Visualisation of Ultrasound Computer Tomography Breast Dataset

Nicholas Tan Jerome[1], Zhassulan Ateyev[1, 2], Vladislav Lebedev[1, 2], Torsten Hopp[1], Michael Zapf[1], Suren Chilingaryan[1], and Andreas Kopmann[1]

[1] *Karlsruhe Institute of Technology, Eggenstein-Leopoldshafen, Germany*
E-mail: nicholas.jerome@kit.edu
[2] *Tomsk Polytechnic University, Tomsk, Russia*

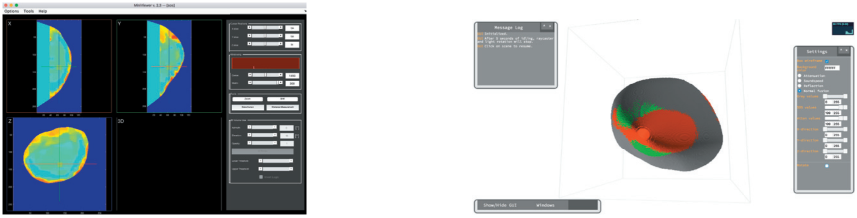
Abstract

Medical visualisation plays a vital role in diagnosing and detecting early symptoms. In particular, visualising the anatomy of breast model allows doctors or practitioners to identify first signs of the breast cancer. However, despite the advancement in visualisation techniques, most standard visualisation approaches in the medical field still rely on analysing 2D images which lack spatial information. In this paper, we present an interactive web-based 3D visualisation tool for ultrasound computer tomography (USCT) breast dataset. We base our implementation on the Web-based Graphics Language (WebGL) technology that utilises the GPU parallel architecture. The tool serves as a common platform among research collaborators to analyse and share findings on their dataset. We render the data using state-of-the-art algorithms of interactive computer graphics and produce results with quality comparable to the desktop application. Aside from that, our tool enables researchers to perform arbitrary view slicing, modality thresholding and multiple rendering modes. In the evaluation, our tool maintains an interactive frame rate above 30 fps on a standard desktop.

Keywords: Web-based Visualization, Scientific Visualization, Multiresolution Slicemap, Normal fusion

1 Introduction

A most common cancer among woman is the breast cancer [1]. In the United States, the breast cancer continues to rank second after lung cancer [2], and significant efforts have been undertaken to improve early breast cancer detection. Primarily due to the improvement in screening methods and treatments, the mortality from breast cancer has declined steadily since 1995 [3].



Message Log

GUI Initialized.
GUI After 5 seconds of idling, raycaster and light rotation will stop.
GUI Click on scene to resume.

Settings

Box wireframe ☑
Background color FFFFFF
○ Attenuation
○ Soundspeed
○ Reflection
● Normal fusion
Grey values 0 255
SOS values 199 255
Atten values 199 255
X-direction 0 255
Y-direction 0 255
Z-direction 0 255
Rotate □

(a)

(b)

Figure 1: The left image shows the Matlab MiniViewer user interface [5] showing the speed of sound modality ranging from 1300 m s^{-1} to 1600 m s^{-1}. The right image shows the screenshot of the 3D web-based visualisation tool [15].

A somewhat new and attractive screening method is the ultrasound computer tomography method (USCT) which does not use any ionisation radiation, and yet produces high-quality 3D dataset using the 3D synthetic aperture focusing technique [4].

Despite the benefits of the USCT method, most of the research focused on comparing the screening results to the more familiar mammography screening methods [6, 7]. Although visualisation is a crucial step in diagnosing the dataset, less attention was given to improving visualisation techniques which could potentially improve data interpretation. The current state of visualisation tools still bases on analysing a stack of 2D slice images and rely on specific software dependencies, i.e., Matlab MiniViewer [5]. Such approach requires doctors to mentally imagine the breast anatomy based on a series of 2D images which could lead to mental fatigue.

Our goal is to create a web-based 3D visualisation of the breast dataset which allows remote and collaborative visualisation. On the one hand, we compromise for lesser quality on client devices with low GPU requirements. On the other hand, we adopted the state-of-the-art algorithms in computer graphics to interactively render the breast dataset using direct volume rendering and surface rendering on client devices with high GPU requirements. To aid the analysis process, we integrated data fusion methods where we merge multimodal information into a single volume. In particular, we integrated the *normal fusion* approach [12] which projects the region of interest along the *normal* direction on top of the surface, and the *image level intermixing* approach [16] which blends the multimodal information by varying its opacity level. Furthermore, our tool allows users to perform arbitrary view slicing, modality grey values thresholding and multiple rendering modes.

In this paper, we present a web-based visualisation tool (Figure 1b) that incorporates interactive rendering of the USCT data. The tool can serve a broad range of client devices, ranging from mobile phone to powerful workstation.

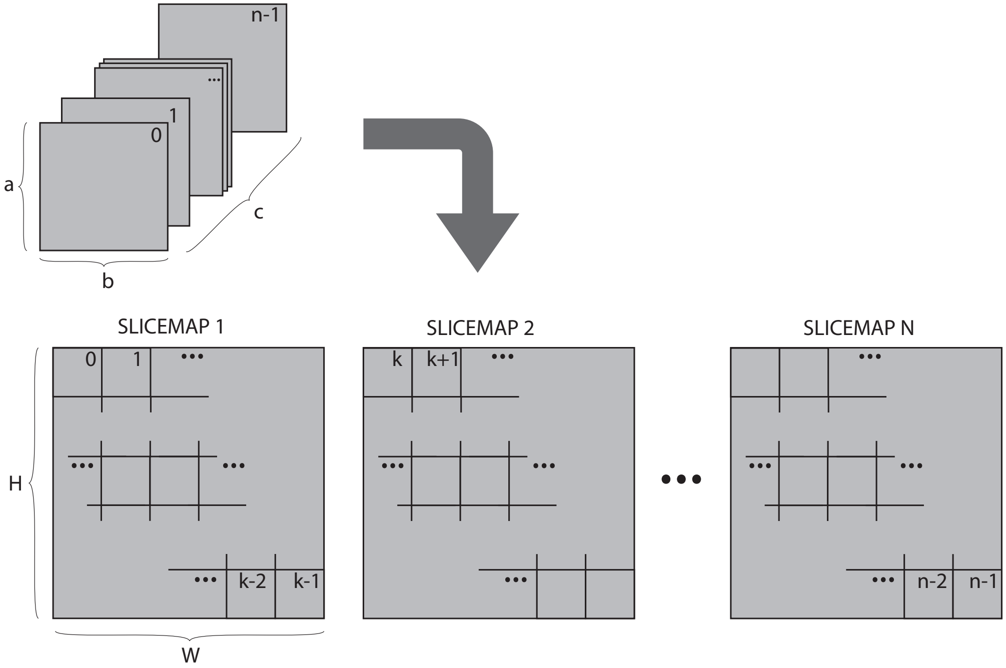


Figure 2: The composition of slicemaps: A 3D volume (image slices) is arranged into multiple slicemaps.

2 Methods

Our approach to design a web-based visualisation tool follows an overarching principle: visual quality and interactive scalability should be limited by the capability of the target client and network. With varying computational resources at the client side, no single visualisation method provides the best performance and the best rendering quality at the same time. Moreover, the limited network bandwidth is also a primary consideration. Although rendering original data may yield the best visual quality, but transferring large size of data may result in a less responsive tool. In this section, we first describe our approaches to find a balance between visual quality and interactive response. The latter parts cover the visualisation approaches and user interactions that are integrated into our tool.

2.1 Input Data

Often, a tomographic dataset presents itself as a stack of image slices, with each slice containing a cross-sectional image that represents an intersection of the volume. Using the standard volume rendering technique, we often load the data into the GPU 3D texture to take advantage of the hardware interpolation which is fast and efficient. However, not all browser supports

3D texture which motivates us to emulate the feature in the Graphics Processing Unit (GPU) shaders [14, 15]. A straightforward approach would be loading input data into the 2D texture memory, but the browser technology further limits the number of images (*texture unit*) and size of each image (*texture size*). These limitations vary across different client devices. For instance, a *Nexus 5* Android phone (GPU: Adreno 330) can accept up to 16 images with each image must not exceed the image resolution of 4096^2 texels.

Hence, we transform the image slices into *slicemap* [15] which comprises a series of 2D cross-section images stacked in a mosaic gridded format. Figure 2 shows the composition of slicemaps from a series of image slices. Here, we generate multiple slicemaps to maximise the capacity of the allowable texture memory in the GPU. Figure 3 shows a slicemap created from the USCT dataset. The USCT dataset consists of three modalities: *sound speed*, *attenuation*, and *reflection*. To this end, we map each modality into its respective colour channel. We express the colour representation of the pixel within the slicemap by a three-tuple $(R,G,B) \in [0,1]$, where $SoundSpeed \mapsto R$, $attenuation \mapsto G$, and $reflection \mapsto B$.

To address the diverse client requirements, we introduce the use of multiresolution slicemap—a hierachy of mutiresolution slicemaps differing in image resolution. The main idea is to transfer a suitable size of data that the best performance at the client. We characterise the client by its *texture unit* and *texture size* properties. To deal with the bandwidth limitation, we included the load-on-demand approach where low-resolution data is firstly served followed by a high-resolution data loading in the background [15].

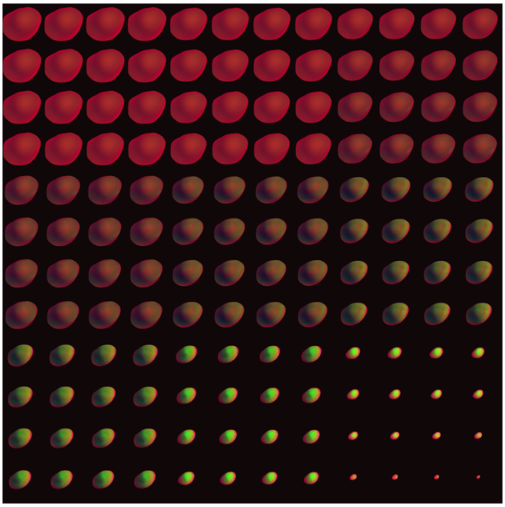


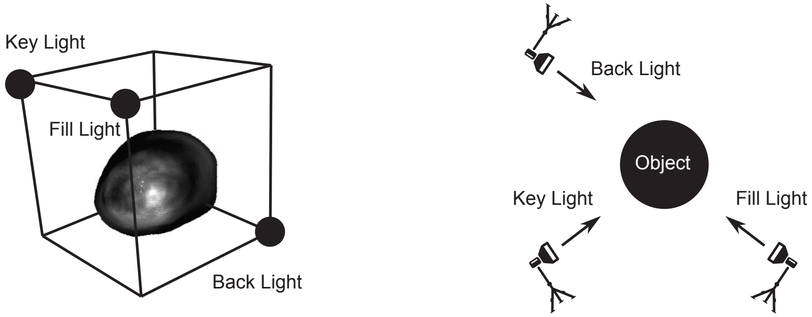
Figure 3: A 12×12 slicemap of a breast dataset. A mosaic-format image comprises a stack of 2D slice images. The iteration of the stack images is from left-right and top-bottom. Each slice consists of three modalities—*sound speed*, *attenuation* and *reflection*—that are mapped onto the colour channels respectively.

2.2 Visualisation Methods

2.2.1 Direct Volume Rendering

The objective of visualising the USCT dataset is to allow doctors or researchers to identify possible tumour regions efficiently. Hence, showing the spatial information of the dataset is crucial. For this reason, we incorporate direct volume and surface renderings as the main visualisation methods. The direct volume rendering is commonly used to display the scientific dataset [8]. In particular, we adopted the particle model variation described by Max [9] where the intensity at each sampled position is constant along the viewing ray direction. Thus, resulting in an image similar to an X-ray negative—brightest where the data is most dense but saturates at the maximum intensity as can be seen in Figure 4a.

Furthermore, we further improve the visual quality by adopting the real-world light simulations — reflection and refraction. In particular, we include the empirical Phong illumination model which describes the light interaction on each surface point [10]. Inspired by the lighting setup in photography studio [11], we mimic the lighting setup to provide an adequate lighting source to the data model. Since our model lies within a bounding box, we place our lights at the corners of the box (Figure 4a). The lighting design consists three light components: (1) key light, (2) fill light, and (3) back light (Figure 4b). Typically in a photography studio, the lightings and the model are static while the photographer moves around the model to find the best shooting angle. In contrast to our setup, the viewer is static while the lightings and model rotate along the model axes.



(a) (b)

Figure 4: The lighting positions within the bounding box. The key light, fill light and back light are attached to the three corners of our model to provide adequate lightings on the data.

2.2.2 Data Fusion Approach

Whenever we visualise the multimodal USCT dataset, we get three full range of volumes—each volume representing each modality. Despite having 3D models, manually analysing multiple datasets is cumbersome. Instead, we merge the three modalities into a single representation for better data interpretation (data fusion). Mainly, we allow fusion scheme which determines the projection of each modality, i.e., Ranger's fusion scheme [6]. In this section, we will discuss on two data fusion approaches where we utilise both surface and volume rendering techniques interchangeably: the *normal fusion* approach [12] and the *image level intermixing* approach [16] (Figure 5). Throughout the data fusion approaches, we label thresholded region of *sound speed* as orange colour and *attenuation* as green colour.

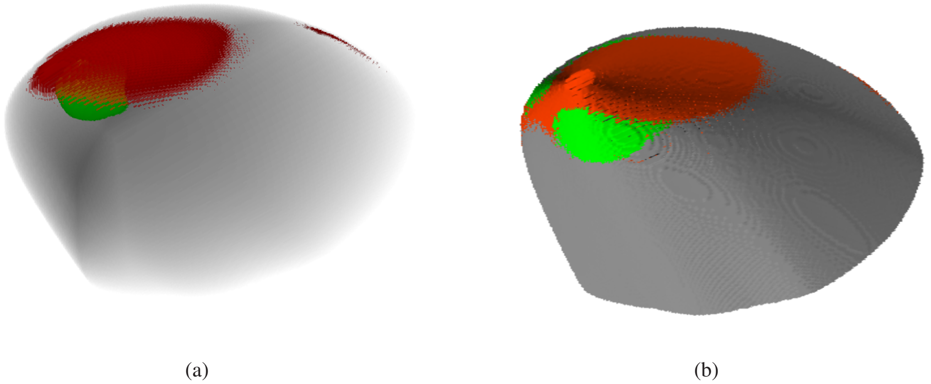


Figure 5: The data fusion approaches: (a) *image level intermixing* and (b) *normal fusion*.

The image level intermixing: In the rendering process (Algorithm 1), we combine the information stored in the slicemap to produce the final image. Mainly, we use the direct volume rendering and extend the classification step where we choose values at each point selectively. The *sound speed* modality is prioritised first, followed by the *attenuation* modality and lastly the *reflection* modality. Firstly, we set the opacity of the background base on the *reflection* modality where the structure of the data can be determined. Then, we look at whether the *sound speed* or the *attenuation* falls within the predefined threshold regions (regions of interest). In the case where any of the two modalities are within the region, we set the colour and opacity of the voxel to the corresponding modality opacity and its predefined colour according to the modality priority.

The normal fusion approach: Algorithm 2 describes the process of projecting the interior information of the volume onto the surface. Here, we utilise volume rendering to collect information within the volume and surface rendering to produce a realistic image. Similar to the *image level intermixing*, we prioritise the *sound speed* modality over *attenuation* modality, and we use *reflection* modality to present the structure information. However, if there is an overlapping region, we show the *sound speed* modality only. To produce the final *normal fusion* result, we firstly cast a ray into the bounding box until we hit the surface of the object. Then, we calculate the normal vector at the intersection point. Rather than continuing in the ray direction, we continue to traverse further in the normal direction instead (secondary ray). Along the secondary ray, we return the orange label if we detect any *sound speed* within the predefined threshold. Otherwise, we return green label if we find *attenuation* within the predefined threshold with no *sound speed* presence. With the colour label ready, we perform the surface rendering using the returned label as our ambient colour.

```
Ray setup (entry position and ray direction);
while ray in volume do
    Interpolate data value at current position;
    if current value (sound speed) within sound speed threshold then
        colour ← orange;
        opacity ← current sound speed opacity;
    else
        if current value (attenuation) within attenuation threshold then
            colour ← green;
            opacity ← current attenuation opacity;
        else
            colour ← grey;
            opacity ← current reflection opacity;
        end
    end
    Perform compositing (colour and opacity);
    Propagate position along ray;
end
```

Algorithm 1: Image level intermixing: pseudo code

2.3 Data Interactions

To study and explore the data, we enable arbitrary model view slicing, grey values thresholding, and multiple rendering modes. Since interesting information often lies within the volume, we allow users to slice and inspect the interior of the volume. The slicing angle can be any arbitrary position which the user can freely define. Given the many visualisation approaches adopted within the tool, we allow the user to change the visualisation mode interactively. Aside, the user can select the grey value threshold and show only the region of interest. Depending on the visualisation mode, the threshold parameter may filter out data from the view or define a particular region to be displayed. We also allow the user to interact with the 3D model by including rotation and zoom features.

```
Ray setup (entry position and ray direction);
Define W as length of the cube;
while ray in volume do
    if intersect then
        calculate normal direction;
        read current position and value;
        output ← grey;
        Secondary ray setup(current position and normal direction);
        while secondary ray less than $\frac{W}{2}$ do
            if current value within sound speed threshold then
                output ← orange;
                break;
            end
            if current value within attenuation threshold then
                output ← green;
            end
        end
        ambient ← output;
        Perform compositing (ambient + diffuse + specular);
        break;
    else
        propagate position along ray
    end
end
```

Algorithm 2: Normal fusion: pseudo code

3 Evaluations

The basis for all frame rate comparison is the client's local server which is primarily affected by its available GPU resource. Hence, to measure the efficiency and performance of our tool, we evaluate the mentioned visualisation methods—direct volume rendering (DVR), the *image level intermixing* approach (ILI), the *normal fusion* approach (NF)—on several clients with different GPU resources. The methods are tested on (i) a mobile phone (GPU: Adreno 510), (ii) a standard desktop (GPU: integrated graphics card HD4000), (iii) a laptop (GPU: GT750M), and (iv) a powerful workstation (GPU: Tesla C2). We use the USCT data with the size of $256 \times 256 \times 144$ which is transformed into a single slicemap (Figure 3). The three modalities are encoded into the colour channels of the slicemap.

Base on the results shown in Figure 6, the direct volume rendering has the best performance due to its simplicity. There is no classification step nor any fusion scheme involved. In the case of the ILI and NF, both fusion methods involve priority-based modality selection to determine the visual output. The ILI is based solely on volume rendering algorithm and extends the classification step; whereas the NF consists of both volume and surface renderings. Moreover, the NF approach involves a primary ray in the view direction and a secondary ray in the normal direction. In comparision with the ILI approach, the addtional secondary ray in the NF approach results in higher traversal steps. However, the performance of the ILI and NF is comparable which suggests the priority-based selection in the NF serves as an *early secondary ray termination*—the secondary ray terminates upon the detection of *sound speed* modality.

With exception to the mobile device, our tool resulted in an overall frame rate higher than 30 fps which suggest the suitability of our tool in clinic or laboratory environments. Worth noting is the performance shown by the desktop with only an integrated graphics card (integrated GPU HD4000) with no dedicated GPU. Hence, we can conclude that our tool is capable of running on any modern desktop with acceptable performance. On the other hand, the usage of the web-based 3D tool in mobile phones in the current state is not feasible. We can optimise the existing algorithm by reducing the sampling step number along with the primary ray. Also, we can adopt less resource demanding algorithm such as additive blending. To minimise the processing load at the client, we can shift the data-intensive operation to the server side such as precalculating normal vectors. Also, reducing the data size can improve the client side performance immensely but at the cost of the visual quality [15].

4 Conclusions and Future Work

We have presented a web-based 3D visualisation tool that can be served by a broad range of client devices. Our tool supports the direct volume rendering, the *image level intermixing* approach, and the *normal fusion* approach which are useful in identifying the interesting region of the USCT data. Not only web-based platform encourages collaborative research, but the visual results were promising and comparable to desktop applications.

Future work of our tool encompasses several opportunities. Firstly, given the simple geometry of the USCT dataset, fusing the multimodal data is much more interesting. In this paper, we have shown a small fraction of the many multimodality visualisations imaginable. For example, we can consider the work by Bramon [17] to fuse multimodal data using the mutual information approach which requires probability map of the interested anatomy features. Since our input data are inherently images, we can improve the network latency by introducing compression schemes such as LZO algorithm.

Throughout this paper, we assume a simple client-server architecture with no high-performance computer (HPC). With hardware commodity being affordable, it is an exciting prospect to improve the image quality and network latency by deploying an image streaming approach, which the approach shifts computation intensive processes to the server and send the resulting image—small memory footprint—to the client. Furthermore, a new trend arises where many applications are emphasising on the visual analytic system. Having only multimodality visualisation is maybe useful, but integrating more domain knowledge into the visualisation system can enlighten the doctors or practitioners in their daily work.

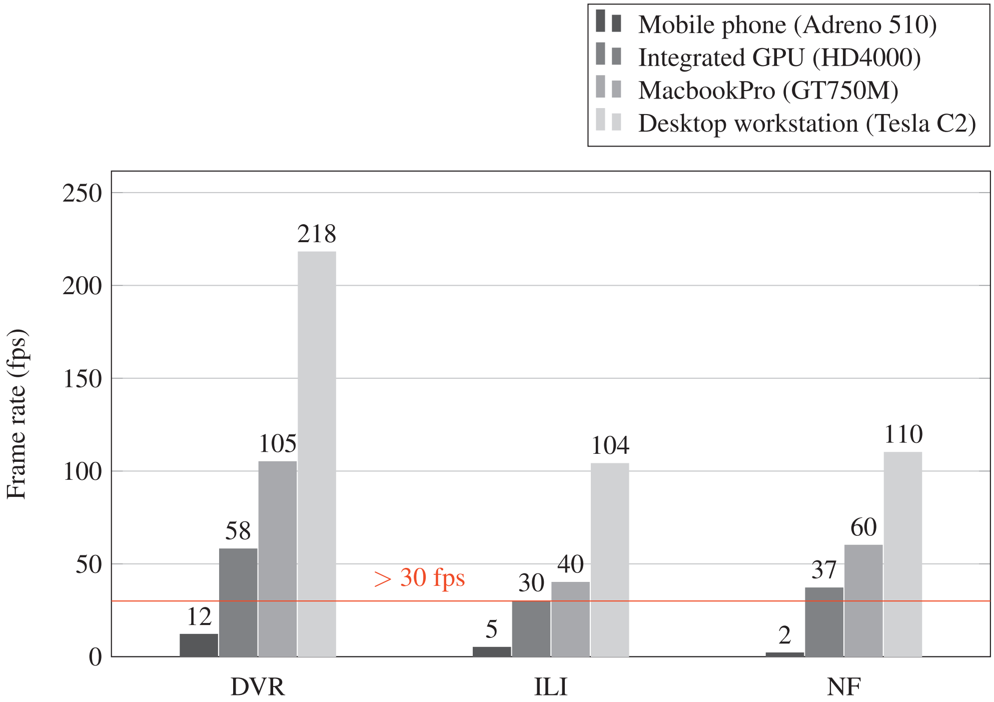


Figure 6: The frame rate of three visualisation methods—the direct volume rendering (DVR), the image level intermixing (ILI) and the normal fusion approach (NF)—tested on the mobile phone, the desktop with integrated GPU, the MacbookPro and the powerful workstation. The higher the frame rate, the better.

References

[1] L. A. Torre, F. Bray, R. L. Siegel, J. Ferlay, J. Lortet-Tieulent, A. Jemal.: Global cancer statistics, 2012. CA: a cancer journal for clinicians 65(2) (2015) 87-108.

[2] K. C. Oeffinger, E. T. Fontham, R. Etzioni, A. Herzig, J. S. Michaelson, Y.-C. T. Shih, L. C. Walter, T. R. Church, C. R. Flowers, S. J. Lamonte, A. M. Wolf.: Breast cancer screening for women at average risk: 2015 guideline update from the american cancer society. Jama 314 (2015) 1599-1614.

[3] R. L. Siegel, K. D. Miller, A. Jemal.: Cancer statistics, 2016. CA: A Cancer Journal for Clinicians 66 (2016) 7-30.

[4] H. Gemmeke, T. Hopp, M. Zapf, C. Kaiser, N. V. Ruiter.: 3D ultrasound computer tomography: Hardware setup, reconstruction methods and first clinical results. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, (2017).

[5] T. Hopp, G. Schwarzenberg, M. Zapf, N. V. Ruiter.: A MATLAB GUI for the analysis and exploration of signal and image data of an ultrasound computer tomograph. IEEE Advances in Computer-Human Interaction (2008) 53-58.

[6] B. Ranger, P. J. Littrup, N. Duric, P. Chandiwala-Mody, C. Li, S. Schmidt, J. Lupinacci.: Breast ultrasound tomography versus MRI for clinical display of anatomy and tumor rendering: preliminary results. American Journal of Roentgenology 198 (2012) 233-239.

[7] N. V. Ruiter, M. Zapf, R. Dapp, T. Hopp, W. Kaiser, H. Gemmeke.: First results of a clinical study with 3D ultrasound computer tomography. IEEE International Ultrasonics Symposium (2013) 651-654.

[8] M. Levoy.: Display of surfaces from volume data. IEEE Computer graphics and Applications 8 (1988) 29-37.

[9] N. Max.: Optical models for direct volume rendering. IEEE Transactions on Visualization and Computer Graphics 1 (1995) 99-108.

[10] B. T. PHONG, Illumination for computer generated pictures, Commun. ACM, 18 (1975), pp. 311-317.

[11] Y. Zhang, K.-L. Ma.: Lighting design for globally illuminated volume rendering. IEEE transactions on visualization and computer graphics 19 (2013) 2946-2955.

[12] R. Stokking, K. J. Zuiderveld, M. A. Viergever.: Integrated volume visualization of functional image data and anatomical surfaces using normal fusion. Human Brain Mapping 12 (2001) 203-218.

[13] S. D. Roth.: Ray casting for modeling solids. Computer graphics and image processing 18 (1982) 109-144.

[14] J. Congote, A. Segura, L. Kabongo, A. Moreno, J. Posada, O. Ruiz.: Interactive visualization of volumetric data with webgl in real-time. ACM. Proceedings of the 16th International Conference on 3D Web Technology (2011) 137-146.

[15] N. Tan Jerome, S. Chilingaryan, A. Shkarin, A. Kopmann, M. Zapf, A. Lizin, T. Bergmann.: WAVE: A 3D online previewing framework for big data archives. INSTICC SciTePress. Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications 3 (2017) 152-163.

[16] W. Cai, G. Sakas.: Data Intermixing and Multivolume Rendering. Blackwell Publishers Ltd. In Computer Graphics Forum 18(3) (1999) 359-368.

[17] R. Bramon, I. Boada, A. Bardera, J. Rodriguez, M. Feixas, J. Puig, M. Sbert.: Multimodal data fusion based on mutual information. IEEE Transactions on Visualization and Computer Graphics 18(9) (2012) 1574-1587.