Model based video coding using open-loop architecture

Zhengrong Yao ¹
Dept. Applied Physics and Electronics
Umeå University
SE-90187, Umeå Sweden
e-mail: Zhengrong.Yao@tfe.umu.se

Haibo Li ²
Dept. Applied Physics and Electronics
Umeå University
SE-90187, Umeå Sweden
e-mail: haibo.li@tfe.umu.se

ISSN Number: 1652-8441
Report Date: July 5, 2004
Abstract

This paper addresses an important issue in Model-based coding, namely, how to extract the motion information of a head object from a video sequence. Traditional methods use a closed-loop coding architecture. In this paper we try to use an open-loop coding architecture instead. Active tracking is chosen for motion estimation. Two cameras are utilized to estimate global and local motion independently. Our theoretical analysis and experiment show that this is a cost effective way to set up the Model-based coding system.
1 Introduction

Model-based coding (MBC) is a promising video coding technique aiming at very low bit rate video transmissions [4]. Major contributions in this field include [2][6][1]. The idea behind the technique is to parameterize a talking face based on a 3D face model. Parameters describing facial movements and texture information are extracted at the transmitter side. The extracted parameters are then sent to the receiver to synthesize the talking head. Very high compression is achieved since only high-level, semantic motion parameters are transmitted.

Today, MBC solutions are more required than ever. Consider a typical networked conference application: a roaming mobile user is travelling when an important conference is held at his company. He wants to be a "remote" attendant using a high-end mobile phone, so that he can see what happens at the conference. The mobile user will experience different communication networks during his trip. The network bandwidth along his trip consequently undergoes huge variations. The handovers between different networks often make the situation even worse. Streaming service in this case is often available at a rate of 50-1500 kbps [12]. At extremely bad network conditions, the common streaming service schemes cannot handle video with an acceptable quality. One has to resort to coding adaptation based on media type, such as speech to text or video to animation [8]. A MBC system is an obvious need for such cases.

For a MBC system, in terms of the coupling relationship between the encoder and decoder and how image analysis and synthesis interact, the coding architecture can be divided into two types: open-loop and closed-loop architectures. The coding architecture decides which coding scheme that is proper for the architecture, it is thus a key factor needed to be considered.

1.1 Closed-Loop Architecture

In the closed-loop architecture, image synthesis is embedded into the image analysis loop as shown in Fig. 1. The encoder contains both image analysis and synthesis and so called Analysis-by-synthesis (ABS). Only image synthesis is contained in the decoder, that is the part in the shaded area in Fig. 1. With a closed-loop architecture, it is possible to check the quality of the extracted parameters directly through an image synthesis module. The encoder is fully aware of the quality of the rendered image at the decoder and knows if updated parameters make a real improvement. The closed-loop ABS scheme is powerful and has been the traditional scheme used in MBC since the early 1990s [10][7][9]. Although being powerful, the closed-loop architecture is not always devoid of difficulties. The motion estimation problem has not been successfully solved yet, due to the inherent difficulties in modelling the illumination changes, non-rigid motion, etc.

In this paper, we suggest using open-loop architecture to set up a MBC system. The following sections will discuss the advantage of open-loop architecture and our trial.

2 Open-loop architecture

In an open-loop structure, image analysis and image synthesis are two totally independent modules as shown in Fig. 2. One analysis model is used in the encoder while another synthesis model is used in the decoder. The only connection between image analysis and image synthesis is given by the scene parameters.

Serious problems arise from a communication point of view when using open-loop architecture. The most obvious one is that it is impossible for the encoder to control the quality of the image synthesis. For a MBC application, the idea of adopting an open-loop architecture is often ignored and never catch researcher interests.

As the development of the coding techniques, more and more people have realized that it is not a good idea to couple the encoder and decoder with the exact same physical model and other shared settings. Instead, a description protocol could be a wiser choice. MPEG-4 has already adopted this strategy. A set of facial features are selected and defined in MPEG-4. Only animation protocols, face animation parameters (FAP) and face
definition parameters (FDP) are standardized, but not any particular face model. This naturally suggest the usage of an open-loop architecture.

Figure 1: The encoder in the close-loop MBC coding architecture.

Figure 2: The open-loop MBC coding architecture. The model at the decoder is independent of the one at encoder.
2.1 Passive and active tracking

It has been realized that the 3D motion estimation of the non-rigid head object is a key problem in MBC, no matter what is the coding architecture. In traditional MBC, a passive tracking system is often used. The camera is mounted in front of the user (Fig 3). In each coding loop, the synthesized image is subjected to comparison with input frame.

When using open-loop architecture, this comparison step is omitted. This suggests that the input frame is unnecessary, and so the camera! In theory, any method could be used to extract the motion parameters, for example, a magnetic sensor could be used to provide the motion parameter under an open-loop architecture. Since in work we are still interested in using vision-based methods to extract motion parameters, the active tracking (head-mounted camera) could naturally be a candidate. The advantages of using active tracking include:

- The global motion estimation and local motion estimation could be separated.
- The global motion estimation becomes easier, since the camera could now look at a rigid environment (object, background), this is much better compared to looking at the non-rigid face object.
- The local motion estimation becomes easier, since it looks at a rather "stable" face, which is much easier to code.

The problem with passive tracking is that usually only a small region contains changes associated with the movement. A passive tracking system can have difficulties to estimate small movements with high accuracy. It is easier to detect even small changes using active tracking. We will use a simple example to compare passive tracking and active tracking in estimating motion parameters and will show the advantage of using active tracking.

Consider a simplified rotation example where there is only rotation around the y-axis (see Fig. 3). The two possibilities of camera setting are passive tracking (camera at point A) and active tracking (camera at B). As the head turns, the changes in the resulting images are calculated for both cases based on a perspective camera model.

\[
\Delta x_1 = \frac{f}{\sqrt{2r_2 - r_1}}
\]

\((1)\)

Figure 3: A head and a camera viewed from above. Two different possibilities of mounting a camera: 1. The camera is positioned at A. 2. The camera is at B. The dashed lines represent the new positions after rotation.
and for active tracking:

$$\Delta x_2 = f \frac{r_1}{r_2}$$  \hfill (2)$$

$$f - \frac{r_1}{\sqrt{2}r_2 - r_1} \ll f \Rightarrow \Delta x_1 \ll \Delta x_2 \text{ as } r_1 \ll r_2$$  \hfill (3)$$

As an example, $f = 100$, $r_2 = 80$ cm, and $r_1 = 15$ cm (the radius of the head). Using Equation (3), we get an estimate of the change in pixels, for passive tracking and active tracking respectively:

$$\Delta x_1 = \left( \frac{0.15}{\sqrt{2} \cdot 0.8 - 0.15} \right) \cdot 0.01 \approx 15.3 \text{ pixels}$$

$$\Delta x_2 = \frac{1}{0.01} = 100 \text{ pixels}$$

It is clear that the change is easier to detect rotation with active tracking compared to passive tracking. The estimation error, for one pixel in our example, is with active tracking in the order of 10 times less prone to error than with passive tracking. There is no difference in changes in the images between the two cases for translation. The advantage of using the active tracking lies in that it gives changes in the entire images, while the passive tracking only gives changes in the head region.

In our work, we try to use active motion tracking given the advantage of open-loop architecture. An overview of the camera setting is shown in Fig. 4, where two cameras are mounted on the user’s head, one facing the user and one facing forward. Our intention is to use the camera in front of the user as a ”local” camera that estimates the local facial motion. The camera on top of the user head is used as a ”global” camera responsible for global motion estimation.

### Figure 4: The camera settings of MBC using active tracking. A web camera is mounted in front of the user’s face and estimates the local motion, another camera is mounted on top of the user’s head to estimate the global motion. The head mounted device is shown from another angle on the top left corner.

In our work, we try to use active motion tracking given the advantage of open-loop architecture. An overview of the camera setting is shown in Fig. 4, where two cameras are mounted on the user’s head, one facing the user and one facing forward. Our intention is to use the camera in front of the user as a ”local” camera that estimates the local facial motion. The camera on top of the user head is used as a ”global” camera responsible for global motion estimation.

### 3 System implementation

By using active tracking cameras, the motion estimation task is greatly simplified compared to traditional passive estimation. Traditional MBC techniques ”jointly” estimate the global and local motion [7]. In our case the aim of motion estimation could be described as: estimating the facial expression with a rather fixed face pose (local camera) first, and then adding the global motion by observing the environmental movement to enhance the understanding of the talking head.
3.1 Motion estimation

The "global" motion is inferred through the moving environmental image, in our case the video image of a bright rectangular computer screen. This is a much easier case than non-rigid face motion since computer screens often have a rigid, planar surface for which a lot of solutions have been proven to be effective [11, 5].

In our system we adopt the POSIT [3] algorithm to estimate the global motion. Four marker points are used as object points. Three markers are mounted at three corners of the screen and one is placed in front of the screen on the end of a pen (which is illustrated in Fig. 5)

The local facial motion estimation is also simplified due to the very small motions of the camera relative to the head. The changing facial expression is estimated using PCA technique.

The decoder side also has the flexibility of using a different model and synthesis through any favorable scheme. In our system setting, the analysis at the encoder uses two active tracking camera systems to individually extract the global motion (rotation and translation) and the local facial motion (animation). The facial animation extraction at the encoder uses a 2D model (PCA in 2D) while at the decoder side, the 3D wireframe model Candide [1] is used to synthesize the video.

This motion estimation strategy could be also implemented in close-loop architecture, although much more complicated due to the need of cameras calibration.

3.2 Experiment and conclusion

We set up an inexpensive MBC system with two cameras and four markers on screen. In our experiment, we use one third camera to act as a "checking" camera located in front of the user as the traditional passive camera. Fig. 5 shows two screen shot of our system. On the left are video frames recorded from the "checking" camera at encoder side, on the right are the video frame from the global camera overlapped with a synthesized talking head (at decoder side).

In the top row, the artifacts in the face are due to the synthesized texture. With our local camera we catch only one part of the face, including the central area. The whole facial texture is made up of two parts: the synthesized inner area and an unchanged outer area. More work is being done to improve the synthesis quality.

The bottom row shows another scene which uses another head model and texture. The two cases used the same "global" test video sequence. The rotation of the synthesized head shows the correct motion as can be seen from the "checking" video, although the difference would be too small to detect with a passive tracking camera.

We conclude that using an open-loop architecture with an active tracking scheme is a very cost efficient way to set up a Model-based coding system.

References


Figure 5: Screen shots of our MBC system. The left column shows the "checking" video from a third camera at encoder side. The right column shows the "global" video overlapped with the synthesized talking head at decoder side.