MHP Oriented Interactive Augmented Reality System for Sports Broadcasting Environments

Igor G. Olaizola*, Iñigo Barandiaran Martirena* and Tobias D. Kammann†
*VICOMTech (Visual Communication Technologies) Spain
†Uni-Koblenz Germany
email: iolaizola[at]vicomtech.es
ibarandiaran[at]vicomtech.es
kammann[at]augmented.org

Abstract

Television and movie images have been altered ever since it was technically possible. Nowadays embedding advertisements, or incorporating text and graphics in TV scenes, are common practice, but they can not be considered as integrated part of the scene. The introduction of new services for interactive augmented television is discussed in this paper. We analyse the main aspects related with the whole chain of augmented reality production.

Interactivity is one of the most important added values of the digital television: This paper aims to break the model where all TV viewers receive the same final image. Thus, we introduce and discuss the new concept of interactive augmented television, i. e. real time composition of video and computer graphics - e.g. a real scene and freely selectable images or spatial rendered objects - edited and customized by the end user within the context of the user's set top box and TV receiver.

We demonstrate a sample application introducing "Interactive Augmented Television" for sport broadcasts additionally with 3D virtual objects in order to enhance or alter the presentation of the match with a new interface. We also introduce a pure virtual world where the user can select the camera position.

Digital Peer Publishing Licence
Any party may pass on this Work by electronic means and make it available for download under the terms and conditions of the current version of the Digital Peer Publishing Licence (DPPL). The text of the licence may be accessed and retrieved via Internet at
http://www.dipp.nrw.de/.

First presented at the 4th European Interactive TV Conference EuroITV 2006, extended and revised for JVRB

Keywords: Interactive Television (ITV), Augmented Reality (AR), Multimedia Home Platform (MHP), Digital Video Broadcasting (DVB), Virtual Reality (VR), User Interfaces, Customization.

1 Introduction

Ever since television signals exist, different methods were proposed to enrich the quality of the broadcast image. Thus, special effects have been broadly used to show virtual worlds that either do not exist or would be to difficult to get. Current technologies allow depicting virtual elements in real time where the same difficulties appeared for special effects must be solved in real time. The main problems are tracking, segmentation, 3D registering and rendering.

Chroma Keying environments have been typically used to segment the real objects and insert them in a virtual scenario. It provides a very easy way for segmentation but only can be used in controlled environments where the background has a predefined colour and the lighting is controlled. In sport broadcast where the scenario does not accomplish these requirements other techniques must be used.

Most of the TV programs compose the real images with 2D elements like banners, texts, pictures, etc. There is no relation between the scene and these objects which overlay the real picture. They are used for example when presenting additional information below a news report or adding commercials inside a sport show. These productions still have one major limitation in common: the character of a priori uneditable and uncustomizable image for the viewer.

Although there are some augmented reality applications in sport broadcasts (American football, hockey, Spanish soccer broadcast, etc.) they use some additional input information like inertial sensors on cameras or transmitters within a puck. These are very expensive setups which could be avoided getting all the information needed for the image analysis of the video provided by the broadcast cameras.

As the global releases of movies and shows increase, the need for an advance in technology to specifically customize these broadcasts also increases. A method to exchange video elements in live while broadcasting is needed. These elements can be plain images or 3D objects. By knowing the camera's position and orientation through tracking, these elements can be inserted into the real scene, producing traditional augmented television broadcast. In this paper we describe a methodology to add interactivity by granting the viewer with full control over the editing process of the video material - possibly resulting in a perfect blend of TV and interactive applications or games. An application is implemented using the television standard of Multimedia Home Platform (MHP) [Pla06] running alongside the Digital Video Broadcasting (DVB) [Sta06]. This example demonstrates the concept of augmented interactive television and shows restrictions and pitfalls that occurred during the development of our application using currently available technologies.

2 Concept Design

Our sample application deals with the presentation of the traditional Basque ball sport pelota. It is played with two or four players inside a court and is typically filmed by two or more cameras from the right side (see Figure 1). The left and front walls are usually green and it is common practice to hang sponsoring posters or banners. These posters cannot be changed during the match and they cover only the lower part of the wall. Putting up virtual banners we could change the content and the position of the advertisement at our free will.

A tracking of cameras is needed and a 3D rendering has to generate the augmentation overlay. Furthermore, players moving in front of exchanged objects have to be masked to still appear in front of the virtual objects.

The video material is transmitted via DVB and the presentation should be done by set top boxes and the Java-based interactive TV standard of MHP. MHP data are broadcast multiplexed with the DVB audiovisual contents. Additionally, a complete virtual 3D world allows observation of the sport game from any angle and at the push of a button the viewer may switch to a real camera point of view. This option makes it possible to render points of view which would be impossible with real cameras. For instance, a camera would be placed just behind the ball or on the head of one player.

Figure 1: Pelota game snapshot

3 Technical Implementation

3.1 Introduction

Digital Television allows the delivery of application data together with audio and video contents, which makes it possible to provide interactive services using the TV sets. Interactive digital television (iDTV) seems to be a very promising technology, providing a large range of new services to the TV user population. While several studies have predicted an explosion of the iDTV market [Sri02], the actual development of valuable applications presents several challenges which still need to be overcome.

Theoretically, digital TV offers a new platform for services which are currently supported by PC environments. However, the underlying technology differs in ways which greatly influence the iDTV applications design strategy. Moreover, iDTV applications target a far more diverse user population, whose demographics, skills and goals significantly differ from those of computer users.

As a consequence, the applications designed to be displayed in TV sets cannot be directly ported from PC-oriented designs. When developing an application for digital TV, one of the most important tasks is thus to identify the requirements and constraints of iDTV environments. Augmented Reality demands high hardware capabilities making difficult the reception and rendering in commercial set-top boxes which in most of the cases have low processing and memory features.

3.2 Hardware

Digital TV represents fundamentally new technology from the computer [Ben02]. At the heart of iDTV is the set-top box. The primary purpose of this box is to convert the digital signal it receives (via satellite, terrestrial broadcast or cable) into video and audio to play through the TV set [O'D00].

Typical commercial set-top boxes do not have any graphic card and most of them use the microprocessor to decode the MPEG-2 stream by software in order to avoid the using of DSPs[1] and get lower prices for the devices. The real-time operating system of the set-top box will assign a low priority level to the MHP processes and the hardware features (up to 200MHz and 64MB RAM in the best commercial set-top boxes in 2005) are far from current computers' state of the art. Due to these limitations, commercial MHP set top boxes are not able to fulfil our system's requirements.

Furthermore, a TV set presents several differences to a computer monitor, which implies some rethinking of the interface design for iDTV [Poy03]: bigger fonts, simpler graphics and clear colours need to be used.

In this project, we focus on Phase Alternating Line (PAL) TV sets, as being the dominant European television standard. We choose to adopt an architecture based on Digital Terrestrial Television (DTT) network using standard DVB-MHP compliant set-top boxes.

3.3 MHP, Java Environment

MHP is a middleware which allows interactivity on television. It has been adopted by the DVB consortium as the standard for interactivity and is currently being used in Europe as the main middleware for DTT. Countries like Italy have sold about 2.2 millions of MHP set-top boxes (May 2005).

MHP middleware typically uses a slim version of Sun's Java programming language to execute applications. This can be the older personalJava (pJava) or the Java Micro Edition (JME[2]).

A 3D renderer is needed to show the virtual objects on the TV set and to mix them with the real images. Currently there are no vailable 3D renderers within the mentioned Java environments used in commercial set-top boxes. For this purpose, we have implemented the components with 3D contents in a PC based system where these limitations have been overcome using Java libraries that at this moment are not included in the MHP specification but could be easily added to the standard in the future.

We opted for a renderer called Xith3D [Xit06]. It is an open source 3D API for Java, including a scenegraph, and at the same time allowing full access to the OpenGL[3] state machine. The scenegraph brings us a high abstraction level and the code can be optimized accessing directly to the OpenGL instruction set.

Some tests have been done using other Java 3D renderers like Anfy[4], but even though they run on a MPH set-top box, the performance obtained was too far from real-time behaviour. For instance, we tried to rotate three simple 3D cube representations using commercial set-top boxes. We used the same application in Humax, Samsung, Philips and ADB devices. All of them were too far from real time behaviour. Even a more advanced ADB development set-top box (166MHz CPU, 72MB RAM) was too slow for real

[1] Digital Signal Processor, a microprocessor with a special instruction set and architecture, totally oriented to the signal processing. They are more efficient for some special arithmetical operations like MAC (Multiply-accumulate) and more expensive than general purpose microprocessors

[2] https://jme.dev.java.net/

[3] Open Graphics Library, a cross-language cross-platform specification for 3D computer graphics production (http://www.opengl.org/).

[4] http://www.anfyteam.com/

time purposes. Although some code optimizations could be done, the performance is too poor to run efficiently with more complex 3d models.

Java3D seems as the best choice, offering the broadest support and the most add-ons, but, due to still missing features regarding rastered images and the inability of OpenGL-state access, we decided the using of Xith3D API.

The usage of the JME in combination with the Mobile 3D Graphics API (M3G) has been inspected as well, and it seems to fulfil our above mentioned requirements. The M3G is an optional package for the JME offering 3D graphic capabilities. The main target lies on implementation for devices with very little and restricted calculation and memory power, such as mobile devices and handhelds. The renderer does not resort to hardware acceleration allowing usage in low-budget environments. However, the API scales up to higher-end devices, featuring bigger colour displays, floating point unit abilities or even 3D graphics chips support.

Up to now, the MHP standard does not include the M3G API, but if in the future it would be integrated in MHP set-top boxes, the whole system would be easily ported to M3G

For video display we resort to the Java Media Framework (JMF[5]), which is one of the two ways for controlling the display of video signals and to choose which signal to present [Web06]. The Java Media Framework allows further control over streamed media and DVB-API classes for event handling or persistent storage of settings and other data.

The MHP stack is formed by different APIs. The MHP 1.1 allows to add new plug-ins to provide customized capabilities to the system (Figure 2). Although, currently there are no MHP 1.1 compatible set-top boxes in the market, this extendibility could be used in the future to add the needed software features to the system.

3.4 Interactivity and Return Channel

Any type of interactivity needs a bidirectional channel. A user can interact with the set-top box using the remote control and the TV set as channels. This local interactivity has some limitations which only can overcome using the return channel. When the interactivity level must be extended to the application source, the broadcast channel can not be used due to its unidirectional being.

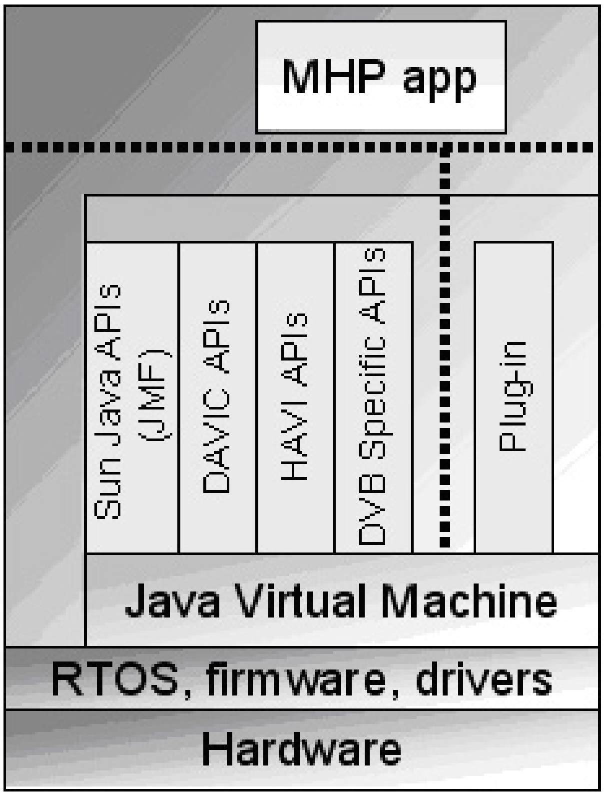

Figure 2: MHP stack

The return channel introduces a bidirectional unicast connection into the TV world, which up to now, has been totally broadcast oriented. Nowadays, most of the commercial set-top boxes use V90 modems to get the connectivity. Although the available bandwidth is not enough to allow personalized video stream provision, it can be used to load personalized textures and models. Furthermore, the future improvement of the return channel bandwidth will allow client-server architecture where the rendering could be done by the server or where the videos could be provided *on demand*.

The union of the broadcast with the return channel results in a new kind of services where the limitations of each of them can be overcome using this combination as a new communication method. These limitations are:

- Broadcast: The same content for all people. There is no possibility to send personalized contents to each user. Furthermore, users can not upload information since the broadcast channel is an unidirectional transmission way.

[5] http://java.sun.com/products/java-media/jmf/

- Return Channel: The bitrate of the return channel depends on external network conditions, number of users connected, type of return channel (modem, DSL, DOCSIS[6], etc.) A large number of connections with the same content (broadcast) requires very expensive infrastructures.

Thus, the broadcast is the best way to send information to millions of receivers without any network congestion limitations. The return channel can establish a point-to-point connection adding personalized data to each of the receivers running at this moment.

In our application we used the broadcast channel to transmit the application itself and the common data. The return channel is used to load personalized textures and to get information about the users habits. The return channel could also be used for betting, voting, buying, etc.

3.5 Video Overlay and Alignment

MHP receivers distinguish between background, video and graphics layer. A 100% fitting alignment between these different parts is not guaranteed. Thus, we can not draw the video inside the video layer and overlay graphics into a separate one. Instead, all rendering has to be done within the graphics layer (violating today's mandatory features of set-top boxes): video display resorting to the JMF inside the graphics layer is only offered optionally, but here we have full control over placement. Inside the 3D world we render the JMF video to a textured quad, positioned in front of the camera in a perpendicular angle and at fixed distance. Camera movements through real 3D space can be mapped directly to a virtual world using the same scaling. Placed virtual objects inside this virtual world will appear at their corresponding position in the real court.

The augmented data can include simple image files (JPEG, PNG) as well as more complex 3D geometry. A file loader for ASCII-encoded geometry is integrated (ASE, ASCII Scene Exporter). Commercial 3D rendering products usually include an exporter or offer plug-ins to generate these files. Textures and UV-alignment are supported as well.

To realise the augmentation, we pass pre-calculated tracking data for the real camera position to the receiver. The camera of the virtual world will be repositioned frame by frame and the rendered image can be put on top of the video with fitting perspective.

Synchronization of streamed media and our renderer remains difficult in a set-top box environment: using data carousel transmission in conjunction with NPT events in DVB, small offsets are still noticeable, causing offsets in time of +/- 5 fps [LGF$^+$03], which is still too high for fast camera movements. The overlay would fail. We propose encoding of tracking data inside the DVB-MPEG-stream to get better results. Private MPEG fields can be used to store additional or they could be stored in dynamic MHP files or sent as stream events. But current set-top boxes do not allow direct access to the MPEG-2 TS data: we can not retrieve our encoded tracking information with the JMF 1.x. we could only rely to a simulation using the JMF in version 2.

3.6 Virtual World Representation and Camera Selection

Digital TV allows transmission of so called bouquets for a single program: more than one video stream might be available, for instance offering different points of view for the same show. These streams are multiplexed into the same MPEG-2 *Transport Stream* and broadcast together with the MHP DSM-CC Object Carousels[7] which contain all the application data. We use this feature for our application: a virtual world showing the pelota court is displayed and small camera models represent available streams at their actual position. From a virtual viewpoint floating above the court, the user can jump to different positions and select a stream, pushing the associated number on the remote control (Figure 3). Each virtual camera can provide real video information, a virtual representation of the court, the players and the ball or an augmented reality render, where the virtual objects enhance the real video.

Transitional flights within the 3D space help orientation when switching from one viewpoint to another. If tracking data for players and the ball are available as well, the viewer can stay within the 3D space - without selecting a stream at all - and watch the game from an arbitrary angle.

[6]Data Over Cable Service Interface Specification, a standard for bidirectional data transmission over cable systems. Cable TV operators use DOCSIS to offer Internet browsing or interactive TV applications with return channel

[7]Digital storage media command and control, this standard defines how to transmit files, directories and streams associated to a file system

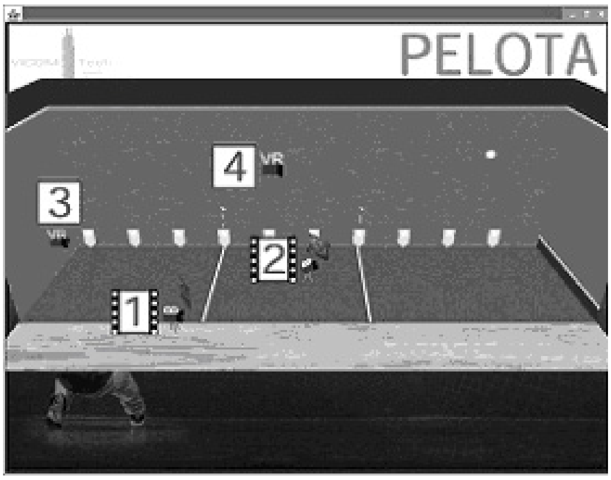

Figure 3: Graphical 3D interface of the pelota application

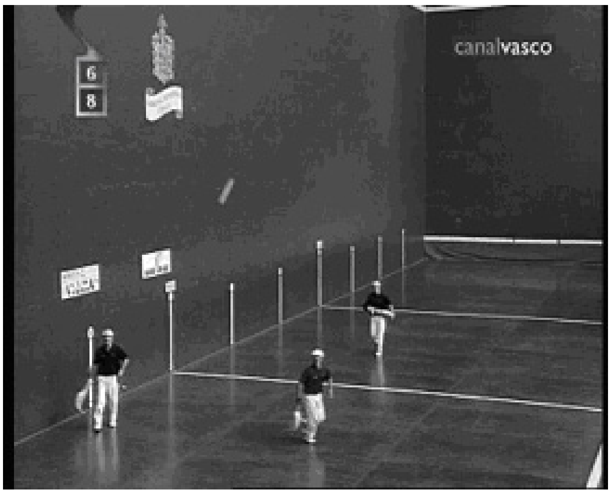

Figure 4: Real pelota match broadcast by ETB

3.7 Video Processing, 3D Data Extraction

Although the 3D model extraction is not the main goal of this project, this information is necessary to be able to accurately register the virtual models with the real world. 2D cameras do not provide information about the depth of the image, but it can be estimated assuming some properties of the court used to play the pelota game.

The left low corner of the court can be easily detected optically since the colour and mainly the luminance features are different in each surface. It allows us to define the orthogonal axis of the real world, as shown on Figure 4. The pelota game lines drawn on the left wall are also easily extracted and are used to calibrate the distances. This information gives us the transformation matrices used to register the virtual world with the real one.

In order to fix the lack of information about the depth of the image introduced by the 2D camera, we consider that the lower part of the body (typically one of the feet) will have $z \approx 0$. The restrictions could be defined as:

$$\begin{cases} z \geq 0 \nabla x, y \\ x, y > 0 \\ min(z_1, z_2, \ldots, z_n) \end{cases}$$

With these simple restrictions we can fix the value of z of the lower foot. Afterwards, the value of y can be extracted with an orthogonal projection to the yz plane. The x value can be obtained with the same projection on the xz plane. The distance references must be calibrated before the systems starts up.

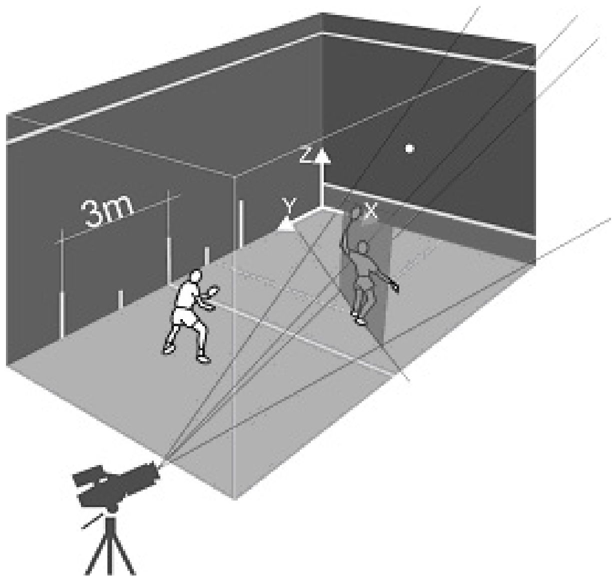

Figure 5: 3D Model extraction scheme

3.8 Tracking

For the 3D data extraction players' bodies must be segmented and tracked. Although we can get tracking information for external systems (STT tracking system) a specific tracker has been developed. This tracker has been tested in real pelota TV matches using broadcast PAL cameras.

The tracking process has been solved in a semiautomatic mode. The first reference points (feet of players) are given manually and then the system is able to track them until an occlusion happens. The tracker has been implemented based on a Gaussian prefiltered 2D correlator. This method has shown good results of robustness and real time performance under different lighting conditions. The tracker is based on a balanced double correlator where one of them belongs to the previous frame and the other one is fixed and sometimes is generated before the tracker starts running.

The ball tracking requires some adaptation of the tracking algorithm. The ball is much smaller than the players and the speed it can fly can be very fast. The small size of the ball means a very low signal-noise ratio and wall texture noises can be considered as the ball by mistake.

In some type of pelota matches mostly known as *jai alai* (Basque expression that means happy party) the ball can reach speeds of over 250km/h. In the slowest type where the ball is hit directly with hands the ball flies up to 120 km//h. Even in these conditions the broadcast PAL cameras are too slow to offer a clear picture of the ball. The low framerate blurs the image and the interlace breaks the shape of it. Furthermore, many occlusions and background changes happen in few frames. All these difficulties make it much more difficult to track the ball than the players.

Deinterlacing filters have been introduced to get a continuous shape and the Kalman filter helps to estimate the position of the ball in frames where it can not be directly identified. Abstract models of the different shapes that the ball can take due to the velocity are used together with the previous frame image

3.9 Occlusions

Up to now, added 3D graphics always overlay all parts of the video. The problem of occlusion arises when a player moves in front of a virtual object supposed to be behind him and therefore, it should not be visible (Figure 6). A quick solution is to decide to only insert graphics in positions where we know that an occlusion will not occur. This method strongly restricts the insertion of virtual objects in the real scene. Instead we implemented two different approaches to deal with this issue: the usage of masked areas and occlusion geometry.

Masking areas implies defining a foreground and background part of the video. For the pelota game the players and the ball are foreground elements and the court is in the background. While rendering the 3D objects we can now clip out parts, where foreground

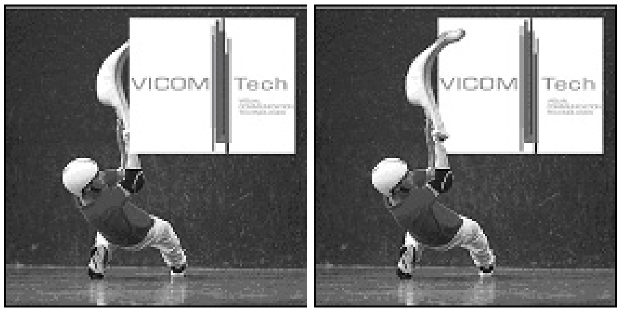

Figure 6: A false occlusion breaks the augmented reality effect.

objects of the video appear. If a stencil buffer is available, a copy of the mask into this buffer can do the work fastest. To generate this mask we calculate a difference image from the grabbed video (Figure 7) and the same scene without players or ball. We clean up the resulting image using linear filters (erosions) and convert it to a binary mask by applying a low threshold to the greyscale mask.

To improve the outcome further, another erosion cleans up fringes and fragments and an expansion of pixels (dilation) closes small gaps and errors caused by the interlaced signal. The resulting composition shown in Figure 8 is produced in real-time. The mask cannot be generated in real time yet. A live broadcasting of pelota would need an offset of some minutes to allow this calculation.

Masks could then be transmitted within an additional DVB-stream (inside the bouquet) or using the MHP DSM-CC object carousel. Another object based encoding standards like MPEG-4 could be also used to define different regions of the video. Since DVB normally resorts to MPEG-2 encoding, the MHP object carousel would be the best way to send the data and ensure the compatibility. Almost no additional bandwidth is needed due to the good compression factor of the binary mask.

The masks can be directly used to distinguish the players and the ball from the background, providing a direct way to have occlusions of flat elements located on the three planes (two walls and the ground). But more information is needed to solve occlusions produced by 3D elements which occupy the inner part of the court. To do this, our approach uses 3D information taken from the 3D data extracting part.

A 3D form approximation formed by few elemental 3D elements (cylinders and prisms) can be used to

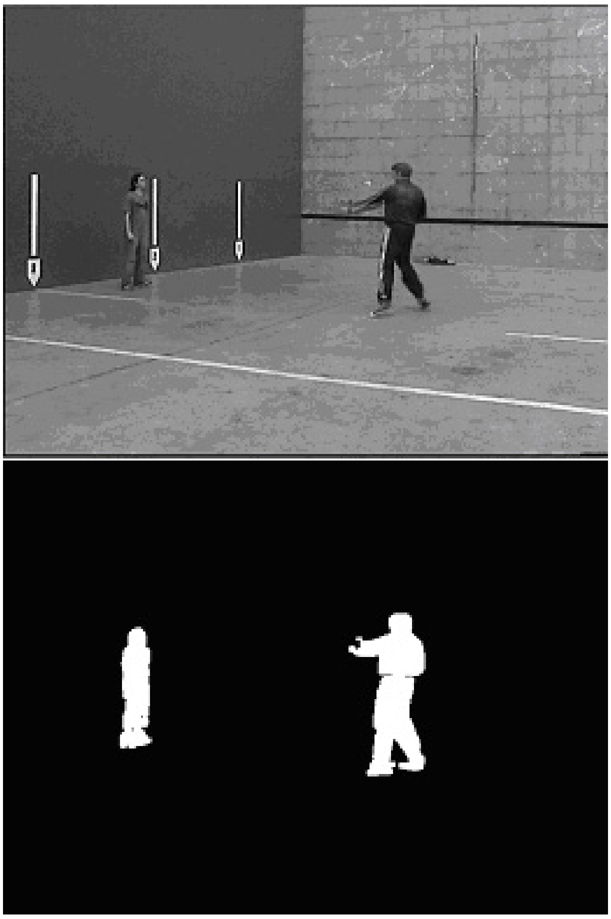

Figure 7: MHP stack Source video and automatically generated mask

Figure 8: Two versions of a real time augmented video stream using occlusion masks

simplify the complex 3D model of the represented human body. If a simpler model is needed, the 2D representation of the mask placed at the proper depth fulfils the most basic requirements to solve most of the occlusions.

3.10 Interactivity and Augmentation Purposes

Since we find ourselves in a TV environment rather than in a PC workstation situation, a simplified user interface and a more restrictive control are needed. Viewers usually watch television to relax and do not want to get overwhelmed with too many options. Typically, distance to a TV set is much bigger than to a PC screen, and therefore requiring bigger fonts, less text and in the best case a simple iconographic language. In our application, we allow selection of cameras (video streams) and virtual viewpoints as well as control over the set of augmented objects to be inserted. Stuck up advertisements can be exchanged freely or mandatory (for instance due to a set user profile). The viewer can turn on visual aids for the pelota game. The trajectory of the ball can be displayed or the current position of the ball itself can be highlighted by additional 3D geometry, drawing focus at the current position of the ball. If the return channel of MHP is available, online betting or downloads of additional player or match information to selected players could be integrated.

If 3D geometry of the players and the ball, were available, occlusions and collisions among virtual objects (flying saucers or a second ball controlled by the viewer, etc) could be handled, allowing to integrate interactive games and more advanced entertainment. Our application offers a basic platform which can be easily extended to more advanced services.

3.11 Broadcast

The integration of the service in a TV broadcaster's headend is not a straightforward task. The features and the quality of the generated signal must fulfil certain conditions defined by the headend's input signal quality control system. The MHP data must be encoded in an Object Carousel (DSM-CC). This carousel structure allows the start-up of the application at any time during the broadcast.

Our application has been developed to be broadcast by the Basque Public Television ETB. The DVB-T parameters used by the Basque broadcaster are:

- Bandwidth: 8MHz
- Guard interval: Ts/4
- FEC: 2/3
- Carrier Modulation: 64 QAM
- Mode: 8K, 6817 carriers (6048 useful)
- Reed Solomon: Enabled, 204 byte packets

Under these parameters, the useful bitrate is fixed to 19.9 Mb/s. Typically, 4 MPEG-2 video channels are broadcast at 4-5Mb/s each of them. It means that there are about 2 or 3 Mb/s available for applications. Current Spanish legislation specifies a maximum of 20% of the whole bandwidth for interactive services, i.e. up to about 2Mb/s. It can be a strong limitation for applications where the data flow is considerable.

3.12 System Architecture

The system is formed by different modules shown in Figure 9. The input signal is processed and the 3D model is extracted according to the mentioned criteria in sections 3.7, 3.8 and 3.9. Once the 3D model is defined, this information is sent to the receiver. There is no rendering process in this part because it would not allow the free camera selection option to the user.

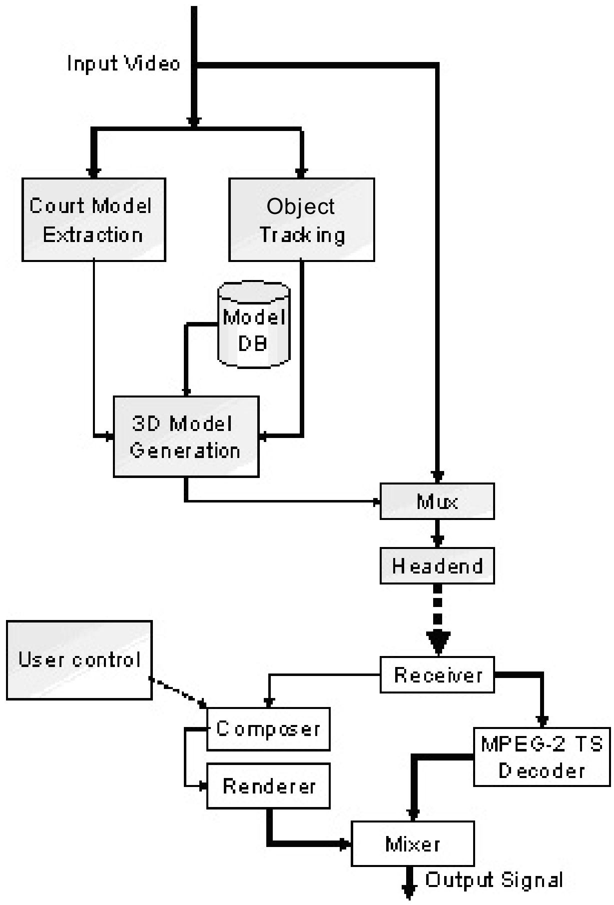

Figure 9: System architecture

4 Discussion - Our Future Work

While our current application only supports the virtual insertion of banner ads and additional visual information of the ball's trajectory many other scenarios for augmented television can be foreseen. Especially localisations of movies can benefit from this idea. Special effects could be inserted live, allowing dynamic effects and more interactive and game like experiences. Just like a DVD-menu enabling spoken language selection, we can take this to a higher level, altering the presented image as well. Visual censorship can be realized, if the user has selected a special profile (children lock) or the set top box has a specific country code.

Moreover, certain elements could just be replaced for purposes of product placement or highlighted to get the attention of the user who could get access to more information about the product selecting the active objects. An online shop could be build according to this advertising model.

User interaction can change the presented image. If a TV program offers an application that takes advantage of the return channel, viewers could participate interacting with the source of the service and taking part in the final result of the video becoming more users than simple passive viewers.

A pure virtual representation could be used for gaming, to develop physical simulations and also to improve the bandwidth. In narrowband scenarios, once the virtual world is loaded, movement vectors of vertices could be enough to play out the video. The bandwidth needed to transmit this information is much lower than the one required by the best video codecs. Some Matlab simulations have been developed to send their dynamic models to our virtual court model. These tests have demonstrated that the data rate requirements can be really low, but the complexity of the simulation and the render demand much more processing power than the current MHP set-top boxes can offer.

The development has been tested in a laboratory environment. Tests with real users would be a necessary step to know their preferences, to improve the user interface and the functionalities of the application. The Basque broadcaster ETB supports this project and will provide the transmission channel when set-top boxes are able to carry out the mentioned requirements.

5 Limitations

While a complete PC environment offers Augmented Reality functionality easily, set-top boxes still hinder an implementation with their current specifications. Important issues that still need to change, include:

- some optional MHP features have to be declared mandatory for future revisions, such as display of AWT-based JMF-players (for correct alignment).
- the personalJava base should generally be substituted by its official successor JME (Java Micro Edition).
- still missing direct video frame access to MPEG data using MHP would allow a guaranteed synchronization.
- a decision on a 3D renderer has to be made, if Java version moves to the JME, the M3G would probably be the best option since it allows 3D software rendering for the JME; if Java3D eventually supports OpenGL state and buffer access, a port from the used Xith3D is done fast due to the very similar structure.
- MHP middleware implementations have to opt for the latest MHP version (including the above changes). Currently most receivers are far behind the latest MHP revision number.
- bandwidth conditions have to increase. Both, the return channel and the broadcast bandwidth dedicated for applications are critical aspects.

6 Conclusion

We described a complete implementation of a possible scenario for enhanced television where interactivity and Augmented Reality techniques are used, granting the viewer control over the finally presented image. A real time compositing of a real scene and freely selectable images or spatial rendered objects is feasible (see the user interface in Figure 10). Although the 3D rendering is still only possible in a simulation, we could describe fields of use and technical approaches in the context of the Digital Television Broadcasting standard and its extension for interaction - MHP Synchronisation, tracking and occlusions were covered and further interaction can be integrated easily resorting to yet available 3D information of the court and moving objects. A new interface for video stream selection has been introduced - allowing not only a real camera viewpoint, but moreover the observation from any arbitrary angle.

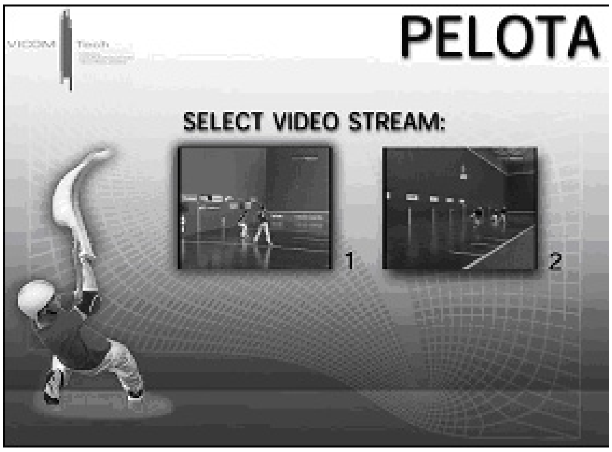

Figure 10: Pelota application user interface

Still, many limitations have to overcome to see augmented television live. We listed important issues and hope for hardware and middleware producers to follow - allowing an augmentation of TV in the near future.

References

[BB95] Shawn C. Becker and V. Michael Bove, *Semiautomatic 3-D model extraction from uncalibrated 2-D camera views*, SPIE Proceedings no. 2410, Visual Data Exploration and Analysis II, no. 2410-49, 1995, ISBN 0-8194-1757-2.

[Ben02] Hervé Benoit, *Digital television: MPEG-1, MPEG-2 and principles of the DVB system*, 2nd edition ed., Focal Press, Oxford, 2002, ISBN 0-240-51695-8.

[LGF+03] A. López, D. Gonzàlez, J. Fabregat, A. Puig, J. Mas, M. Noé, E. Villalón, F. Enrich, V. Domingo, and G. Fernàndez, *Synchronized MPEG-7 Metadata Broadcasting over DVB Networks in an MHP Application Framework*, IBC 2003 (Amsterdam), 2003.

[O'D00] Gerard O'Driscoll, *The essential guide to digital set-top boxes and interactive TV*, Prentice Hall, Upper Saddle River, N.J., 2000, ISBN 0-13-017360-6.

[Pla06] Digital Video Broadcasting Multimedia Home Platform, *Dvb-mhp*, www.mhp.org, 2006, Last visited August 25th, 2006.

[Poy03] Charles A. Poynton, *Digital Video and HDTV, algorithms and interfaces*, The Morgan Kaufmann series in computer graphics and geometric modeling, Morgan Kaufmann, Amsterdam [u.a.], 2003, ISBN 1-558-60792-7.

[Sri02] Hari Om Srivastava, *Interactive TV technology and markets*, Digital audio and video series, Artech House, Boston, 2002, ISBN 1-58053-321-3.

[Sta06] Digital Video Broadcasting Standard, *Dvb*, www.dvb.org, 2006, Last visited August 25th, 2006.

[Web06] Interactive TV Web, *Media control apis*, www.interactivetvweb.org, 2006, Last visited August 25th, 2006.

[Xit06] Xith3D, *Xith3d*, www.xith.org, 2006, Last visited August 25th, 2006.

Citation

Igor G. Olaizola, Iñigo Barandiaran Martirena, and Tobias D. Kammann, *Interactive Augmented Reality in Digital Broadcasting Environment*, Journal of Virtual Reality and Broadcasting, 3(2006), no. 13, February 2007, urn:nbn:de:0009-6-7863, ISSN 1860-2037.