Enabling Object Based In-content Placement for Digital Video Television Systems

Rudolf Jaeger¹, Peter Dunker², Andreas Hanusch³,

¹) University of Applied Science Giessen-Friedberg
Fachbereich IEM - Digital TV Laboratory
Wilhelm–Leuschner-Strasse 13
61169 Friedberg; Germany
rudolf.jaeger@iem.fh-friedberg.de

2) Fraunhofer Gesellschaft, IDMT, 98693 Ilmenau, Germany

³) Accenture Technology Solutions GmbH
61476 Kronberg, Germany

Abstract
This paper presents a framework for object based interactivity in Digital Video Broadcast (DVB) Television systems. Based upon pre-recorded MPEG-2 video material an object annotation and synchronization scheme has been designed and implemented enabling the viewer to actively interact on previously marked video objects. The framework comprises an authoring process which extracts the necessary Meta data the introduction of a new timeline in the transport stream and the synchronization process at the client site. To show its feasibility at the client site the MHP platform was used. Obtained results of various content samples indicate that our approach is promising and effective thus offering new service models such as e.g. automatic in-program content placements.

1. Introduction

The motivation of our work lies in the fact that advanced digital television systems will provide the viewers with new interactive services, which extends the simple provisioning of event related information regardless provided over the broadcast channel or via a return channel.

Future interactivity will be personalized and the degree of granularity will move to object based interactivity. Although future media systems will employ the MPEG-4 standard for compression, current DVB Television systems use the MPEG-2 standard for compression and delivery the media. Enabling object based interactivity to a DVB/MPEG-2 video stream requires the following two functionalities

(1) Marking a moving object, e.g. with a colored overlay, on the TV screen thus showing the viewer that there is some interactivity upon the marked object possible.

(2) Enabling the viewer to apply some functionality upon this marked object, e.g. to retrieve some additional information of the marked object or to save the shot of the video for further processing later on.

First possible means of synchronizing video streams with additional data on a frame-based level were presented in [1], [2]. In [3] a complex object segmentation algorithm was described which is applied to uncompressed video allowing a classification of every pixel in every frame of a video sequence. In [4] an object tracking scheme is presented using the Multiple Hypothesis Tracking (MHT) and Hausdorff image matching algorithm. Here the aim is to organize individual edges into objects given their two-dimensional models.

In chapter two the requirements to fulfill object based interactivity at the play out site and at the client of DVB systems are listed. Chapter three presents the employed object based tracking algorithm for extracting the necessary Meta data and the timing model allowing object based synchronization for each single video frame. The last chapter provides an implementation example verifying the usefulness of our approach and showing its practical usage.

2. Requirements for marking object in DVB/MPEG-2 systems

In order to achieve object based synchronization the two steps Extracting Meta Data and Synchronizing Meta data with the Video Stream need to be carried out.
2.1 Extracting Meta data
Since the interactivity scenario should be applied for pre-recorded MPEG-2 video material an annotation process of the considered video material needs to be carried out in advance. Allowing object based interactivity requires that for every frame the X- and Y-coordinates and the proper MPEG-2 timing information of the marked objects are extracted. The timing information is needed to place e.g. an overlay frame accurate on the marked object. These so-called Meta data are essential because they provide a link to the marked object of the video material. In chapter 3.1 the employed algorithm and the developed annotation tool for extracting these Meta data will be explained.

2.2 Synchronizing Meta Data with the Video Stream
In order to synchronize the Meta data with the video the Meta data need to be present at the client when the video is decoded. The synchronization at the client can be achieved by implementing a background process which does a comparison of the MPEG-2 timing part of the Meta data with the actual video stream. The employed timing model will be discussed in chapter 3.2.

3. A Model for object based synchronization in DVB/MPEG-2 systems

3.1 Extracting Meta data
As stated in chapter two overlaying a video object for a certain period of time the
(1) x-/y-coordinates of the marked object and
(2) Presentation Time Stamp (PTS) i.e. the presentation time for each MPEG-2 frame
are essential. Based upon the work of [5] an object segmentation tool has been developed [6]. Its software architecture consists of a MPEG-2 demultiplexer, a MPEG-2 decoder with an analysis unit for obtaining the PTS-values, an enhanced version of the CAMSHIFT algorithm [7] of Intel’s OPENCV lib and a rendering unit all implemented as DirectShow filter. The CAMSHIFT algorithm that comes with the OpenCV lib was modified and implemented using Microsoft’s DirectShow dataflow architecture. In order to obtain the Meta data the composer selects an object by pulling over the mouse and than either specifies the end time of the desired video sequence or the Meta data generation process automatically stops at the time when the object disappears from the scene. During the tracking process the CAMSHIFT algorithm will track the location of the marked object and reports the spatial coordinate of the center of mass of the tracked object and its orientation with respect to the x-y plane as well as the PTS timing values for every frame. Figure 1 shows the U/I of the annotation tool while extracting the Meta data of the marked shirt. The orientation of the tracked object is depicted by the rectangle. Figure 2 shows the corresponding shape of the marked object.

Figure 1: U/I of the annotation tool while tracking a marked object (rectangle on the shirt)
A slider (1) in figure 1 allows the composer to quickly move to the desired object for the easy usage of the tool. The composer who marks the object can also add additional information (e.g. shot name, object description, etc.) (3) in figure 1 which will become part of the Meta data and can be used for further interactions or for archiving additional information. It also contains a control slider (2) in figure 1 to adjust the brightness and saturation of the video frame thus a clear distinction of the background and the marked object can be achieved for the optimal usage of the CAMSHIFT algorithm.

Figure 2: shape of the annotated object (back projection image)
The generated Meta data are stored into an XML formatted file that is in accordance to the MPEG-7 standard. In figure 3 an excerpt of a semi automatic produced XML file is shown, reflecting the PTS timing information (underlined), coordinates (italicized) and further description of the marked object related to figure 1. The KeyTimePoint entry is repeated for every frame of the considered sequence.

Add 1) Clock References provide a way to ‘transfer’ the encoders System Time Clock (STC) to the decoder. Clock References carried in Transport Streams (TS) are called Program Clock References (PCR). Within a multiplexed transport stream (i.e. a TV service) PCR values that are samples of the STC are copied into the TS packets. Thus it is possible to regenerate the encoder’s STC.

Add 2) Timestamps are also samples of the encoder’s STC. They are located in the header of PES packets and represent times (in terms of the encoder’s STC) of the audio or video frames. A PTS indicates the time at which a frame leaves the decoder to be presented on the output device.

Reviewing the MPEG-2/DVB Timing Model
MPEG-2 allows that audio and video is prepared and edited separately prior to multiplexing and transferring (s. Figure 4). Therefore a timing model is specified which can be categorized in two main parts:

1. Clock References (CR)
2. Decoding and Presentation Timestamps (DTS, PTS)

Figure 4: The DVB/MPEG-2 timing model

Introducing a new Timeline for frame based Synchronization at the Client site
In order to synchronize the overlay frame accurately, accessing the PTS parameter at the Decoder would be required. Due to the fact that accessing those timing parameters is MPEG-2 chip and firmware specific and thus highly dependant upon the STB suppliers an independent timing model approach was defined. A new time line of the MPEG-2 transport stream (TS) as private section is added. Figure 5 shows a TS excerpt where a new time line (section data) is added. This means the TV service normally consisting of a TS excerpt where a new time line (section data) is added. This means the TV service normally consisting of a TS excerpt where a new time line (section data) is added. This means the TV service normally consisting of a TS excerpt where a new time line (section data) is added.

Figure 3: Sample of coded Meta data for the marked object

3.2 Synchronizing Meta data with the video stream
In order to overlay the video frame accurately with some data e.g. a colored overlay at the client, a synchronization mechanism of Meta data with video stream is required. This will be achieved by introducing a new time line in the multiplexed video/audio stream and compared with the extracted timing information of the Meta data. Following a brief repetition of the MPEG-2 timing model is given.

Add 1) Clock References (CR)
Add 2) Timestamps are also samples of the encoder’s STC.

Figure 5: TS excerpt where a new time line (section data) is added.
packets. The only requirement for the introduction of private section is that there are stuffing packets available. Some of these stuffing packets will be replaced by packets representing the new time line. In order to minimize the error the PCR sample, sent as private section, should be close to the TS packet containing the original PCR value. If the distance is too high a correction of the PCR is needed.

These timing information needs to be compared with the Meta data at the client site. At the client site only a section filter mechanism that is usually available at any middleware and a timing comparison algorithm is needed. This unique and generic approach is applicable for any MPEG-2 transportation model which means it is usable for the broadcast as well as for the on demand service model. For obtaining the PCR values of the considered TV service a head end software tool was developed [8] which browses the designated video component of the transport stream and then copies the PCR value (usually bound to the video component) in the private section. Following to this an update of the Program Map Table i.e. adding a new component was done.

Figure 5: Adding private section to generate a new timeline

Figure 6: UML classes to synchronize the overlay with the marked video object at the client

The main purpose of the client application is

1) To show that an overlay is synchronized frame accurately to a moving object and

2) To Enables simple interaction upon this object.

The crucial part of the client application concerns accurate synchronization of the overlay with marked video object. It comprises the permanent reading of the introduced PCR timeline (private section), the comparison of the PTS timing values provided by the Meta data and displaying the overlay on the marked object for every frame. In order to achieve this several control classes have been designed and implemented. Figure 6 shows the core classes and its relationship using the UML notation. The reading of the PCR timing value that rebuilds the timeline is handled within the class “PcrValueFilter”. The MHP API provides methods, which return the relevant section i.e. the PCR timing values. Due to the fact that for every frame a PCR is generated a separate thread was implemented for this class. The class „TheComparer“ the heart of the synchronization mechanism compares the PCR timeline values with the PTS time values and triggers the overlay process. The latter one is in charge for the display of the overlay.

4. An Implementation example

4.1 The Client Application
In order to show the feasibility of the developed approach a client application with some degree of interactivity was developed [8]. As stated in chapter three the approach is generic thus any kind of client can be used as long as the section filtering mechanism is provided by the middleware of the client. We used a client with the Multimedia Home Platform (MHP) as middleware. The scope of MHP is a common Java-based API, which provides a platform independent interface between applications from different providers and the STB manufacturer [9]. MHP-Java applications are called Xlet applications, which consist of a set of Java classes that operate together. Java classes are stored in object carousels. This carousel cyclically transmits the data modules over a DVB network. The Meta data can either be transmitted as data carousels or provided over a standard Internet network. For our application the client Java application, the Meta Data and additional information are transferred by using the carousel mechanism.

The main purpose of the client application is

1) To show that an overlay is synchronized frame accurately to a moving object and

2) To Enables simple interaction upon this object.
4.2 Discussion of obtained results
At the first stage the robustness of synchronizing an overlay with a marked object was tested. Following several shots of the successful synchronization mechanism based of the tracked object in figure 1 will be discussed. If a shot allows object based interactivity then the viewer will be made aware by the ellipse (“shirt color”) at the bottom line. After pressing the blue colored button a reference to the object is provided (blue circle overlay of the shirt) also depicting that some additional interaction is possible (s. figure 7). The overlay moves with the marked object and by pressing the blue button again further information or interaction of the marked object is possible. Figure 8 shows the result after pressing the blue button again.

5. Summary
In this paper, we have presented a new generic framework for object based synchronization of MPEG-2 video with additional data. Thus the viewer is enabled to interact actively with the linear presentation model in a DVB environment. A new timeline using the PCR values of each frame was introduced and transferred as private sections. With the introduction of an additional time line and referencing the marked object to this time line it is even possible to track and synchronize several objects within one shot. For each object marked for interactivity there will be a Meta data file entry referring to the introduced time line. Displaying e.g. the overlay frame accurate and allowing further interactivity of several objects at the same shot is a matter of an efficient software implementation and a powerful Hardware of the client. Our work successfully demonstrated the possibility of adding interactivity on MPEG-2 coded video objects and is a valuable basis for future services addressing content provider and application developer. Currently usability tests upon real content samples (sport and documentation) are carried out.

The authors would like to thank the hr fernsehen in particular Manuel Meyer for the provision of the video material.

6. References