Computer Vision
Homework Assignment 5 : Image Segmentation

Autumn 2018

Nicolas Marchal

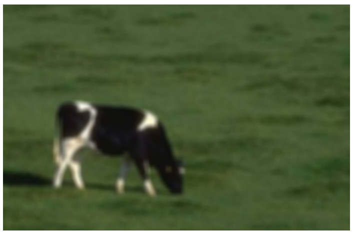
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Computer Vision
and Geometry Lab

1 Image preprocessing

Before doing the Image segmentation, I preprocess the original image (fig. 1a) such that it is smoother (less sensitive to noise). I used a 5x5 Gaussian filter with $\sigma = 5.0$ (fig. 1b). I then convert the image to L^*a^*b space (fig. 1c), which is better than RGB for image segmentation. L^*a^*b is a conversion from RGB (3 color components) to a lightness component L*, and two color components - a* and b*. L^*a^*b is therefore less sensitive to different lightening of the same color, as the brightness affect only the L value (whereas it would affect the three components on a RGB image).

(a) Original Image



(b) Smooth Image

(c) Smooth l-a-b Image

Figure 1: Image Preprocessing

2 Mean-Shift Segmentation

I will now use Mean-Shift Segmentation, which repeatedly computes the mean of all the pixels that lie within a spherical window or radius r shifting this window to the mean until convergence (i.e the nest shift is less than a *threshold*). The two parameters to tune will therefore be :

- Radius: r
- Threshold: $threshold$

I decided to normalize my images such that in the 3D L^*a^*b space their centroid is at the origin and their average distance to the origin is $\sqrt{3}$. I have done this to avoid randomly testing many parameters and eventually finding correct ones (note: I had already written the matlab function `normalisation` in the first homework of this class). By normalising my point, I know that the points lies on average at $\sqrt{3}$ from the origin so I can intuitively set a radius smaller than $\sqrt{3}$. In practice, $r = \frac{2}{3}\sqrt{3}$ works well. The threshold is given in the exercise instructions : $threshold = r/2$.

This algorithm is computationally expensive so I made sure to avoid using for loops in the Matlab code. The computation time varies between computers and depends on the radius as well as the image size. For the testing, I reduced by a factor of 2 the image size to have the algorithm running under a minute. When running the entire picture, it can take several minutes, but the progress is shown on the command window (as well as the number of peaks detected, see fig. 2a).

Fig. 2 shows the result of my algorithm using

- Radius: $r = \frac{1}{3}\sqrt{3}$ and $r = \frac{2}{3}\sqrt{3}$
- Threshold: $threshold = r/2$

The strength of this algorithm is that it automatically finds 8 (or 15, depending on the radius size) number of clusters, without needing to specify this number.

(a) Command Window

(b) $r = \frac{2}{3}\sqrt{3}$, 8 clusters

(c) $r = \frac{1}{3}\sqrt{3}$, 15 clusters

Figure 2: Mean Shift

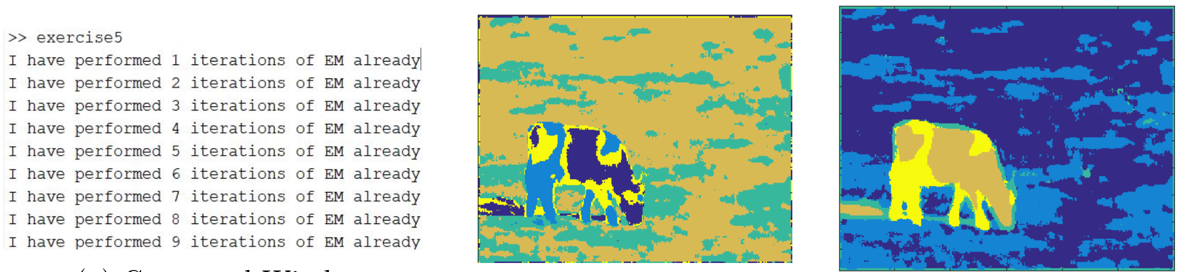
3 EM Segmentation

The last part of this exercise is to implement the Expectation Maximisation (EM) algorithm. In this method we need to chose a number of clusters K (I will evaluate the performance for K = 3,4,5 but I will first talk about the general results).

The first step was to initialise the parameters α, Σ and μ. I followed the method presented in the project slides :

- α : $\alpha_k = \frac{1}{K}$ for all k (same weight at initialization).
- Σ : 3x3 diagonal matrix, with elements corresponding to the range of L*, a*, b*.
- μ : The project slides suggest to initialize all the μ_k equally spread in the $L*a*b$ space. I took a similar approach and I distributed the μ_k uniformly over the $L*a*b$ space (stochastic process).

I then implemented the formulas to evaluate the probability in `expectation` and update the parameters in `maximization`. I set a tolerance that stops the EM algorithm when the difference of the means before and after maximization is lower than a threshold. Just like for the mean shift, I normalized my images. I decided to set the threshold to stop the EM algorithm at $\frac{1}{20} * sqrt(3)$ because we can see on fig. 3 (K=5) that this value identifies better the white area on the cow.



(a) Command Window

(b) $tol = \frac{1}{15}\sqrt{3}$, 9 EM steps

(c) $tol = \frac{1}{20}\sqrt{3}$, 15 EM steps

Figure 3: EM results

3.1 K=3

$$\mu_1 = \begin{bmatrix}123.1 & 123.4 & 142.5\end{bmatrix}^T \quad \mu_2 = \begin{bmatrix}89.4 & 114.5 & 149.1\end{bmatrix}^T \quad \mu_3 = \begin{bmatrix}46.0 & 121.7 & 138.7\end{bmatrix}^T$$

$$\alpha = \begin{bmatrix}0.051 & 0.810 & 0.139\end{bmatrix}$$

$$\Sigma_1 = \begin{bmatrix} 4.6013 & 0.2510 & -0.0892 \\ 0.2510 & 0.0377 & -0.0318 \\ -0.0892 & -0.0318 & 0.0629 \end{bmatrix} \qquad \Sigma_2 = \begin{bmatrix} 0.1018 & -0.0001 & 0.0010 \\ -0.0001 & 0.0015 & -0.0003 \\ 0.0010 & -0.0003 & 0.0029 \end{bmatrix}$$

$$\Sigma_3 = \begin{bmatrix} 1.2979 & -0.2684 & 0.3916 \\ -0.2684 & 0.0727 & -0.0938 \\ 0.3916 & -0.0938 & 0.1337 \end{bmatrix}$$

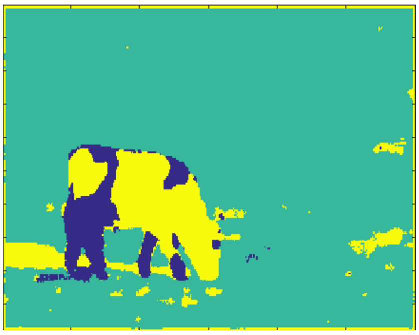



Figure 4: EM results : K = 3

3.2 K=4

$$\mu_1 = \begin{bmatrix}127.2 & 124.1 & 141.7\end{bmatrix}^T \qquad \mu_2 = \begin{bmatrix}92.2 & 114.52 & 148.9\end{bmatrix}^T$$

$$\mu_3 = \begin{bmatrix}85.1 & 114.4 & 149.4\end{bmatrix}^T \qquad \mu_4 = \begin{bmatrix}41.8 & 122.8 & 137.2\end{bmatrix}^T$$

$$\alpha = \begin{bmatrix}0.046 & 0.461 & 0.372 & 0.121\end{bmatrix}$$

$$\Sigma_1 = \begin{bmatrix} 4.6972 & 0.2104 & -0.0288 \\ 0.2104 & 0.0318 & -0.0250 \\ -0.0288 & -0.0250 & 0.0572 \end{bmatrix} \qquad \Sigma_2 = \begin{bmatrix} 0.0489 & -0.0006 & 0.0004 \\ -0.0006 & 0.0008 & -0.0002 \\ 0.0004 & -0.0002 & 0.0024 \end{bmatrix}$$

$$\Sigma_3 = \begin{bmatrix} 0.1391 & -0.0003 & 0.0062 \\ -0.0003 & 0.0029 & -0.0004 \\ 0.0062 & -0.0004 & 0.0036 \end{bmatrix} \qquad \Sigma_4 = \begin{bmatrix} 1.2045 & -0.2360 & 0.3524 \\ -0.2360 & 0.0626 & -0.0815 \\ 0.3524 & -0.0815 & 0.1187 \end{bmatrix}$$


Figure 5: EM results : K = 4

3.3 K=5

$$\mu_1 = \begin{bmatrix}90.4 & 114.6 & 149.8\end{bmatrix}^T \mu_2 = \begin{bmatrix}39.6 & 123.3 & 136.5\end{bmatrix}^T \mu_3 = \begin{bmatrix}82.9 & 115.0 & 149.0\end{bmatrix}^T$$

$$\mu_4 = \begin{bmatrix}125.0 & 123.6 & 142.5\end{bmatrix}^T \qquad \mu_5 = \begin{bmatrix}89.5 & 114.3 & 148.7\end{bmatrix}^T$$

$$\alpha = \begin{bmatrix}0.228 & 0.112 & 0.112 & 0.0479 & 0.4998\end{bmatrix}$$

$$\Sigma_1 = \begin{bmatrix}0.0951 & -0.0049 & 0.0036 \\ -0.0049 & 0.0014 & -0.0006 \\ 0.0036 & -0.0006 & 0.0037\end{bmatrix} \qquad \Sigma_2 = \begin{bmatrix}1.1572 & -0.2273 & 0.3404 \\ -0.2273 & 0.0645 & -0.0818 \\ 0.3404 & -0.0818 & 0.1174\end{bmatrix}$$

$$\Sigma_3 = \begin{bmatrix}0.2006 & -0.0103 & 0.0189 \\ -0.0103 & 0.0030 & -0.0030 \\ 0.0189 & -0.0030 & 0.0077\end{bmatrix} \qquad \Sigma_4 = \begin{bmatrix}4.5678 & 0.2474 & -0.0924 \\ 0.2474 & 0.0354 & -0.0303 \\ -0.0924 & -0.0303 & 0.0628\end{bmatrix}$$

$$\Sigma_5 = \begin{bmatrix}0.0952 & 0.0039 & -0.0014 \\ 0.0039 & 0.0014 & -0.0002 \\ -0.0014 & -0.0002 & 0.0015\end{bmatrix}$$

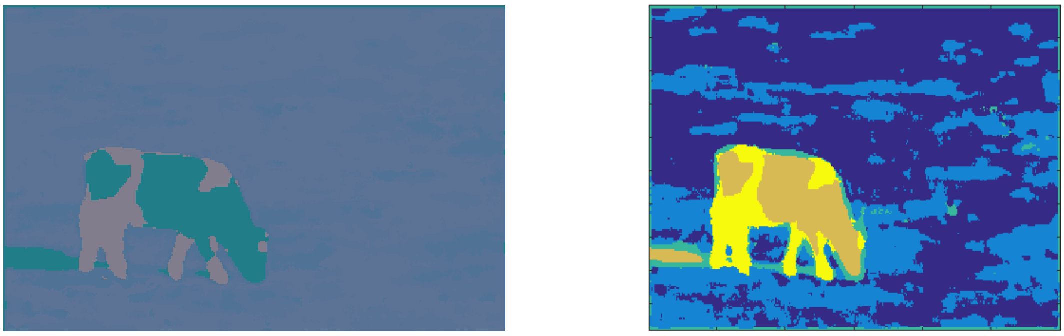


Figure 6: EM results : K = 5

3.4 Zebra

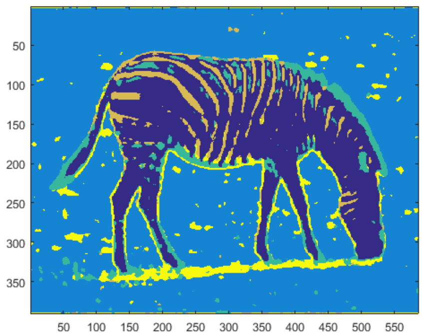


Figure 7: K=5

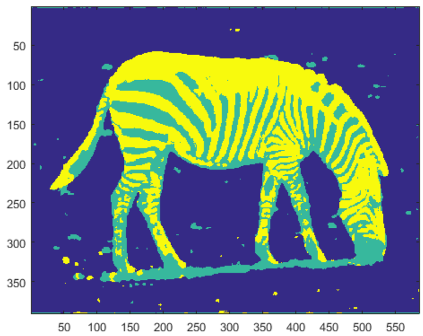


Figure 8: K=5