Multiview Video Presentation and Encoding

LÁSZLÓ LOIS, TAMÁS LUSTYIK, BÁLINT DARÓCZY,
Budapest University of Technology and Economics, Department of Telecommunications
lois@hit.bme.hu
TIBOR AGÓCS, TIBOR BALOGH
Holografika Kft.
t.agocs@holografika.com

Keywords: multi-view video, video encoding, MPEG-4, 3-dimensional animation and visualization, 3-dimensional television

In this paper the multi-view video encoding and presentation is introduced. In the first part the current multi-view video displays and systems are described. After then we review the developed multi-view video encoding algorithms and the related computer graphics tools. Based the OpenGL system, we show a Depth Image-base Representation (DIBR) method to render an image from several existing reference pictures in a multi-view environment. We also present a new method to encode the depth image efficiently and build a whole multi-view encoding system where the images in the reference views are encoded by using MPEG-4 AVC and the other images are rendered by DIBR. We compare the distortion of this hybrid algorithm and a standard video encoding method in order to establish new multi-view video formats for Holografika's holographic display. Finally we give some further research objectives to complete the developed hybrid method.

1. Introduction

The three-dimensional television system is likely to play an important role in the future broadcasting. Currently this research area is mostly related to the computer graphics and animation since the multi-view camera systems and displays have not entered into practical use, hence the capturing of the multi-view sequences are implemented by a computer graphic software.

The hybrid video encoding systems like MPEG-1 and MPEG-2 are based on the motion compensated DCT coding scheme. The block-based prediction scheme could be easily used for removing the redundancy between the adjacent views (disparity compensation), hence the conventional video encoding methods could be used for motion and disparity compensation. The multi-view profile of the MPEG-2 video encoding contains both motion and disparity compensation, and the several tools in MPEG-4 support to view an object from any viewpoint. In an MPEG-4 scene the virtual and natural video objects could be mixed since in the MPEG-2 multi-view profile uses only binocular representation.

Beside the motion and disparity compensation hybrid schemes the Depth Image-based Representation (DIBR) could be also used for multi-view video coding. This toolkit is also the part of the MPEG-4 video encoding tools.

This paper organized as follows. In the next section, the prevalent multi-view rendering devices are introduced. In Section 3 we describe the Depth Image-based Representation of the three-dimensional objects. In Section 4 we review the multi-view image and video encoding systems where both the motion and/or disparity compensation techniques and the methods based on Depth Image-based Representation are introduced. In Section 5 we show the developed multi-view encoding system which uses DIBR. In our experiments the reference views are encoded and the others are rendered. The MPEG-4 AVC is used to encode the color information by using motion compensation and the depth information in encoded by using a lossless codec. Finally we give some further research objectives to complete the developed hybrid method.

2. Three-dimensional display systems

There are several companies and universities offering 3D display solutions worldwide but all of them can be categorized according to the next groupings of basic principle. Many of those provide new opportunities for 3D presentation, but also present new challenges. This section surveys the capabilities and characteristics of different three-dimensional display technologies.

2.1. Volumetric Displays

Volumetric displays use some media positioned or moved in space where they project light beams and so light beams are scattered/reflected from that point of this media which is generally a semi transparent or diffuse surface.

Moving screen

One solution is when there is a moving screen and the different perspectives of the 3D object are projected on it. A well known solution (Actuality Systems) is a lightweight screen sheet that is rotated at very high speed in a protecting globe and the light beams from an array of LED-s or microdisplays (DMD from Texas Instruments) are projected onto it. By proper synchronization 3D objects can be seen in the globe.

Double or Multi-layer screen

Other commonly used technique in volumetric display technology when two or more LCD layers act as a

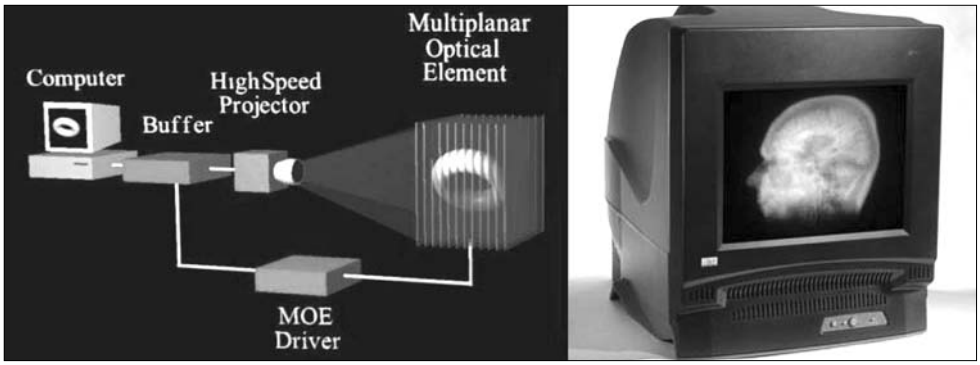


Figure 1. A multi-layer based volumetric display solution

projection screen, creating the vision of depth. Deep Video Imaging produced a 17" display which consist two LCD with the resolution of 1280x1024. The Depth Cube from LightSpace Technologies has 20 XGA (1024x 768) layers inside. The layers are LCD sheets that are transparent/opaque (diffuse) when switched on/off, and are acting as a projection screen positioned in 20 positions. Switching the 20 layer is synchronized to the projection and an adapting optics is keeping the focus.

In volumetric displays the portrayed objects appear transparent, since the light energy addressed to points in space cannot be absorbed by foreground pixels. Thus, practical applications seem to be limited to fields where the objects of interest are easily represented by wire frame models *(Figure 1)*.

2.2. Autostereoscopic Displays

Autostereoscopic displays provide 3D perception without the need for special glasses or other head-gear, the separation for the left/right eye could be implemented using various optical or lens raster techniques directly above the screen surface. Two basic technologies exist to make autostereoscopic displays: stereoscopic and multi-view displays.

2.2.1. Stereoscopic Technology

It has long been known how to make a two-view autostereo display using parallax barrier, lenticular sheet or micropolarizer-based technology. These divide, into two sets, the horizontal resolution of the underlying, typically liquid crystal display device. One of the two visible images consists of every second column or row of pixels, the second image consists of the other columns or rows. The two images are captured or generated so that one is appropriate for the viewer's left eye and one appropriate for the right. The two displayed images are visible in multiple zones in space. If the viewer stands at the ideal distance and in the correct position he or she will perceive a stereoscopic image. The downside of this is that there is a 50% chance of the viewer being in the wrong position and seeing an incorrect, pseudoscopic image. These serious limitations necessitate the use of another autostereo solution. This is either to increase the number of views or to introduce head tracking.

Passive Stereoscopic Technology

This type of displays requires the viewer to be carefully positioned at a specific viewing angle, and with her head in a position within a certain range, otherwise the stereoscopic view will disappear. The information provided by these systems is only twice of the amount contained in a 2D image. Moreover, there are physiological side effects e.g., the contradiction between accommodation and focusing, that can produce discomfort.

Tracking Stereoscopic Technology

To overcome the aforementioned limitations, manufacturers of stereoscopic displays are developing head/eye-tracking systems capable of following the viewer's head/eye movement. Even if this solution cannot support multiple viewers, and there could be latency effects, it provides the viewer with parallax information and it is, therefore, a good solution for single user applications.

2.2.2. Multi-view Technology

This display projects different images to multiple zones in space. The whole viewing space is divided into a finite number of horizontal windows. In each window only one image (view) of the scene is visible. The viewer's two eyes see a different image, and the images

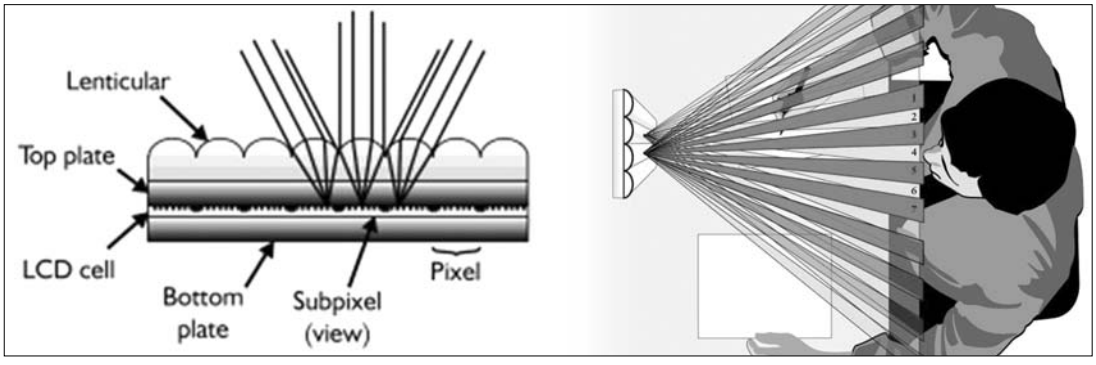


Figure 2. Lenticular display

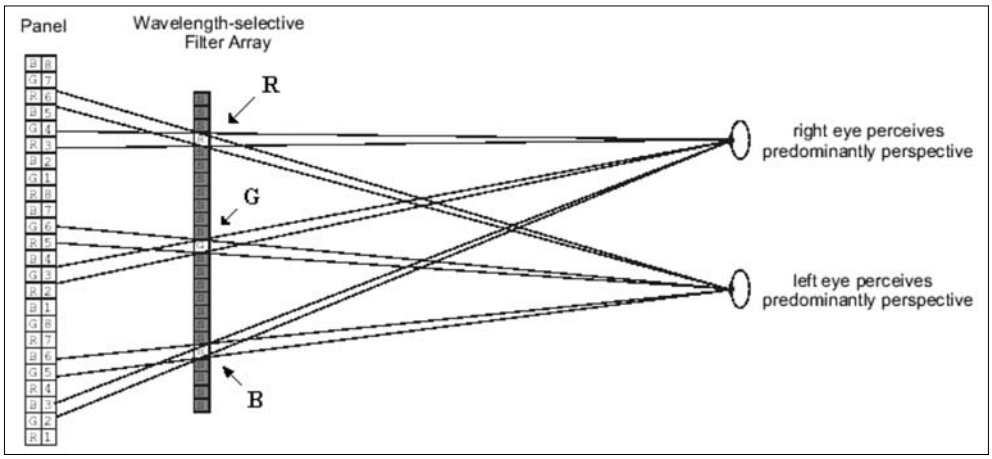


Figure 3.
Wavelength selective filter based autostereoscopic display

change when the viewer moves his head, causing "jumps" as the viewer moves from window to window. It does not require 3D eyeglasses and allows multiple simultaneous viewers, restricting them, however, to be within a limited viewing angle.

Lenticular Displays

Multi-view displays are often based on an optical mask, on a lenticular lens array. This is a sheet of cylindrical lenses placed on top of a high resolution LCD in such a way that the LCD image plane is located at the focal plane of the lenses. The effect of this arrangement is that different LCD pixels located at different positions underneath the lenticulars fill the lenses when viewed from different directions. Provided these pixels are loaded with suitable stereo information, a 3D stereo effect is obtained in which left and right eyes see different but matching information. Lenticular state of the art displays typically use 8-10 images *(Figure 2)*.

Parallax Barrier Displays

A parallax barrier, which comprises an array of slits spaced at a defined distance from a high resolution LCD, is one such a micro-optical component like the lenticular lens array. The parallax effect is created by this lattice of very thin vertical lines, causes each eye to view only light passing through alternate image columns.

Displays with different optical filters and rasters

These displays are based on wavelength dependent filters creating the necessary divided viewing space for the 3D vision. The wavelength-selective filter array is placed on a flat LCD panel, and a combination of several perspective views (state of the art displays provide eight views) is represented to the observer.

The images for wavelength-selective filter arrays contain different views which are combined in a regular pattern. The filter array itself is positioned in front of the display and radiates the light of the pixels from the combined image into different directions, depending on their wavelengths. As seen from the viewer position the different spectral components are blocked, filtered or transmitted. So a perception of different images in the viewing space is enabled *(Figure 3)*.

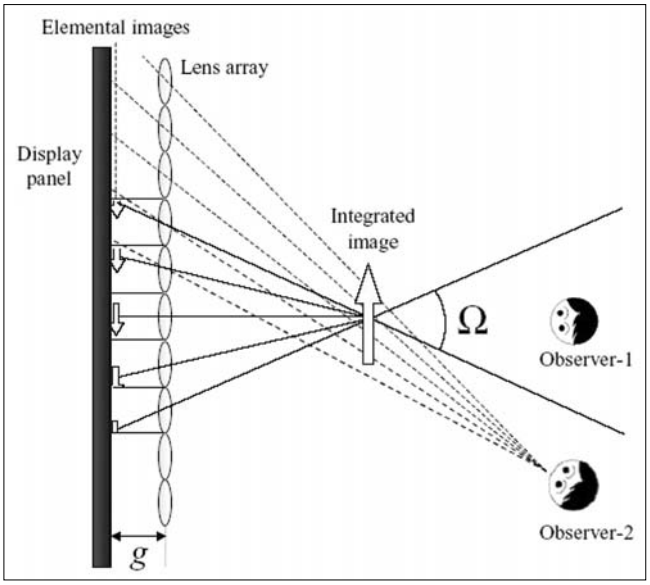
Integral imaging

Integral imaging (InIm) uses a lens array and a planar display panel. Each elementary lens constituting the lens array forms each corresponding elemental image based on its position relative to the object, and these elementary images displayed on the display panel are integrated at the original spatial position of the object forming a 3D image. A primary disadvantage of InIm is its narrow viewing angle. The viewing angle, the angle within which observers can see the complete image reconstructed by InIm, is limited due to the restriction of the area where each elemental image can be displayed. Generally, in the InIm system each elemental lens has its corresponding area on the display panel. To prevent image flipping, the elemental image that exceeds the corresponding area is discarded optically in direct pick up method or electrically in computer-generated integral imaging (CGII) method. Therefore, the number of the elemental images is limited and an observer outside the viewing zone cannot see the integrated image *(Figure 4)*.

2.3. Holographic Techniques

The holographic technology has the ability to store and reproduce the properties of light waves. Attempts have been made to use acousto-optic material. Further moving holograms have been created using optically addressed spatial light modulators. Pure hologram tech-

Figure 4. Integral imaging displaying method



nology utilizes 3D information to calculate a holographic pattern. This technology generates true 3D images by computer control of laser beams and the position of a system of mirrors. Compared to stereoscopic and multi-view technologies, the main advantage of a hologram is in the quality of the picture. The historical disadvantages are: the huge amount of information contained in the hologram which limits its use to mostly static 3D models; and the total incompatibility with existing displaying conventions.

Holographic technology from Holografika

Each point (voxel) of the holographic screen of Holo Vizio system emits light beams of different colour and intensity to the various directions (exactly how a point of a window does), in a controlled manner The light beams are generated through a patented specially arranged light modulation system and the holographic screen makes the necessary optical transformation to compose these beams into a perfect 3D view. The light beams cross each other in front of the screen or they propagate as if they were emitted from a common point behind the screen. With a proper software control of the light beams viewer or viewers see objects behind the screen or floating in the air in front of the screen. The system can be upgraded to large scale (wall-size holographic screens), resolution, brightness, etc. is not limited by principle *(Figure 5)*.

The main advantage of this approach is that, similarly to the pure holographic displays, it is able to provide all the depth cues and it is truly multi-user within a reasonably large field of view. This is a high-end solution compared to other technologies and fulfils all the requirements of real 3D displaying simultaneously, it creates all light beams that are present in a natural 3D view, that is the reason why one sees the same as in reality. The display is able to provide all the depth cues and is truly multi-user within a reasonably large field of view. This is qualitatively very different from other contemporary multi view technologies that force users into approximately fixed positions, because of the abrupt view-image changes that appear at the crossing of discrete viewing zones. By contrast, this holographic display provides continuous horizontal parallax with 0.8° angular resolution for the full 50° field of view, free of any transitions between view images. Reconstruction of the perfect 3D view makes this system a really working 3D display technology *(Figure 6)*.

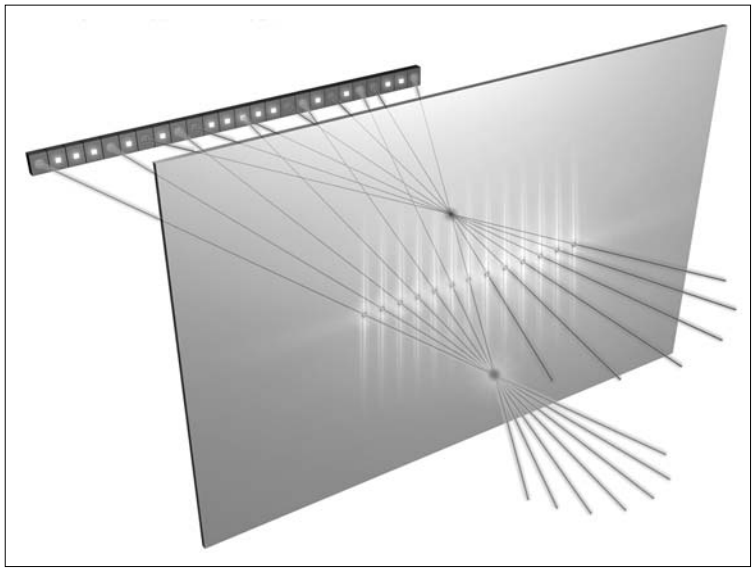
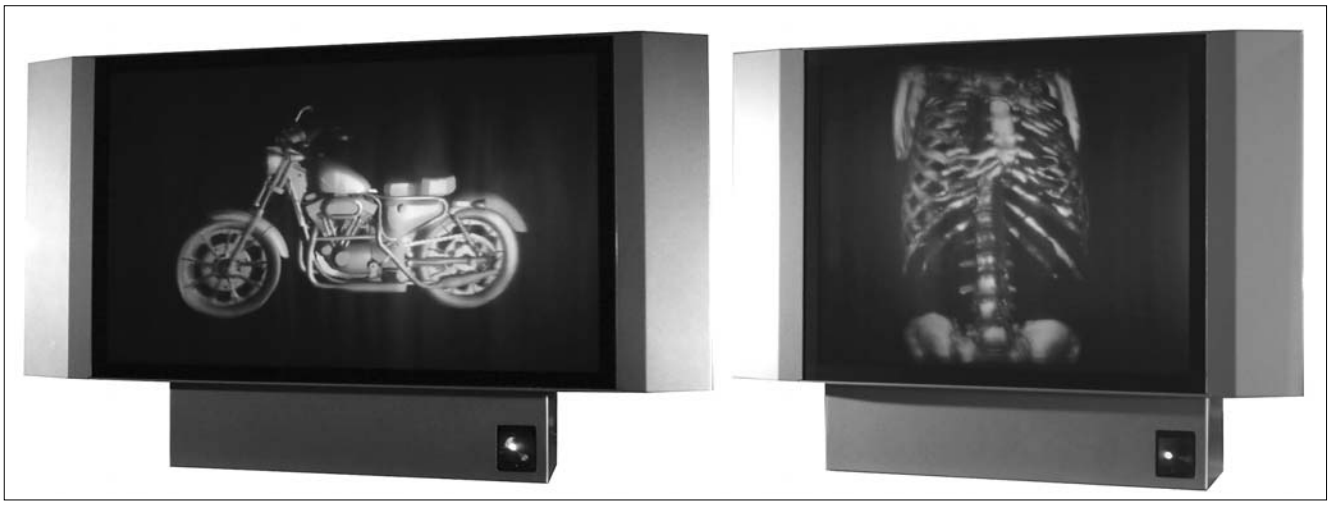


Figure 5. HoloVizio system from Holografika

3. Depth Image Based Representation of 3D objects

Image based rendering (IBR) techniques have been proposed as an efficient way of generating novel views of real and synthetic objects. The first step in the DIBR developed was the generalization of sprites, a technology already widespread in the 80's. The sprite is a planar projected image of an object or a part of a 3D scene, which is moveable by means of an affine transfor-

Figure 6. HoloVizio in operation – CAD and medical application



mation according to the camera. However, in the case of fast camera movement or changes in orientation, this method results in noticeable distortion. If we also store the depth value of the point of the sprites beside its colour (sprite with depth [1]), then we can handle the slight non-affine distortions of texture and shape depending on the camera parameters. The depth value of a pixel means the distance from the spectator.

Based on the sprites with depth values, an extended picture with depth information was defined (relief image) which contains the depth values in addition to the colour components.

There are some cameras on the market that can sense the distance between the point and the camera, but their accuracy is currently not acceptable, but in the near future it would possible to create relief images with cameras. Besides, there are several algorithms which are able to recognize the convergent points of different sequences and set the depth value of transparent points by triangulation.

When scanning the 3D scene, the screen position of points in an other view can be calculated with the help of the depth values. However, it is easy to see that there are several points which will be positioned outside the screen on the actual view or covered by an another point. This means that the rendered image will be incomplete. To fill the gaps in multi-view sequences, we can take another image from a different view, or for smaller gaps, apply an interpolation filter.

The missing pixels on a rendered image could be covered on the reference images by an another object. For these cases, the Layered Depth Image (LDI) was introduced in [1]. LDI contains potentially multiple depth pixels at each discrete location in the image. Instead of a 2D array of pixels with associated depth information, it stores a 2D array of layered depth pixels. A layered depth pixel stores a set of depth pixels along one line of sight sorted in front to back order. The front element in the layered depth pixel samples the first surface seen along that line of sight; the next pixel in the layered depth pixel samples the next surface seen along that line of sight, etc. When rendering from an LDI, the requested view can move away from the original LDI view and expose surfaces that were not visible in the first layer. The previously occluded regions may still be rendered from data stored in some later layer of a layered depth pixel. Using this description, we can calculate some of the missing points, but it is not possible to create such descriptions with a camera.

4. Multi-view video coding

In the case of multi-view video coding, a number of cameras are used for recording a given scene, which operate in a synchronized manner. Consequently, the pixels that are visible from more than one camera will be sampled at the same time. Owing to the spatial arrangement of the cameras, pixels representing the same point map to different coordinates along the views. Moreover, in case of a real illuminated surface, intensity and chrominance are likely to vary as well.

4.1. MPEG-2 multi-view profile

While MPEG-1 does not contain any recommendations aiming at the efficient coding of stereo images or other redundant sequences, MPEG-2 comes with the support for stereo coding. The MPEG-2 multi-view profile defined in 1996 as an extension to the MPEG-2 standard, introduced a new concept for temporal scaling and also made it possible to send various camera parameters within the standard bitstream [3].

This profile is based on that of MPEG-2 temporal scalability mode. The first, higher priority bitstreams codes video at a lower frame rate, and the intermediate frames can be coded in a second bitstream using the first bitstream reconstruction as prediction. The same principle is used in the multi-view profile, though in this case multiple refinement layers can be predicted from the view considered as the 'middle one'.

Here, of course, the term 'refinement layer' refers to a separate view along which the appropriate camera parameters are also transmitted in the bitstream. This way the prediction between layers becomes prediction between views *(Figure 7)*.

In MPEG-2 multi-view profile, coding gain is achieved only when the additional layer(s) could be encoded at a lower bit-rate than the sum of the bitrate of the separately coded original sequences. According to the observations, layers refining the side views cannot be compressed significantly better at the same quality, hence the bit-rate needed is proportional to the number of views.

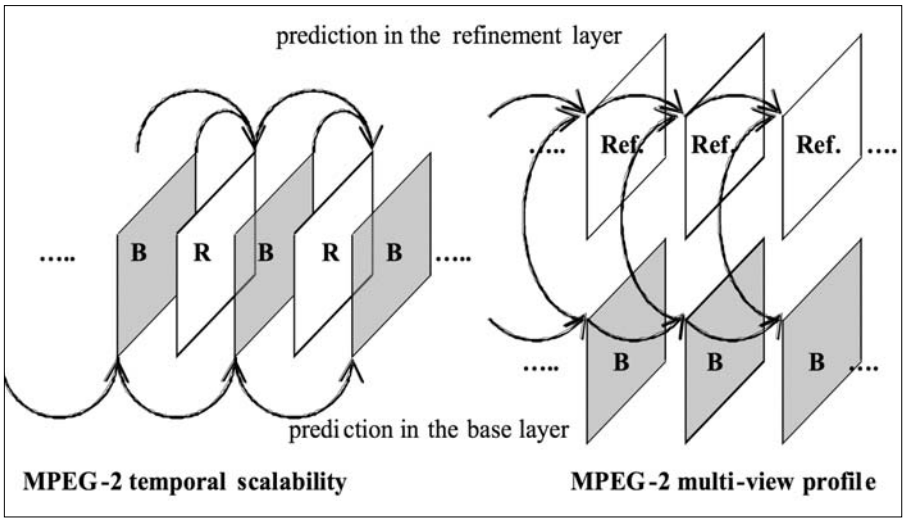


Figure 7. Prediction method of MPEG-2 temporal scalability and multi-view profile

4.2. Implementing inter-view prediction in another MPEG encoder

The aforementioned principle can be applied in case of other codecs using inter-frame prediction. Choosing MPEG-4 AVC, the most efficient MPEG video coding tool at the moment, is beneficial since the use of AVC inherently means a significant increase in coding gain.

MPEG-4 AVC [4] is designed to encode rectangular objects, including natural camera sequences, too. In multi-view systems, one of the toughest problems concerns the proper calibration of cameras, since the more cameras we have, the more difficult it becomes to set the same level of light sensitivity. For this reason, in [5] block-based inter-frame motion compensation is complemented by an illumination compensation which models the effect of illumination by offsetting and scaling the intensity component. These two illumination parameters are also encoded beside the usual motion vector and motion segmentation information. Regarding the tests, this method resulted in 0.5-1.0 dB coding gain, and subjective evaluations also stated that the number of errors appearing in critical regions decreased significantly.

4.3. MPEG-4 Part 16: Animation Framework eXtension (AFX)

MPEG-4 AFX [6] contains the tools for depth-image based representation to a large extent. Static and animated 3D objects of AFX have been developed mainly on the basis of VRML (Virtual Reality Modeling Language), consequently the tools are compatible with the BIFS node and the synthetic and audiovisual bitstreams as well [7-9]. There are two of the main data structures of AFX which are essential concerning DIBR: these are *SimpleTexture* and *PointTexture*.

The *SimpleTexture* data structure holds a 2D image, a depth image and the camera parameters (position, orientation, projection). The pixels of the depth image can be transmitted either pixel-by-pixel as a fourth coordinate beside the intensity and the chrominance signals, or as a separate grayscale bitstream. In most cases, the coding of depth images is lossless since the distortion of depth information along with changing the camera parameters can lead to the miscalculation of pixel positions.

However, a SimpleTexture structure is not enough for describing an object, apart from viewing it only from one direction and allowing minimal deviations of position and orientation from that of the middle camera. For the sake of more complex cases, the *PointTexture* data structure should be used for characterizing objects. Point Texture stores intersections of the object and various straight lines by means of assigning depth value and colour information to each point of intersection.

4.4. Depth-image based multi-view image and video coding

A number of articles deal with depth-image based coding beyond the tools of MPEG-4. One of the most comprehensive achievement is described in [10], which introduces a new algorithm for the efficient coding of the MPEG-4 AFX *OctreeImage* representation and also gives an exhaustive and interesting overview of DIBR issues.

MPEG-4 AFX is not the only available tool for LDI coding. A solution enumerating a couple of alternatives can be found in [11]:

• Store the number of layers (namely the number of colour-depth pairs) for each pixel. This image can be encoded as a grayscale image with usually a few gradation levels. The JPEG-LS [12] algorithm has proven to be significantly more efficient than the Deflate (ZIP) algorithm regarding either compression ratio or elapsed time.

• The different colour component and depth layers are encoded separately. Since the larger are the layers, the smaller is the number of pixels, the following alternatives have been examined:
- MPEG-4 Shape Adaptive DCT
- Rectangular shape image completion, encoded by JPEG 2000
- MPEG-4 arbitrary shape codec
- VOW (Video Object Wavelet) codec that is able to encode arbitrary shaped images just the one above. According to the observations, VOW performed best at both the color and distance layers.

JPEG 2000 is also used for encoding a still picture in a multi-view manner [13]. In this research, the authors experienced that in case of 16 bit depth images the JPEG 2000 encoder produced such reconstructed picture quality at 0.3 bpp compression ratio that is just acceptable for DIBR. Important regions for ROI (Region of Interest) coding in JPEG 2000 are set by examining the depth image and the picture itself. Additionally, depth information is companded before encoding.

5. Experiments and results

In this section, we enumerate the effectiveness of the depth image based multi-view video encoding by implementing the DIBR rendering algorithm, the encoding of the depth and the color images and generating a natural-like synthetic multi-view video sequence. The target bit-rate was between 24 and 30 Mbps which fits into one DVB-T multiplex and the target number of cameras was 60 which corresponds the holographic display by Holografika.

The 60 cameras capture a virtual reality scene where 12 different cars turn in a simple 90-degree bend while there are trees and a high-rise blocks of flats in the background. The resolution was 512x320 for each camera. The car models of the Need For Speed 3 Hot Pursuit are used, this format could be easily imported and contains a car-specific texture-mapped polygon representation of 3-D objects (car body and several – usually four – wheels). Large amount of overlaps occurred on the rendered images and due to the transpa-

rent texture parts some image areas could be captured only by one camera. Only some viewpoint was compressed where the RGB image is encoded by using MPEG-4 AVC and the depth image is compressed by a lossless method. The missing views are rendered by using the decoded RGB and depth image.

In the first step we determine that 9 bit depth value in each pixel is capable of rendering the images at PSNR of 38 dB when the DIBR rendering uses the original images as references.

Number of reference views	Camera angle
13	0°, ±5°, ±10°, ±15°, ±20°, ±25°, ±30°
11	0°, ±6°, ±12°, ±18°, ±24°, ±30°
7	0°, ±10°, ±20°, ±30°
5	0°, ±15°, ±30°

Table 1. The angle of camera of the reference views

5.1. Image synthesis by using depth information and reference images encoded by MPEG-4 AVC

To develop a depth-based image rendering system we must define the proper parameters of each camera. Since we use OpenGL to produce the multi-view sequence the camera parameters could be extracted from the OpenGL transform matrices.

In the OpenGL, a point (*x,y,z*) in the virtual space is transformed to display coordinates (*u,v,d*) using homogeneous coordinates as follows:

1. Modelview transform:

$$[x', y', z', h'] = [x, y, z, 1] \cdot T_{view} \quad (1)$$

where T_{view} denotes the modelview transform matrix which can be requested from the OpenGL system after defining the viewing transform.

2. Perspective transform:

$$[x'', y'', z'', h''] = [x', y', z', h'] \cdot T_{pers} \quad (2)$$

where T_{pers} is the perspective transform matrix.

3. Display transform:

$$u = (x'' + 1) \cdot \frac{V_{sx}}{2} + V_x \qquad v = (y'' + 1) \cdot \frac{V_{sy}}{2} + V_y$$
$$z_d = \frac{Z_{max} - Z_{min}}{2} \cdot Z'' + \frac{Z_{max} + Z_{min}}{2} \quad (3)$$

where *u* and *v* denotes the horizontal and vertical pixel position on the rendered picture and z_d denotes the depth information, Z_{min} and Z_{max} denote the minimal and maximal depth value, respectively, the width and height of the display is v_{sx} and v_{sy} and the top-left coordinate is (v_{sx}, v_{sy}).

This transform could be also represented using a 4x4 matrix, this matrix is simply called T_3- in this paper, hence

$$[u, v, z_d, \tilde{h}] = [x'', y'', z'', h''] \cdot T_3 \quad (4)$$

By using the above equations, the DIBR-based image reconstruction could be implemented as follows:

• Let $R_k(,)$ denote the reference image in the *k*-th view and $Zbuff_k(,)$ the corresponding depth image which contains the depth value of rendered pixels and there are $+\infty$ in the untouched pixel positions. To render the target image in the *n*-th view by using the reference image in the *k*-th view we use the z-buffer algorithm.

• For every (u_k,v_k) on the :

– Get the depth information z_k at the (u_k,v_k) position on the depth image

$$z_k = Zbuff_k(u_k, v_k) \quad (5)$$

– Determine the original 3D coordinate (*x,y,z*) as follows (6):

$$[x, y, z, h] = [u_k, v_k, z_k, 1] \cdot (T_{view,k} \cdot T_{pers,k} \cdot T_{3,k})^{-1}$$

– Calculate the position and depth on the picture in the *n*-th view as follows (7):

$$[u_n, v_n, z_n, h_n] = [x, y, z, h] \cdot T_{view,n} \cdot T_{pers,n} \cdot T_{3,n}$$

After dividing the vector by scalar to keep the last component = 1 in the homogenous representation we get the position and the depth value on the target image. Both coordinates of the position must be quantized to get an integer value and the pixel of the target image at this integer coordinate position must be updated according to the z-buffer algorithm. By scanning all reference pictures the target image could be rendered but some pixels could be untouched.

The quantization of the position coordinated could cause a misplacement error by one pixel position, while the quantization of the depth value can enlarge the misplacement error by more pixel position.

The steps of the image synthesis are shown in *Figure 8.*

First the nearest reference view is used to render the target image, after that all reference views are used in ascending order of distance from the target image. Finally, the missing pixels are produced by an interpolating filter. The covered area is mainly covered by more pixels than the area contains since several pixel occurs on more reference images but the shading could be different due to the illumination. This case could be detected by comparing the shading, the position and depth value on the target picture, and if these values show significant similarity we assume that these pixels from the reference images correspond a same pixel in the 3D space hence we use the shading of the pixel form the most adjacent view.

Due to misplacement, illumination and coverage problems the subjective and objective quality of the rendered images are very different. While the objective quality in PSNR of the rendered image was very poor, in contrast the subjective tests showed significant degra-

Figure 8/a.
Prediction from the nearest left reference image (PSNR = 24.10 dB)

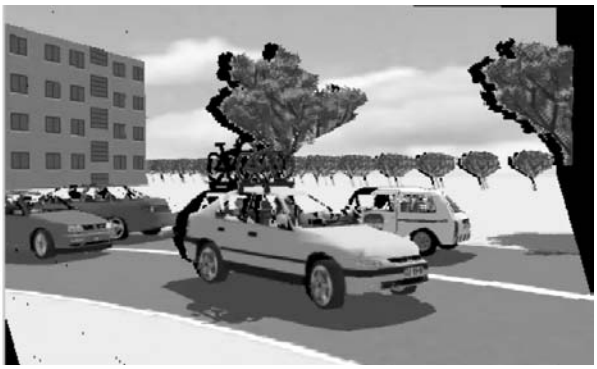


Figure 8/b.
Prediction from the nearest right reference image (PSNR = 27.63 dB)

Figure 8/c.
Prediction from the nearest left and right reference image (PSNR = 33.45 dB).
There are uncovered pixels around the tree on left and the white car.

Figure 8/d.
Prediction by using all reference views (PSNR = 34.64 dB).
The missing pixels are almost covered.

Figure 8/e.
The result of the interpolation filtering of the missing pixels on Figure 8/d (PSNR = 36.25 dB)

Figure 8/f.
The original image

dation only by the boundary of the objects. To find a suitable objective distance measure, we define two new distortion measures based on the well-known PSNR measure. The first measure called $PSNR_{covered}$ is used to handle the coverage problem, this measure is the average PSNR evaluated only on the covered pixels. To handle the misplacement error we define $PSNR_u$ where the horizontal coordinate denoted by u in (7) is rounded to the nearest integer position where the squared error between the rendered pixel and the original pixel is smaller. The latter solution can not be applicable in a real system and used only for test purposes.

Table 2 shows the results of these distance measures for different number of reference views where the reference views are compressed by using our H.264/AVC codec with quantization parameter of 32.

The results in Table 2 show that the best performance could be achieved by a proper rounding of the horizontal coordinate u in (7) but this could not be implemented in a real system in this way. Furthermore, the uncovered pixels cause only slight quality degradation.

Finally, the larger number of reference views generate insignificant quality improvement while the encoding of the depth images of reference views are very

costly. Hence we suggest to use 5 or 7 reference views in the following experiments.

5.2. Lossless compression of depth images

The depth image pixels are quantized with 9 bits and these 9 bit symbols must be restored without error. The current lossless compression algorithms are designed for 8 bit alphabet hence one has to develop a new method or modify the current algorithms. In our experiments we modified the following well-known algorithms to support 9 bit images:

Deflate (ZIP):

Deflate compression is an LZ77 derivative used in zip, gzip, pkzip and related programs. In 1977 Abraham Lempel and Jacob Ziv presented their dictionary based scheme for text compression which outputs offset and lengths to the previous text seen and also outputs the next byte after the match.

GIF87a (LZW):

The GIF is an implementation of LZW (Lempel-Ziv-Welch algorithm) with some special code. LZW is a dictionary based scheme based on LZ78 which outputs bytes and codewords: pairs of offsets and lengths. LZW outputs always codewords, they refer to a dictionary that has 4096 entries.

Burrows-Wheeler transform with WFC and RLE-BIT algorithm:

This hybrid method is based on the implementation of Burrows-Wheeler transform by Jürgen Abel [14]. It is based on a permutation of the input sequence – the Burrows-Wheeler Transformation (BWT) –, which groups symbols with a similar context close together. It the used version, this permutation was followed by a Weighted Frequency Count (WFC) ranking algorithm and a final entropy coding stage using RLE-BIT algorithm and adaptive arithmetic coding.

PNG (Portable Network Graphics):

The image compression algorithm of PNG works on a byte basis. First, the pixels of the image are encoded as bytes. Optionally, the bytes are filtered by one of the 4 predictors to improve compression. The prediction error bytes are compressed to by using the deflate algorithm and the resulting symbols are Huffman encoded.

JPEG-LS:

This method is the predictive lossless part of the JPEG image coding standard. Similarly to the PNG, is uses 8 predictors and the prediction error in each pixel is Huffman encoded.

The JPEG-LS is only method that supports the 9-bit alphabet, hence the other methods are modified to support 9 bit input symbols. The result of the 9 bit variants are shown in *Table 3*. The performance of the 9 bit BWT algorithm is significantly better compared to the other methods hence we use this result to calculate the required bit-rate for encoding the multi-view video sequence.

5.3. The bandwidth of the encoded RGB and depth images

Based upon the latter results, *Table 4.* shows the required bandwidth of the encoded RGB and depth images of the reference views. The results show that at 20.74 Mbps the PSNR value of 26.7 dB can be achie-

Number	Images rendered by DIBR			Every image		
of ref. views	PSNR	$PSNR_{covered}$	$PSNR_u$	PSNR	$PSNR_{covered}$	$PSNR_u$
5	24.988	24.625	28.029	25.726	25.394	28.518
7	25.589	25.282	28.831	26.555	26.283	29.425
11	25.783	25.527	29.082	27.269	27.059	29.973
13	25.776	25.535	29.073	27.535	27.345	30.129

Table 2. The resulting values of the distance measures for different number of reference views

Lossless image compression method	Total size of 610 images [Bytes]	Average bit rate for one camera [Mbps]
LZW: 9 bit..13 bit	19332712	6.34
WinZIP (LZ77, 8 bit)	12049986	3.95
9 bit LZ77 encoding	10474861	3.43
9 bit PNG encoding (PNG prediction + LZ77)	11782353	3.86
JPEG-LS	16188900	5.31
9 bit BWT + RLE-BIT + WFC	8321324	2.73

Table 3. Lossless encoding of the depth images in the test sequence by using the 9 bit variant of the compression algorithms

Number of reference views	AVC QP	RGB images [Mbit/s]	Depth images [Mbit/s]	Total bitrate [Mbit/s]	Average PSNR [dB]
5	**32**	7.44	13.65	21.09	25.726
7	**32**	10.42	19.1	29.52	26.555
11	**32**	16.37	30.01	46.38	27.269
13	**32**	19.34	35.47	54.81	27.535
61	**42**	20.74	none	20.74	26.678

Table 4. Comparison of total bit rate and average PSNR values for the hybrid DIBR-AVC schemes with several number of reference views. Note that the rendered images require no bandwidth in this scheme but their PSNR values are also calculated in the average PSNR values.

ved by using MPEG-4 AVC with quantization parameter of 42. Almost the same quality could be obtained at 21 Mbit/s by using 5 reference views, where the RGB and depth images of the 5 views are encoded and the images in the remaining views are reconstructed by using DIBR.

The two most promising configurations are capable of transmitting the multiview sequence of 61 cameras on a single DVB-T channel. The single MPEG-4 AVC based scheme has a slightly better performance in quality, but the implementation of DIBR reconstruction is cheaper.

6. Conclusions and further work

In this paper, the two basic methods of multi-view video encoding were shown and compared. First we introduced the multi-view video representation and the devices capable of displaying the multi-view video sequences. We described the Depth Image-based Representation of the three-dimensional objects and reviewed the multi-view image and video encoding systems.

In Section 5 we showed the developed multi-view encoding system using DIBR. In our system the MPEG-4 AVC is used to encode the color pictures of reference views, and the corresponding depth information is encoded by a 9-bit lossless codec based on BWT. The other views are rendered by using the decoded depth and color image of the reference views. The developed hybrid method AVC could achieve the performance of the single-view MPEG-4 AVC scheme.

Further research will be performed on improvement of the performance of lossless compression of the depth images by utilizing the redundancy between the depth images. This redundancy can be by exploited since several depth value in a depth picture of a reference view can be predicted from an other reference view by using (7). As we have shown in the results the misplacement problem after (7) is also an important topic, this could be handled i.e. by using offset buffer on sub-pixel basis or an object warping technique could be also capable of solving this problem.

Another interesting aspect could be a stronger integration of the DIBR reconstruction and the hybrid motion compensated transform coding. In this topic the rendering error of the rendered images could be further compressed by a transform coding engine or the rendered images could be used as reference images in motion compensation.

References

[1] Jonathan Shade, Steven Gortler, Li-wei Hey, Richard Szeliski, “Layered Depth Images”, SIGGRAPH 98, Orlando Florida, July 19-24, 1998, Computer Graphics Proceedings, pp.231–242.

[2] M. Oliveira, G. Bishop, D. McAllister, “Relief textures mapping,” in Proc. SIGGRAPH, July 2000, pp.359–368.

[3] Jens-Rainer Ohm, “Stereo/Multiview Video Encoding Using the MPEG Family of Standards”

[4] ISO/IEC 14496-10:2003, Information technology – Coding of audio-visual objects – Part 10: Advanced Video Coding.

[5] Joaquin Lopez, Jae Hoon Kim, Antonio Ortega, George Chen, “Block-based Illumination Compensation and Search Techniques for Multiview Video Coding”, Picture Coding Symposium, San Francisco, CA, Dec. 2004.

[6] Information Technology – Coding of Audio-Visual Objects – Part 16: AFX – Animation Framework eXtension, ISO/IEC Standard JTC1/SC29/WG11 14 496-16:2003

[7] Information Technology – Coding of Audio-Visual Objects – Part 1: Systems, ISO/IEC Standard JTC1/SC29/WG11 14 496-1.

[8] Information Technology – Coding of Audio-Visual Objects – Part 2: Visual, ISO/IEC Standard JTC1/SC29/WG11 14 496-2.

[9] Information Technology – Coding of Audio-Visual Objects – Part 3: Audio, ISO/IEC Standard JTC1/SC29/WG11 14 496-3.

[10] Leonid Levkovich-Maslyuk, Alexey Ignatenko, Alexander Zhirkov, Anton Konushin, In Kyu Park, Mahnjin Han, Yuri Bayakovski, “Depth Image-Based Representation and Compression for Static and Animated 3-D Objects”, IEEE Transactions On Circuits and Systems for Video Technology, Vol. 14, No.7, July 2004, pp.1032–1045.

[11] Jiangang Duan, Jin Li, “Compression of the Layered Depth Image”, IEEE Transactions On Image Processing, Vol. 12, No.3, March 2003, pp.365–372.

[12] M. Weinberger, G. Seroussi, G. Sapiro, “The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS,” IEEE Trans. Image Processing, Vol. 9, Aug. 2000. pp.1309–1324.

[13] Ravi Krishnamurthy, Bing-Bing Chai, Hai Tao, Sriram Sethuraman, “Compression and Transmission of Depth Maps for Image-Based Rendering”.

[14] www.data-compression.info/JuergenAbel/Preprints/Preprint_After_BWT_Stages.pdf