

Multiple Objects Tracking in videos by a Graph-based Particle Filter

Donatello Conte

Laboratoire d'Informatique - Tours

October 1, 2015



Plan

- 1 Introduction
- 2 Particle Filters
 - Particle Filter in Objects Tracking Problem
- 3 Graph-based Particle Filters
- 4 Conclusions

Plan

- 1 Introduction
- 2 Particle Filters
- 3 Graph-based Particle Filters
- 4 Conclusions

Object Tracking

The goal of object tracking is extracting trajectories of moving objects

State of the arts

- Large literature: many methods, but still an open problem
- Some exemples:
 - KSP: Multiple Object Tracker Using K-Shortest Paths
 - Continuous Energy Minimization for Multi-Target Tracking
 - MonteCarlo methods and Particle Filters

Plan

- 1 Introduction
- 2 Particle Filters
 - Particle Filter in Objects Tracking Problem
- 3 Graph-based Particle Filters
- 4 Conclusions

A filtering system

$$\mathbf{x}_t = f_t(\mathbf{x}_{t-1}, \mathbf{v}_t)$$

$$\mathbf{y}_t = h_t(\mathbf{x}_t, \mathbf{w}_t)$$

where $\mathbf{x}_t$: the hidden state at time t ;

$\mathbf{y}_t$: the measurement state;

f_t : the temporal evolution of $\mathbf{x}_t$;

h_t : the measurement equation;

$\mathbf{v}_t$ and $\mathbf{w}_t$: independent white noises.

Goal

Estimating the posterior density function

$$p(\mathbf{x}_t | \mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t | \mathbf{x}_t) \cdot p(\mathbf{x}_t | \mathbf{y}_{1:t-1})}{\int_{\mathcal{X}} p(\mathbf{y}_t | \mathbf{x}'_t) \cdot p(\mathbf{x}'_t | \mathbf{y}_{1:t-1}) d\mathbf{x}'_t}$$

The solution: particles filters

- Approximating the density function by a weighted sum of N Dirac masses $\delta_{\mathbf{x}_t^{(n)}}(d\mathbf{x}_t)$ centered on hypothetic state realizations $\{\mathbf{x}_t^{(n)}\}_{n=1}^N$ of the state $\mathbf{x}_t$, also called particles;
- $\mathbb{P}(\mathbf{x}_t|\mathbf{y}_{1:t})$ is recursively approximated by the empiric distribution $P_N(\mathbf{x}_t|\mathbf{y}_{1:t}) = \sum_{n=1}^N w_t^{(n)} \delta_{\mathbf{x}_t^{(n)}}$, where $\mathbf{x}_t^{(n)}$ is the n^{th} particle and $w_t^{(n)}$ its weight.

Particle Filter Processing

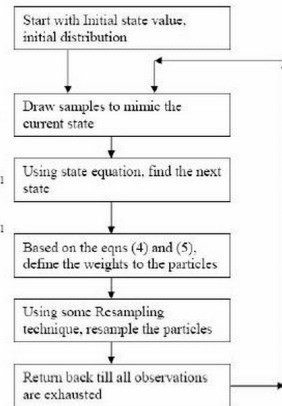
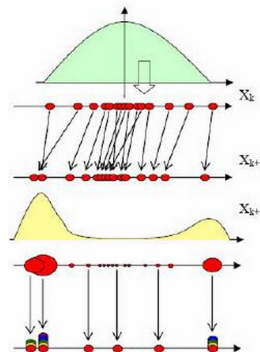
❶ **Diffusion step:** propagating the particle swarm $\{\mathbf{x}_t^{(n)}, w_t^{(n)}\}_{n=1}^N$ using a probabilistic model for the state function;

❷ **Update step:** computes new particle weights using the new observation $\mathbf{y}_t$;

❸ **Resampling step:** generating new particles according to their weight.

❹ **Output:**

- $\hat{\mathbf{x}}_t^{MMSE} = \sum_{n=1}^N w_t^{(n)} \mathbf{x}_t^{(n)}$
- $\hat{\mathbf{x}}_t^{MAP} = \operatorname{argmax}_{\mathbf{x}_t} \sum_{n=1}^N w_t^{(n)} \delta_{\mathbf{x}_t^{(n)}}$



Object Tracking and Particle Filters

- Widely used in the mono-object tracking, because of its reliability to deal with non-linear systems
 - usually the state of the system is the coordinates of the object to track

Object Tracking and Particle Filters

- Problems in multi-objects tracking
 - estimating more objects needs substantially more particles (*curse of dimensionality*: the number of particles increases exponentially with the number of objects)
 - the association problem between measures and objects has to be solved
 - interactions between objects should be modeled
- The problem is still open

Plan

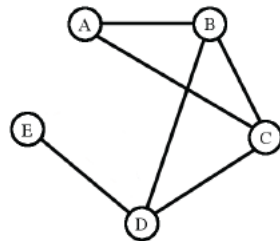
- 1 Introduction
- 2 Particle Filters
- 3 Graph-based Particle Filters**
- 4 Conclusions

Graph-based Particles Filter

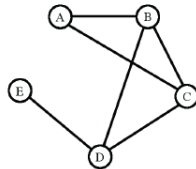
A new idea

- Using graphs for representing particles
- One graph $\equiv$ One particle $\equiv$ The entire ensemble of objects in the scene
- Using graph kernels for measuring likelihood

The Representation

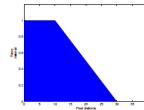
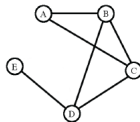


The Representation



Node attributes											
ID	label	visibility	w	h	c_x	c_y	$\dot{w}(\frac{px}{fr})$	$\dot{h}(\frac{px}{fr})$	$\dot{c}_x(\frac{px}{fr})$	$\dot{c}_y(\frac{px}{fr})$	$h_o(A)$
A	1	1	41	104	94	207	3	0	15	0	
B	2	1	57	150	151	220	4	1	12	4	
C	3	1	94	197	235	233	4	2	8	8	
D	4	1	40	81	326	212	1	1	0	10	
E	5	1	105	253	505	308	0	0	0	0	

The Representation



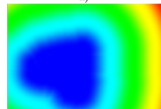
a)



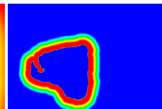
b)



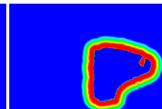
c)



d)



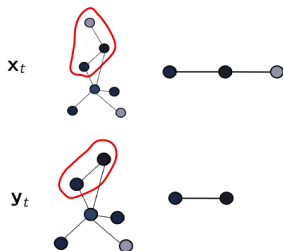
e)



f)

<i>Edge attributes</i>		
ID	fuzzy adjacency measure adj (with $\tau = 0.1$)	
e(A,B)	0.62	$adj(near(R_1), R_2) = \frac{\sum_{x \in R_2} \mu_{near(R_1)}(x)}{\sum_{x \in S} \mu_{R_2}(x)}$
e(A,C)	0.43	
e(B,C)	0.58	
e(B,D)	0.32	
e(C,D)	0.27	
e(D,E)	0.12	

The likelihood: an Edition Path Kernel



$$p(\mathbf{y}_t | \mathbf{x}_t) \simeq K_{edit}(\mathbf{x}_t, \mathbf{y}_t)$$

Graph Kernel based on editions costs



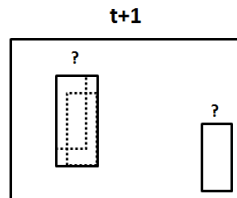
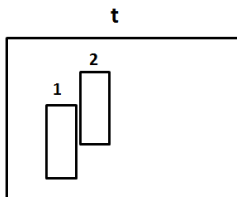
Edition paths: $(h, \kappa(h), \dots, \kappa^D(h))$

$$K_{edit}(h_1, h_2) = \frac{1}{D^2} \sum_{k=0}^D \sum_{l=0}^D e^{-\frac{cost_k(h_1) + cost_l(h_2)}{2\sigma_{cost}^2}} K_{classic}(\kappa^k(h_1), \kappa^l(h_2))$$

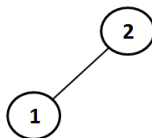
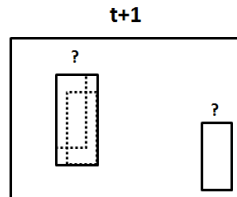
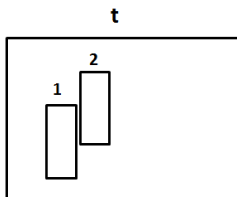
Graph-based Particle Filter

- **Diffusion step:** propagating the particle swarm $\left\{\mathbf{x}_t^{(n)}, w_t^{(n)}\right\}_{n=1}^N$ using a probability model for the hidden state;
 - **Updating step:** computing new particle weights using the new observation $\mathbf{y}_t$;
 - **Resampling step:** generating new particles based on old particles weights.
 - **Output:**
 - $\hat{\mathbf{x}}_t^{MMSE} = \sum_{n=1}^N w_t^{(n)} \mathbf{x}_t^{(n)}$
 - $\hat{\mathbf{x}}_t^{MAP} = \operatorname{argmax}_{\mathbf{x}_t} \sum_{n=1}^N w_t^{(n)} \delta_{\mathbf{x}_t^{(n)}}$
- **Diffusion step:** Graph Edition (adding a node, changing an attribute, ...) according to a probabilistic model;
 - **Updating step:** at time t the observation of the scene is also represented by a graph $\Rightarrow$ the weights of “particle graphs” are computed by means of Graph Kernels;
 - **Output:** MAP: the graph with the highest weight is taken as state representation at time t

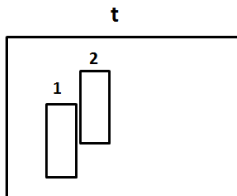
A toy example



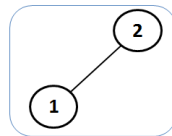
A toy example



A toy example



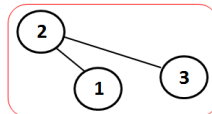
$p = 0.2$



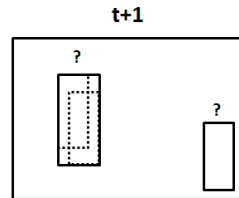
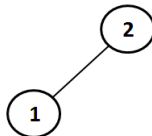
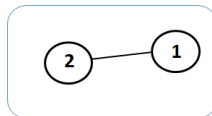
$p = 0.1$



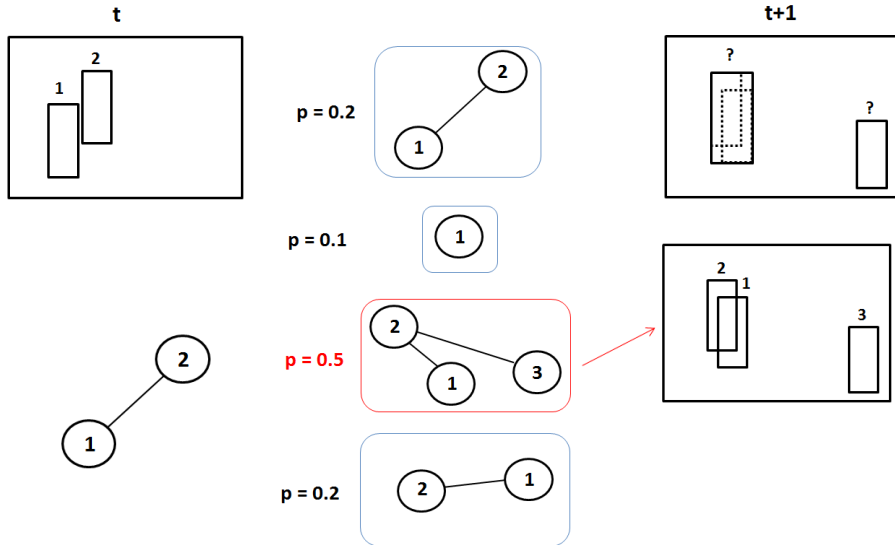
$p = 0.5$



$p = 0.2$



A toy example



Preliminary results (work in progress)

- Public dataset : **PETS2009**
- Common indexes : **Kasturi et al. [PAMI2009]**

ATA	MOTA	MOTP
0.10	0.57	0.61
0.31	0.69	0.75

Plan

- 1 Introduction
- 2 Particle Filters
- 3 Graph-based Particle Filters
- 4 Conclusions**

Conclusions

- A new idea for facing Particles Filters open problems
- Graph Representation of the scene
- Likelihood in terms of graphs
- Future work: assessing and improving the method