Mixture of Gaussians
Expectation Maximization (EM)

Part 2

Most of the slides are due to Christopher Bishop

BCS Summer School, Exeter, 2003.

The rest of the slides are based on lecture notes by A. Ng

Limitations of K-means

- Hard assignments of data points to clusters – small shift of a data point can flip it to a different cluster
- Not clear how to choose the value of K
- Solution: replace ‘hard’ clustering of K-means with ‘soft’ probabilistic assignments
- Represents the probability distribution of the data as a *Gaussian mixture model*

Gaussian Mixtures

- Linear super-position of Gaussians

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- Normalization and positivity require

$$\sum_{k=1}^{K} \pi_k = 1 \qquad 0 \leqslant \pi_k \leqslant 1$$

- Can interpret the mixing coefficients as prior probabilities

$$p(\mathrm{x}) = \sum_{k=1}^{K} p(k)\, p(\mathrm{x} \mid k)$$

Maximum Likelihood for the GMM

- The log likelihood function takes the form

$$\ln p(D|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

- Note: sum over components appears *inside* the log
- There is no closed form solution for maximum likelihood
- How to maximize the log likelihood
  - solved by expectation-maximization (EM) algorithm

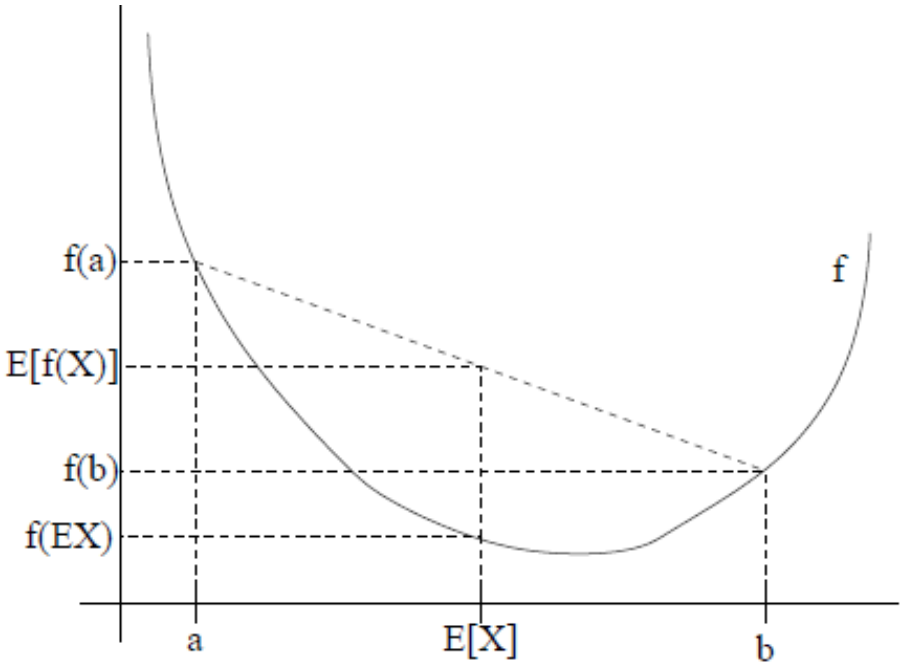
EM Algorithm – Informal Derivation

- The solutions are not closed form since they are coupled
- Suggests an iterative scheme for solving them:
  - Make initial guesses for the parameters
  - Alternate between the following two stages:
    1. E-step: evaluate responsibilities
    2. M-step: update parameters using ML results

General View of the EM algorithm

Jensen's inequality:

Let f be a convex function ($f''(x) \geq 0$ for all $x \in \Re$) and X be a and random variable. Then $E[f(X)] \geq f(E[X])$.



f is convex
X =a with probability 0.5
X =b with probability 0.5

E[X] is given by the midpoint between a and b.
E[f(X)] is the midpoint between f(a) and f(b).
E[f(X)] ≥f(EX).

Jensen's inequality (cont.)

Further, if f is a strictly convex function ($f''(x) > 0$), then $E[f(X)] = f(E[X])$ holds true if and only if $E[X] = X$ with probability 1 (i.e., if X is a constant).

Jensen's inequality also holds for concave functions ($f''(x) \leq 0$), but with the direction of all the inequalities reversed ($E[f(X)] \leq f(E[X])$, etc.).

Problem Definition

- Suppose we have an estimation problem in which we have a training set $\{x_1,.., x_m\}$ of iid samples.
- We wish to fit the parameters of a model $p(x, z; \theta)$ to the data.

observed hidden parameters

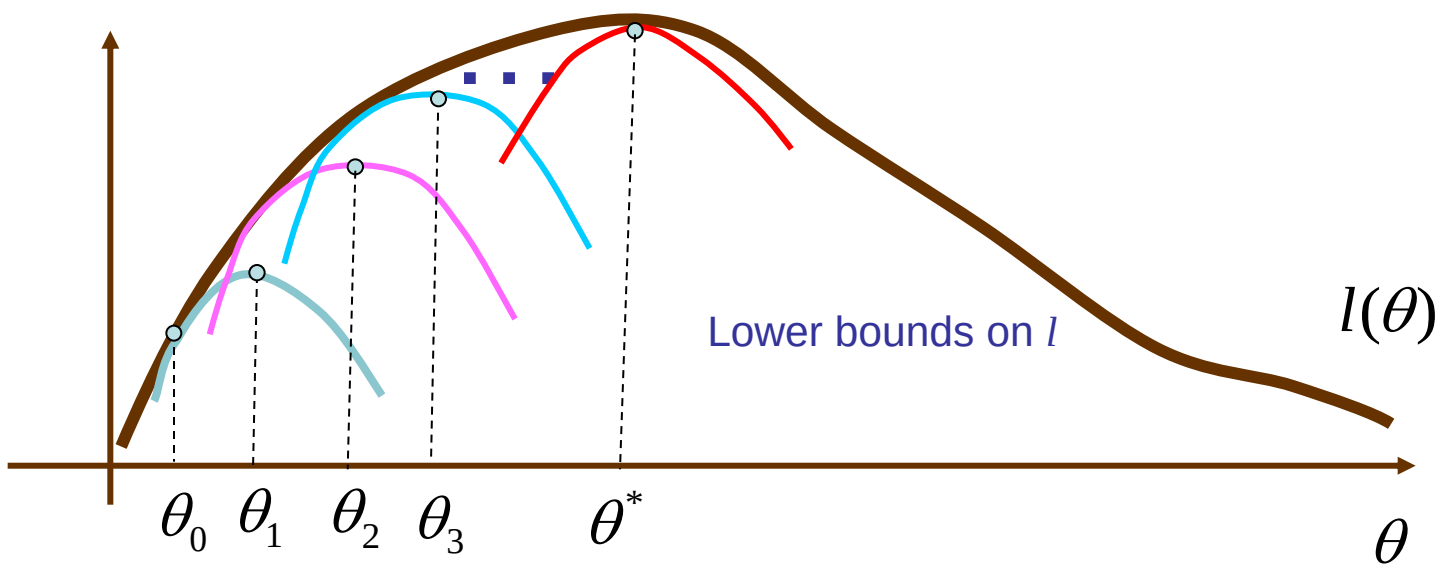
- We want to maximize the likelihood

$$l(\theta) = \sum_{i=1}^{m} \log p(x, \theta) = \sum_{i=1}^{m} \log \sum_{z} p(x, z; \theta)$$

- Doing it explicitly may be hard, since z's are the non-observed.
- If z's were observed, then (often) maximum likelihood estimation would be easy.

EM at glance

- Our strategy will be to repeatedly
  - construct a lower-bound on $l(\theta)$ (E-step),
  - optimize that lower-bound (M-step).



EM algorithm derivation

$$l(\theta) = \sum_{i=1}^{m} \log p(x_i, \theta) = \sum_{i=1}^{m} \log \sum_{z_i} p(x_i, z_i; \theta)$$

$$= \sum_{i=1}^{m} \log \boxed{\sum_{z_i} Q_i(z_i) \frac{p(x_i, z_i; \theta)}{Q_i(z_i)}}$$

$\forall i, \quad Q_i(z)$ is some distribution over z's

$$\sum_{z} Q_i(z) = 1, \; Q_i(z) \geq 0$$

$$= \sum_{i=1}^{m} \log \boxed{E\left[\frac{p(x_i, z_i; \theta)}{Q_i(z_i)}\right]}$$

$E\left[\frac{p(x_i, z_i; \theta)}{Q_i(z_i)}\right]$ is with respect to z_i drawn according to the distribution given by Q_i

$$\geq \sum_{i=1}^{m} E\left[\log \frac{p(x_i, z_i; \theta)}{Q_i(z_i)}\right]$$

Jensen's inequality: $\log\left(E[X]\right) \geq E[\log X]$

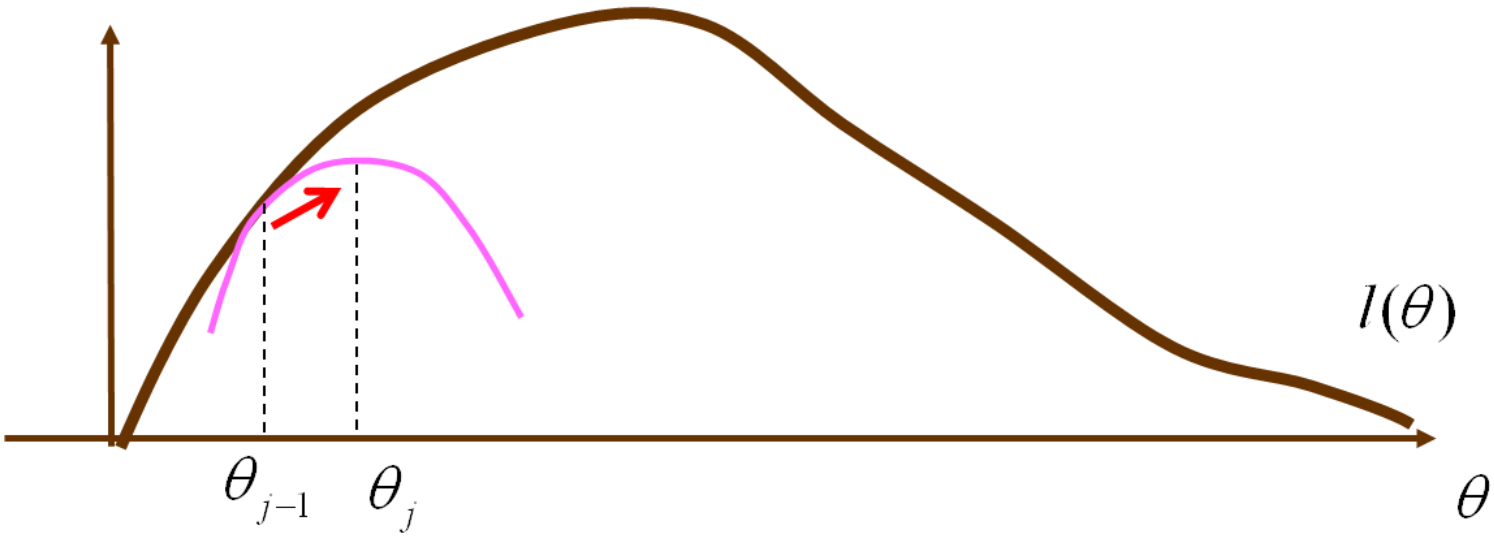
$$= \sum_{i=1}^{m} \sum_{z_i} Q_i(z_i) \log \frac{p(x_i, z_i; \theta)}{Q_i(z_i)}$$

This is a lower bound on $l(\theta)$

EM algorithm derivation (cont.)

$$l(\theta) \geq \sum_{i=1}^{m} \sum_{z_i} Q_i(z_i) \log \frac{p(x_i, z_i; \theta)}{Q_i(z_i)}$$

We want a lower bound to be equal to l at the previous θ



EM algorithm derivation (cont.)

To ensure that, we should choose $Q_i(z)$ such that inequality in our derivation above would hold with equality. We require that:

$$\frac{p(x_i, z_i; \theta)}{Q_i(z_i)} = const \quad \Rightarrow \quad Q_i(z_i) \propto p(x_i, z_i; \theta)$$

Since we know that $\sum_z Q_i(z_i) = 1$, then

$$Q_i(z_i) = \frac{p(x_i, z_i; \theta)}{\sum_z p(x_i, z_i; \theta)} = \frac{p(x_i, z_i; \theta)}{p(x_i; \theta)} = p(z_i | x_i; \theta)$$

General EM Algorithm

Repeat until convergence {

– E-step:

For each i set $Q_i(z_i) := p(z_i | x_i; \theta)$

– M-step:

$$\theta := \arg\max_{\theta} \underbrace{\sum_{i=1}^{m} \sum_{z_i} Q_i(z_i) \log \frac{p(x_i, z_i; \theta)}{Q_i(z_i)}}_{\text{Lower bound on } l(\theta)}$$

}

EM for MoG revisited

- For $1 \le i \le N,\ 1 \le j \le K$, define hidden variables z_{ij}

$$z_{ij} = \begin{cases} 1 & \text{if sample i was generated by component } \mathbf{k} \\ 0 & \text{otherwise} \end{cases}$$

- z_{ij} are indicator random variables, they indicate which Gaussian component generated sample x_i

- Let $z_i = \{z_{i1}, \ldots, z_{iK}\}$ indicator r.v. correspond to sample x_i .

  We say that $z_i = k$, when its k'st coordinate is 1 and the rest are 0.

- Conditioned on z_i, distribution of x_i is Gaussian

$$p(\mathrm{x}_i \mid z_i = k) \sim N(\mu_k, \Sigma_k)$$

EM for MoG revisited

E-step:

$$Q_i(z_i = k) = p(z_i = k \mid x_i; \mu, \Sigma, \pi)$$

$$= \frac{\pi_k N(x_i \mid \mu_k, \Sigma_k)}{\sum_j \pi_j N(x_i \mid \mu_j, \Sigma_j)} \quad = \gamma_k(x_i)$$

EM for MoG revisited

M-step: $$\max_{\mu,\Sigma,\pi} \boxed{\sum_{i=1}^{m}\sum_{z_i} Q_i(z_i)\log\frac{p(x_i,z_i;\theta)}{Q_i(z_i)}}$$

$$= \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_k(x_i)\log\frac{\pi_k N(x_i \mid \mu_k,\Sigma_k)}{\gamma_k(x_i)}$$

$$\nabla_\mu(\ldots) \overset{\text{set}}{=} 0 \Longrightarrow \mu_k = \frac{\sum_{i=1}^{N}\gamma_k(x_i)x_i}{\sum_{i=1}^{N}\gamma_k(x_i)}$$

Similarly,

$$\pi_k = \frac{1}{N}\sum_{i=1}^{N}\gamma_k(x_i), \qquad \Sigma_k = \frac{\sum_{i=1}^{N}\gamma_k(x_i)(x_i-\mu_k)(x_i-\mu_k)^T}{\sum_{i=1}^{N}\gamma_k(x_i)}$$

K-means Algorithm

- Goal: represent a data set in terms of K clusters each of which is summarized by a prototype $\boldsymbol{\mu}_k$
- Initialize prototypes, then iterate between two phases:
  - E-step: assign each data point to nearest prototype
  - M-step: update prototypes to be the cluster means

Responsibilities

- *Responsibilities* assign data points to clusters

$$r_{nk} \in \{0, 1\}$$

such that

$$\sum_k r_{nk} = 1$$

- Example: 5 data points and 3 clusters

$$(r_{nk}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

K-means Cost Function

data

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

responsibilities

prototypes

Minimizing the Cost Function

- E-step: minimize J w.r.t. r_{nk}
  - assigns each data point to nearest prototype
- M-step: minimize J w.r.t $\boldsymbol{\mu}_k$
  - gives

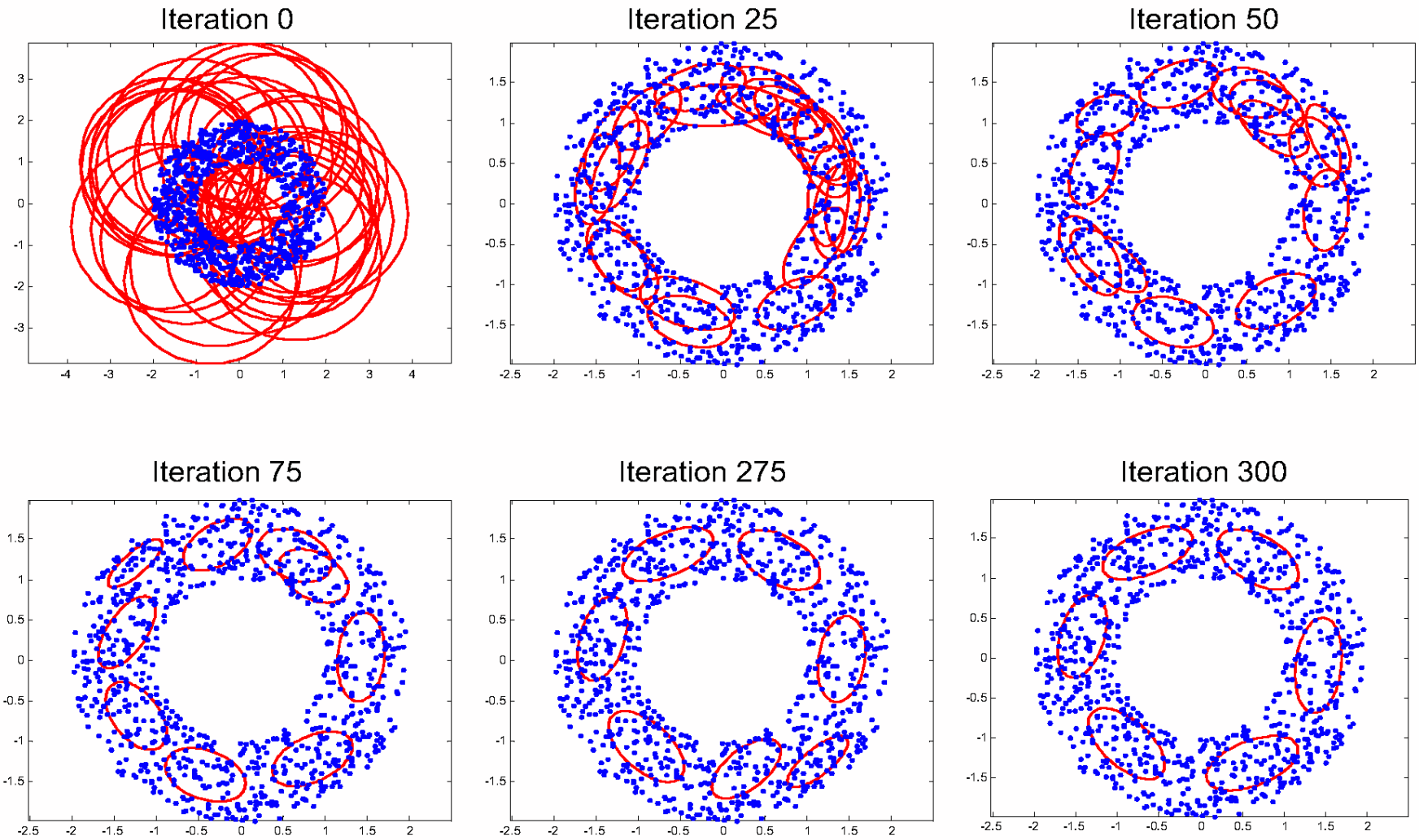
$$\boldsymbol{\mu}_k = \frac{\sum_n r_{kn}\mathbf{x}_n}{\sum_n r_{kn}}$$

  - each prototype set to the mean of points in that cluster
- Convergence guaranteed since there is a finite number of possible settings for the responsibilities

EM Example

- Example from R. Gutierrez-Osuna
- Training set of 900 examples forming an annulus
- Mixture model with **m** = 30 Gaussian components of unknown mean and variance is used
- Training:
  - Initialization:
    - means to 30 random examples
    - covaraince matrices initialized to be diagonal, with large variances on the diagonal (compared to the training data variance)
  - During EM training, components with small mixing coefficients were trimmed
    - This is a trick to get in a more compact model, with fewer than 30 Gaussian components

EM Example



from R. Gutierrez-Osuna

EM Motion Segmentation Example

Three frames from the MPEG “flower garden” sequence

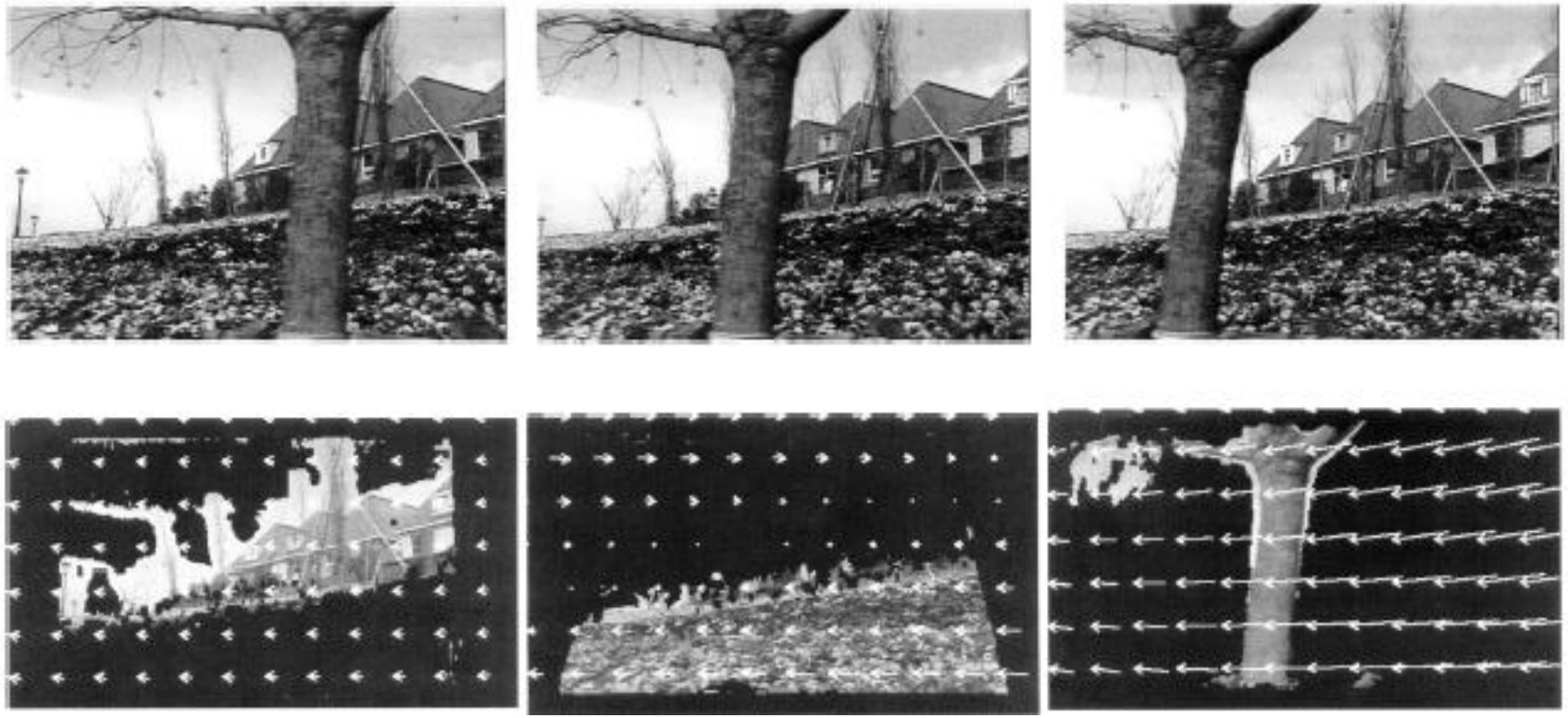


Figure from “Representing Images with layers,”, by J. Wang and E.H. Adelson, IEEE Transactions on Image Processing, 1994, c 1994, IEEE