Computer Vision and Geometry Lab

Computer Vision

Exercise Session 6

Assignment 6

- 3 Tasks:
  - Preprocess the image for image segmentation in the L*a*b* color space.
  - Implement the mean-shift algorithm for image segmentation.
  - Implement the Expectation-Maximization algorithm for image segmentation.

Preprocessing the Image

- Smoothing the image
  - Use a 5x5 Gaussian kernel with $\sigma = 5.0$.
  - Use matlab “fspecial” and “imfilter” functions.
- Converting RGB to L*a*b* color space
  - Use matlab “makecform” and “applycform” functions.

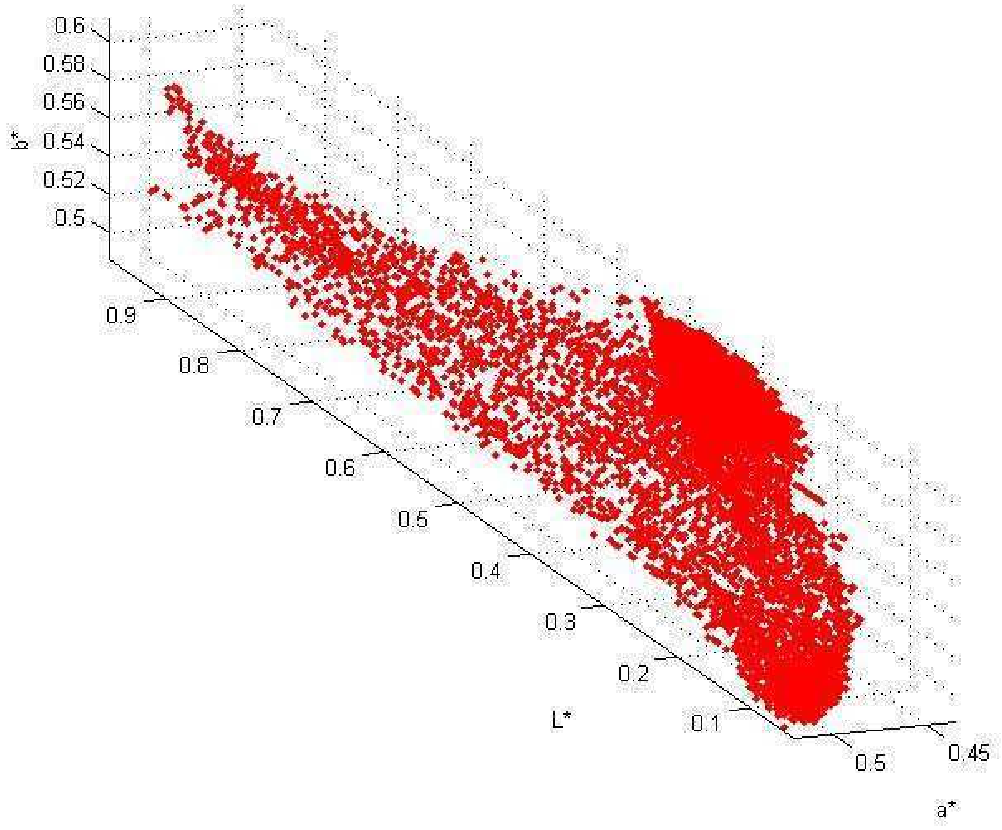
original image

smoothed image

l*a*b* image

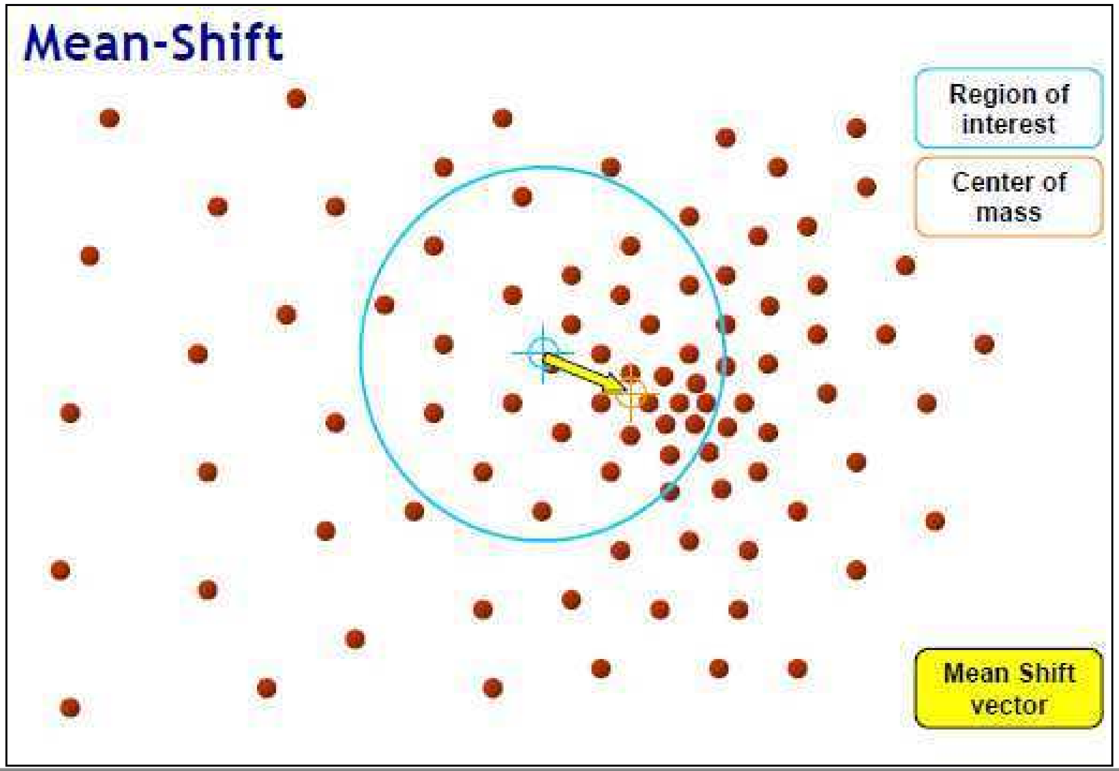
Mean-Shift Segmentation

- Create the density function X in the L*a*b* space.
  - L x 3 matrix, where L = no. of pixels in the image and 3 for the dimension of the L*a*b* space.



Mean-Shift Segmentation

- For each pixel
  - Compute the mean of all the pixels that lie within a spherical window of radius r (e.g. r=0.03).
  - Shift this window to the mean.
  - Repeat until convergence.



EM Segmentation – Color Model

- We assume that the colors originate from a probability density given by a Gaussian Mixture Model (GMM):

$$p(\mathbf{x}) = \sum_{k=1}^{K} \alpha_k N(\mathbf{x} | \mu_k, \Sigma_k) = \sum_{k=1}^{K} \alpha_k \frac{1}{(2\pi)^{n/2} \det(\Sigma_k)^{1/2}} \exp(-\frac{1}{2}(\mathbf{x} - \mu_k)^T \Sigma_k^{-1} (\mathbf{x} - \mu_k))$$

α_k : prior probability of component k

Σ_k: covariance of the k-th component

μ_k : mean of the k-th component

EM Segmentation – Latent Variables

- Observed variables (colors) $\mathbf{X} = \{\mathbf{x}_l, l = 1..L\}$
- Latent (unobserved) variables $Z = \{z_l, l = 1..L\}$
  - Each sample is drawn from a component of the Gaussian mixture, but we do not observe/know from which
  - $z_l = k$ means that pixel l belongs to component k and

$$p(\mathbf{x}_l | z_l = k) = N(\mathbf{x}_l | \mu_k, \Sigma_k)$$

EM Segmentation – Latent Variables

- Log of the likelihood function after marginalizing over latent variables:

$$\ln p(\mathbf{X} | \alpha, \mu, \Sigma) = \sum_{l=1}^{L} \ln \left\{ \sum_{k=1}^{K} \alpha_k N(\mathbf{x}_l | \mu_k, \Sigma_k) \right\}$$

- Difficult to maximize w.r.t. model parameters (sum inside logarithm)

EM Segmentation – Latent Variables

- Consider instead the maximization of the log of the ‘complete’ data likelihood (observed AND unobserved data)

$$\ln p(\mathbf{X}, Z | \alpha, \mu, \Sigma) = \ln\left\{ \prod_{l=1}^{L} \prod_{k=1}^{K} \alpha_k^{I_k(z_l)} N(\mathbf{x}_l | \mu_k, \Sigma_k)^{I_k(z_l)} \right\}$$

$$= \sum_{l=1}^{L} \sum_{k=1}^{K} I_k(z_l) \{ \ln \alpha_k + \ln N(\mathbf{x}_l | \mu_k, \Sigma_k) \}$$

over α, μ and Σ.

$I_k(z_l)$ is an indicator function which returns value 1 if $z_l = k$ and 0 otherwise.

EM Segmentation – Latent Variables

- Now, the maximization step can be given in closed form
- Unfortunately, Z is not observed
- But we can maximize the expected value of the complete log-likelihood and iterate expectation and maximization steps:

$$\max_{\mu,\Sigma,\alpha} \mathrm{E}\left[\ln p(\mathrm{X},\mathrm{Z}|\mu,\Sigma,\alpha)\right]_{P(\mathrm{Z}|\mathbf{X},\mu,\Sigma,\alpha)}$$

$$= \max_{\mu,\Sigma,\alpha} \sum_{l=1}^{L} \sum_{k=1}^{K} \mathrm{P}(z_l = k|\mathbf{X},\mu,\Sigma,\alpha)\left\{\ln \alpha_k + \ln N(\mathbf{x}_l|\mu_k,\Sigma_k)\right\}$$

EM Segmentation – Expectation Step

- Determine $P(Z|\mathbf{X},\mu,\Sigma,\alpha)$ i.e. the posterior probabilities for Z

$$\gamma_{lk} = P(z_l = k|\mathbf{X},\mu,\Sigma,\alpha) = \frac{\alpha_k N(x_l \mid \mu_k, \Sigma_k)}{\sum_{k=1}^{K} \alpha_k N(x_l \mid \mu_k, \Sigma_k)}$$

- The probabilites can be stored in an LxK matrix.

EM Segmentation – Maximization Step

- Example α
  - Condition $\sum_{k=1}^{K} \alpha_k = 1$ has to be fulfilled
  - Use Lagrange multiplier:

$$f(\alpha, \lambda) = \mathrm{E}\left[\ln p(\mathrm{X}, \mathrm{Z} | \mu, \Sigma, \alpha)\right]_{P(\mathrm{Z}|\mathbf{X}, \mu, \Sigma, \alpha)} + \lambda(\sum_{k=1}^{K} \alpha_k - 1)$$

$$= \sum_{l=1}^{L} \sum_{k=1}^{K} \gamma_{lk} \left\{\ln \alpha_k + \ln N(\mathbf{x}_l | \mu_k, \Sigma_k)\right\} + \lambda(\sum_{k=1}^{K} \alpha_k - 1)$$

  - Solve $\nabla_{\alpha,\lambda} f = 0$ which leads to $\alpha_k = \frac{1}{L} \sum_{l=1}^{L} \gamma_{lk}$

EM Segmentation – Maximization Step

- Use the same approach for μ and Σ (in contrast to α no additional condition has to be fulfilled)
- Solution of the maximization for all model parameters:

$$\alpha_k = \frac{1}{L}\sum_{l=1}^{L}\gamma_{lk} \qquad \mu_k = \frac{\sum_{l=1}^{L} x_l \gamma_{lk}}{\sum_{l=1}^{L}\gamma_{lk}} \qquad \Sigma_k = \frac{\sum_{l=1}^{L}\gamma_{lk}\left\{(x_l - \mu_k^{(s+1)})(x_l - \mu_k^{(s+1)})^T\right\}}{\sum_{l=1}^{L}\gamma_{lk}}$$

EM Segmentation - Initialization

- Initialization:
  - Need to initialize $\alpha_{1:k}^{(0)}$, $\Sigma_{1:k}^{(0)}$ and $\mu_{1:k}^{(0)}$.
  - $\alpha_k^{(0)} = \frac{1}{K}$ which gives every segment the same weight at initialization.
  - $\Sigma_k^{(0)}$ is a 3x3 matrix and could be initialized as a diagonal matrix with elements corresponding to the range of the L*, a* and b* values.
  - Each of $\mu_{1:k}^{(0)}$ is a 3x1 vector that represents a point in the L*a*b* space. A possible way to initialize $\mu_{1:k}^{(0)}$ is to spread them equally in the L*a*b* space.

EM Segmentation - Algorithm

- Initialize $\alpha^{(0)}$, $\Sigma^{(0)}$ and $\mu^{(0)}$
- Iterate (s indicates current step)
  - Expectation step: calculate ${\gamma_{lk}}^{(s)}$ using $\alpha^{(s)}$, $\mu^{(s)}$ and $\Sigma^{(s)}$
  - Maximization step: update α, μ and Σ

$$ {\gamma_{lk}}^{(s)} = \frac{\alpha_k^{(s)} N(x_l \mid \mu_k^{(s)}, \Sigma_k^{(s)})}{\sum_{k=1}^{K} \alpha_k^{(s)} N(x_l \mid \mu_k^{(s)}, \Sigma_k^{(s)})} \qquad \alpha_k^{(s+1)} = \frac{1}{L} \sum_{l=1}^{L} {\gamma_{lk}}^{(s)} $$

$$ \mu_k^{(s+1)} = \frac{\sum_{l=1}^{L} x_l {\gamma_{lk}}^{(s)}}{\sum_{l=1}^{L} {\gamma_{lk}}^{(s)}} \qquad \Sigma_k^{(s+1)} = \frac{\sum_{l=1}^{L} {\gamma_{lk}}^{(s)} \left\{ (x_l - \mu_k^{(s+1)})(x_l - \mu_k^{(s+1)})^T \right\}}{\sum_{l=1}^{L} {\gamma_{lk}}^{(s)}} $$

EM Segmentation

- A detailed description and justification of the EM algorithm can be found in the book
  ‘**Pattern Recognition and Machine Learning**’
  by **Christopher M. Bishop**

Matlab hints

- Avoid for-loops!
  - Use element-wise operations (.*, .^, etc.) on matrices, repmat, sum, etc.
  - Reduce image size for debugging
  - See e.g. visualizeMostLikelySegments.m
  - Very helpful for mean-shift segmentation…