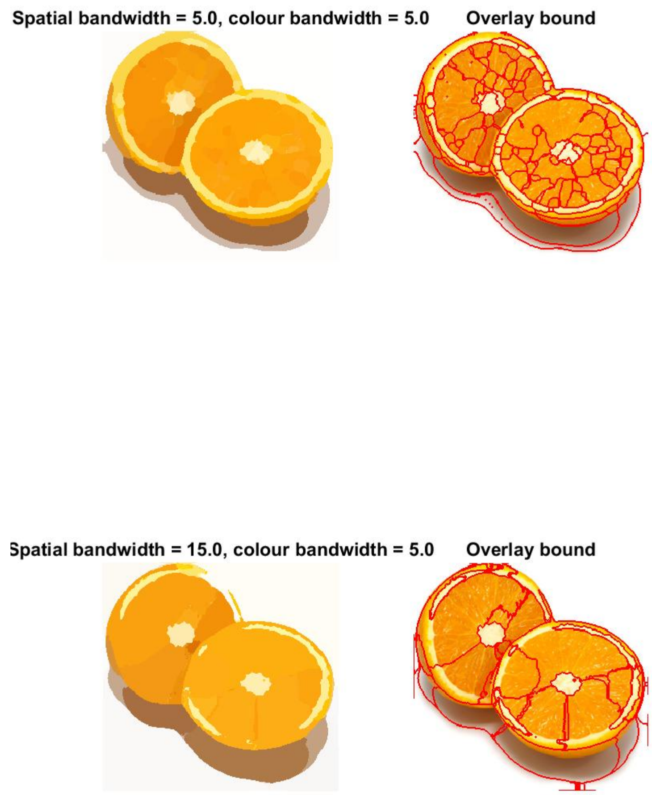
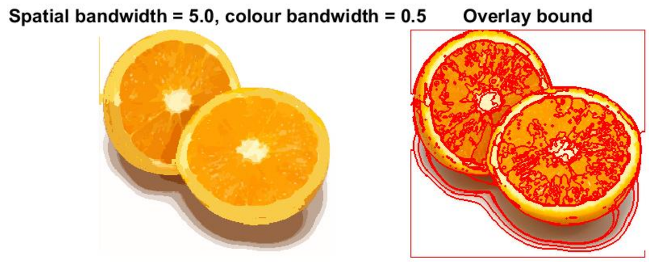


Figure 3: Results of Mean-shift on orange.jpg with spatial_bandwidth = {5.0, 15.0} and colour_bandwidth = 5.0.



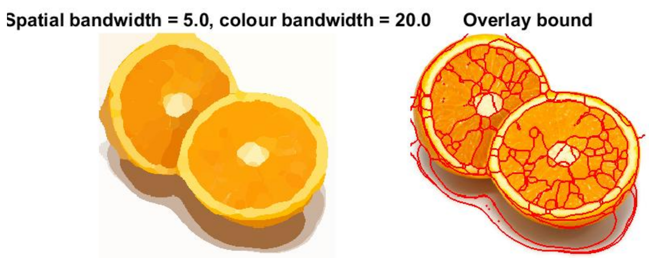


Figure 4: Results of Mean-shift on orange.jpg with spatial_bandwidth = 5.0 and colour_bandwidth = $\{0.5, 20.0\}$.

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

A key difference is that Mean-shift cannot output a predefined number of segments but find the number of modes by itself while K-means can only generate K clusters as defined by the user in the beginning. The choice of K is often critical, and some optimization methods include running k-means with multiple K*s* finding the one for which the intra-class variance is minimized and/or the inter-class variance is maximized. K-means only takes colour dimension into consideration (in the way it is implemented in this laboratory) while Mean-shift also uses the spatial information of the pixels. This means that the segments generated by K-means could span over multiple regions while they spatially concentrate by Mean-shift segmentation.
Similarly instead, they both treat the colour and/or position of pixels as samples from a probability distribution and try to determine its clusters or modes.

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 setting varies depending on the images. Parameter `min_area` depends on the size of the features and the complexity of the feature's structure; parameter `ncut_thresh` depends on the diversity in colour of the image. Since the images of tigers are more diverse in colour and rich in details, more accurate cuts will be required to get suitable segments. Therefore, we need to decrease the `min_area` and increase the `ncut_thresh` to preserve the characteristics of the shape and colour. For `colour_bandwidth` the same principle of before is applied, therefore for a image with a smaller range of colours a smaller `colour_bandwidth` is needed.

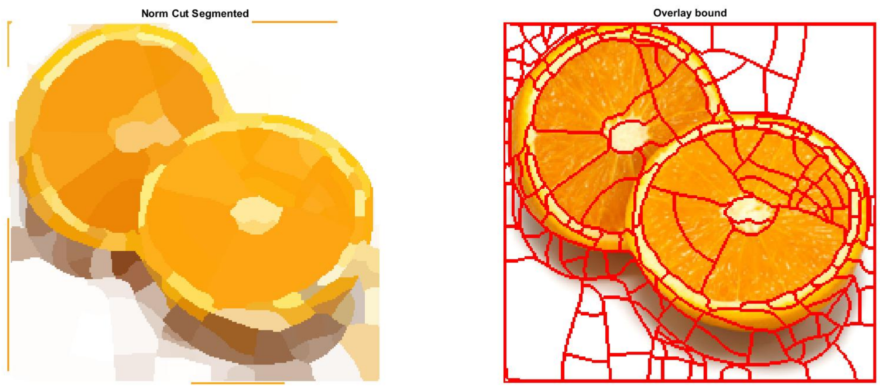


Figure 5: NormCut on orange.jpg: colour_bandwidth = 20, radius = 3, ncuts_thresh = 0.4, min_area = 50, max_depth = 8.

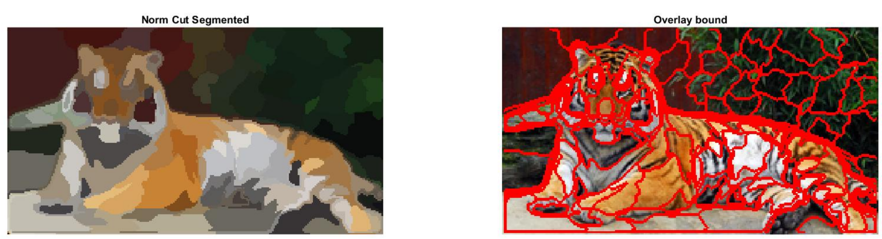


Figure 6: NormCut on tiger1.jpg: colour_bandwidth = 10, radius = 6, ncuts_thresh = 0.6, min_area = 20, max_depth = 8.

Figure 7: NormCut 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?*

Parameter `min_area`, `ncut_thresh` and `max_depth` are candidates for reducing the subdivision and resulting a satisfactory segmentation.

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

We will explain through the equations (1), (2) and (3):

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

Since both $assoc(A, V)$ and $assoc(B, V)$ include those edges whose one vertex is from the opposite subset of vertices, $assoc(V) = assoc(A, V) + assoc(B, V) - cut(A, B)$ should hold. Then we could get:

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

To minimize $Ncut(A, B)$ we can compute the derivative:

$$\frac{dNcut(A, B)}{dassoc(A, V)} = \frac{cut(A, B)(assoc(V) + cut(A, B))(-2assoc(A, V) + assoc(V) + cut(A, B))}{assoc^2(A, V)(assoc(V) - assoc(A, V) + cut(A, B))^2} = 0 \tag{3}$$

Therefore, $assoc(A, V) = \frac{1}{2}(assoc(V) + cut(A, B)) = \frac{1}{2}(assoc(A, V) + assoc(B, V))$, which means $assoc(A, V) = assoc(B, V)$. Hence, the Normalized Cut prefers cuts of approximately equal size theoretically, but this does not always happen in practice because of the NP-hard problem ((non-deterministic polynomial-time).

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

As we can see from Fig. 8, increasing `radius` will include more neighborhood pixels into computation, and will result in a decrease on the number of segments, but the colour approximation will sort of get ruined.

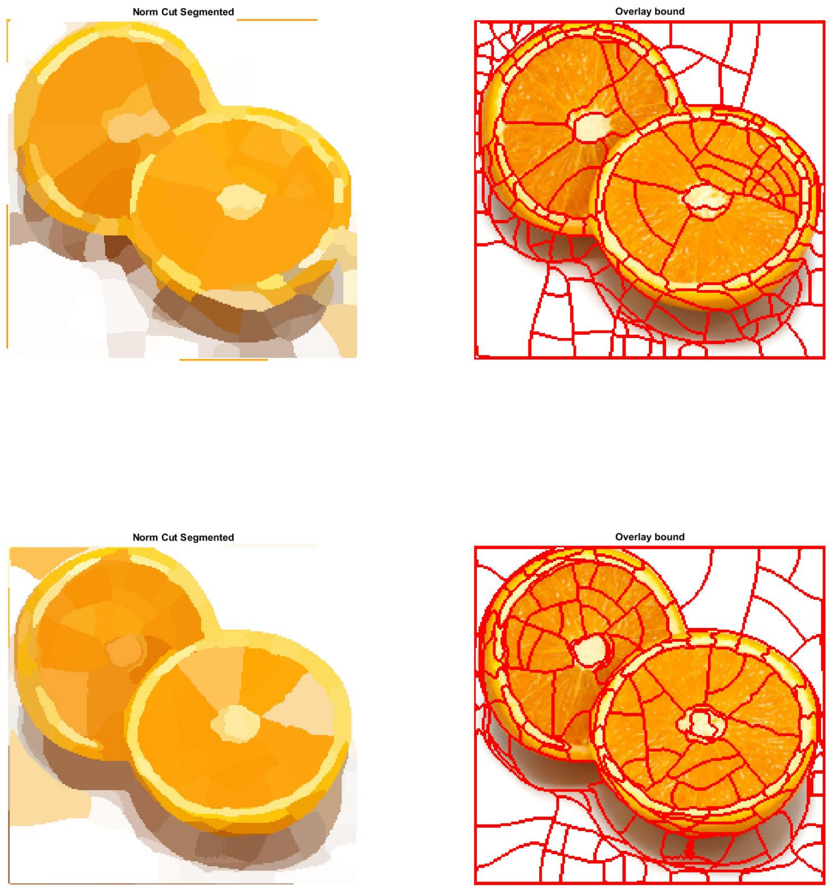


Figure 8: NormCut on orange.jpg with radius = {3, 6}.

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.*

For image `tiger1.jpg`, the ideal values of `alpha` and `sigma` should all be around 8 and 10 respectively. For `tiger2.jpg`, alpha = 30 and sigma = 25 will be the best choice and alpha = 12 and sigma = 14 for `tiger3.jpg`. Generally, if we decrease the two parameters, more inaccurate subdivision will be presented; if we increase the parameters, the result looks similar to the optimal one but seems a little bit coarse.

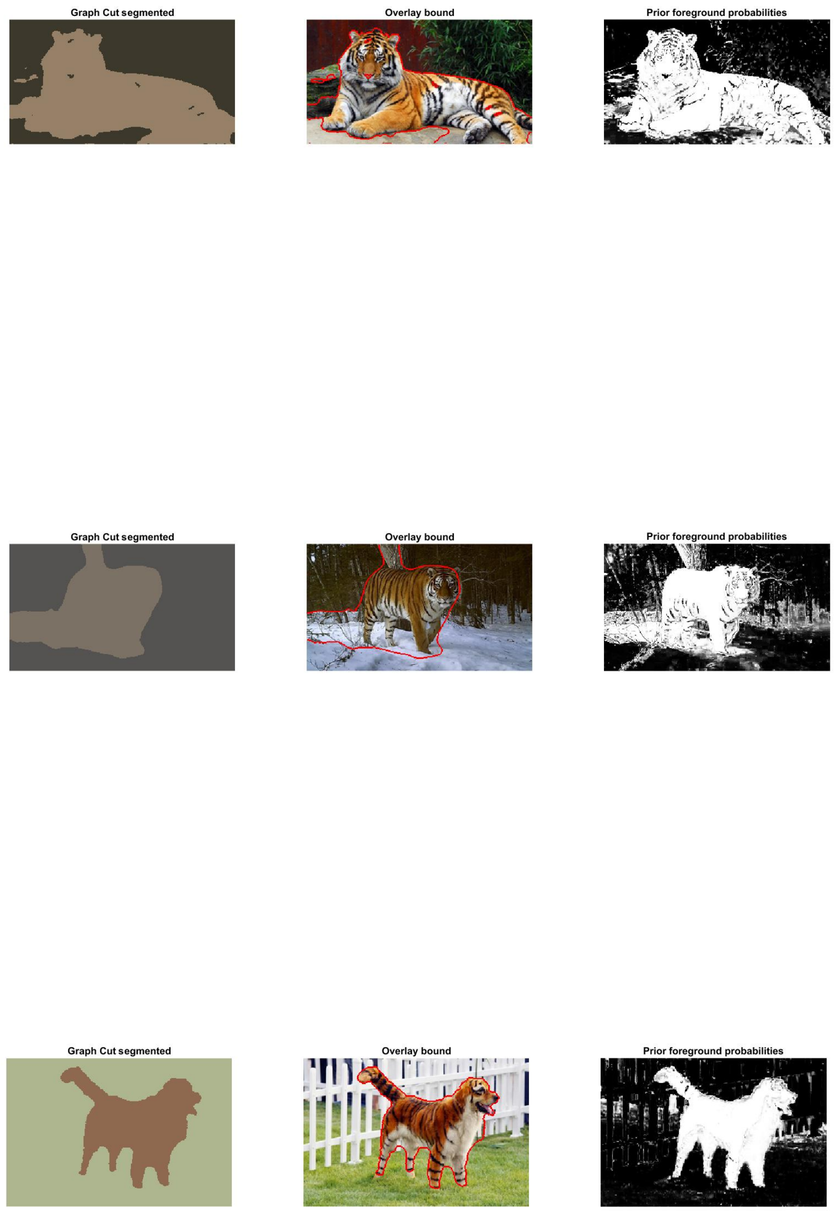


Figure 9: GraphCut on tiger1.jpg (alpha = 8, sigma = 10), on tiger2.jpg (alpha = 30, sigma = 25) and on tiger3.jpg (alpha = 12, sigma = 16).

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

Testing on image `tiger1.jpg` we observed that we can reduce `K` down to 4 and still get acceptable results. In Fig. 10 we show that with `K` = 2 and 3 the image is not enough well-segmented.

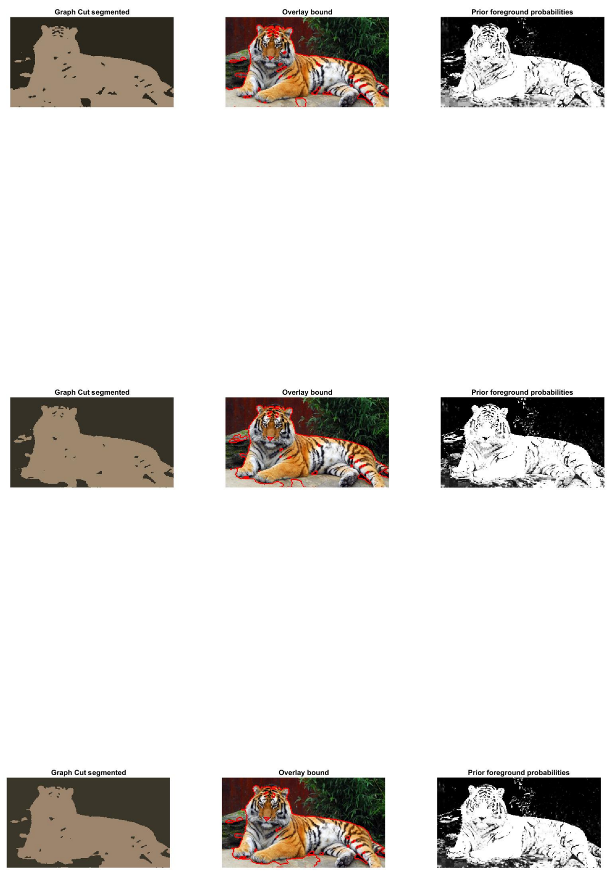


Figure 10: GraphCut on tiger1.jpg with K = {2, 3, 4 }.

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!*

If most parts of the foreground objects are located within the rectangle, then it will be much helpful to use the information inside to obtain an accurate training set, and provide classification probability for each pixel in the image. Otherwise, it will not be a good idea to use this bounding box since the result from the training data will not build a reliable inference to 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!!*

Regarding similarities, all the segmentation methods try to label those similar points and group them to form different clusters. In this way, the image will be segmented into various parts. Differently, K-means does not take spatial information into consideration while Mean-shift does. Normalized Cut and energy-based segmentation with Graph Cuts treat each image as a graph and construct the weights (edges) based on the similarity between pixels. The difference between these two methods is that the Graph Cuts also need prior information of the pixels (probability to be foreground or background element) to obtain a superior segmentation.

References

[1] Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing*, Prentice Hall, 2nd ed., 2002

[2] R. Szeliski, *Computer Vision: Algorithms and Applications*, Springer, 2010