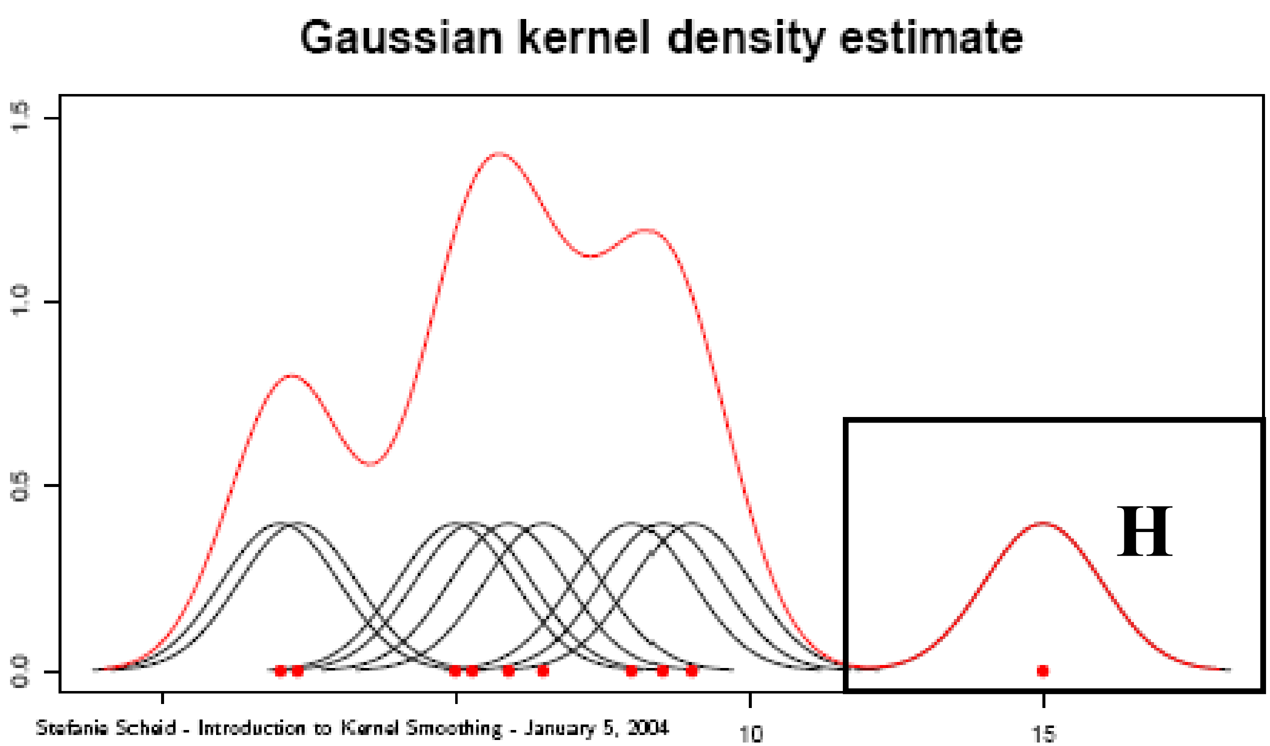
Mean-shift, continued

R.Collins, CSE, PSU

CSE598C Fall 2012

Background: Kernel Density Estimation

Given a set of data samples x_i; $i=1...n$
Convolve with a kernel function H to generate a smooth function $f(x)$
Equivalent to superposition of multiple kernels centered at each data point



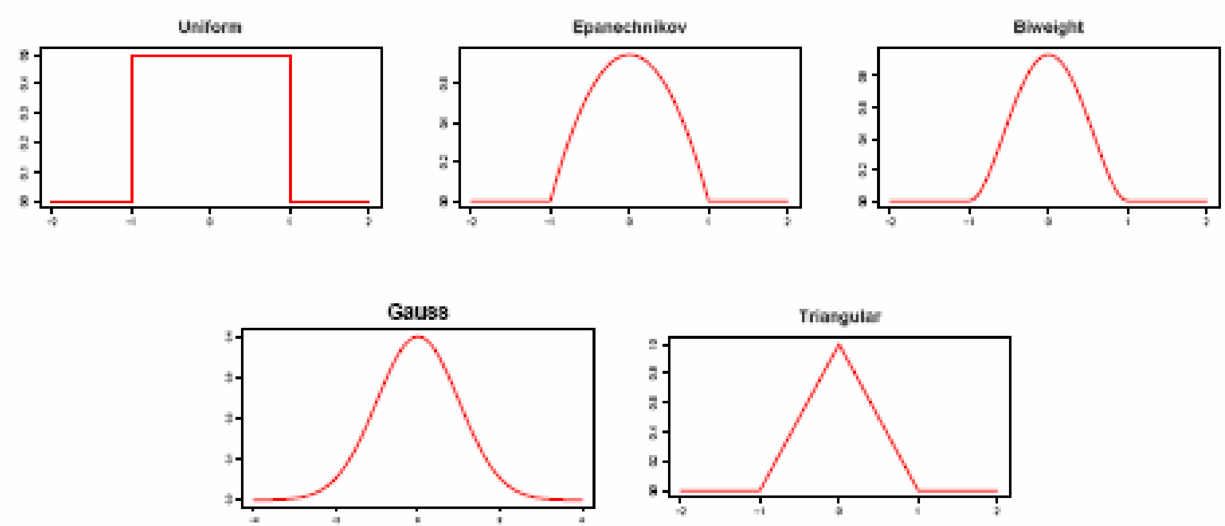
Kernel Density Estimation

For kernel H with bandwidth h, estimated function value f at location x is

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

superposition of kernels centered at x_i

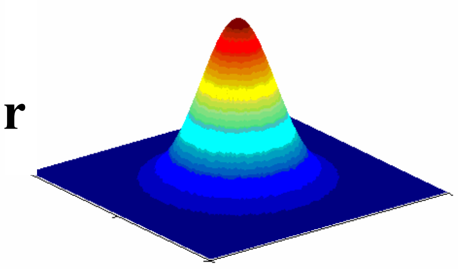
some sample kernels:



Stefanie Scheid - Introduction to Kernel Smoothing - January 5, 2004

Robert Collins
CSE598C, PSU

Radially Symmetric Kernels



Height at point is function only of distance from center

Can be written in terms of a 1D <u>Profile</u> function that is is a function of the radius (we will use squared radius below)

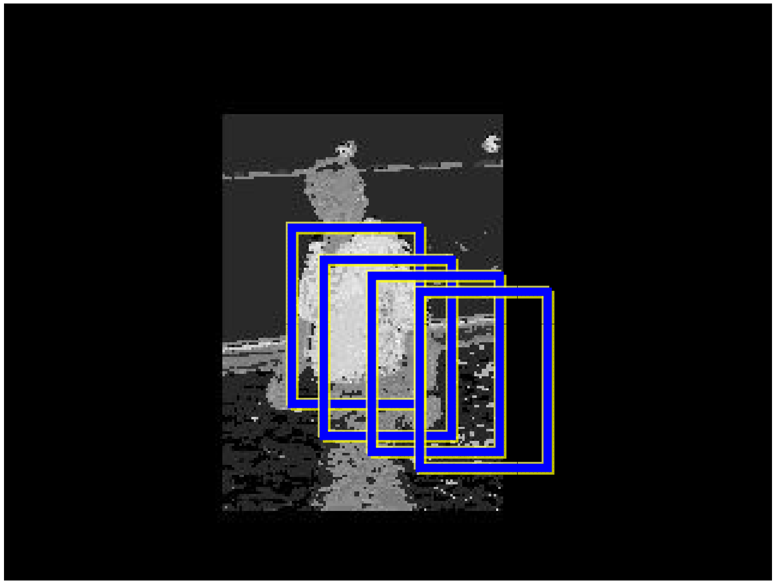
$$H(x_i - x) \equiv h(\|x_i - x\|^2)$$

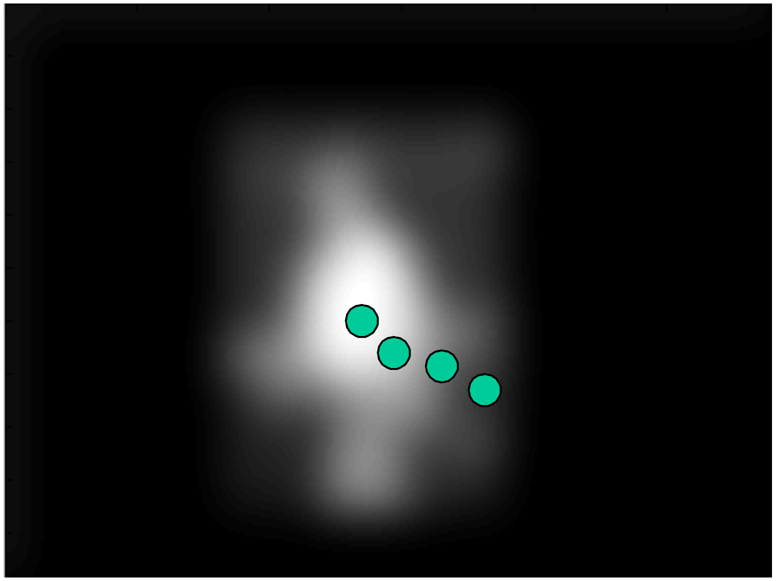
$$\equiv h((x_i - x)^T (x_i - x))$$

$$\equiv h(r) \qquad r(x) = (x_i - x)^T (x_i - x)$$

Recall: Nice Property

Running mean-shift with kernel K on weight image w is equivalent to performing gradient ascent in a (virtual) image formed by convolving w with some "shadow" kernel H.





Computational savings: only have to compute convolution values at the relatively small number of points you visit on the path to the mode.

Robert Collins
CSE598C, PSU

Kernel-Shadow Pairs

Given a convolution kernel H, what is the corresponding mean-shift kernel K?
Perform change of variables $r = ||x_i - x||^2$
Rewrite $H(x_i-x) \Rightarrow h(||x_i-x||^2) \Rightarrow h(r)$.

Then kernel K must satisfy **$h'(r) = - c\, k(r)$**

Examples

Epanichnikov	$h(r) = 1 - r \quad h'(r) = -1 \quad \Rightarrow k(r) = 1$	**Flat**
Biweight	$h(r) = (1-r)^2 \quad h'(r) = -2(1-r) \Rightarrow k(r) = 1 - r$	**Epanichnikov**
Gaussian	$h(r) = e^{-\beta r} \quad h'(r) = -\beta e^{-\beta r} \Rightarrow k(r) = e^{-\beta r}$	**Gaussian**

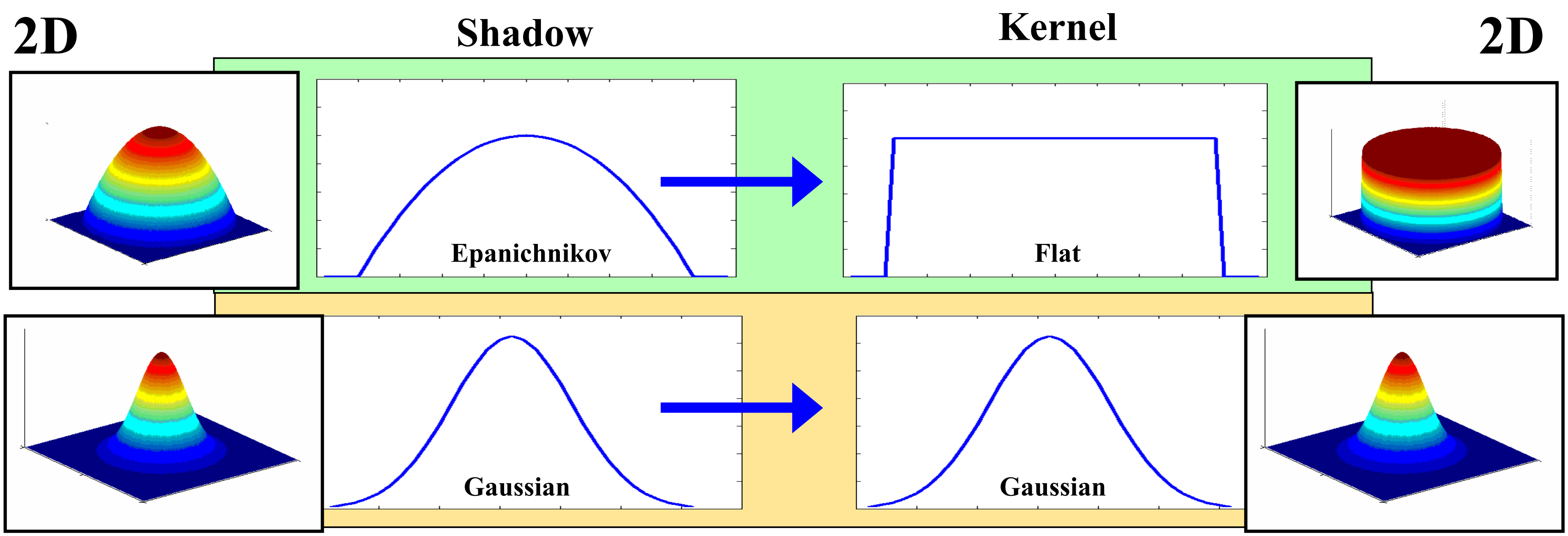
self-replicating!

Kernel-Shadow Pairs

Given a convolution kernel H, what is the corresponding mean-shift kernel K?
Perform change of variables $r = \|x_i - x\|^2$
Rewrite $H(x_i - x) \Rightarrow h(\|x_i - x\|^2) \Rightarrow h(r)$.

Then kernel K must satisfy $$h'(r) = - c\, k\, (r)$$

Examples



Robert Collins
CSE598C, PSU

Mean-Shift and Gradient Ascent

We will derive an explicit equation relating the mean-shift procedure using kernel K with gradient ascent on the KDE surface formed by using the shadow kernel H.

KDE $f(x) = \sum H(x_i - x)$

$= \sum h(\|x_i - x\|^2)$ **rewrite using profile function**

gradient of KDE

$\nabla f(x) = \nabla \sum h(\|x_i - x\|^2) = \sum \nabla h(\|x_i - x\|^2)$

Robert Collins
CSE598C, PSU

Mean-Shift and Gradient Ascent

$$\nabla f(x) = \nabla \sum h(\|x_i - x\|^2) = \sum \nabla h(\|x_i - x\|^2)$$

Sidebar derivation:

$$\nabla h = \frac{\partial h}{\partial x} = \frac{\partial h(r)}{\partial x}$$

change of variables

$$r(x) = (x_i - x)^T (x_i - x)$$

$$= \frac{\partial h}{\partial r} \frac{\partial r(x)}{\partial x}$$

chain rule

$$= h'(r) \frac{\partial}{\partial x} (x_i^T x_i - 2x^T x_i + x^T x)$$

$$= h'(r) [-2x_i + 2x]$$

$$= -2h'(r)(x_i - x)$$

$$= -2\, h'(\|x_i - x\|^2)(x_i - x)$$

change variables back

Robert Collins
CSE598C, PSU

Mean-Shift and Gradient Ascent

We will derive an explicit equation relating the mean-shift procedure using kernel K with gradient ascent on the KDE surface formed by using the shadow kernel H.

$$\text{KDE} \quad f(x) = \sum H(x_i - x)$$

$$= \sum h(\|x_i - x\|^2)$$ **rewrite using profile function**

gradient of KDE

$$\nabla f(x) = \sum \nabla h(\|x_i - x\|^2)$$

$$= -2 \sum h'(\|x_i - x\|^2)(x_i - x)$$ **change of vars + chain rule**

$$= c \sum k(\|x_i - x\|^2)(x_i - x)$$ **definition of kernel shadow pairs h'(r) = - c k(r)**

$$= c \sum K(x_i - x)(x_i - x)$$ **rewrite as kernel K**

Robert Collins
CSE598C, PSU

Mean-Shift and Gradient Ascent

cont.

$$\nabla f(x) = c \sum K(x_i - x)(x_i - x)$$

$$\frac{\nabla f(x)}{c} = \sum K(x_i - x)\, x_i - \left[\sum K(x_i - x)\right] x$$

x this does not depend on i, so can come out of the summation

call this p(x).
It is another KDE, using kernel K instead of H

$$= \left[\sum K(x_i - x)\, x_i - p(x)\, x\right] \left[\frac{p(x)}{p(x)}\right]$$

this is just 1 !

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

this is the relationship we wanted to derive.

Robert Collins
CSE598C, PSU

Mean-Shift and Gradient Ascent

cont.

weighted center of mass

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

mean shift vector

$$\frac{\nabla \sum H(x_i - x)}{c \sum K(x_i - x)}$$

- **same direction as gradient of KDE_H**
- **step size is inversely proportional to KDE_K**
- **mean-shift performs gradient ascent with adaptive step size**

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

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

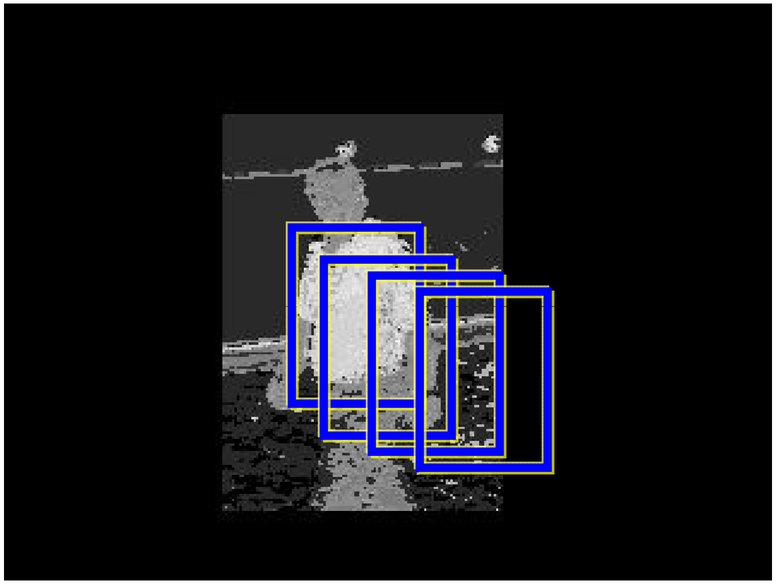
mean shift vector is still a gradient ascent process

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

Mean-Shift Tracking

Let pixels form a uniform grid of data points, each with a weight (pixel value) proportional to the "likelihood" that the pixel is on the object we want to track. Perform standard mean-shift algorithm using this weighted set of points.



$$\Delta x = \frac{\sum_i K(x_i - x)\, w(x_i)\, (x_i - x)}{\sum_i K(x_i - x)\, w(x_i)}$$

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).