

# DD2423 - Image Analysis and Computer Vision

## Lab 3

Gabriela Zarzar Gandler  
gzrsm@kth.se

Huijie Wang  
huijiew@kth.se

December 2016

## 2 K-means clustering

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

We initialize the clusters by picking  $K$  random pixel values in the image. We compared this method with picking  $K$  random colors in the RGB space. We believe the first approach is more effective, because it avoids picking values that are too far from the values in the image.

The result is shown as Figure 1:

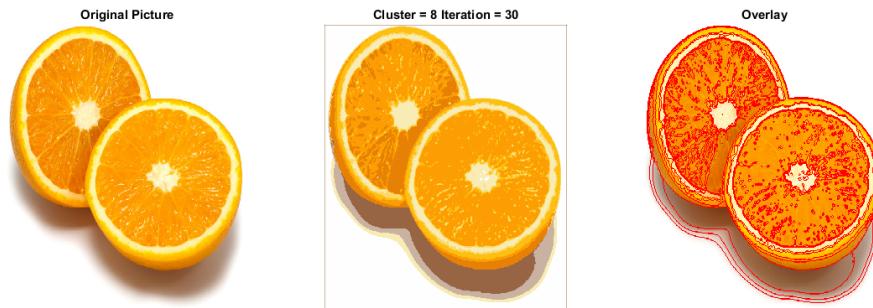


Figure 1: Result of K-means

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?

For  $K = 8$ , between 49 and 88 iterations are necessary. But it is noticeable that the centers changes very little after iteration = 10. For  $K = 16$ , between 107 and 156 iterations are necessary. The fewer clusters, the shorter the convergence time. In order to measure the number of iterations until convergence, we check if the centers of the clusters are the same between one iteration and the other.

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?

It is hard to find an optimal  $K$  for this.  $K = 18$  still doesn't offer an optimal segmentation. This was expected, because the colors are not reliable criteria to segment the halves of the oranges.

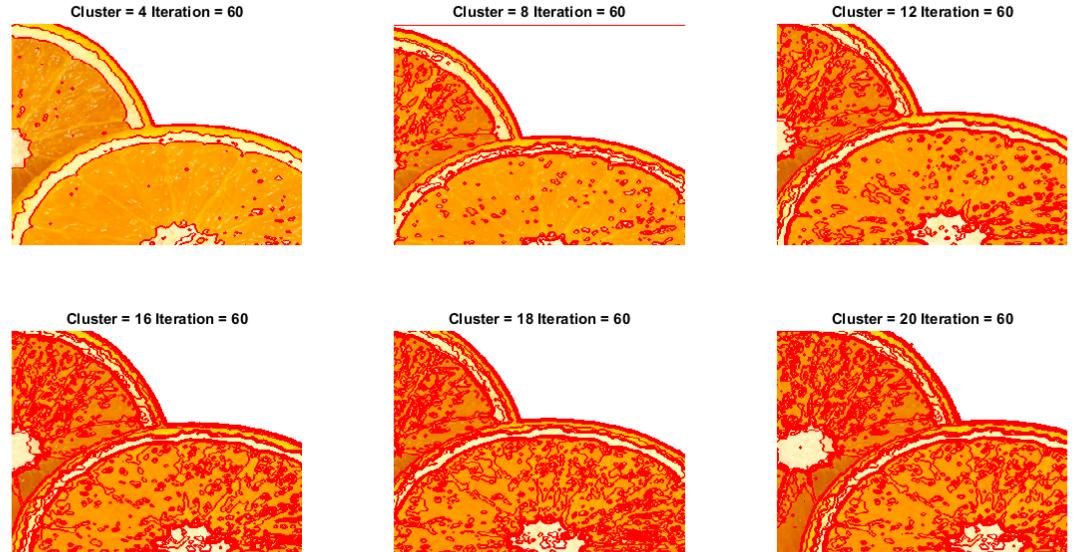


Figure 2: Test on superpixels

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

Parameter K should be changed and L should be set to suitable size. The results are shown as Figure 3, 4 and 5.



Figure 3: Result of *tiger1.jpg*

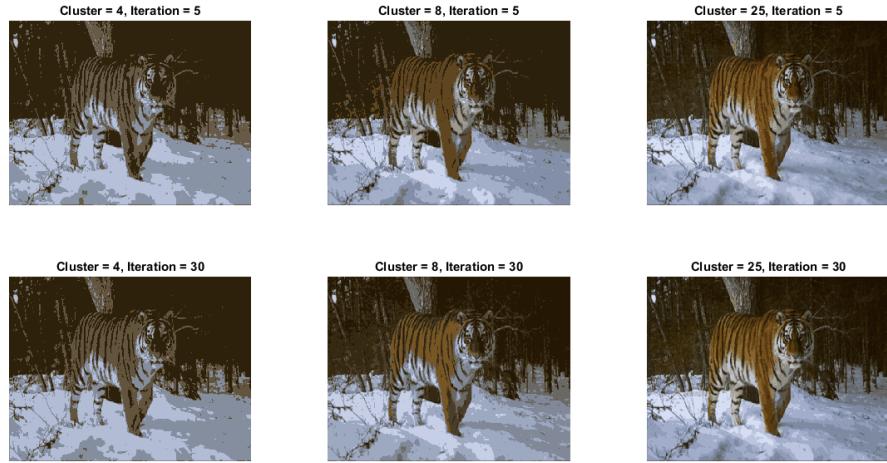


Figure 4: Result of *tiger2.jpg*

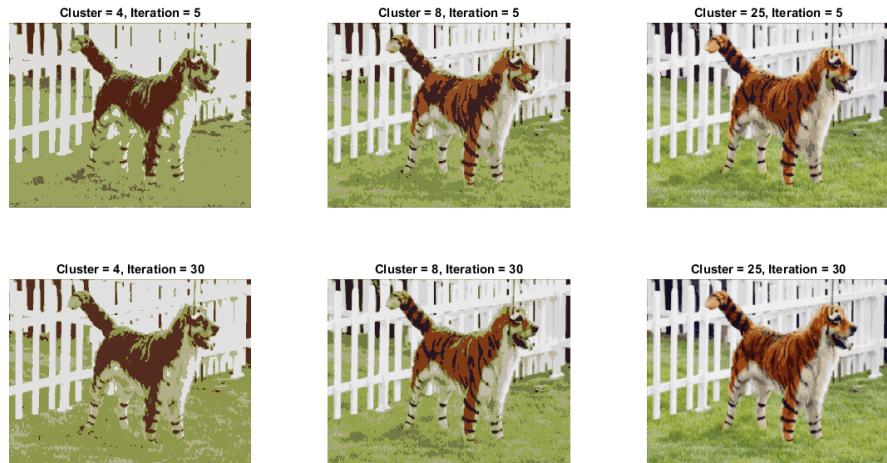


Figure 5: Result of *tiger3.jpg*

### 3 Mean-shift segmentation

5. How do the results change depending on the bandwidths? What settings did you prefer for the different images?

If the bandwidth is too high, the estimation of the density function may be too flattened, i.e. the estimation obscures much of the underlying structure. This tends to create few segments in the image. However too low bandwidths may result in noisy estimations, affected by spurious data artifacts. This tends to create highly segmented image, because of too

many modes. So a compromise between these two situations needs to be met. We noticed that applying bandwidth 3 both in the spatial and color domains results in a fairly good segmentation.

The result is shown as Figure 6



Figure 6: Result of meanshift

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

Mean-shift segmentation is a non-parametric approach, except by the hyperparameter (bandwidth) in the kernel definition. This means the mean-shift algorithm segments the image by assigning each sample point to a mode, without however actually estimating the whole probability density function that generated the samples. K-means, however, implicitly models the probability density as a superposition of spherically symmetric distributions - but it still does not require any probabilistic modeling. This all means that, differently from k-means, mean-shift segmentation doesn't require pre-determining the number of clusters, which is great. Additionally, in the case of this assignment, we are aggregating positional information to the features in mean-shift segmentation, while only using information about color in the k-means algorithm.

It can be said that both algorithms make use of clustering to find regions that are somewhat similar in the image.

## 4 Normalized Cut

#### 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?

Yes, as expected. The number of objects are different in different images (which may affect the maximum depth), the formats of the segments differ as well (which may affect the radius), the difference in colors (background x foreground) also differ (which may affect the color bandwidth). Figure 7 shows the result of applying NormCuts in the image tiger1.



Figure 7: NormCuts:  $\text{colour}_{bandwidth} = 20$ ;  $\text{radius} = 3$ ;  $\text{ncuts}_{thresh} = 0.2$ ;  $\text{min}_{area} = 50$ ;  $\text{max}_{depth} = 6$

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

The parameters  $\text{ncuts}_{thresh}$  and  $\text{min}_{area}$  were most useful to reduce the subdivision and still result in a satisfactory segmentation. Figures 8 and 9 show the results.

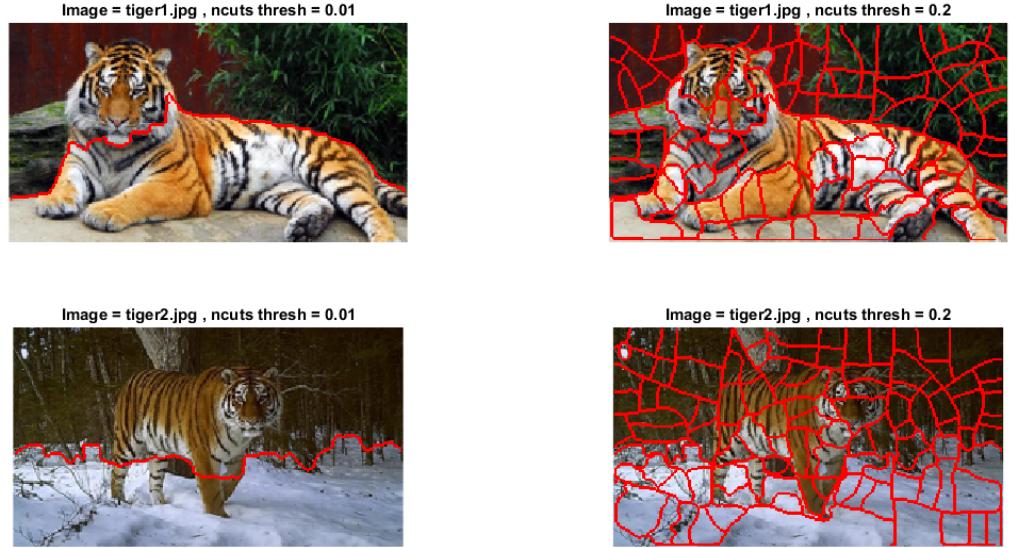


Figure 8: NormCuts:  $\text{colour}_{bandwidth} = 20; \text{radius} = 3; \text{ncuts}_{thresh} = 0.01$  and  $0.2; \text{min}_{area} = 200; \text{max}_{depth} = 8$



Figure 9: NormCuts:  $\text{colour}_{bandwidth} = 20; \text{radius} = 3; \text{ncuts}_{thresh} = 0.2; \text{min}_{area} = 20$  and  $200; \text{max}_{depth} = 8$

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

Because the objective function itself induces the areas to be more or less of the same size. Yes, this can be observed in practice, via the given figures.

#### 10. Did you manage to increase radius and how did it affect the results?

It makes the contours follow the true edges better, which was expected. Figure 10 shows the difference.

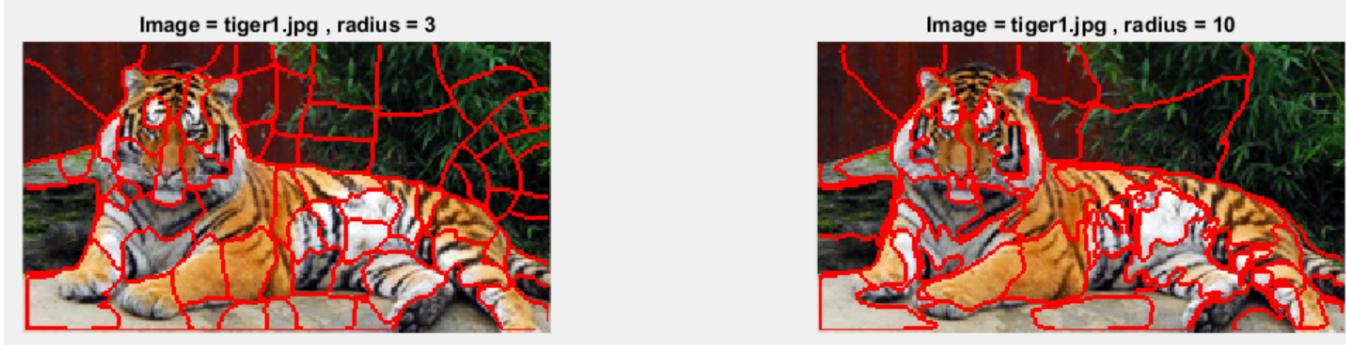


Figure 10: NormCuts:  $\text{colour}_{bandwidth} = 20$ ;  $\text{radius} = 3$  and  $10$ ;  $\text{ncuts}_{thresh} = 0.2$ ;  $\text{min}_{area} = 50$ ;  $\text{max}_{depth} = 6$

## 5 Segmentation using graph cuts

#### 11. Does the ideal choice of *alpha* and *sigma* vary a lot between different images?

- (a) orange : Not very sensible to the parameter since the shape is not very complex.  $\text{alpha} = 4$  and  $\text{sigma} = 6$  seems to be fine.

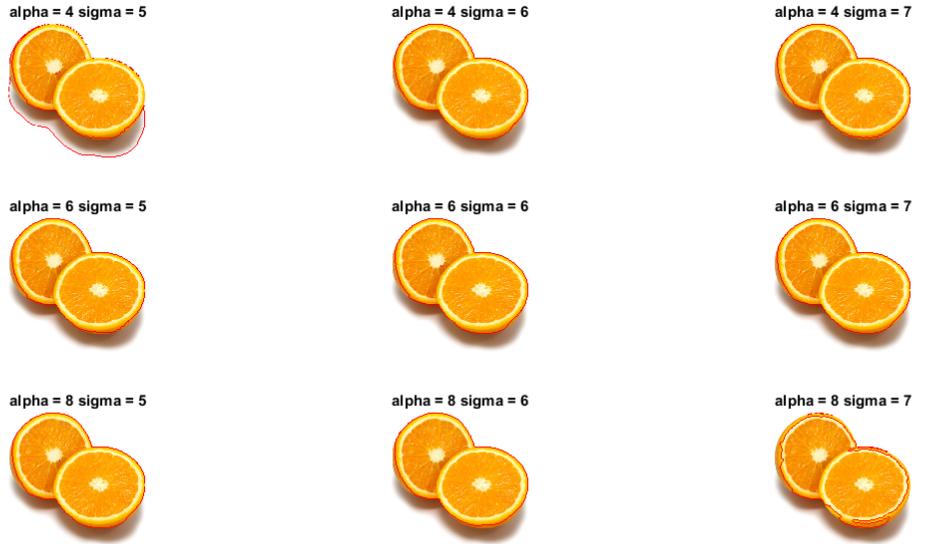


Figure 11: Results with different  $\alpha$  and  $\sigma$  for *orange.jpg*

(b) tiger1 :  $\alpha = 8$  and  $\sigma = 10$



Figure 12: Results with different  $\alpha$  and  $\sigma$  for *tiger1.jpg*

(c) tiger2 : hard to find the optimal.  $\alpha = 15$  and  $\sigma = 10$  seems to be OK.



Figure 13: Results with different  $\alpha$  and  $\sigma$  for *tiger2.jpg*

(d) tiger3 :  $\alpha = 15$  and  $\sigma = 15$  seems to be optimal.



Figure 14: Results with different  $\alpha$  and  $\sigma$  for *tiger2.jpg*

## 12. How much can you lower K until the results get considerably worse?

$K = 6$  is still generates acceptable result for *tiger1.jpg*, as shown in Figure 15.

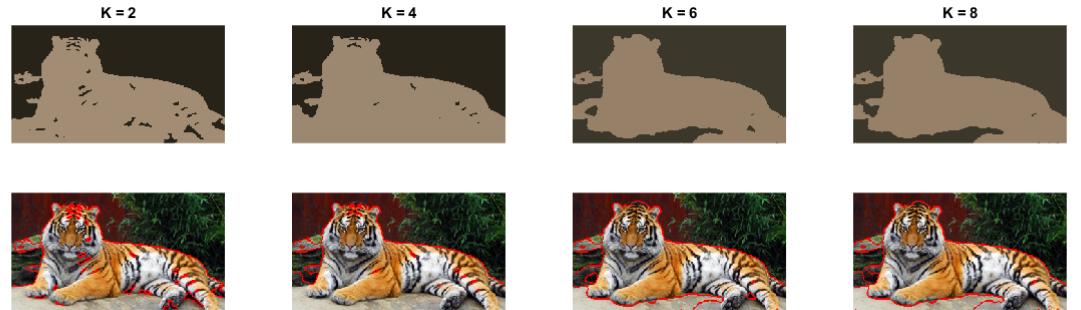


Figure 15: Results with different  $K$  for *tiger1.jpg*

$K = 2$  is still generates acceptable result for *tiger3.jpg*,

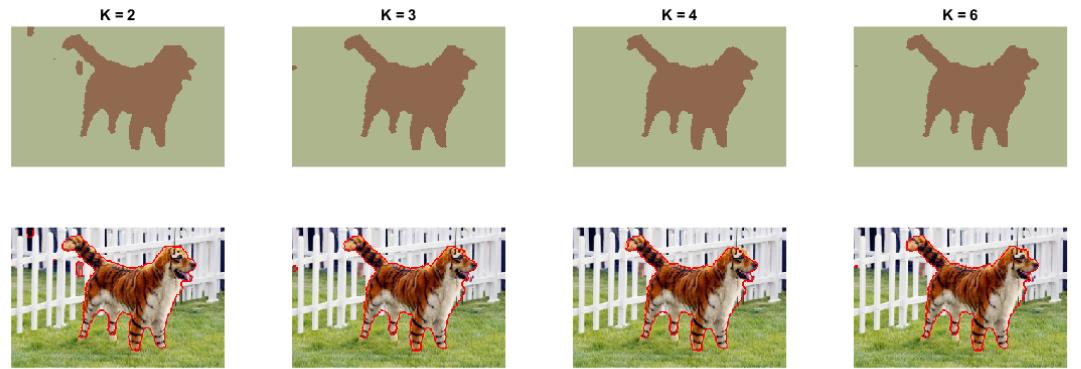


Figure 16: Results with different  $K$  for *tiger3.jpg*

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!

Yes, the results of different initialization area is shown as below:

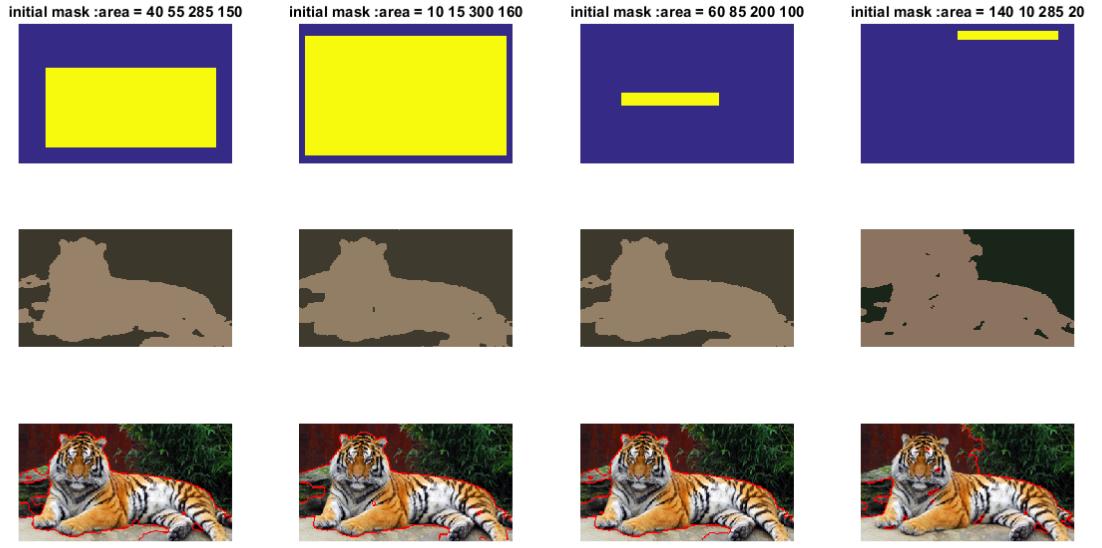


Figure 17: Results with different initialization

The initial rectangle gives the prior knowledge on the foreground and background. If the initialization is too large, it will include more pixels into foreground in the end and if too biased it will lead to converging to wrong object as foreground.

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 K-means and graph-cut have very similar approaches in the sense that they associate pixels to a class. However graph-cut makes use of a smooth association, while k-means makes use of a harsh association.

Both K-means and mean-shift take averages in order to find the segmentations. The graph-cut also takes a similar approach.

Normalized Cut and Graph-cut are graph-based methods. Mean-shift takes a "non-parametric" method to find the segments, while graph-cut takes a parametric method.