Image and Vision Computing 29 (2011) 594–606

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

A hierarchical feature fusion framework for adaptive visual tracking ☆

Alexandros Makris [a,b,*], Dimitrios Kosmopoulos [a], Stavros Perantonis [a], Sergios Theodoridis [b]

[a] NCSR Demokritos, Institute for Informatics and Telecommunications, Computational Intelligence Laboratory, 15310, Aghia Paraskevi, Athens, Greece
[b] University of Athens, Department of Informatics, 15771 Athens, Greece

ARTICLE INFO

Article history:
Received 4 June 2009
Received in revised form 10 March 2011
Accepted 1 July 2011

Keywords:
Visual tracking
Particle filter
Sequential Monte-Carlo

ABSTRACT

A *Hierarchical Model Fusion* (HMF) framework for object tracking in video sequences is presented. The Bayesian tracking equations are extended to account for multiple object models. With these equations as a basis a particle filter algorithm is developed to efficiently cope with the multi-modal distributions emerging from cluttered scenes. The update of each object model takes place hierarchically so that the lower dimensional object models, which are updated first, guide the search in the parameter space of the subsequent object models to relevant regions thus reducing the computational complexity. A method for object model adaptation is also developed. We apply the proposed framework by fusing salient points, blobs, and edges as features and verify experimentally its effectiveness in challenging conditions.

1. Introduction

The problem of visual tracking consists in the localization of moving scene objects (targets) in consecutive frames acquired by static or moving sensors. It has a broad scope of applications ranging from human–computer interfaces, to surveillance. Its general solution might be very challenging especially when the targets are deformable, move abruptly in front of heavily cluttered background under varying illumination conditions and are partially or fully occluded.

A very popular approach is the probabilistic Bayesian tracking methods. The Sequential Monte Carlo (SMC) approximation methods [3–7] known as particle filters (PF) which belong to the Bayesian approach are among the most promising approaches for robust tracking. These methods treat the location of the target as a probability density function, which they attempt to estimate using a set of samples. Their main advantage lies in their ability to cope with multi-modal distributions, such as those emerging from a cluttered environment, due to the maintenance of multiple hypotheses. However, the application of particle filtering methods still has many problems to resolve before it can be considered robust for tracking targets in natural scenes in real-time. Among the most important issues are the efficient and information-rich *target representation* and the selection of the *proposal distribution* (hypothesis generation).

In this work we propose the *Hierarchical Model Fusion* (HMF) framework for fusing visual cues. The target is represented by several *object models* of increasing dimension, which are probabilistically linked. The parameter update for each object model takes place hierarchically so that the simpler object models, which are updated first, guide the search in the state space of the more complex object models to relevant regions. The most complicated object model (in terms of state dimension) and the last in hierarchy, is called *main model* and its parameters fully describe the target. The rest of the object models are referred to as *auxiliary* as the estimation of their state is not required by the application. A method to adapt the auxiliary object models to cope with target appearance changes is also proposed. The method deletes the auxiliary object models which seem to lose track based on a measure of their compatibility with the main object model. When the number of auxiliary models is low new ones are added.

A simple example (see Fig. 1) will clarify the proposed concept. Let us consider a case of a target of which we want to estimate the bounding box. We will use two object models, an auxiliary that tracks a salient point in the target and the main model, the bounding rectangle (blob). The state of the first object model has two parameters, the salient point's coordinates $x_s=[i_{s_x};i_{s_y}]$, while the blob object model has three, the coordinates of its center and a scale parameter, $x_b=[i_{b_x};i_{b_y};s_b]$. When the tracking is initialized the relative position of x_b with respect to x_s is measured. If the tracked object is rigid this relative position should be almost constant between two consecutive frames. Thus if the location of the point is found on the next frame we can significantly narrow the search for the position of the blob thus the search in the three-dimensional space is simplified.

The contribution of our work consists in the following:

- We extended the Bayesian framework to allow the integration of multiple object models which may lead to a better *target representation*.
- We derived a particle filtering based approximation algorithm which leads to efficient *hypothesis generation*. This algorithm integrates

☆ This paper has been recommended for acceptance by Dimitris Samaras.

* Corresponding author. Tel.: +30 2106503141.
E-mail addresses: amakris@iit.demokritos.gr (A. Makris), dkosmo@iit.demokritos.gr (D. Kosmopoulos), sper@iit.demokritos.gr (S. Perantonis), stheodor@di.uoa.gr (S. Theodoridis).

0262-8856/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2011.07.001

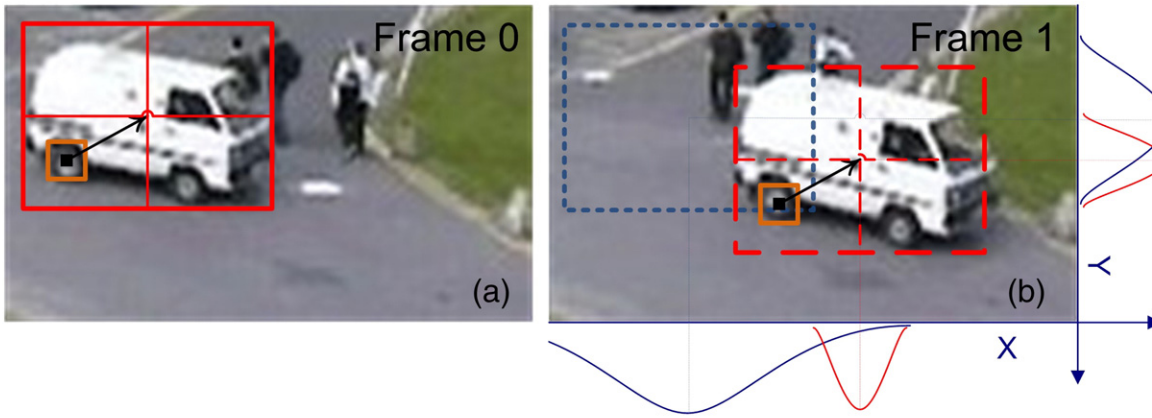

Fig. 1. In (a) the tracking is initialized with two object models describing the target, a blob and a salient point. The arrow shows the relative position of the point and the center of the blob. In (b) the position of the point is updated and using the stored relative distance the proposal for the blob given this position is shown in the x and y axis (red). This proposal is much closer to the target than the proposal derived by the state evolution model of the blob (blue).

multiple object models of different complexity with redundancy of information.
- We developed an *adaptation* technique, to automatically select appropriate auxiliary models.

To test the proposed HMF framework we implemented a tracker using the following cues: salient points within the target, color, and edge information. The object models used, in increasing state dimension order, are salient points and blobs within the target as auxiliary and the target's contour as the main model. A possible drawback of our method that we are going to handle in the future is the dependence on the main model.

The rest of the paper is structured as follows. Section 2 describes the merits of our approach compared to the related works. Section 3 provides the background knowledge to particle filtering methods, presents the HMF framework, and explains its requirements and constraints. Section 4 gives an implementation of the framework. Section 5 contains the experimental results, which demonstrate the merits of the proposed HMF framework using some challenging video shots and simulated data. Section 6 concludes this work.

2. Related work

In this section we overview the related works on tracking and how they treat the *target representation* (object model), the *hypothesis generation/evaluation* (proposal distribution/measurement model), and the *adaptation* compared to our approach. A recent survey on tracking methods can be found here [8].

The choice of the *object model* is crucial and depends on the speed and accuracy requirements of the application. Simple object models such as [9–13] with 3–4 state dimensions are efficiently calculated, but the amount of information they provide is low and requires significant post-processing for video understanding. On the other hand, object models with many state parameters have high computational cost. Examples are [14,1] with 6 state parameters, or target specific models with even more parameters [15–17]. The acquisition of those parameters is often required by the application and the high level knowledge they provide results in a detailed target representation, thus these object models are more difficult to be distracted by clutter. In this work we propose the use of several object models of varying complexity in an attempt to maintain the benefits of both simple and complicated models, by using a coarse to fine strategy.

Another important decision when designing a tracking algorithm is the choice of the *measurement model* and the *proposal distribution*. There are many works in the literature using particle filters with only a single cue (e.g. edges, color) and use the state evolution (dynamics) as a proposal [1,9]. However, using only one cue limits the robustness and sampling from dynamics is inefficient as is now acknowledged by many recent works [13]. The target's motion is hard to predict and its direction or velocity might change abruptly. To account for this kind of motion the range of search should be very wide. This results in inefficient search because many hypotheses are created in low likelihood regions. Furthermore, a large search range increases the probability of tracker distraction by similar objects in the target's neighborhood. Feature fusion is a popular approach to overcome these difficulties, and several methods which attempt it have recently appeared [10,18–22]. These approaches differ in the way they fuse the cues and can be classified in three categories: the first concerns methods which combine several cues during the measurement process to increase robustness. The methods of the second category try to improve the proposal distribution by using some of the cues to guide the new hypotheses in high likelihood areas. The third category contains methods which partition the state space and use different visual cues to sequentially update the resulting sub-states. In the following we are going to outline some methods from all three categories. The main challenge for these methods is to fuse the cues in a way that will increase robustness while maintaining a low computational cost since many tracking applications require on-line performance.

In [18,2] two frameworks for fusion during the measurement process are presented. In [2], a method to automatically estimate the reliability of each feature is also proposed. Similarly in [23], a method to evaluate and select the most suitable features for a given application is presented. Another relative approach is [24], which builds likelihood maps from each feature and combines them based on their classification confidence scores. A limitation of the aforementioned methods is that the object models that are coupled with the various cues must share the same state space. The number of particles required increases exponentially with the number of state parameters, rendering these methods inefficient. In [21], the authors overcome this drawback by using several object models to fuse different cues. This strategy maintains information redundancy and lower computational complexity by splitting the state into several subspaces, however, the cues are fused during the measurement process which does not improve the proposal distribution and thus might result in inefficient search of the state space. The same limitation holds for [25], where two object models are used for head tracking with the particles of each model updated using a Monte Carlo approximation to sequential belief propagation.

The second category of the fusion methods combine the cues during the hypothesis generation stage. Some of them, propose the use of some sort of low level information such as color [26], and