DD2423

Image Analysis and Computer Vision

Laboratory Report

Lab 3: Image Segmentation

Jiang, Sifan
sifanj@kth.se

December 8, 2018

1 K-meas clustering

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

  Since we have priori information with the image `orange` and based on the image, the main colors in the image are white, which is the background and the textures of the oranges, orange, which is the oranges themselves, and gray, which is the shadow of the oranges. So, we can set the centers based on these three colors.

  However, if we don't know the priori information and that's how we implement the code, the best way to initialize the clustering process is to chose the centers randomly.

- **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 number of clusters `K`, the number of the pixels, the initial cluster centers and the complexity of the image. In our case, the blurring degree could also affect the iteration amount.

  For `orange`, we typically need iteration time varies from 10 to 30 times to obtain a nice result of segmentation(clear boundary between two halves of the orange). The specific number of iteration would change according to the parameter setting.

- **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.

  From figure 1 and 2, the minimum value for `K` to get no superpixels that covers parts form both halves of the orange in my implementation is 17. When `K` is 16, there is no boundary between two halves of the orange, while if `K` is 17, there exists a clear boundary.

  In this case, I still use the default value for $\sigma = 1.0$ which is the image preblurring scale. If this value is smaller, then the minimum `K` to have obvious boundary between two halves of the orange could decrease.

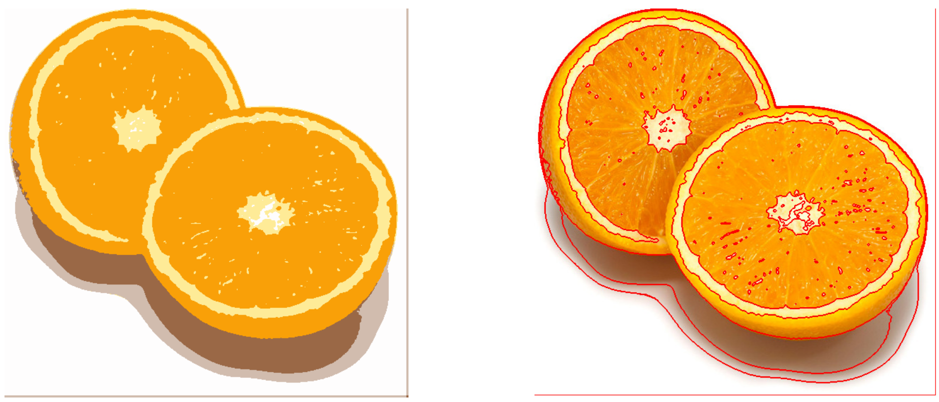


Figure 1: Segmented `orange` with 16 clusters.

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

  We need to increase the cluster number `K` since the images `tiger` are more complex than `orange` which have more color and details. Also, it would need more number of iterations, so the iteration time `L` should be increased at the same time.

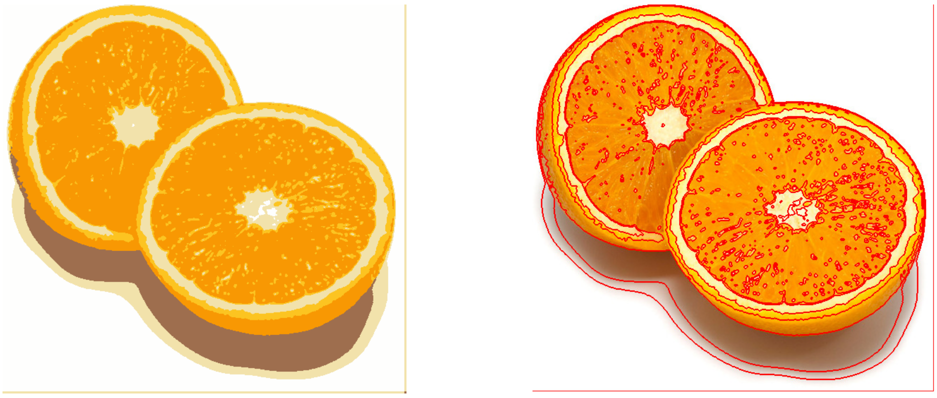


Figure 2: Segmented `orange` with 17 clusters.

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.

  The spatial bandwidth σ_s^2 is responsible for the range of the density function around pixels in 5 dimension spatial and color space. As σ_s^2 increases, the range of the Gaussian increases, thus the density function spread over more space and more likely to mix with other pixels. So, more pixels would be assigned to same mode and the number of modes decreases. Also, the spatial bandwidth determines the area of the segmentations. The larger the spatial bandwidth is, the larger the area of the segmentations could be, thus the less the number of modes would obtained.

  When the color bandwidth σ_c^2 increase, the diversity of color information of modes would decreases. Also, the overall color information would be more similar to the original image.

  For image `tiger_1`, the best parameters setting is spatial bandwidth equals to 4 and color bandwidth equals to 2, which is illustrated in Figure 3.

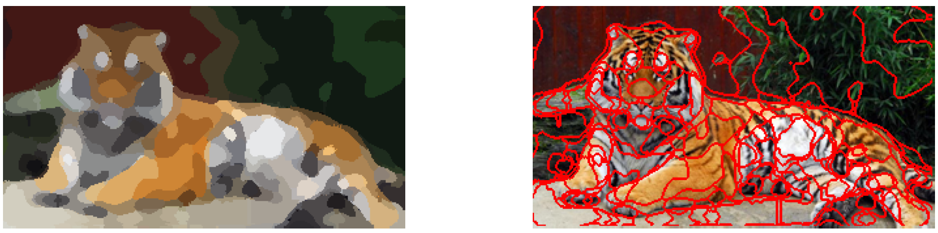


Figure 3: Best result for Mean-shift segmented `tiger_1` with spatial bandwidth 4 and color bandwidth 2.

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

  K-means and mean-shift segmentation are both treat with the color and position of pixels and are to find the cluster centers and modes.

However, mean-shift segmentation would not determine the number of modes before the segmentation start, which is different from K-means segmentation. Also, in our implementation, spatial information of the pixels is not taken into consideration in K-means segmentation. While mean-shift segmentation considers both spatial and color information.

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. Figure 4 shows the default parameters setting for Normalized Cut segmenting the image `tiger2`, with parameters noted in the title.

Figure 4: Normalized Cut segmented `tiger2` with default parameters setting: `colour_bandwidth` = 20.0, `radius` = 3, `ncuts_thresh` = 0.2, `min_area` = 200, and `max_depth` = 8.

`ncut_thresh` determines the color diversity of the result. The choice of `ncut_thresh` depends on the overall complexity of the image. As illustrated in Figure 5, the decrease of the maximum allowed value for a cut would result in the decrease the color diversity of the segmented image.

Figure 5: Normalized Cut segmented `tiger2` with changing `ncuts_thresh` to 0.02.

`min_area` determines the minimum size of each segmentation, so the choice of `min_area` depends on the size of the image, the size of the complexity of the objects in the image. As shown in Figure 6, the decrease of `min_area` would result in the decrease of segmented areas and more modes.

`max_depth` determines the maximum number of recursion to apply the implementation. As illustrated in Figure 7, the lower the `max_depth` is the less the segmentation would be and the less modes would the modes be found.

Also, as illustrated in Figure 8, the increase in `colour_bandwidth` would also result in the increase of modes found and the color complexity as well.

Figure 6: Normalized Cut segmented `tiger2` with changing `min_area` to 20.

Figure 7: Normalized Cut segmented `tiger2` with changing `max_depth` to 2.

Figure 8: Normalized Cut segmented `tiger2` with changing `colour_bandwidth` to 50.

In Figure 9, the best result for Normalized Cut segmentation of `tiger2` is shown. The parameters setting is `colour_bandwidth` = 20.0, `ncuts_thresh` = 0.2, `min_area` = 10, and `max_depth` = 16.

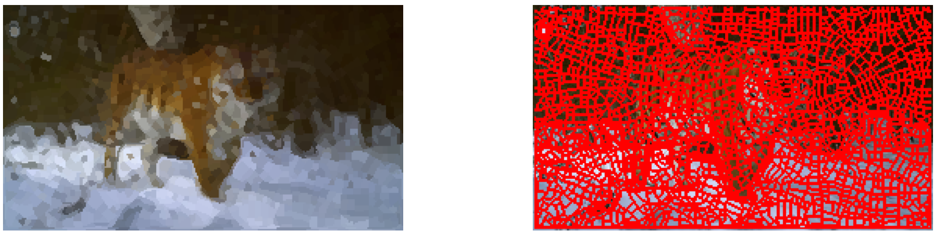


Figure 9: Best result of Normalized Cut segmented `tiger2` with parameters setting: `colour_bandwidth` = 20.0, `ncuts_thresh` = 0.5, `min_area` = 10, and `max_depth` = 16.

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

  `ncuts_thresh`, `min_area`, and `max_area` are all effective for reducing the subdivision. However, to get resulting image with a satisfactory segmentation, the parameters `ncuts_thresh` and `min_area` are the most effective.

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

  Normalized Cut prefers cuts of approximately equal size. Based on the equation:

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

  and the nature of symmetry, $\mathrm{Ncut}(A,B)$ would be minimized if $\mathrm{assoc}(A,\mathcal{V}) = \mathrm{assoc}(B,\mathcal{V})$. So if the pixels on the vertices $\mathcal{V}$ are similar to each other, the Normalized Cut would tend to cut the image to approximately equal size.

  In practice, if we set the parameter `colour_bandwidth` to a big value which means we would have larger tolerace on the difference of color, as illustrated in Figure 8, the image would be segmented into many equal sized parts. When this value is small, the image in general would not likely to be segmented into segements with equal size.

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

  Increasing the value of `radius` would reduce the subdivision of the segmented image and the segmentation is good which is shown in Figure 10 and 11 with comparison to Figure 9. What is expected is that, if we increase the `max_depth` and decrease the `min_area` when increasing the `radius` to get equal number of subdivision as the best result, the segmentation could be better.

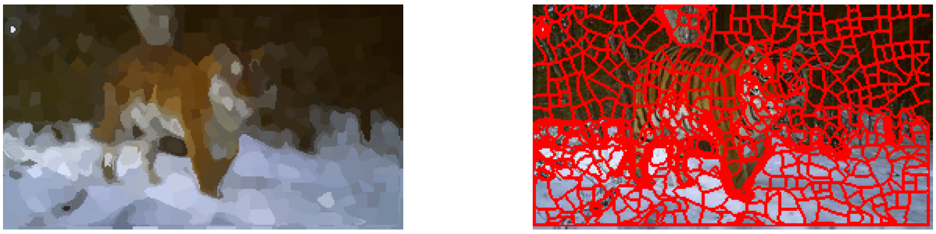


Figure 10: Best result of Normalized Cut segmented `tiger2` but change `radius` to 6.

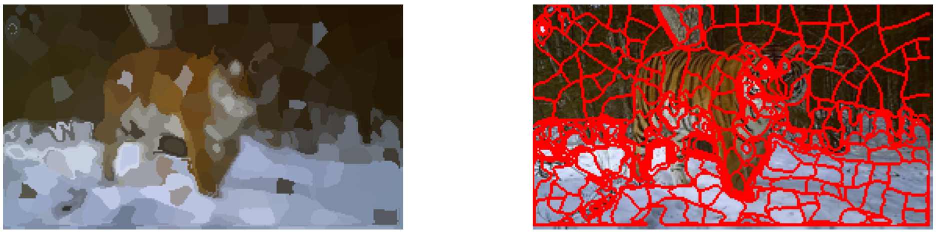


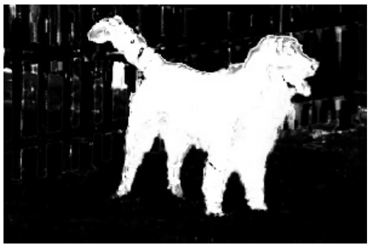
Figure 11: Best result of Normalized Cut segmented `tiger2` but change `radius` to 9.

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.

  The ideal choice of `alpha` and `sigma` doesn't vary a lot between different images. The ideal choice of `alpha` and `sigma` for image `tiger_3` are 10 and 16 correspondingly with result illustrated in Figure 12.

(a) Original image of `Tiger_3`.



(b) The likelihood for each pixels to be assigned to foreground.

(c) Threshold the likelihood to 0.5.

(d) Result of segmentation.

Figure 12: Segmentation of `tiger_3` using graph cuts with `alpha` 10 and `sigma` 16.

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

  When segmenting image `tiger_3`, even if `K` is lowered to 2, the result is still good enough for me.

  Take image `tiger_1` as the example in this case, when `K` is lowered to 7, the result is still good enough. While when `K` becomes to 6, the result gets considerably worse, which is shown in Figure 13.

(a) Result of segmentation when `K` is 7.

(b) Result of segmentation when `K` is 6.

Figure 13: Comparison between the result when `K` is 7 and 6.

- **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!

  The answer should depends on the context of what kind of segmentation we need. The

Graph Cut segmentation worth the effort to deal with the images having clear background and foreground. Also, the Graph Cut is only useful when there is on object.

However, when there is no obvious objects or difference between background and foreground, the Graph Cut does not worth all the changes.

- **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!!

  Differences:

  - The K-means segmentation method we use in this lab only find the similarity of the pixels in the color domain, thus the output of K-means segmentation are seperated into different areas.
  - Graph Cuts need prior information for better segmentation.
  - Normalized Cut and energy-based segmentation with Graph Cuts view the images as graphs.

  Similarities:

  - All the segmentation methods are to find the similarities among the pixels and put the similar pixels into one segmentation.