Probabilistic Segmentation And Tracking

Michael Trumpis

Electrical Engineering
The Polytechnic Institute of NYU

May 15, 2012

Outline

Feature Based Segmentation

What is a feature? could be

- color vector: (R,G,B), (L,A,B), (G,B), etc
- color and pixel location: (X,Y,R,G,B)
- intensity and edge-strength
- color and wavelet coefficient
- etc, etc

We are talking only about variants of the first 2

Feature Based Segmentation

Some terminology

- Feature space: $\mathcal{F} \subset \mathbb{R}^N$
- Feature map: $f : (x, y) \mapsto \mathcal{F}$
  - e.g. The image itself $\Psi(x, y)$, a quantizing function, ...
- Segment map / label map

Feature Distribution

Model

The combined features in an image are distributed according to a complex, multi-modal probability density.

features from a given class (segment) instantiated from single mode

1. **how to find the probability density?**
2. how to identify which mode corresponds to the feature at any pixel?

Classical Density Estimation

Parzen kernel density estimation

A sample of n points $S = \{x\}_i \subset \mathbb{R}^n \quad x_i \sim F$

- Begin with empirical distribution

$$F_n(x) = \frac{1}{n} \sum_{i=1}^{n} \theta(x - x_i)$$

- minimize quadratic functional for f:

$$R(f, F_n) = \|Af - F_n\|_{L_2}^2 + \gamma \|K * f\|_{L_2}^2$$

- where $Af = \int_{-\infty}^{x} f(t)dt = F(x)$ is the true (unknown) distribution

Classical Density Estimation

- Note that in Fourier space:
  $f(x) = \frac{d}{dx}F(x) \rightarrow \bar{f}(\omega) = j\omega\bar{F}(\omega) \rightarrow \bar{F}(\omega) = \frac{1}{j\omega}\bar{f}(\omega)$
- Also $\mathcal{F}\{F_n\} = \frac{1}{n}(j\omega)^{-1}\sum_{i=1}^{n} e^{-j\omega x_i}$,
  and convolution$\rightarrow$multiplication
- Fourier Transform preserves L_2 norm, so the problem is now

$$R(f, F_n) = \left\| \frac{\bar{f}(\omega) - \frac{1}{n}\sum_{i=1}^{n} e^{-j\omega x_i}}{j\omega} \right\|_{L_2}^2 + \gamma \|\bar{K}(\omega)\bar{f}(\omega)\|_{L_2}^2$$

$$\rightarrow \bar{f}_n(\omega) = g_\gamma(\omega)\frac{1}{n}\sum_{i=1}^{n} e^{-j\omega x_i} \quad g_\gamma(\omega) = (1+\gamma\omega^2\bar{K}(\omega)\bar{K}^*(\omega))^{-1}$$

Kernel Density Estimator

- Estimator of density in Fourier space

$$\bar{f}_n(\omega) = g_\gamma(\omega)\frac{1}{n}\sum_{i=1}^{I} e^{-j\omega x_i}$$

- Transform back to X

$$f_n(x) = \mathcal{F}^{-1}\{\bar{f}_n\} = \frac{1}{n}\sum_{i=1}^{n}\int g_\gamma(\omega)e^{j\omega(x-x_i)}d\omega$$

- The density at any x is estimated as a kernel-weighted sum of the sample points

$$f_n(x) = \frac{1}{n}\sum_{i=1}^{n} G_\gamma(x - x_i)$$

Kernel Density Estimator

Consider a simplified kernel estimator

$$\hat{f}_{h,K}(x) = \frac{1}{n}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \qquad K(x) = c_{k,h,n}k(\|x\|^2) \qquad k : \mathbb{R} \mapsto \mathbb{R}$$

- K is a radially symmetric kernel, normalized to unit mass
- k is the kernel profile, which is a 1D real-valued function, e.g.
  - Gaussian $k_G(x) \propto e^{-x/2}$
  - Epanechnikov $k_E(x) \propto (1 - x)\ 0 \leq x \leq 1$
- h is a "bandwidth" parameter, controlling the kernel window size

Feature Distribution

Model

The combined features in an image are distributed according to a complex, multi-modal probability density.

features from a given class (segment) instantiated from single mode

1. how to find the probability density?
2. **how to identify which mode corresponds to the feature at any pixel?**

Mode Seeking

Problem

How to find a mode?

It's just a local maximum, so walk up the gradient.

Density Gradient Estimate

The gradient of the density estimate is taken for the density gradient estimator

$$\hat{\nabla}_x f_{h,K}(x) = \nabla_x \hat{f}_{h,K}(x) \propto \sum_{i=1}^{n} (x - x_i) k' \left(\left\| \frac{x - x_i}{h} \right\|^2 \right)$$

Substitute $g(x) = -k'(x)$, *and use algebra trick*

$$\rightarrow \left[\sum_i g \left(\left\| \frac{x - x_i}{h} \right\|^2 \right) \right] \left[\frac{\sum_i x_i g \left(\left\| \frac{x - x_i}{h} \right\|^2 \right)}{\sum_i g \left(\left\| \frac{x - x_i}{h} \right\|^2 \right)} - x \right]$$

We have found the ***mean shift!***

Mean Shift

$$\left[\sum_i g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)\right]\left[\frac{\sum_i x_i g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}{\sum_i g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)} - x\right]$$

density estimate at x
under kernel $g(x)$: $\hat{f}_{h,G}(x)$

mean shift vector $m_{h,G}(x)$

Mean Shift Gradient Ascent

- The gradient $\nabla_x \hat{f}_{h,K}(x)$ is proportional to the mean shift vector

$$m_{h,G}(x) = \left[\frac{\sum_i x_i g\left(\left\| \frac{x - x_i}{h} \right\|^2 \right)}{\sum_i g\left(\left\| \frac{x - x_i}{h} \right\|^2 \right)} - x \right]$$

- To find a local ***maximum***, do Gradient ***Ascent*** iterations

$$x^{(p+1)} = x^{(p)} + \alpha \nabla_x \hat{f}(x^{(p)}) = x^{(p)} + m_{h,G}(x^{(p)})$$

- α auto-tuned by $1/Prob\{x\}$ (estimated with kernel g)
  - bottom/top of the hill takes larger/smaller steps
- For Gaussian $k(x)$, $g(x)$ is Gaussian
- For Epanechnikov $k(x)$, $g(x)$ is ***uniform*** within bandwidth h

Mode Based Segmentation

Definition

Given a feature mapping x from pixels to feature space, we partition all pixels z in image $\Psi(z)$ into classes $L_k \equiv \{z : \text{Ascent}\{x(z)\} = M_k\}$

- mean shift iteration leads from any point x on the probability density manifold to a stationary point*: one of the modes M_i of $f(x)$.
- label pixel z by the mode found from $x(z)$

Mode Based Segmentation

Note that the proposed segmentation:

- is agnostic towards specific probability distribution (just general regularity requirements)
- makes no assumptions about the number of different classes

Also note that:

- *stationary point can be saddle point! (use postprocessing)
- naive implementation requires iterating from every pixel in Ψ (***slow!***)

Mode Based Segmentation

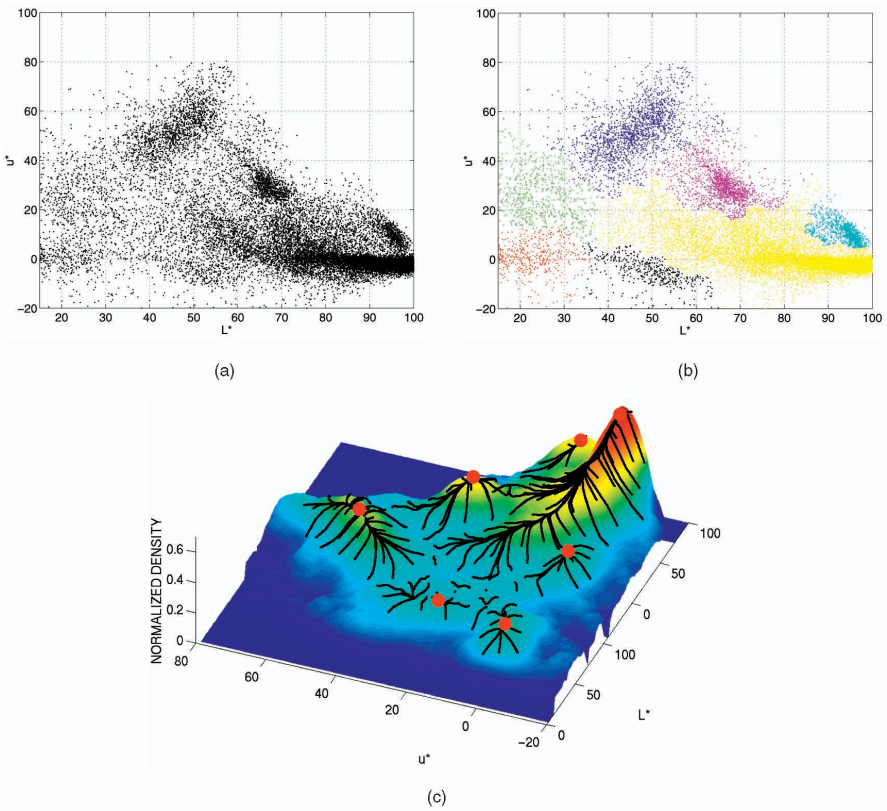


Figure from Comaniciu and Meer 2002

- (a) empirical density
- (b) mode based labels
- (c) mean shift paths

Outline

Direct Density Estimation

Definition

Recall density estimator $\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i)$,
a manifold over feature space $\mathcal{F} \subset \mathbb{R}^n$

Definition

Suppose there is a grid $\mathcal{G}$ covering $\mathcal{F}$. $\mathcal{G}$ is composed of a set of hypercube cells $\mathcal{S}$, and is equipped with a quantizing function $Q(x) : \mathcal{F} \mapsto \mathcal{S}$. The edge size of the hypercubes is equal to bandwidth h, such that $\|s_i - s_{i+1}\|_\infty = h$ for edge-connected cells.

Direct Density Estimation

We can attempt to compute the density estimator directly through histogramming the image features.

$$\sum_{i=1}^{I} K_h(x - x_i) \approx \sum_{r \in \mathcal{S}} K_1(p - r) b(r) = D(p)$$

$$b(r) = \sum_{i=1}^{I} \delta(r - Q(x_i))$$

The grid density estimator $D(p)$ over $\mathcal{G}$ is the convolution of unit-bandwidth kernel K_1 with the N-dimensional histogram function $b(r)$.

Grid Mean Shift

Consider the mean shift iterate at feature vector x

$$x + m(x) = \frac{\sum_{i=1}^{I} x_i g_h(x - x_i)}{\sum_{i=1}^{I} g_h(x - x_i)}$$

- Recall if K is Gaussian, then g is Gaussian
  $\rightarrow$denominator can be approximated (interpolated) from grid density $D(p)$ using nodes at the corners of the hypercube centered at $Q(x)$

Grid Mean Shift

Consider the mean shift iterate at feature vector x

$$x + m(x) = \frac{\sum_{i=1}^{I} x_i g_h(x - x_i)}{\sum_{i=1}^{I} g_h(x - x_i)}$$

Similarly approximate the numerator using a grid accumulation function $\sigma(r) = \sum_{i=1}^{I} x_i \delta(r - Q(x_i))$

$$\mu(x) = \sum_{x_i \in \mathcal{N}(x;h)} x_i g_h(x - x_i) \approx \sum_{q \in \mathcal{N}(Q(x),1)} \sigma(q) g_1(x - q)$$

$\mu(x)$ can be interpolated from the function
$\mu(r) = \sum_q \sigma(q) g_1(r - q)$

Grid Mean Shift

The mean shift vector can be approximated over the grid via an interpolation function H and the quantization function Q:

$$x + m(x) = \frac{\sum_{i=1}^{I} x_i g_h(x - x_i)}{\sum_{i=1}^{I} g_h(x - x_i)} \approx \frac{H(D, Q, x)}{H(\mu, Q, x)}$$

Topological Mode Seeking

Suppose we know a manifold estimating the probability density over our ND feature space. Imagine the following thought experiment in 3D space (e.g. mountains and foothills):

Move a plane normal to the "altitude" axis from as high as possible to as low as possible. Only consider points above the plane:

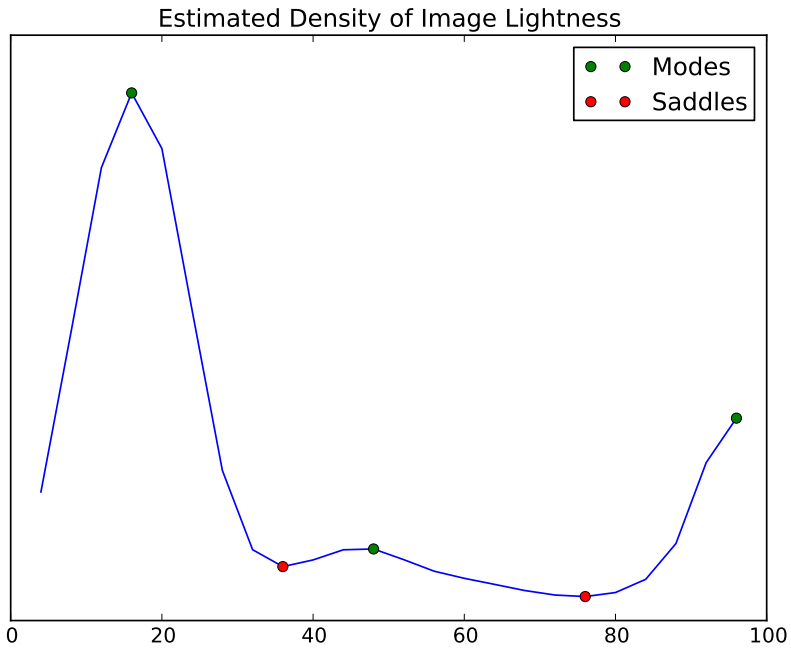
- Modes emerge one by one as isolated points on the plane
- Modes are isolated until we reach saddle points, at which point boundaries form

Topological Hierarchy + Mean Shift

Using the topological information, we can build the following segment system

1. Directly estimate the density manifold by smoothing the histogram function with a Gaussian kernel
2. Sort the density values from high to low
3. Stepping through points, consider the following cases:
   1. if isolated, then label a new mode
   2. if cell-edge neighbors are unimodal, then label as same mode
   3. if cell-edge neighbors are multimodal, then
      1. mark as a boundary
      2. if it is the highest point connecting modes (i.e. the modes are not already connected), also mark as a saddle point
4. After visiting all ponits, use (grid) mean shift to resolve the mode of boundary points.

1D Example



- saddles in 1D? Just an example...
- walking downhill from modes uniquely identifies labels of all points (until boundaries are discovered)

Mode Based Segmentation II

Definition

Given a label manifold Λ over the feature space, we partition all pixels z in image $\Psi(z)$ into classes $L_k \equiv \{z : \Lambda(x(z)) = M_k\}$

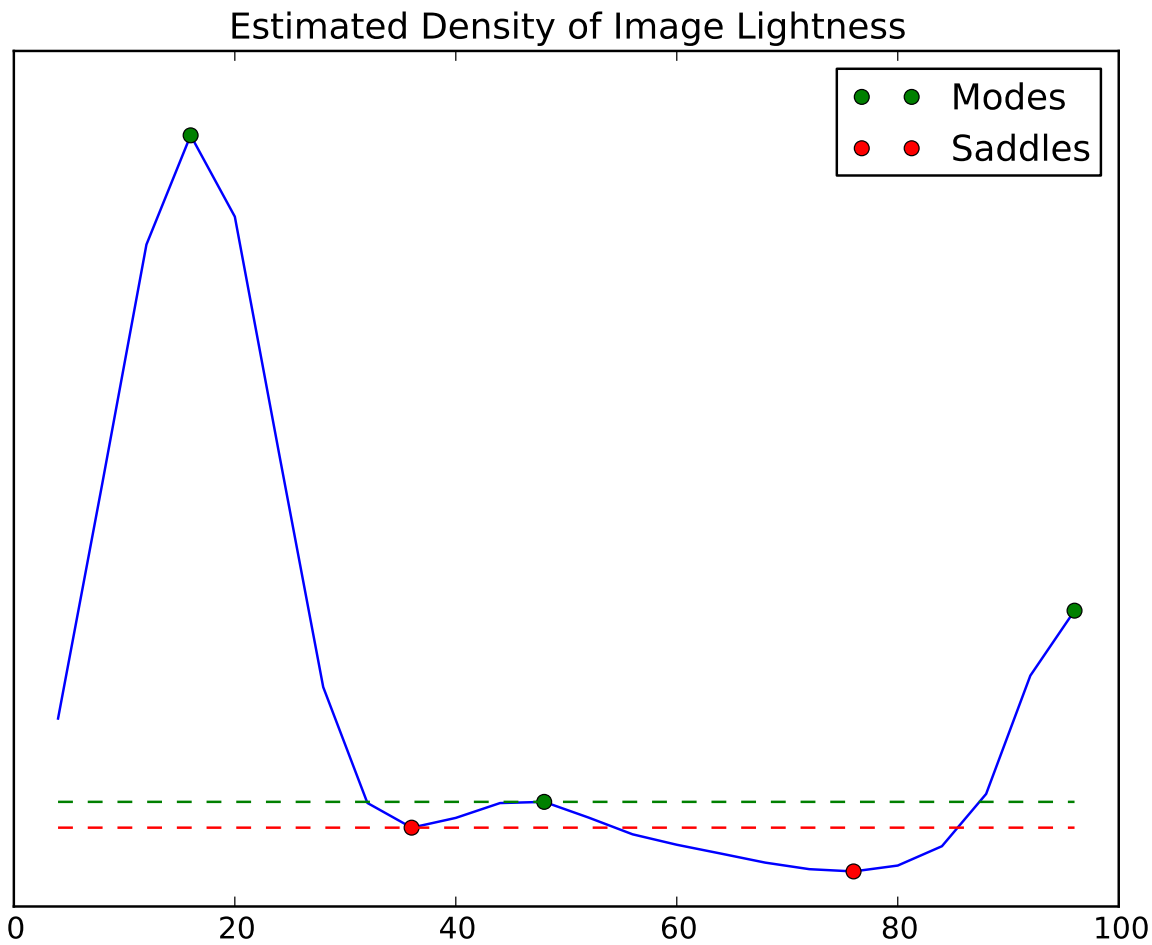
- an equivalent form of mode based segmentation
- just use the label map over feature space to determine the label of pixel z

Mode Persistence

The ***persistence*** of a mode is its distance to the highest neighboring saddle.

- bigger persistence $\rightarrow$ better mode separation $\rightarrow$ more clear image partition

Mode Persistence



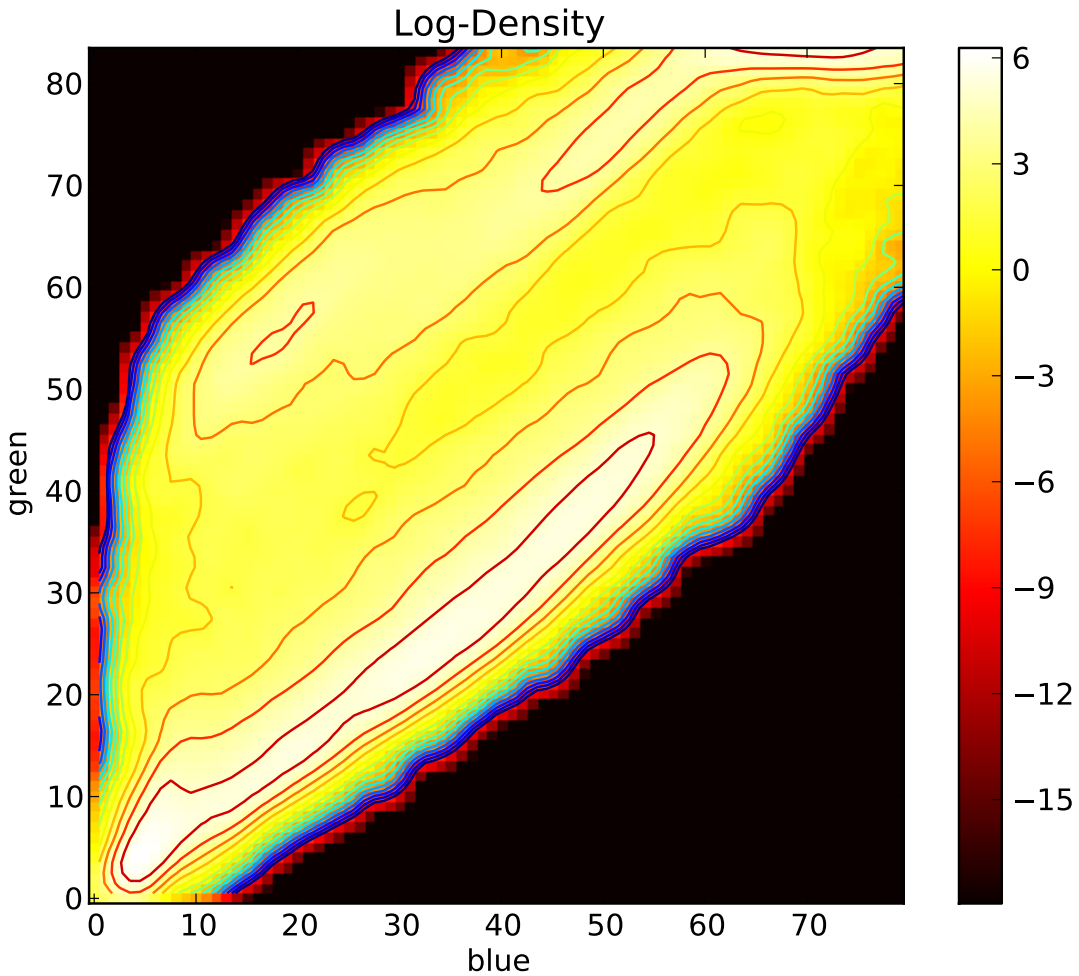
Hierarchical Clustering

Modes and Saddles form a disconnected graph

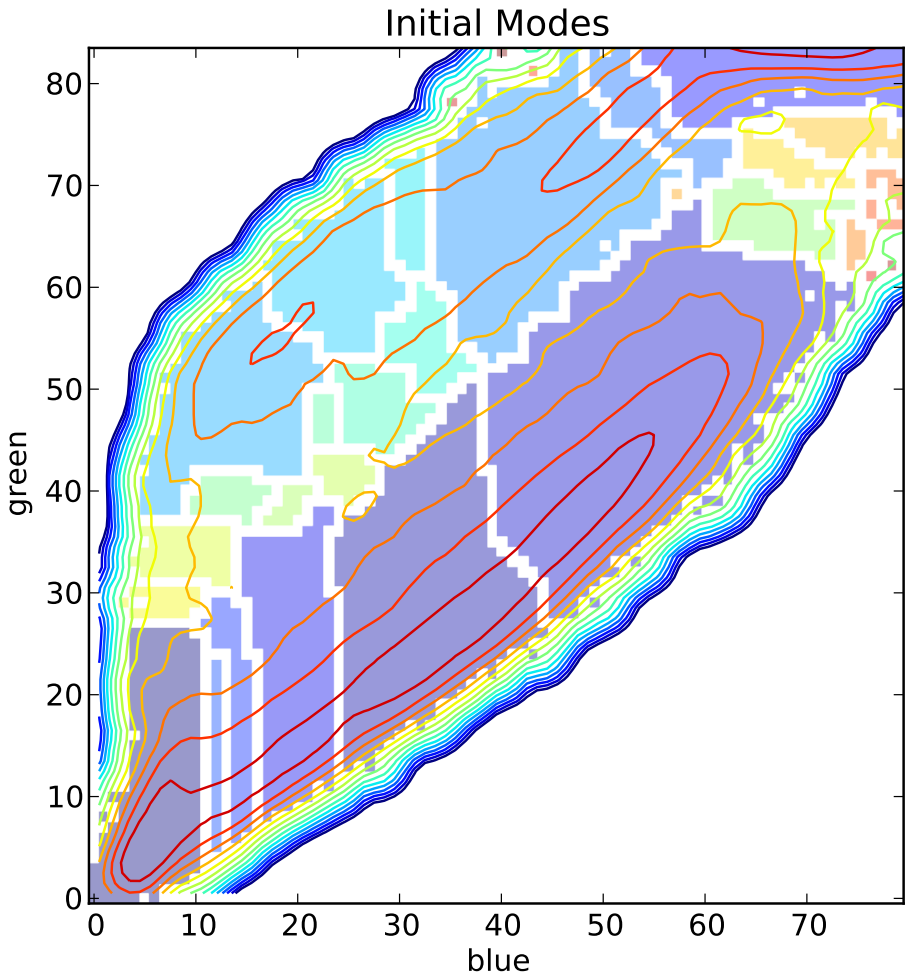
- Nodes are Modes
- Edges are Saddles, with edge weights being Persistence

We can merge clusters hierarchically based on the persistence.
Reversible, and not recursive: i.e. no need to recompute labeling function!

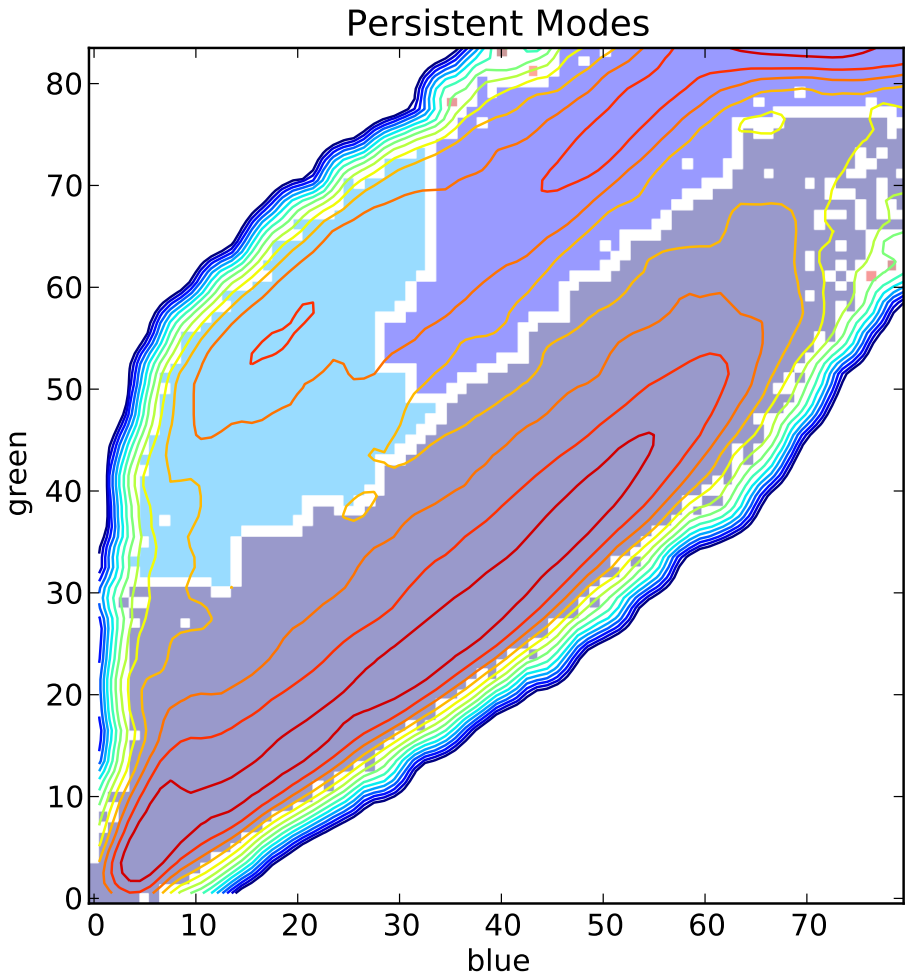
2D Example



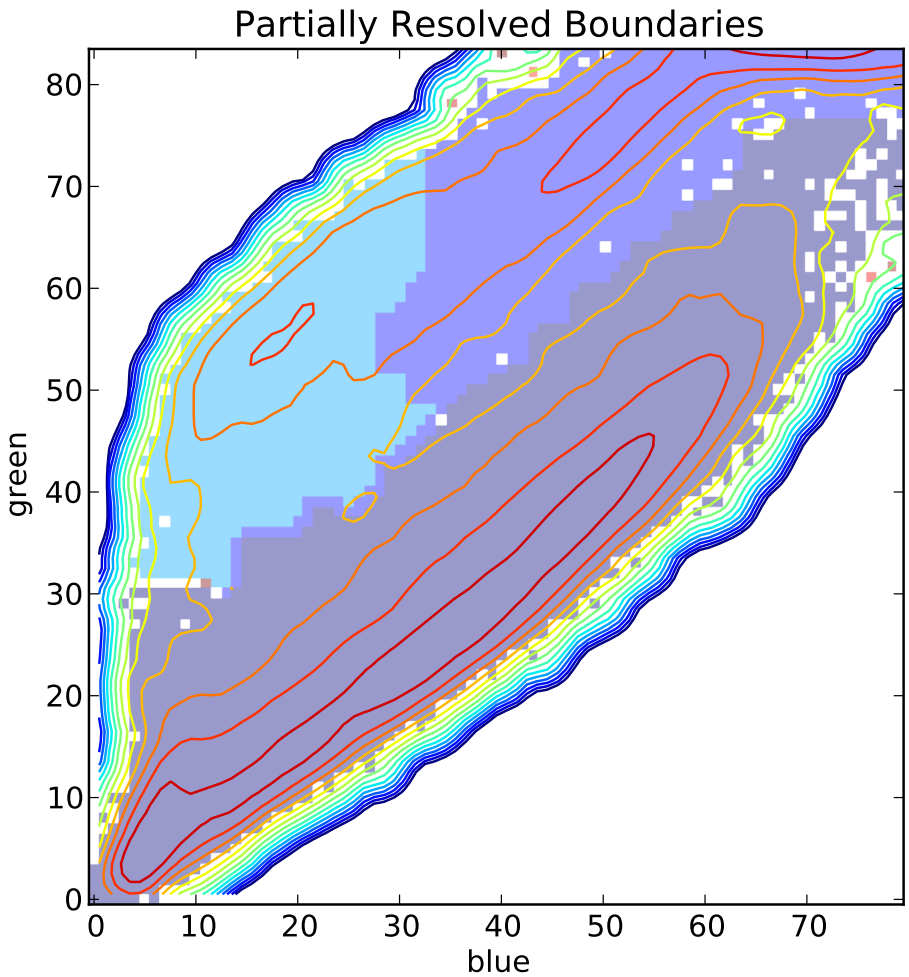
2D Example



2D Example



2D Example


Partially Resolved Boundaries
green
blue
0
10
20
30
40
50
60
70
80

Outline

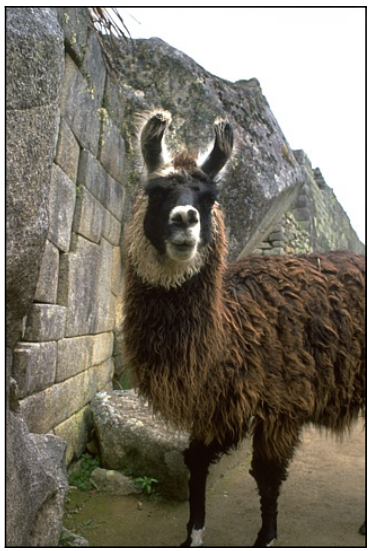
Examples

Easy, good spatial/color contrast. Solution: medium color bandwidth, large spatial bandwidth

(raw image from BSDS500 dataset [?])

Examples

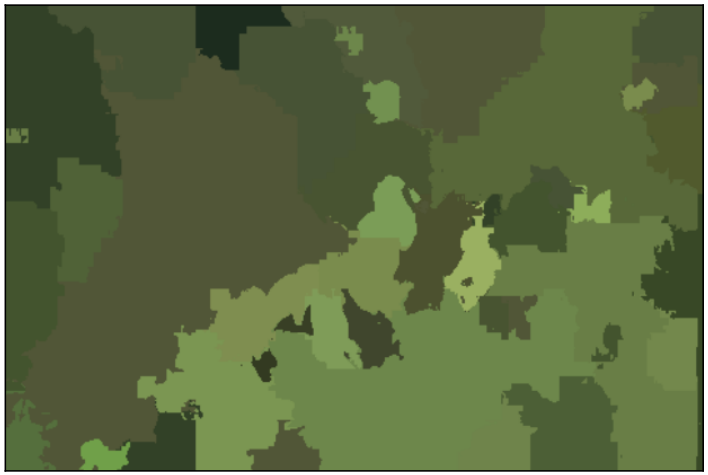
Harder, poor spatial/color contrast. Solution: high color bandwidth, medium spatial bandwidth



(raw image from BSDS500 dataset [**?**])

Examples

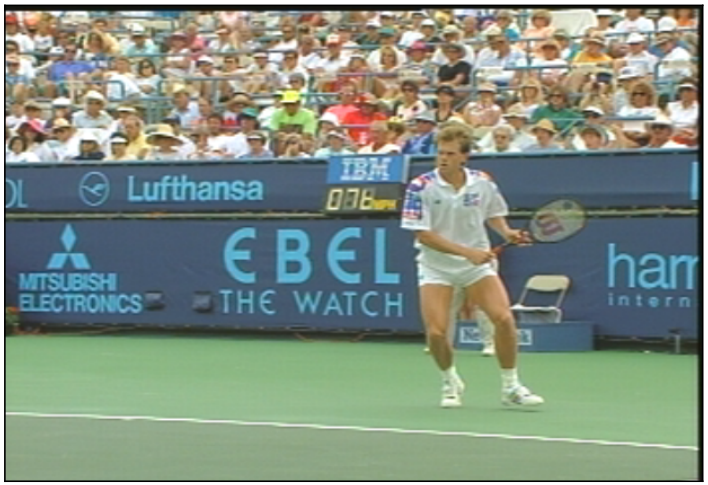
Worst, awful spatial/color contrast. Solution: high color/spatial bandwidth



(raw image from BSDS500 dataset [?])

Application: Compression/Computer Sketch

The computer can make a "child's cartoon" of a scene



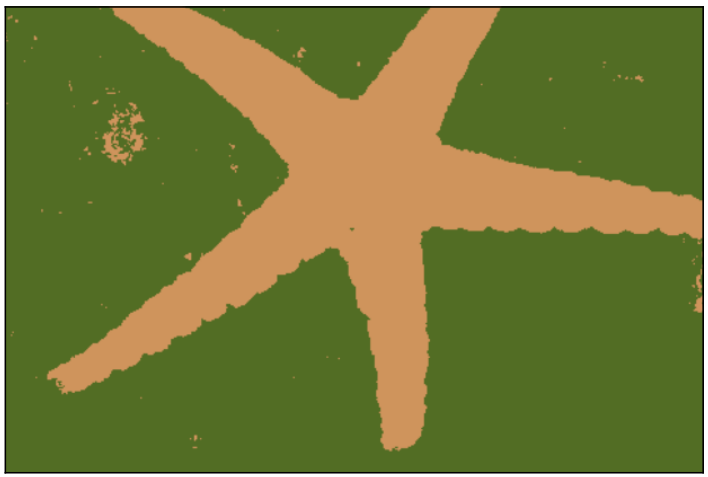
Application: Compression/Computer Sketch

I've taught the computer to make an *artistic impression* of a scene

(raw image from BSDS500 dataset [**?**])

Application: Object Detection?

Merge modes with smallest persistence until there are up to N modes.
2 modes:



(raw image from BSDS500 dataset [**?**])

Application: Object Detection?

3 modes:

(raw image from BSDS500 dataset [**?**])

Application: Tracking?

Given an estimate of the probability over colorspace, label subsequent frames

Ex: Crew Sequence
Ex: Tennis Sequence

- Questionable approach:
  - classifies entire scene–slow
  - object is not "tracked" but repeatedly discovered in each farme

Outline

Object Model

- Can be specific to an object class
  - E.G. deformable contours constrained to match *a priori* templates like faces or car frames
- Can be simple shapes
  - ellipses, boxes, rectangles

We choose simple shapes, and track based on the local probability density estimate

Distribution Matching Problem

Definitions

In one frame, we have a "model" distribution located at the centroid of the model object $q(u; x_0)$.

We want to find the best "target" distribution $p(u; y)$ in each new frame, the center of which is the new location of the model object

notation switches between $p_u(y)$ ***to emphasize variation with*** y***, and*** $p(u; y)$ ***to emphasize that the density is over*** u***.***

Distribution Matching Problem

Definitions

Use metric based on the Bhattacharrya coefficient of two densities:
$\rho(p, q) = \sum_{u=1}^{m} \sqrt{p_u q_u}$

The distance is: $d(p, q) = \sqrt{1 - \rho(p, q)}$

Local Density Estimates

The density estimators within a small window parameterized by bandwidth h are kernel density estimators

$$p(u;y) = \sum_{x_i \in \mathcal{N}(y;h)} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \delta[b(x_i) - u] \quad 1 \leq u \leq m$$

- The "quantizing" function b is a feature map from pixel location $x \in \mathbb{R}^2$ to feature space "bin" $u \in \mathcal{F}$.
- The local density estimator is a spatially-weighted histogram over m bins–I.E., each histogram count is weighted by where that feature originates in space

Optimizing the Match

Clearly to minimize the distance between $q(u; x_0)$ and $p(u; y)$, the Bhattacharrya coefficient must be maximized.
Expanding $\rho(p, q)$ with only linear terms, we have

$$\begin{aligned}
\rho(p+s, q) &\approx \sum_{u=1}^{m} \sqrt{p_u q_u} + \frac{1}{2} \sum_{u=1}^{m} s_u \sqrt{\frac{q_u}{p_u}} \\
\text{if } p_u(y) &= p_u(y_0) + s_u(dy) \\
\rightarrow \rho(p(y), q) &\approx \frac{1}{2} \sum_{u=1}^{m} \sqrt{p_u(y_0) q_u} + \frac{1}{2} \sum_{u=1}^{m} p_u(y) \sqrt{\frac{q_u}{p_u(y_0)}}
\end{aligned}$$

Optimizing the Match

The free variable is y, so we are left maximizing the term

$$\frac{1}{2}\sum_{u=1}^{m} p_u(y)\sqrt{\frac{q_u}{p_u(y_0)}}$$

But,

$$p(u;y) = \sum_{x_i \in \mathcal{N}(y)} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right)\delta[b(x_i) - u]$$

Optimizing the Match

So we maximize

$$\frac{1}{2}\sum_{x_i \in \mathcal{N}(y)} w_i k\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \qquad w_i = \sqrt{\frac{q(b(x_i); x_0)}{p(b(x_i); y_0)}}$$

Look familiar?
This is a kernel density estimator (KDE) over the weights w_i–we can find the local mazimum using ***mean shift***

Tracking Principle

Given a model distribution $q(u; x_0)$ and a starting point y_0, find the mean shift path

$$y^{(k+1)} = \frac{\sum_{x_i \in \mathcal{N}(y^{(k)})} x_i w_i g\left(\left\| \frac{y^{(k)} - x_i}{h} \right\|^2\right)}{\sum_{x_i \in \mathcal{N}(y^{(k)})} w_i g\left(\left\| \frac{y^{(k)} - x_i}{h} \right\|^2\right)}$$

But, re-weight w_i each step

Mean Shift Tracking Algorithm

Given $\{q_u\}$ of the model, and location x from the previous frame

1. $k \leftarrow 0$, initialize location in the current frame as $y^{(0)} = x$
2. Compute $p(u; y^{(k)})$ and weights w_i
3. Apply mean shift to step to new location

$$y^{(k+1)} = \frac{\sum_{x_i \in \mathcal{N}(y^{(k)})} x_i w_i g\left(\left\| \frac{y^{(k)} - x_i}{h} \right\|^2\right)}{\sum_{x_i \in \mathcal{N}(y^{(k)})} w_i g\left(\left\| \frac{y^{(k)} - x_i}{h} \right\|^2\right)}$$

4. check convergence
   1. if $\|y^{(k+1)} - y^{(k)}\|_\infty < 1$ then converged
   2. else $k \leftarrow k + 1$, go to **(2)**

Outline

Performance

- If motion is not too great, then starting locations are close to local maxima and convergence is fast

Outline

Enhancements

- Adaptive scale
- Continuation when object leaves and enters frame
- Second order optimization in Gaussian case (Newton step)

Conclusion

- mean shift is
  - efficient
  - adaptable
- mode based classification can be applied for various tasks in imaging
  - segmentation
  - tracking