Robert Collins
CSE598C, PSU

Mean-shift and Color Histograms

R.Collins, CSE, PSU

CSE598C Fall 2012

Robert Collins
CSE598C, PSU

Generalizing to Weighted Points

Let each point sample x_i have an associated <u>nonnegative</u> weight $w(x_i)$

Can rederive the equations with $w(x_i)$ factors in them:

$$f(x) = \sum H(x_i - x)\, w(x_i)$$

KDE using shadow kernel H

$$p(x) = \sum K(x_i - x)\, w(x_i)$$

KDE using kernel K

$$\frac{\nabla f(x)}{c\, p(x)} = \frac{\sum K(x_i - x)\, w(x_i)\, x_i}{\sum K(x_i - x)\, w(x_i)} - x$$

mean shift vector is still a gradient ascent process

This is important for running on images. Since pixels form a lattice, spatial density of samples is fixed, so need a weight to denote sample density at each point.

Robert Collins
CSE598C, PSU

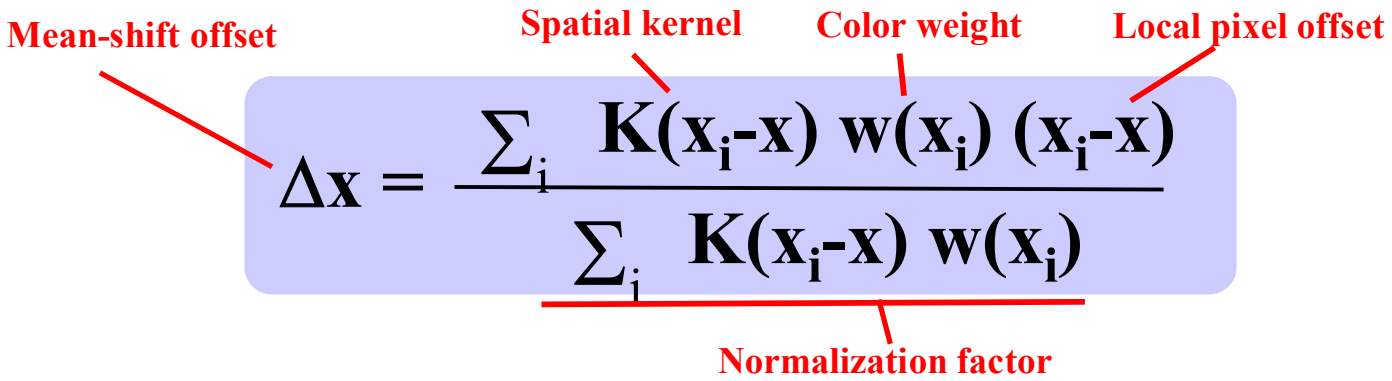
Story So Far...

Given a set of sample points, mean shift does mode-seeking to find locations having a high density of samples.

Mean-shift uses a spatial kernel to 1) define the size of a local neighborhood and 2) to specify weights of points within the neighborhood.

Mean-shift using a spatial kernel K can be understood as doing hill-climbing on a surface estimated from the sample points using kernel density estimation with a shadow kernel of H, where in terms of profile functions, h'(r) = - c k(r)

When used on pixels, the "points" are now centers of pixels, which form a uniform grid. Do do anything useful, we need to specify another weight for each pixel. Typically, this weight is based on color similarity with a model histogram.



Robert Collins
CSE598C, PSU

Mean-shift on Weight Images

Ideally, we want an indicator function that returns 1 for pixels on the object we are tracking, and 0 for all other pixels

Instead, we compute likelihood maps where the value at a pixel is proportional to the likelihood that the pixel comes from the object we are tracking.

Computation of likelihood can be based on

- color
- texture
- shape (boundary)
- predicted location

Claim: these weight images are all the mean-shift algorithm "sees", whether they be explicitly computed (e.g. Bradski) or implicitly computed (e.g. Comaniciu, Ramesh and Meer).

Robert Collins
CSE598C, PSU

Explicit Weight Images

histogram backprojection

histogram is an empirical estimate of p(color | object) = p(c | o)

Bayes rule says p(o | c) = p(c | o) p(o) / p(c)

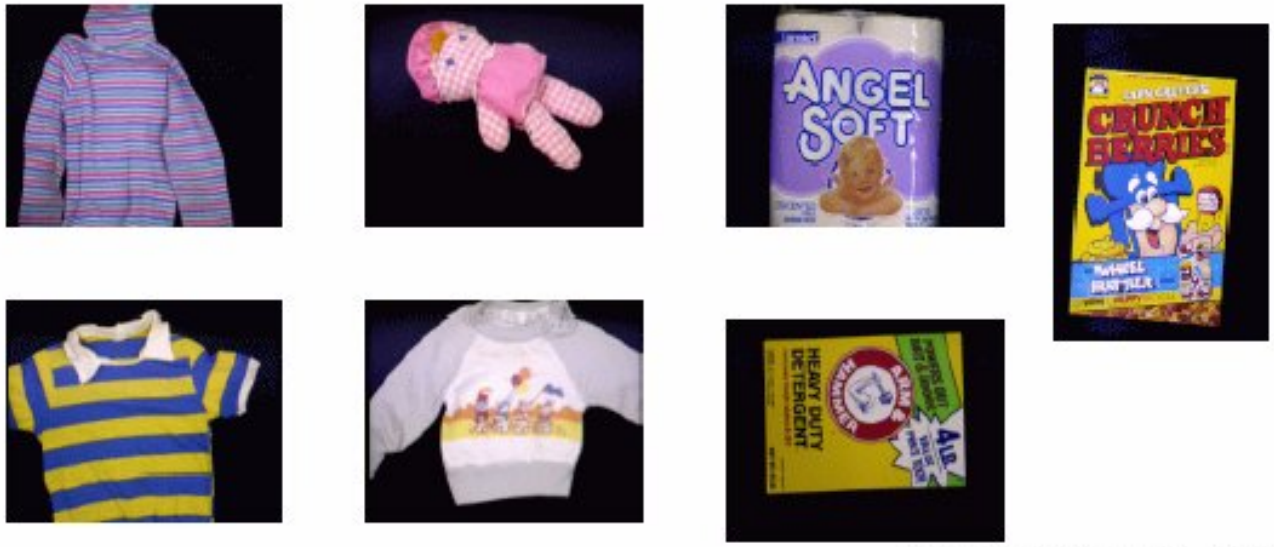
Simplistic approx: assume (p(o)/p(c)) is constant. Then p(o|c) = p(c|o).

Use histogram h as a lookup table to set pixel values in the weight image. (if pixel maps to bucket i in histogram h, set weight for that pixel to h(i)

Sidebar: Swain and Ballard, 1991

Using color histograms for recognition.

- Works surprisingly well
- In the first paper (1991), 66 objects could be recognized almost without errors



[Swain & Ballard, 1991]

Swain and Ballard

- Histogram backprojection:
  - „Where in the image are the colors being looked for?"
  - Query: object with histogram M
  - Given: image with histogram I
  - Compute the „ratio histogram":

$$R_i = \min\left(\frac{M_i}{I_i}, 1\right)$$

    - R reveals how important an object color is, relative to the current image.
    - This value is projected back into the image (i.e. the image values are replaced by the values of R that they index).
    - The result image is convolved with a circular mask the size of the target object.
    - Peaks in the convolved image indicate detected objects.

22

does this sound familiar?

Swain and Ballard

Object Localization

- Example result after backprojection
  - Looking for blue pullover...

Note: relationship between recognition and tracking. This will come up again later.

[Swain & Ballard, 1991]

Robert Collins
CSE598C, PSU

Implicit Weight Images

Sometimes the weight image is not explicitly created. An example is Comaniciu, Ramesh and Meer. Here, the "weight image" is embedded in the procedure (taking derivatives of bhattacharyya coeff with respect to image location of the window of samples).

Interesting point: their weight image changes between iterations of mean-shift, as compared to iterating to convergence on an explicit weight image!

Robert Collins
CSE598C, PSU

Comaniciu et.al.

Color Histogram Representation:

target model: $\hat{\mathbf{q}} = \{\hat{q}_u\}_{u=1...m}$ $\quad \sum_{u=1}^{m} \hat{q}_u = 1$

target candidate: $\hat{\mathbf{p}}(\mathbf{y}) = \{\hat{p}_u(\mathbf{y})\}_{u=1...m}$ $\quad \sum_{u=1}^{m} \hat{p}_u = 1$.

Distance between histograms measured by:

$$d(\mathbf{y}) = \sqrt{1 - \rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}]} ,$$

note: this is a function of window of location y

where

$$\hat{\rho}(\mathbf{y}) \equiv \rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(\mathbf{y})\hat{q}_u} ,$$

Bhattacharyya coefficient

Comaniciu et.al.

the histograms are computed via Parzen estimation:

$$\hat{q}_u = C \sum_{i=1}^{n} k(\|\mathbf{x}_i^\star\|^2) \delta\left[b(\mathbf{x}_i^\star) - u\right] ,$$

$$\hat{p}_u(\mathbf{y}) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{\mathbf{y} - \mathbf{x}_i}{h}\right\|^2\right) \delta\left[b(\mathbf{x}_i) - u\right] ,$$

where k is some radially symmetric smoothing kernel (profile)

This gathers histograms over a local neighborhood. Also allows interpolation of histograms centered around an off-lattice location.

Robert Collins
CSE598C, PSU

Comaniciu et.al.

via Taylor series expansion about current values $p_u(y_0)$

this does not depend on y

so just need to maximize this.
Note: it is a KDE!!!!

$$\rho[\hat{\mathbf{p}}(\mathbf{y}),\hat{\mathbf{q}}] \approx \frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{p}_u(\hat{\mathbf{y}}_0)\hat{q}_u} + \frac{C_h}{2}\sum_{i=1}^{n_h} w_i k\left(\left\|\frac{\mathbf{y}-\mathbf{x}_i}{h}\right\|^2\right)$$

where $$w_i = \sum_{u=1}^{m}\sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{\mathbf{y}}_0)}}\delta\left[b(\mathbf{x}_i)-u\right] .$$

find mode of second term by mean-shift iterations:

$$\hat{\mathbf{y}}_1 = \frac{\sum_{i=1}^{n_h}\mathbf{x}_i w_i g\left(\left\|\frac{\hat{\mathbf{y}}_0-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{\hat{\mathbf{y}}_0-\mathbf{x}_i}{h}\right\|^2\right)} \quad \text{where } g(x) = -k'(x)$$

Comaniciu et.al.

At each iteration:

$$\hat{\mathbf{y}}_1 = \frac{\sum_{i=1}^{n_h} \mathbf{x}_i w_i g\left(\left\|\frac{\hat{\mathbf{y}}_0 - \mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{\hat{\mathbf{y}}_0 - \mathbf{x}_i}{h}\right\|^2\right)}$$

which is just standard mean-shift on (implicit) weight image w_i

Let's look at the weight terms more closely. For each pixel x_i:

$$w_i = \sum_{u=1}^{m} \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{\mathbf{y}}_0)}}\, \delta\left[b(\mathbf{x}_i) - u\right] .$$

This term is only 1 once in the summation

If pixel x_i's value maps to histogram bucket B,

then w_i = sqrt(q_B / $p_B(y_0)$)

Comaniciu et.al.

So if model histogram is $q_1, q_2, ..., q_m$

and current data histogram is $p_1, p_2, ..., p_m$

form weights $q_1/p_1, q_2/p_2, ..., q_m/p_m$

and then do "histogram backprojection" of these values into the image to get the weight image w_i

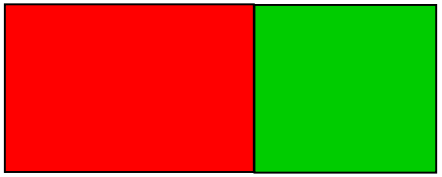
note to self:
Swain and Ballard

also note, $p_1, p_2, ..., p_m$ changes at each iteration, and therefore so does the weight image w_i

Robert Collins
CSE598C, PSU

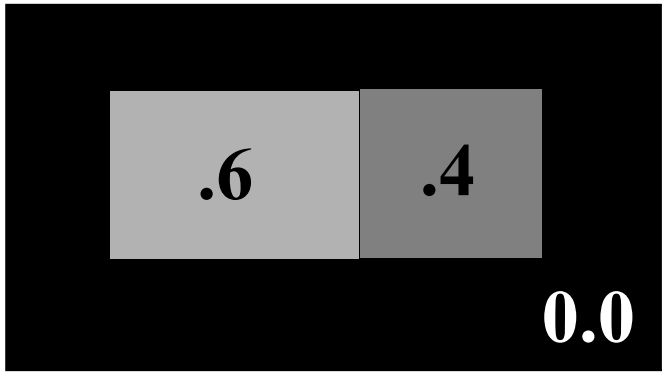
Qualitative Intuition

Assume some object that
is 60% red and 40%green



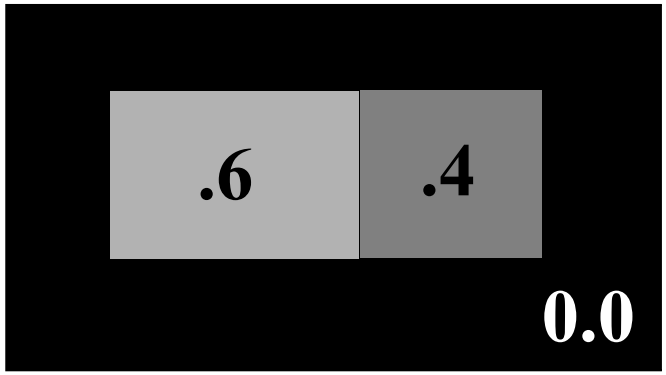
q1 = .6, q2 = .4, qi = 0 for all other i

If we just did histogram backprojection of these likelihood values (ala Bradski), we would get a weight image



Robert Collins
CSE598C, PSU

Qualitative Intuition



Mean shift does a weighted center of mass computation at each interation

Mean shift window will be biased towards the region of red pixels, since they have higher weight

Robert Collins
CSE598C, PSU

Qualitative Intuition

Now use Comaniciu et.al.'s weights
Let's say the data histogram is perfectly located

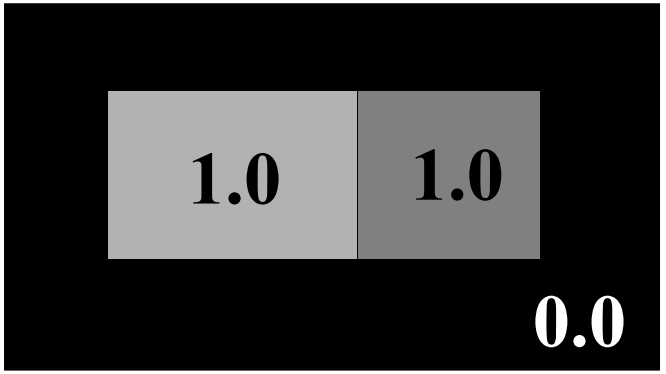
$$q1 = .6,\ \ q2 = .4,\ \ qi = 0 \text{ for all other } i$$

say something about 0 values

$$p1 = .6,\ \ p2 = .4,\ \ pi = 0 \text{ for all other } i$$

$$w1 = \text{sqrt}(.6/.6),\ \ w2 = \text{sqrt}(.4/.4),\ \ wi = 0 \text{ for all other } i$$

Resulting weight image:



This is the indicator function image we always hope for!

Robert Collins
CSE598C, PSU

Qualitative Intuition

Say we have too little percentage of red in data hist

$$q1 = .6,\ \ q2 = .4,\ \ qi = 0 \text{ for all other } i$$

$$p1 = .5,\ \ p2 = .5,\ \ pi = 0 \text{ for all other } i$$

$$w1 = \text{sqrt}(.6/.5),\ \ w2 = \text{sqrt}(.4/.5),\ \ wi = 0 \text{ for all other } i$$

$$>1 \qquad\qquad <1$$

So red pixels will be favored in center of mass computation, hopefully causing more of them to be included at the next iteration.

Robert Collins
CSE598C, PSU

Qualitative Intuition

Say we have very little percentage of red in data hist

$$q1 = .6,\ \ q2 = .4,\ \ qi = 0 \text{ for all other } i$$

$$p1 = .2,\ \ p2 = .8,\ \ pi = 0 \text{ for all other } i$$

$$w1 = \text{sqrt}(.6/.2),\ \ w2 = \text{sqrt}(.4/.8),\ \ wi = 0 \text{ for all other } i$$

$$>>1 \qquad\qquad <<1$$

So red pixels will be even more highly weighted relative to green pixels.

Robert Collins
CSE598C, PSU

Qualitative Intuition

Say we have too much percentage of red in data hist

$$q1 = .6,\ \ q2 = .4,\ \ qi = 0 \text{ for all other } i$$

$$p1 = .7,\ \ p2 = .3,\ \ pi = 0 \text{ for all other } i$$

$$w1 = \text{sqrt}(.6/.7),\ \ w2 = \text{sqrt}(.4/.3),\ \ wi = 0 \text{ for all other } i$$

$$<1 \qquad\qquad >1$$

So green pixels will now be favored.

Other Features

We've only talked about color, but of course we could use histograms of other features like edge orientations (or any filter-bank response)

However, one point I've been trying to make today is that we don't need histograms at all. We just need a way to label pixels with the likelihood that they belong to the object (rather than the background).

That is, we just need to be able to specify a weight image, either explicitly or implicitly.

Robert Collins
CSE598C, PSU

Discuss using mean-shift to find modes in correlation surfaces

Explicit is easy. Implicit??