Sondre Langeland Hisdal

Frequency Modulated Shading

Master Thesis

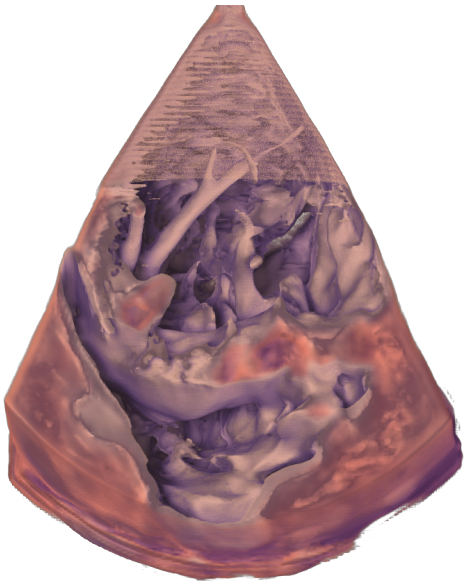
supervised by

Veronika Šoltészová

Ivan Viola

Department of Informatics

University of Bergen



Abstract

In this thesis we describe a frequency modulated shading technique. The technique performs a local area analysis for each voxel in the volume. The analysis creates a local subvolume around the voxel currently being investigated. This subvolume is transformed into frequency domain, where some frequencies are filtered out. The filtered volume is transformed back into spatial domain, where it is used to compute gradients. The filtered gradients are then used in combination with a shadow renderer, to provide both gradient based shading and shadows.

Contents

CHAPTER

1 Introduction

Nature and Nature's laws lay
hid in night: God said, 'Let
Newton be!' and all was light..

epitaph of Sir Isaac Newton by
Alexander Pope

Medical imaging is the process of creating images of the human anatomy and physiology, and it is much used for clinical purposes. An example of this is the use of X-ray to discover bone fractures or to discover cavities in teeth. Another important use of medical imaging is to understand how different systems in the body behave. This can for instance be perfusion imaging which is used to see how blood is delivered to a particular part of the body. Based on the observation of the blood flow, the clinician can discover cancerous cells in the body. Medical imaging allows clinicians to get a better understanding of the processes occurring in patients, and it can be an important aid when setting a diagnosis. There are several different scanning modalities used for medical imaging. All the different modalities have their own strengths and weaknesses. In the following paragraphs we will discuss some of these modalities.

Ultrasonography (US) is a medical imaging technique that uses the properties of sound to generate images of the scanned tissue. US is an important tool for medical doctors when diagnosing a patient [20]. US has many benefits, which

make it a popular scanning modality. US is non-invasive and it is not harmful for the patient. US can provide detailed images of the tissue. It has a high temporal resolution, meaning that using ultrasound one can accurately locate structures at a particular time. As well as a high temporal resolution ultrasound can have a very good spatial resolution. Unlike other scanning modalities it is portable, and can be applied bedside, which is a major benefit for patients that may be to sick or injured to be moved safely. Being able to use hand-carried [9] scanners and pocket sized scanners [10] make ultrasound usable in very many situations where other modalities can not be used, such as in the field where larger equipment can not be brought.

A US-probe is used to generate a pulse of ultrasound waves. The sound waves created by this pulse traverse the tissue of a patient. The sound waves are reflected by structures in the patient tissue, and the echoes are measured by the US-probe. From the measured echoes images and volumes can be constructed. Ultrasound is widely used as a medical imaging tool, and different types of US can be used depending on the situation. B-Mode US is used in obstetric ultrasound, where ultrasound is used during pregnancy to visualize the development of the fetus. The same technique is also used for scanning other parts of the body such as the liver and the heart. Other modes of ultrasound include Doppler ultrasound and ultrasound elastography. Doppler ultrasound uses the Doppler-effect to measure blood flow, while US Elastography is used to measure strain in tissue.

Other structural medical imaging modalities include X-ray, CT and MRI. X-ray is still widely used by doctors when looking for bone fractures, as well as by dentists when looking for cavities in their patients teeth. Computed tomography (CT) uses a series of X-ray projections to create a 3D volume of the scanned object. CT is a very important tool in medical imaging. It is used when looking for a wide range of illnesses, and can be used to scan the entire human body. Magnetic resonance imaging (MRI) is another medical imaging technique. MRI uses a magnetic field to systematically manipulate the magnetization of atoms in the human body. A scanner measures the alterations in the magnetic fields, and from this it is possible to construct an image or a volume of the scanned area. MRI provides good contrast between different types of soft tissues. Due to this, it is very often used for scanning areas, such as the brain, where contrast between

Figure 1.1: Acuson x300 Ultrasound machine [19].

different soft tissues is important.

Because of the large area of applicability of US it is important to get as much valuable information out of the ultrasound data as possible. Having a best possible 3D visualization of ultrasound data will help doctors when diagnosing patients. The most dominant issue with ultrasound is the noise in the data. Ultrasound has a low signal-to-noise ratio which means that there is a lot of noise in the data. Noise in the data makes it difficult to discern which parts of the data that contain valuable information and which parts that are just noise. Due to this, an important part of the visualization pipeline when working with ultrasound data, is minimizing the effects of noise. Another problem when dealing with 3D ultrasound is speckle. Speckle is an effect of the reflection from the surface. In 2D it gives valuable information about the texture of the structure, but this property can not be exploited in 3D. The reduction of noise and speckle artifacts in 3D visualization is the main focus of this thesis and it is a much studied topic in the ultrasound imaging research community. In the next section, we will give a short description of the physics behind ultrasound scanning and describe the problems that occur due to the natural properties of ultrasound.

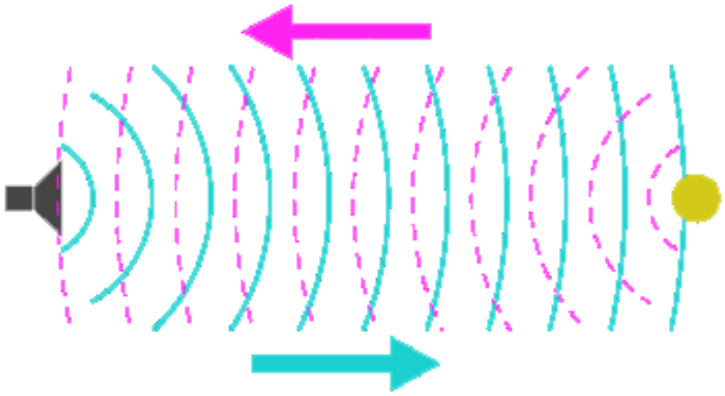


Figure 1.2: Illustration of the ultrasound principle. The US-node represented by the speaker icon emits sonic waves, represented by the cyan color. When these waves reach a structure, like the one represented by the yellow color, parts of the waves are reflected. The reflected waves, represented by the purple dotted lines, traverses back through the tissue until they reach the node, where they are measured.

1.1 Physics of ultrasound

Basic principle

Ultrasound is used to describe sound waves with a frequency above the audible frequency range, i.e. frequencies above 15-20 kHz. The use of ultrasound in medicine usually operates with frequencies between 2-10 MHz. In medical ultrasound a US-probe is placed on the skin of a patient. The probe emits sound waves, the sound waves traverses the tissue of the patient and are reflected by structures in the tissue. The reflected waves traverses the tissue back to the US-probe, which measures the reflected waves. Figure 1.2 illustrates this basic principle. The probe, represented by the gray speaker icon, emits sound waves, represented by the cyan color in the figure. When the waves reach the structure, represented by the yellow circle, some of the waves are reflected. The reflected waves are represented by the purple dotted lines in the figure. The reflected waves are then measured by the probe.

Resolution

There are two types of resolution to consider when working with ultrasound, radial resolution and lateral resolution. Radial resolution is the resolution along the direction of the wave, and lateral resolution is the resolution transverse to the wave. The radial resolution depends on the frequency of the wave, and the lateral resolution depends on the width of the initial pulse. The radial resolution is inversely proportional to the frequency of the wave, meaning that the higher the frequency, the better the radial resolution. Unfortunately, attenuation of the wave increases with the frequency. As the wave traverses the tissue, some of the energy is absorbed, or attenuated. This makes the wave lose its energy as it traverses through tissue, and the higher the frequency is, the bigger part of the energy is absorbed. Due to this, it is necessary to find a balance between depth and resolution. Naturally, it is beneficial to have a high resolution, but using a too high frequency may prevent the wave from penetrating deep enough to get to the target structures. Scanning deep organs requires a lower frequency than scanning shallow tissue. Due to this attenuation factor different frequencies are used when scanning different structures. Frequencies between 2.5-5 MHz are commonly used when scanning deep organs in adults. For more shallow scans higher frequencies can be used. Frequencies up to 40MHz have been used in intra-vascular imaging. Reflections from deep structures naturally have a lower amplitude than reflections from close structures, due to the energy absorption in the tissue. To compensate for this, an amplifier is introduced that amplifies the measured reflections depending on time past from the original pulse. This amplifier is called time gain-compensation.

Reverberation

At the boundary of two tissues, the pulse will be partially reflected and partially transmitted. How much is transmitted and how much is reflected depends on the difference in acoustic impedance of the two boundary tissues. Acoustic impedance is a characteristic of the tissue material. Different tissues such as fat and muscle have different acoustic impedances. At the interface between two

materials with differing acoustic impedance some of the sound wave will be reflected. The acoustic impedance of a material is defined as,

$$Z = \frac{P}{U}, \tag{1.1}$$

where Z is the acoustic impedance, P is the pressure amplitude and U is the particle velocity in the material. At the interface between two soft tissues most of the energy is transmitted, since the difference in impedance is low, while at tissue-bone interface most of the energy is reflected. This causes the problem of shadowing effect from bones as mentioned previously.

When a wave reaches the second interface it will be partially reflected, creating a backwards wave. When this backwards wave reaches the first interface it will again be partially reflected. This creates a second forwards wave. As before when this wave reaches the second interface it will be partially reflected. This creates multiple waves from multiple reflections and the produced effect is called reverberations. Reverberations create an effect called ghosting, which makes things appear in the image which are not physically there.

Speckle

As mentioned previously, there are two types of resolutions to consider, i.e. the radial resolution and the lateral resolution. The lateral resolution depends on the width of the beam. If there are two targets so close to each other that they are hit by the same pulse, there will be interference between the reflected waves from the two targets. In a homogeneous tissue the interference creates a pattern which is called speckle. The speckle texture depends on the frequency, and using a higher frequency will create a more finely grained speckle texture. Since speckle is dependent on the frequency and on the tissue material, it is not considered to be random noise. Speckle is, in fact, used to identify types of tissue when interpreting at a 2D-slice. When visualizing the entire acquired volume, speckle does not provide any useful information and is considered to be an artifact.

1.2 Scope of the thesis

The goal of the thesis is to improve the visualization of 3D medical ultrasound data by improving the rendering step in the visualization pipeline. To achieve this, we explore the use of different illumination techniques. We also investigate the frequency spectrum of the data and look at ways to reduce the noise by frequency modulation.

The thesis is structured as follows. In Chapter 2, an overview of the current state of the art in visualization of volumetric data is given. In Chapter 3, the techniques which have used to get a best possible visualization of 3D medical ultrasound data are described. In Chapter 4, details on how the techniques from the previous chapter have been implemented is given. In Chapter 5, results of our work is shown, and finally in Chapter 6, the thesis is concluded.

CHAPTER

2 State of the Art

Copy from one, it's plagiarism;
copy from two, it's research.

Wilson Mizner

In this chapter we give an overview of existing work in relevant fields, such as preprocessing of ultrasound data, and rendering. Visualization of ultrasound data is difficult because of its natural properties such as noise and speckle. Preprocessing of the data can be useful to minimize the effects of these properties, and us discussed below. Using advanced rendering techniques can further minimize the effects of these properties.

2.1 Preprocessing and Enhancement of Ultrasound Data

An important part of handling ultrasound data is data enhancement. Ultrasound data has some properties that make it challenging interpret. It has a low dynamic range, a low signal-to-noise ratio and in 3D visualization effects of speckle is no longer a valuable property [24]. Due to these properties it is important to enhance the data by reducing the negative effects of the aforementioned properties, and important works are reviewed in the following sections.

Reconstruction

To acquire volumes of ultrasound data either a 3D-probe or a freehand ultrasound can be used by putting the images into a spatial context. Techniques to create a correct volumes from freehand ultrasound images exist. Garret et al. used a binary positioning tree to order the images correctly on the fly from incoming 2D ultrasound slices [7]. To reduce the problem of cluttering in large volumes Gee et al. presented a re-slicing technique to create narrow-band volumes. These volumes contain less elements and have less cluttering [8].

Speckle reduction

Speckle patterns are useful when looking at 2D images, since it contains information about the texture of the tissue. This is valuable information for doctors who are investigating the images. In 3D, the speckle gives no valuable information and is considered to be an artifact. It would be beneficial to remove or minimize the effects of speckle when working with 3D data. Techniques for reducing speckle effects have therefore been developed and can improve the visualization of 3D ultrasound data. A simple approach to reduce speckle is to use median filtering. This can be effective, but there have been proposed several more advanced techniques. Adaptive weighted median filter (AWMF) [18] applies integer weighted coefficients to control the influence of the pixel values in the median calculations. Homogeneous region growing mean filter (HRGMF) [15] uses a region growing technique based on a constant homogenity value, to obtain a homogeneous area, before smoothing the area with a mean filter. Karaman et al. [12] presented an adaptive smoothing technique for suppressing speckle in B-mode ultrasound data. They apply a local statistics based region growing technique to obtain homogeneous region around a center pixel. Mean or median smoothing techniques are then applied to this homogeneous area. Chen et al. [5] uses an adaptive filtering algorithm.

Frequency analysis

The Fourier transform is a much used tool in image processing. Using the Discrete Fourier transform (DFT) finite sequences of discrete data can be transformed into frequency domain. For transforming multidimensional data into frequency domain, the DFT are run once for each dimension. The frequency representation of the data can be used to see how the data changes. Low frequencies correspond to small changes that occur over time, and higher frequencies correspond to more sudden changes. Inspecting volumetric data from ultrasound low frequencies correspond to homogeneous regions with little change. Higher frequencies correspond to where there are more sudden changes in the data, such as on the interface between two structures, or they correspond to noise in the data.

Frequency modulation can be used to reduce noise in the data. Vucini et al. [30] introduce the concept of Frequency-based transfer functions. They transform the volumetric data into frequency domain using the Fast Fourier Transform, an efficient version of DFT. They then modify the amplitude of the frequencies according to some transfer function. A standard one-dimensional transfer function in spatial domain, maps scalar values to opacities and color. Transformed into frequency domain, each location in the volume consists of a real value and an imaginary value. There is no intuitive way of mapping the values to a trans-

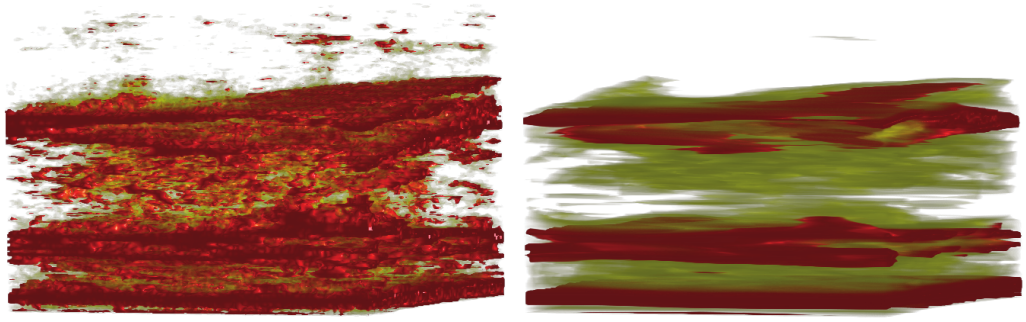


Figure 2.1: A 3D abdominal ultrasound dataset with no frequency modulation (left), compared to the same dataset with modulated frequencies (right).Images from *Enhancing Visualization with Real-Time Frequency-based Transfer Functions* [30]

fer function. Instead Vucini et al. uses the relative distance to the zero frequency (DC-component) in their transfer function. The higher the frequencies, the farther away they are from the DC-component. The transfer function maps the relative distance to the DC-component to a scaling factor. Both, the real and imaginary values, are scaled by their corresponding scaling factor. Using this approach, they created a one-dimensional transfer function for frequency domain that are intuitive to change. Figure 2.1 shows a 3D abdominal ultrasound dataset with no frequency modulation (left), compared to the same dataset with modulated frequencies using their approach (right).

2.2 Rendering of Ultrasound Data

The last stage of the visualization pipeline is rendering. This is where a visual representation of the data is created. How light interacts with the scene, is a very important part of how we perceive it. Naturally, to achieve a good volume visualization we have to consider light propagation in the volume. Shading using surface normals gives surface perception. This improves the shape perception and is a very important tool in rendering. Shadows provide depth cues.

We separate between different types of illumination models depending on their scope of influence. Local illumination models only take into account the immediate neighborhood when shading a voxel, or surface. Semi-global illumination models take into account at a larger area around a the geometry, but are still limited to a local area. Global illumination models take into account the entire scene. Global illumination models are expensive to compute and approximations to these have been developed.

Local Illumination

Local illumination models use local properties when creating shading. Levoy, who introduced the concept of direct volume rendering, proposed the use of gradient based shading. Gradients can be used to evaluate a surface-based local illumination, according to the Phong illumination model [16]. The Phong illumi-

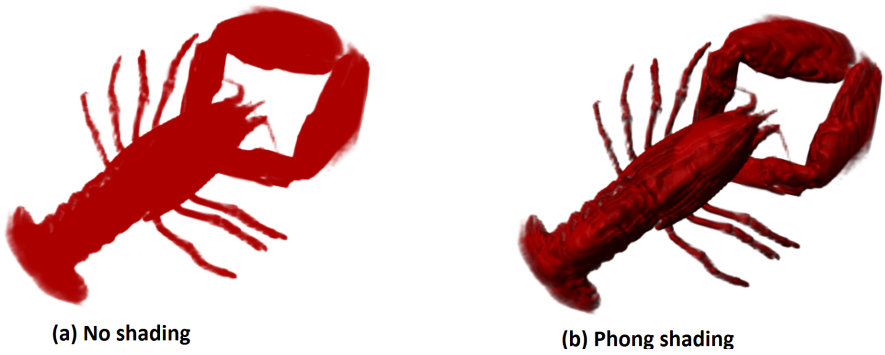


Figure 2.2: No shading (a) vs. Phong Shading (b)

nation model uses surface normals to create shading. Gradients can be used as a substitute for surface normals in volume rendering. This has become a widely used approach in volume rendering. Local illumination models are computationally inexpensive, and already provide structural information. Figure 2.2 shows the lobster dataset; (a) rendered without shading and (b) with Phong shading . Since the estimation of gradients is based on finite differences, it is very sensitive to noise and often gives poor results for noisy data. As stated before, ultrasound has a low signal-to-noise ratio and naturally gradient estimation suffers from this. In addition, gradients do not give any useful information in regions that are nearly homogeneous. Due to the poor quality of gradient-shading in volume rendering of ultrasound data, this technique is not often used. In fact, no-shading is usually a preferred option as compared to standard gradient-based shading. Illumination models that look at more than the very immediate neighborhood can be used to provide more visual cues, and improve the visualization.

Semi-Global Illumination

Semi-global Illumination models considers a larger area than local illumination models, but are still limited to a local area around the voxel being evaluated.

Ambient occlusion, introduced by Zhukov et al. [31], obscure a point in space by the environment around the point. This technique has been recently incorpo-

rated in volume visualization. Hernell et al. proposed a local ambient occlusion (LAO) technique which uses the area around each voxel to modify the ambient lighting for that voxel [11]. The traditional rendering equation used for volume rendering consists of three parts, an ambient, a diffuse, and a specular part. The ambient part is typically set to a constant factor. LAO enhance the ambient part of the equation to include occlusion from local features. LAO considers the occlusion from voxels within a sphere centered around the voxel being evaluated. Since LAO only looks at a local area it avoids global shadows that might cause information hiding, while still getting shadows from the local area. To modify the ambient term using LAO, a number of rays are shot from the voxel being evaluated and the ambient term is modified depending on the opacity of the sample points. Each ray is evaluated according to Equation 2.1, where $A_k(x)$ is the ambient term according to ray k, M is the number of samples, and α_i is the opacity at sample i.

$$A_k(x) = \frac{1}{M} \sum_{m=1}^{M} \prod_{i=1}^{m-1} (1 - \alpha_i) \tag{2.1}$$

The final value of the ambient term is given by the normalized sum of all individual ray values, according to Equation 2.2. $A(x)$ is the ambient term of voxel x, and K is the number of rays.

$$A(x) = \frac{1}{K} \sum_{k=1}^{M} A_k(x) \tag{2.2}$$

Figure 2.3 shows resulting shading using local ambient occlusion compared to diffuse Phong shading. The quality of local ambient occlusion shading is improved by considering more directions.

Global Illumination

Global illumination models, as the name indicates, take into account a larger area when computing the shading. Global illumination models are more computationally expensive than local illumination models. They do however result in a more realistic appearance than the local models, and give good spatial perception. Global illumination models are, however, too expensive to compute in real

(a) Diffuse Illumination (23.8 FPS) (b) LAO with 1 direction (21.6 FPS) (c) LAO with 8 directions (21.7 FPS)

Figure 2.3: Images from *Local Ambient Occlusion in Direct Volume Rendering* [11]. Images courtesy of Hernell et al.

time. To lower the computational costs, several techniques for approximations have been proposed.

Kniss et al. describe a technique for creating volumetric shadows and appearance of translucency [14]. Their technique accounts for scattering effects in slice-based volume rendering. Adding shadows to a scene increases realism. Shadows are important monocular and binocular depth cues [22]. Schott et al. have proposed an interactive directional occlusion shading [25]. By restricting occlusion computation in a cone, oriented towards the viewer, they were able to achieve interactive rendering with occlusion effects. Šoltészová et al. generalized the work of Schott et al. [25], for user-defined light directions [28]. They use a tilted cone-shaped phase function which leaves elliptic footprints in an opacity buffer to approximate the light transport. The opacity buffer is incrementally blurred with an elliptic kernel. For rendering they use a slice-based renderer with an additional opacity buffer. This results in high quality soft shadows. Figure 2.4 shows visualizations of a human hand using ray casting (a) and slice-based volume rendering (b). Both (a) and (b) use the Phong illumination model. The last two figures shows directional occlusion shading with the light positioned at the viewer (c), and with the light positioned to the top-left (d). Positioning the light in the top-left corner causes soft shadows to be cast from the fingers which gives strong depth perception. Illumination from the top-left direction is a common approach used by the medical illustrators to give more depth to the image.

Lindemann and Ropinski use spherical harmonic lighting to simulate advanced light material interactions [17]. Ropinski et al. have proposed a volumetric lighting model to mimic the effects of scattering and shadowing [23]. They use slice-

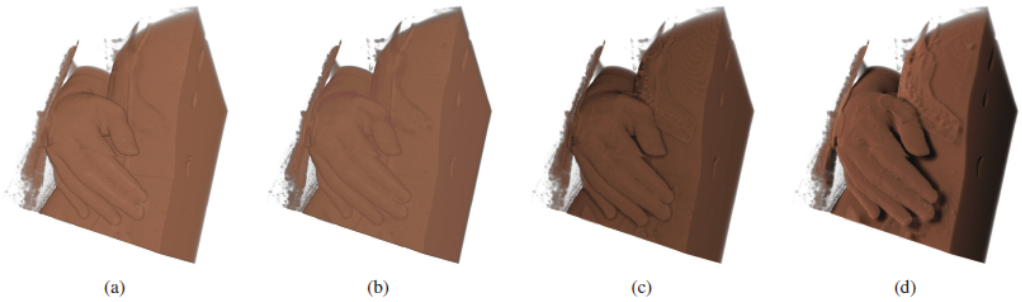


Figure 2.4: A human hand visualized using (a) raycasting with Phong illumination, (b) slice-based volume rendering with Phong illumination, (c) directional occlusion shading with a headlight, and (d) directional occlusion shading illuminated from the top left. Images from *A Multi-directional Occlusion Shading Model for Direct Volume Rendering* [28]. Images courtesy of Šoltészová et al.

based volume rendering from the light sources point of view to generate a light volume and then use ray casting to render the final image.

Adding shadows to a visualization can improve the depth cues, but it can also lead to information hiding. Areas that are largely covered by shadows lose their structural information. Because the shadows darken the shadowed areas, the elements in these areas become less visible, and we lose the detailed information. The color of an object in shadow should be darker than the color of the same object not in shadow. To make the object appear to be in shadow, it is natural to modify the luminance of the object color. However, illustrators use a different approach. To compensate for the problem of over-darkening, they modify the luminance less, and instead add a chromatic part, often blue or of a complementary color to the color of the material. This way the shadowed area is still visible, but the color conveys that it is in shadow. Šoltészová et al. have presented a single-pass method for light scattering in volumes, that incorporates the use of a chromatic element in shadows [29]. They add a scalar attribute, *shadowiness*, to each voxel in the volume. Then they use a shadow transfer function to map the shadowiness to a color and blend factor. The object color is blended with the shadow color depending on the blend factor from the transfer function. This way they circumvent the problem of information hiding by shadows, in the same

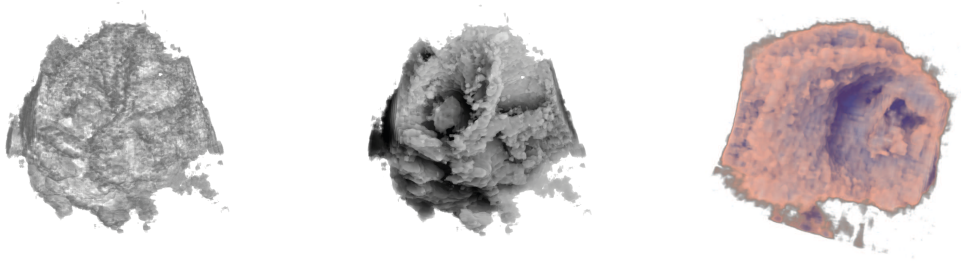


Figure 2.5: Heart dataset visualized using (left) Phong illumination, (middle) technique proposed by Ropinski et al., and (right) the technique propsed by Šoltészová et al. [29] (right). Images from The Ultrasound Visualization Pipeline – A Survey [3]. Images courtesy of Birkeland et al.

way as the illustrators have circumvented them. By having shadows as a color and blending it with the object color, information hiding does not occur while the depth cues from shadows are preserved. Šoltészová et al. have used a blue tone for the shadows, as suggested by medical illustrators. Figure 2.5 shows a 3D US dataset of a heart rendered with Phong illumination (left), with the technique proposed by Ropinski et al. [23] (middle), and with the technique proposed by Šoltészová et al. [29] (right).

Physically-based illumination

Physically-based rendering techniques aim to render scenes with as much realism as possible, by using physical models for the propagation of light. An important aspect of this, is to model reflection of light correctly, and much work has been devoted to this area. The Torrance-Sparrow [26] model simulates surfaces as a collection of mirror-like micro facets. Depending on the distribution of the micro facets and how much the micro facets shadow each other, the reflection is modelled.

PBRT, Physically based rendering toolkit, is a library of photo-realistic rendering techniques [21]. PBRT can be used to create high quality photo-realistic images of complex scenes. Rendering using PBRT is, however, not interactive. Exposure Render [1] is a volume ray caster, enhanced with physically based light

transport. It uses Monte Carlo Ray Tracing for direct volume rendering to simulate physically-based lighting effects. The ray tracing in Exposure Render is performed on the GPU making it an efficient parallelized renderer.

CHAPTER

3 Frequency Modulated Shading

An algorithm must be seen to be believed.

Donald Knuth

Our work aims to improve the visualization of 3D ultrasound data by combining the surface perception provided by local illumination shading, and the depth cues provided by shadows using global illumination techniques. As stated earlier, local illumination models using gradients are very sensitive to noise, and since ultrasound data suffers from noise, using gradient-based shading is not well suited for ultrasound data. To improve the surface perception we employ a frequency analysis to remove noise and create smoothed gradients that are better suited to be used for shading.

Our approach is inspired by the method, which Vucini et al. presented in their paper about frequency-based transfer functions [30]. Our method performs a frequency analysis of a local area around each voxel. After the analysis is performed, the amplitude of the frequencies are modified, depending on the result of the analysis. The data are then transferred back into spatial domain, and gradients are computed using central differences. The gradients are stored in a new volume which is loaded as a normal texture in the renderer. Rendering is then performed using the approach of chromatic shadows, with an added step of gradient based shading. In the next sections we will describe each step in detail.

3.1 Frequency analysis

In our approach we use the same general approach as Vucini et al. used with their frequency-based transfer function. We are however not looking at modulating the frequencies of the entire volume. To compute gradients which suffer less from noise, a local area analysis is performed and a new local volume is created using only high magnitude low frequencies. The low frequencies are less likely to be an effect of noise. The majority of these can be used when creating the new volume. The higher frequencies are more likely to be results of noise and many of these can be excluded from the new volume. Some higher frequencies are however a part of the true data, and should not be removed. Our technique uses a thresholding technique, where frequencies with an amplitude above a certain threshold contribute to the new filtered new volume. Frequencies with amplitudes below this threshold are discarded. The relative distance to the DC-component is used to scale the threshold, so that the higher the frequencies, meaning frequencies farther from the DC-component, have a higher threshold than lower frequencies. Figure 3.1 illustrates the overall process of the frequency analysis. In the following paragraphs each step is described in more detail.

Step One: Creating Local Volume

The first step in our approach is to create a local volume that should be analyzed. The analysis runs for each voxel in the volume. The local volume is created with the current voxel in the center. A volume of size $8 \times 8 \times 8$ is created. The current voxel, for which the analysis is performed, is placed at $(4, 4, 4)$ which is considered the center of the volume. The surrounding voxels fills the rest of the volume. If the current voxel is at or near an edge, the surrounding area is zero-padded as needed, filling any voxels that would be outside the volume with zeros. To avoid ghosting artifacts from the frequency modulation that is performed later, this volume is zero-padded with two voxels in every direction, resulting in a new volume of size $12 \times 12 \times 12$, with the center voxel at $(6, 6, 6)$. This newly created volume is represented by the top-left image in Figure 3.1.

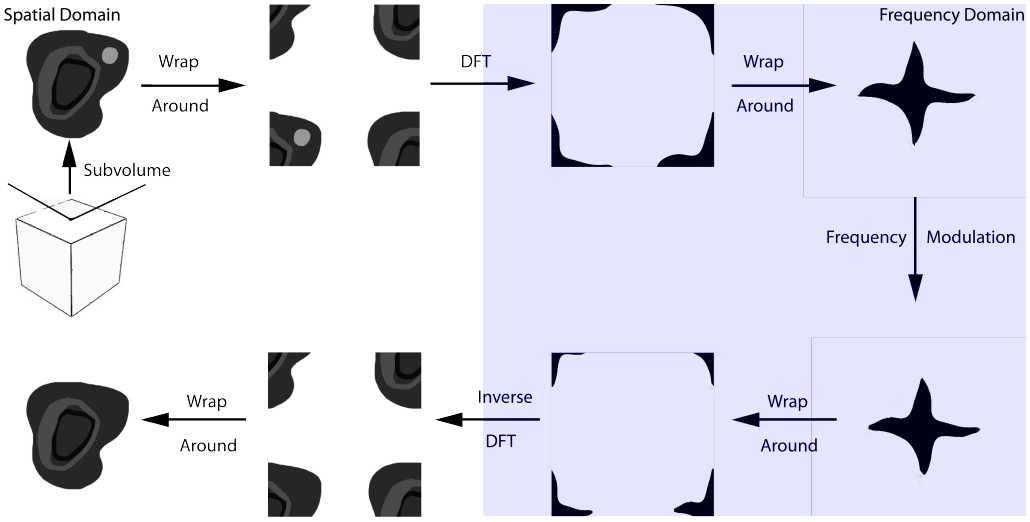


Figure 3.1: Illustration of the algorithm. A local area volume is created by a subvolume of the dataset volume. A re-shuffling (wrap around) of the subvolume is performed to position the center voxel in $(0,0,0)$. A Discrete Fourier Transform (DFT) is performed to transform the subvolume into the frequency domain. Another wrap around is performed. Then the frequency modulation is performed and a new subvolume is created. Wrap arounds and inverse DFT is performed to transform the subvolume back into spatial domain.

Step Two: Fourier Transform

The next step is to transform this local volume into frequency domain using the Discrete Fourier transform (DFT). DFT, however, considers the voxel at $(0,0,0)$ to be the center. The analysis should be performed with the current voxel, positioned at $(6,6,6)$, at the center. Therefore, before performing the DFT the local volume is shifted to place the current voxel at $(0,0,0)$. This is achieved by re-shuffling parts of the volume in a so-called wrap around operation [27]. To do the wrap around the local volume is divided into 8 subvolumes. The position of these subvolumes are then swapped, first horizontally, then vertically, and finally in depth. Figure 3.2 illustrates this procedure in 2D. Volume 1 is first swapped with volume 2, and volume 3 is swapped with volume 4. Volume 1 is then swapped with volume 4, and volume 2 is swapped with volume 3.

Performing zero-padding and wrap around is necessary to avoid ghosting

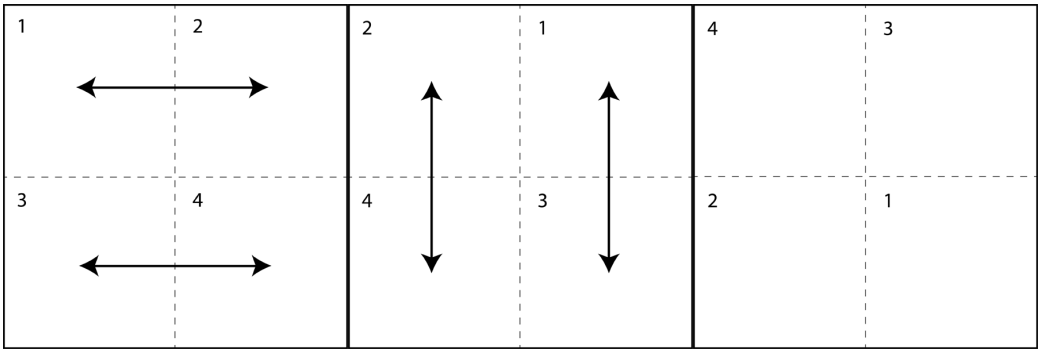


Figure 3.2: Illustration of wrap around in 2D. Subsections are shuffled as indicated by the arrows. In 3D there is the added step of shuffling in depth.

when the modulated frequencies are transformed back into spatial domain. Figure 3.3 illustrates these ghosting artifacts in the lobster data set. Frequency modulation in Figure 3.3 is done on the entire volume, using a frequency transfer function, as described by Vucini et al. Our method uses the same approach to modulating the frequencies as Vucini et al., in a local area, and the same precautions have to be taken.

Figure 3.3: Lobster with no frequency modulation (left), with modulation without padding and wrap around (middle), and with modulation with padding and wrap around (right).

After the local volume has been created and the voxels rearranged to place the current voxel at $(0,0,0)$ a 3D DFT is performed. The DFT produces the frequency representation, $F(x,y,z)$, of the local volume, according to Equation 3.1, where

$V(k_1,k_2,k_3)$ is the local area volume, and N is the size of the dimensions in the local area volume.

$$F(x,y,z) = \frac{1}{N^3} \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} \sum_{k_3=0}^{N-1} V(k_1,k_2,k_3) \times e^{-2\pi j(\frac{x}{N}+\frac{y}{N}+\frac{z}{N})} \tag{3.1}$$

Step Three: Frequency Modulation

The aim of the frequency analysis is to remove frequencies that produce noise in the data which do not give true structural information. Frequencies with a high amplitude are the frequencies that have the larges impact on the structures. Low frequencies with a high amplitude are the frequencies which represent the core structure of the data with small changes. Higher frequencies represent more rapid changes in the data. Noise is present in the higher frequency ranges. Higher frequencies do, however, not only contain noise, but also structural information in places where the data changes suddenly, such as on the interface between two types of tissue. In order to remove noise, while still aiming to keep the structural information in the high frequency ranges a scaled threshold is used. Frequencies with an amplitude larger than the threshold are used to create a new volume. The new volume will contain only high amplitude frequencies, but the threshold is lower for low frequencies. This allows low frequencies to be passed along to the new volume more easily than higher frequencies.

The threshold value should increase for higher frequencies, allowing low frequencies to easier be passed along. To modify the threshold value the distance to the DC-component is used. The DC-component is located at $(0,0,0)$ in the frequency volume. To use the distance to the DC-component as a scaling factor for the threshold, it should be positioned at the center of the volume, with negative frequencies located before the DC-component. In the current state of the frequency volume, the DC-component is located at $(0,0,0)$ followed by positive frequencies and negative frequencies following after the positive frequencies. To position the DC-component at the center of the frequency volume, another wrap around has to be performed. The current state of the frequency volume is represented by the top-right image in Figure 3.1. Now that the DC-component is positioned at the center, the distance to it can easily be used as a scaling factor

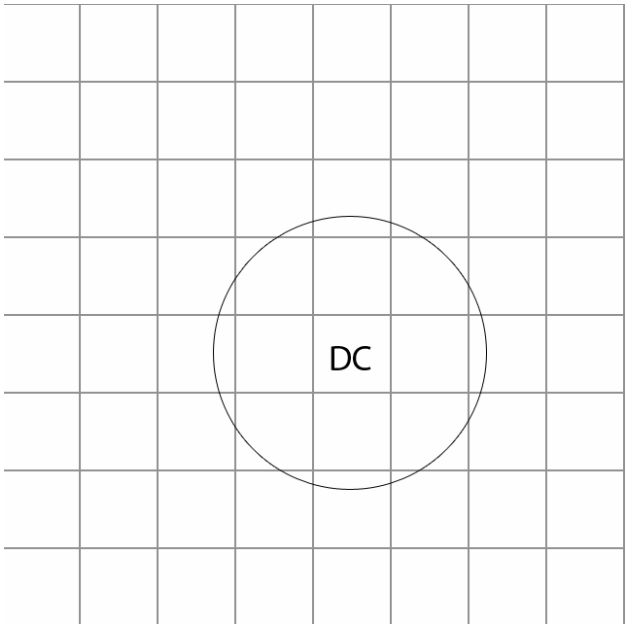


Figure 3.4: DC-component is positioned at the center. The threshold is scaled by the distance to the DC-component. The scale increases at the same speed in all directions, meaning that the threshold is scaled by the same value at all points on a circle centered at the DC-component.

for the threshold. Let the position of the DC-component be (x_{DC}, y_{DC}, z_{DC}). The scaling factor for the threshold is then given by the distance to the DC-component as in Equation 3.2.

$$distance = \sqrt{(x - x_{DC})^2 + (y - y_{DC})^2 + (z - z_{DC})^2} \tag{3.2}$$

The new volume can now be created by traversing the frequency volume and passing along the frequencies with $amplitude > threshold \times distance$.

Step Four: Inverse Fourier Transform

When the new volume of frequencies has been created, the next step is to transform it back into the spatial domain. This is done by doing an inverse 3D DFT. Before the data can be transformed back into spatial domain, it is necessary to perform another wrap around. The wrap around done before creating a new volume

has placed the DC-component in the center of the volume. The DFT considers the DC-component to be positioned at $(0,0,0)$. Due to this it is necessary to perform another wrap around to undo the previous rearrangement. The inverse DFT, according to Equation 3.3, is then used to transforms the data back into the spatial domain. In the current state of the volume what should be the center voxel is positioned at (0, 0, 0). This is due to the first wrap around that was performed before transforming the data into the frequency domain. To undo this rearrangement, one final wrap around has to be performed.

$$V(k_1,k_2,k_3) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1}\sum_{z=0}^{N-1} V(x,y,z) \times e^{2\pi j(\frac{k_1}{N}+\frac{k_2}{N}+\frac{k_3}{N})} \tag{3.3}$$

Step Five: Computing Gradients

Finally, gradients can be computed from the new smoothed volume. Gradients are computed using central differences according to Equation 3.4, where $d(x,y,z)$ is the data value at position (x, y, z) in the local volume.

$$g(x) = \frac{1}{2}\begin{pmatrix} d(x_{i+1},y_j,z_k) - d(x_{i-1},y_j,z_k) \\ d(x_i,y_{j+1},z_k) - d(x_i,y_{j-1},z_k) \\ d(x_i,y_j,z_{k+1}) - d(x_i,y_j,z_{k-1}) \end{pmatrix} \tag{3.4}$$

The analysis is performed once for each voxel in the original volume, and the gradients are stored as a vector in a gradient volume with the same dimensions as the original volume. The original volume and the volume with corresponding gradients will both be loaded into the renderer, which is described in more detail in the next section.

3.2 Rendering

Our renderer is based on the chromatic shadows renderer method described by Šoltészová et al. [29]. The paper by Šoltészová et al. describes a way to add colors to shadows to make a part of the scene that is in shadow more visible. They applied this method to a shadowing model, called multi-directional occlusion shading (MOS), from their previous work [28]. Our goal is to use this method

in combination with the newly computed gradients to get better visualization of the data.

Our renderer uses an implementation of MOS, with shadows being mapped to a color as described in the paper about chromatic shadows. In addition to computing chromatic shadows our renderer uses the gradients computed during the frequency analysis to provide gradient based shading. Our renderer first computes the color according to the transfer function and the gradient based shading algorithm, using our newly computed normals. This color is then combined with the shadow color, obtained from a transfer function for shadow. The transfer function for shadow depends on the "shadowiness"-value, which is mapped to a color and an opacity. The combined gradient shaded color, and shadow color is combined as described in Equation 3.5.

$$\begin{aligned} combinedColor.rgb = {} & shadedColor.rgb * shadowOpacity \\ & + shadowColor.rgb * (1 - shadowOpacity) \end{aligned} \tag{3.5}$$

CHAPTER

4 Implementation

A fact is a simple statement that everyone believes. It is innocent, unless found guilty. A hypothesis is a novel suggestion that no one wants to believe. It is guilty, until found effective.

Edward Teller

Our application is implemented as a proof-of-concept of the methods described in Chapter 3. The implementation has been tested on a computer running Windows 7, with 8GB of RAM and an Intel Core i5 processor operating at 2.6 GHz. The workstation was equipped with a NVIDIA GeForce GTX 570 Graphics card with 1280 MB graphic memory. All performance measurements were conducted with this hardware and operating system. In this chapter we describe some selected implementation details.

Rendering have been implemented in C++, and uses OpenGL with GLSL vertex and fragment shaders. The renderer have been implemented as a plug-in for an existing volume visualization framework called Volumeshop [4]. A screen shot from the Volumeshop framework as be seen in Figure 4.1.

In the next section we describe the implementation of the frequency analysis in more detail.

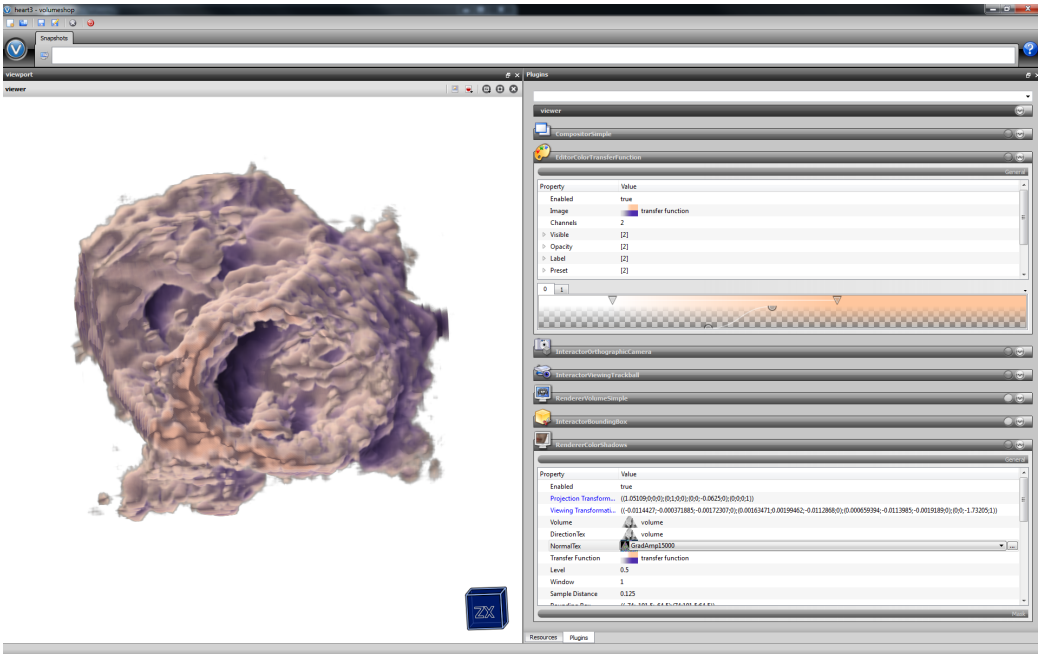


Figure 4.1: Screen shot from the Volumeshop framework.

4.1 Frequency Analysis

The frequency analysis has been implemented using Python and CUDA. We used Python to load the data, while the frequency analysis itself is performed on the GPU using CUDA [6], via the PyCuda wrapper [13]. CUDA allows general-purpose computation to be performed on the GPU (GPGPU), using C as a programming language. Since our method is implemented using CUDA our program requires a computer with a NVIDIA graphics card supporting CUDA.

Step One: Creating Local Volume

The frequency analysis consists of five steps, as described in Section 3.1. The first step is to build local area volume around the voxel currently being analyzed. Algorithm 1 describes the implementation logic in pseudocode. The algorithm should build a local volume around the current voxel, as well as zero-padding the local volume.

Algorithm 1 Algorithm for building local volume

```
Input:
Data Volume, Volume
Position of current voxel, Pos
Size of local volume dimensions, Size
_____________________________-
x = Pos.x - Size/2;
y = Pos.y - Size/2;
z = Pos.z - Size/2;
//Create local volume with dimension 1.5 times the number of voxels in each
//dimension. This is to allow zero-padding.
LocalVolume = new Volume[Size*1.5, Size*1.5, Size*1.5]
for i = 0 → Size do
    for j = 0 → Size do
        for k = 0 → Size do
            q = Size/4;
            LocalVolume[q + i, q + j, q + k] = Volume[x + i, y + j, z + k];
        end for
    end for
end for
```

In our implementation Size was set to 8, resulting in a 8x8x8 sub-volume being taken from the original volume, and with zero-padding equal to a quarter of the size, i.e. 2, on each side. The final local volume is a 12x12x12 volume.

Step Two/Four: Fourier Transforms

To transform the data into the frequency domain where the analysis is performed, we perform a 3D DFT. A 3D DFT is done by performing three consecutive DFTs, one in each direction. Algorithm 2 shows how the 1D DFT have been implemented. Algorithm 3 shows how the 3D DFT have been implemented. Algorithm 2 is used several times within Algorithm 3. These algorithms are used for both DFT, and Inverse DFT. The algorithms take a direction value as input, and this value defines which type of DFT should be performed.

Algorithm 2 Algorithm for DFT

Input:
SubArray of Local Volume (complex), LocArr
Length of SubArray, m
Direction, dir //1 if DFT, -1 if Inverse DFT

---

for $i = 0 \rightarrow m$ **do**
  $arg = -dir * 2 * \pi / m$;
  for $j = 0 \rightarrow m$ **do**
    $cosarg = cos(arg * j)$;
    $sinarg = sin(arg * j)$;
    $TempArr[i].real += LocArr[j].real * cosarg - LocArr[i].img * sinarg$;
    $TempArr[i].img += LocArr[j].real * sinarg - LocArr[i].img * cosarg$;
  end for
end for
$LocArr = TempArr$;

Step Three: Frequency Modulation

The next step in the process is to remove the low amplitude high frequencies. Algorithm 4 describes how this have been implemented. The algorithm takes a local area frequency volume as input. This frequency volume should have the DC-component in the center. The local area volume created in Algorithm 1 is in spatial domain, and before it can be forwarded to Algorithm 4 it needs to be transformed into frequency domain. As was described in Section 3.1 a wrap around also have

Algorithm 3 Algorithm for 3D DFT

```
Input:
Local Volume (complex), LocVol
Size of dimensions, n
Direction, dir //1 if DFT, -1 if Inverse DFT
---------------------------------------
//Transform depth
for i = 0 → n do
    for j = 0 → n do
        for k = 0 → n do
            LocArr[k] = LocVol[i,j,k]
        end for
        DFT(LocArr, n, dir);
        for k = 0 → n do
            LocVol[i, j, k] = LocArr[k]
        end for
    end for
end for
//Transform cols
for k = 0 → n do
    for i = 0 → n do
        for j = 0 → n do
            LocArr[j] = LocVol[i,j,k]
        end for
        DFT(LocArr, n, dir);
        for j = 0 → n do
            LocVol[i, j, k] = LocArr[j]
        end for
    end for
end for
//Transform rows
for j = 0 → n do
    for k = 0 → n do
        for i = 0 → n do
            LocArr[i] = LocVol[i,j,k]
        end for
        DFT(LocArr, n, dir);
        for i = 0 → n do
            LocVol[i, j, k] = LocArr[i]
        end for
    end for
end for
```

to be performed to position the DC-component in the center. Once this has been performed the local volume is forwarded to Algorithm 4.

Algorithm 4 Algorithm for frequency modulation

```
Input:
Local Volume, LocVol
Threshold, T
-----------------------------------
for i = 0 → Size do
    for j = 0 → Size do
        for k = 0 → Size do
            Mid = Size/2;
            Dist = sqrt((i-Mid)*(i-Mid) + (j-Mid)*(j-Mid) + (k-Mid)*(k-Mid));
            if LocVol[i, j, k] > Dist*T then
                ModifiedLocalVolume[i, j, k] = LocVol[i, j, k];
            else
                ModifiedLocalVolume[i, j, k] = 0;
            end if
        end for
    end for
end for
```

Algorithm 4 produces a new local frequency volume in which some frequencies have been removed. $LocVol[i, j, k] > Dist * T$ ensures that low amplitude frequencies are removed, as well as ensuring that the DC-component always is included in the new volume.

Step Five: Computing Gradients

The final step is to compute gradients using the modified local volume from Algorithm 4. The gradients are put in a new volume with the same size as the original volume, at the position corresponding to the voxel currently being analyzed. Algorithm 5 describes how the gradients are computed and put in the new gradient volume. The output from Algorithm 4 is in frequency domain, and needs to be transformed back to spatial domain before it can be forwarded to Algorithm 5. When this has been performed the modified local volume is sent to Algorithm 5.

Algorithm 5 Algorithm for gradient computation

Input:
Modified Local Volume, LocVol
Position of current voxel, Pos
Gradient Volume, GradVol
Size of local volume dimensions, Size
———————————————-

```
Mid = Size/2;
gradX = 0.5*(LocVol[Mid + 1, Mid, Mid] - LocVol[Mid - 1, Mid, Mid]);
gradY = 0.5*(LocVol[Mid, Mid + 1, Mid] - LocVol[Mid, Mid - 1, Mid]);
gradZ = 0.5*(LocVol[Mid, Mid, Mid + 1] - LocVol[Mid, Mid, Mid - 1]);
GradVol[Pos.x, Pos.y, Pos.z] = Vector(gradX, gradY, gradZ);
```

When Algorithm 5 has been performed for every voxel in the volume, the gradient volume is ready to be loaded into the renderer.

Performance

To provide an overview of how time-consuming the frequency analysis is, we have performed the analysis on several different datasets with varying data sizes. The results of the performance tests can be seen in Table 4.1 below.

Volume Size	Runtime in Seconds
$50 \times 50 \times 50$	3.02 sec
$50 \times 50 \times 100$	5.37 sec
$50 \times 100 \times 100$	10.18 sec
$100 \times 100 \times 100$	19.76 sec
$200 \times 200 \times 200$	153.80 sec
$256 \times 256 \times 256$	196.35 sec
$128 \times 200 \times 128$	63.26 sec
$148 \times 203 \times 129$	74.66 sec
$181 \times 245 \times 190$	161.66 sec

The first five volumes listed in Table 4.1 are phantom datasets created for performance testing. The last three volumes are actual datasets from heart and liver ultrasound scans. As can be seen in the table small volumes are analyzed in

a matter of seconds, while larger volumes, like $256 \times 256 \times 256$, require three to four minutes to perform the frequency analysis.

CHAPTER

5 Results

People love chopping wood. In this activity one immediately sees results.

Albert Einstein

This chapter will show results that have been achieved using our algorithm for frequency-modulated shading. Section 5.1 will illustrate the effects of changing the threshold value. Section 5.2 will compare our gradients to gradients computed using a low pass filter instead of our more complex and more expensive algorithm.

5.1 Results using varying threshold

The threshold value used in the frequency modulation algorithm can be varied. The next set of images, in Figures 5.1 - 5.4, aims to illustrate the effects of increasing the threshold while value using our method. The figures also compare the results to resulting images using gradients computed on the fly, as well as results without using gradient-based shading.

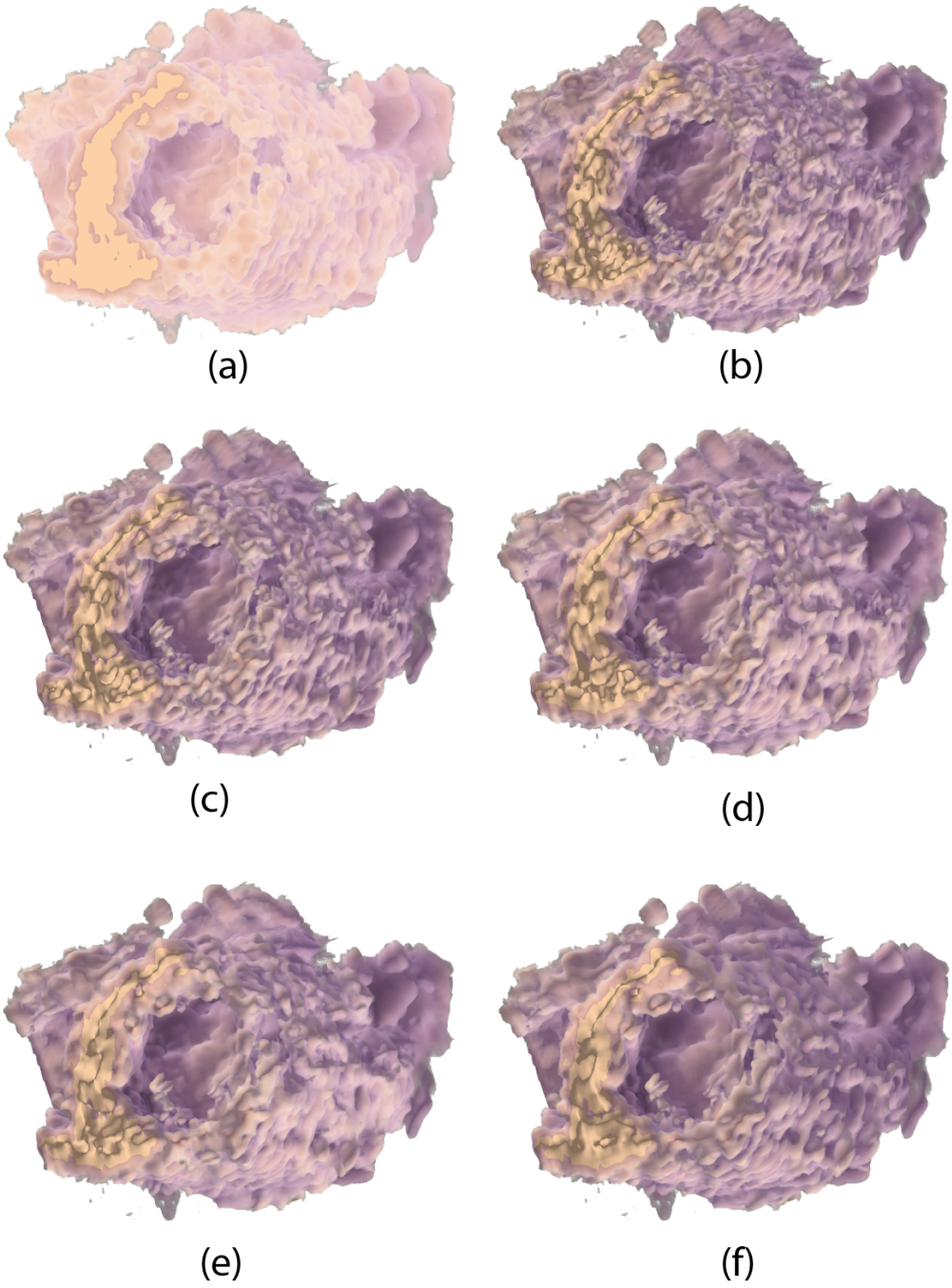


Figure 5.1: Heart dataset with (a) no gradient based shading, (b) gradients computed on the fly, and (c) - (f) using our precomputed gradients with an increasing threshold value. Increasing the threshold results in smoother gradients.

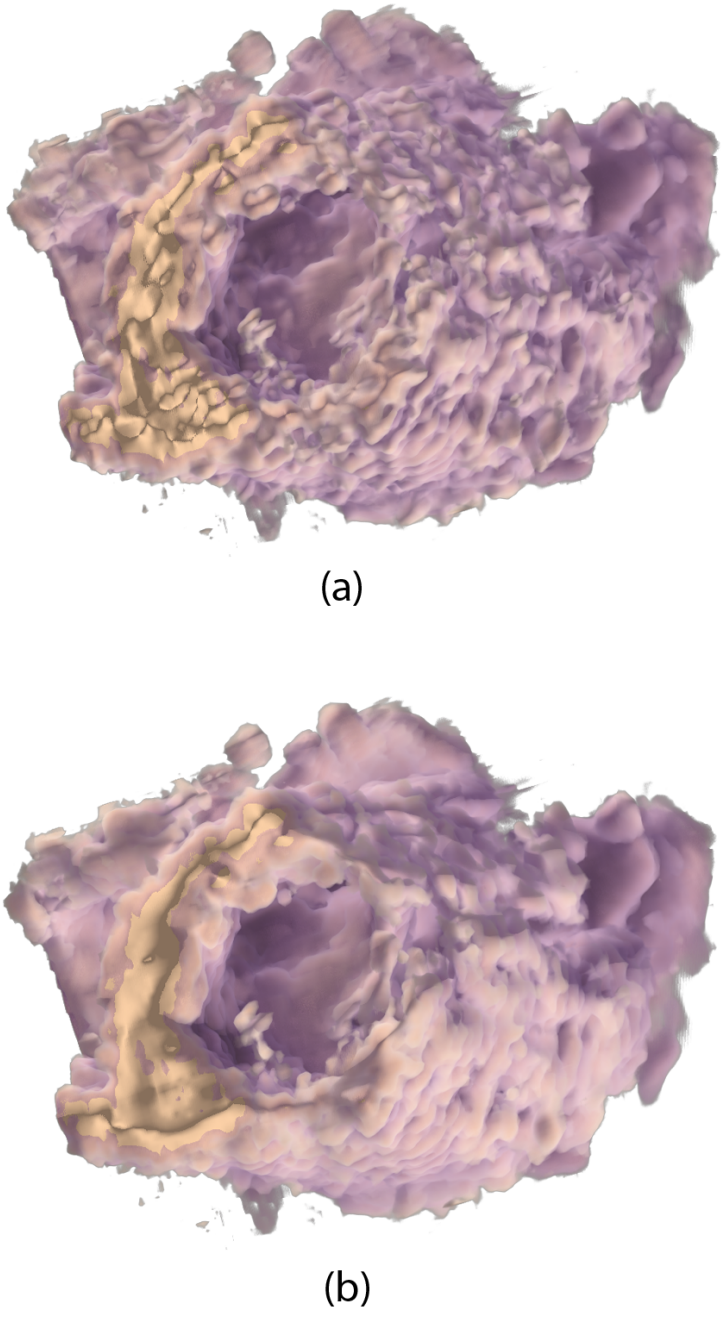


Figure 5.2: Heart dataset with (a) gradients computed using a very low threshold, and (b) gradients computed using a very high threshold. It can clearly be seen that increasing the threshold results in smoother gradients.

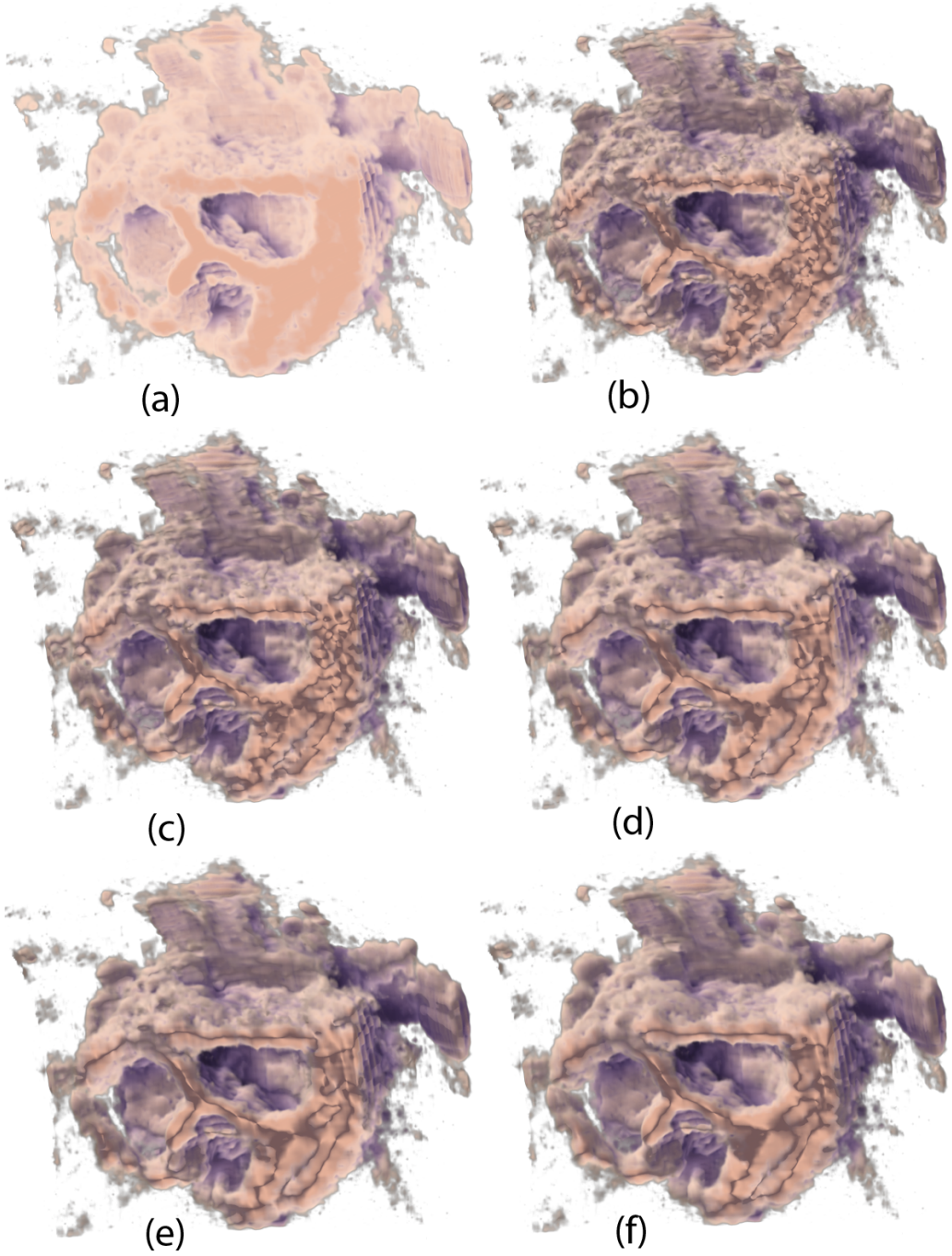


Figure 5.3: Heart dataset with (a) no gradient based shading, (b) gradients computed on the fly, and (c) - (f) using our precomputed gradients with an increasing threshold value. Increasing the threshold results in smoother gradients.

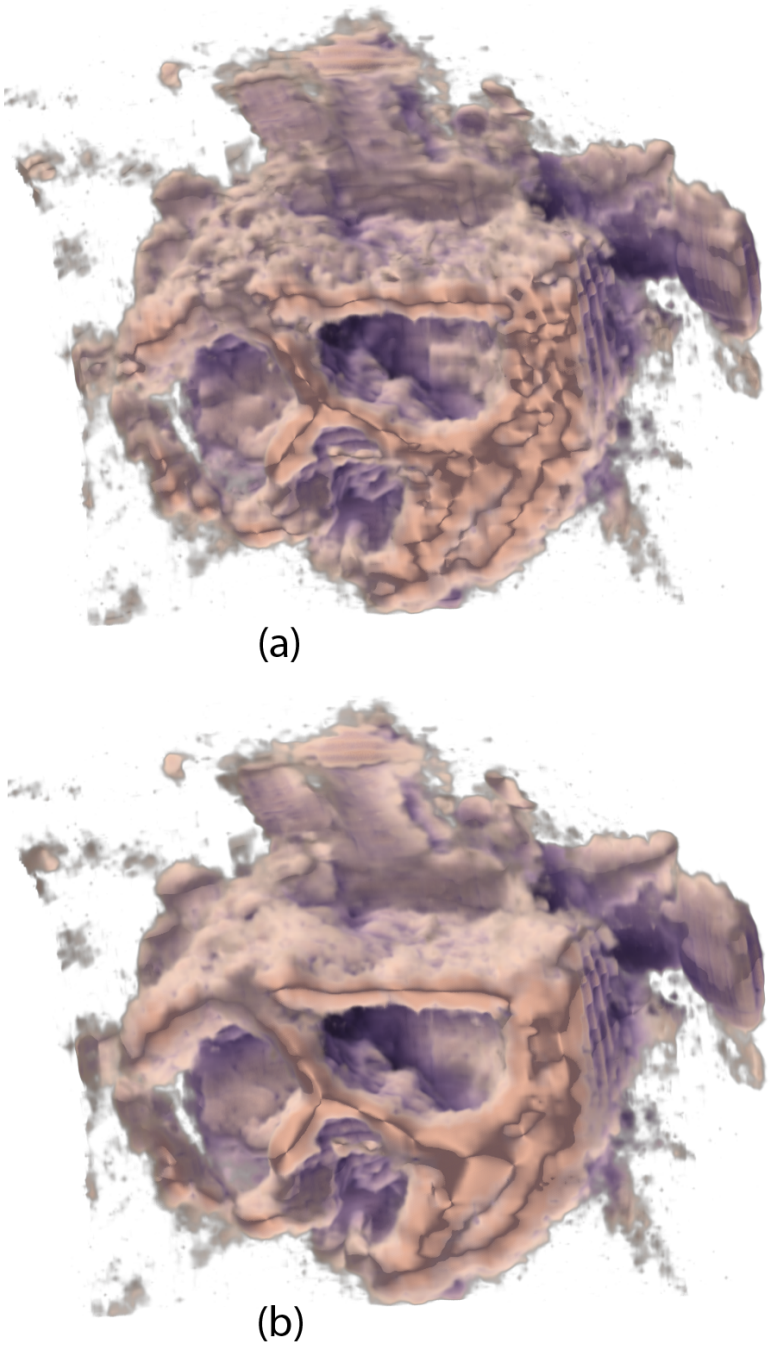


Figure 5.4: Heart dataset with (a) gradients computed using a low threshold, and (b) gradients computed using a very high threshold. It can clearly be seen that increasing the threshold results in smoother gradients.

Increasing the threshold value provides smoother gradients, as can be seen in the figures on the previous pages. Using a low threshold does not give results much different from using gradients computed on the fly, but as the threshold is increased, the results are smoother and less cluttered from noise. Increasing the threshold too much, will however result in over smoothing, and valuable information can be lost. The higher the threshold is made, the more frequencies are excluded when computing the gradients. If the threshold value is increased too much, there will be too few frequencies left to provide any valuable data. Figure 5.5 shows a heart dataset with gradients computed using a very high threshold. Figures 5.6 - 5.8 shows more results using different datasets. Figure 5.6 show results from a liver dataset, while the rest shows results from different heart datasets.

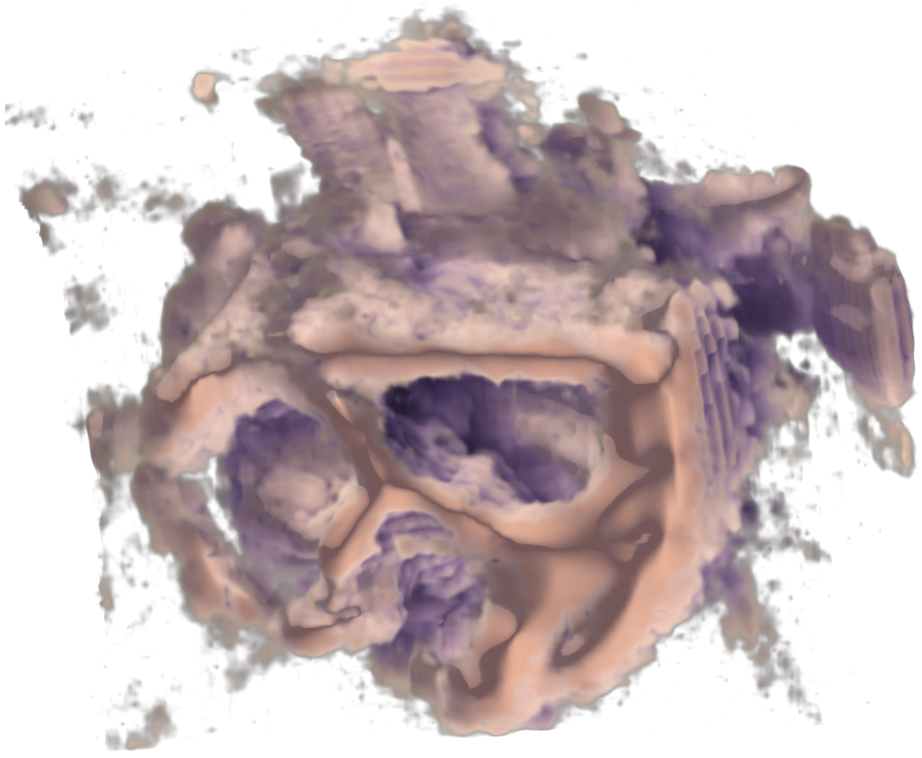


Figure 5.5: Heart dataset with gradients computed using a very high threshold.

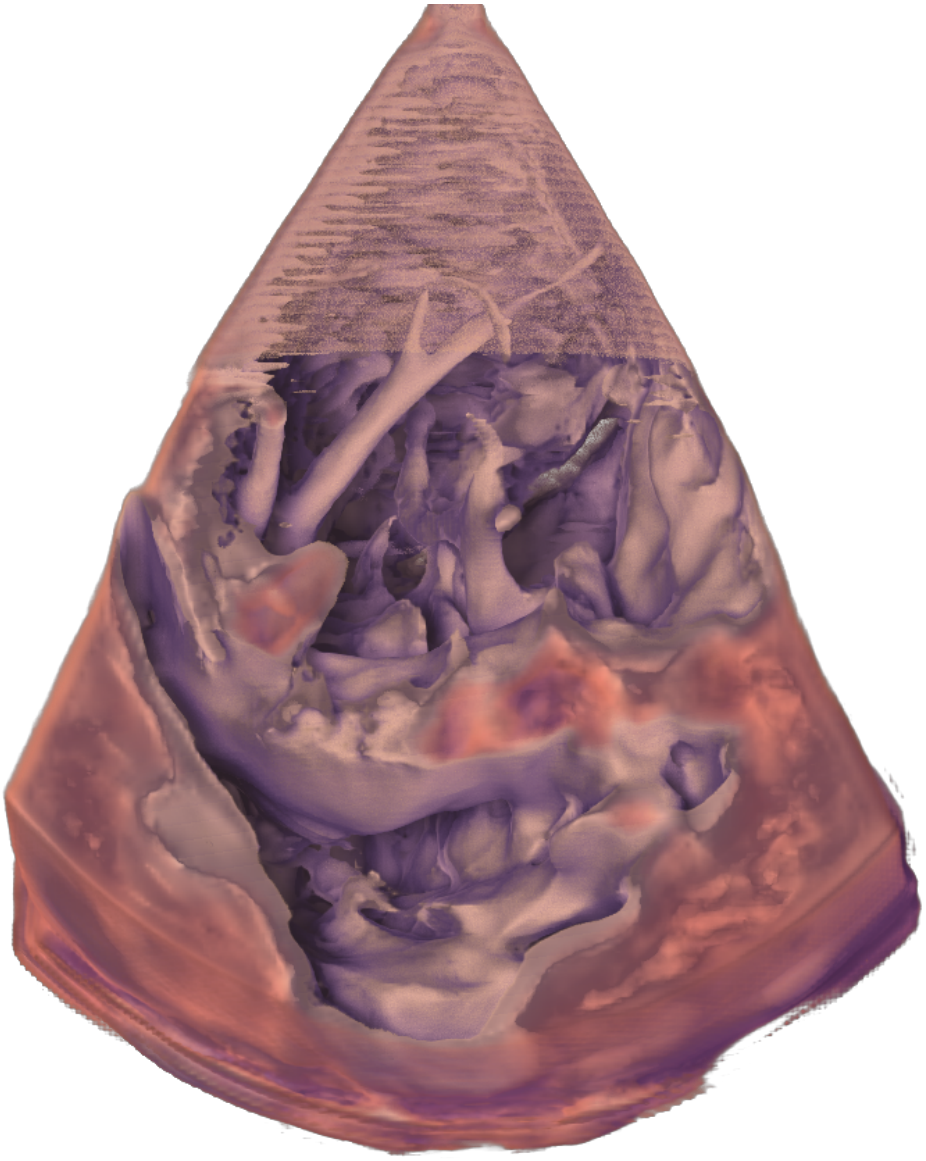


Figure 5.6: Liver dataset with gradients computed using our method.

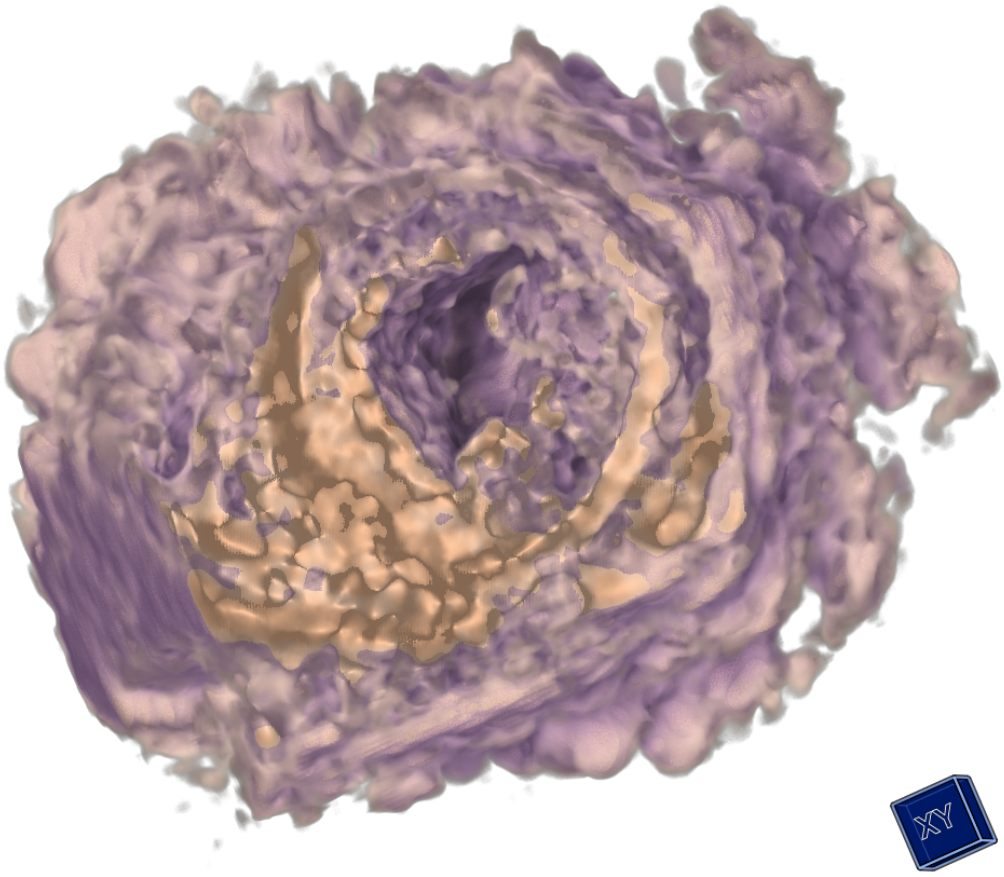


Figure 5.7: Heart dataset with gradients computed using our method.

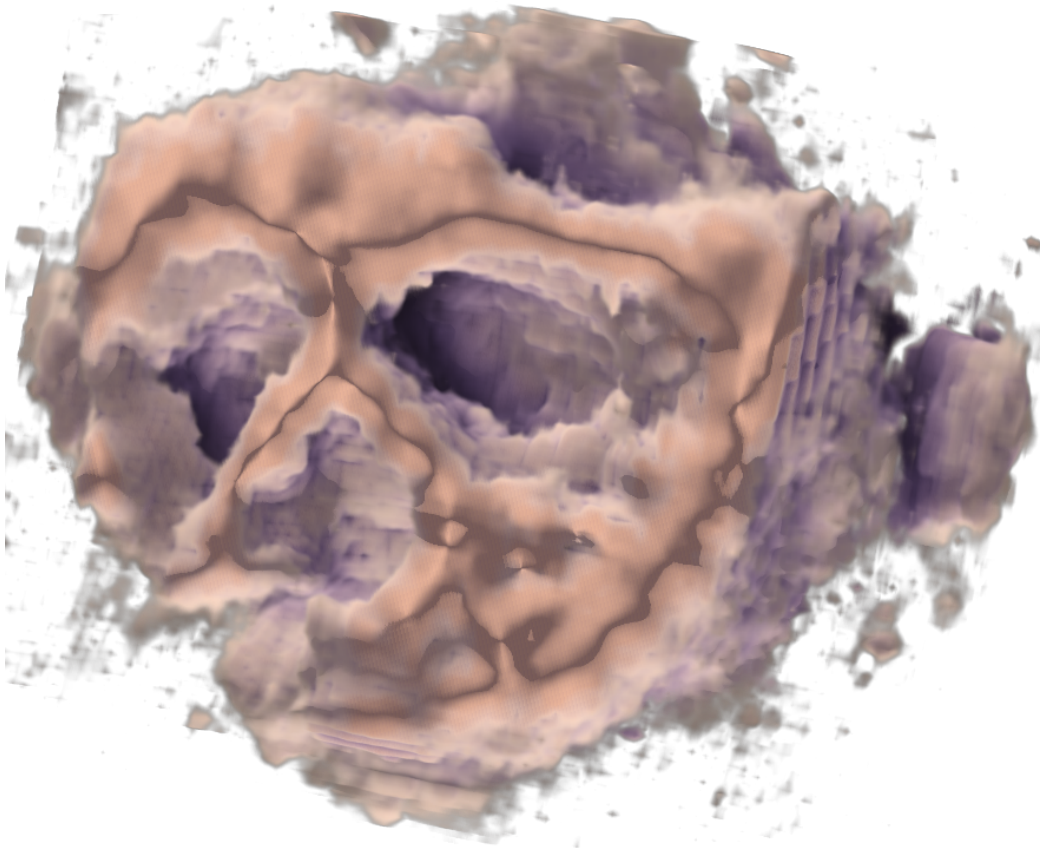


Figure 5.8: Heart dataset with gradients computed using our method.

5.2 Results compared to low pass filtering

A simpler approach to the frequency-modulation is to only consider low frequencies. In our implementation, this can easily be achieved by only including frequencies which are a short distance from the DC-component, instead of considering their amplitudes. Figure 5.9 shows our method compared to using a low-pass filter. Figure 5.10 shows the same comparison, but using a wider filter, meaning the filter takes more frequencies into account. Figure 5.11 shows another comparison, using a different dataset.

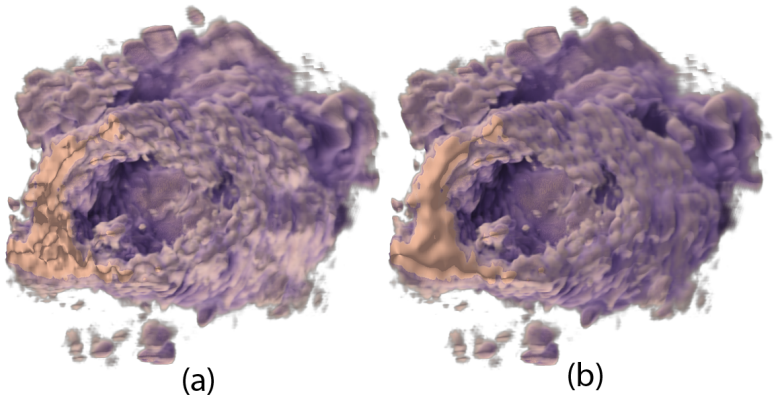


Figure 5.9: Heart dataset with gradients using our method (a), and using low-pass filterer gradients(b).

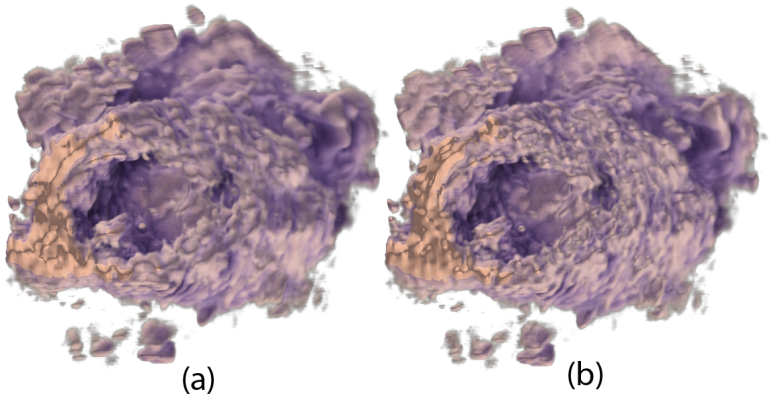


Figure 5.10: Heart dataset with gradients using our method (a), and using low-pass filterer gradients(b) (using a wider filter than in the previous figure).

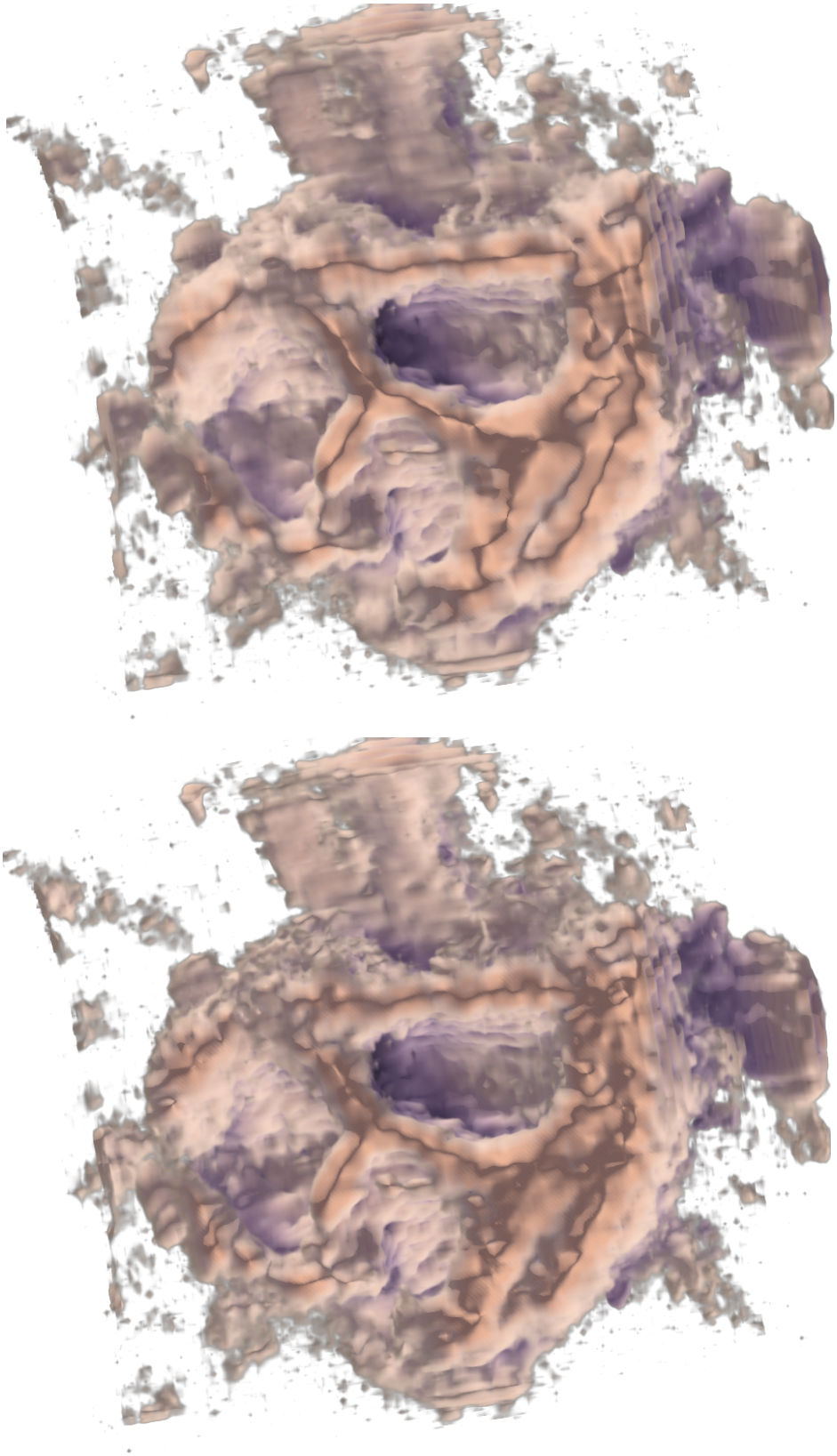


Figure 5.11: Heart dataset with gradients using our method (top), and using low-pass filterer gradients(bottom).

CHAPTER

6 Summary and Conclusions

It's more fun to arrive a conclusion than to justify it.

Malcolm Forbes

The use of medical imaging to get an image of patient's anatomy and processes in the patients body is an important part of medicine today. Different scanning modalities can be used to create images or volumes that contain anatomic and metabolic information about patients. Ultrasonography is an important medical imaging tool, that uses sound waves to generate images of scanned tissue. It has a wide range of applicability and has a high resolution. Due to its portability it can be used in many different situations, not just in a diagnosis or operating room in a hospital. The natural properties of ultrasound, do however create some artifacts in the data and the signal-to-noise ratio is low. Due to this, an important part of working with US-data is minimizing the influence of noise. In order to minimize noise in the data several different approaches have been used. As described in Chapter 2 there are several pre-processing that have been applied to US-data as well as different rendering techniques. A basic approach to direct volume rendering is to use gradients to compute normals and use these with surface shading techniques such as Phong shading. Gradients are computed using finite differences, and due to this, gradients are very influenced by noise in the data. Since US-data has a low signal to noise ratio direct use of gradients for shading does not provide good re-

sults. More advanced global illumination and semi global illumination techniques have been used to circumvent this problem. Global illumination techniques, or approximations to these, provide better results that are less influenced by noisy data, but that are more computationally expensive. Global illumination and semi global illumination techniques take into account a wider area when computing shading, while gradient based shading considers the immediate neighborhood. In Section 2.2 several different approaches to rendering of 3D data are presented. Our technique uses both global illumination techniques as well as gradient based shading with gradients based on a frequency modulated volume.

We have presented a technique for frequency-modulated shading. The technique uses a combination of frequency-modulated gradients and chromatic shadows to render 3D ultrasound data. In order to reduce the impact of noise when computing gradients the technique transforms the data into frequency spectrum, where after an analysis the amplitude of the frequencies is modulated. The technique allows computation of gradients that are affected less by noise in the original data. The frequency modulation is inspired by a paper by Vucini et al. on frequency based transfer functions. This paper is described in Section 2.2. To modulate the frequencies, a threshold scaled by distance to the DC-component is used. Frequencies with amplitudes higher than the threshold are used to create a new volume from which the gradients are computed. The procedure of frequency modulation and gradient computation is described in section 3.1. Our method uses these frequency modulated gradients in combination with a more advanced rendering technique using shadows, to render images. A more advanced rendering technique is the use of chromatic shadows as described by Šoltészová et al. This technique is described in Section 2.2. The shading for each voxel in the volume is first computed using gradient based shading with the newly computed normals. This color is then blended with the shadow color. This way structural information from gradient-based shading is brought together with depth cues provided by shadows.

6.1 Conclusions

The goal of this work was to improve the visualization of 3D ultrasound data. To achieve this we have combined a rendering approach using chromatic shadows with gradient based shading. Due to the low signal-to-noise ratio of ultrasound data, and gradients being computed using finite differences, which suffers from noise, a new way of computing gradients had to be invented. Our method for computing frequency modulated gradients, provides a new way of reducing the effects of noise in 3D ultrasound data. By performing the frequency analysis and removing some of the frequencies we get a smoothed dataset that produces gradients, which are less affected by noise. By combining gradient based shading using these gradients and a shadow renderer, we are able to generate images with good structural information as well as good depth cues. There is still much room for improvement, but initial results indicate, that performing such a frequency-modulation can help improve the visualization of ultrasound data.

6.2 Future Work

In this thesis we have presented a technique for frequency modulated shading, where we used a scaled threshold to perform an analysis of the frequency spectrum, and then modified the amplitude of the frequencies depending on the threshold. For future work alternative methods for performing the frequency analysis can be considered. For instance, instead of using a scaled threshold a region growing approach can be considered.

As a part of this thesis we compared our frequency analysis to low-pass filtering. For future work comparisons to other filtering techniques, for instance using wavelets, should be performed.

Acknowledgements

There will come a time when you believe everything is finished. Yet that will be the beginning.

Louis L'Amour

I would like to thank my supervisors Veronika Šoltészová and Ivan Viola for their invaluable guidance and feedback while working on this thesis. I also want to thank the rest of the Visualization group at the Department of Informatics, and especially Prof. Helwig Hauser for his assistance with planning the Master Thesis.

This work was done in connection to the IllustraSound research project (# 193180), which is funded by the VERDIKT program of the Norwegian Research Council with support of the MedViz network in Bergen, Norway. The focus of the IllustraSound project is on visualization of medical ultrasound data. It aims to develop new visualization technology to improve the readability of ultrasound data. Reading ultrasound images requires training and improving the visualization of ultrasound data can improve communication between doctors and patients, as well as giving doctors better tools when developing a diagnosis for a patient [2].

I would like to thank Erald Vucini for his help and additional explanation while I was working with his paper on Frequency Transfer functions. His help was much appreciated.

I would also like to thank Prof. Kwan-Liu Ma for his suggestions while planning the thesis. My thanks also goes out to Stefan Bruckner for providing and maintaining Volumeshop.

Last but not least, I would like to thank: My friends and fellow students, who make studying more enjoyable. Everyone who has influenced me in my choice of academic path, my previous teachers and probably most of all my brother.

Bibliography

[1] Exposure render. http://code.google.com/p/exposure-render/, 2012.

[2] Illustrasound. http://www.ii.uib.no/vis/projects/illustrasound, 2012.

[3] Åsmund Birkeland, Veronika Šoltészová, Dieter Hönigmann, Odd Helge Gilja, Svein Brekke, Timo Ropinski, and Ivan Viola. The ultrasound visualization pipeline – a survey. *ArXive – e-prints*, 2012.

[4] S. Bruckner and M. E. Gröller. VolumeShop: An interactive system for direct volume illustration. *Proceedings of IEEE Visualization*, pages 671–678, 2005.

[5] Y. Chen, R. Yin, P. Flynn, and S. Broschat. Aggressive region growing for speckle reduction in ultrasound images. *Pattern Recognition Letters*, 24(4-5):677–691, 2003.

[6] NVIDIA Corporation. Cuda. http://www.nvidia.com/object/cuda_home_new.html, 2012.

[7] W. F. Garret, H. Fuchs, M. C. Whitton, and A. State. Real-time incremental visualization of dynamic ultrasound volumes using parallel BSP trees. *Proceedings of IEEE Visualization*, pages 235–240, 1996.

[8] A. Gee, R. Prager, G. Treece, and L. Berman. Narrow-band volume rendering for freehand 3D ultrasound. *Computers & Graphics*, 26(3):463–476, 2002.

[9] O. H. Gilja, T. Hausken, S. Ødegaard, Ø. Wendelbo, and M. Thierley. Mobile ultrasonography in a medical department. *Tidskrift for Norsk Lægeforening*, 19:270–285, 2003.

[10] GE Healthcare. GE's Vscan Aids Doctors Across Frontiers. `http://www.gereports.com/ges-vscan-aids-doctors-across-frontiers/`, March 2012.

[11] F. Hernell, P. Ljung, and A. Ynnerman. Local ambient occlusion in direct volume rendering. *Visualization and Computer Graphics, IEEE Transactions on*, 16(4):548 –559, 2010.

[12] Mustafa Karaman, M. Alper Kutay, and Gozde Bozdagi. An adaptive speckle suppression filter for medical ultrasonic imaging. *IEEE Transactions on Medical Imaging*, 14(2):283–292, 1995.

[13] Andreas Klöckner. Pycuda. `http://mathema.tician.de/software/pycuda/`, 2012.

[14] J. Kniss, S. Premoze, C. Hansen, and D. Ebert. Interactive translucent volume rendering and procedural modeling. *Proceedings of IEEE Visualization*, pages 109–116, 2002.

[15] J. I. Koo and S. B. Park. Speckle reduction with edge preservation in medical ultrasonic images using a homogeneous region growing mean filter (hrgmf). *Ultrasonic Imaging*, 13:211–237, 1991.

[16] M. Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(3):29–37, 1988.

[17] F. Lindemann and T. Ropinski. Advanced Light Material Interaction for Direct Volume Rendering. *Proceedings of IEEE/EG International Symposium on Volume Graphics*, pages 101–108, 2010.

[18] T. Loupas, W. N. McDicken, and P. L. Allan. Adaptive weighted median filter for speckle suppression in medical ultrasonic images. *IEEE Transactions on Circuits and Systems*, 36:129–135, 1989.

[19] Siemens Medical. Acuson x300 ultrasound system, premium edition. `http://www.medical.siemens.com/`, 2012.

[20] S. Ødegaard, L. B. Nesje, and O. H Gilja. *Atlas of Endoscopic Ultrasonography.* Fagbokforlaget, 2007.

[21] M. Pharr and G.. Humphreys. *Physically Based Rendering, Second Edition: From Theory To Implementation*. Morgan Kaufmann, 2010.

[22] M. Puerta. The power of shadows: shadow stereopsis. *Journal of the Optical Society of America. A, Optics and image science*, 6(2):302–311, 1989.

[23] T. Ropinski, C. Döring, and C. Rezk-Salama. Interactive volumetric lighting simulating scattering and shadowing. *Proceedings of IEEE Pacifc Visualization*.

[24] G. Sakas, L. Schreyer, and M. Grimm. Preprocessing and volume rendering of 3D ultrasonic data. *IEEE Computer Graphics and Applications*, 15(4):47–54, 1995.

[25] M. Schott, V. Pegoraro, C. Hansen, K. Boulanger, and K. Bouatouch. A Directional Occlusion Shading Model for Interactive Direct Volume Rendering. *Proceedings of EUROGRAPHICS*, pages 855–862, 2009.

[26] K. Torrance and E. Sparrow. Theory for off-specular reflection from roughened surfaces. *J. Optical Soc. America*, 57:1105–1114, 1976.

[27] I. Viola, A. Kanitsar, and M. E. Gröller. Gpu-based frequency domain volume rendering. *Proceedings of SCCG'04*, pages 49–58, 2004.

[28] V. Šoltészová, D. Patel, and I. Viola. A Multidirectional Occlusion Shading Model for Direct Volume Rendering. *Computer Graphics Forum*, 29(3):883–891, 2010.

[29] V. Šoltészová, D. Patel, and I. Viola. Chromatic shadows for improved perception. *Proceedings of ACM SIGGRAPH*, 1(212):105–116, 2011.

[30] E. Vucini, D Patel, and M. E. Gröller. Enhancing visualization with real-time frequency-based transfer functions. *Proceedings of SPIE conference on visualization and data analysis*, 2010.

[31] S. Zhukov, A. Iones, and G. Kronin. An ambient light illumination model. *Rendering Techniques*, pages 45–56, 1998.