Answers to questions in

Lab 3: Image segmentation

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Program: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**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!

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Answers: There are different ways. You could randomly generate some colours, but then you might generate a cluster colour for which there is no similar pixel, such that that a cluster gets no pixels assigned to it. You could instead randomly generate some pixels and use their colours as initial cluster colours. However, then you might get some clusters with exactly the same colours, where only one cluster gets pixels assigned to it and the other clusters no pixel. A better solution is to randomly select a pixel and select the next pixel that is most distant to the first one in colour space. You add new colours by repeating the process always selecting the pixel that has the most different colour compared to the pixels you have so far selected.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**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: It depends on how you define convergence, how many clusters you have and what image you have. If you base it on how the end result changes, 10 iterations it often enough, but if you look at the mean square error, you might need many more.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**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: It varies a bit depending on how clusters were initiated. Something like K=7 clusters is reasonable.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Answers: Students might give very different answers, most of them being more or less correct, depending on the assumptions. The reason for the question is to force students to actually explore the parameter space.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**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: The bandwidths define how different two pixels can be and the method still considers them as “similar”. Small bandwidths lead to over-segmentation. Small bandwidths in spatial domain tend to make segments small and round. Small bandwidths in colour tend to create edges around real object boundaries, but easily lead to fragmentation. It’s not easy to find the best solution. High contrast images for example, need larger bandwidths in colour.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Answers: They are both iterative procedures in order to find high concentration of pixels. The purpose of K-means is to find a representation of the overall distribution, which is not really the same as the mean-shift that tries to maximize the density of points. You might thus get a cluster with K-means with few points assigned to it, with a low concentration of points, if that is necessary for K-means to cover the whole distribution. Unlike K-means that includes all points in the iteration, mean-shift does the iterations point by point, letting the original pixels initiate the iterations. With K-means pixels are clustered together if they will eventually belong to the same cluster in colour space, whereas with mean-shift the same is true if their respective iterations converge to the same mode (high density point). With mean-shift the points may in fact originate from quite different parts of the space. Many students will claim that one difference is that K-means only uses colours, while mean-shift also uses positions. This is true given how the methods are normally used for segmentation, but there is nothing that prevents K-means from being extended to also the spatial domain.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**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: Yes, indeed. The contrasts of images and the sizes of objects are different, which affects the parameter settings. Also here students might respond in different ways and there is no clear-cut right answer.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Answers: The theoretically best parameter is the cut threshold, but the maximum depth and minimum size are also practical, since they prevent fragmentation and limit the time it takes to compute.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Answers: The optimization function consists of two terms that each includes a division. Since the sum of the denominators is almost constant (the sum of the links with the links on the cut counted twice), you get the lowest total cost, if both denominators are of approximately the same size. This means that the two halves of the cut are often also about equal in size.

Assuming that the sum of cut links c is constant and s = a+b–c is the sum of all links.

f(a) = c/a + c/b = c/a + c/(s+c–a)

f’(a) = -c/a^2 + c/(s+c-a)^2 = 0 => a = s+c-a => a = b = (s+c)/2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Answers: It usually improves the results (and the computation time) up to a radius of about 5 to 7 pixels. For images that are slightly blurred, changes in brightness across an edge might be a couple of pixels wide. To capture such wide changes, it’s good to have a radius that covers the full width of the change. In fact, due to slight focal blur and motion blur, most images are slightly blurred. The difference between two nearby pixels is thus not a good indication of an edge.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**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: Alpha controls the maximum cost of a boundary pixel, while sigma controls how similar the colours need to be for the method to consider the pixels as “similar”. For low contrast images, you thus need a lower sigma, and the opposite for high contrast images. The alpha parameter directly controls the number of edges.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Answers: It can be as low as 2 or 3, which is quite impressive, but also quite logical. Colours are initially defined in RGB-space, e.g. in three dimensions. Thus with GMM K=3 should be necessary.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**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: In general, results can be considerably better with some limited human input. It depends on the application whether it’s worth it. For autonomous systems, it’s not practical at all. You would then need some detector to select a window, depending on what you intend to segment.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**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: The first three methods are image segmentation methods, with which images are divided into segments, hopefully corresponding to the real objects in the scene, while the last method is a figure-ground segmentation method, where you have a specific object of interest. K-means creates a very fragmented segmentation, which needs to be followed by some kind of merging process to be really of use. It’s more for data reduction than for segmentation per se. Mean-shift and Normalized Cut create larger segments that can be directly used for some applications. Normalized Cut and Graph Cuts uses a graph structure to capture spatial similarity, whereas Mean-shift uses a kernel. K-means does not take spatial similarity into consideration at all, at least not the implementation here. There is nothing that prevents K-means from being extended to the spatial domain though. K-means and mean-shift are iterative processes, whereas Normalized Cut solves and eigenvalue problem and Graph Cuts a maximum flow problem. The GMMs in the Graph Cut method are simply used to model the distribution of foreground and background pixels. They are used to create a prior, which is used in the energy-based formulation.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_