View Interpolation for Medical Images on Autostereoscopic Displays

Svitlana Zinger, Daniel Ruijters, Luat Do, and Peter H. N. de With, *Fellow, IEEE*

***Abstract*—We present an approach for efficient rendering and transmitting views to a high-resolution autostereoscopic display for medical purposes. Displaying biomedical images on an autostereoscopic display poses different requirements than in a consumer case. For medical usage, it is essential that the perceived image represents the actual clinical data and offers sufficiently high quality for diagnosis or understanding. Autostereoscopic display of multiple views introduces two hurdles: transmission of multi-view data through a bandwidth-limited channel and the computation time of the volume rendering algorithm. We address both issues by generating and transmitting limited set of views enhanced with a depth signal per view. We propose an efficient view interpolation and rendering algorithm at the receiver side based on texture+depth data representation, which can operate with a limited amount of views. We study the main artifacts that occur during rendering – occlusions, and we quantify them first for a synthetic model and then for real-world biomedical data. The experimental results allow us to quantify the Peak Signal-to-Noise Ratio (PSNR) for rendered texture and depth as well as the amount of disoccluded pixels as a function of the angle between surrounding cameras.**

***Index Terms*—Depth Image Based Rendering (DIBR), view interpolation, rendering quality, three-dimensional displays, biomedical imaging.**

I. INTRODUCTION

THE INTRODUCTION of autostereoscopic displays to a clinical setting allows physicians to perceive depth in medical images. The addition of depth perception leads to a faster and better interpretation of the morphology of the patient's pathology and contextual anatomy. Stereoscopic images are being used in various clinical applications, such as surgical planning [1], surgical navigation [2]–[4], minimally invasive endoscopic surgery [5], autostereoscopic intracranial MRA visualization [6], etc. Autostereoscopic visualization of the patient's anatomy has the potential to be combined with augmented reality, which has been reported to increase the surgical instrument placement accuracy [3]. Displaying medical images in a clinical context imposes several restrictions on the image transmission chain. A digital imaging chain may contain errors and artifacts. Such errors involve image digitization, compression, transfer function limitations, dynamic range limitations, etc., which have to be kept below a stringent threshold. The obvious reason for this is the fact that medical decisions are taken based on these images, and flaws in the image may lead to misinterpretations.

The development of high-resolution LCD grids (such as QuadHD grids) has brought high-resolution autostereoscopic screens within reach. However, these screens introduce a new challenge, since the amount of the visualized data becomes enormous, while the images have to be rendered and conveyed to the display in real-time. To cope with this, we intend to transmit fewer views than the autostereoscopic display emits, and to render the missing views at the receiver side using 2D texture+depth information. To this end, the display unit in the operating room is extended with embedded receiver hardware. The development of such embedded hardware is the objective of the iGlance project [7]. This approach alleviates the huge computational costs involved with a multi-view rendering system, as fewer views need to be generated. Further, it reduces the strain on the transmission channel between the image processing unit and the receiver in the operating room, since less data has to be transmitted, which is cost efficient and allows existing bandwidth-limited infrastructures to be used.

To this end, we introduce the context of stereoscopic visualizations for clinical purposes (see e.g., Figure 1). We present our solution of dealing with a bandwidth-limited channel while facing processing resources constraints at the medical workstation. A further contribution of this article is the step-by-step description of our improved rendering algorithm and its quantitative results on a synthetic model as well as on real-world biomedical data. Although limiting the amount of views seems an attractive choice for both bandwidth and rendering at first glance, it generates another problem for visualization. Using fewer views, the reconstruction of the missing views that represent the 3D scene will become more vulnerable for artifacts such as occlusions at the borders of objects. For this reason, the view interpolation and rendering algorithm has to be designed for a proper handling of those reconstruction problems. This forms a considerable part of the processing studied in this paper.

The paper is organized as follows. Section II introduces stereoscopic displaying and its properties. Section III discusses view interpolation, the artifacts it produces as well as the constraints that we impose on it. We describe the view interpolation algorithm in Section IV. Our experimental results are shown in Section V, and our conclusions are presented in Section VI.

S. Zinger, L. Do, and P. H. N. de With are with the Video Coding and Architectures Research group, Eindhoven University of Technology, Eindhoven, the Netherlands. Email: {s.zinger, q.l.do, p.h.n.de.with}@tue.nl.

D. Ruijters is with the interventional X-Ray (iXR) Innovation department, Philips Healthcare, Best, the Netherlands. Email: danny.ruijters@philips.com.

Manuscript received July 14, 2010; revised March 6, 2011.

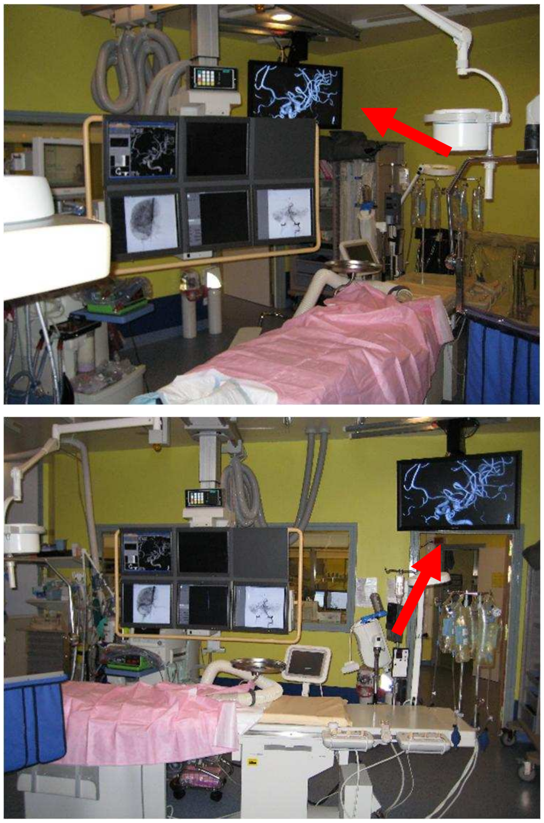


Fig. 1. An 42" autostereoscopic display (top right display, red arrow), which has been mounted in the operating room (OR) at a viewing distance of 3 meters to the work spot of the surgeon.

II. Autostereoscopic display

A stereoscopic display presents the viewer with different images for the left and the right eye. Provided that these images contain proper stereoscopic information, the viewer will have the sensation of seeing depth. Principally there are two kinds of stereoscopic displays: the first type requires the viewer to wear goggles or glasses, and the second type, called autostereoscopic display, allows stereoscopic viewing without any external aid. For the usage of such displays during medical interventions, the absence of goggles is a significant benefit, since there is no compromise of sterility by any external attributes and the goggles might be considered to be disturbing when the clinician is not looking at the stereoscopic display (which typically will be the case during a major part of the clinical procedure) [8]. In contrast to the binocular stereoscope, mutli-view autostereoscopic displays emit more than two views (which in principle would be enough for stereoscopy), in order to have a wider range where the observer can see a proper stereoscopic image, and to allow multiple viewers to perceive the stereoscopic image [3]. Especially in the operating room, where the observer is not fixed to a single spot, this is of relevance.

Popular techniques to achieve autostereoscopy are parallax barrier and lenticular displays. Both techniques can be used to implement multi-view displays. In parallax barrier displays, a raster of slits is placed at a small distance in front of the display, showing a different subset of pixels when viewing it from various angles. A multi-view autostereoscopic lenticular display consists of a cover sheet of cylindrical lenses (lenticulars) placed on top of an LCD, in such a way that the LCD image plane is positioned at the focal plane of the lenses [9]. As a consequence of this arrangement, different LCD pixels underneath the lenses become visible when viewed from a set of predetermined directions. The fact that mutually exclusive subsets of LCD pixels are assigned to different views (spatial multiplex), leads to a lower effective resolution per view than the intrinsic resolution of the LCD grid [10]. In order to distribute this loss of resolution over the horizontal and vertical axis, the lenticular cylindrical lenses are not placed vertically and parallel to the LCD column, but slanted at a small angle.

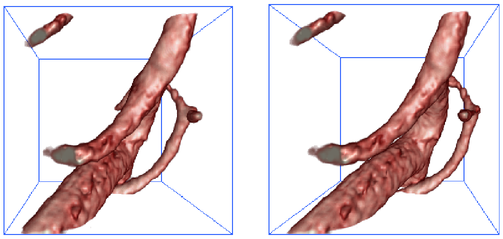


Fig. 2. The same scene rendered from the most left and most right viewpoint.

When the pixels of the autostereoscopic display are loaded with suitable stereo information, a 3D stereo effect is obtained, in which the left and right eye see different, but corresponding, information, see Figure 2 [11]. Most commercially available display lines offer eight or nine distinct views, but our technology will be applicable to any number of views. The stereoscopic views have to be loaded with the images corresponding to the angle they are emitted. The angle between the views is typically in the range of 1-5°, depending on the monitor setup. In clinical interventional applications, the screen is typically mounted at the opposite side of the patient table with respect to the surgeon. The viewing distance amounts from approximately 1.5 meters to 5 meters. For example, assuming a 7-cm distance between the left and right eye, and 3 meters from the viewer to the screen, delivers a 1.3° viewing angle between the view for the left and right eye.

Medical data typically consists of voxel data sets of several hundreds of Megabytes, which can be visualized through a technique called volume rendering [11]–[13]. The raw voxel data typically consists of 12-bit or 16-bit scalar data. Often a transfer function is used to zoom on a user-defined scalar range. This transfer function can also be used to map the scalar data on RGB colors and transparencies. The result of the volume rendering is typically stored as 24-bit RGB images. The individual views displayed on the autostereoscopic display are generated on the medical workstation. A naïve approach would be to simply generate an image for every view that is displayed. In case of displaying a scene at 25 frames per second on a 9-view autostereoscopic screen, this would require rendering 225 views per second. In case of a 256-MB 3D data set (512^3 voxels, 16-bit per voxel), the naïve approach would need to parse more than 56 GB per second. Alternatively, Hübner and Pajarola have proposed to generate the composed multi-view volume rendered data in a single pass [12]. Though

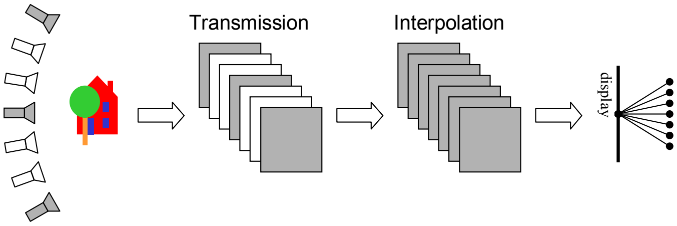


Fig. 3. Only the images of the gray cameras are rendered and transmitted. For the white cameras only their parameters (position, field of view, etc.) are transmitted. The missing views are interpolated at the receiver side. Finally all views are emitted to their respective angle by the lenticular display.

this approach leads to a slightly better image quality, it does not reduce the strain on volume rendering bottleneck, nor on the transmission channel. In this paper, we propose a technique that avoids rendering all views, and transmits only a selection of those views and then interpolates the missing views at the receiver (see Figure 3). In this way, the load on the 3D volume rendering pipeline is reduced. This method does require, though, a depth map for the views that are being rendered. Luckily, this depth map can be easily created during the volume rendering process, without demanding any significant processing resources. The volume rendering process involves traversing a ray through the voxel volume for each pixel in the image. The depth value of a given pixel is then defined as the distance to the first opaque voxel (non-zero alpha value) that the ray encounters during traversal [14].

III. View interpolation

A. Principle

In order to reduce the load of the 3D rendering on the workstation and the transmission channel, we create fewer views than those displayed on the lenticular screen. The missing views are interpolated after decoding the video stream at the receiver side, see Figure 4 [7].

As a start, the concept of our interpolation is adopted from free-viewpoint rendering [15]. This method takes into account the camera parameters and the depth information per pixel. This depth quantifies the distance between the screen and the object displayed at a particular pixel. The camera focus point and the pixel location define a ray in a virtual ray space [16]. By using this information, more accurate interpolated views can be created than using a naïve interpolation method, such as linear interpolation. The conceptual chain for multi-view imaging from rendering to display is illustrated in Figure 3.

A QuadHD LCD grid consists of 3840 × 2160 pixels. Assuming that a 9-view QuadHD autostereoscopic display is used, it would make sense to build up a single view in a resolution of 1280 × 720 pixels [11]. The views that are used for the interpolation algorithm can consist of 32 bits per pixel, where 24 bits are required for the RGB components and 8 bits for depth information. Usage in clinical interventions requires a minimum frame rate of 24 frames per second (fps). When for example four views are transmitted at 24 fps, and the others are interpolated, this would require a bandwidth of 4 views × 1280 × 720 pixels × 24 fps × 32 bits = 2.6 Gbit/s (for uncompressed video data) versus 4.4 Gbit/s for 9 views without depth information. Furthermore, the load on the volume rendering bottleneck on the medical workstation is reduced considerably, since only 4/9 of the data rendered in the naïve approach needs to be generated.

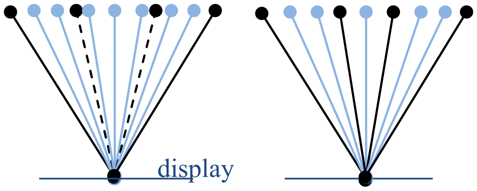


Fig. 4. Two possible configurations for 4 transmitted views, and 9 displayed views. Solid black: transmitted views that can be mapped directly on an output view. Dashed: transmitted views that cannot be mapped on an output view. Light blue: interpolated view.

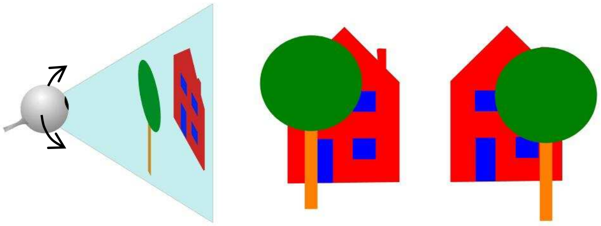


Fig. 5. Depending on the viewing position, different parts of a background object are occluded.

B. Artifacts resulting from rendering

A key artifact that may appear when using free-viewpoint interpolation is the occurrence of disocclusions [17], [18]. When this happens, a part of the scene becomes visible that was hidden in any of the transmitted views, see Figure 5. Consequently, there is no proper information available that should be filled in at the affected pixels. Fortunately, the impact of this effect is very limited for our application, since the transmitted views and the interpolated views are very close to each other. Disocclusions mainly occur for views that are rather far apart, which is not the case for our setup.

Another related artifact of free-viewpoint interpolation concerns semi-transparent parts of the depicted scene. The free-viewpoint algorithm expects a single depth value per pixel. However, when the object that is being depicted is semi-transparent, a pixel can contain visible information that is composed of the light that is reflected by several objects. Once the reflected light has been blended into a single pixel color, it is impossible to dissect it. In the rendering of 3D medical data, this effect is even amplified, since the depicted data is often the result of a volume rendering process [8], [11], whereby a ray of light traverses through a continuous range of semi-transparent material. Therefore, the proposed technique is limited to representations whereby the majority of the volume consists of completely transparent data together with more or less opaque data, as is demonstrated in Section V.

C. Constraints of the rendering process

Since the medical context demands that there are no severe artifacts, the angle between the virtual cameras has to be small,

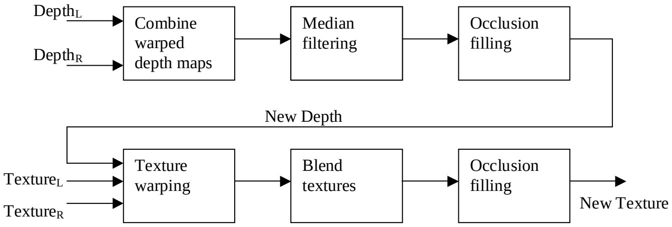


Fig. 6. Block diagram for the proposed DIBR algorithm.

because it will lead to high similarities between the interpolated and the original views, and thus fewer interpolation artifacts. Furthermore, the small angles will lead to fewer and smaller disoccluded areas. This is particularly important, as the content of the disoccluded areas has to be extrapolated from the surrounding information. Because the images are input to medical decisions, it is essential that the artifacts do not lead to misinterpretations of the interpolated images.

It is our aim to achieve a real-time hardware implementation of the proposed interpolation method. Therefore the complexity of the post-processing has to be limited. The rendering algorithm should be simple, while providing an acceptable quality of the rendering results.

IV. Interpolation algorithm

A. Image warping

Depth Image Based Rendering (DIBR) algorithms are based on warping the image from a camera view to another view [19]. Let us specify this in some more detail. Consider a 3D point at homogeneous coordinates $P_w = (X_w, Y_w, Z_w, 1)^T$, captured by two cameras and projected onto the reference and synthetic image plane at pixel positions p_1 and p_2. The 3D position of the original point P_w in the Euclidean domain can be written as

$$P_w = (K_i R_i)^{-1}(\lambda_i p_i + K_i R_i C_i), \quad (1)$$

where matrix R_i describes the orientation of the camera i, K_i represents the 3×3 intrinsic parameter matrix of camera i, and C_i gives the coordinates of the camera center. Parameter λ_i represents the positive scaling factor defining the position of the 3D point on the ray through point p_i. Assuming that Camera 1 is located at the world coordinate system origin and looking into the Z direction, i.e., the direction from the origin to P_w, we can write the warping equation as

$$\lambda_2 p_2 = K_2 R_2 K_1^{-1} Z_w p_1 - K_2 R_2 C_2. \quad (2)$$

Equation (2) constitutes the 3D image warping equation that enables the synthesis of the virtual view from a reference texture view and a corresponding depth image. This equation specifies the computation for one pixel only, so that it has to be performed for the entire image. The depth map of the virtual view can be obtained in a similar manner, given the depth map of the real view.

B. Algorithm backbone

The aforementioned warping forms the basis ingredient for our view interpolation algorithm [20]. The algorithm is illustrated in Figure 6. In multi-view video, the information for warping is taken from the two surrounding camera views, I_L and I_R, to render a new synthetic view I_{new}. Typically, two warped images are blended to create a synthetic view at the new position:

$$I_{new} = Warp(I_L) \oplus Warp(I_R), \quad (3)$$

where $Warp$ is a warping operation and the operation $\oplus$ denotes blending of the warped views. Such an approach requires several post-filtering algorithms, in order to improve the visual quality of the results and it is especially important to close the empty areas on the resulting image caused by occlusions. Initial images I_L and I_R may be divided into layers, prior to performing the warping as described in [21].

The latest research has shown that a better rendering quality is possible when we first create a depth map for a new image [22], [23], since it provides better handling of disoccluded areas. Using this depth map, we perform an "inverse mapping" in order to obtain texture values for I_{new} - the new image to be rendered. In this case, we have two main stages of the rendering algorithm: 1) create a depth map $Depth_{new}$ for I_{new}; 2) create a texture of the new image I_{new}.

The above two stages, which form the backbone of the rendering algorithm, are similar to the approach of Morvan [22] and Mori *et al.* [23]. Our method additionally employs the surrounding depth information to fill in the disoccluded regions more reliably, avoiding cracks in the interpolated image. A detailed evaluation can be found in [24].

C. Depth map creation

The depth map creation consists of the following steps.

1. *Combine warped depth maps.* Combine the depth maps warped from the closest left and right cameras:

$$Depth_{comb} = C\left(Warp(Depth_L), Warp(Depth_R)\right), \quad (4)$$

where C defines the operation of combining two depth maps of both neighboring input cameras by taking the depth values that are closer to the virtual camera. For example, C is set to $Warp(Depth_L(x,y))$ when it is closer to the camera. In practice, combining warped depth maps means taking a maximum or minimum value of each couple of corresponding pixels.

2. *Median filtering.* The filtering function called $Median(Depth_{comb})$ involves applying median filtering to the depth map obtained at the previous step. We take a 3×3 window for the median filter, which allows us to find pixel values that occurred due to rounding of pixel coordinates during the warping. Typically, the rounding of pixel coordinates occasionally produces line-shaped gaps of one pixel width, and a 3×3 window is sufficient to close these gaps.

3. *Occlusion processing.* The resulting image still contains empty regions - disoccluded areas. The following operation finds values for filling in these regions by taking the closest

found background value of the depth map obtained at the previous step. We perform search in eight directions from each empty pixel and take the value closest to the depth of the background:

$$Depth_{new} = Occ\_filling\left(Median(Depth_{comb})\right). \tag{5}$$

For a practical implementation, it means that we take a minimum or a maximum depth value (depending on which depth value represents the background). We aim at finding eight values around an empty pixel. If one of the surrounding eight pixels is also empty, which is often the case, then we move further in the same direction until we find a pixel containing a depth value.

D. Texture creation

Besides the depth image, we need to compute the texture of the new view, which is the final objective. The texture I_{new} is created by the following operations.

1. *Warping textures for the new view.* The new texture image is based on pixels from the left and right texture images. We select the pixels from the left and right image according to the "inverse warping" of the intermediate depth image. This results in

$$Texture_i = Warp_i^{-1}(Depth_{new}) \tag{6}$$

where index i represents the left or right camera, and $Warp^{-1}$ is "inverse warping" - from the location of the new image to a position where the existing left (or right) view is located. When warping, we use the coordinates of the depth map to obtain the corresponding coordinates of the texture at the left (or right) view.

2. *Blending.* The textures obtained at the previous step are blended, specified by:

$$Texture_{blended} = Dilate(Texture_L) \oplus Dilate(Texture_R), \tag{7}$$

where $Dilate$ is depth map-based filtering which aims at preventing ghosting artifacts. These artifacts result from warping the contours of objects that are often represented in the texture as a mixture of foreground and background pixels. We dilate the empty areas on the textures with a square 5×5 structural element. This removes the ghosting artifacts already present on the contours of these areas. Afterwards, the textures are blended. The drawback of this method is that it leads to larger empty areas in the textures, therefore requiring more occlusion filling operations. Experimental results have shown that the ghosting artifacts are typically two pixels wide, which means that at least a 5×5 structural element is needed to avoid them [24]. For computational efficiency we do not use a larger kernel.

3. *Occlusion filling.* Finally, the rendered image is created from the blended texture and by filling the occlusion holes:

$$I_{new} = Occ\_filling(Texture_{blended}). \tag{8}$$

In our optimized version of this rendering approach, both texture and depth are warped simultaneously but kept separated [25]. This leads to less warping operations, thereby increasing the execution speed of the algorithm.

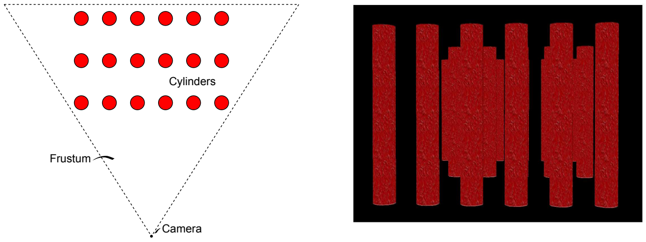


Fig. 7. Synthetic model containing three rows of six cylinders in each row. Left: Schematic setup. Right: Rendering of the synthetic model.

V. Experimental results

In this section, we present the results regarding the image quality of the view interpolation applied to biomedical images. We study the view interpolation quality as a function of the angle between the two nearest cameras. The angle between cameras has to be chosen such that the quality of the rendered views stays sufficiently high for medical application. With the development of multi-view technologies for medical applications, we expect that the amount of occluded pixels will be bounded and subject to regulation. Therefore we measure explicitly the number of occluded pixels as a function of the angle between two nearest cameras. The angle between the cameras then will have to be selected such that the amount of occluded pixels stays limited and below a quality threshold. In the following, we show our rendering results for synthetic and real data.

Varying the angles between the two nearest cameras, we calculate the following quality metrics: the Mean Absolute Error (MAE) of depth values as a percentage of the maximum depth value for this data, defined as

$$MAE = \frac{1}{N}\sum_{s\in S}|Y_v(s) - Y_{ref}(s)|, \tag{9}$$

where S is the discrete image space, N the number of pixels in S, Y_v and Y_{ref} the luminance of the interpolated and reference image respectively. The Peak Signal-to-Noise Ratio (PSNR) for textures, defined as

$$\begin{aligned} PSNR &= 20\log_{10}\left(\tfrac{255}{\sqrt{MSE}}\right), \ with \\ MSE &= \tfrac{1}{N}\sum_{s\in S}\left(Y_v(s) - Y_{ref}(s)\right)^2 \end{aligned} \tag{10}$$

and the Percentage of occluded pixels (OP) that belong to the cylinders, defined as

$$OP = 100 \cdot \frac{N_{occ}}{N}, \tag{11}$$

where N_{occ} is the amount of occluded pixels (which will be filled by the inpainting procedure).

Our first data set is a synthesized model that consists of three rows composed of six cylinders each, shown in Figure 7. The form of the cylinders corresponds roughly to the form of the blood vessels which are to be visualized in medical applications. We took three layers of cylinders to emulate the common situation that the blood vessels occlude each other. The quality of the rendering algorithm will therefore depend on its ability to handle the disoccluded areas. Figures 8-10

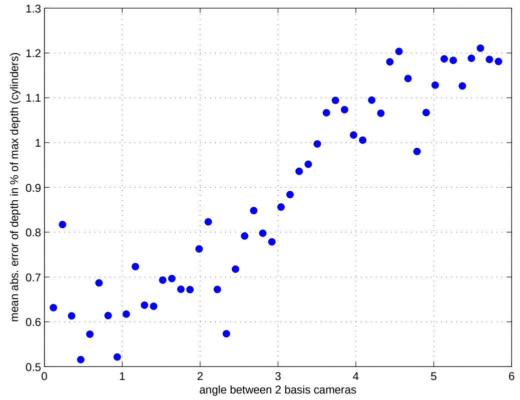


Fig. 8. Mean absolute error of depth values in percentage of the maximum depth value calculated for the model of cylinders.

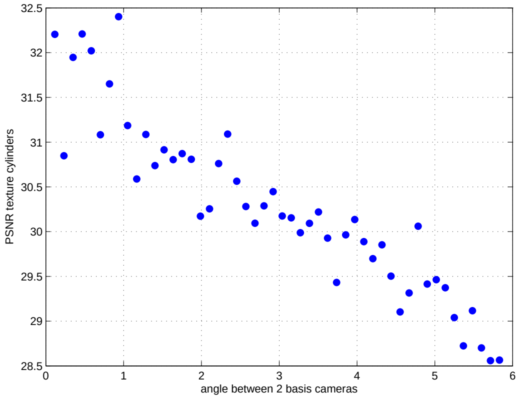


Fig. 9. PSNR of texture values for the model of cylinders.

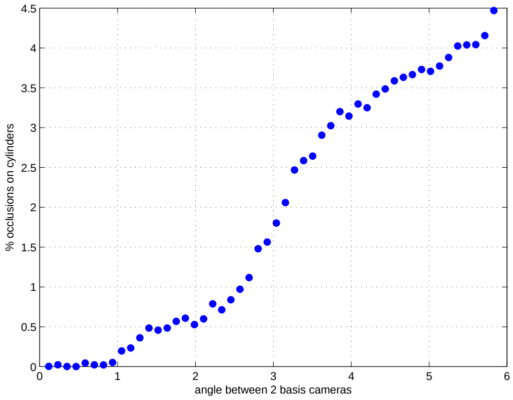


Fig. 10. Percentage of the occluded pixels as a function of the angle between nearest cameras for the model of cylinders.

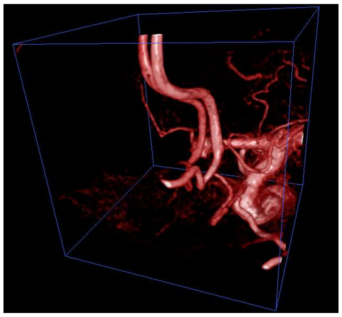


Fig. 11. Visualization of blood vessels which is based on real 3D data measurements.

illustrate the performance of the rendering algorithm when the angle between the initial left and right views changes. Figures 8 and 9 characterize the rendering algorithm by its quality for depth maps and textures, respectively. Obviously, the results deteriorate when the angle between the initial views increases.

Figure 10 shows the number of occluded pixels on the cylinders as a function of the angle between the surrounding views. The number of such pixels is an important parameter of the model and of the camera setting. Pixels in disoccluded areas will be filled in by our rendering algorithm, and this filling procedure is error-prone. These are the pixels where errors may occur, causing wrong visualization and therefore mistakes in the medical diagnosis and treatment. This motivates us to quantify the number of such pixels and to study how this number changes when increasing the angle between the surrounding cameras. It can be observed that the results show a consistent behavior as a function of the increasing camera angle. As will become clear later, the results are also in accordance with the results of a real-world data set showing the same consistent behavior.

In order to check the relevance of our synthetic model of cylinders, we have performed the quality assessment of a mesh extracted from a real-world medical data set, as shown in Figure 11. A mesh representing the segmented vessels was created, and the interpolated image given a left and right camera image has been compared to the ground-truth image, see Figure 12. The mean error in depth, PSNR and percentage of occluded pixels has been calculated for increasing camera angles (Figures 13-15), illustrating the errors that occur with increasing camera angles that can lead to image artifacts for real-world data. Such plots can be used to determine which camera angle is still possible, provided that a desired maximal and average artifact level is specified.

Comparing the results from the synthetic model and the real world mesh, we can see that the view interpolation quality provided by our rendering is higher for the real world mesh, when compared to the model with cylinders. This holds for all three types of measurements that we have performed: the mean absolute error of depth as a percentage of the maximal depth value is about two times lower for the real data; the PSNR of the texture is about 2 dB higher for the real data; the percentage of occluded pixels is significantly higher for the synthetic data (4.5%) compared to the real data (1.3%). From these observations, we can conclude that the model with cylinders can be used for estimating the worst-case scenario of rendering and for finding a suitable angle between the neighboring views.

In order to analyze to which extend the observed results are valid for volume rendered medical data, we have also

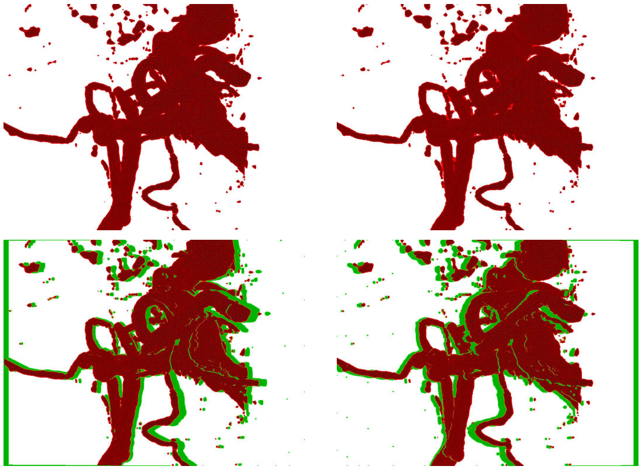


Fig. 12. Top left: Reference rendering of the mesh model representing the vessels. Top right: interpolated view for the same camera position. Bottom left: interpolated image using only the left camera. Green pixels identify the disoccluded areas. Bottom right: interpolated image using only the right camera.

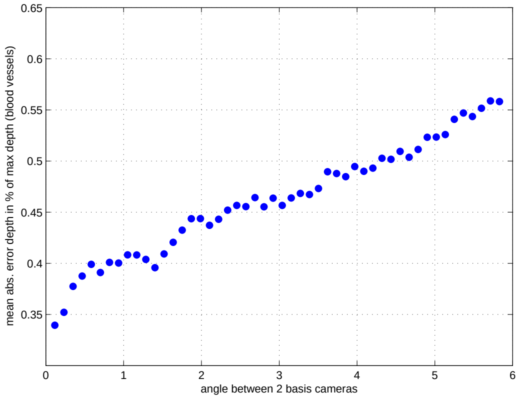


Fig. 13. Mean absolute error of depth values in percentage of the maximum depth value calculated for a mesh extracted from real world data.

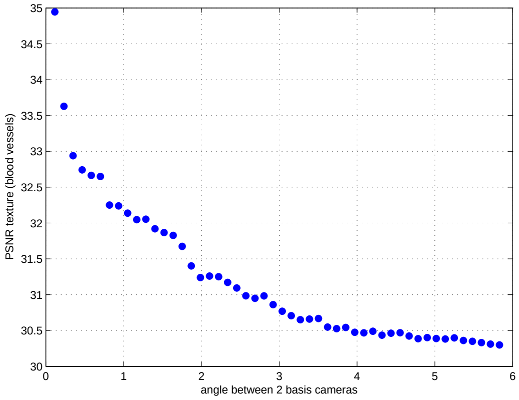


Fig. 14. PSNR of texture values for a mesh extracted from real world data.

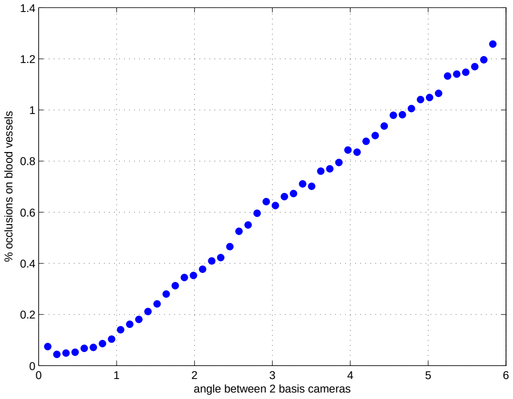


Fig. 15. Percentage of the occluded pixels as a function of the angle between nearest cameras for a mesh extracted from real world data.

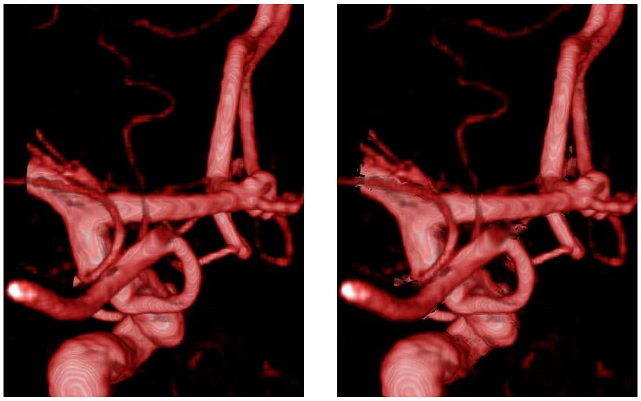


Fig. 16. Left: ground truth reference volume rendering for a given camera position. Right: interpolated image, using two neighboring volume rendered images, whereby the cameras are 2.5° apart.

interpolated volume rendered images of the same real world data set. As can be seen from Figure 16, the interpolated image and the ground truth image are very similar. Most pronounced differences can be observed at the edges of the vessels, which contain semi-transparent voxels. Figure 17 shows the resulting PSNR of interpolating volume rendered images versus mesh rendered images, using the same data set. For small angles, the interpolated volume rendered images are significantly better than the mesh results. The mesh was covered with a high-frequent texture, which leads to different pixel values even for small deviations in the sample locations. The volume rendered images, on the other hand, are relatively smooth, which results in far smaller deviating pixel values. Only for relatively large angles ($> 5°$) the PSNR becomes comparable to the PSNR of the mesh.

The opacity A of a ray being traced through a voxel volume is given by

$$A_{i+1} = (1 - A_i) \cdot \alpha_i + A_i, \tag{12}$$

whereby α represents the opacity of the current sample i [13]. From Equation 12 it can be understood that even modestly transparent voxels saturate a ray rather soon, e.g., three samples with opacity $\alpha = 0.35$ already yield $A = 0.73$. In order to examine the influence of transparency in the volume rendering

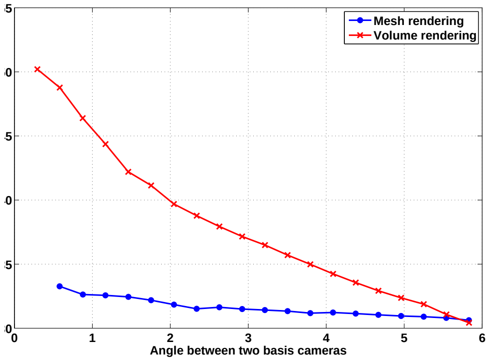


Fig. 17. PSNR of interpolation of volume rendered and mesh rendered images of the data set in Figure 11.

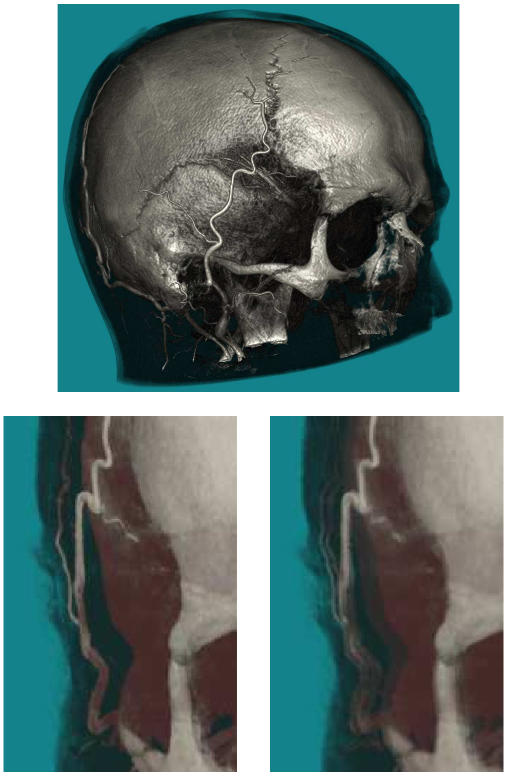


Fig. 18. Top: Extremely low opacities are assigned to the soft-tissue voxels, resulting in depth values that correspond to the location of the skin of the patient. Bottom left: fragment of the ground-truth image. Bottom right: fragment of the interpolated image when the cameras are 2.5° apart. Especially for objects that lie far from the skin surface, such as the small vessels at the left, a ghost image appears.

algorithm, we generated images whereby the soft-tissues in the volume were assigned extremely low opacities. It should be noted that this is not common when rendering vascular data in the clinic, and the purpose was just to understand the effect on the interpolated result. As becomes clear from Figure 18, this leads to severe artifacts for objects that lie considerably deeper than the first non-transparent voxel.

TABLE I
COMPARISON OF VIEW INTERPOLATION, GIVEN TWO NEIGHBORING VIEWS AND DEPTH MAPS, VERSUS VOLUME RENDERING OF VOXEL DATA CONSISTING OF 128^3 (SEE FIGURE 11) AND $512^2 \cdot 396$ VOXELS RESPECTIVELY. THE VIEW INTERPOLATION AND THE VOLUME RENDERING IMPLEMENTATIONS WERE GPU ACCELERATED. THE FIGURES REPRESENT THE AVERAGE TIME IN MILLISECONDS TO GENERATE A SINGLE FRAME IN THE REQUESTED RESOLUTION. ALL MEASUREMENTS WERE TAKEN ON AN INTEL XEON 3.6 GHZ MACHINE WITH AN NVIDIA QUADRO 2000 GPU WITH 1GB ON-BOARD MEMORY.

view resolution	interpolation	128^3 voxels	$512^2 \cdot 396$ voxels
$512 \cdot 512$	5.54	31.67	81.97
$800 \cdot 600$	9.63	34.20	114.63
$1024 \cdot 768$	16.38	50.83	163.93
$1280 \cdot 1024$	25.58	97.09	218.34
$1920 \cdot 1080$	40.94	114.94	280.90

Table I compares the average time necessary to generate a view using view interpolation and volume rendering. Obviously, those times become larger when more pixels need to be rendered and more voxels need to be processed. As can be seen from the table, the view interpolation is always significantly faster than volume rendering, even when rendering relatively small volumes.

VI. Discussion and Conclusions

We have presented an approach for efficient rendering and transmitting views to a high-resolution autostereoscopic display for medical purposes. Stereoscopic vision can be introduced to medical imaging as it facilitates surgeon's knowledge about key objects in 3D during an intervention. The stereoscopic image allows interpreting the 3D shape, including the out-of-plane curvature (the curvature in the z-direction of the image), in a single glance without any additional input interaction. Therefore, it reduces the mental stress on the clinician during the intervention. Further, the stereoscopy reduces the risk of misinterpreting pathologies, due to a biased interpretation.

Autostereoscopic display requires a number of views of the same scene taken from a slightly different angle. These views have to be transported from the control room, where the views are rendered, to the display in the operating room. The contribution of this paper is twofold. At first, it describes a setup of rendering fewer views by the medical workstation and interpolating the missing views at the receiver side. This is a considerable reduction of the load on the 3D rendering pipeline in the medical workstation, and therefore aids in reaching interactive frame rates. Furthermore, the fewer views also relieve the load on the transmission channel. Secondly, this paper describes an efficient algorithm for the interpolation of the views, and examines its signal-to-noise characteristics.

In order to examine the artifacts introduced by the view interpolation, we have quantified the errors that were caused by our approach, using both artificial and real-world data. We introduce a quality metric based on the number of occluded pixels. We expect this metric to contribute to setting the standards for stereoscopic visualization of medical data. The quantitative measurements have shown that for the mesh models the PSNR is higher than 30 dB and the number of affected pixels is lower than 1%, when the angle between the cameras is less than 2.5°. For volume rendering of the same clinical vascular data set the PSNR was even considerably better (starting over 50 dB) for small angles, and comparable for larger angles. We have also investigated the effect of large volumes with very low (but non-zero) opacities. This can lead to severe artifacts, and therefore the described approach is not suitable for such visualizations. It should be noted though, that for realistic vascular data sets, these kind of transfer functions are rather uncommon.

The observed PSNR levels correspond approximately to a compression ratio of 20:1, when using JPEG compression [26]. The literature suggests that lossy compression ratios of 15:1 to 20:1 are still acceptable for medical images [27]. This leads us to conclude that a misdiagnosis in this range is unlikely to happen, and therefore our approach could be evaluated in a clinical trial. The experiments also point out that the behavior and quality degradation for an increased angle between the cameras shows a stable behavior and may be well modeled. The percentage of occluded pixels as a function of the angle between the cameras is smoothly increasing with the angle value. This enables us to ensure a sufficiently high quality for the clinical usage of our technique. Besides the previous discussion, it should be considered that 3D rendering in multi-view imaging is a quite novel research area which may show significant progress in performance and quality in the near future, as it is also a focal point in the introduction of 3D television. This progress can be readily exploited for the medical application discussed in this paper. Future work can further evaluate the trade-off between lossy compression of the transmitted texture and depth maps in order to save bandwidth and its influence on the quality (PSNR) of the interpolated image [20], when applied in a medical context.

Acknowledgements

This work has been done as part of the iGLANCE project. The iGLANCE project is sponsored in the framework of the EUREKA (E! 2365) MEDEA+ program by the French and Dutch national research programs. We would also like to acknowledge all the partners in the iGLANCE project, especially Michel Imbert, Regis Miquel and Emiliano Piccinelli from STMicroelectronics, Luc Verhaegh from Task24, Menno Lindwer and Aleksandar Beric from Silicon Hive, Egor Bondarev from TU/e, Nick de Koning and Dion Cornelissen from Prodrive, Henk Katerberg and Ladislau Posta from Verum, Sander Denissen, Willemien van der Linden and Paul Merkus from Philips, Florian Geffray from 4D View Solutions, Matthieu Millequant from Logica, Frédéric Pétrot and Olivier Muller from TIMA, Edmond Boyer and Radu Horaud from INRIA and all others who contributed.

References

[1] A. T. Stadie, R. A. Kockro, R. Reisch, A. Tropine, S. Boor, P. Stoeter, and A. Perneczky, "Virtual reality system for planning minimally invasive neurosurgery," *Journal of Neurosurgery*, vol. 108, pp. 382–394, 2008.

[2] M. Liévin and E. Keeve, "Stereoscopic augmented reality system for computer-assisted surgery," *International Congress Series, Computer Assisted Radiology and Surgery*, vol. 1230, pp. 107–111, Jun 2001.

[3] H. Liao, T. Inomata, I. Sakuma, and T. Dohi, "3-D augmented reality for MRI-guided surgery using integral videography autostereoscopic image overlay," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 6, pp. 1476–1486, Jun 2010.

[4] H. Liao, H. Ishihara, H. H. Tran, K. Masamune, I. Sakuma, and T. Dohi, "Fusion of laser guidance and 3-D autostereoscopic image overlay for precision-guided surgery," *Lecture Notes in Computer Science, Medical Imaging and Augmented Reality*, vol. 5128, pp. 367–376, 2008.

[5] S. Röhl, S. Speidel, G. Sudra, T. Gehrig, B. P. Müller-Stich, C. Gutt, and R. Dillmann, "A surface model for intraoperative soft tissue registration," *International Journal of Computer Assisted Radiology and Surgery*, vol. 4 Suppl. 1, pp. S106–S107, Jun 2009.

[6] A. Abildgaard, A. Kasid Witwit, J. Skaarud Karlsen, E. A. Jacobsen, B. Tennøe, G. Ringstad, and P. Due-Tønnessen, "An autostereoscopic 3D display can improve visualization of 3D models from intracranial MR angiography," *International Journal of Computer Assisted Radiology and Surgery*, vol. 5, pp. 549–554, 2010.

[7] D. Ruijters and S. Zinger, "IGLANCE: transmission to medical high definition autostereoscopic displays," in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, Potsdam, Germany, May 2009, p. 4 pages.

[8] D. Ruijters, "Integrating autostereoscopic multi-view lenticular displays in minimally invasive angiography," in *Proc. MICCAI AMI-ARCS workshop*, New York, USA, Sep 2008, pp. 87–94.

[9] C. van Berkel, "Image preparation for 3D-LCD," in *Proc. SPIE, Stereoscopic Displays and Virtual Reality Systems VI*, vol. 3639, 1999, pp. 84–91.

[10] N. A. Dodgson, "Autostereo displays: 3D without glasses," in *EID: Electronic Information Displays*, 1997.

[11] D. Ruijters, "Dynamic resolution in GPU-accelerated volume rendering to autostereoscopic multiview lenticular displays," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, p. Article ID 843753, 2009.

[12] T. Hübner and R. Pajarola, "Single-pass multiview volume rendering," in *Proc. IADIS International Conference on Computer Graphics and Visualization (CGV 07)*, Lisbon, Portugal, Jul 2007.

[13] M. Levoy, "Effcient ray tracing of volume data," *ACM Transactions on Graphics*, vol. 9, no. 3, pp. 245–261, 1990.

[14] J. Mensmann, T. Ropinski, and K. H. Hinrichs, "Accelerating volume raycasting using occlusion frustums," in *Proc. Eurographics/IEEE 7th International Symposium on Volume and Point-Based Graphics*, 2008, pp. 147–154.

[15] Y. Morvan, D. Farin, and P. H. N. de With, "System architecture for free-viewpoint video and 3D-TV," *IEEE Transactions on Consumer Electronics*, vol. 54, pp. 925–932, 2008.

[16] M. Tanimoto, "FTV (free viewpoint TV) and creation of ray based image engineering," *ECTI Trans. Electrical Eng., Electronics, Communications*, vol. 6, no. 1, pp. 3–14, Feb 2008.

[17] C. Vázquez and W. J. Tam, "3D-TV: Coding of disocclusions for 2D+depth representation of multiview images," in *Proc. Tenth IASTED International Conference on Computer Graphics and Imaging (CGIM)*, Innsbruck, Austria, 2008, pp. 26–32.

[18] S. Ince and J. Konrad, "Geometry-based estimation of occlusions from video frame pairs," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP05)*, vol. 2, Philadelphia, USA, 2005, pp. ii/933–ii/936.

[19] L. McMillan and R. S. Pizer, "An image based approach to three-dimensional computer graphics," University of North Carolina at Chapel Hill, Tech. Rep. TR97-013, 1997.

[20] L. Do, S. Zinger, and P. H. N. de With, "Quality improving techniques for free-viewpoint DIBR," in *IST / SPIE Electronic Imaging*, vol. 7524, San Jose, USA, Feb 2010, p. 10 pages.

[21] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," in *ACM SIGGRAPH 04*, 2004, pp. 600–608.

[22] Y. Morvan, "Acquisition, compression and rendering of depth and texture for multi-view video," Ph.D. dissertation, Eindhoven University of Technology, 2009.

[23] Y. Mori, N. Fukushima, T. Fujii, and M. Tanimoto, "View generation with 3d warping using depth information for FTV," *Image Communication*, vol. 24, no. 1-2, pp. 65–72, 2009.

[24] S. Zinger, L. Do, and P. H. N. de With, "Free-viewpoint depth image based rendering," *Journal Visual Communication and Image Representation*, vol. 21, no. 5-6, pp. 533–541, 2010.

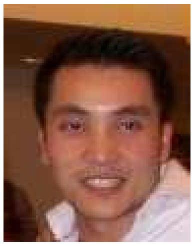
[25] L. Do, S. Zinger, Y. Morvan, and P. H. N. de With, "Quality improving techniques in DIBR for free-viewpoint video," in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, Potsdam, Germany, May 2009, p. 4 pages.

[26] R. Shahnaz, J. F. Walkup, and T. F. Krile, "Image compression in signal-dependent noise," *Appl Opt.*, vol. 38, no. 26, pp. 5560–5567, Sep 1999.

[27] Y.-H. Shiao, T.-J. Chen, K.-S. Chuang, C.-H. Lin, and C.-C. Chuang, "Quality of compressed medical images," *Journal of Digital Imaging*, vol. 20, no. 2, pp. 149–159, Jun 2007.

Svitlana Zinger received the M.Sc. degree in computer science in 2000 from the Radiophysics faculty of the Dnepropetrovsk State University, Ukraine. She received the Ph.D. degree in 2004 from the Ecole Nationale Superieure des Telecommunications, France. Her Ph.D. thesis was on interpolation and resampling of 3D data. In 2005 she was a postdoctoral fellow in the Multimedia and Multilingual Knowledge Engineering Laboratory of the French Atomic Agency, France, where she worked on creation of a large-scale image ontology for content based image retrieval. In 2006-2008 she was a postdoctoral researcher at the Center for Language and Cognition Groningen and an associated researcher at the Artificial Intelligence department in the University of Groningen, the Netherlands, working on information retrieval from handwritten documents. She is currently a postdoc at the Video Coding and Architectures Research group in the Eindhoven University of Technology.

Daniel Ruijters is employed by Philips Medical Systems since 2001. Currently he is working as Sr. Scientist 3D Imaging at the iXR innovation department in Best, the Netherlands. He received his engineering degree at the University of Technology Aachen (RWTH), and performed his master thesis at ENST in Paris. Next to his work for Philips, he has recently finished a joint Ph.D. thesis at the Katholieke Universiteit Leuven and the University of Technology Eindhoven (TU/e). His primary research interest areas are medical image processing, 3D visualization, image registration, fast algorithms and hardware acceleration.



Luat Do In 2009 Luat Do obtained his M.Sc. degree in Electrical Engineering at the Eindhoven University of Technology (TU/e), Eindhoven, The Netherlands. In September 2009, he joined the Video Coding and Architectures group at the TU/e as an Ph.D. student and is currently working on free viewpoint interpolation algorithms which is a part of the iGlance project.

Peter H. N. de With (M'81-SM'97-F'07) graduated in electrical engineering from the University of Technology in Eindhoven and received his Ph.D. degree from the University of Technology Delft, The Netherlands in 1992. He joined Philips Research Labs Eindhoven in 1984, where he became a member of the Magnetic Recording Systems Department. From 1985 to 1993, he was involved in several European projects on SDTV and HDTV recording. In this period, he contributed as a principal coding expert to the DV standardization for digital camcording. In 1994, he became a member of the TV Systems group at Philips Research Eindhoven, where he was leading the design of advanced programmable video architectures. In 1996, he became senior TV systems architect and in 1997, he was appointed as full professor at the University of Mannheim, Germany, at the faculty Computer Engineering. In Mannheim he was heading the chair on Digital Circuitry and Simulation with the emphasis on video systems. Between 2000 and 2007, he was with LogicaCMG (now Logica) in Eindhoven as a principal consultant. Early 2008, he joined CycloMedia Technology, The Netherlands, as vice-president for video technology. Since 2000, he is professor at the University of Technology Eindhoven, at the faculty of Electrical Engineering and leading a chair on Video Coding and Architectures. He has written and co-authored over 200 papers on video coding, architectures and their realization. Regularly, he is a teacher of the Philips Technical Training Centre and for other post-academic courses. In 1995 and 2000, he co-authored papers that received the IEEE CES Transactions Paper Award, and in 2004, the VCIP Best Paper Award. In 1996, he obtained a company Invention Award. In 1997, Philips received the ITVA Award for its contributions to the DV standard. Mr. de With is a Fellow of the IEEE, program committee member of the IEEE CES, ICIP and VCIP, board member of the IEEE Benelux Chapters for Information Theory and Consumer Electronics, co-editor of the historical book of this community, former scientific board member of LogicaCMG, scientific advisor to Philips Research, and of the Dutch Imaging school ASCII, IEEE ISCE and board member of various working groups.