ECE4580 Homework #12

Due: Apr. 17, 2014

Problem 1. (25 pts) Implement the k-means algorithm for image segmentation. Of course initialization is always a problem, so for the images provided give the initial guess of means to start out with.

Recall that the energy to minimize is:

$$\mathcal{E}(\mu, s) = \int_D \left|\left|I(\mathbf{x}) - \mu_{\mathbf{s}(\mathbf{x})}\right|\right|_\Sigma^2 \, \mathrm{d}\mathbf{x} \, ,$$

where $I : D \to \mathbb{R}$ is the image proper, $\mu = (\mu_1, \ldots, \mu_k)$ are the means, and $s : D \to \{1, \ldots, k\}$ is the segmentation map. Usually, Σ is one or the identity matrix. It will be an optional argument for the function.

The algorithm is:

1. Start with initial guess of means.
2. Using means, segment by minimizing error energy.
3. Recompute means based on the segmentation.
4. Repeat (2)-(3) until segmentation does not change or maximum iterations hit.

The function stub is called `segKmeans.m`, and the script file you should prepare is called `segmentK.m`. As usual, it should select two of the images from the given Matlab file to process and go for it. Turn in the code and the results. Explain the output and your selection (image plus initial conditions).

Problem 2. (35 pts) Implement the Iterative Conditioning Mode algorithm for image segmentation, which is essentially k-means with regularization. Of course initialization is always a problem, so for the images provided give the initial guess of means to start out with.

Recall that the energy to minimize is:

$$\mathcal{E}(\mu, s) = \int_D \left|\left|I(\mathbf{x}) - \mu_{\mathbf{s}(\mathbf{x})}\right|\right|_\Sigma^2 \, \mathrm{d}\mathbf{x} + \lambda \int_{\mathbf{D}} \int_{\mathcal{N}(\mathbf{x})} \left(\mathbf{1} - \delta_{\mathbf{1}}(\mathbf{s}(\mathbf{x}), \mathbf{s}(\mathbf{y}))\right) \mathrm{d}\mathbf{y} \; \mathrm{d}\mathbf{x}$$

where $I : D \to \mathbb{R}$ is the image proper, $\mu = (\mu_1, \ldots, \mu_k)$ are the means, $s : D \to \{1, \ldots, k\}$ is the segmentation map, and δ_1 is the Kronecker delta function.

One algorithm is:

1. Starting with initial guess of means, generate segmentation by using standard k-means (w/out regularization).
2. Using means, update segmentation by minimizing error energy.
   - Here, this involves minimizing both the matching energy and the neighbor disagreement penalty (e.g., the regularization term).
3. Recompute means based on the segmentation.
4. Repeat (2)-(3) until segmentation does not change or maximum iterations hit.

The function stub is called `segICM.m`, and the script file you should prepare is called `segmentI.m`. Select two of the images from the Matlab file to process and go for it. Turn in the code and the results. Explain the output and your selection. Compare to just k-means (one way is to set lambda to zero or to perform no iterations and run `segICM`).

Problem 3. (50 pts) Implement the Bayesian relaxation algorithm for image segmentation. This requires not only the means, but also the standard deviations associated to the means. Allow for two options in the invocation, (i) run for a number of iterations as decided by user, and (ii) run to convergence based on a negative argument for the number of iterations.

Recall that the energy to maximize is:

$$\mathcal{E}(\mu, s) = \int_D \mathcal{N}\left(I(\mathbf{x}); \mu_{\mathbf{s}(\mathbf{x})}, \mathbf{\Sigma}_{\mathbf{s}(\mathbf{x})}\right) \,\mathrm{d}\mathbf{x} - \lambda \int_{\mathbf{D}} ||\nabla \mathbf{P}(\mathbf{x})||_{\mathbf{2}}^{\mathbf{2}} \,\mathrm{d}\mathbf{x}$$

where $I : D \to \mathbb{R}$ is the image proper, the pairings (μ_c, Σ_c) for $c = \{1 \dots k\}$ are the class means and covariances, $s : D \to \{1, \dots, k\}$ is the segmentation map, and $\mathcal{N}(\cdot; \mu, \Sigma)$ is the normal distribution. As before, λ is a weighting factor that tries to modulate the importance of the second term relative to the first.

One important thing to note about the stubs is that the argument list requires the standard deviation and not the (co)variances. This is because of the simple relationship between the two for scalar random variables. With that in mind, a stripped down version of the algorithm is:

1. Start with initial guess of Gaussian parameters, e.g., means and standard deviations, and homogeneous priors.
2. Using Gaussian parameters, generate likelihoods for each class.
3. Using Bayes' rule, generate the posteriors (normalize).
4. Smooth the posteriors using your favorite smoother (you choose the smoothing parameters).
5. Normalize the probabilities (across the classes for each pixel).
6. Segment based on the *maximum a posteriori* estimate.
7. Use segmentation to update parameters.
8. Repeat (2)-(7) until desired number of iterations, or to convergence.

The function stub is called `segBayes.m`, and the script file you should prepare is called `segmentB.m`. Select two of the images from the Matlab file to process and go for it. Turn in the code and the results. Explain the output and your selection. Compare to at least one of the other segmentation strategies (k-means or ICM).