Authoring Interactive Mobile Services Using MPEG-7 and MPEG-4 LASeR

Albert Hofmann, Andreas Kriechbaum, Werner Bailer

Institute of Information Systems & Information Management
JOANNEUM RESEARCH Forschungsgesellschaft mbH
Steyrergasse 17, 8010 Graz, Austria
{firstName.lastName}@joanneum.at

Abstract: Mobile TV is becoming increasingly popular. As hand held devices are personal and used in a "lean-forward" mode, mobile devices are well suited for interactive TV. The porTiVity system provides authoring tools for mobile interactive content that allow to attach interactivity to moving objects in the video. We describe automatic and semi-automatic annotation tools that create content descriptions represented as MPEG-7 documents. The descriptions of moving objects are then transformed into MPEG-4 LASeR, which is a suitable lightweight format for visualisation and interactivity on the mobile device. We show the use of these authoring components in porTiVity in two example scenarios.

1 Introduction

1.1 Motivation

With the introduction of the mobile broadcast standard DVB-H [DVB04] mobile TV is becoming increasingly popular. The number of mobile TV users is increasing at an enormous rate and consumers are showing their interest to enjoy innovative mobile applications. Market analysis predicts growth rates of 100% and more[1] for the number of users in the coming years and within the last year the average daily time a user spent watching mobile TV has nearly doubled[2]. Mobile interactive TV can be one of these emerging applications. Results from previous interactive TV projects such as GMF4iTV [Car05] show that hand held devices are better suited for interactivity than classic TV sets as they are personal and used in "lean-forward" mode.

This paper presents results from the porTiVity project [Por] which aims to provide interaction with objects in video shown on mobile devices. The interactive experience for the end-user is achieved by simply clicking on highlighted objects overlaid on the video on the touch screen of the mobile device. This interaction triggers e.g. downloading of additional content from the web, participating in a quiz, in mobile network based games

[1] http://www.3gnewsroom.com/3g_news/apr_05/news_5811.shtml
[2] http://futurezone.orf.at/stories/1500388/

or purchasing a chosen product. In a live scenario such as a football game interaction on the soccer players or via menu buttons allows to retrieve content from a player's website, game statistics, watching replays of interesting events etc.

The efficient semantic annotation of audiovisual content is an important part of this workflow. MPEG-4 LASeR annotation tools supporting moving objects and interaction with those objects were not available when the project started. Relevant annotation tools for audiovisual content are among others M-OntoMat-Annotizer[3] developed by the aceMedia project or the IBM VideoAnnEx Tool[4]. However, these tools lack object redetection feature and automatic object tracking functionality.

We describe in this paper an authoring system for creating interactive TV content with interactivity based on moving objects. Section 2 describes the automatic and semi-automatic annotation of content described using MPEG-7 [MPE01]. In Section 3 we discuss how this description is transformed into a suitable representation for the mobile device, namely MPEG-4 LASeR [LAS06]. The integration of these authoring steps into the porTiVity system and the end-user experience in two scenarios is described in Section 4. Section 5 concludes this paper.

1.2 Authoring for mobile interactive TV

The architecture of the end-to-end platform for providing rich media interactive TV services for portable and mobile devices developed in the porTiVity project is shown in Figure 1 and described in Section 4 in more detail. In the authoring phase the main video content is analysed and annotated to link it to additional content (images, web pages, etc.) and the possible user interactions are defined. The tools support workflows for both offline and online scenarios. The result of the authoring phase is a scene description in MPEG-4 LASeR which is multiplexed into an MXF [MXF04] stream together with the main and additional content. MPEG-4 LASeR is a standard for rich media scene description on mobile and embedded devices. In our work it is used to link additional content to the main video stream which can be simple text, HTML pages, pictures, audio and video clips. In order to reduce the bandwidth and processing requirements it has been used as a more lightweight alternative to MHP [MHP07]. MXF is a container format defined by SMPTE that can hold various types of audiovisual essence as well as metadata, with capabilities for multiplexing.

In Figure 2 the workflow between the tools of the proposed Semantic Video Annotation Suite in a typical authoring chain and the metadata formats used on the interfaces are shown. Media-Analyze is an automatic video preprocessing tool (cf. Section 2.1). In the Semantic Video Annotation Tool (SVAT) the results from the automatic preprocessing are used to support the user in annotating the video by providing navigation aids, redetection and tracking of objects. The Interactive Application Tool (IAT) uses the MPEG-7 de-

[3] `http://www.acemedia.org/aceMedia/results/software/m-ontomat-annotizer.html`

[4] `http://www.research.ibm.com/VideoAnnEx`

scription of the annotated objects generated by the SVAT, transforms them into MPEG-4 LASeR objects, performs the LASeR segmentation and packs all the required data into an MXF file suitable for the playout system.

2 Content annotation

The content annotation is divided into two steps: an automatic preprocessing step done with the Media-Analyze tool and a semi-automatic annotation step performed with the Semantic Video Annotation Tool (SVAT). Both tools are bundled together in the Semantic Video Annotation Suite [SVA] and are described here in the context of authoring interactive mobile TV services. The output of both steps is described using MPEG-7, a comprehensive standard for multimedia content description covering both low-level and high-level description. In particular, the Detailed Audiovisual Profile (DAVP) [BS06], an MPEG-7 subset aimed at fine-grained description of audiovisual content in media production and archiving, is used.

2.1 Automatic preprocessing

In a first step a video file is automatically analysed by the Media-Analyze tool. We extract metadata for efficient navigation in videos and structuring the content. These are shot boundaries, key-frames, stripe images and camera motion.

In order to aid the user in efficient object annotation we extract SIFT descriptors [Low99] of interest points of key-frames and build a visual vocabulary [SZ03]. By doing this it is possible to redetect similar objects and shots in the same video in a very efficient way [NTM07, RBN^{+}07].

After identifying single occurrences by the user these occurrences need to be tracked within the same shot. Since this step is interactive it requires a very efficient tracking algorithm. For this purpose feature points and descriptors are extracted within this automatic preprocessing step [TM07, NTM07].

Both the visual vocabulary of SIFT descriptors and the descriptors for object tracking are stored in a proprietary binary format for efficiency reasons.

2.2 Semi-automatic object annotation

After this automatic annotation step objects of interest can be annotated with interaction. This is done with the Semantic Video Annotation Tool and its support for spatio-temporal object annotation.

As operators often wish to assign the same additional content to similar objects we provide

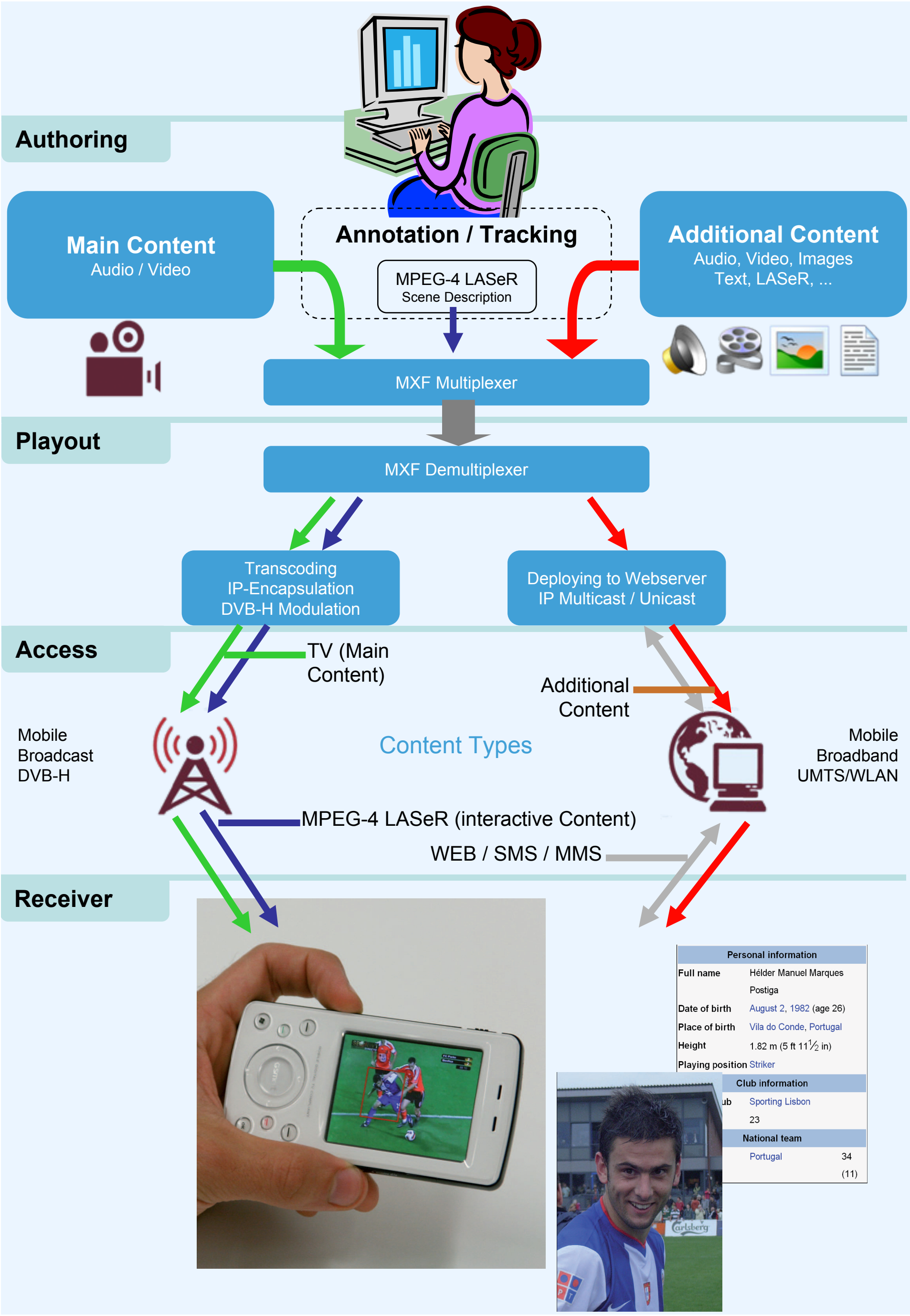

Figure 1: porTiVity system overview.

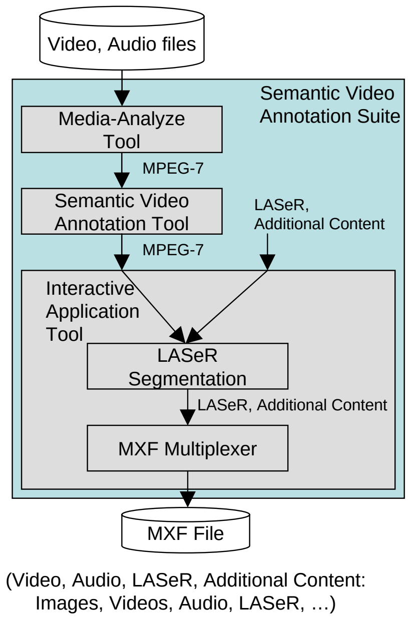

Figure 2: Annotation chain in the Semantic Video Annotation Suite including exchanged metadata formats.

an object redetection feature. The input for the object redetection is a region of interest identified in a video player component. This can be done by simply marking a bounding box around the region, or by using a color segmentation functionality in order to get the exact border of the object. By using the visual vocabulary from the preprocessing step we can search for appearances of similar SIFT descriptors as extracted from the query region. For this application scenario higher recall is preferred since false positives in the result list can be easily deleted.

In the next step these single occurrences can be tracked over time within the boundaries of the shot. The input for the tracking algorithm can be a bounding box as well as a complex polygon around an object. The resulting moving region is always described as bounding box. The tool also supports manual tracking in case the automatic tracking fails. In this case the operator specifies key-regions over time that will be connected with a spline interpolation algorithm.

The output of the Semantic Video Annotation Tool is an MPEG-7 description which includes the metadata from the automatic analysis step enriched with the annotation of moving objects. An example of such a moving region is shown in Listing 1. The moving region shows an object with name *bottle* (*SemanticRef* element) and a bounding box that changes over time (*ParameterTrajectory* elements as defined by [MPE01, part 3]). The values inside the *Coords* element define the three points that build a closed bounding box.

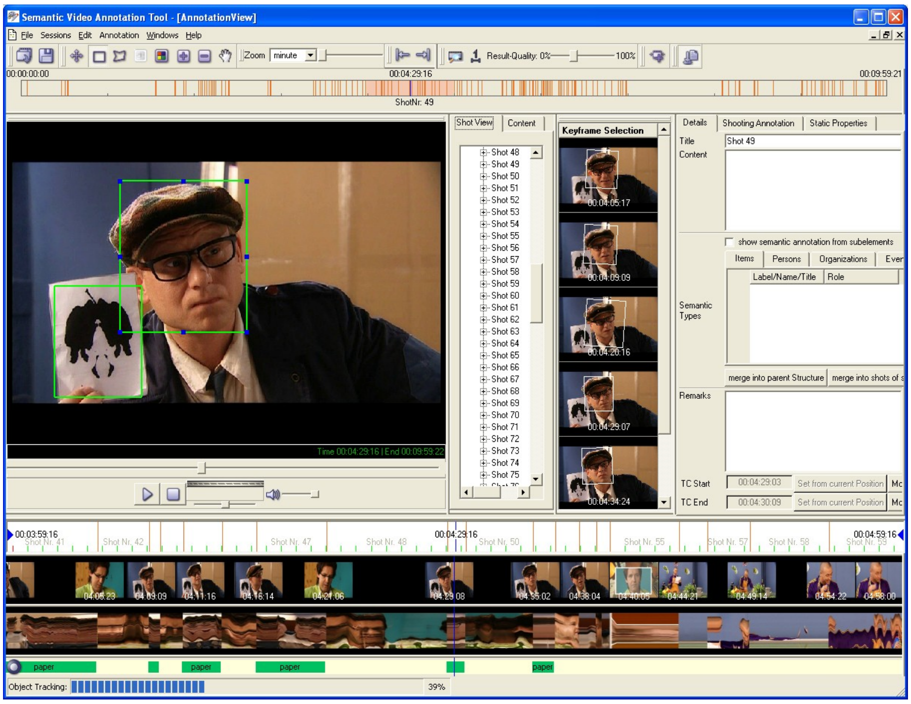

Figure 3: Semantic Video Annotation Tool for object annotation.

3 MPEG-7 to MPEG-4 LASeR transformation

3.1 Transforming moving object descriptions

In this step the representation of moving regions described in MPEG-7 format are transformed to MPEG-4 LASeR, a suitable delivery format for the mobile device. MPEG-7 is intended for content description, but does not provide visualisation and interaction possibilities. In addition, a large part of the MPEG-7 description is not needed at the mobile device. We present an approach for transforming shape descriptions of moving objects (like polygons and rectangles) to MPEG-4 LASeR.

MPEG-4 LASeR is based on SVG[5] Tiny 1.2 and is thus very powerful regarding shape representation. We have focused on transforming polygons and simple bounding boxes to SVG taking different personalisation options for the end user in consideration. Currently three options are supported which are also reflected in the MPEG-4 LASeR snippet in Listing 2 and the images in Figure 4: a rectangle with solid or dashed line style, a contact point in the middle of the area and rectangle with only corner indicators.

[5]Scalable Vector Graphics (SVG) is a W3C recommendation for the XML based representation of 2D vector graphics (`http://www.w3.org/Graphics/SVG/`).

Listing 1: Moving object described in MPEG-7.

```
<MovingRegion>
  <SemanticRef idref="bottle" />
  <SpatioTemporalLocator>
    <ParameterTrajectory motionModel="still">
      <MediaTime>
        <MediaTimePoint>T00:01:23:1F25</MediaTimePoint>
        <MediaDuration>P0DT0H0M0S2N25F</MediaDuration>
      </MediaTime>
      <InitialRegion>
        <Polygon>
          <Coords dim="2␣4">421 96 0 -96 300 0 151 0</Coords>
        </Polygon>
      </InitialRegion>
    </ParameterTrajectory>
    <ParameterTrajectory motionModel="still">
      <MediaTime>
        <MediaTimePoint>T00:01:23:3F25</MediaTimePoint>
        <MediaDuration>P0DT0H0M0S1N25F</MediaDuration>
      </MediaTime>
      <InitialRegion>
        <Polygon>
          <Coords dim="2␣4">422 95 0 -95 301 0 151 0</Coords>
        </Polygon>
      </InitialRegion>
    </ParameterTrajectory>
    ...
  </SpatioTemporalLocator>
</MovingRegion>
```

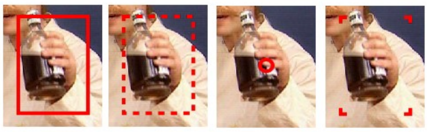

Figure 4: Styles for highlighting moving objects.

For the SVG description of the animation of the moving object over time we considered two options: specifying the coordinates in a fixed interval for the whole time range, or using a variable interval depending on the movement of the object and specifying those time points in the *keyTimes* attribute. Using a variable interval has advantages in terms of description size for moving objects that hardly move or keep their position for a longer period, but increases the complexity of the synchronised broadcast (see LASeR Segmentation in Section 3.2). So we have chosen the approach using a fixed interval (e.g. 0.2 sec) that can be configured by the user.

The result of the conversion from the moving region described in the MPEG-7 snippet in Listing 1 to MPEG-4 LASeR is shown in Listing 2. The id from the MPEG-7 description is reflected in the *title* element (unchanged) and also in the *id* attribute, taking into account that XML requires unique ids. Also when accessing specific elements later during MPEG-4 LASeR command processing we require unique ids when adding, modify or removing objects.

The graphic object is embedded inside the SVG document of the main MPEG-4 LASeR scene. In order to reduce the complexity of MPEG-4 LASeR handling within the authoring tools, no LASeR commands are used in this authoring step for time related updates of the scene. This is done in a separate post processing step – the LASeR segmentation – which is described in Section 3.2. The animation of objects over time is specified for the whole object time range. Also the visibility of the object is modeled by setting the SVG *display* attribute to 'none' and 'block' for the respective time points. So up to this step the produced MPEG-4 LASeR document just contains one scene with one SVG document containing the whole description for the whole video time range. Also no interaction possibilities are added up to this point.

Listing 2: Moving object described in MPEG-4 LASeR.

```
<g pointer-events="all" fill="none" stroke="red" id="bottle0" stroke-width="2">
  <title>bottle</title>
  <desc>Moving Object</desc>
  <rect width="96" x="421" y="300" height="151">
    <animate values="421;426;432;439;447;..."
      fill="freeze" begin="83.040s" dur="9.680s" attributeName="x" />
    <animate values="300;305;302;277;229;192;..."
      fill="freeze" begin="83.040s" dur="9.680s" attributeName="y" />
    <animate values="96;94;93;95;99;..."
      fill="freeze"begin="83.040s" dur="9.680s" attributeName="width" />
    <animate values="151;149;148;149;153;..."
      fill="freeze" begin="83.040s" dur="9.680s" attributeName="height" />
  </rect>
  <circle cx="469" cy="375" display="none" r="10">
    <animate values="375;379;376;351;305;..."
      fill="freeze" begin="83.040s" dur="9.680s" attributeName="cy" />
    <animate values="469;473;478;486;496;..."
      fill="freeze" begin="83.040s" dur="9.680s" attributeName="cx" />
  </circle>
  <g display="none">
    <desc>corner rectangle</desc>
    <polyline points="421,310␣421,300␣431,300">
      <animate values="421,310␣421,300␣431,300;426,315␣426,305␣436,305;..."
        fill="freeze" begin="83.040s" dur="9.680s" attributeName="points" />
    </polyline>
    <polyline points="507,300␣517,300␣517,310">
      <animate values="507,300␣517,300␣517,310;510,305␣520,305␣520,315;..."
        fill="freeze" begin="83.040s" dur="9.680s" attributeName="points" />
    </polyline>
    <polyline points="517,441␣517,451␣507,451">
      <animate values="517,441␣517,451␣507,451;520,444␣520,454␣510,454;..."
        fill="freeze" begin="83.040s" dur="9.680s" attributeName="points" />
    </polyline>
    <polyline points="431,451␣421,451␣421,441">
      <animate values="431,451␣421,451␣421,441;436,454␣426,454␣426,444;..."
        fill="freeze" begin="83.040s" dur="9.680s" attributeName="points" />
    </polyline>
  </g>
  <set begin="0s" attributeType="XML" to="none" attributeName="display" />
  <set begin="83.040s" attributeType="XML" to="block" attributeName="display" />
  <set begin="92.720s" attributeType="XML" to="none" attributeName="display" />
</g>
```

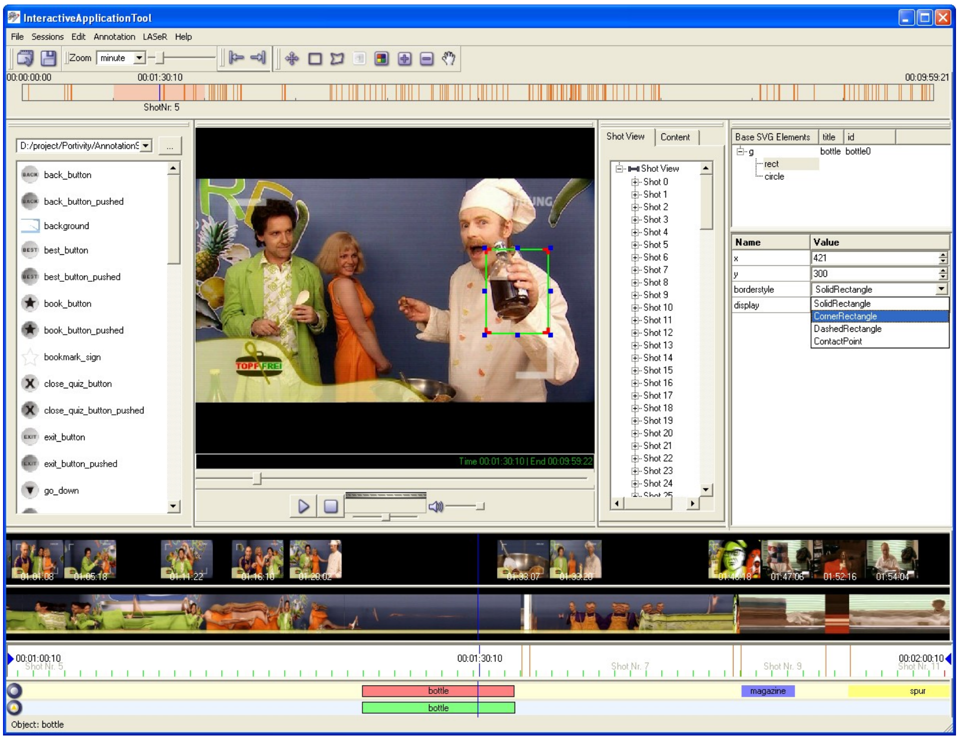

Figure 5: Interactive Application Tool for creating MPEG-4 LASeR scenes.

3.2 Adding interactivity

Next to the powerful shape representation capabilities of MPEG-4 LASeR, it provides support for interactivity to the end user. In our scenario we link additional content to moving objects like statistics and photos of soccer players, background information on actors etc. If the user clicks on a moving region this additional content should be displayed. Moreover, the additional content can be personalised, which means that different additional content items can be linked for different target groups (e.g. fans of the different teams in sports broadcasts, male and female viewers, . . .). We can edit these actions in the Interactive Application Tool (IAT, see Figure 5). The resulting MPEG-4 LASeR snippet after adding an additional content is shown in listing 3. For non personalised additional content we place an *a* element around the shapes for highlighting the moving object. For personalised additional content (not shown in the listing) we use a script function that evaluates the additional content link based on the local user profile and makes the respective HTTP request.

Before sending the MPEG-4 LASeR description to the playout system, a post processing step is needed to make the service ready for transmitting over a broadcast channel. Users can join the TV channel any time, so it is necessary to transmit the LASeR content period-

Listing 3: Interactivity described in MPEG-4 LASeR.

```
<g pointer-events="all" fill="none" stroke="red" id="bottle0" stroke-width="2">
  <title>bottle</title>
  <desc>Moving Object</desc>
  <a xlink:href="http://server.portivity.org/PoE1_a.xsr">
    <rect width="96" x="421" y="300" height="151">
      <!-- animation of rect -->
    </rect>
    <circle cx="469" cy="375" display="none" r="10">
      <!-- animation of circle -->
    </circle>
    <g display="none">
      <desc>corner rectangle</desc>
      <!-- polylines of corner rectangle -->
    </g>
        </a>
</g>
```

ically. This is required for the general scene description containing the static objects (like menus), while moving objects just need to be transmitted during the time they are visible. This is done by the LASeR segmentation module. It takes the whole LASeR scene from the Interactive Application Tool as input, and splits the objects into different access units. An example of how the MPEG-4 LASeR snippet presented in Listing 2 is segmented is shown in Listing 4:

Listing 4: LASeR Segmentation into Access Units.

```
<!-- Initial object description in the main SVG document: -->

<g pointer-events="all" fill="none" stroke="red" id="bottle0" stroke-width="2">
  <!-- shape representation cut here -->
  <set begin="0s" attributeType="XML" to="none" attributeName="display" />
  <set begin="83.040s" attributeType="XML" to="block" attributeName="display" />
  <set begin="92.720s" attributeType="XML" to="none" attributeName="display" />
</g>

<!-- After the segmentation into access units for insert and delete: -->

<saf:sceneUnit time="83040">
  <lsr:Insert ref="parent_id">
    <g id="bottle0" />
  </lsr:Insert>
</saf:sceneUnit>

<saf:sceneUnit time="92720">
  <lsr:Delete ref="bottle0"/>
</saf:sceneUnit>
```

Another task of this module is the periodic generation of RefreshScenes, which are access units containing a refresh of the scene as it should be at that precise moment. These access units are generated at a given interval – usually every two seconds – and they are ignored by the terminal if it already has this information (either from a NewScene or from a previous RefreshScene). These access units will contain the information on static and dynamic objects that become visible or invisible, and will also update the position of visible dynamic objects. The additional binary MPEG-4 LASeR stream only costs a small amount of bandwidth compared to the TV stream. An alternative system design to periodically poll a web server would lead to a traffic bottleneck due to the huge amount of requests to be handled by the web server.

The final step of the authoring is the MXF multiplexing. The video and audio content as well as all additional content files are packed into an MXF stream to ensure time synchronisation.

4 porTiVity system

A detailed description of the porTiVity system is given in [DFL$^+$08]. This section summarises the steps after the authoring as shown in Figure 1.

4.1 Playout system

The playout system consists of several modules making the video stream and MPEG-4 LASeR content ready for transmission over DVB-H and the additional content accessible on a web server.

The MXF demultiplexer takes the produced MXF file from the authoring system as input, extracts all tracks and forwards them to the respective modules in the chain. The transcoder module converts the high-quality video and audio streams used in the production environment to the target formats on the mobile device: MPEG-4 AVC (H.264) and MPEG-4 AAC. Additional content files are also transcoded to the target format if necessary. The Metadata Processor Module receives LASeR access units from the MXF demultiplexer, encapsulates them over RTP (Real-Time Transport Protocol), and sends them to the DVB-H module responsible for broadcasting.

4.2 Receiver

The mobile device receives the DVB-H stream consisting of an audio, video and LASeR MPEG-4 elementary stream encapsulated in RTP. These streams are synchronised on the terminal by RTCP.

The software responsible for rendering the porTiVity service is based on the Osmo player of the open source multimedia framework GPAC [FCM07]. An implementation with the full support of the porTiVity features is available for Windows Mobile version 5 and 6, Windows XP and an experimental version for Symbian OS. The developed scenarios work on normal off-the-shelf mobile phones with DVB-H receivers and touch screens. Furthermore any kind of web access is required for downloading the additional content files requested by the user clicking on the annotated objects. For various demonstrations we used terminals with limited processing power like a Gigabyte GSmart T600, a QTec 9000 and an LG KS20. As a high-end terminal an HTC Shift and a Samsung Q1 Ultra were used.

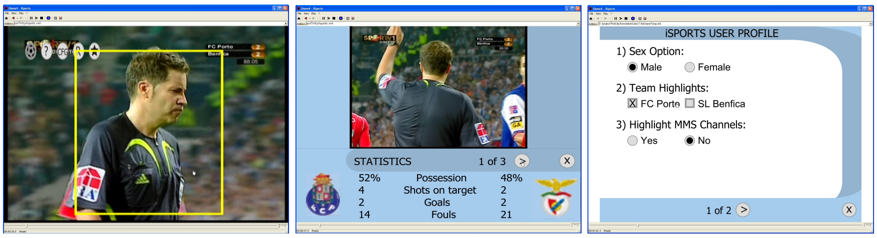

Figure 6: Interactive sports scenario.

4.3 End-user experience

iSports – interactive sports scenario

The iSports scenario (see Figure 6) demonstrates how interactivity can be applied to live sport events. For this case a separate annotation tool – the Live Annotation Tool [DFK$^+$09] – has been developed in the porTiVity project. With this tool it is possible to define objects of interest and link additional content, track the objects in realtime and directly send the results to the playout system – with an overall delay of 5 seconds to the input stream.

Next to the interaction possibility with the highlighted objects, a menu with buttons is shown in the top left corner. The menu, which can be collapsed in order not to consume valuable display space, allows to access replays of interesting events like goals, to access a help page that explains the available features of the service and to configure personalisation settings.

Spur & Partner – interactive crime series for children

The Spur & Partner scenario (see Figure 7) is based on a successful interactive TV crime series for children produced by the public German broadcaster ARD. Users can interactively collect hints by clicking on the objects annotated by the producer. They also can answer questions referring to those pieces of evidence and receive points for correct answers, which gives the TV program a little game characteristic.

The end-users can also access background information on actors and learn from an interactive detective magazine more about the approach that the detective is using to identify the culprit. And finally at the end of the series the users have the chance to name the culprit by themselves and directly transmit their results.

Spur & Partner was evaluated by Rundfunk Berlin Brandenburg (RBB) with 6 children at the age of 9 to 11 years. They had normal computer skills and were familiar with the original TV series and the interactivity by its MHP application. Evaluation results have shown that recognizing the bounding box of the moving objects and further interaction with the additional content improve with the experience and are no problem after a little trying.

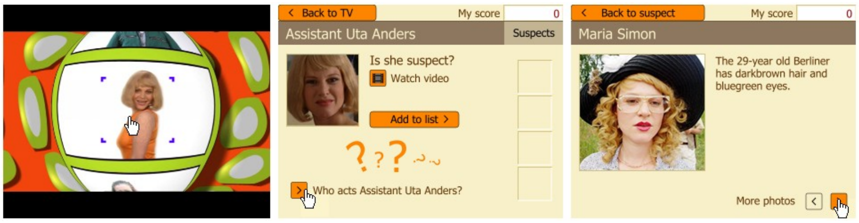

Figure 7: Interactive detective story Spur & Partner.

4.4 Evaluation of the authoring tool

The authoring system was evaluated by RBB with professional users from various departments. An evaluation of the tools has shown that the time required to annotate objects are realistic and acceptable in a TV production environment. The system was tested on the interactive detective story with a video length of 8:40 minutes and consisting of 16 interactive elements. Not considering the automatic preprocessing step, the whole annotation process took 1:15 hours with the additional content already prepared. Missing shortcuts in the evaluation version for common task could further increase the annotation performance.

5 Conclusion

This work has shown that MPEG-7 is suitable as description format for video including the description of moving objects. We presented tools that aid an operator in efficient object annotation by using automatically extracted metadata in MPEG-7 format and enrich it with annotations for moving objects. On the delivery side, MPEG-4 LASeR has proved to be a powerful standard for enabling rich multimedia interactive services on mobile devices with limited processing power. Due to the fact that MPEG-4 LASeR is based on SVG it allows scalability for a wide range of target devices. The chosen approach to use existing annotation tools based on MPEG-7 and adding MPEG-4 LASeR functionality combined the advantages of both standards and resulted in a full featured MPEG-4 LASeR authoring suite.

Acknowledgements

The authors would like to thank their colleagues at the Institute of Information Systems & Information Management at JOANNEUM RESEARCH, especially Helmut Neuschmied, Georg Thallinger and Werner Haas, as well as the other partners of the porTiVity project for the good collaboration.

The research leading to this paper has been partially supported by the European Commission under the contracts FP6-027098, “porTiVity – Portable Interactivity” (http://www.portivity.org), FP6-045032, “SEMEDIA - Search Environments for Media” (http://www.semedia.org) and FP7-231161, “PrestoPRIME”.

References

[BS06] Werner Bailer and Peter Schallauer. The Detailed Audiovisual Profile: Enabling Interoperability between MPEG-7 Based Systems. In Huamin Feng, Shiqiang Yang, and Yueting Zhuang, editors, *Proceedings of 12th International Multi-Media Modeling Conference*, pages 217–224, Beijing, CN, Jan. 2006.

[Car05] B. Cardoso. Hyperlinked Video with Moving Objects in Digital Television. In *International Conference on Multimedia and Expo*, pages 486–489, 2005.

[DFK+09] J. Deigmöller, G. Fernandez, A. Kriechbaum, A. Lopez, B. Mérialdo, H. Neuschmied, F. Pinyol Margalef, R. Trichet, P. Wolf, R. Salgado, and F. Milagaia. Active Objects in Interactive Mobile Television. In *15th International Conference on MultiMedia Modeling*, Sophia-Antipolis, FR, Jan. 2009.

[DFL+08] J. Deigmöller, G. Fernandez, A. Lopez, B. Mérialdo, H. Neuschmied, F. Pinyol, R. Trichet, P. Wolf, R. Salgado, F. Milagaia, S. Glaser, and A. Duffy. Portivity : Object Based Interactive Mobile TV System Networked and Electronic Media. In *NEM SUMMIT 2008, Networked and Electronic Media, October 13-15, 2008, Saint-Malo, France*, Oct 2008.

[DVB04] Transmission System for Handheld Terminals (DVB-H). ETSI EN 302 304, Jun. 2004.

[FCM07] Jean Le Feuvre, Cyril Concolato, and Jean-Claude Moissinac. GPAC: open source multimedia framework. In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, pages 1009–1012, New York, NY, USA, 2007. ACM.

[LAS06] MPEG-4 LASeR. ISO/IEC 14496-20, 2006.

[Low99] David G. Lowe. Object Recognition from Local Scale-Invariant Features. In *Proceedings of the International Conference on Computer Vision*, pages 1150–1157, 1999.

[MHP07] MHP 1.2. ETSI 102 590, 2007.

[MPE01] Information Technology—Multimedia Content Description Interface. ISO/IEC standard 15938, 2001.

[MXF04] Material Exchange Format (MXF) – File Format Specification (Standard). SMPTE 377M, 2004.

[NTM07] Helmut Neuschmied, Rémi Trichet, and Bernard Mérialdo. Fast annotation of video objects for interactive TV. In *MM 2007, 15th international ACM conference on multimedia, September 24-29, 2007, Augsburg, Germany*, Sep 2007.

[Por] porTiVity web page. http://www.portivity.org.

[RBN+07] Herwig Rehatschek, Werner Bailer, Helmut Neuschmied, Sandra Ober, and Horst Bischof. A Tool Supporting Annotation and Analysis of Videos. In Stefanie Knauss and Alexander D. Ornella, editors, *Reconfigurations. Interdisciplinary Perspectives on Religion in a Post-Secular Society*, pages 253–268. LIT, Vienna, 2007.

[SVA] Semantic Video Annotation Suite. http://www.joanneum.at/en/fb2/iis/products-solutions-services/semantic-video-annotation.html.

[SZ03] J. Sivic and A. Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1470–1477, October 2003.

[TM07] Rémi Trichet and Bernard Mérialdo. Generic object tracking for fast video annotation. In *VISAPP 2007, 2nd International Conference on Computer Vision Theory and Applications, 8 - 11 March, 2007 Barcelona, Spain*, Mar 2007.