Mixture Models and EM

Goal: Introduction to probabilistic mixture models and the expectation-maximization (EM) algorithm.

Motivation:

- simultaneous fitting of multiple model instances
- unsupervised clustering of data
- coping with missing data
- segmentation? (... stay tuned)

Readings: Chapter 16 in the Forsyth and Ponce.

Matlab Tutorials: modelSelectionTut.m *(optional)*

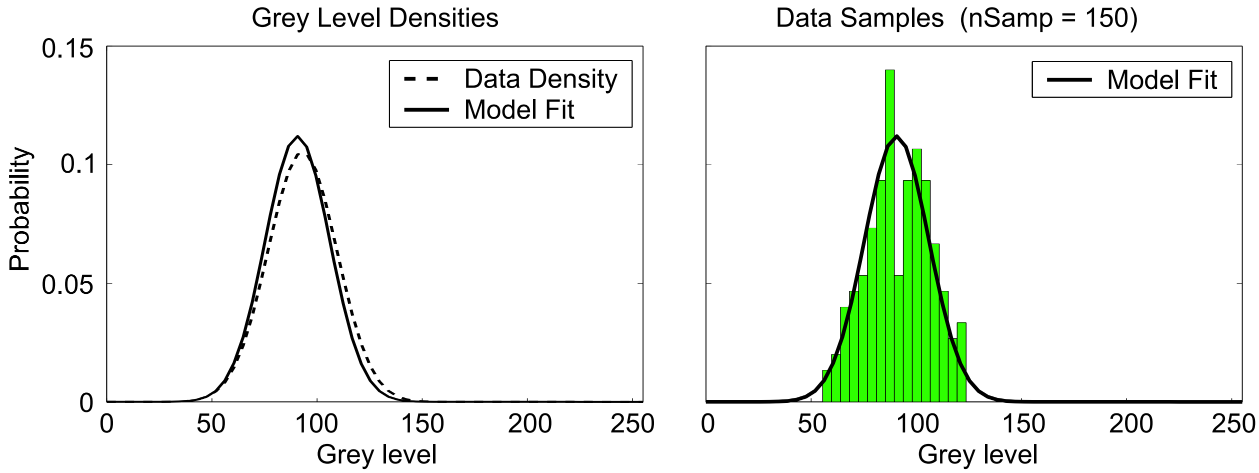
Model Fitting: Density Estimation

Let's say we want to model the distribution of grey levels $d_k \equiv d(\vec{\mathbf{x}}_k)$ at pixels, $\{\vec{\mathbf{x}}_k\}_{k=1}^{K}$, within some image region of interest.

Non-parametric model: Compute a histogram.

Parametric model: Fit an analytic density function to the data.
For example, if we assume the samples were drawn from a Gaussian distribution, then we could fit a Gaussian density to the data by computing the sample mean and variance:

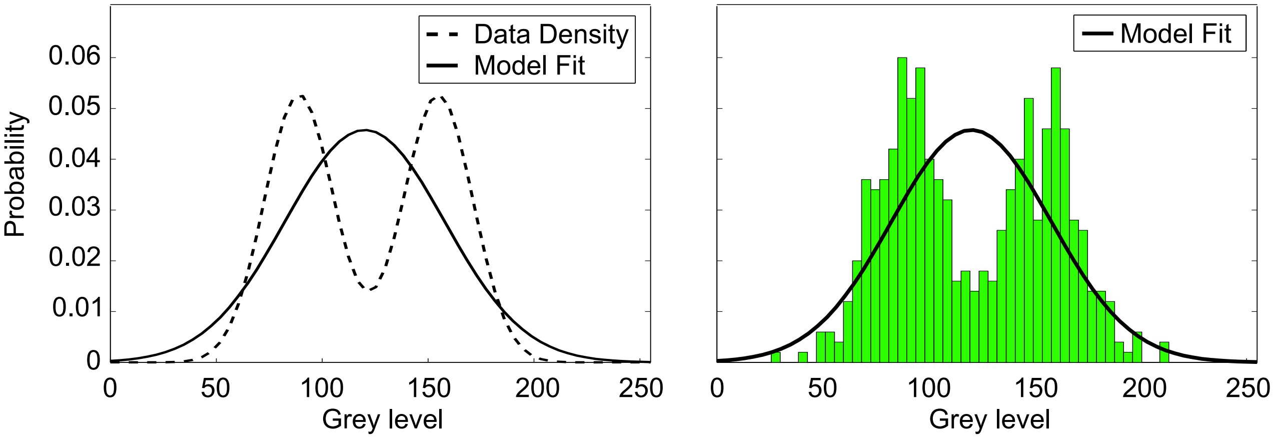
$$\mu = \frac{1}{K}\sum_k d_k \quad , \qquad \sigma^2 = \frac{1}{K-1}\sum_k (d_k - \mu)^2$$



Right plot shows a histogram of 150 IID samples drawn from the Gaussian density on the left (dashed). Overlaid is the estimated Gaussian model (solid).

Model Fitting: Multiple Data Modes

When the data come from an image region with more than one dominant color, perhaps near an occlusion boundary, then a single Gaussian will not fit the data well:



Missing Data: If the assignment of measurements to the two modes were *known*, then we could easily solve for the means and variances using sample statistics, as before, but only incorporating those data assigned to their respective models.

Soft Assignments: But we don't know the assignments of pixels to the two Gaussians. So instead, let's infer them:

Using Bayes' rule, the probability that d_k is owned (i.e., generated) by model $\mathcal{M}_n$ is

$$p(\mathcal{M}_n \,|\, d_k) \;=\; \frac{p(d_k \,|\, \mathcal{M}_n)\, p(\mathcal{M}_n)}{p(d_k)}$$

Ownership (example)

Above we drew samples from two Gaussians in equal proportions, so

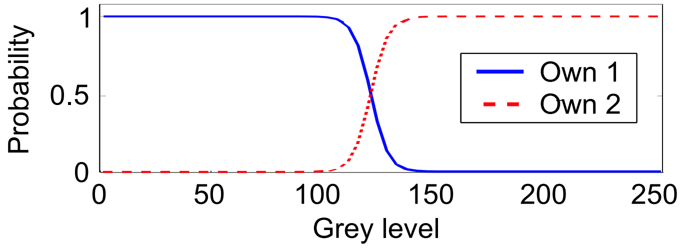
$$p(\mathcal{M}_1) = p(\mathcal{M}_2) = \frac{1}{2} \quad , \quad \text{and} \quad p(d_k \,|\, \mathcal{M}_n) \;=\; G(d_k;\, \mu_n, \sigma_n^2)$$

where $G(d;\, \mu, \sigma^2)$ is a Gaussian pdf with mean μ and variance σ^2 evaluated at d. And remember $p(d_k) = \sum_n p(d_k \,|\, \mathcal{M}_n)\, p(\mathcal{M}_n)$.

So, the ownerships, $q_n(d_k) \equiv p(\mathcal{M}_n \,|\, d_k)$, then reduce to

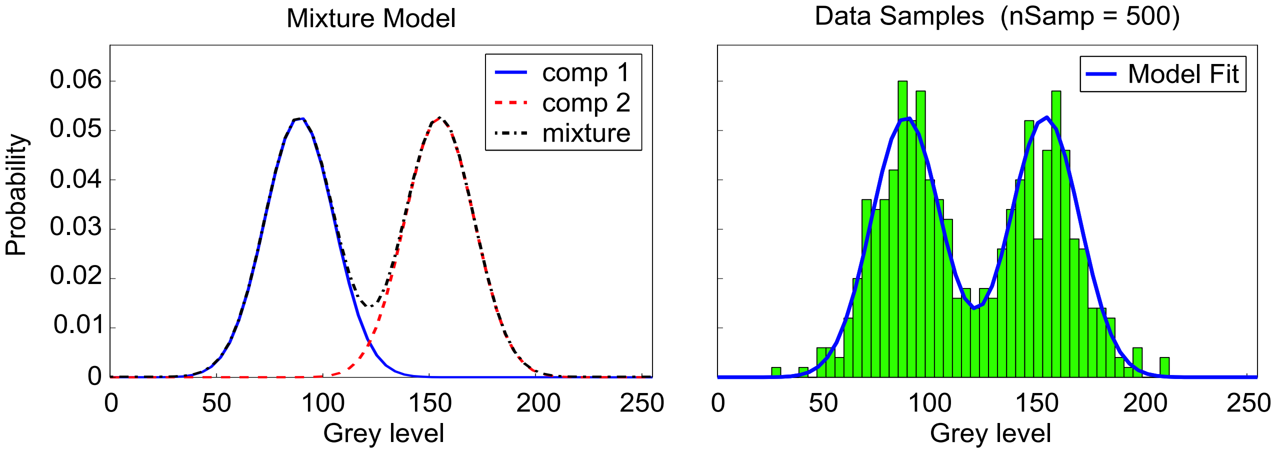
$$q_1(d_k) \;=\; \frac{G(d_k;\, \mu_1, \sigma_1^2)}{G(d_k; \mu_1, \sigma_1^2) + G(d_k; \mu_2, \sigma_2^2)} \quad , \quad \text{and} \quad q_2(d_k) \;=\; 1 - q_1(d_k)$$

For the 2-component density below:



Then, the Gaussian parameters are given by weighted sample stats:

$$\mu_n = \frac{1}{S_n} \sum_k q_n(d_k) d_k \quad , \quad \sigma_n^2 = \frac{1}{S_n} \sum_k q_n(d_k)(d_k - \mu_n)^2 \; , \quad S_n = \sum_k q_n(d_k)$$



Mixture Model

Assume

- N processes, $\{\mathcal{M}_n\}_{n=1}^N$, each of which generates some data (or measurements).
- Each sample d from process $\mathcal{M}_n$ is IID with density $p_n(d \,|\, \vec{\mathbf{a}}_n)$, where $\vec{\mathbf{a}}_n$ denotes parameters for process $\mathcal{M}_n$.
- The proportion of the entire data set produced solely by $\mathcal{M}_n$ is denoted $m_n = p(\mathcal{M}_n)$ (it's called a *mixing probability*).

Generative Process: First, randomly select one of the N processes according to the mixing probabilities, $\vec{\mathbf{m}} \equiv (m_1, ..., m_N)$. Then, given n, generate a sample from the observation density $p_n(d \,|\, \vec{\mathbf{a}}_n)$.

Mixture Model Likelihood: The probability of observing a datum d from the collection of N processes is given by their linear mixture:

$$p(d \,|\, \mathcal{M}) \;=\; \sum_{n=1}^{N} m_n \, p_n(d \,|\, \vec{\mathbf{a}}_n)$$

The *mixture model*, $\mathcal{M}$, comprises $\vec{\mathbf{m}}$, and the parameters, $\{\vec{\mathbf{a}}_n\}_{n=1}^N$.

Mixture Model Inference: Given K IID measurements (the data), $\{d_k\}_{k=1}^K$, our goal is to estimate the mixture model parameters.

Remarks: One may also wish to estimate N and the parametric form of each component, but that's outside the scope of these notes.

Expectation-Maximization (EM) Algorithm

EM is an iterative algorithm for parameter estimation, especially useful when one formulates the estimation problem in terms of *observed* and *missing* data.

- Observed data are the K intensities. Missing data are the assignments of observations to model components, $z_n(d_k) \in \{0, 1\}$.

Each EM iteration comprises an E-step and an M-step:

E-Step: Compute the expected values of the missing data given the current model parameter estimate. For mixture models one can show this gives the ownership probability: $\mathrm{E}[z_n(d_k)] = q_n(d_k)$.

M-Step: Compute ML model parameters given observed data and the expected value of the missing data. For mixture models this yields a weighted regression problem for each model component:

$$\sum_{k=1}^{K} q_n(d_k) \, \frac{\partial}{\partial \vec{\mathbf{a}}_n} \log p_n(d_k \,|\, \vec{\mathbf{a}}_n) \; = \; \vec{\mathbf{0}} \, .$$

and the mixing probabilities are $m_n \; = \; \frac{1}{K} \sum_{k=1}^{K} q_n(d_k)$.

Remarks:

- Each EM iteration can be shown to increase the likelihood of the observed data given the model parameters.
- EM converges to local maxima (not necessarily global maxima).
- An initial guess is required (e.g., random ownerships).

Derivation of EM for Mixture Models

The mixture model likelihood function is given by:

$$p\left(\{d_k\}_{k=1}^{K} \,|\, \mathcal{M}\right) \;=\; \prod_{k=1}^{K} p\left(d_k \,|\, \mathcal{M}\right) \;=\; \prod_{k=1}^{K} \sum_{n=1}^{N} m_n \, p_n(d_k \,|\, \vec{\mathbf{a}}_n)$$

where $\mathcal{M} \equiv \left(\vec{\mathbf{m}},\, \{\vec{\mathbf{a}}_n\}_{n=1}^{N}\right)$. The log likelihood is then given by

$$L(\mathcal{M}) \;=\; \log p\left(\{d_k\}_{k=1}^{K} \,|\, \mathcal{M}\right) \;=\; \sum_{k=1}^{K} \log\left(\sum_{n=1}^{N} m_n \, p_n(d_k \,|\, \vec{\mathbf{a}}_n)\right)$$

Our goal is to find extrema of the log likelihood function subject to the constraint that the mixing probabilities sum to 1. The constraint that $\sum_n m_n = 1$ can be included with a Lagrange multiplier. Accordingly, the following conditions can be shown to hold at the extrema of the objective function:

$$\frac{1}{K} \sum_{k=1}^{K} q_n(d_k) \;=\; m_n$$

and

$$\frac{\partial L}{\partial \vec{\mathbf{a}}_n} \;=\; \sum_{k=1}^{K} q_n(d_k) \, \frac{\partial}{\partial \vec{\mathbf{a}}_n} \log p_n(d_k \,|\, \vec{\mathbf{a}}_n) \;=\; \vec{\mathbf{0}} \,.$$

The first condition is easily derived from the derivative of the log likliehood with respect to m_n, along with the Lagrange multiplier.

The second condition is more involved as we show here, beginning with form of the derivative of the log likelihood with respect to the motion parameters for the m^{th} component:

$$\begin{aligned}
\frac{\partial L}{\partial \vec{\mathbf{a}}_n} \;&=\; \sum_{k=1}^{K} \frac{1}{\sum_{n=1}^{N} m_n \, p_n(d_k \,|\, \vec{\mathbf{a}}_n)} \, \frac{\partial}{\partial \vec{\mathbf{a}}_n} \left(\sum_{n=1}^{N} m_n \, p_n(d_k \,|\, \vec{\mathbf{a}}_n)\right) \\
&=\; \sum_{k=1}^{K} \frac{m_n}{\sum_{n=1}^{N} m_n \, p_n(d_k \,|\, \vec{\mathbf{a}}_n)} \, \frac{\partial}{\partial \vec{\mathbf{a}}_n} p_n(d_k \,|\, \vec{\mathbf{a}}_n) \\
&=\; \sum_{k=1}^{K} \frac{m_n p_n(d_k \,|\, \vec{\mathbf{a}}_n)}{\sum_{n=1}^{N} m_n \, p_n(d_k \,|\, \vec{\mathbf{a}}_n)} \, \frac{\partial}{\partial \vec{\mathbf{a}}_n} \log p_n(d_k \,|\, \vec{\mathbf{a}}_n)
\end{aligned}$$

The last step is an algebraic manipulation that uses the fact that $\frac{\partial \log p(a)}{\partial a} = \frac{1}{p(a)} \frac{\partial p(a)}{\partial a}$.

Derivation of EM for Mixture Models (cont)

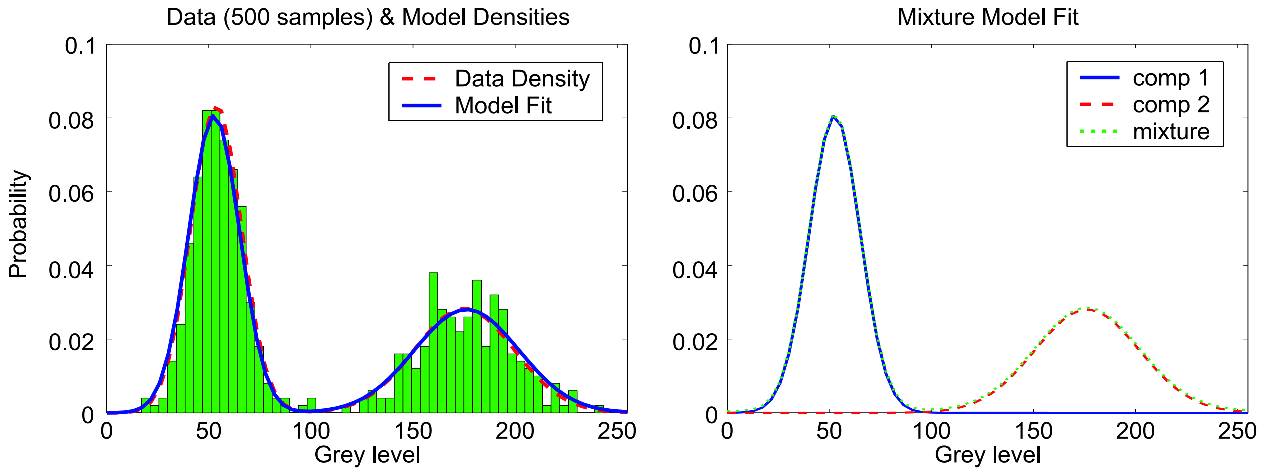
Notice that this equation can be greatly simplified because each term in the sum is really the product of the ownership probability $q_n(d_k)$ and the derivative of the component log likelihood. Therefore

$$\frac{\partial L}{\partial \vec{\mathbf{a}}_n} \;=\; \sum_{k=1}^{K} q_n(d_k)\, \frac{\partial}{\partial \vec{\mathbf{a}}_n} \log p_n(d_k \,|\, \vec{\mathbf{a}}_n)$$

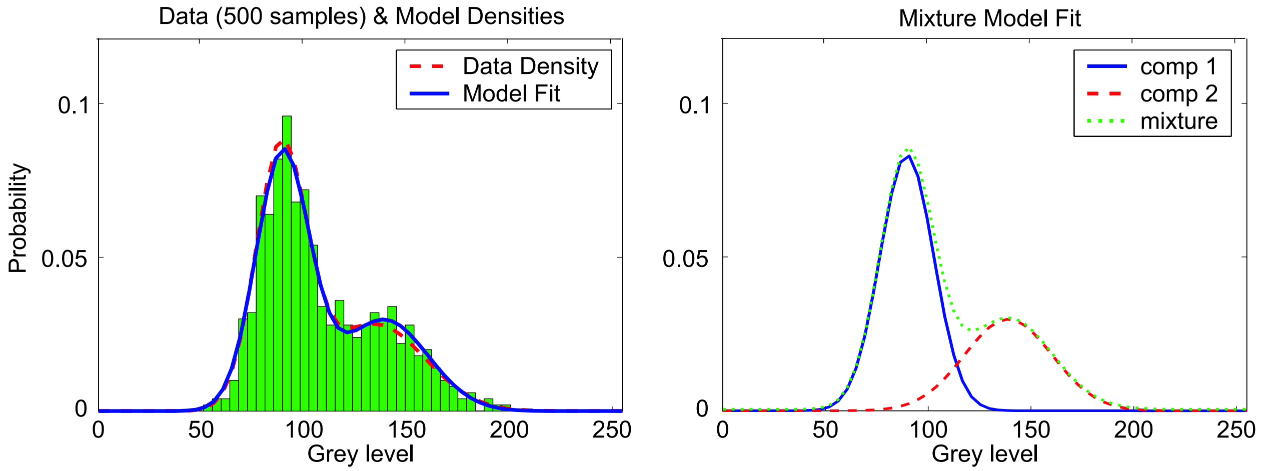
This is just a weighted log likelihood. In the case of a Gaussian component likelihood, $p_n(d_k \,|\, \vec{\mathbf{a}}_n)$, this is the derivative of a weighted least-squares error. Thus, setting $\partial L/\partial \vec{\mathbf{a}}_n = \vec{\mathbf{0}}$ in the Gaussian case yields a weighted least-squares estimate for $\vec{\mathbf{a}}_n$.

Examples

Example 1: Two distant modes. (We don’t necessarily need EM here since *hard* assignments would be simple to determine, and reasonably efficient statistically.)

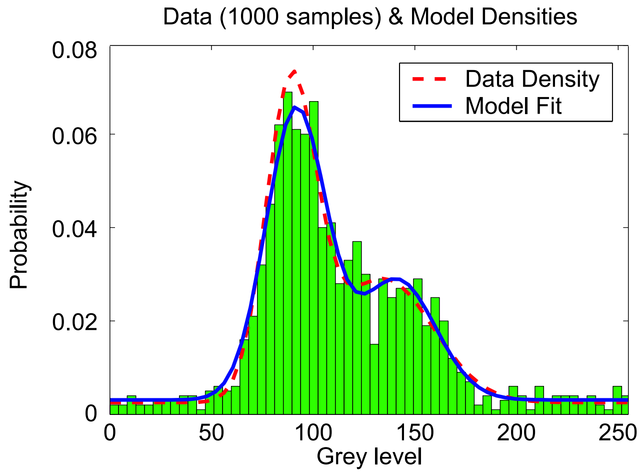


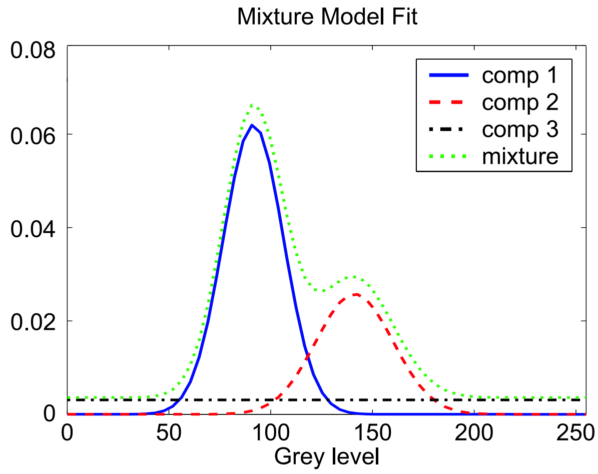
Example 2: Two nearby modes. (Here, the soft ownerships are essential to the estimation of the mode locations and variances.)



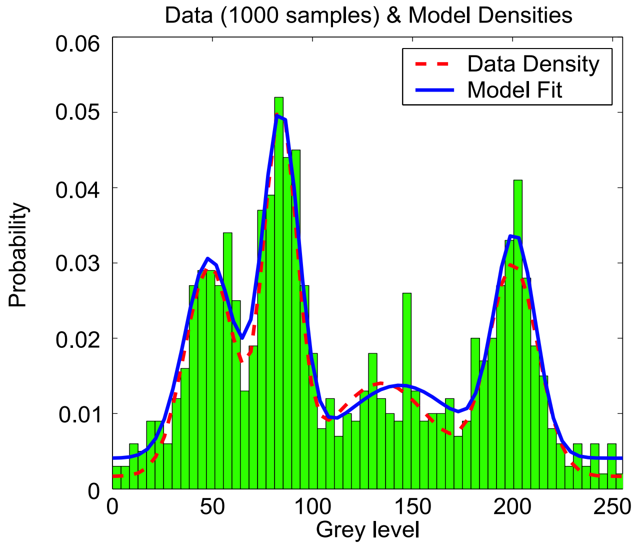
More Examples

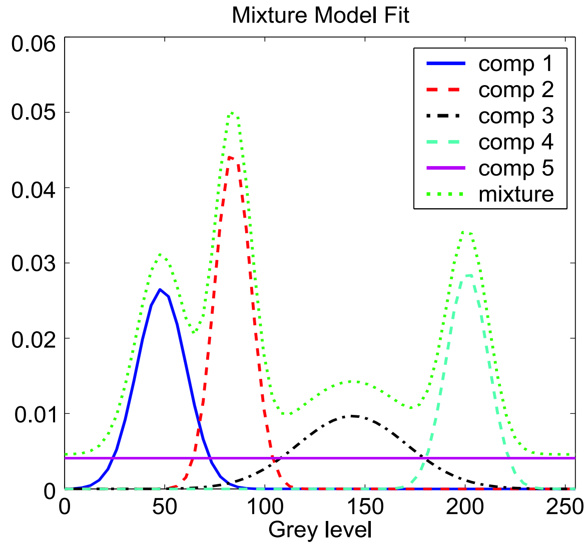
Example 3: Nearby modes with uniformly distributed outliers. The model is a mixture of two Gaussians and a uniform outlier process.





Example 4: Four modes and uniform noise present a challenge to EM. With only 1000 samples the model fit is reasonably good.



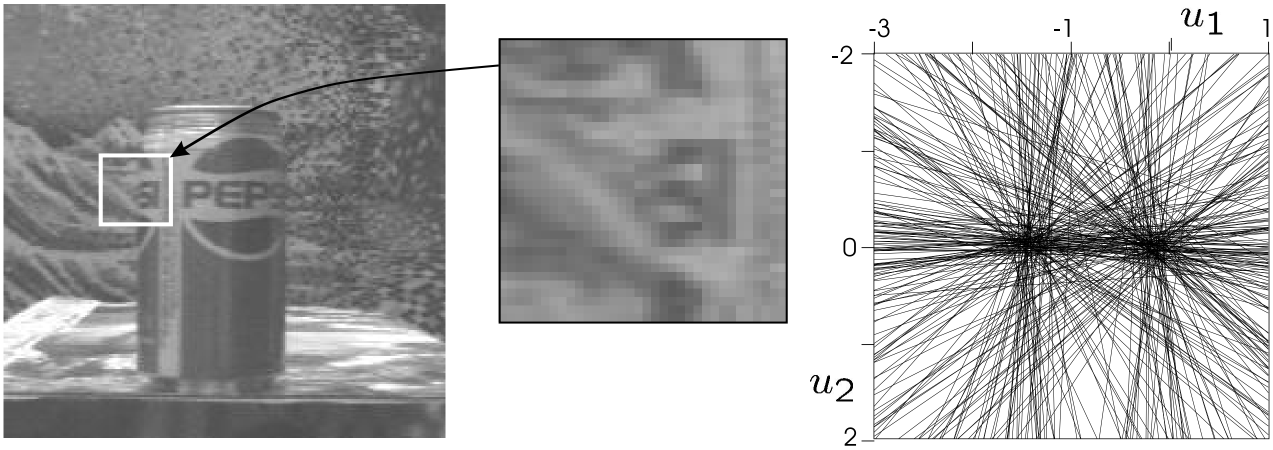


Mixture Models for Layered Optical Flow

Layered motion is a natural domain for mixture models and the EM algorithm. Here, we believe there may be multiple motions present, but we don't know the motions, nor which pixels belong together (i.e., move coherently).

- Two key sources of multiple motions, even within small image neighbourhoods, are occlusion and transparency.
- Mixture models are also useful when the motion model doesn't provide a good approximation to the 2D motion within the region

Example: The camera in the Pepsi sequence moves left-to-right. The depth discontinuities at the boundaries of the can produce motion discontinuities. For the pixels in the box in the left image, the right plot shows some of the motion constraint lines (in velocity space).



Mixture Models for Optical Flow

Formulation: Assume an image region contains two motions.

- Let the pixels in the region be $\{\vec{\mathbf{x}}_k\}_{k=1}^{K}$.
- Let's use parameterized motion models $\vec{\mathbf{u}}_j(\vec{\mathbf{x}};\, \vec{\mathbf{a}}_j)$ for $j = 1, 2$, where $\vec{\mathbf{a}}_j$ are the motion model parameters. (E.g., $\vec{\mathbf{a}}$ is 2D for a translational model, and 6D for an affine motion model.)
- One gradient measurement per pixel, $\vec{\mathbf{c}}_k \equiv (f_x(\vec{\mathbf{x}}_k), f_y(\vec{\mathbf{x}}_k), f_t(\vec{\mathbf{x}}_k))$.
- Like gradient-based flow estimation, let $\vec{\nabla} f(\vec{\mathbf{x}}_k) \cdot \vec{\mathbf{u}} + f_t(\vec{\mathbf{x}}_k)$ be mean-zero Gaussian with variance σ_v^2, that is,

$$p_n(\vec{\mathbf{c}}_k \,|\, \vec{\mathbf{x}}_k,\, \vec{\mathbf{a}}_n) \;=\; G(\vec{\nabla} f(\vec{\mathbf{x}}_k) \cdot \vec{\mathbf{u}} + f_t(\vec{\mathbf{x}}_k);\, 0,\, \sigma_v^2)$$

- Let the fraction of measurements (pixels) owned by each of the two motions be denoted m_1 and m_2.
- Let m_0 denote the fraction of outlying measurements, inconsistent with either motion, and assume a uniform density for the outlier likelihood, denoted p_0.
- With three components, $m_0 + m_1 + m_2 = 1$.

Mixture Models for Optical Flow

Mixture Model: The observation density for measurement $\vec{\mathbf{c}}_k$ is

$$p(\vec{\mathbf{c}}_k \,|\, \vec{\mathbf{m}},\, \vec{\mathbf{a}}_1,\, \vec{\mathbf{a}}_2) \;=\; m_0\, p_0 \,+\, \sum_{n=1}^{2} m_n\, p_n(\vec{\mathbf{c}}_k \,|\, \vec{\mathbf{x}}_k, \vec{\mathbf{a}}_n)\ .$$

where $\vec{\mathbf{m}} \equiv (m_0, m_1, m_2)$.

Given K IID measurements $\{\vec{\mathbf{c}}_k\}_{k=1}^{K}$ the joint likelihood is the product of individual component likelihoods:

$$L(\vec{\mathbf{m}},\, \vec{\mathbf{a}}_1,\, \vec{\mathbf{a}}_2) \;=\; \prod_{k=1}^{K} p(\vec{\mathbf{c}}_k \,|\, \vec{\mathbf{m}},\, \vec{\mathbf{a}}_1,\, \vec{\mathbf{a}}_2)\ .$$

EM Algorithm:

- *E Step:* Infer the ownership probability, $q_n(\vec{\mathbf{c}}_k)$, that constraint $\vec{\mathbf{c}}_k$ is owned by the n^{th} mixture component. For the motion components of the mixture ($n=1,2$), given $\vec{\mathbf{m}}$, $\vec{\mathbf{a}}_1$ and $\vec{\mathbf{a}}_2$, we have:

  $$q_n(\vec{\mathbf{c}}_k) \;=\; \frac{m_n\, p_n(\vec{\mathbf{c}}_k \,|\, \vec{\mathbf{a}}_n)}{m_0 p_0 + m_1 p_1(\vec{\mathbf{c}}_k \,|\, \vec{\mathbf{x}}_k,\, \vec{\mathbf{a}}_1) + m_2 p_2(\vec{\mathbf{c}}_k \,|\, \vec{\mathbf{x}}_k,\, \vec{\mathbf{a}}_2)}\ .$$

  And of course the ownership probabilities sum to one so:

  $$q_0(\vec{\mathbf{c}}_k) \;=\; 1 - q_1(\vec{\mathbf{c}}_k) - q_2(\vec{\mathbf{c}}_k)$$

- *M Step:* Compute the maximum likelihood estimates of the mixing probabilities $\vec{\mathbf{m}}$ and the flow field parameters, $\vec{\mathbf{a}}_1$ and $\vec{\mathbf{a}}_2$.

ML Parameter Estimation

Mixing Probabilities: Given the ownership probabilities, the mixing probabilities are the fractions of the total ownership probability assigned to the respective components:

$$\frac{1}{K}\sum_{k=1}^{K} q_n(\vec{c}_k) \;=\; m_n$$

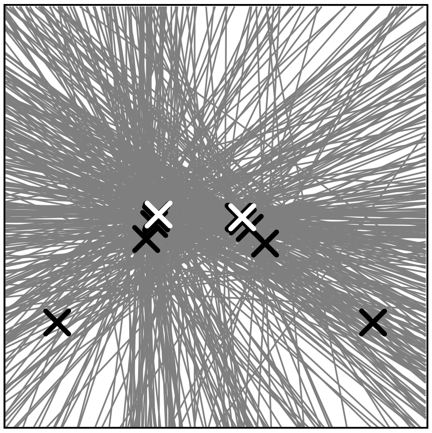
Flow Estimation: Given the ownership probabilities, we can estimate the motion parameters for each component separately with a form of weighted, least-squares area-based regression.

E.g., for 2D translation, where the $\vec{a}_n \equiv \vec{\mathbf{u}}_n$, this amounts to the minimization of the weighted least-squares error

$$\begin{aligned} E(\vec{\mathbf{u}}_n) \;&=\; \sum_{k=1}^{K} q_n(\vec{\mathbf{c}}_k)\, d^2(\vec{\mathbf{u}}_n,\, \vec{\mathbf{c}}_k) \\ &=\; \sum_{k=1}^{K} q_n(\vec{\mathbf{c}}_k)\, \left[\vec{\nabla} f(\vec{\mathbf{x}}_k, t) \,\cdot\, \vec{\mathbf{u}}_n \,+\, f_t(\vec{\mathbf{x}}_k, t)\right]^2 \;, \end{aligned}$$

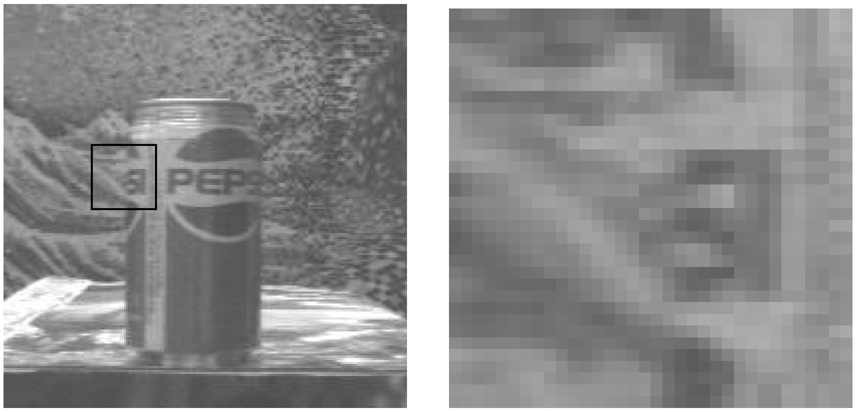
where $\vec{\nabla} f \equiv (f_x,\, f_y)^T$ (*cf.* iteratively reweighted LS for robust estimation).

Convergence Behaviour:
(in velcocity space)

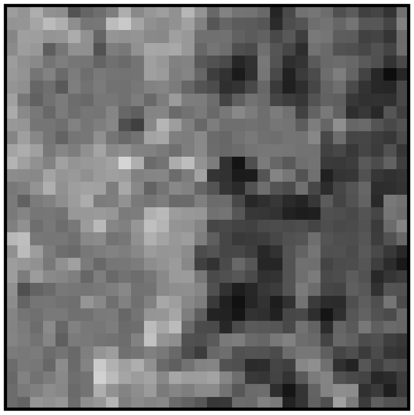


Representation for the Occlusion Example

Region at an occlusion boundary.

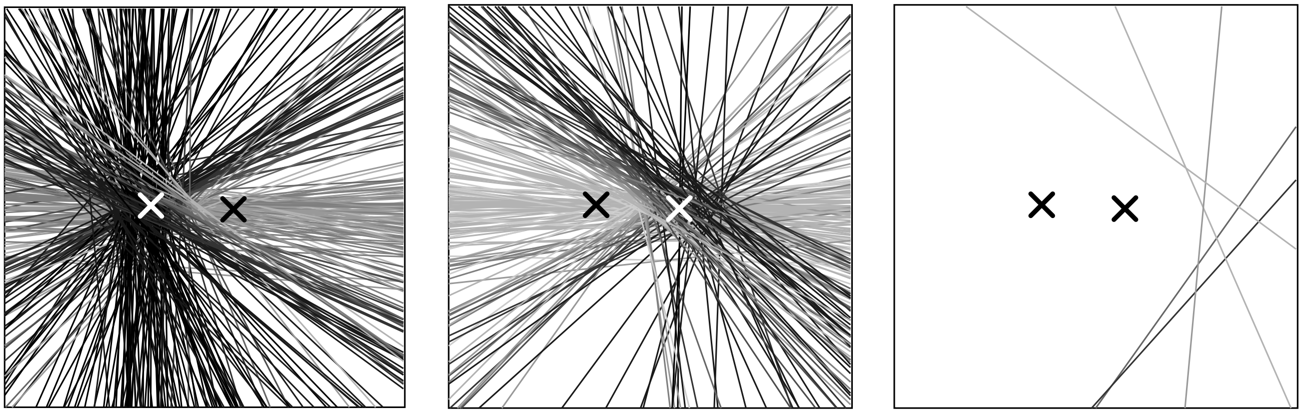


Pixel Ownership:



for Layer #2

Constraint Ownership:



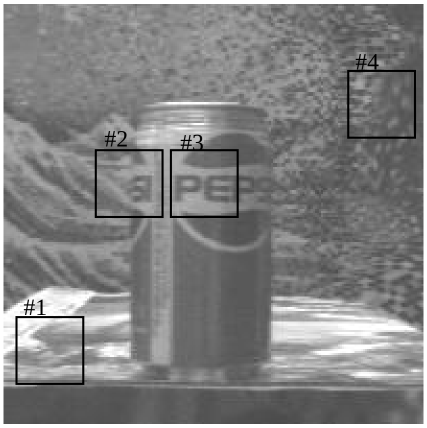
Layer #1

Layer #2

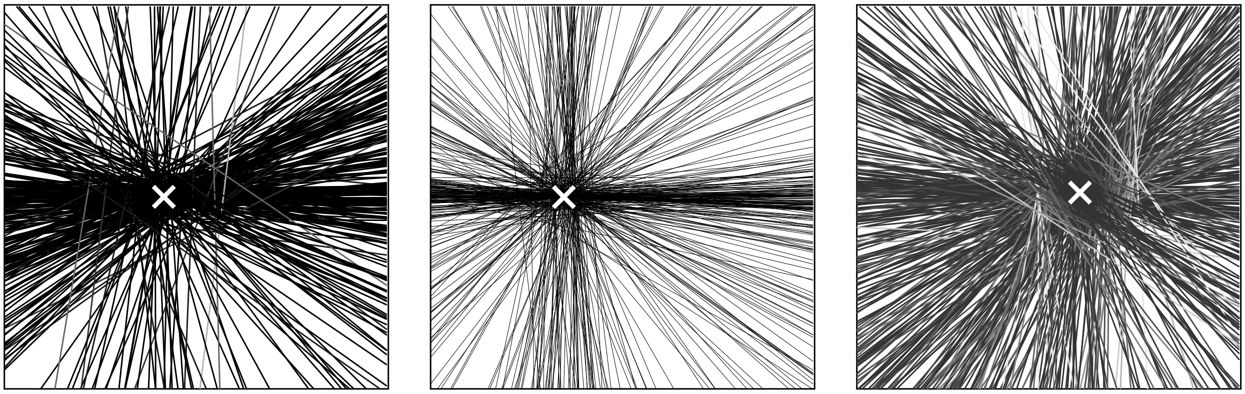
Outliers

Additional Test Patches

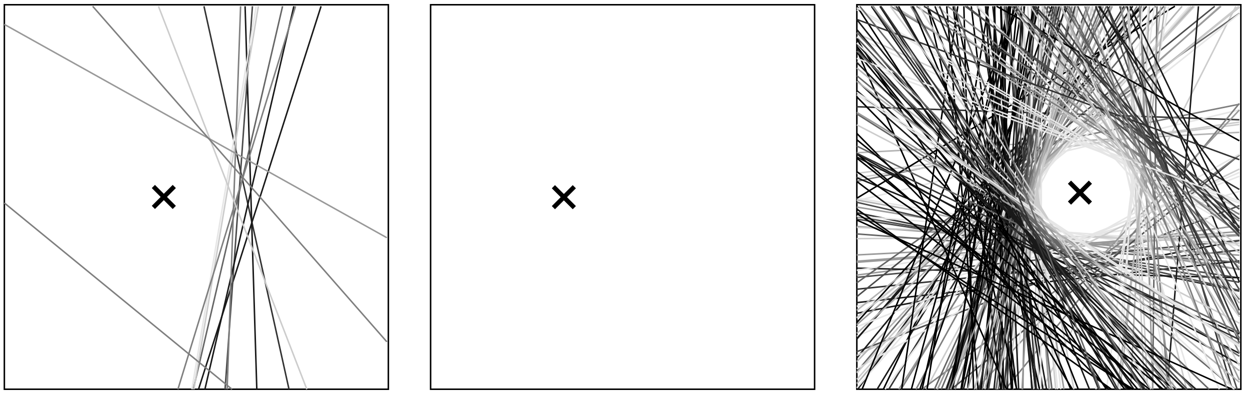
Test Patches:



Motion constraints for regions #1, #3 and #4:



Outliers for regions #1, #3 and #4:



Further Readings

Papers on mixture models and EM:

A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society Series B*, pp. 1–38, 1977.

Papers on mixture models for layered motion:

S. Ayer and H. Sawhney. H.S. Sawhney and S. Ayer. Compact Representations of Videos Through Dominant and Multiple Motion Estimation, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(8):777–784, 1996.

A.D. Jepson and M. J. Black. Mixture models for optical flow computation. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 760–761, New York, June 1993.

Y. Weiss and E.H. Adelson. A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. *IEEE Proc. Computer Vision and Pattern Recognition*, San Francisco, pp. 321–326, 1996.