Problem

- Given a target (object) in the first frame track it through all other frames.

Key Concepts

- Mean
- Mean shift
- Probability Density Functions (PDFs)
- Kernel Density
- Gradient of Kernel Density
- Bhattacharya coefficient

Mean Shift Vector

Given:
Data points and approximate location of the mean of this data.

Task:
Estimate the exact location of the mean of the data by determining the shift vector from initial mean.

Mean Shift Vector Example

\[ M_n(y) = \frac{1}{n} \sum_{i=1}^{n} (x_i - y) \]

- \( n \): number of points
- \( y \): initial mean location
- \( x_i \): data points

Mean shift vector always points towards the direction of the maximum increase in the density.
### Modified Mean Shift (weighted)

\[
M_n(y_0) = \frac{\sum_{i=1}^{n_i} w_i(y_0) x_i}{\sum_{i=1}^{n_i} w_i(y_0)} - y_0
\]

- \( n_i \): number of points in the kernel
- \( y_0 \): initial mean location
- \( x_i \): data points
- \( h \): kernel radius

Weights are determined using kernels (masks):
- Uniform, Gaussian or Epanechnikov

### Properties of Mean Shift

- Mean shift vector has the direction of the gradient of the density estimate.
- It is computed iteratively for obtaining the maximum density in the local neighborhood.

### Probability Density Functions (PDFs)

- Parametric models
  - Uniform
  - Gaussian
  - Exponential
  - ...
- Some PDFs can not be modeled by parametric models

### Kernel Density Estimation

- Kernel Density estimate can be used to represent any non-parametric pdf. (general)
- All data points are saved. (large storage)
- The probability of any given value is calculated by using all the data points.
Kernel Density Estimate

\[ \hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K \left( \frac{x - x_i}{h} \right) \]

- \( n \): number of points in the kernel
- \( h \): window radius
- \( x \): mean vector
- \( d \): number of dimensions
- \( K \): Kernel density function

Possible Kernels

- **Uniform kernel**
  \[ K_u(x) = \frac{1}{2\pi^d} \]

- **Normal kernel (convex, monotonic decreasing)**
  \[ K_n(x) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)} \]
  - \( \mu \): mean vector
  - \( \Sigma \): covariance matrix
  - \( d \): number of dimensions
  - \( |\Sigma| \): determinant of \( \Sigma \)

- **Epanechnikov kernel (convex, monotonic decreasing)**
  \[ K_e(x) = \begin{cases} \frac{1}{2} c_d i(d + 2)(1 - |x|^2) & \text{if } |x| < 1 \\ 0 & \text{otherwise} \end{cases} \]
  - \( c_d \): volume of unit d-dimensional sphere
  - \( d \): number of dimensions

Kernel Density Estimation

\[ \hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K \left( \frac{x - x_i}{h} \right) \]

Estimate of Density Gradient

\[ \nabla \hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^{n} \nabla K \left( \frac{x - x_i}{h} \right) \]

Mean Shift Vector in Terms of Epanechnikov Kernel

\[ \nabla \hat{f}(x) = \nabla \hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^{n} \nabla K \left( \frac{x - x_i}{h} \right) \]

- \( n \): number of points in unit d-dimensional sphere

Target Model for Tracking

- Features used for tracking include:
  - Gray level

- Feature probability distribution are calculated by using weighted histograms.
- The weights are derived from Epanechnikov kernel.
Target Model for Tracking

\[ p(u) = C \sum_{i=1}^{K} \left( \frac{1}{h} \right)^{K/2} \exp \left( -\frac{|\sum S(x_i) - u|^2}{2h^2} \right) \]

\( x_1, x_2, x_3, x_4 \) has the same value of the feature, such as gray level, \( u \).

Target Gray Level Feature

![Target Gray Level Feature](image)

Target Gray Level Feature

\[ \rho(y) = \rho[p(y), q] = \sum_{i=1}^{m} \lambda_i \phi_i(y) \]

\( \rho(y) \) : Bhattacharya coefficient.

Similarity of Target and Candidate Distributions

Target : \( q_u \)
Candidate : \( p_u \)

\[ d(y) = \sqrt{1 - \rho(y)} \]

\[ \rho(y) = \rho[p(y), q] = \sum_{i=1}^{m} \lambda_i \phi_i(y) \]

Distance Minimization

Minimizing the distance corresponds to maximizing Bhattacharya coefficient.

\[ \rho[p(y), q] = \sum_{i=1}^{m} \lambda_i \phi_i(y) \]

Taylor expansion around \( \hat{p}(y) \)

Maximizing Bhattacharya coefficient can be obtained by maximizing the blue term

\[ \rho[p(y), q] \approx \rho[p(y), q] + \frac{1}{2} \sum_{i=1}^{m} \lambda_i \phi_i(y) \]

\[ \rho(y) = q_u \]

\[ \rho(y) = \rho[p(y), q] = \sum_{i=1}^{m} \lambda_i \phi_i(y) \]

Likelihood Maximization

\[ \rho[p(y), q] \approx \rho[p(y), q] + \frac{1}{2} \sum_{i=1}^{m} \lambda_i \phi_i(y) \]

\[ \rho(y) = q_u \]

\[ \rho(y) = \rho[p(y), q] = \sum_{i=1}^{m} \lambda_i \phi_i(y) \]

\[ \rho(y) = \rho[p(y), q] = \sum_{i=1}^{m} \lambda_i \phi_i(y) \]

\[ \rho(y) = \rho[p(y), q] = \sum_{i=1}^{m} \lambda_i \phi_i(y) \]

Likelihood Maximization Using Mean Shift Vector

Maximization of the likelihood of target and candidate depends on the weights:

\[ w_i(y) = \sum_{k=1}^{m} \lambda_k \phi_k(y) \]

\[ 0 \leq w_i \leq 1 \]

\[ M_i(y) = \sum_{k=1}^{m} w_i(y) x_k \sum_{k=1}^{m} w_i(y) \]

Thus, new target center is

\[ \hat{y} = y + M_i(y) \]

\[ \lambda_i \phi_i(y) \]

\[ \rho(y) = \rho[p(y), q] = \sum_{i=1}^{m} \lambda_i \phi_i(y) \]

\[ \rho(y) = \rho[p(y), q] = \sum_{i=1}^{m} \lambda_i \phi_i(y) \]

\[ \rho(y) = \rho[p(y), q] = \sum_{i=1}^{m} \lambda_i \phi_i(y) \]

\[ \rho(y) = \rho[p(y), q] = \sum_{i=1}^{m} \lambda_i \phi_i(y) \]
Algorithm

1. Calculate \( q \)
2. Initialize estimated center \( y_0 \)
3. Calculate \( p \)
4. Calculate \( w \)
5. Estimate new target center \( y_1 \)
6. Update target center \( y_0 \rightarrow y_1 \)

Repeat until end of the sequence

Tracking Results

Papers


- Target-Tracking in Airborne Forward Looking Infrared Imagery