A Dynamic Bayesian Network Approach to Multi-cue based Visual Tracking

Tao Wang, Qian Diao, Yimin Zhang, Gang Song, Chunrong Lai, Gary Bradski
Intel China Research Center, Beijing, P.R. China, 100020
{ tao.wang, qian.diao, yimin.zhang, gang.song, chunrong.lai, gary.bradski}@intel.com

Abstract

Visual tracking has been an active research field of computer vision. However, robust tracking is still far from satisfactory under conditions of various background clutter, poses and occlusion in the real world. To increase reliability, this paper presents a novel Dynamic Bayesian Networks (DBNs) approach to multi-cue based visual tracking. The method first extracts multi-cue observations such as skin color, ellipse shape, face detection, and then integrates them with hidden motion states in a compact DBN model. By using particle-based inference with multiple cues, our method works well even in background clutter without the need to resort to simplified linear and Gaussian assumptions. The experimental results are compared against the widely used CONDENSATION and KF approaches. Our better tracking results along with ease of fusing new cues in the DBN framework suggest that this technique is a fruitful basis to build top performing visual tracking systems.

1. Introduction

Reliable visual tracking is an important task in human computer interaction (HCI), surveillance, teleconferencing and video compression etc. Within last decades, lots of methods have been proposed including Kalman Filter(KF) [1], CONDENSATION[5], JPDAF[12] and Mean Shift [2] etc. However, most of them use only a single visual cue and are limited to a particular environment that is, typically, static, controlled and known a priori. It is likely that no single cue will be general and robust enough to deal with various scenes in the real world. Motivated by this idea, many methods have been proposed for combining a multitude of complementary cues to increase robustness [3][6][11]. How to optimally fuse multi-cues then becomes the key challenge for developing reliable visual tracking systems.

Dynamic Bayesian networks (DBNs) are a powerful, flexible statistical tool in machine learning and pattern recognition [4][9]. In this paper, we reformulate the well known CONDENSATION[5] method into a more flexible and theoretically sound Dynamic Bayesian network (DBNs) framework. The proposed approach gives substantially improved tracking performance. It differs from previous methods in three ways: (1) Using multi-cue observations to increase the reliability of tracking in a compact DBNs framework; (2) Using particle-based inference to estimate the hidden motion state and thus overcoming the difficulties of non-linear, non-Gaussian and multi-modality cases; (3) Improving ellipse shape features by using skin color to exclude impossible regions.

The rest of the paper is organized as follows: In Section 2, we introduce some important concepts of DBNs before diving into the details of the proposed approach. In Section 3, we give a comprehensive description of the DBN-based approach to visual tracking. In Section 4, two real-world experiments are reported. Finally, concluding remarks are given in Section 5.

2. Concepts of Dynamic Bayesian Networks

DBNs are a branch of Bayesian networks (BNs) for modeling sequential data. A Bayesian network is defined as a directed acyclic graph (DAG) that represents the joint probabilistic distribution for a set of random variables (see in fig.1). Nodes in the graph represent random variables and directed arcs connect pairs of nodes to denote probabilistic relations between variables.

Bayesian networks provide an elegant factorization mechanism through breaking up the complex joint distribution \( P(X_1,X_2,...,X_n) \) into a simple product of some local probability distributions \( P(X_i | Pa(X_i)) \). Using the chain rule, it can be easily proved that:

\[
P(X_1,\ldots,X_n) = \prod_{i=1}^n P(X_i | Pa(X_i)) \quad (1)
\]

where \( Pa(X_i) \) is the parent set of variable \( X_i \). The individual factor \( P(X_i | Pa(X_i)) \) is called conditional probability distribution (CPD) that quantifies the effects of the parents on the node. For continuous variables, the commonly used CPDs include linear Gaussian CPD and mixture of Gaussians CPD.
Dynamic Bayesian Networks (DBNs) are the extension of BNs for modeling time-series data. The term “dynamic” means modeling a dynamic system, and does not mean the graph structure changes along with time. For each step, a DBN model normally processes two time slices to capture the fact that time flows forward from anterior slice1 to current slice2 (see in fig.1). Directed arcs within a time-slice denote intra-correlations and the arcs between two slices indicate inter-correlations, reflecting the causal flow of time.

DBNs are a powerful and convenient tool for modeling dynamic systems. They have particular advantages as follows: (1) They provide a graphical representation of the model which is modular and thus manageable. (2) They exploit the conditional independence properties that allow computational efficiency. (3) They can integrate various observations together into a single consistent probability framework.

3. DBN-based visual tracking

In this section, we focus on how to build a probabilistic visual tracking model using DBNs. The presented approach first extracts multi-cue observations. These observations are then adaptively integrated into a compact DBN framework. Finally, particle-based approximate inference is used to estimate the real motion states.

3.1 Multi-cue observations

(1) Skin color: A common misconception is that different color models are needed for different races of people e.g., blacks and whites. Under uniform illuminations, the fact is that humans are almost the same color in the “hue” except for albinos [2]. Therefore, a hue color histogram is adopted in our approach. When a candidate face region is considered, pixels inside the region are collected to build a hue histogram. The intersection of such a histogram and a pre-trained face histogram gives the color observation’s likelihood of the region. This method is robust to many complicated, non-rigid motions [10]. However, it suffers from illumination and backgrounds with similar color objects.

(2) Ellipse shape: The head can be modeled by an ellipse with aspect ratio of 1:1.2. When a candidate region is considered, the sum of dot products of the intensity gradient vectors and the ellipse normals around the ellipse perimeter gives the shape observation’s likelihood [10]. Since shape features are easily disturbed by background clutter, we overcome the shortcoming by using skin color to avoid impossible candidate region. That is, only if the color likelihood of a candidate region is big enough, will its shape be analyzed. In practice, this method runs well in uniform backgrounds including moderate amounts of occlusions, but fails in heavy background clutter.

(3) Face detection: We use the implementation of Intel OpenCV’s Adaboost face detector to find face like regions [7]. On a P4 2.0G PC, this detection has a speed of about 14 frames/second(fps) in the image resolution 128×96. Face detection is a robust cue for the frontal face. However, it suffers from the head rotation.

One thing worth pointing out is that the proposed DBN-based model is an open framework into which other cues can readily be incorporated for more robust tracking.

3.2 DBN-based Tracking model

The proposed DBN model is illustrated in fig.1. The \( V_i \) node, \( i=1,2 \), represents the head’s velocity \((v_x, v_y, v_s)\) in \( x, y, \) and \( \text{scale} \) dimensions. The \( X_i \) node, \( i=1,2 \), denotes the head’s position \((x_i, y_i, s_i)\) in \( x, y, \) and \( \text{scale} \) dimensions. The \( W_i \) node, \( i=1,2 \), is the weights \((w_{i1}, w_{i2}, ..., w_{ik})\) of the \( K \) multi-cue observations. The \( O_i \) node, \( i=1,2 \), is the head likelihood \( P(O_i | X_i) = (o_{i1}, o_{i2}, ..., o_{ik}) \) of the \( K \) multi-cue observations in the candidate position \( X_i \).

![Figure 1. DBN-based visual tracking model, unrolled for two slices](image)

The DBN model builds on the intuition of exploiting conditional independence properties to allow a compact and natural representation of the joint probability. For computational convenience, we assume all the CPDs are linear Gaussian. Given continuous variables \( Pa(X_i) \) and \( X_i \), the linear Gaussian CPD is defined as:

\[
P(X_i \mid Pa(X_i)) = N(\alpha_i + \beta^T_i Pa(X_i); \sigma_i^2)
\]

where \( \alpha_i + \beta^T_i Pa(X_i) \) is the conditional mean and \( \sigma_i^2 \) is the variance. In fig.1, the velocity variable \( V_2 \) conditionally depends on the variable \( V_1 \) as a linear Gaussian CPD \( P(V_2 \mid V_1) = N(V_1, \sigma_1^2) \). The head position \( X_2 \) equals to the sum of the anterior position \( X_1 \) and the current velocity \( V_2 \) as \( P(X_2 \mid X_1, V_2) = N(X_1, V_2; \sigma_2^2) \).

The multi-cue weights \( W_2 \) conditionally depend on frontal weights \( W_1 \), namely, \( P(W_2 \mid W_1) = N(W_1, \sigma_2^2) \). The total
observation likelihood is decided by each cue’s weight and the position of the head. It can be approximated as:

\[ P(O_2 \mid X_2, W_2) = \sum_{i=1}^{K} w_{2i} \cdot P(O_2 \mid X_2)^T \]

\[ = \sum_{i=1}^{K} w_{2i} O_{2i} \]  

(3)

3.3 Particle-based approximate inference

In the DBN frame of fig.1, a particle-based approximate inference algorithm is used to estimate object’s real motion states (show in procedure 1). By likelihood-weight sampling from the DBN’s model (see procedure 2), many particles {\(X_i = [V_i, X_i, W_i, O_i]\), \(i=1,2,\ldots,N_i\)} with weights \(\omega_i\) are generated to represent various possible states of the tracking object. According to the particle weight \(\omega_{i}\) in anterior slice t-1 and the likelihood \(O_i\) of multi-cue observations in slice t, Eq.4 [4] can update the \(i\)th particle’s likelihood \(\omega_i\) of slice t as:

\[ \omega_{i} = \omega_{i-1} \times P(O_i \mid X_i, W_i) \]  

(4)

In the next time slice, the particles with higher likelihood \(\omega_i\) will be selected with higher probability to predict the object’s new position. Given the evidence \(O_t\), these \(N_i\) particles and their weights \(\omega_i\), \(i=1,2,\ldots,N_i\), approximate the joint distribution of the variables in the network. From these weighted particles, Eq.5 can estimate the real motion state as:

\[ \hat{X_t} = \sum_{i=1}^{N_i} \omega_i X_i, \quad s.t. \sum_{i=1}^{N_i} \omega_i = 1 \]  

(5)

The effective number of particles is \(N_{\text{eff}} = 1 / \sum_{i=1}^{N_i} (\omega_i)^2 \). When \(N_{\text{eff}} < \text{threshold}\), the weights of particles are very skew (e.g. near the degeneracy case). A re-sampling step is then required to increase the effective number of particles.

Particle-based inference has several advantages: It’s easy to implement; It works on almost any kind of model including non-linear, non-Gaussian CPDs; It can convert heuristics into provably correct algorithms by using them as proposal distributions and in the limit of an infinite number of particles, it is guaranteed to give the exact answer[4]. Furthermore, combined with DBNs, it provides an intuitive fusion mechanism for dynamic adaptation among multi-cue observations as well as the parameterization for each single cue.

4. Experiment

The tracking system was developed in VC++ on Win2K platform. The kernel algorithms ofKF and CONDENSATION come from the Intel’s OpenCV functions [7]. For the implementation of the DBN, Intel’s Probabilistic Network Library (OpenPNL) [8] provides building blocks. For a fair comparison, DBNs, KF, CONDENSATION methods are tested on the same data set [10]. The video sequences simulate various tracking conditions including appearance changes, quick movement, rotation, shape deformation, partial occlusion and camera zoom in/out, which together impose great challenges on any visual tracking system.

Running on a Pentium4 2.0G with 400 particles, the average kernel speed of particle-based inference is 401.6 frames/second(fps). In the image resolution 128×96, the average total speed of DBN tracking is 71.9 fps for the color cue, 60.3 fps for the shape cue, 13.0 fps for the face cue and 10.6 fps for the color+shape+face cue respectively. The tracking results for sequence A are shown in fig.2. The top row is the CONDENSATION tracking result using only one cue such as color, face or shape. In image (a), since the box color is similar to the face, tracking will be lost by only using the color cue. In image (b), using only the face cue fails to track when the head rotates. In image (c), the tracking result of just using shape is confused by the strong edges of the shoulder. However, the multi-cue based DBN tracking result is more accurate and robust (see the bottom row of fig.2). This fact can be explained by the adaptive weight curves \(\hat{\omega}_i\) of the DBN (see fig.3). For example, although the color cue is lost in the 78th frame, the correct shape and face cue are generated to represent various possible states of the tracking object. According to the particle weight \(\omega_{i}\) in anterior slice t-1 and the likelihood \(O_i\) of multi-cue observations in slice t, Eq.4 [4] can update the \(i\)th particle’s likelihood \(\omega_i\) of slice t as:

\[ \omega_{i} = \omega_{i-1} \times P(O_i \mid X_i, W_i) \]  

(4)

In the next time slice, the particles with higher likelihood \(\omega_i\) will be selected with higher probability to predict the object’s new position. Given the evidence \(O_t\), these \(N_i\) particles and their weights \(\omega_i\), \(i=1,2,\ldots,N_i\), approximate the joint distribution of the variables in the network. From these weighted particles, Eq.5 can estimate the real motion state as:

\[ \hat{X_t} = \sum_{i=1}^{N_i} \omega_i X_i, \quad s.t. \sum_{i=1}^{N_i} \omega_i = 1 \]  

(5)

The effective number of particles is \(N_{\text{eff}} = 1 / \sum_{i=1}^{N_i} (\omega_i)^2 \). When \(N_{\text{eff}} < \text{threshold}\), the weights of particles are very skew (e.g. near the degeneracy case). A re-sampling step is then required to increase the effective number of particles.

Particle-based inference has several advantages: It’s easy to implement; It works on almost any kind of model including non-linear, non-Gaussian CPDs; It can convert heuristics into provably correct algorithms by using them as proposal distributions and in the limit of an infinite number of particles, it is guaranteed to give the exact answer[4]. Furthermore, combined with DBNs, it provides an intuitive fusion mechanism for dynamic adaptation among multi-cue observations as well as the parameterization for each single cue.

4. Experiment

The tracking system was developed in VC++ on Win2K platform. The kernel algorithms of KF and CONDENSATION come from the Intel’s OpenCV functions [7]. For the implementation of the DBN, Intel’s Probabilistic Network Library (OpenPNL) [8] provides building blocks. For a fair comparison, DBNs, KF,
can remedy the error. When the frontal face disappears from the 165th to the 255th frame, the weight of face and color cue will automatically decrease. However, the shape cue correspondingly increases to keep tracking robustly.

Another tracking result of sequence B is shown in fig.4. Compared with KF and CONDENSATION, DBNs can fuse observations of skin color, ellipse shape and face detection adaptively and achieve better performance.

5. Conclusion

Dynamic Bayesian networks are a powerful tool in machine learning and pattern recognition. In this paper, we propose an approach to visual tracking of integrating multi-cue observations and hidden motion states into a compact DBN framework. Further, particle-based approximate inference is used to estimate the real motion state efficiently. Experimental results demonstrate that the DBN-based tracking system performs better than the conventional methods such as KF and CONDENSATION.

References