# Particle Filtering

CS6240 Multimedia Analysis

Leow Wee Kheng

Department of Computer Science
School of Computing
National University of Singapore

# Introduction

Video contains motion information that can be used for

- detecting the presence of moving objects
- tracking and analyzing the motion of the objects
- tracking and analyzing the motion of camera

Basic tracking methods:

- Gradient-based Image Flow:
  - Track points based on intensity gradient.
  - Example: Lucas-Kanade method [LK81, TK91].
- Feature-based Image Flow:
  - Track points based on template matching of features at points.
- Mean Shift Tracking:
  - Track image patches based on feature distributions, e.g., color histograms [CRM00].

## Strengths and Weaknesses

- Image flow approach:
  - Very general and easy to use.
  - If track correctly, can obtain precise trajectory with sub-pixel accuracy.
  - Easily confused by points with similar features.
  - Cannot handle occlusion.
  - Cannot differentiate between planner motion and motion in depth.
  - Demo: lk-elephant.mpg.
- Mean shift tracking:
  - Very general and easy to use.
  - Can track objects that change size & orientation.
  - Can handle occlusion, size change.
  - Track trajectory not as precise.
  - Can't track object boundaries accurately.
  - Demo: ms-football1.avi, ms-football2.avi.

Basic methods can be easily confused in complex situations:

frame 1

frame 2

- In frame 1, which hand is going which way?
- Which hand in frame 1 corresponds to which hand in frame 2?

Notes:

- The chances of making wrong association is reduced if we can correctly predict where the objects will be in frame 2.
- To predict ahead of time, need to estimate the velocities and the positions of the objects in frame 1.

To overcome these problems, need more sophisticated tracking algorithms:

- Kalman filtering: for linear dynamic systems, unimodal probability distributions
- Extended Kalman filtering: for nonlinear dynamic systems, unimodal probability distributions
- Condensation algorithm: for multi-modal probability distributions

# CONDENSATION

Conditional Density Propagation over time [IB96, IB98].
Also called particle filtering.

Main differences with Kalman filter:

1. Kalman filter:
    - Assumes uni-modal (Gaussian) distribution.
    - Predicts single new state for each object tracked.
    - Updates state based on error between predicted state and observed data.
2. CONDENSATION algorithm:
    - Can work for multi-modal distribution.
    - Predicts multiple possible states for each object tracked.
    - Each possible state has a different probability.
    - Estimates probabilities of predicted states based on observed data.

# Probability Density Functions

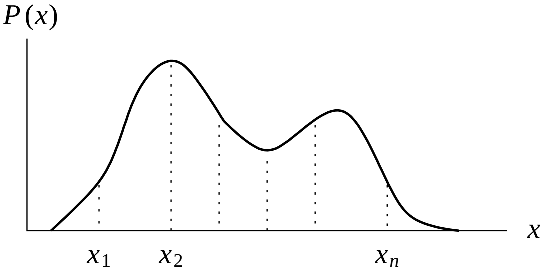
Two basic representations of probability density functions $P(x)$:

1. Explicit
   - Represent $P(x)$ by an explicit formula, e.g., Gaussian

$$P(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right) \quad (1)$$

   - Given any $x$, can compute $P(x)$ using the formula.

2. Implicit
   - Represent $P(x)$ by a set of samples $x_1, x_2, \ldots, x_n$ and their estimated probabilities $P(x_i)$.
   - Given any $x' \neq x_i$, cannot compute $P(x')$ because there is no explicit formula.

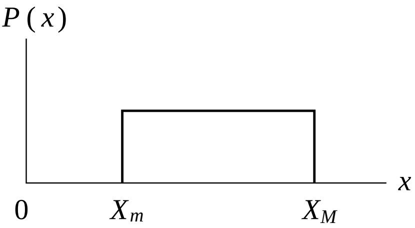



CONDENSATION algorithm predicts multiple possible next states.

- Achieved using sampling or drawing samples from the probability density functions.
- High probability samples should be drawn more frequently.
- Low probability samples should be drawn less frequently.

# Sampling from Uniform Distribution

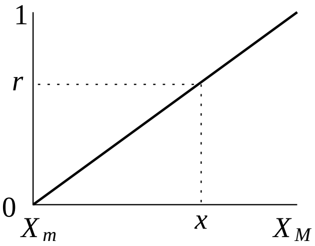
Uniform Distribution:




- Equal probability between $X_m$ and $X_M$:

$$P(x) = \begin{cases} \dfrac{1}{X_M - X_m} & \text{if } X_m \leq x \leq X_M \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Sampling Algorithm:




1. Generate a random number $r$ from [0, 1] (uniform distribution).
2. Map $r$ to $x$:

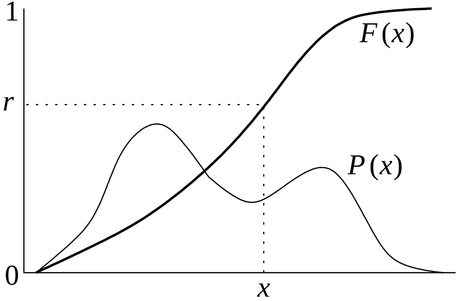
$$x = X_m + r(X_M - X_m) \tag{3}$$

The samples $x$ drawn will have uniform distribution.

# Sampling from Non-uniform Distribution

Let $P(x)$ denote the probability density function.
$F(x)$ is the indefinite integral of $P(x)$:

$$F(x) = \int_0^x P(x)dx \tag{4}$$



Sampling Algorithm:

1. Generate a random number $r$ from $[0, 1]$ (uniform distribution).
2. Map $r$ to $x$:
    - Find the $x$ such that $F(x) = r$, i.e., $x = F^{-1}(r)$.
    - That is, find the $x$ such that the area under $P(x)$ to the left of $x$ equals $r$.

The samples $x$ drawn will fit the probability distribution.

# Sampling from Implicit Distribution

The method is useful when

- it is difficult to compute $F^{-1}(r)$, or
- the probability density is implicit.

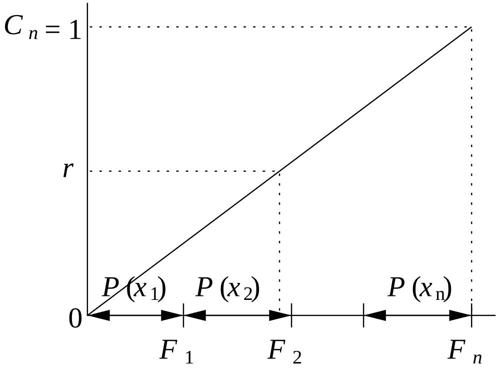
The basic idea is similar to the previous method:

- Given $x_i$ and $P(x_i)$, $i = 1, \ldots, n$.
- Compute cumulative probability $F(x_i)$:

$$F(x_i) = \sum_{j=1}^{i} P(x_j) \tag{5}$$

- Compute normalized weight $C(x_i)$:

$$C(x_i) = F(x_i)/F(x_n) \tag{6}$$




Sampling Algorithm:

1. Generate a random number $r$ from $[0, 1]$ (uniform distribution).
2. Map $r$ to $x_i$:
   - Find the smallest $i$ such that $C_i \geq r$.
   - Return $x_i$.

Samples $x = x_i$ drawn will follow probability density.

- The larger the $n$, the better the approximation.

# Factored Sampling

- $x$: object model (e.g., a curve)
- $z$: observed or measured data in image
- $P(x)$: a priori (or prior) probability density of $x$ occurring.
- $P(z|x)$: likelihood that object $x$ gives rise to data $z$.
- $P(x|z)$: a posteriori (or posterior) probability density that the object is actually $x$ given that $z$ is observed in the image.
  So, want to estimate $P(x|z)$.

From Bayes' rule:

$$P(x|z) = k\,P(z|x)\,P(x) \tag{7}$$

where $k = P(z)$ is a normalizing term that does not depend on $x$.

Notes:

- In general, $P(z|x)$ is multi-modal.
- Cannot compute $P(x|z)$ using closed form equation.
  Has to use iterative sampling technique.
- Basic method: factored sampling [GCK91]. Useful when
  - $P(z|x)$ can be evaluated point-wise but sampling it is not feasible, and
  - $P(x)$ can be sampled but not evaluated.

Factored Sampling Algorithm [GCK91]:

1. Generate a set of samples $\{s_1, s_2, \ldots, s_n\}$ from $P(x)$.
2. Choose an index $i \in \{1, \ldots, n\}$ with probability $\pi_i$:

$$\pi_i = \frac{P(z|x = s_i)}{\sum_{j=1}^{n} P(z|x = s_j)}. \tag{8}$$

3. Return $x_i$.

The samples $x = x_i$ drawn will have a distribution that approximates $P(x|z)$.

- The larger the $n$, the better the approximation.
- So, no need to explicitly compute $P(x|z)$.

# CONDENSATION Algorithm

## Object Dynamics

- state of object model at time $t$: $\mathbf{x}(t)$
- history of object model: $\mathbf{X}(t) = (\mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(t))$
- set of image features at time $t$: $\mathbf{z}(t)$
- history of features: $\mathbf{Z}(t) = (\mathbf{z}(1), \mathbf{z}(2), \ldots, \mathbf{z}(t))$

General assumption: object dynamic is a Markov process:

$$P(\mathbf{x}(t+1) \,|\, \mathbf{X}(t)) = P(\mathbf{x}(t+1) \,|\, \mathbf{x}(t)) \tag{9}$$

i.e., new state depends only on immediately preceding state.

- $P(\mathbf{x}(t+1) \,|\, \mathbf{X}(t))$ governs probability of state change.

## Measurements

Measurements $\mathbf{z}(t)$ are assumed to be mutually independent, and also independent of object dynamics. So,

$$P(\mathbf{Z}(t) \,|\, \mathbf{X}(t)) = \prod_{i=1}^{t} P(\mathbf{z}(i) \,|\, \mathbf{x}(i)) \,. \tag{10}$$

# CONDENSATION Algorithm

Iterate:

At time $t$, construct $n$ samples $\{\mathbf{s}_i(t), \pi_i(t), c_i(t), i = 1, \ldots, n\}$ as follows:

The $i$th sample is constructed as follows:

1. Select a sample $\mathbf{s}$ from the probability distribution $\{\mathbf{s}_i(t-1), \pi_i(t-1), c_i(t-1)\}$.

2. Predict by selecting a sample $\mathbf{s}'$ from the probability distribution

$$P(\mathbf{x}(t) \,|\, \mathbf{x}(t-1) = \mathbf{s})$$

   Then, $\mathbf{s}_i(t) = \mathbf{s}'$.

3. Measure $\mathbf{z}(t)$ from image and weight new sample:

$$\pi_i(t) = P(\mathbf{z}(t) \,|\, \mathbf{x}(t) = \mathbf{s}_i(t))$$

- normalize $\pi_i(t)$ so that $\sum_i \pi_i(t) = 1$
- compute cumulative probability $c_i(t)$:

$$\begin{aligned} c_0(t) &= 0 \\ c_i(t) &= c_{i-1}(t) + \pi_i(t) \end{aligned}$$

# Example

Track curves in input video [IB96].

Let

- $\mathbf{x}$ denote the parameters of a linear transformation of a B-spline curve, either affine deformation or some non-rigid motion,
- $\mathbf{p}_s$ denote points on the curve.

Notes:

- Instead of modeling the curve, model the transformation of curve.
- Curve can change shape drastically over time.
- But, changes of transformation parameters are smaller.

## Model Dynamics

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\boldsymbol{\omega}(t) \tag{11}$$

- $\mathbf{A}$: state transition matrix
- $\boldsymbol{\omega}$: random noise
- $\mathbf{B}$: scaling matrix

Then, $P(\mathbf{x}(t+1) \,|\, \mathbf{x}(t))$ is given by

$$P(\mathbf{x}(t+1) \,|\, \mathbf{x}(t)) = \exp\left\{-\frac{1}{2}\|\mathbf{B}^{-1}[\mathbf{x}(t+1) - \mathbf{A}\mathbf{x}(t)]\|^2\right\}. \tag{12}$$

$P(\mathbf{x}(t+1) \,|\, \mathbf{x}(t))$ is a Gaussian.

## Measurement

- $P(\mathbf{z}(t) \,|\, \mathbf{x}(t))$ is assumed to remain unchanged over time.
- $\mathbf{z}_s$ is nearest edge to point $\mathbf{p}_s$ on model curve, within a small neighborhood $\delta$ of $\mathbf{p}_s$.
- To allow for missing edge and noise, measurement density is modeled as a robust statistics, a truncated Gaussian:

$$P(\mathbf{z}|\mathbf{x}) = \exp\left\{ -\frac{1}{2\sigma^2} \sum_s \phi_s \right\} \tag{13}$$

where

$$\phi_s = \begin{cases} \|\mathbf{p}_s - \mathbf{z}_s\|^2 & \text{if } \|\mathbf{p}_s - \mathbf{z}_s\| < \delta \\ \rho & \text{otherwise.} \end{cases} \tag{14}$$

$\rho$ is a constant penalty.

Now, can apply CONDENSATION algorithm to track the curve.

Further Readings:

1. [IB96, IB98]: Other application examples of CONDENSATION algorithm.

# Reference I


R. G. Brown and P. Y. C. Hwang.
*Introduction to Random Signals and Applied Kalman Filtering.*
John Wiley & Sons, 3rd edition, 1997.

D. Comaniciu, V. Ramesh, and P. Meer.
Real-time tracking of non-rigid objects using mean shift.
In *IEEE Proc. on Computer Vision and Pattern Recognition*, pages 673–678, 2000.

U. Grenander, Y. Chow, and D. M. Keenan.
*HANDS. A Pattern Theoretical Study of Biological Shapes.*
Springer-Verlag, 1991.

# Reference II


- M. Isard and A. Blake.
  Contour tracking by stochastic propagation of conditional density.
  In *Proc. European Conf. on Computer Vision*, volume 1, pages 343–356, 1996.

- M. Isard and A. Blake.
  CONDENSATION — conditional density propagation for visual tracking.
  *Int. J. Computer Vision*, 29(1):5–28, 1998.

# Reference III


- B. D. Lucas and T. Kanade.
  An iterative image registration technique with an application to stereo vision.
  In *Proceedings of 7th International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.
  http://www.ri.cmu.edu/people/person_136_pubs.html.

- C. Tomasi and T. Kanade.
  Detection and tracking of point features.
  Technical Report CMU-CS-91-132, School of Computer Science, Carnegie Mellon University, 1991.
  http://citeseer.nj.nec.com/tomasi91detection.html,
  http://www.ri.cmu.edu/people/person_136_pubs.html.