ETH Zürich
Computer Science Department

Prof. Marc Pollefeys
Prof. Luc Van Gool
HS 2015

Computer Vision

Exercise 6

Hand-out: 29-10-2015
Hand-in: 05-11-2013 14:00

Objective:
In this exercise you will implement image segmentation with the mean-shift and Expectation-Maximization (EM) approaches. Image segementation will be done in the $L^*a^*b^*$ color space.

6.1 Image Preprocessing (Total: 10%)

Image segmentation is usually done by clustering an n-dimensional data set obtained from an image where n is determined by the type of feature we choose from the image. In this assignment, we shall use the $L^*a^*b^*$ color space ($n = 3$) as the feature for image segmentation.

a) It is neccessary to smooth the image so that the color becomes more uniform before image segmentation. Load the given image "cow.jpg" into Matlab and apply a $5x5$ Gaussian filter with $\sigma = 5.0$ to smooth the image. (**5**%)

b) Using the functions "makecform" and "applycform" from Matlab, convert the image from RGB to $L^*a^*b^*$ color space. Explain in your report why is it better to do segmentation in the $L^*a^*b^*$ color space as compared to RGB color space.(**5**%)

6.2 Mean-Shift Segmentation (Total: 20%)

The Mean Shift algorithm clusters the 3-dimensional $L^*a^*b^*$ color space by finding the mode of the density function for each pixel where all the pixels with the $L^*a^*b^*$ values form the discrete samples of this density function. Mean-shift algorithm repeatedly computes the mean of all the pixels that lies within a spherical window of radius r and shifting this window to the mean until convergence (i.e. the next shift is less than a threshold).

a) Implement the algorithm to find the mode of the density function for a given pixel x_l as the function:

$$function\ \ peak = find\_peak(X, x_l, r)$$

where X is the discrete samples of the density function and it is a matrix with size $L \times n$. L is the total number of pixels in the image and $n = 3$ for the $L^*a^*b^*$ value. (**10**%)

b) Now implement the mean-shift algorithm that calls the $find\_peak$ function written in (a) for each pixel in X. A check on the distances between the peaks should be done after processing each pixel. Peaks that have distance lesser than $r/2$ should be merged together. The mean-shift function call should be:

$$function \; [map, peaks] = mean\_shift(X, r)$$

where map is a matrix with the size of the image and it holds the id of the associated peak for each pixel. Display the segmentation result with an unique color for each pixel associated with the same peak. (**10**%)

6.3 EM Segmentation (Total: 70%)

Image segmentation can also be done probilistically with the EM algorithm. Here, we assume that the number of segments K is known. Each of these segments is modeled as a Gaussian with parameters $\theta_k = (\mu_k, \Sigma_k)$ and together these segments form a Gaussian mixture model with each segment weighted by a mixing weight α_k The task is to find out these parameters $\Theta = (\alpha_1, ..., \alpha_K, \theta_1, ..., \theta_k)$ given the observations X. Where X was defined previously as a $L \times 3$ matrix with the $L^*a^*b^*$ color values from the image. Formally, we define the Gaussian mixture model for each pixel x_l as

$$p(x_l|\Theta) = \sum_{k=1}^{K} \alpha_k p(x_l|\theta_k) \tag{1}$$

where each component density is the usual Gaussian

$$p(x_l|\theta_k) = \frac{1}{(2\pi)^{n/2} |\Sigma_k|^{1/2}} \exp -\frac{1}{2}(x_l - \mu_k)^T \Sigma_k^{-1} (x_l - \mu_k) \tag{2}$$

EM solves for Θ by alternating between the Expectation and Maximization steps. In the Expectation step, we compute the probability I that x is in segment k, given current guess of Θ. In the Maximization step, we maximize the expectation of the complete log likelihood $p(X|\Theta)$ under I over the parameters Θ. This process is done repeatedly until convergence.

a) Implement a function to compute the probability that x is in segment k, given current guess of Θ. The function call should be

$$function \; I = expectation(\Theta, X, K)$$

Take note that $\theta_{1:k}$ should be initialized to random values within the range of the $L^*a^*b^*$ values in X, and $\alpha = 1/K$ which indicates uniform weightage for all segments. Fix $K = 3$ in your implementation. (**30**%)

b) With the probability I from the previous step known, implement the Maximization step to compute the new values of Θ. Your function call should be

$$function \; \Theta = maximization(I, X, \mu_{1:K}, K)$$

where I is the expected values from the previous step. (**30**%)

c) Now write the code to iterate between the Expectation and Maximization steps until convergence. Show the final values for Θ in your report. (**5**%)

d) Repeat steps (a)-(c) for $K = 4$ and 5. Report the values for Θ. (**5**%)

6.4 Hand in:

Write a short report explaining the main steps of your implementation and discussing the results of the methods. Send the report together with your source code to **fede@inf.ethz.ch**.