A Causal And-Or Graph Model for Visibility Fluent Reasoning in Tracking Interacting Objects

Yuanlu Xu\textsuperscript{1}, Lei Qin\textsuperscript{2}, Xiaobao Liu\textsuperscript{3}, Jianwen Xie\textsuperscript{4} and Song-Chun Zhu\textsuperscript{1}

\textsuperscript{1}University of California, Los Angeles (UCLA), \textsuperscript{2}Inst. Computing Technology, Chinese Academy of Sciences\textsuperscript{3}San Diego State University (SDSU), \textsuperscript{4}Hikvision Research Institute, USA

\section*{Motivation}

\textbf{a)} Tracking interacting objects is a very challenging problem. An object of interest might undergo frequent interactions with the surrounding, e.g., entering a vehicle or a building, or with the other objects, e.g., passing behind another subject.

\textbf{b)} The key idea of our method is to introduce a fluent variable for each subject of interest to explicitly indicate his/her visibility status in videos.

\section*{Method Overview}

\textbf{a)} We introduce a Causal And-Or Graph (C-AOG) model to represent object visibility fluents varying over time to represent the causal relationships between object’s activities (actions/sub-events) and object’s visibility fluent changes.

\textbf{b)} Our goal is to jointly find all subject locations and estimate their visibility fluents from the input video sequence.

\section*{Problem Formulation}

\textbf{a)} According to Bayes’ rule, we can solve our joint object tracking and fluent reasoning problem by maximizing a posterior (MAP),

\textbf{Prior term:} a temporal model to measure object motion consistency and action-fluent transition, as illustrated in (a)

\textbf{Likelihood term:} appearance model to measure object-object interactions, as illustrated in (b) and (c)

\begin{itemize}
  \item[(1)] We represent the prior term as a Markov Chain model (1). The likelihood term is represented as a term of the likelihood of object fluents and a term of the likelihood of object action/interactions, learned upon human pose (2) and vehicle parts (3).
  \item[(c)] We formulate the posterior as an Integer Linear Programming and constraining one object only allowing entering a nearby container. The simplified graph structure is a directed acyclic graph and we use dynamic programming to optimize.
\end{itemize}