# Virtually moving cameras for digital TV using IBR

Wolfgang Leister*and Henrik Olsen

**Abstract:** *We present an IBR method in order to see whether it is feasible to implement non-trivial graphical applications for use in a digital TV environment.*

Television has been an information channel for the mass market for over fifty years. However, in recent years changes are emerging: (a) Analogue transmission technology will be phased out, while digital broadcast will take over. (b) Many viewers want to be more interactive, and new technology promises to allow that. For the broadcasters, the possible increase of bandwidth on the order of 6–8 times is most interesting.

The providers are less interested in introducing new possibilities. The reasons are high development costs for new products in a conservative market, where new technologies are only carefully introduced. Possible extensions to television are the subject of research, e.g., 3D-Television [1]. Today, proprietary systems and APIs within digital TV (e.g., MediaHighway, Beta) hinder interoperability, increase development costs for all platforms, and thus hinder a broader acceptance.

Digital TV services are meant to offer more than just viewing the broadcast content. Interactivity and applications (games, e-commerce, etc.) are an important issue for the future TV experience. With the venue of MHP (Media Home Platform) defined by DVB forum, and hardware that supports the new standard, some obstacles for the introduction of the new technology will disappear. In addition, the MHP standard has been defined for accessing additional content, navigating in the Electronic Programme Guide (EPG), accessing Internet content, and programming user interfaces. The MHP shall also be able to support more advanced applications like games.

In the following we present an application for digital TV that makes use of IBR methods in order to provide dynamic views from virtual moving cameras for preprocessed TV scenes. The viewer will be able to control the virtual camera using the remote control. This study is intended (1) as a feasibility study, to see whether hardware, software and technology within digital TV are ready for non-trivial applications; (2) to identify what kinds of applications are feasible; and (3) which potential extensions of APIs could be useful.

*Norsk Regnesentral, Postboks 114 Blindern, NO-0314 Oslo, email: Wolfgang.Leister@nr.no

## Digital TV

Digital TV in Europe is driven by DVB Forum (see `http://www.dvb.org`). The technology is mainly based on the definition of a set top box (STB), the MPEG standards (mostly MPEG-2 – ISO standard 13818), and the Media Home Platform (MHP) [2]. In MHP, standard applications are written as so-called "Xlets" in DVB-Java.

The STB with MHP implements a Java Virtual Machine (JVM). The Xlet is compiled to byte code, downloaded via the MPEG-TS, and installed by the class loader. The Xlet can access resources in the DVB stream.

An Xlet is like an applet, which is adjusted to the special needs for digital TV with regard to class loading from the DVB stream. The applications can access additional libraries for accessing DVB streams, providing user interfaces, and other functionalities defined by HAVi, DAVIC, and DVB. More on digital TV can be found in a separate report [3].

## Rendering

Here, we suggest an application where the viewer can interactively walk around in the scene, which she can control via a remote control. The application is implemented using a view interpolation method as described by Chen and Williams [4]; see also [5].[1]

In our view interpolation method we use depth maps in order to create coloured reliefs in space, which can be viewed from other positions. We have also experimented with disparity maps. These depth maps could be provided by e.g., a camera with special depth sensors; such products are available, see e.g. 3DV (`http://www.3dsystems.com`). For our experiments we use the output of a raytracer.

After the content has arrived on the STB, the image for another viewpoint is rendered from one or several sources and presented to the viewer. Note that the rendering process has to be done for each frame, thus the rendering algorithm limits the achievable frame rate.

[1] The purpose of this work is not to present or implement a new IBR method. The IBR method used is well-known and too simple for a real-world application. Therefore, the resulting images are of lower quality and contain artifacts, as shown in [5]. The intention of this work is to see whether the use of an IBR method is feasible in a digital TV context.

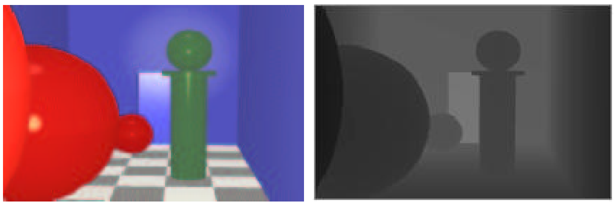

Figure 1: The original colour image and depth map

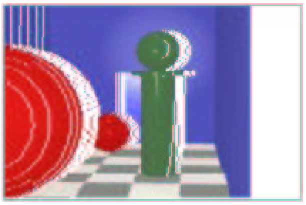

Figure 2: The reconstructed image using a simple IBR method

## Transmission

The input of the view interpolation method is a raster image, a depth image, and information about the camera parameters; this inormation together with the images must be transmitted in the DVB stream. Data are encoded into the MPEG-TS, which can be done in several ways. For images the MPEG standard offers compression using the DCT method; see [6].

## Tests

Image size on TV is normally 780 × 576 pixels for the PAL standard. Our test images were of the size 200 × 130 in order to get a suitable frame rate on the PC for our study. In our experiments, we could achieve a frame rate of about 12 frames per second on a PC (ca. 800 MHz), while the Nokia Mediaterminal only reached 1 frame per second. This rate is far too low. However, the development of faster processors makes us believe that we could reach the necessary frame rate in the near future.

We are aware that the choice of the programming language Java might not be optimal with respect to processing time. However, the digital TV platform makes it necessary to use DVB Java, and one intention of this report is to make conclusions about whether it is possible to use DVB Java for non-trivial functionality (i.e. more than handling menus on the screen).

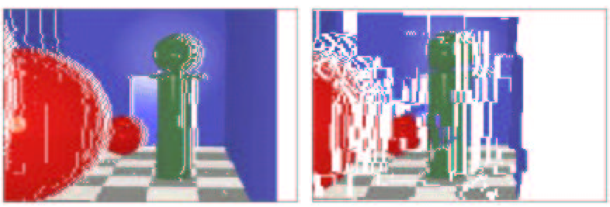

Figure 3: The reconstructed image using compressed depth map (left) and disparity map (right)

In our example, the image in Figure 2 is reconstructed from the image and depth map in Figure 1. The white parts in the reconstructed image are areas where no information about depth or colour is available, as they are occluded by some objects. Some dots and lines in the reconstructed image are caused by the algorithm, when the original view is "stretched". These problems can be eliminated as outlined by Chen and Williams [5]. However, this is beyond the scope of this presentation.

For transmission the MPEG standard defines the DCT method as a compression method, in order to increase bandwidth. The DCT method is lossy, and therefore has a visible impact on the image quality. The compression of the raster image reduces colour quality in the same manner as MPEG or JPEG does.

Due to the compression of the depth map, the geometry of the relief gets distorted; therefore the reconstructed image is visibly reduced in quality. However the problems are moderate, as shown in Figure 3. The impact of compression methods in a disparity map is much larger, which means that the disparity data are much more sensitive to information loss. Therefore, our conclusion is that the use of depth maps is preferable over disparity maps, since quality and execution time are improved.

More suitable library functions of the MHP libraries could accelerate the rendering, but MHP is not optimised for 3D graphics at all.

## References


[1] C. Fehn, P. Kauff, M. Op de Beeck, F. Ernst, W. IJsselsteijn, M. Pollefeys, L. Van Gool, E. Ofek, and I. Sexton. An Evolutionary and Optimised Approach on 3D-TV. In *Proc. IBC 2002, Amsterdam*. available on `http://bs.hhi.de/~fehn/publications/IBC02.pdf`, 2002.

[2] DVB. Multimedia Home Platform 1.0.1. DVB Blue Book A057 Rev.1, 2001.

[3] W. Leister, S. Boudko, O. Aamot, and P. Holmes. Digital TV — and overview. report 988, Norsk Regnesentral, Oslo, 2002.

[4] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. *Computer Graphics*, 27(Annual Conference Series):279–288, 1993.

[5] A. Watt and F. Policarpo. *The Computer Image*. Addison-Wesley, 1998. chapter 21.

[6] J. Mitchell, W. Pennebaker, C. Fogg, and D. LeGall. *MPEG Video Compression Standard*. Chapman Hall, New York, 1997.