Figure 1: A Markov chain for object tracking.

Figure 2: A Dynamic Bayesian Network.

2 ARTICULATED OBJECT TRACKING

In this paper, articulated object tracking consists of estimating a state sequence \( \{x_t\}_{t=1}^{T} \), whose evolution is given by equation \( x_t = f_t(x_{t-1}, n^x_t) \), from observations \( \{y_t\}_{t=1}^{T} \) related to the states by \( y_t = h_t(x_t, n^y_t) \). Usually, \( f_t \) and \( h_t \) are nonlinear functions, and \( n^x_t \) and \( n^y_t \) are i.i.d. noise sequences. From a probabilistic viewpoint, this problem can be represented by the Markov chain of Fig. 1 and it amounts to estimate, for any \( t \), \( p(x_{1:t}|y_{1:t}) \) where \( x_{1:t} \) denotes the tuple \( (x_1, \ldots, x_t) \). This can be computed iteratively using Eq. (1) and (2), which are referred to as a prediction step and a correction step respectively.

\[
p(x_{1:t}|y_{1:t-1}) = p(x_t|x_{t-1})p(x_{1:t-1}|y_{1:t-1}) \quad (1)
p(x_{1:t}|y_{1:t}) \propto p(y_t|x_t)p(x_{1:t-1}|y_{1:t-1}) \quad (2)
\]

with \( p(x_t|x_{t-1}) \) the transition corresponding to \( f_t \) and \( p(y_t|x_t) \) the likelihood corresponding to \( h_t \).

The PF framework [Gordon et al., 1993] approximates the above densities using weighted samples \( \{x^{(i)}_t, w^{(i)}_t\}_{i=1}^{N} \), \( i = 1, \ldots, N \), where each \( x^{(i)}_t \) is a possible realization of state \( x_t \) called a particle. In its prediction step (Eq. (1)), PF propagates the particle set \( \{x^{(i)}_{t-1}, w^{(i)}_{t-1}\} \) using a proposal function \( q(x_t|x^{(i)}_{t-1}, y_t) \) which may differ from \( p(x_t|x^{(i)}_{t-1}) \) (but, for simplicity, we will assume they do not); in its correction step (2), PF weights the particles using a likelihood function, so that \( w^{(i)}_t \propto w^{(i)}_{t-1}p(y_t|x^{(i)}_{t-1}) \frac{p(x^{(i)}_{t-1}|x^{(i)}_{t-1})}{q(x^{(i)}_{t-1}|x^{(i)}_{t-1}, y_t)} \), with \( \sum_{i=1}^{N} w^{(i)}_t = 1 \). The particles can then be resampled: those with the highest weights are duplicated while the others are eliminated. The estimation of the posterior density \( p(x_t|y_{1:t}) \) is then given by \( \sum_{i=1}^{N} w^{(i)}_t \delta_{x^{(i)}_t} \), where \( \delta_{x^{(i)}_t} \) are Dirac masses centered on particles \( x^{(i)}_t \).

As shown in [MacCormick and Isard, 2000], the number of particles necessary for a good estimation of the above densities grows exponentially with the dimension of the state space, hence making PF’s basic scheme unusable in real-time for articulated object tracking. To cope with this problem, different variants of PF have been proposed, ranging from local search-based methods like the Annealed Particle Filter [Deutscher and Reid, 2005, Gall, 2005] and hierarchical-refining methods [Chang and Lin, 2010] to decomposition techniques like Partitioned Sampling (PS) [MacCormick and Blake, 1999] and its siblings [Rose et al., 2008, Besada-Portas et al., 2009]. Here, we focus on decomposition-based particle filters like PS.

PS’s key idea is that the state and observation spaces \( X \) and \( Y \) can often be naturally decomposed as \( X = X^1 \times \cdots \times X^P \) and \( Y = Y^1 \times \cdots \times Y^P \) where each \( X^j \) represents some “part” of the object. For instance, on Fig. 2, a human body is decomposed as 6 parts (head, torso, etc.) numbered from 1 to 6. The state of the \( j \)th part at time \( t \) is denoted \( x^j_t \). Then, by exploiting conditional independences among different subspaces \( (X^j, Y^j) \), PS estimates \( p(x_{1:t}|y_{1:t}) \) using only sequential applications of PF over \( (X^j, Y^j) \). For instance, on Fig. 2, given the position of the torso, the left and right arm positions may be independent so, after applying PF on the torso, PS can apply it sequentially to the left and right arms and still compute a correct estimation of \( p(x_{1:t}|y_{1:t}) \). As the \( (X^j, Y^j) \) subspaces are “smaller” than \( (X, Y) \), the distributions to estimate at each iteration of PF have fewer parameters than those defined on \( (X, Y) \), which significantly reduces the number of particles needed for their estimation and, thus, speeds up the computations.

The exploitation of the conditional independences among the \( (X^j, Y^j) \) leads to generalizing the Markov chain of Fig. 1 by the Dynamic Bayesian Network (DBN) of Fig. 2.