

# Tracking a detected face with dynamic programming

Zhengrong Yao \*, Haibo Li

*Digital Media Lab, Applied Physics and Electronics, Umea University, 901 87 Umea, Sweden*

Received 30 November 2004; received in revised form 11 August 2005; accepted 15 September 2005

## Abstract

In this paper, we consider the problem of tracking a moving human face in front of a video camera in real-time for a model-based coding application. The 3D head tracking in a MBC system could be implemented sequentially as 2D location tracking, coarse 3D orientation estimation and accurate 3D motion estimation. This work focuses on the 2D location tracking of one face object through continuously using a face detector. The face detection scheme is based on a boosted cascade of simple Haar-like feature classifiers. Although such a detector demonstrated rapid processing speed, high detection rate can only be achieved for rather strictly near front faces. This introduces the ‘loss of tracking’ problem when used in 2D tracking. This paper suggests an easy method of solving the pose problem by using the technique of Dynamic Programming. The Haar-like facial features used in the 2D face detector are spatially arranged into a 1D deformable face graph and the Dynamic Programming matching is used to handle the ‘loss of track’ problem. Dynamic Programming matches the deformed version of the face graph extracted from a rotated face with the template taken online before ‘loss of tracking’ happens. Since the deformable face graph covers a big pose variation, the developed technique is robust in tracking rotated faces. Embedding Haar-like facial features into a deformable face graph is the key feature of our tracking scheme. A real time tracking system based on this technique has been set up and tested. Encouraging results have been got and are reported.

© 2005 Elsevier B.V. All rights reserved.

*Keywords:* Model-based coding; Face tracking; Face detection; Dynamic Programming

## 1. Introduction

We consider the problem of detecting and tracking an arbitrarily moving human face in front of a video camera in real-time. Real-time face tracking is a key step in many real-world applications such as Model-based coding (MBC), teleconference, human-machine interface, face animation, visual surveillance, etc. The targeted application in this work is MBC. The goal of face tracking in MBC is to extract motion parameters of the 3D head motion [1].

### 1.1. Model-based Coding

MBC is a promising video coding technique aimed for very low bit rate video transmission [2–5]. The aim is realized through making use of a 3D face model. At the sender side, the 3D motion information and facial texture information are extracted and parameterized. Only these high-level semantic parameters are transmitted to the receiver side to reconstruct

the video. Very high compression is thus achieved. Fig. 1 shows a basic block diagram of a MBC system. At the encoder (sender) side, the scene parameters are extracted making use of 3D head model. The decoder (receiver) then synthesizes the video with these parameters. The 3D motion extraction consists of a major module to help to realize the low bit rate aim of MBC. Although huge effort has been put into the 3D motion extraction problem, a fully automatic solution is still a research dream. In practice, the extraction of 3D motion could often be implemented in three steps: locating the 2D position, estimating the coarse orientation, and extracting accurate pose parameters [6]. Multiple clues could be used in all these stages to improve the estimation performance. In this work, we focus on the first step and try to robustly locate the face position even when big rotation happens. For simplicity we limit ourselves to a single user case. The task is to try to estimate the 2D location of one single face in video stream and the main concern is its robustness.

### 1.2. Face tracking in Model-based Coding

The 3D motion could often be implemented in a hierarchical manner and the estimation consists of three steps: firstly the 2D position, then the coarse orientation, and finally the accurate pose parameters. The advantage of using a hierarchical

\* Corresponding author. Tel.: +46 46 2864252; fax: +46 46 2864257.

E-mail address: [zhengrong.yao@scalado.com](mailto:zhengrong.yao@scalado.com) (Z. Yao).

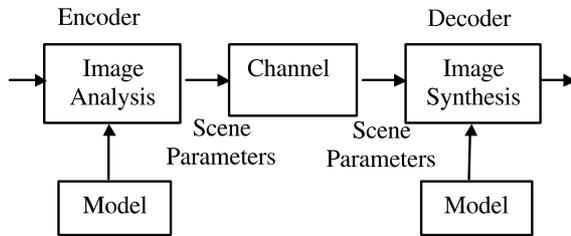


Fig. 1. Block diagram for a Model-based coding system.

manner is that it can make use of multiple clues and tends to be robust [6].

One key issue for vision based face tracking is the selection of the observations to work with. The possible choices often include image intensity or extracted features (edges, Gabor jets). The accurate face tracking schemes often track the selected ‘good’ features. To judge if a set of facial features is good, two general criteria exist. One criterion is to check if the selected features could capture or encode the face object efficiently and reliably. Another criterion is to check if the selected features are easy to be tracked.

Performance of face tracking will depend very much on the selection of suitable facial features. Different types of facial features such as skin color, edges, feature points, Gabor jets and motion have been used for face tracking. Skin color is tried for tracking face motion in  $X$ ,  $Y$  direction and out-of-plane rotation in [7]. It is often too simple to encode structural knowledge of face, it is thus good for coarse face tracking. An optical flow field [1] has been adopted for face tracking. Dense motion information makes face tracking easier. A major constraint is that optical flow estimation is subject to the aperture effect and usually fails at big movement [8]. Salient facial feature points [9] are a better choice for accurate face tracking. The main shortcoming is that tracking of point features is easily impaired by noise and often the face appearing in the video has to be large enough to facilitate feature tracking. By contrast, the facial features characterized by Gabor jet [10] have been proved to be better than other types of features.

Since 2D face tracking could be implemented as continuously employing a face detector, the face tracking benefits directly from the technique of face detection. A milestone in face detection is the introduction of a general framework for learning based object detection [11,12]. In this framework, the object class is represented by an overcomplete dictionary of wavelet basis function. Through learning, a compact representation consisting of a subset of wavelet basis functions could be derived for rapid object detection. Based on this framework, a boosted cascade scheme [13] speeds up object detection very much and demonstrates both extremely rapid processing speed and also high detection rate, which is comparable to some much more expensive existing systems. The false alarm may reduce further by 12.5% by adding a new set of rotated Haar-like features and introducing a novel post optimization procedure [14].

This give us the hints of using this efficient face detector continuously to fulfill the task of 2D face tracking. Since we

limit us to the single user case in this work, we suppose the continuously detected face is the same subject face.

## 2. Face tracking with face detector

In order to study the feasibility of utilizing Haar-like features for face tracking, we perform a face tracking experiment by sequential employment of the Haar-like feature based face detector provided by the OpenCV library [15]. The tests are performed on each frame of testing head-and shoulder video sequences.

The testing video sequences come from the well-known work of head motion estimation [16]. They are also publicly available for downloading on the web site: <http://www.cs.bu.edu/groups/ivc/HeadTracking/>. Fig. 2 shows some example cases where the face detector failed to detect the face object. Among all the provided video sequences of a moving subject, we tested with only those whose head experienced large rotation. In other word, we tested only the ‘difficult’ sequences. We selected 30 sequences from them for testing. Each video sequence consists of 200 frames. Altogether there are 6000 video frames from all the testing video sequences. The detector works well with near frontal face and could handle small rotation (approximately less than  $20^\circ$ ), but fails in a big head rotation case. Among all video frames, 2085 of them failed to report a face (The sequences used for testing are listed in Table 1). With the default parameter setting, only around 65% frames have correctly reported as having one face.

Since the classifiers used in face detection were trained by learning from near frontal faces, most failures are simply due to lack of knowledge of faces under big rotation. One may think that this problem can be solved by adding largely rotated faces into the training data set. This might work but would dramatically increase the complexity of both the leaning processing and the detecting processing. In [17] side view is considered when training and the searching is much more complicated compared to the frontal face detector. In fact simple analysis shows that the limitation of pose variation is



Fig. 2. Face detector based on Haar-like has problem treating big pose variation. The original Haar-like feature based face detector failed to detect the faces in these examples.

متن کامل مقاله

دریافت فوری ←

**ISI**Articles

مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات