# Computer Vision

# Assignment 5 : Image Segmentation

Autumn 2018

Nikita Rudin

November 7, 2018

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Computer Vision
and Geometry Lab

# 1 Image Processing

The first task of this assignment is to pre-process the image in order to have better segmentation results. We first smooth the image with a Gaussian filter (fspecial() and imfilter()) and then convert it to L*a*b color (makecform() and applycform()) space that represents the image closer to the way a human perceives it. It is as such superior to RGB for segmentation. It is for example more robust to illumination changes because only the L parameter is affected.

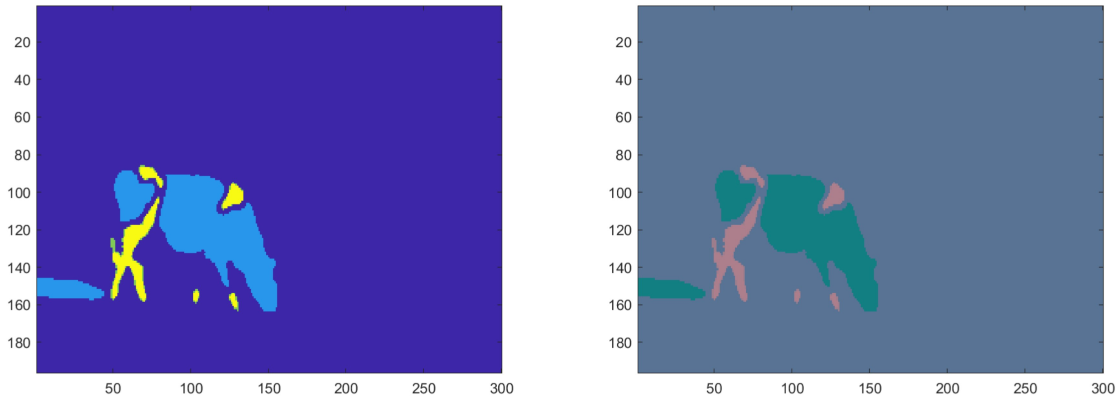
(a) Original image (b) Smoothed and converted image

Figure 1: Comparison of the original and processed images

# 2 Mean-Shift algorithm

We now implement the Mean-Shift algorithm to cluster the pixels in the L*a*b space. We flatten the image MxNx3 matrix to a Lx3 matrix where L=MxN. For each pixel we create a circular window and iteratively shift it towards the maximum density. The final center of that window is then the center assigned to the pixel. If this center is close to a previously detected center we group the pixels together. This provides us with a mapping of pixels to a small number of clusters.

We can see that the resulting image separates the image in 3 segments. Correctly separating the white and black parts of the cow from the background. We see some misclassification at the edges, where an edge of black and white is classified as background. It is worth noting that this method is computationally very heavy (takes around 6 min) and requires a fine tuning or the radius of the circle (here r=20 was used on not normalized points).

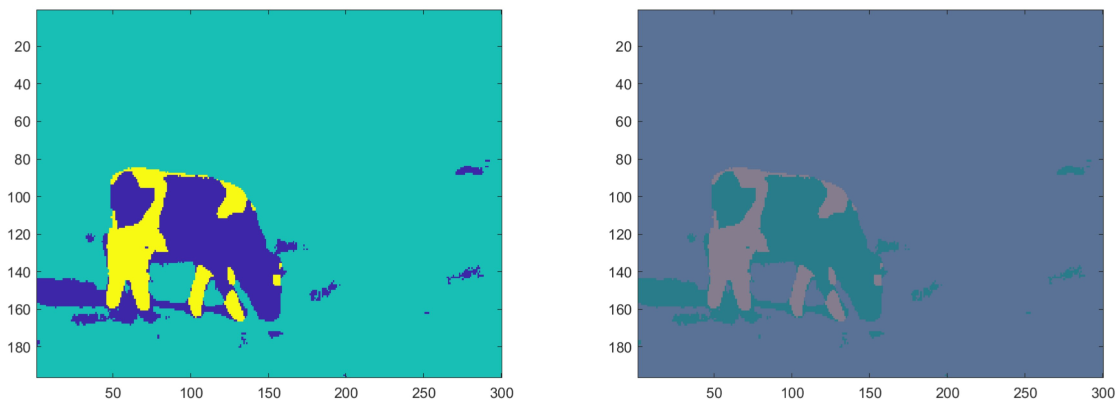



(a) Segments (b) Re-projected image

Figure 2: We can display the mapping to different clusters and re-project the image to L*a*b giving each segment the value of its center.

# 3 EM algorithm

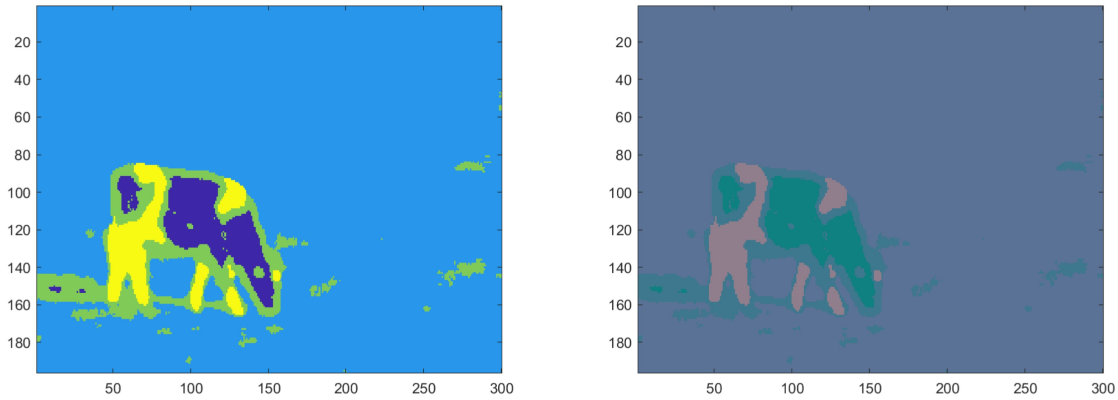
In this part we implement an EM algorithm to group our points into K clusters. Each cluster is represented by a gaussian with mean $\mu_i$ and covariance $\Sigma_i$. This time the number of clusters has to be fixed in advance we show results for K=3,4,5.




(a) Segments (b) Re-projected image

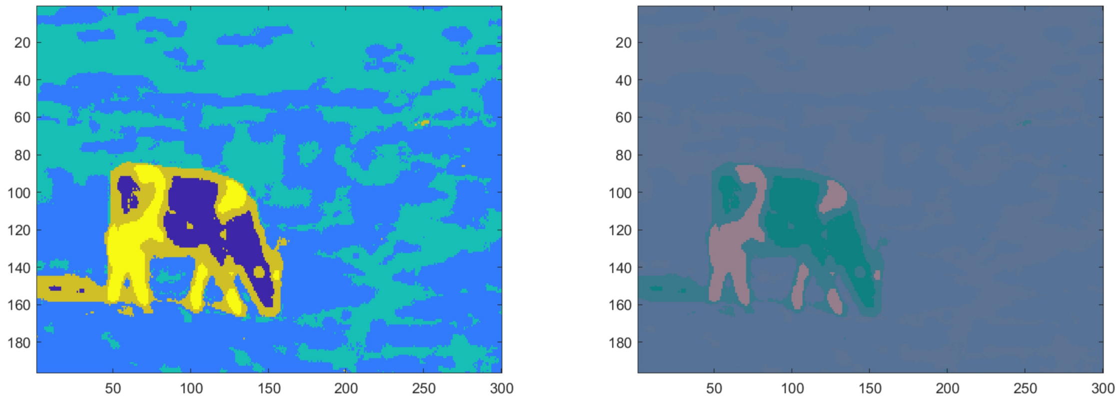
Figure 3: Segments and re-projected image to L*a*b space with K=3

The results provided by the K=3 case seem to be close to the previous algorithm. We can see a difference at the borders which are not misclassified any more. There are some errors in the background but in general the results seem good. This algorithm is much faster as it takes only 4.5 seconds to compute. A drawback is the fact that we need to specify the number of clusters which requires a user input. On the other hand we don't need to tune any other hyper-parameter. We also see that the results for K=4 and 5 are




(a) Segments (b) Re-projected image

Figure 4: Segments and re-projected image to L*a*b space with K=4

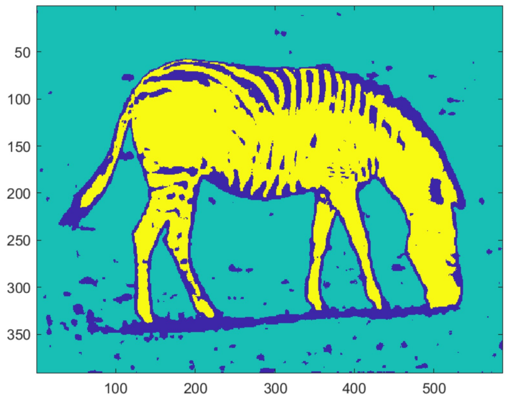



(a) Segments (b) Re-projected image

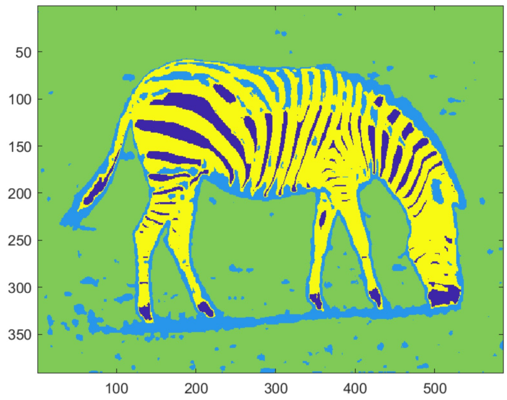
Figure 5: Segments and re-projected image to L*a*b space with K=5

worse. This is logical since we only have three main colours: black, white and green.

We can check the performance of the EM algorithm on the zebra image. We see that the results are reasonnable for K=3 and 4. The stripes seem to be a big challenge in this case.




(a) K=3




(b) K=4

Figure 6: Segments from EM applied to the zebra image

# 4 Matrices from EM (cow image)

Matrices for K=3

$$\mu_1 = \begin{pmatrix} 40.6354 \\ 123.7311 \\ 136.7329 \end{pmatrix} \Sigma_1 = \begin{pmatrix} 777.3157 & -134.1227 & 233.5033 \\ -134.1227 & 32.9492 & -47.3048 \\ 233.5033 & -47.3048 & 78.8975 \end{pmatrix}$$

$$\mu_2 = \begin{pmatrix} 89.107 \\ 114.4195 \\ 149.0968 \end{pmatrix} \Sigma_2 = \begin{pmatrix} 56.5438 & 0.35138 & 0.43407 \\ 0.35138 & 0.83987 & -0.17299 \\ 0.43407 & -0.17299 & 1.5386 \end{pmatrix}$$

$$\mu_3 = \begin{pmatrix} 132.6799 \\ 124.7418 \\ 140.8422 \end{pmatrix} \Sigma_3 = \begin{pmatrix} 2439.1055 & 89.3349 & 26.0992 \\ 89.3349 & 13.64 & -8.5564 \\ 26.0992 & -8.5564 & 24.906 \end{pmatrix}$$

$$\alpha = \begin{pmatrix} 0.10195 \\ 0.85686 \\ 0.041187 \end{pmatrix}$$

Matrices for K=4

$$\mu_1 = \begin{pmatrix} 14.2719 \\ 128.6195 \\ 128.3784 \end{pmatrix} \Sigma_1 = \begin{pmatrix} 9.2384 & 2.1976 & -1.4738 \\ 2.1976 & 1.774 & -0.65139 \\ -1.4738 & -0.65139 & 2.3785 \end{pmatrix}$$

$$\mu_2 = \begin{pmatrix} 89.1723 \\ 114.4261 \\ 149.0958 \end{pmatrix} \Sigma_2 = \begin{pmatrix} 55.5351 & 0.25009 & 0.38659 \\ 0.25009 & 0.80632 & -0.1623 \\ 0.38659 & -0.1623 & 1.5221 \end{pmatrix}$$

$$\mu_3 = \begin{pmatrix} 60.5788 \\ 120.2497 \\ 142.3695 \end{pmatrix} \Sigma_3 = \begin{pmatrix} 613.148 & -92.8004 & 160.2337 \\ -92.8004 & 29.7416 & -36.0265 \\ 160.2337 & -36.0265 & 56.1175 \end{pmatrix}$$

$$\mu_4 = \begin{pmatrix} 143.91 \\ 125.6787 \\ 140.2987 \end{pmatrix} \Sigma_4 = \begin{pmatrix} 2259.1299 & 54.8973 & 56.1233 \\ 54.8973 & 9.0315 & -5.0514 \\ 56.1233 & -5.0514 & 23.5656 \end{pmatrix}$$

$$\alpha = \begin{pmatrix} 0.037199 \\ 0.84992 \\ 0.079266 \\ 0.033618 \end{pmatrix}$$

Matrices for K=5

$$\mu_1 = \begin{pmatrix} 14.1778 \\ 128.6026 \\ 128.3601 \end{pmatrix} \Sigma_1 = \begin{pmatrix} 8.6996 & 2.0863 & -1.4971 \\ 2.0863 & 1.6914 & -0.63932 \\ -1.4971 & -0.63932 & 2.3162 \end{pmatrix}$$

$$\mu_2 = \begin{pmatrix} 85.7697 \\ 114.3583 \\ 149.6741 \end{pmatrix} \Sigma_2 = \begin{pmatrix} 72.6749 & -0.29218 & 5.0226 \\ -0.29218 & 1.3212 & -0.12034 \\ 5.0226 & -0.12034 & 1.7548 \end{pmatrix}$$

$$\mu_3 = \begin{pmatrix} 92.8158 \\ 114.5311 \\ 148.4151 \end{pmatrix} \Sigma_3 = \begin{pmatrix} 16.8071 & 0.1423 & -0.072743 \\ 0.1423 & 0.38197 & -0.02447 \\ -0.072743 & -0.02447 & 0.46352 \end{pmatrix}$$

$$\mu_4 = \begin{pmatrix} 56.8294 \\ 122.129 \\ 139.7748 \end{pmatrix} \Sigma_4 = \begin{pmatrix} 726.2817 & -86.188 & 153.6748 \\ -86.188 & 24.6214 & -29.5306 \\ 153.6748 & -29.5306 & 47.2296 \end{pmatrix}$$

$$\mu_5 = \begin{pmatrix} 150.5054 \\ 126.0604 \\ 140.2485 \end{pmatrix} \Sigma_5 = \begin{pmatrix} 2051.2459 & 40.7969 & 56.9538 \\ 40.7969 & 7.1761 & -3.9301 \\ 56.9538 & -3.9301 & 23.4509 \end{pmatrix}$$

$$\alpha = \begin{pmatrix} 0.036422 \\ 0.48567 \\ 0.38368 \\ 0.064113 \\ 0.030116 \end{pmatrix}$$