Tracking Algorithms

- Deterministic methods
  Given input video and current state, tracking result is always same.
  Local search
  - Least-square tracking
  - Mean-shift
  - Gradient ascent (or decent) algorithms
  - Correlation filter

- Probabilistic or stochastic methods
  Each trial may produce a different tracking result.
  Statistical modeling and search (e.g., by sampling)
  - Kalman filter/extended Kalman filter
  - Particle filter
  - Simple dense sampling

Probabilistic Tracking

- Target state is determined through a statistical process.
  - For example, the target state is estimated based on density function, sometimes by sampling for target state and corresponding observation.

Joint Probability and Graphical Model

- Graphical model
  - Probabilistic model for which a graph denotes the conditional independence structure between random variables
  - Enables a simpler representation of joint probability
  - Directed or undirected

\[
p(A, B, C, D) = p(D | A, B, C)p(C | A, B)p(B | A)p(A)
\]

\[
p(D | A, B) = p(C | A)p(B | A)p(A)
\]
Markov Property

- In stochastic process
  - The conditional probability distribution of future states of the process depends only upon the present state, not on the sequence of events that preceded it.
  - In the discrete time stochastic process, Markov property can be defined as
    \[ p(x_t|x_{t-1}, x_{t-2}, \ldots, x_0) = p(x_t|x_{t-1}) \]
- In graphical model

```
\[ x_0 \longrightarrow x_{t-2} \longrightarrow x_{t-1} \longrightarrow x_t \]
```

Sequential Bayesian Filtering

- Estimation of an unknown probability density function
  - PDF is estimated recursively over time using
    - Incoming measurement
    - Mathematical process model (dynamic model)
  - Extension of (static) Bayesian estimation
    - Observation changes overtime

```
x_t: state variable
z_t: observation variable

p(x_t|z_{1:t}) \propto p(z_t|x_t)p(x_t|z_{1:t-1})
```

Components of Sequential Bayesian Filter

- Process model (dynamic model)
  \[ x_{t|t-1} = f(x_{t-1}, a_{t-1}) + u_t \]
- Observation model
  \[ z_t = h(x_t, v_t) \]

Derivation of Sequential Bayesian Filtering

- Markov assumption for process model
  \[ p(x_t|x_{t-1}, x_{t-2}, \ldots, x_0) = p(x_t|x_{t-1}) \]
- Conditional independence in observation
  \[ p(z_t|x_t, x_{t-1}, \ldots, x_0) = p(z_t|x_t) \]
- Joint distribution of all variables
  \[ p(x_t, x_{t-1}, \ldots, x_0, z_t, \ldots, z_0) = p(x_0) \prod_{t=1}^{T} p(z_t|x_t)p(x_t|x_{t-1}) \]

We should estimate the marginal posterior \( p(x_t|z_{1:t}) \) sequentially.
Derivation of Sequential Bayesian Filtering

\[
\begin{align*}
\mathbb{P}(x_{t} | z_{1:t}) &= \frac{\mathbb{P}(z_{1:t} | x_{t}) \mathbb{P}(x_{t})}{\mathbb{P}(z_{1:t})} \\
&= \frac{\mathbb{P}(z_{t}, z_{1:t-1} | x_{t}) \mathbb{P}(x_{t})}{\mathbb{P}(z_{1:t})} \\
&= \frac{\mathbb{P}(z_{t} | z_{1:t-1}, x_{t}) \mathbb{P}(z_{1:t-1} | x_{t}) \mathbb{P}(x_{t})}{\mathbb{P}(z_{1:t})} \\
&= \frac{\mathbb{P}(z_{t} | x_{t}) \mathbb{P}(x_{t} | z_{1:t-1}) \mathbb{P}(z_{1:t-1}) \mathbb{P}(x_{t})}{\mathbb{P}(z_{1:t})} \\
&= \frac{\mathbb{P}(z_{t} | x_{t}) \mathbb{P}(x_{t} | z_{1:t-1})}{\mathbb{P}(z_{1:t})}
\end{align*}
\]

normalization constant

Examples of Sequential Bayesian Filtering

\[
p(x_{t} | z_{1:t}) \propto \mathbb{P}(x_{t} | x_{t}) \int p(x_{t} | x_{t-1}) p(x_{t-1} | z_{1:t-1}) dx_{t-1}
\]

- Kalman filter
- Extended Kalman filter
- Unscented Kalman filter
- Particle filter
  - Sequential Importance Sampling (SIS)
  - Sampling Importance Resampling (SIR)
  - Regularized particle filter
  - Auxiliary particle filter
  - ...

Monte Carlo Approximation

- Basic idea
  - Sample based
  - The more we draw samples, the more accurate the estimation is.
  - Useful when analytical solution is unknown or hard to compute
- MC in sequential Bayesian filtering
  - Estimating unknown probability distribution using a set of samples
  - MC can easily be used to compute marginal posterior distribution, \( p(x_{t} | z_{1:t}) \)
  - It does not require any assumption on the underlying distribution.

Particle Filter

- Most flexible implementation of sequential Bayesian filtering
  - Representation of the state with \((\text{location}, \text{weight})\) pair of each sample
  - No restriction of posterior density representation
  - No restriction of process and measurement models
  - Also known as Sequential Monte Carlo (SMC)

The posterior is estimated in a sequential manner by
the population of samples.

Actually, there are some other SMC techniques other than PF.

- Advantages
  - Able to handle multi-modal (arbitrary) posterior density functions
  - Able to handle non-linear process model
  - Very simple implementation

**Posterior Representation**

- **Probability State Weighted Sample**
  - $x_t^{(i)}, \omega_t^{(i)}$

**Procedure**

- **Sequential Importance Sampling (SIS)**

- **State Transition**

- **Observation**

- **Unknown Measurement Function**

**Limitations**

- **Degeneracy Problem**
  - Most particles have very small and negligible weights.
  - Most of the weights are concentrated on a few particles.
  - Most of particles are useless.
  - Density estimation becomes inaccurate.

**Condensation Algorithm**

- **Giving more diversity in samples by resampling (SIR)**

- **State Transition**

- **Observation**

- **Unknown Measurement Function**
Resampling

- Benefits
  - Identical sample weights
  - More sample diversity
  - Less degeneracy

However, resampling does not solve the degeneracy problem completely.


Particle Filter for Visual Tracking

- Sample a set of particles (samples) from the prior.
- Perform an observation for each particle.
- Obtain the target state

Observation: Color Histogram

- Likelihood computation for each particle
  - By histogram comparison

\[
p(y_t | x_t) \propto \exp \left( -\lambda D^2(q^*, q_t(x_t)) \right)
\]

where

\[
D^2(q^*, q_t(x_t)) = \left(1 - \frac{1}{n} \sum_{n}^{q^*(n)} q_t(n; x_t) \right)
\]

[17] Isard and Blake, 1998

[18] Particle Filter for Visual Tracking

Procedure 1 Particle filter iteration for single object color-based tracking

Input: \( q^* = \{q^*(n)\}_{n=1}^{N} \) reference color histogram

- Current particle set: \( \{x_t^n\}_{n=1}^{M} \)
- Prediction: for \( m = 1 \cdots M \), draw \( x_{t+1}^m \) from second-order AR dynamics.
- Computation of candidate histograms: for \( m = 1 \cdots M \), compute \( q_{t+1}(\hat{x}_{t+1}^m) \) according to (4).
- Weighting: for \( m = 1 \cdots M \) compute

\[
\pi_{t+1}^m = K \exp \sum_{n=1}^{N} \lambda \sqrt{q^*(n)q_{t+1}(n; x_{t+1}^m)}
\]

with \( K \) such that \( \sum_{m=1}^{M} \pi_{t+1}^m = 1 \)

- Selection: for \( m = 1 \cdots M \), sample index \( a(m) \) from discrete probability \( \{a_{t+1}^m\} \) over \( \{1 \cdots M\} \), and set \( x_{t+1}^m = \hat{x}_{t+1}^{a(m)} \).


[20]
Characteristics

- It is practically (almost) impossible to find the proper dynamic model.
  - A suggestion: Auto-Regressive (AR) model
    \[ x_{t+1} = Ax_t + Bx_{t-1} + Cv_t, \quad v_t \sim N(0, \Sigma) \]
  - Random walk is frequently used.
- Tracking control and observation are independent.
  - Any reasonable observation technique can be integrated into the observation model (e.g., histogram comparison)
    \[ p(y_t \mid x_t) \propto \exp\left(-\lambda D^2(q^*, q_t(x_t))\right) \]
    where \( D^2(q^*, q_t(x_t)) = \left(1 - \sum_n \sqrt{q^*(n)q_t(n; x_t)}\right) \)

Sparse Representation

- Main idea
  - Reconstructs an input sample with a sparse linear combination of templates
    \[
    \begin{align*}
    l(x_t) &= a_1 t_1 + a_2 t_2 + \cdots + a_n t_n \\
    &\quad + e_1^1 i_1 + e_2^1 i_2 + \cdots + e_d^1 i_d + e_1^2 (-i_1) + e_2^2 (-i_2) + \cdots + e_d^2 (-i_d) \\
    \end{align*}
    \]
    \[
    l(x_t) = [T I - I] \begin{bmatrix} a \\ e^+ & e^- \end{bmatrix} \equiv Bc \quad \text{where} \quad B = [T I - I] \quad \text{and} \quad c = \begin{bmatrix} a \\ e^+ & e^- \end{bmatrix}
    \]

Tracking Results

- Optimization
  - Based on an interior-point method: very slow
  - Can be accelerated by various techniques including compressed sensing
Tracking-by-Detection

- Combination of tracking and detection techniques
  - Exploits the recent advance of object detection techniques
  - Typically needs to design online classifiers
  - Requires to handling outliers and noises effectively
- Examples
  - Online boosting
  - STRUCK: structural SVM
  - Deep learning: convolutional neural networks
  - etc.

Applications of Particle filter

- Computer vision
  - Visual tracking
  - Dynamic parameter estimation
- Robotics
  - SLAM: Simultaneous Localization And Mapping
- Signal processing
- Financial engineering
  - Prediction of stock price