Mean Shift
Theory and Applications

Yaron Ukrainitz & Bernard Sarel
Agenda

• Mean Shift Theory
  • What is Mean Shift?
  • Density Estimation Methods
  • Deriving the Mean Shift
  • Mean shift properties

• Applications
  • Clustering
  • Discontinuity Preserving Smoothing
  • Object Contour Detection
  • Segmentation
  • Object Tracking
Mean Shift Theory
Intuitive Description

Objective: Find the densest region
Distribution of identical billiard balls
Intuitive Description

Objective: Find the densest region
Distribution of identical billiard balls
Intuitive Description

Objective: Find the densest region
Distribution of identical billiard balls
Intuitive Description

Objective: Find the densest region
Distribution of identical billiard balls
Intuitive Description

Objective: Find the densest region
Distribution of identical billiard balls
Intuitive Description

Objective: Find the densest region
Distribution of identical billiard balls
Intuitive Description

Objective: Find the densest region
Distribution of identical billiard balls
What is Mean Shift?

A tool for:
Finding modes in a set of data samples, manifesting an underlying probability density function (PDF) in \( \mathbb{R}^n \)

PDF in feature space
• Color space
• Scale space
• Actually any feature space you can conceive
• …

Non-parametric Density \textit{GRADIENT} Estimation (Mean Shift)

PDF Representation

PDF Analysis
Non-Parametric Density Estimation

Assumption: The data points are sampled from an underlying PDF

Data point density implies PDF value!

Assumed Underlying PDF

Real Data Samples
Non-Parametric Density Estimation

Assumed Underlying PDF

Real Data Samples
Non-Parametric Density Estimation

Assumed Underlying PDF

Real Data Samples
**Parametric Density Estimation**

Assumption: The data points are sampled from an underlying PDF

\[
\text{PDF}(x) = \sum_i c_i \cdot e^{-\frac{(x-u_i)^2}{2\sigma_i^2}}
\]

Assumed Underlying PDF  \hspace{2cm} \text{Real Data Samples}
Kernel Density Estimation
Parzen Windows - General Framework

\[ P(x) = \frac{1}{n} \sum_{i=1}^{n} K(x - x_i) \]

A function of some finite number of data points \( x_1 \ldots x_n \)

Kernel Properties:
• Normalized
  \[ \int_{\mathbb{R}^d} K(x) \, dx = 1 \]

• Symmetric
  \[ \int_{\mathbb{R}^d} xK(x) \, dx = 0 \]

• Exponential weight decay
  \[ \lim_{\|x\| \to \infty} \|x\|^d K(x) = 0 \]

• ???
  \[ \int_{\mathbb{R}^d} x x^T K(x) \, dx = cI \]
Kernel Density Estimation

Parzen Windows - Function Forms

\[ P(x) = \frac{1}{n} \sum_{i=1}^{n} K(x - x_i) \]

A function of some finite number of data points \( x_1 \ldots x_n \)

In practice one uses the forms:

\[ K(x) = c \prod_{i=1}^{d} k(x_i) \]

or

\[ K(x) = ck\left(\|x\|\right) \]

Same function on each dimension

Function of vector length only
Kernel Density Estimation
Various Kernels

\[ P(x) = \frac{1}{n} \sum_{i=1}^{n} K(x - x_i) \]

A function of some finite number of data points \( x_1 \ldots x_n \)

**Examples:**

- **Epanechnikov Kernel**
  \[ K_E(x) = \begin{cases} 
  c \left(1 - \|x\|^2\right) & \|x\| \leq 1 \\
  0 & \text{otherwise} 
  \end{cases} \]

- **Uniform Kernel**
  \[ K_U(x) = \begin{cases} 
  c & \|x\| \leq 1 \\
  0 & \text{otherwise} 
  \end{cases} \]

- **Normal Kernel**
  \[ K_N(x) = c \cdot \exp\left(-\frac{1}{2}\|x\|^2\right) \]
Kernel Density Estimation

\[ \nabla P(x) = \frac{1}{n} \sum_{i=1}^{n} \nabla K(x - x_i) \]

Give up estimating the PDF! Estimate **ONLY** the gradient.

Using the Kernel form:

\[ K(x - x_i) = ck \left( \frac{||x - x_i||^2}{h} \right) \]

We get:

\[ \nabla P(x) = \frac{c}{n} \sum_{i=1}^{n} \nabla k_i = \frac{c}{n} \left[ \sum_{i=1}^{n} g_i \right] - \frac{\sum_{i=1}^{n} x_i g_i}{\sum_{i=1}^{n} g_i} - x \]

\[ g(x) = -k'(x) \]
Kernel Density Estimation

Computing The Mean Shift

\[ \nabla P(x) = \frac{c}{n} \sum_{i=1}^{n} \nabla k_i = \frac{c}{n} \left[ \sum_{i=1}^{n} g_i \right] \begin{bmatrix} \sum_{i=1}^{n} x_i g_i \\ \sum_{i=1}^{n} g_i \end{bmatrix} \]

\[ g(x) = -k'(x) \]
Computing The Mean Shift

Yet another Kernel density estimation!

Simple Mean Shift procedure:
• Compute mean shift vector

\[ m(x) = \frac{\sum_{i=1}^{n} x_i g_i \left( \frac{\|x - x_i\|^2}{h} \right)}{\sum_{i=1}^{n} g_i \left( \frac{\|x - x_i\|^2}{h} \right)} - x \]

• Translate the Kernel window by \( m(x) \)

\[ g(x) = -k'(x) \]
Mean Shift Mode Detection

What happens if we reach a saddle point?

Perturb the mode position and check if we return back.

**Updated Mean Shift Procedure:**
- Find all modes using the Simple Mean Shift Procedure
- Prune modes by perturbing them (find saddle points and plateaus)
- Prune nearby – take highest mode in the window
Mean Shift Properties

- Automatic convergence speed – the mean shift vector size depends on the gradient itself.
- Near maxima, the steps are small and refined
- Convergence is guaranteed for infinitesimal steps only ➔ infinitely convergent, (therefore set a lower bound)
- For Uniform Kernel ( ), convergence is achieved in a finite number of steps
- Normal Kernel ( ) exhibits a smooth trajectory, but is slower than Uniform Kernel ( ).
Real Modality Analysis

Tessellate the space with windows

Run the procedure in parallel
Real Modality Analysis

The blue data points were traversed by the windows towards the mode.
Real Modality Analysis

An example

Window tracks signify the steepest ascent directions
Adaptive Mean Shift
Mean Shift Strengths & Weaknesses

**Strengths:**

- Application independent tool
- Suitable for real data analysis
- Does not assume any prior shape (e.g. elliptical) on data clusters
- Can handle arbitrary feature spaces
- Only ONE parameter to choose
- \( h \) (window size) has a physical meaning, unlike K-Means

**Weaknesses:**

- The window size (bandwidth selection) is not trivial
- Inappropriate window size can cause modes to be merged, or generate additional “shallow” modes
  - Use adaptive window size
Mean Shift Applications
Clustering

Cluster: All data points in the attraction basin of a mode

Attraction basin: the region for which all trajectories lead to the same mode
Clustering
Synthetic Examples

Simple Modal Structures

Complex Modal Structures
Clustering
Real Example

Feature space: L*u*v representation

Initial window enters

Pruning
Clustering
Real Example

L*u*v space representation
Clustering
Real Example

2D (L^*u) space representation

Final clusters

Not all trajectories in the attraction basin reach the same mode
Discontinuity Preserving Smoothing

Feature space: Joint domain = spatial coordinates + color space

\[ K(x) = C \cdot k_s \left( \frac{x_s}{h_s} \right) \cdot k_r \left( \frac{x_r}{h_r} \right) \]

Meaning: treat the image as data points in the spatial and gray level domain
Discontinuity Preserving Smoothing

The image gray levels... ... can be viewed as data points in the x, y, z space (joined spatial And color space)
Discontinuity Preserving Smoothing

Flat regions induce the modes!
Discontinuity Preserving Smoothing

The effect of window size in spatial and range spaces

Original

$(h_s, h_r) = (8, 8)$

$(h_s, h_r) = (8, 16)$

$(h_s, h_r) = (16, 4)$

$(h_s, h_r) = (16, 8)$

$(h_s, h_r) = (16, 16)$

$(h_s, h_r) = (32, 4)$

$(h_s, h_r) = (32, 8)$

$(h_s, h_r) = (32, 16)$
Discontinuity Preserving Smoothing

Example
Discontinuity Preserving Smoothing

Example
Segmentation

Segment = Cluster, or Cluster of Clusters

Algorithm:
• Run Filtering (*discontinuity preserving smoothing*)
• Cluster the clusters which are closer than window size

Mean Shift: A robust Approach Toward Feature Space Analysis, by Comaniciu, Meer
http://www.caip.rutgers.edu/~comanici
Segmentation

Example

...when feature space is only gray levels...
Segmentation
Example
Segmentation
Example
Segmentation
Example
Segmentation
Example
Segmentation
Example
Segmentation
Example
Non-Rigid Object Tracking
Non-Rigid Object Tracking

Real-Time

Surveillance

Driver Assistance

Object-Based Video Compression
Mean-Shift Object Tracking
General Framework: Target Representation

Choose a reference model in the current frame → Choose a feature space → Represent the model in the chosen feature space
Mean-Shift Object Tracking

General Framework: Target Localization

Start from the position of the model in the current frame

Search in the model’s neighborhood in next frame

Find best candidate by maximizing a similarity func.

Repeat the same process in the next pair of frames

... Current frame ...
Mean-Shift Object Tracking
Target Representation

Choose a reference target model

Choose a feature space

Represent the model by its PDF in the feature space

Quantized Color Space

Kernel Based Object Tracking, by Comaniciu, Ramesh, Meer
Mean-Shift Object Tracking
PDF Representation

Target Model
(centered at 0)

Target Candidate
(centered at \( y \))

\[
\tilde{q} = \{ q_u \}_{u=1}^{m} \quad \sum_{u=1}^{m} q_u = 1
\]

\[
\tilde{p}(y) = \{ p_u(y) \}_{u=1}^{m} \quad \sum_{u=1}^{m} p_u = 1
\]

Similarity Function:

\[
f(y) = f[\tilde{q}, \tilde{p}(y)]
\]
Mean-Shift Object Tracking
Smoothness of Similarity Function

Similarity Function: \( f(y) = f[\tilde{p}(y), \tilde{q}] \)

Problem:
- Target is represented by color info only
- Spatial info is lost
- Large similarity variations for adjacent locations
- \( f \) is not smooth
- Gradient-based optimizations are not robust

Solution:
- Mask the target with an isotropic kernel in the spatial domain
- \( f(y) \) becomes smooth in \( y \)
Mean-Shift Object Tracking

Finding the PDF of the target model

\[ \{x_i\}_{i=1}^{n} \] Target pixel locations

\( k(x) \)

A differentiable, isotropic, convex, monotonically decreasing kernel

- Peripheral pixels are affected by occlusion and background interference

\( b(x) \)

The color bin index (1..m) of pixel \( x \)

Probability of feature \( u \) in model

\[
q_u = C \sum_{b(x_i)=u} k\left(\|x_i\|^2\right)
\]

Normalization factor

Pixel weight

Probability of feature \( u \) in candidate

\[
p_u(y) = C_h \sum_{b(x_i)=u} k\left(\frac{\|y-x_i\|^2}{h}\right)
\]

Normalization factor

Pixel weight
Mean-Shift Object Tracking

Similarity Function

Target model: \( \vec{q} = (q_1, \ldots, q_m) \)

Target candidate: \( \vec{p}(y) = (p_1(y), \ldots, p_m(y)) \)

Similarity function: \( f(y) = f\left[ \vec{p}(y), \vec{q} \right] = ? \)

The Bhattacharyya Coefficient

\( \vec{q}' = (\sqrt{q_1}, \ldots, \sqrt{q_m}) \)

\( \vec{p}'(y) = (\sqrt{p_1(y)}, \ldots, \sqrt{p_m(y)}) \)

\( f(y) = \cos \theta_y = \frac{\vec{p}'(y)^T \vec{q}'}{||\vec{p}'(y)|| \cdot ||\vec{q}'||} = \sum_{u=1}^{m} \sqrt{p_u(y)q_u} \)
Mean-Shift Object Tracking
Target Localization Algorithm

Start from the position of the model in the current frame

Search in the model’s neighborhood in next frame

Find best candidate by maximizing a similarity func.$f\left[\vec{p}(y), \vec{q}\right]$
Mean-Shift Object Tracking
Approximating the Similarity Function

\[ f(y) = \sum_{u=1}^{m} \sqrt{p_u(y)} q_u \]

Model location: \( y_0 \)
Candidate location: \( y \)

Linear approx. (around \( y_0 \))

\[ f'(y) \approx \frac{1}{2} \sum_{u=1}^{m} \sqrt{p_u(y_0)} q_u \]

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

\[ p_u(y) = C_h \sum_{b(x_i) = u} k \left( \frac{||y - x_i||^2}{h} \right) \]

\[ \frac{C_h}{2} \sum_{i=1}^{n} w_i k \left( \frac{||y - x_i||^2}{h} \right) \]

Independent of \( y \)

Density estimate! (as a function of \( y \))
Mean-Shift Object Tracking
Maximizing the Similarity Function

The mode of
\[ \frac{C_h}{2} \sum_{i=1}^{n} w_i k \left( \frac{y - x_i}{h} \right)^2 \]
= sought maximum

Important Assumption:
The target representation provides sufficient discrimination

One mode in the searched neighborhood
Mean-Shift Object Tracking

Applying Mean-Shift

The mode of
\[ \frac{C_h}{2} \sum_{i=1}^{n} w_i k \left( \frac{\| y - x_i \|^2}{h} \right) \]

= sought maximum

Original Mean-Shift:

Find mode of
\[ c \sum_{i=1}^{n} k \left( \frac{\| y - x_i \|^2}{h} \right) \]

using
\[ y_1 = \frac{\sum_{i=1}^{n} x_i g \left( \frac{\| y_0 - x_i \|^2}{h} \right)}{\sum_{i=1}^{n} g \left( \frac{\| y_0 - x_i \|^2}{h} \right)} \]

Extended Mean-Shift:

Find mode of
\[ c \sum_{i=1}^{n} w_i k \left( \frac{\| y - x_i \|^2}{h} \right) \]

using
\[ y_1 = \frac{\sum_{i=1}^{n} x_i w_i g \left( \frac{\| y_0 - x_i \|^2}{h} \right)}{\sum_{i=1}^{n} w_i g \left( \frac{\| y_0 - x_i \|^2}{h} \right)} \]
Mean-Shift Object Tracking
About Kernels and Profiles

A special class of radially symmetric kernels:

$$K(x) = ck\left(\|x\|^2\right)$$

The profile of kernel $K$

$${k}'(x) = -g(x)$$

Extended Mean-Shift:
Find mode of $c \sum_{i=1}^{n} w_i k\left(\frac{\|y - x_i\|^2}{h}\right)$ using

$$y_1 = \frac{\sum_{i=1}^{n} x_i w_i g\left(\frac{\|y_0 - x_i\|^2}{h}\right)}{\sum_{i=1}^{n} w_i g\left(\frac{\|y_0 - x_i\|^2}{h}\right)}$$
Mean-Shift Object Tracking
Choosing the Kernel

A special class of radially symmetric kernels:

\[ K(x) = ck\left(\|x\|^2\right) \]

Epanechnikov kernel:

\[ k(x) = \begin{cases} 1 - x & \text{if } \|x\| \leq 1 \\ 0 & \text{otherwise} \end{cases} \]

Uniform kernel:

\[ g(x) = -k(x) = \begin{cases} 1 & \text{if } \|x\| \leq 1 \\ 0 & \text{otherwise} \end{cases} \]

\[
y_1 = \frac{\sum_{i=1}^{n} x_i w_i g\left(\frac{\|y_0 - x_i\|}{h}\right)}{\sum_{i=1}^{n} w_i g\left(\frac{\|y_0 - x_i\|}{h}\right)}
\]

\[
y_1 = \frac{\sum_{i=1}^{n} x_i w_i}{\sum_{i=1}^{n} w_i}
\]
Mean-Shift Object Tracking
Adaptive Scale

Problem:
- The scale of the target changes over time
- The scale \((h)\) of the kernel must be adapted

Solution:
- Run localization 3 times with different \(h\)
- Choose \(h\) that achieves maximum similarity
Mean-Shift Object Tracking

Results

Feature space: $16 \times 16 \times 16$ quantized RGB
Target: manually selected on 1st frame
Average mean-shift iterations: 4
Mean-Shift Object Tracking

Results

Partial occlusion

Distraction

Motion blur
Mean-Shift Object Tracking
Results
Mean-Shift Object Tracking

Results

Feature space: $128 \times 128$ quantized RG
Mean-Shift Object Tracking
The Scale Selection Problem

Problem:
- Kernel too big: Poor localization
- Kernel too small: Poor localization

In uniformly colored regions, similarity is invariant to $h$
Smaller $h$ may achieve better similarity
Nothing keeps $h$ from shrinking too small!
Tracking Through Scale Space
Motivation

Spatial localization for several scales

Simultaneous localization in space and scale

Previous method

This method

Mean-shift Blob Tracking through Scale Space, by R. Collins
Lindeberg’s Theory
Selecting the best scale for describing image features

- Scale-space representation
- Differential operator applied
- 50 strongest responses
Lindeberg’s Theory

The Laplacian operator for selecting blob-like features

\[
f(x) = \nabla^2 G(x; \sigma_1) \\
G(x; \sigma_1) \\
\text{LOG}(x; \sigma_1) \\
G(x; \sigma_2) \\
\text{LOG}(x; \sigma_2) \\
\vdots \\
G(x; \sigma_k) \\
\text{LOG}(x; \sigma_k)
\]

2D LOG filter with scale \( \sigma \)

3D scale-space representation

\[
\forall x \in f, \forall \sigma_{1..k} : \\
L(x, \sigma) = \text{LOG}(x; \sigma) \ast f(x)
\]

Best features are at \((x, \sigma)\) that maximize \(L\)
Lindeberg’s Theory
Multi-Scale Feature Selection Process

Original Image

Convolve

3D scale-space function

Maximize

250 strongest responses
(Large circle = large scale)

\[ L(x, \sigma) = \text{LOG}(x; \sigma) * f(x) \]
Tracking Through Scale Space
Approximating LOG using DOG

\[ \text{LOG}(x; \sigma) \approx \text{DOG}(x; \sigma) = G(x; \sigma) - G(x; 1.6\sigma) \]

Why DOG?

- Gaussian pyramids are created faster
- Gaussian can be used as a mean-shift kernel

3D spatial kernel

\[ K(x, \sigma) = \]

Scale-space filter bank

2D LOG filter with scale \( \sigma \)
2D DOG filter with scale \( \sigma \)
2D Gaussian with \( \mu=0 \) and scale \( \sigma \)
2D Gaussian with \( \mu=0 \) and scale 1.6\( \sigma \)
Tracking Through Scale Space
Using Lindeberg’s Theory

**Recall:**

- **Model:** \( \tilde{q}(q_1, \ldots, q_m) \) at \( y_0 \)
- **Candidate:** \( \tilde{p}(y) = (p_1(y), \ldots, p_m(y)) \)
- **Color bin:** \( b(x) \)
- **Pixel weight:** \( w(x) = \sqrt{\frac{q_{b(x)}}{p_{b(x)}(y_0)}} \)

**Centered at current location and scale**

**Modes are blobs in the scale-space neighborhood**

**Need a mean-shift procedure that finds local modes in** \( E(x, \sigma) \)

- 1D scale kernel (Epanechnikov)
- 3D spatial kernel (DOG)

**3D scale-space representation**
Tracking Through Scale Space

Example

Image of 3 blobs

A slice through the 3D scale-space representation
Tracking Through Scale Space
Applying Mean-Shift

Use interleaved spatial/scale mean-shift

Spatial stage:
- Fix $\sigma$ and look for the best $x$

Scale stage:
- Fix $x$ and look for the best $\sigma$

Iterate stages until convergence of $x$ and $\sigma$
Tracking Through Scale Space

Results

Fixed-scale

± 10% scale adaptation

Tracking through scale space
Thank You