

Rendering and Reconstruction of Astronomical Objects

Diplomarbeit

Kristian Hildebrand
Studiengang Mediensysteme
Bauhaus-Universität Weimar

Betreuer
Dr. Marcus Magnor
Max-Planck-Institut Informatik
Professor Dr. Bernd Fröhlich
Bauhaus-Universität Weimar

August 29, 2005

0.1 Acknowledgement

First, I am indebted to Dr. Marcus Magnor, head of the independent research group 'graphics-optics-vision' at MPI Saarbruecken, for having proposed the topic and for his interest and advice during the preparation of this work. Also, I would like to thank Prof. Bernd Froehlich, professorship for virtual reality at the Bauhaus University Weimar, for kindly having accepted the supervision of this thesis.

Futhermore, I would like to thank everybody who has contributed ideas, discussions and proof-reading to this work, especially Timo Stich for always having time for discussions, Andre Lințu for always providing me with good ideas and important astronomical knowledge and Christian Lessig for his helpful advice on GPU programming problems. I would like to say thanks to Ivo Ihrke, Lukas Ahrenberg, Timo Stich and Abhijeet Ghosh for proof-reading this thesis and to everybody at NWG3 and AG4 for keeping me sane with drinking some beers now and then and playing some games.

Last, but most importantly I am grateful my parents for giving me always the possibility to realize the things I wanted to, my sister and my loved Jule, who was always supporting me (Sorry, that you had to wait so long).

0.2 Abstract

The beauty of space and of astronomical objects is always fascinating to people and give rise to enormous research efforts to discover the secrets of the Universe. To get an impression on how these objects look like from a different viewpoint than our terrestrial confined position, one has to visualize them in their three-dimensional shape.

We present a new visualization technique for the rendering of astronomical objects, like *reflection nebulae* and provide two approaches for reconstructing the volumetric structure of *spiral galaxies* from conventional 2D images. Our interactive visualization tool renders the physically correct, commonly very colorful, appearance of arbitrary three-dimensional interstellar dust distributions surrounding illuminating stars. The proposed reconstruction algorithm incorporates computerized tomography methods as well as far-infrared information to plausibly recover to the shape of spiral galaxies. With our GPU-based volume rendering driving a non-linear optimization, we estimate the galaxy's dust density. The optimization refines the model by minimizing the differences between the rendered image and the original astronomical image.

These techniques can be used to create virtual fly-throughs of astronomical objects, such as *reflection nebulae* or *spiral galaxies* for either interactive desktop visualizations or scientifically accurate animations for educational purposes to gain a deeper and more intuitive understanding of the complex interaction of light and dust in real astrophysical settings.

Contents

0.1	Acknowledgement	iii
0.2	Abstract	v
1	Introduction	1
1.1	Introduction	1
1.2	Motivation	2
1.3	Contribution	3
1.4	Related Work	5
1.4.1	Computer Graphics and Astronomy	5
1.4.2	Astrophysics	6
1.4.3	Volume Rendering	7
1.4.4	Reconstruction	8
2	Realistic Rendering of Reflection Nebulae	9
2.1	Reflection Nebulae	9
2.1.1	Interstellar Dust	10
2.1.2	Colors and Astrophysical Settings	15
2.2	Visualization Model	17
2.2.1	Voxel Characteristics	20
2.2.2	Scattering Table Generation	21
2.3	Ray-Casting Implementation	23
2.3.1	GPU-Ray-casting	23
2.3.2	Fly-through Application	25
2.3.3	Reflection Nebula Rendering on GPU	26
2.3.4	Intensity Correction	27
2.4	Multi-resolution Multiple Scattering	28
2.5	Rendering several stars	32
2.6	Nebula Generation	34

2.7	Results and Discussion	35
3	Reconstruction of Galaxies from Telescope Images	39
3.1	Galaxy Background	39
3.2	Image Calibration and Normalization	41
3.3	Reconstruction from Images to 3D Volume	43
3.4	Image Morphing	45
3.4.1	Thin-Plate Splines	48
3.5	Computerized Tomography	50
3.5.1	Algebraic Reconstruction Tomography	51
3.5.2	Filtered Back-projection	54
3.6	Dust Distribution Reconstruction	57
3.7	Analysis by Synthesis Reconstruction	59
3.8	Results, Validation and Discussion	65
4	Summary and Outlook	69
A	Johnson System and Star Types	73

List of Figures

1.1	Computer graphics and astronomical objects	6
2.1	Reflection Nebulae	11
2.2	Light Scattering by Small Particles	12
2.3	Phase Function Symmetry	13
2.4	Albedo and Anisotropic Factor	15
2.5	Total-to-Selective Extinction	18
2.6	Visualization Model	19
2.7	Scattering Characteristics	21
2.8	Color Variations	22
2.9	Volume Rendering	24
2.10	Fly-through Images	25
2.11	HDR and Gamma Correction	28
2.12	Multiple Scattering Algorithm	30
2.13	Effects of Multi-resolution Multiple Scattering	31
2.14	Results of Multi-resolution Multiple Scattering	32
2.15	Performance	33
2.16	Visualization Model for Several Stars	34
2.17	Rendering Several Stars	35
2.18	Perlin Noise	36
2.19	Rendering Results	36
3.1	Galaxy Shape	40
3.2	Hubble Classification Scheme	41
3.3	Image Normalization	42
3.4	Galaxy Scattering	45
3.5	Generating new views	46

3.6	Image Morphing	47
3.7	Projection	51
3.8	Reconstruction Comparison of ART and Filtered Backprojection . .	53
3.9	Radon Transform	55
3.10	Computerized Tomography	55
3.11	Reconstruction by Backprojection	56
3.12	Dust Distribution	58
3.13	Reconstruction from Infrared	59
3.14	Analysis by Synthesis	60
3.15	Conjugate Gradient	62
3.16	Analysis by Synthesis M81 Galaxy	64
3.17	Analysis by Synthesis Galaxy M81	64
3.18	Fly-by Galaxy M81	65
3.19	Validation on synthetic data	67
4.1	Reflection Nebula Results	69
4.2	M81 Reconstruction Results	70

List of Tables

A.1	Star Colors	73
A.2	Johnson Color System	73

1 Introduction

1.1 Introduction

Astronomical objects in the night sky are always fascinating to people and give rise to enormous research efforts to discover the secrets of our universe. The very colorful and esthetically most attractive appearance of these objects inspires astronomers to find new objects, like other planets, to draw conclusions on the origin of the cosmos. Therefore thousands of telescopes around the world and in space record and collect data that is not only useful for physicists but also stunning and beautiful to everybody on earth.

By looking through an eye-piece of a telescope one can see various different objects in the night sky, e.g. planets, stars, all sorts of nebulae and galaxies. To get an impression on how these objects look like from a different viewpoint than our terrestrial position, one has to visualize astronomical objects in their three-dimensional shape and simulate the visual effects by exploring physical properties of light in interstellar material. There are several ways to mimic the objects in space.

One approach is to find a way of generating synthetic objects and visualize them in a physically correct manner to compare the results with the hypothetical or recorded objects in the sky. Another way is trying to find a plausible explanation for a correct three-dimensional shape out of original images to recover the actual spatial structure. Also artistic creation of objects from images by three-dimensional modeling and pleasant looking light modeling is a way to make the static astronomical objects more lively [18]. In this thesis we focus on the first two approaches.

Reflection Nebulae are among the most colorful objects in the night sky, which is the reason why they are often featured in science fiction movies and popular science journals. Besides their fascinating appearance they have significant scientific relevance because they are places where stars are formed. The first part of this thesis focuses on the physically correct visualization of *Reflection Nebulae* as one class of astronomical objects which incorporates several general optical properties of interstellar material.

Reconstructing the *Reflection Nebulae*'s spatial structure is a very hard problem, because they do not have a general shape or symmetries that would constrain a reconstruction process. For this reason we decided to determine the three-dimensional shape of *Spiral Galaxies*, from conventional 2D images. From our terrestrially confined point of view the actual spatial shape of the distant astronomical object is, in general, still very challenging to reconstruct. One has to take into account several physical information about the object, like a general shape evolved from its formation [43]. On the other hand, one could take into account different band-filtered data from the object to gain more insights of the material it consists of, to recover its form. Moreover, one could use several different snapshots from different but similar looking objects and try to fill out missing 3D information from the other objects. Our approaches take into account only CCD image information and reconstruct a plausible three-dimensional volume model to be able to authentically visualize existing galaxies.

Furthermore, the visualization of *Reflection Nebulae* and reconstructed *Spiral Galaxy* models enable us to create impressive fly-through animations to captivate the audience as well as illustrate the underlying physics for educational purposes.

This thesis is structured in the following way. After an introductory motivation and contribution section, we outline fundamental previous work to give an overview of the general topics that are related. A more detailed summary of previous work is given in the according sections. Section 2 illustrates a physically correct rendering of *Reflection Nebulae*, while Section 3 proposes two reconstruction techniques for *Spiral Galaxies* from 2D images. We discuss and validate our results in Section 2 and Section 3, before we conclude in Section 4.

1.2 Motivation

Why do we want to visualize and reconstruct astronomical objects, like *Reflection Nebula* and *Galaxies*? Depending on the viewpoint, *reflection nebulae* change significantly in brightness and color. Unfortunately we cannot observe these phenomena due to our fixed viewpoint on or near earth. Therefore virtual astronomy animations are a popular target for planetarium shows. Typically current shows begin from real observational data and include varying degrees of artistic reconstruction to generate nearby views of the nebulae [18].

Realistic visualization of astronomical objects is interesting for several other ap-

plications, e.g. TV documentaries and science fiction movies. It is also needed for educational purposes, if the goal is to gain a deeper understanding of the complex interaction of light and dust in interstellar space. Today's animations are often based on a more artistic than physically correct representation of astronomical objects, even though it is desirable to give a more realistic impression of what can be observed. Tools for visualizing astrophysical simulation results are in common use, but only a few publications address visually realistic, physically correct visualizations of astronomical objects, which could add to the understanding of the underlying data. The increasing interest of 3D visualization in games, movies and education makes it desirable to find a realistic representation of all kinds of objects. The problem of finding a plausible representation of astronomical objects is due to the lack of information that is provided by the observed data. Since our position relative to the object doesn't change, there will be no additional information available. Different wavelength measurement data can give more insights about the object.

Furthermore, we can use our results to provide additional information in telescope applications like recently published by Lințu et al. [41] to increase the understanding of the observed data. Our visualization tool for *Reflection Nebulae* can also be used in conjunction with synthetic 3D nebula models as well as dust distributions derived from real nebulae. That yields not only to faithful renderings but also to the possibility to change the nebula's physical properties. Therefore we can provide an essential function for physicists to validate their research hypothesis in a more interactive environment.

1.3 Contribution

In this thesis we present a new approach for rendering astronomical objects like *Reflection Nebulae* and reconstruction of *Spiral Galaxies* from 2D image data as one exemplary type of astronomical objects.

We propose a significant extension to 3D modeling and visualization of *Reflection Nebulae* based on physical models that are used in astronomical research. We make use of new graphics hardware technologies to visualize synthetically generated nebulae at interactive frame rates. Anisotropic scattering characteristics, wavelength-dependent extinction and multiple scattering are taken into account. Moreover, we provide a novel algorithm for approximating global multiple scattering effects without much loss of rendering performance using a multi-resolution method. This way

we can visualize stars that are surrounded by interstellar dust and scatter light at visible wavelength in a physically correct manner. [42].

For the reconstruction process we use knowledge of other image data that can provide missing three-dimensional information on the shape of the object. That means, we incorporate views of other similar looking galaxy images to complete the volume information. To generate new plausible viewpoints from arbitrary viewing directions, we use an image morphing approach and computerized tomography methods to recover the objects actual volumetric shape. We also provide a method for incorporating infrared information for the reconstruction process to derive a realistic looking *Spiral Galaxy*.

1.4 Related Work

1.4.1 Computer Graphics and Astronomy

One of the pioneer work of using computer graphics for virtual space journeys is done by Blinn [4]. The Jet Propulsion Laboratories decided to produce movies about planned NASA missions for public information purposes. The results were a set of animations called 'Voyager Fly-by Animations', 'Cosmos' and 'The Mechanical Universe' from the late 70ies and early 80ies. Later more astronomical simulation and visualization work appeared in the 1990s [69]. Although there is not a large amount of general related work on this topic one should note some more interesting articles on how to visualize and reconstruct astronomical objects.

Many spectacular 3D fly-throughs of astronomical objects can be experienced in planetariums. These animations are almost always purely artistic pieces of work. One notable exception is the effort by Nadeau et al. [48], [47] who employed massive computational power to create scientifically justified views of the orion nebula. The visualization was shown in the Hayden Planetarium in New York for the show 'Passport to the Universe' and at Siggraph's 2000 evening show 'Volume Visualization of the Orion Nebula' [22]. For their visualization they rely on a 3D model of the Orion nebula that was determined by astronomers over decades from various observational data [71]. The final 150-seconds animation at high-resolution took 12 hours to compute on SDSC's IBM RS/6000 SP supercomputer using 952 processors [22].

Hanson et al. [28] did a lot of work on large scale visualization of astronomical data and more recently on exploring the physical Universe as an enormous environment. Since the Universe is so gigantic in size and dominated by empty space, modeling and rendering an environment like this is a very different task compared with any ordinary three-dimensional virtual world [19]. They introduce a so-called powers-of-ten visualization architecture for this visualization problem, which provides an assortment of scale-independent modeling and rendering methods.

Magnor et al. [43] most recently reconstructed and rendered 3D models of planetary nebulae. Former astrophysical research had shown that planetary nebulae exhibit symmetry characteristics due to physical processes in their formation. By making use of symmetry constraints they are able to automatically reconstruct the axisymmetric structure of many planetary nebulae using volume rendering driving a non-linear optimization. Planetary nebulae, which consist of glowing gas, and reflec-

tion nebulae differ in their optical characteristics. Reflection nebulae do not directly emit light. Instead, light from nearby stars is scattered and absorbed by surrounding interstellar dust, see Section 2.1.1. It means, that the appearance of planetary nebulae only changes in shape and brightness when changing the viewpoint, while reflection nebulae also vary in color due to wavelength-dependent scattering and extinction. This physically different and more complex illumination mechanics requires a rendering algorithm that takes these properties into account.

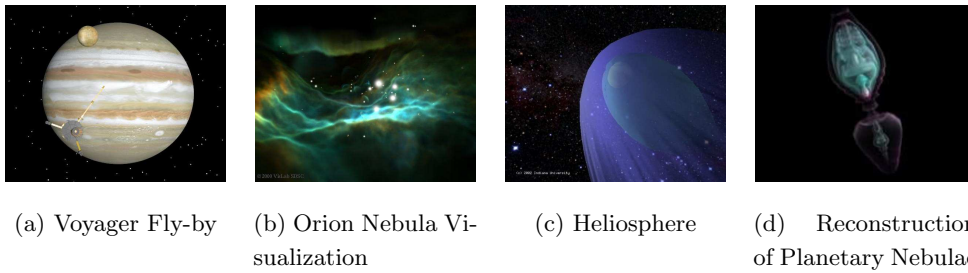


Figure 1.1: (a) shows one image of the so-called Voyager Fly-By Animations created by Blinn [4] image ©<http://www2.jpl.nasa.gov/galileo/dialimages.html>. (b) is a massive volume visualization of the Orion Nebula [22] reconstructed from complex measured data image ©<http://vis.sdsc.edu/research/orion.html> (c) Sun's interaction with its environment. It represents the interaction of the Solar wind with the surrounding interstellar material. Image ©<http://www.cs.indiana.edu/~hanson/> (d) Reconstruction of a three-dimensional volumetric model of planetary nebula M92 from a single image [43]. Image ©<http://www.mpi-sb.mpg.de/~magnor/civr.html>

1.4.2 Astrophysics

In astrophysics, a large body of literature concerns the scattering and absorption of light in interstellar material at different wavelengths [8], [25]. In 1938, Henyey and Greenstein derived analytical expressions for the color and brightness distributions of reflection nebulae for idealized geometrical configurations [30]. Their optical model is widely accepted in the astrophysics and graphics community [25], [70], [62], [51] and forms the basis of our rendering algorithm. Nevertheless there is closely related work that proposes other phase function models, which are experimentally and an-

analytically derived [15], [12] and also describe the scattering properties of interstellar dust grains.

Fundamental basic knowledge and research about galactic astronomy can be found in Binney et al. [3]. They provide a complete overview of colors, morphology and photometry of galaxies, as well as the properties of interstellar material and its effect on observed data. Physicists are mostly interested in specific details of a galaxy instead of the general shape. So, one can find lot of related research about deprojecting galaxies (projecting it to a face-on view) and finding out properties of the spiral arms using fourier transform [23], [57], [58], [2], [61].

However, some work is done by Drimmel et al. [16], who presented a large-scale three-dimensional model of our Galaxy's extinction based on the Galactic dust distribution. The extinction at any point in the Galactic disk can be deduced using a set of three-dimensional Cartesian grids. It can be found by integrating along a line of sights. Nevertheless it is difficult to exploit these results in our settings, so we have to come up with alternative approaches.

1.4.3 Volume Rendering

Synthetically generated or reconstructed volumetric information must be visualized in an interactive way. Rendering *Reflection Nebulae* demands some form of volume rendering, with special attention paid on scattering and absorption effects. Volume rendering, in general, refers to the process of mapping discretized volume elements, called voxels, to RGBA values using a transfer function. There are several volume rendering techniques that composite these values into the frame buffer. Shear-Warp, for example, [40] is fast but is less accurate in sampling compared to the other techniques. The general approach is to axis-align the volume with the nearest face and an off-screen image buffer. The volume is rendered into this buffer and finally warped into the desired orientation. Splatting is an approach using a spherical reconstruction kernel for the influence of a voxel. This kernel is projected onto the image plane and accumulated with other 'splats' to visualize the data set [68]. Another way to visualize the data set is to cast rays through the volume using ray casting [36]. In this technique, a ray is generated for each desired image pixel and passes through the volume to be rendered. The ray is clipped by the boundaries of the volume to save time and is sampled at regular intervals throughout the volume. The data is interpolated at each sample point.

However, to render *Reflection Nebulae* at interactive frame rates, we have to

exploit fast graphics hardware. Real-time methods to directly convey information contents of large volumetric scalar fields are still a challenge to the computer graphics community. To achieve real-time volumetric rendering for large data sets one wants to exploit hardware assisted texture mapping. Fundamentally, these techniques re-sample the volume data, represented in a 2D or 3D texture onto a sampling surface called *proxy-geometry*. This capability was first described by Cullip and Neumann [13] and Cabral [7]. Volume rendering via 3D textures can be performed by slicing the texture block in back-to-front or front-to-back order with planes oriented parallel to the image plane. For each fragment the 3D texture is sampled by trilinear interpolation and the resulting color is blended with the pixel color in the color buffer. Our interactive volume rendering is generally based on the work of Krueger [39] and Scharsach [60], who propose a simple GPU-based ray-casting using a box-shaped proxy geometry and vertex and fragment shader to issue the volume data from a 3D texture.

1.4.4 Reconstruction

Reconstruction methods to recover the three-dimensional structure of an object from two-dimensional images is a challenging problem. Lot of research has been done on reconstruction methods for medical research [26] using computerized tomography (CT) [37]. More recently, the computer graphics community advanced these techniques for image-based modeling [29], [32], [24], [5]. The general idea is to determine the spatial structure of an object from many views by solving an under determined system of linear equations or using back-projection techniques.

Magnor et al. [43] introduce the term constraint inverse volumetric rendering (CIVR) as a GPU-based optimization procedure to reconstruct a volumetric model for planetary nebulae. The basic idea is to use astronomical image data and symmetric structural constraints of the nebulae to reconstruct the three-dimensional volume by an analysis-by-synthesis approach.

3D image analysis and synthesis research relies on recent advances in visualization and graphics research. Curless et al. [14] first employed the term inverse volume rendering to refer to updating volume data from range images. One analysis by synthesis approach is derived in Marchner's Ph.D. thesis [44]. The term inverse rendering is introduced to denote the idea of reconstructing scene properties from image data via computer graphics techniques.

2 Realistic Rendering of Reflection Nebulae

In the first part of this thesis we present a novel approach for the physically correct visualization of synthetic *Reflection Nebulae*. In the following chapter we provide basic physical background knowledge of *Reflection Nebulae* and astrophysical settings in Section 2.1. After an introductory overview of our visualization model in Section 2.2, we describe our interactive rendering algorithm and its implementation in hardware. We will conclude this chapter with results and discussion in Section 2.7.

2.1 Reflection Nebulae

In regions where bright stars lie close to dense gas clouds, light scattered by dust can be directly observed. *Reflection nebulae* are clouds of interstellar dust which are reflecting the light of a nearby star or stars. However, the stars are not hot enough to ionize the gas of the nebula like in emission nebulae but are bright enough to provide the sufficient scattering to make dust visible. The frequency spectrum shown by reflection nebulae is similar to that of the illuminating stars. There are different spectral classes of stars that form the colorful appearance of reflection nebulae. Table A.1 in the appendix shows several types of them and their RGB values which are used in our visualization system. Most stars are also described by their absolute magnitudes additionally to the spectral type O, B, A, F, G, K and M. The stars classification ranges from type O which are very large and bright, to M which is often just large enough to start ignition of the hydrogen. The colors are derived from their observed spectra and their blackbody radiation [10]. Even though reflection nebulae are often associated with very hot and very luminous stars of type O or B (about 50.000-30.000 Kelvin), being strongly blue in color, our interactive visualization tool allows to modify their color in any arbitrary form. That way one can experience how these changes influence the overall appearance.

Such hot and large stars have a shorter lifespan (10^6 compared to 10^{10} years for our sun) resulting in the conclusion that a reflection nebula represents the star's protostellar cloud. A protostellar cloud is a stable cloud that evolves and when it becomes so dense that the heat which is being produced in its center cannot easily escape, the pressure rapidly rises and catches up with the weight, or whatever external force is causing the cloud to collapse. This refers to the fact that it is now destined to become a star. That is one reason, why *Reflection Nebulae* are interesting to astrophysical research, to reveal insights into star formation and stellar composition.

Figure 2.1 (a) and (c) shows Reflection Nebula IC 5146 and Reflection Nebula NGC 1999. One can see very dense regions of interstellar dust surrounding several illuminating stars. Images (b) and (d) show results of artificially generated and rendered nebulae using our visualization tool. The dust scatters illuminating star light in a continuous spectrum which can be identified as the bright regions in the images. Furthermore, the dust extincts direct and scattered light, which are the dark regions in the image. In parts, where the dust of the interstellar medium is concentrated, most light from beyond the dust cloud is absorbed, forming the so called dark-nebula, which highlights a star-poor region. The microscopic particles responsible for scattering consist of carbon compounds and compounds of other elements, in particular iron and nickel [38]. They are usually appearing blue because scattering is more efficient for blue light than for the red part of the visible spectrum (See Section 2.1.1). One can observe similar scattering phenomena on earth as blue skies and red sunsets.

Interesting physics are responsible for the colorful interplay of wavelength-dependent light scattering and extinction that yields a large variation in color hue and brightness. The following sections give more insights into the properties of interstellar dust, scattering and color in astrophysical settings which are important for the realistic visualization of reflection nebula.

2.1.1 Interstellar Dust

In galaxies the space between stars is not empty but filled with cold low density smoke-like gas. It is so rarefied but exceedingly filthy that if the dust would be compressed to the density of ordinary air (that is, by a factor of 10^{21}), the density of smoke in it would be such that objects would disappear at a distance of less than a meter [3]. Some of the smoke particles will later form asteroids, comets and

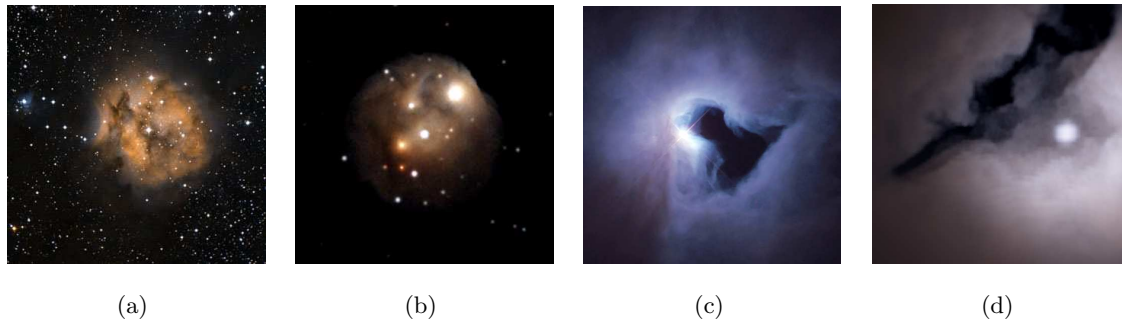


Figure 2.1: (a) Reflection Nebula IC 5146 ©G. Crinklaw [52]. (b) Rendering result image composited with a star field texture. (c) NGC 1999 lies in the Orion Nebula ©NASA/ESA [20]. The very young star illuminates the surrounding interstellar dust. (d) Another rendering result using the visualization tool. One can see the similarities between the real Reflection Nebula and our rendering.

planets but the majority will hang around the interstellar space and obscure the view. Depending on the density, size and temperature of a given dust cloud, the hydrogen in it can be neutral, forming so-called HI regions, ionized, that are the so-called HII regions, or molecular, forming molecular clouds. There are also regions in the galaxy in which stars lie close to dense dust clouds and the light scattering effects by dust can be directly observed. These regions are called *reflection nebulae*.

The *interstellar dust* not only dims the light from stars, it also reddens it. There are two different explanations for that. Firstly, some photons are scattered out of the line of sight. Secondly, photons are absorbed, converting their energy into heat. The size of many of the individual grains of space dust are about the same as the wavelength for blue light varying between $100nm$ and $1\mu m$. Figure 2.2 shows the effect when blue part of the spectrum waves encounter a dust grain. The grain is a big obstacle for the light and scatters off the grain in a random direction. Redder light waves are less affected by dust, since their wavelength is about the same size or greater than the size of the grain. This means, that much of the blue light emitted from stars in the galaxy behind the dust clouds gets scattered away from our direct view, making the stars, as we see them through the clouds look redder and dimmer than they actually are [3]. Furthermore photons that get absorbed from the dust convert the energy into heat. Thus, the dust transforms blue light

into far infrared light and the absorption of starlight warms dust grains to $\approx 10K$. At this temperature they radiate significantly at $\lambda \approx 200\mu m$, and photons of this wavelength can escape.

One should note here that dust distributions in the far-infrared can be observed by special telescopes like Spitzer Space Telescope [64] resulting in the most spectacular images of today's astronomy. These images can be used as additional information in an explicit reconstruction process of astronomical objects, since the distribution of interstellar material is an important factor of the overall shape and appearance. We discuss this topic in more detail in Section 3.6.



Figure 2.2: We show dust is heating up and scattering blue light in arbitrary directions. Smaller wavelengths of blue light encounter the dust grain and are scattered away. Red light is scattered less, encounters the particle and heats it up to 10 Kelvin. At this temperature it releases photons at higher wavelengths. This is called reddening the starlight.

The scattering properties of a single particle are accurately described in the Mie scattering theory [65]. One can model the physical scattering properties of dust by a scattering phase function, e.g. the Henyey-Greenstein phase function. To illustrate the effect of a very large number of different dust grains illuminated by polychromatic, unpolarized light, Gordon describes in [25] that scattering of photons by dust grains can be parameterized by two variables:

First, a single scattering albedo, $a = [0, 1]$, which is the ratio of radiosity to irradiance, i.e. the average percentage of radiation that is being scattered on a single dust particle. It becomes zero when the dust is completely black and all incident radiation is absorbed and one when all photons are deflected by the particle. One can imagine this as the average absorption coefficient σ_{abs} and scattering coefficient

σ_{sct} that define together the dust albedo

$$a = \frac{\sigma_{sct}}{\sigma_{abs} + \sigma_{sct}} = \frac{\sigma_{sct}}{\sigma_{ext}} \quad (2.1)$$

and give rise to light extinction in turbid media. Summing up both yields the extinction coefficient $\sigma_{ext} = \sigma_{abs} + \sigma_{sct}$. Since the size of dust particles is roughly on the order of the wavelength of visible light some wavelengths are scattered more than others which leads to anisotropic scattering, i.e. the scattering probability is direction-dependent. For spherical particles and unpolarized light, scattering is symmetric around the incident radiation direction, so the scattering probability varies only with the deflection angle μ , as shown in Figure 2.3.

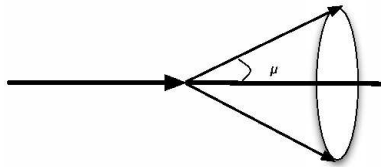


Figure 2.3: The phase function has an axial symmetry and depends only on μ . It is often expressed as the function of the cosine of the scattering angle.

The second parameter that describes scattering of photons by dust grains is the scattering phase function, $\Phi(\mu)$, where μ is the scattering angle. In most astrophysical research studies the Henyey-Greenstein phase function is used as $\Phi_{HG}(\mu)$, which is a probability density function $P(\mu)$ [31]. It is used to characterize the angular distribution of scattered light and is characterized by $g = [-1, 1]$, which parameterizes the function by the average cosine of the scattering angle:

$$\Phi_{HG}(\mu) = P(\mu) = \frac{1 - g^2}{(1 + g^2 - 2g \cos \mu)^{3/2}} \quad (2.2)$$

The variable g is referred to as a measure of the scattering phase function asymmetry, i.e. an anisotropy factor that varies from -1 (complete back scattering) over 0 (isotropic scattering) to +1 (complete forward scattering). That means, the phase function describes, how much light is scattered in a certain direction based on the angle μ . It turns out that these two parameters are sufficient to correctly characterize the optical properties of the interstellar dust to within today's observational accuracy [31], [30].

The Henyey-Greenstein phase function is widely accepted ([25], [70], [62], [51]) as a good approximation for scattering phase functions at wavelengths between

$\sim 0.4\mu m$ to $1\mu m$, but fails at shorter and longer wavelengths [15]. Some other analytical expressions have been proposed by Cornette & Shanks [12] to provide a more realistic approximation for the single scattering of unpolarized light. More recently, Draine [15] derived another phase function to overcome the problem at shorter and longer wavelengths.

Consider the following new phase function

$$\Phi_{\alpha}(\mu) = \frac{1 - g^2}{1 + \alpha(1 + 2g^2)/3} \frac{1 + \alpha \cos^2 \mu}{(1 + g^2 - 2g \cos \mu)^{3/2}} \quad (2.3)$$

with two adjustable parameters α and the anisotropic factor g . For $\alpha = 0$, equation (2.3) reduces to the Henyey-Greenstein phase function (2.2). For $g = 0$ and $\alpha = 1$ this reduces to the *Rayleigh* scattering phase function, which is frequently used in computer graphics for approximating scattering caused by small molecules in the air [51]. For $\alpha = 1$ this corresponds to the phase function proposed by Cornette & Shanks [12]. However, Draine [15] is pointing out that the phase function (2.3) has good properties for wavelengths $\lambda \lesssim 0.4\mu m$ and $\lambda > 1\mu m$ but isn't satisfying when dust is strongly forward-scattered ($g \gtrsim 0.6$).

For our visualization algorithm it is possible to use any arbitrary scattering phase function. However, we decided to implement the Henyey-Greenstein phase function (2.2), because it is widely used and accepted. As pointed out in [25], it is reasonable to say that for the current measured and analyzed data at visible wavelengths the albedo is $a \approx 0.6$ and the anisotropic factor is $g \approx 0.6$. One reason is that observations have shown that the optical properties of the dust throughout the galaxy are the same within acceptable error margins. Figure 2.4 shows how varying the properties of scattering and absorption of the interstellar dust grains changes the overall appearance of the nebula. Increasing albedo from $a = 0.6$ to $a = 0.8$ intensifies the overall brightness. Enhancing forward scattering from $g = 0.6$ to $g = 0.8$ brightens already light regions, while the overall contrast increases.

However, since these factors are consistent over visible wavelengths the average scattering coefficient σ_{set} varies with wavelength. While light from the blue part of the visible spectrum is scattered almost twice as often as that from the red part, the ratio of scattering probability at red and blue wavelengths varies for different regions in the Galaxy [3].

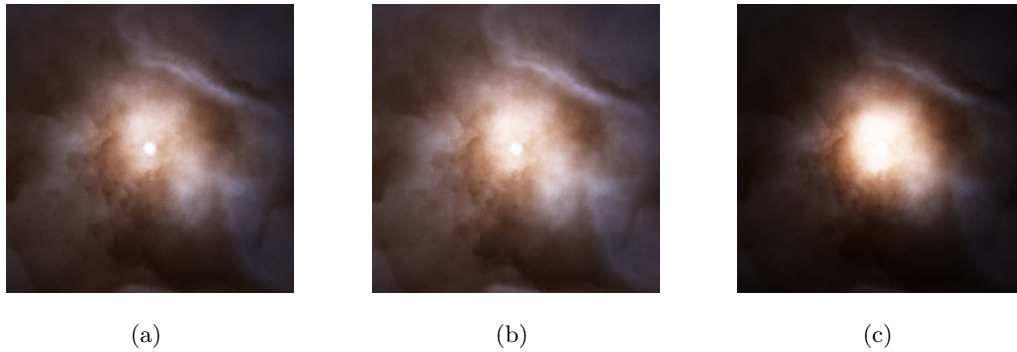


Figure 2.4: Rendering results for different scattering probability lookup tables with varying albedo a and anisotropic factor g . From left to right image (a) albedo $a = 0.6$, anisotropic factor $g = 0.6$, (b) albedo $a = 0.8$, anisotropic factor $g = 0.6$, (c) albedo $a = 0.6$, anisotropic factor $g = 0.8$

2.1.2 Colors and Astrophysical Settings

In astrophysical settings the object's color is commonly referred to in terms of band-filtered observations. It is important to understand how colors are derived from measurements and how they are influenced by the interstellar material. We will also show how to derive the relative extinction coefficients that result in a variable for our visualization model, when computing the overall extinction.

The Johnson color system [3] features three bands in the visible range of the spectrum that approximately correspond to the RGB colors in computer graphics. Table A.2 in the appendix shows the Ultraviolet-Blue-Visible-Red-Infrared (UBVRI) bands for the combined Johnson-Cousins-Glass system and the corresponding wavelength λ . The peak filter wavelengths and half-maximum pass bandwidths for the blue band (B band) are $445 \pm 47 \text{ nm}$, $551 \pm 44 \text{ nm}$ for the green (V band) and $658 \pm 69 \text{ nm}$ for the red band (R band) [3].

As pointed out earlier in Section 2.1, the stars can be classified in several types, depending on their spectra and their blackbody radiation. The color of a star is commonly described in terms of apparent magnitude m , which is a measure of brightness for every band. One should note here, that the sensitivity of a light detector as used in telescopes varies with the wavelength λ of the light, and the way in which it varies depends on the type of light detector. For this reason, it is necessary to specify, how the magnitude is measured in order for the value to

be meaningful. By taking the difference in magnitudes measured in two different bands, from the widely used UBVRI system, we can form a color or color index. A color index is usually written using the letters that denote the different filters involved; that is (B-V) and (U-B) for the standard UBVRI system. It essentially measures the ratio of stellar fluxes near the effective wavelength of the two bands. Since this quantity depends only on a ratio of fluxes, it measures a property of an object's spectrum that is independent of its distance. This is highly desirable, given the uncertainties in the distances to many astronomical objects. The V band was chosen for spectral purposes and records magnitudes closely related to those seen by the light-adapted human eye. Given only the apparent magnitude without any further qualification, it is usually the V magnitude that is referred to.

As we pointed out before, the scattering coefficient σ_{sct} varies with the wavelength, so does extinction. Extinction A or optical depth π_{opt} can be determined as the product of the total path length l (see Figure 2.6) and the average extinction coefficient σ_{ext} along the way

$$\pi_{opt} = A = \sigma_{ext} \cdot l. \quad (2.4)$$

In astrophysical settings the relative amount of extinction A at different wavelengths λ can be expressed as the ratio of total to selective extinction R_v . This is related to the average extinction law $A(\lambda)/A(V)$ which is derived for diffuse and dense regions of interstellar dust. The mean extinction law only depends on one parameter: the value of total-to-selective extinction $R_v = A(V)/E(B - V)$, where the extinction $E(B - V)$ is $(B - V)_{obs} - (B - V)_0$, which refers to the observed and intrinsic colors of the target.

For visible and near infrared wavelength one can derive two wavelength-dependent coefficients $a(x)$ and $b(x)$ [8]. The mean R_v -dependent extinction law then takes the form

$$A(\lambda)/A(V) = a(x) + b(x)/R_v \quad (2.5)$$

To obtain the values F_R , F_V and F_B for the different wavelengths, we compute $F_{RVB} = a(x) + b(x)/R_V$ with

$$\begin{aligned} a(x) = & 1. + 0.17699y - 0.50447y^2 - 0.02427y^3 + \\ & + 0.72085y^4 + 0.01979y^5 - 0.7753y^6 + 0.32999y^7 \end{aligned} \quad (2.6)$$

and

$$b(x) = 1.41338y + 2.28305y^2 + 1.07233y^3 - 5.38434y^4 - 0.62251y^5 + 5.30260y^6 - 2.09002y^7. \quad (2.7)$$

for $y = (x - 1.82)$ where $x = 2.27$ for the B band and $x = 1.43$ for the R band [8]. This results in three coefficients, which describe relative extinction: F_R, F_V, F_B .

$A(\lambda)$ is the absolute extinction at any wavelength which is expressed relative to the $A(\lambda_{ref})$ - the absolute extinction at a chosen reference wavelength - that is $A(V)$ for historical reasons. Therefore, one expresses the extinction A for B and R bands with respect to the V band.

$R_v = 3.1$ is a standard value which can be found in [3] for not densely distributed dust. That yields the relative extinction factor $F_B = A_B/A_V = 1.324$ and $F_R = A_R/A_V = 0.748$. However, scattering increases more slowly with decreasing wavelength in dense clouds. In that case, astronomical research has shown [8], that one should use $R_v = 5.0$ which gives $F_B = A_B/A_V = 1.2$ and $F_R = A_R/A_V = 0.8$ as relative extinction factors.

Figure 2.5 illustrates the different appearance of the nebula, if the dust is not densely distributed, i.e. $R_v = 3.1$ in contrast to $R_v = 5.0$. Colors in the left hand image are somewhat more vivid due to the fact that relative extinction rises more steeply with decreasing wavelength.

In the implementation F_R, F_B and F_V are essentially factors responsible for wavelength dependent extinction and are multiplied with the optical depth π_{opt} and scattering depth π_{sct} . See Section 2.3.3 for more details, where these parameters are used in the implementation.

2.2 Visualization Model

When rendering a participating medium, we have to take into account several volumetric effects such as absorption and extinction, anisotropic scattering and effects of multiple scattering. To simulate the scattering and extinction properties of *reflection nebulae* we use a discretized volumetric model consisting of box-shaped volume elements, called voxels. We assign every voxel its own scattering depth π_{sct} which is proportional to the dust density. For simplicity we assume a single star in the center of the volume. (Section 2.5 describes changes in the model for several stars.)



Figure 2.5: Two images showing the difference in overall nebula appearance using widely accepted total-to-selective extinction factors R_v . Left image with $R_v = 3.1$ for not densely distributed dust. Right image with $R_v = 5.0$ for dense dust clouds.

The second value that we assign to each voxel is the received radiance at the voxel's position (x,y,z) . It depends on its distance r from the star and the optical depth.

Optical depth is a measure of how much of the light is absorbed while travelling through a medium, such as the atmosphere. A completely transparent medium has optical depth zero, while another type of medium, for example interstellar dust, has higher optical depth. It depends on the frequency of radiation as well as the type of the medium. As we saw before, the blue part of the visible spectrum is strongly affected by interstellar dust, so dust clouds have high blue-light optical depth.

In our case optical depth is computed by $\pi_{opt} = \pi_{sct}/a$ accumulated all the way from the star to the voxel. Figure 2.6 shows the radiance L_{ill} that is received by a voxel at a volume position (x,y,z) at distance r from the star depends on the radiant power Φ_{star} of the star and the optical depth π_{opt} from the star to the voxel and is defined as

$$L_{ill} = \frac{\Phi_{star}}{4\pi r^2} \exp^{-\int_0^r \pi_{opt}(r') dr'} \quad (2.8)$$

The star's radiant power Φ_{star} falls off with increasing distance r and is multiplied by an exponentially decreasing factor of optical depth π_{opt} also depending on r . The amount of light reaching a voxel is partially scattered on the dust particle at different wavelengths. $P(\pi_{sct}, \mu)$ denotes the fraction that is being scattering in the observers direction. It can be described as the probability scattering density for a given viewing direction and scattering depth π_{sct} , which is precomputed in a

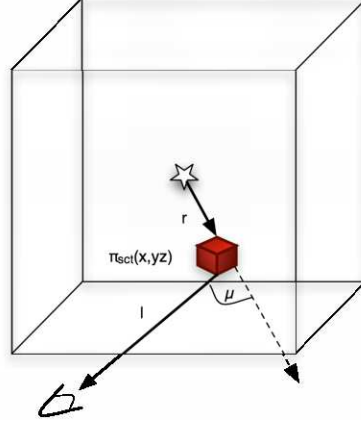


Figure 2.6: Discretization of the space around the illuminating star into voxels. Each voxel receives light that is attenuated along direction r . A fraction of the light is scattered depending on scattering depth π_{sct} and the angle μ between viewing-direction and radial illumination vector. More light is extinct on distance l from the voxel to the observer. Here, we assume the star is in the middle. We can also incorporate several stars. See Section 2.5

Monte-Carlo simulation. See Section 2.2.2 for details on how the probability density scattering table is derived.

$$L_{sct} = L_{ill} P(\pi_{sct}, \mu) \quad (2.9)$$

Once we have the amount of scattered light L_{sct} that the observer perceives for a single voxel we accumulate it along the ray-path to the observer. Again taking optical depth π_{opt} into account to attenuate the light from the voxel at position (x, y, z) to the viewer which has exponential falloff depending on distance l (Figure 2.6).

$$L = L_{sct} \exp^{-\int_0^l (\pi_{opt}(l')) dl'} \quad (2.10)$$

As described in Section 2.1.2 the scattering is wavelength-dependent. Therefore we have to compute scattering and extinction separately for the red, green and blue channel since π_{sct} and π_{opt} vary with wavelength.

2.2.1 Voxel Characteristics

Most important for the correct visualization are the voxel scattering characteristics. For a voxel in our visualization model only a fraction $P(\pi_{sct}, \mu)$ of the radiance is scattered depending on the scattering depth π_{sct} and the deflection angle μ between the incident ray from the star and the viewing direction. We use a Monte-Carlo simulation to precompute a scattering probability table for 1000 scattering depth values between 0 and 10.0 and 72 directions from $\cos\mu = -1$ to $+1$. Figure 2.7 shows direction dependent scattering for a voxel given different scattering depths π_{sct} . As one can see the directional dependence changes drastically with voxel scattering depth because of multiple scattering within the voxel. The diagram shows, how increasing dust concentration leads to an increasing amount of scattering in the beginning but decreases for very dense dust clouds. Forward scattering takes place mostly for thin layers of dust while backward scattering appears in more dense regions.

Figure 2.8 illustrates the combined effects of anisotropy, multiple scattering, absorption, wavelength dependence and 3D distribution of dust on reflection nebulae appearance. In this case we initialize the volume with a thin layer of non-zero scattering depth π_{sct} values in the front and the back of the otherwise empty nebula. The scattering depth π_{sct} increases with the angle from 0.0 up to 10.0. Forward scattering shows how increasing scattering depth yields higher extinction and light becomes redder and dimmer. Light that is back-scattered takes mostly the color of the illuminating star.

As can be seen in Figure 2.4 we created the lookup table for different albedos a and anisotropic factors g (each for 0.6 and 0.8). As suggested in [25] we use the Henyey-Greenstein phase function (2.2) with $a = 0.6$ and $g = 0.6$ in a normal setup. However, it is possible to use any arbitrary single-particle phase function, e.g. as pointed out earlier [15] and [12].

When the dust density is high it is likely that the photon is not only scattered once but several times within the voxel. Using the Monte-Carlo simulation it is possible to take multiple scattering into account when deriving $P(\pi_{sct}, \mu)$ and we can correctly determine multiple scattering on a local per-voxel level. See Section 2.4 for a novel global multiple scattering approach.

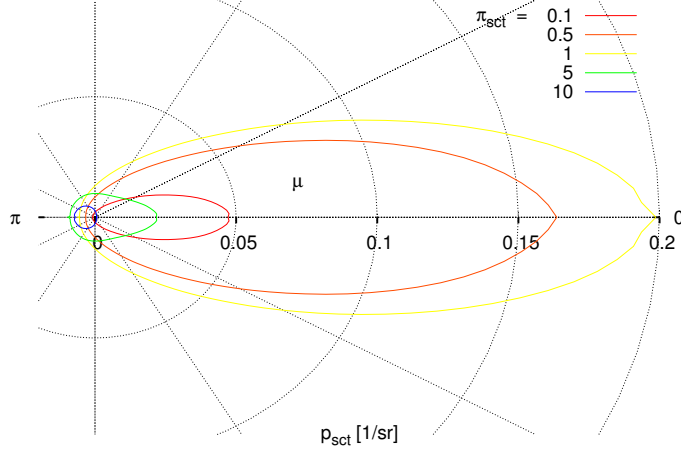


Figure 2.7: Voxel scattering probability $P(\pi_{sct}, \mu)$ for albedo $a = 0.6$ and anisotropy factor $g = 0.6$. Forward scattering is shown to the right with $\mu = 0$. With increasing π_{sct} the scattering first increases then decreases again. Relative forward scattering depth also decrease until for the highest scattering depth $\pi_{sct} = 10.0$, more light is back-scattered due to the extinction within the voxel. Image provided by Andrei Lințu.

2.2.2 Scattering Table Generation

The generation of the scattering probability table is described in detail in [42], in which we propose using a Monte-Carlo simulation. Monte-Carlo simulation describes a class of methods for simulating the behavior of various systems in a non-deterministic, i.e. stochastic, manner.

We are simulating N photons initialized with a weight $w_0 = 1/N$ for every photon. We place it at a starting position x_0 of a sphere with diameter l in the size of a voxel length. In the simulation, we now trace the photon until it leaves the sphere and add w_i to a direction bin $B(\cos \mu)$ of the outgoing angle $\cos \mu = d_i \cdot d_0$. The next scattering event takes place after a travel length of

$$r = -\ln(1 - u)/\sigma_{sct} \quad (2.11)$$

with $u = \text{rand}[0, 1]$ and the scattering coefficient $\sigma_{sct} = 1/\bar{r}$. This gives us a new 3D coordinate $x_{i+1} = x_i + r \cdot d_i$. For the new scattering direction we have to take anisotropy into account which can be evaluated from the adopted Henyey-Greenstein

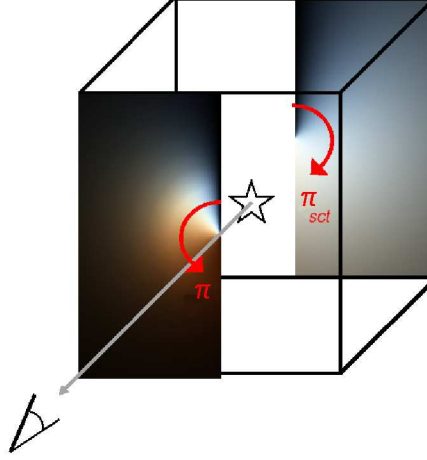


Figure 2.8: Light scattered forward and backward with angular increasing scattering depth π_{sct} from 0 to 10.0.

phase function (2.2) to account for our accumulation bins $B(\cos\mu)$.

$$\cos\mu = \frac{1}{2g} \left(1 + g^2 - \left(\frac{1 - g^2}{1 - g - 2gv} \right)^2 \right) \quad (2.12)$$

For anisotropic scattering, i.e. $g \neq 0$ the horizontal angle is $\phi = 2\pi \cdot w$ with $v, w = rand[0.1]$. For isotropic scattering, i.e. $g = 0$, one can derive

$$\cos\mu = 2v - 1 \quad (2.13)$$

To compute the new direction $d_{i+1} = (d'_x, d'_y, d'_z)$ in Cartesian coordinates we have to distinguish between two cases: if the previous direction $d_i = (d_x, d_y, d_z)$ is parallel to the z-axis then

$$d'_x = \sin\mu \cos\phi$$

$$d'_y = \sin\mu \sin\phi$$

$$d'_z = \frac{d_z}{\|d_z\|} \cos\mu$$

otherwise

$$d'_x = \frac{\sin\mu}{\epsilon} \cdot (d_x d_z \cos\phi - d_y \sin\phi) + d_x \cos\mu$$

$$d_y' = \frac{\sin \mu}{\epsilon} \cdot (d_y d_z \cos \phi - d_x \sin \phi) + d_y \cos \mu$$

$$d_z' = -\epsilon \sin \mu \cos \phi + d_z \cos \mu$$

with $\epsilon = \sqrt{1 - d_z^2}$. Finally, the photon weight is multiplied with the albedo to account for the attenuation of every single particle $w_{i+1} = w_i \cdot a$. The simulation is executed for all N photon particles. The accumulated values in bin $B(\cos \mu)$ represent the row of entries for π_{sct} in the scattering lookup table $P(\pi_{sct}, \mu)$.

2.3 Ray-Casting Implementation

Real-time methods for volumetric rendering are still a challenge to the computer graphics community. To achieve interactive or real-time volume visualization for large data sets one wants to exploit hardware assisted texture mapping. This capability was first described by Cullip and Neumann [13] and further developed for more advanced medical imaging by Cabral [7]. They demonstrated that interactive volume reconstruction and interactive volume rendering was possible with hardware provided 3D texture acceleration. Nowadays, it is a widely accepted technique to render medium sized data sets at interactive frame rates. With rapidly increasing development of computer graphics hardware this technique got extended several times, e.g. [67], [21] and [17]. However, the key to an efficient GPU volume ray-casting is to find an effective stream model that allows to continuously feed multiple, data-parallel fragment units on recent GPU chips. Furthermore the number of operations on each fragment should be minimized.

2.3.1 GPU-Ray-casting

Our rendering algorithm relies on graphics hardware based ray-casting. The basic idea, proposed by Krueger [39], is simple. The dataset is stored in a 3D texture to take advantage of built-in trilinear filtering. Then a bounding box geometry is created encoding the position in the dataset as colors, i.e. we can interpret these as 3D texture coordinates. Figure 2.9 shows the bounding box for the front-faces (left) and back-faces (right).

The proposed multi-pass algorithm renders the front-faces of the volume bounding box first to a 2D RGB texture *TEX*. In a second pass we render the volume bound-

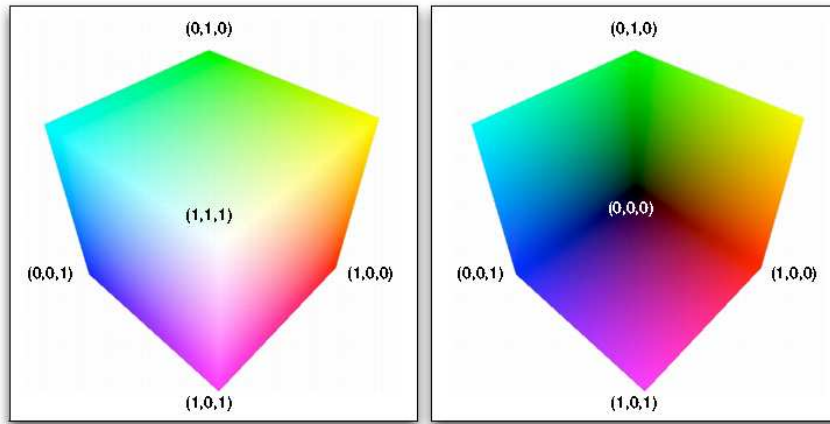


Figure 2.9: Left: Front-faces of the rendered cube. Right: Back-faces of the cube.
The subtraction result gives us the ray direction and ray length

ing box again but this time only the back-faces. The result is stored in a 2D RGBA texture *DIR*. In this pass the fragment shader is issued to fetch the corresponding texel for every fragment of *TEX* and perform a subtraction with the current rendering. That way we accomplish the ray-direction by normalizing the subtraction result. In addition the non-normalized length of the ray is computed and rendered into the alpha channel of the texture *DIR*. In a third pass we use new capabilities of graphics hardware to loop over the entire volume and step in regular intervals along the viewing direction by looking up the rendering results in *DIR*. We summarize the necessary steps:

- (i) Render front faces of the color cube into off-screen buffer like Pbuffer or Frame-bufferobject (FBO).
- (ii) Render back faces and compute difference between the back faces and the off-screen result from the first pass. Normalize the result to get the ray direction and store it together with the initial unnormalized value in a separate off-screen buffer.
- (iii) Render front faces again, taking the colors as starting points for the rays and cast along the viewing vector
- (iv) Loop through the 3D texture in constant steps in ray direction and account for all voxel in the line of sight.

2.3.2 Fly-through Application

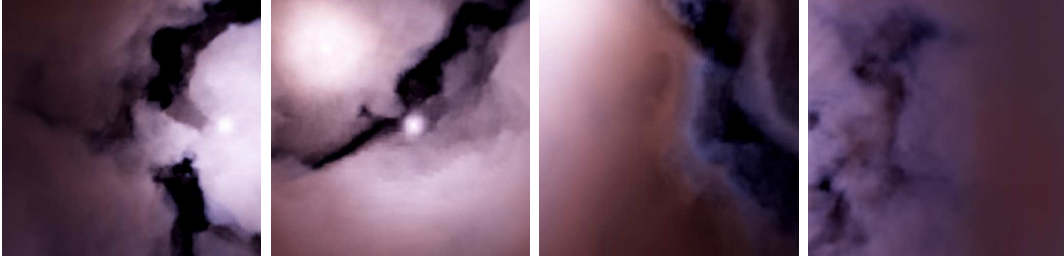


Figure 2.10: Exemplary rendering results of a fly through synthetic nebulae

For many applications such as the exploration of astronomical objects it is interesting to move the viewpoint into the volume and evaluate the data in a fly-through mode. This doesn't matter for our rendering algorithm as long as the camera doesn't touch the geometry. But as soon as the near clipping plane intersects with the bounding geometry holes appear due to the lack of color encoded information that is necessary to compute the ray-direction. To solve the problem that arises when we try to interactively explore the volume we use a way suggested by Scharsach [60]. Whenever an intersection with the near clipping happens, we have to fill the geometry with correctly colored values. The approach would be to draw the near clipping plane first which encodes the color as absolute position in the dataset. Then render the front faces ensuring that whenever there is no front face to start from, due to clipping, the position of the near clipping plane is taken. This doesn't work if there is another object behind the current one, which front faces are visible and would misleadingly be taken as starting positions for the ray. A way to avoid this is to first draw back-faces in the depth buffer only, retrieve the z-value of the nearest back face and render the front-faces afterwards. This way no front face of a rear object will be drawn because its z-value would be greater than the nearest back face. We can summarize the steps as follow:

- (i) Render the color coded near clipping plane with depth buffer turned off.
- (ii) Render back faces only to the depth buffer ensuring that only the first front faces (front faces of our object) will be drawn.
- (iii) Render front faces with depth buffer enabled, resulting in the correct starting position for all rays.

Figure 2.19 gives an example of fly-around and fly-through sequences for synthetic nebulae to illustrate the change in shape and color by moving the viewpoint.

2.3.3 Reflection Nebula Rendering on GPU

So far, we talked about our visualization model based on several physical properties and a general GPU-rendering solution. Now, we want to focus on our GPU visualization algorithm for reflection nebulae, which works as follows.

The rendering algorithm runs in four passes. The first two passes prepare the proxy-geometry, i.e. render front- and back-faces, and compute the ray direction and length. In the third pass we first issue a fragment program to step along each viewing vector from front to back in voxel length intervals. The precomputed scattering table is uploaded as a 2D floating-point texture to graphics memory (see Section 2.2.2 for its creation). A 3D floating point texture stores scattering depth π_{sct} and precomputed illumination L_{ill} for each voxel. In a second 3D texture, we additionally store emissive radiance L_{em} per voxel. The 3D emission texture allows us to simulate ionized, glowing gas clouds that may be intermixed with interstellar dust, as well as to visualize the illuminating star.

At each step along the ray we lookup local scattering depth π_{sct} , voxel illumination L_{ill} and emissive RGB radiance values L_{em} from the 3D volume textures. These values are trilinear interpolated on graphics hardware.

Then the fragment program computes the angle between the ray direction and the incident star-light direction (radial illumination vector) (Figure 2.6) and queries the scattering probability. The values for π_{sct} and μ determine row and column values for our lookup table. The interpolated result gives $P(\pi_{sct}, \mu)$ and L_{sct} can be determined (2.9). The L_{em} radiates isotropically in all directions and can be added directly to the scattered radiance L_{sct} . Both contributions undergo extinction $\sigma_{ext} = \pi_{sct}/a$ on the line of sight L_{obs} (2.10) which is accumulated by stepping along the ray. The extinction depth is evaluated along the ray as $\pi_{opt}^{i+1} = \pi_{opt}^i + \pi_{sct}^{i+1}/a$. Note that the computation is done separately for the red, green and blue channel by weighting π_{sct} according to Section 2.1.2.

Furthermore we query uniform variables to be able to interactively change the star color, extinction offset π_{opt}^i , which is the amount of extinction that we undergo from earth to the nebula and albedo. Additionally we can modify the ratio of total-to-selective extinction R_v , resulting in different values for F_R , F_V and F_B for wavelength-dependent extinction and image correction values (see Section 2.3.4) to

optionally correct for color intensity. The image correction is done in the final fourth pass. The result is an artificial but physically plausible visualization of a *Reflection Nebula* as can be seen in Figure 2.19.

2.3.4 Intensity Correction

In general, extinction (scattering + absorption) methods do not look convincing without a reasonable intensity correction like gamma correction or high-dynamic range (HDR) rendering. This is because these methods can easily generate images that are too bright or have too dark regions. However, one can compensate for these effects by rendering to a floating-point buffer and then correct for over saturated regions in the same rendering pass. Thus, we have to normalize the values from our computation to fall within a displayable RGB range between 0 and 1, by doing

$$p.rgb = (col.rgb - min)/(max - min) \quad (2.14)$$

in the final rendering pass. There are sophisticated ways to compute the *min* and *max* directly in the fragment shader. Nevertheless, we decided to do a complete *readback* from the frame buffer to find these values, even though it is not the most efficient approach.

There are different mapping functions to perform color and image intensity corrections. We can perform a gamma correction which has several disadvantages over HDR rendering (see Figure 2.11). Gamma correction leads to over exposure or color bleaching for lower gamma values. Simple HDR rendering on the other hand can be achieved by using an exponential curve with an adjustable exposure constant. Figure 2.11 shows three different parameters for gamma values to compute the pixels color

$$p = color^{gamma}. \quad (2.15)$$

As one can see increasing the exponent leads to overexposed colors. HDR mapping on the other hand uses the exposure equation simply as

$$p = 1.0 - \exp(-exposure \cdot color). \quad (2.16)$$

The exposure constant works like the aperture of a camera or the pupil of our eyes. Where there is too much light, the exposure constant is reduced to let in less light and visa versa. Figure 2.11 shows the different curve and color mapping for three exposure values. The relative brightness and contrast for each color is better preserved [54].

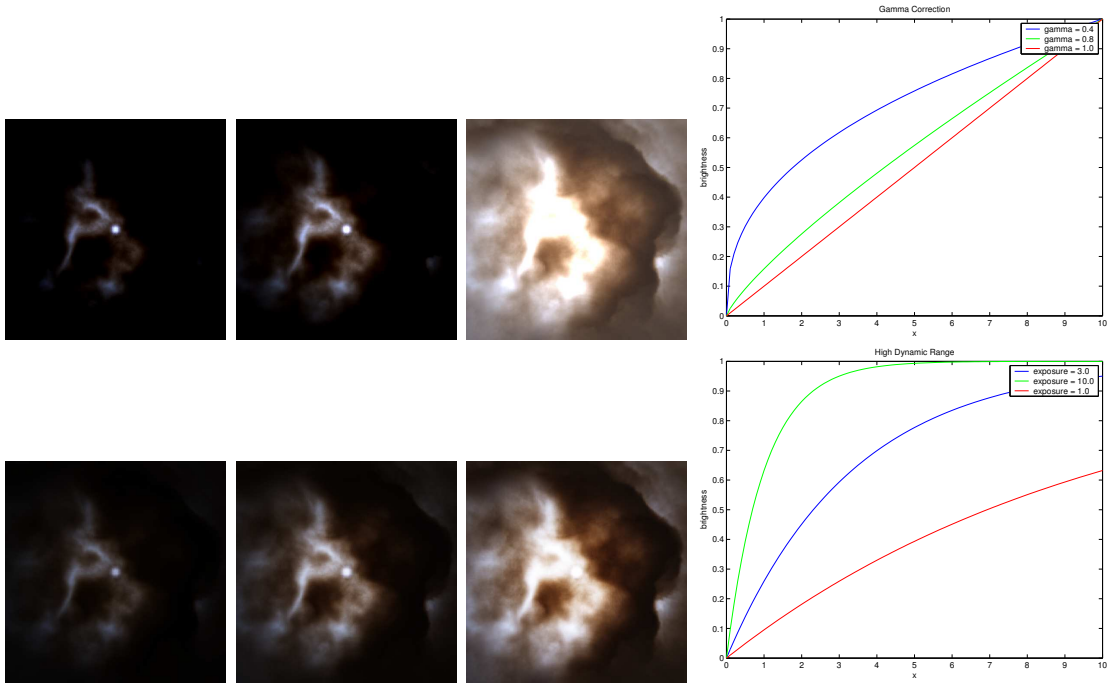


Figure 2.11: Rendering of a synthetic reflection nebula with gamma correction and simple HDR rendering. Upper: Results for gamma corrected images. The diagram shows different gamma correction values for $\gamma = 0.4, 0.8, 1.0$. Lower: The same rendering with simple high dynamic range mapping and a diagram showing several exposure times $\text{exposure} = 1.0, 3.0, 10.0$.

2.4 Multi-resolution Multiple Scattering

Particles in the atmosphere and in interstellar medium are responsible for wavelength-dependent effects like anisotropic scattering and other optical phenomena like rainbows and halos in the sky. To simulate these effects physically accurate a full and computational expensive radiative transfer method is needed. Unfortunately, this is a difficult problem, because the number of paths a photon can take in dust or clouds rises exponentially with the expected number of scattering events.

Lot of work is done to simulate the effects of multiple scattering. Early work in modeling multiple scattering involved spherical harmonics [36]. More recent, diffusion systems model light with a first order angular distribution using a multi-grid scheme [63] and with multiple order scattering reference patterns [51]. Other

methods like [45] and [55] generate high accuracy solutions but are computationally expensive when fine angularly dependent effects are desired.

More recently, Riley et al. [59] solved the rendering of angular dependent effects in the presence of multiple scattering by designing a lighting approximation based upon multiple scattering phase function. It is similar to our method, since they use a precomputed multiple scattering phase function and angle dependent scattering probabilities. However, our approach [42] can be used with any arbitrary (also experimentally measured) single-particle phase function. That is because we use a Monte-Carlo simulation in a preprocessing step to generate our scattering probability table.

In a reflection nebula the dust concentration can reach high densities, resulting in multiple scattering events. Using a Monte-Carlo simulation we can determine the scattering probability on a local per-voxel level (see Section 2.2.2). The scattering depth π_{sct} denotes an average number of the scattering events within a voxel. Thus, albedo values $a < 1$ dim the light in case of multiple scattering events because each scattering event absorbs a fraction $(1 - a)$ of the scattered light. After n events only a^n of the initial radiance is still present.

However, photons that leave a voxel but are scattered back into it from neighboring regions are not taken into account this way. One should note here, that these neglected photons in a single-scattering approach can only increase the overall observed radiance in the final rendition, when taken into account as multiple scattered photons.

To convincingly approximate the global effect of multiple scattering we pursue a new multi-resolution rendering approach. During preprocessing the full-resolution model V_i is repeatedly down-sampled along all dimensions to $V_{i+1...n}$. The scattering depth π_{sct} , again which is proportional to the interstellar dust density, from eight neighbors is averaged and twice this average value is assigned to the corresponding lower-resolution voxel. One can imagine the factor two as a linear extension of the voxel size. This means that increasing the size actually increases the probability that a photon is scattered back from neighboring voxel. Also the illumination is computed separately for each down-sampled volume. The resulting image becomes brighter for a lower resolution model than for a higher resolution volume.

During the visualization we render a pyramid of images from corresponding volume resolutions V_i . The algorithm scheme is shown in Figure 2.12. To obtain the final image we iteratively down-sample the image I_i for every resolution stage which

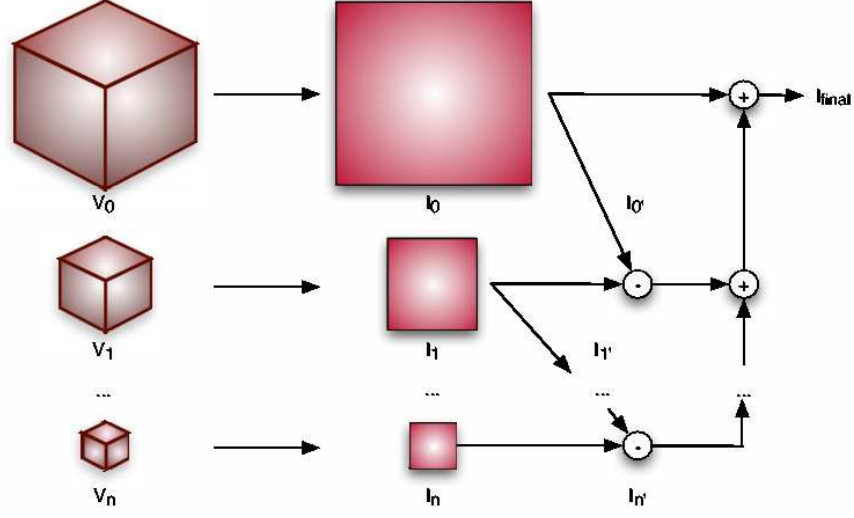


Figure 2.12: Multi-resolution rendering to incorporate for global effects of multiple scattering. A resolution pyramid is rendered from volumes V_i . Resulting images I_i are down sampled to and subtracted from lower resolution results. The difference image is the approximated effect of multiple scattering. It is up sampled and added to the final image I_{final}

yields $I_{i'}$. The two images I_i and $I_{i'}$ have the same resolution but are generated from different volumes (Figure 2.12). I_{i+1} includes the effect that photons at resolution V_{i+1} are scattered back from an adjacent voxel. I_{i+1} is therefore a little bit brighter than I_i . Because multiple scattering can only add to observed radiance, the difference image

$$\Delta I_{i^*} = I_{i+1} - I_{i'} \quad (2.17)$$

is clipped to non-negative values. The computed difference image contains an intermediate result between two consecutive resolutions. Starting from the lowest resolution the difference image is up-sampled and combined until the full resolution level is reached.

The final image I_{final} is obtained by adding the accumulated difference image

$$I_{final} = I_0 + \sum_{i^*=0}^{i^*=n} \Delta I_{i^*} \quad (2.18)$$

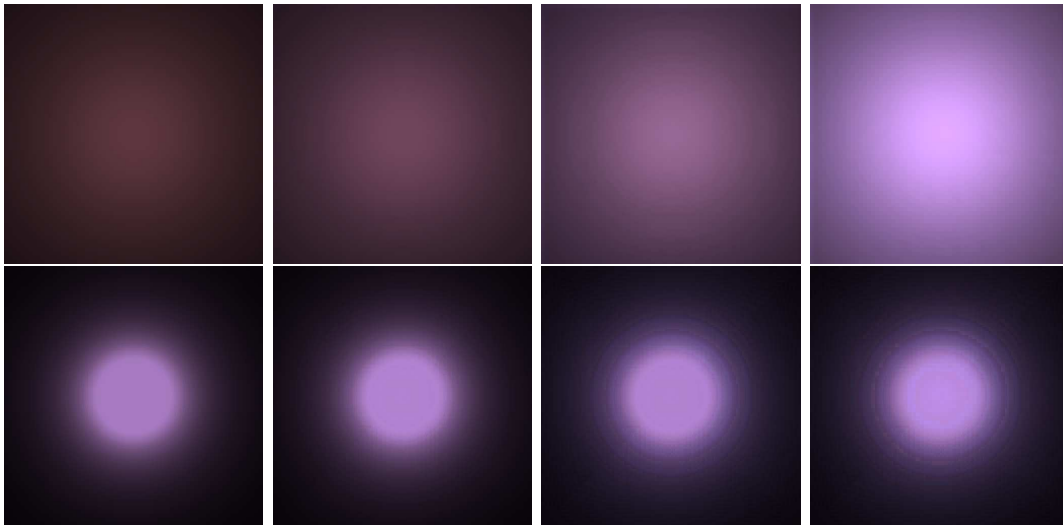


Figure 2.13: Thin slices of dust are rendered in front and behind the star. The effects of our multiple scattering approach are shown. Upper images show the effects if the dust is in front of the illuminating star. Lower images for the back side.

to the image rendered from the full-resolution volume. This image includes the effects of multiple scattering on all scales.

In Figure 2.13 one can see the result for a rendering of a homogeneous layer of dust in front of the star which shows the increasing brightness and a shift to blue because of the increasing scattering probability to shorter wavelengths. The effect of multiple scattering on the back side of the star shows less brightening but a similar blue-shift.

We remark here that this approach assumes that a constant dust density yields the same average scattered radiance as a non-homogeneous distribution of identical average dust density. This is valid for smoothly varying dust concentrations. Even though in regions with high gradients the approach still produces qualitative correct results. However, our method cannot substitute for full radiative transfer simulations in the general case. Still, it turns out that the approach gives reasonable results, as one can see in Figure 2.14.

It is also noticeable that there is a rather large increase in brightness in the resulting image with adding a second scattering 'layer'. This suggests a fairly sparse dust concentration in some regions leading to the fact that lots of photons are

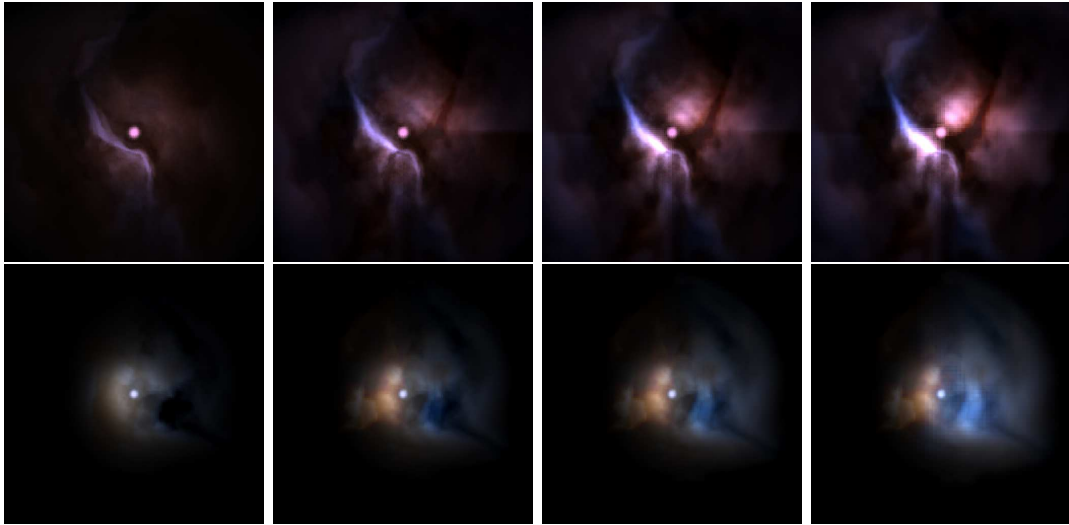


Figure 2.14: Rendering results for the proposed multi-resolution multiple scattering approach. Brightness increases significantly due to global multiple scattering. From left to right one resolution scale is added.

scattered back into the voxel in a second scattering step.

Furthermore, to compensate for resolution problems we suggest to use smoothing filters, such as gaussian blur or more advanced extrapolation algorithms for up-sampling the lower resolution renditions. In the current implementation we haven't solved this problem.

Figure 2.15(a) shows that our multi-resolution approach runs on a nVidia GeForce 6800 Ultra graphics card with a 512x512-pixel resolution and 128^3 , 64^3 , 32^3 and 16^3 volumes with 7.5fps, 6.4fps, 6.2fps and 6.1fps. (with increasing number of resolutions)

To account for global multiple scattering in our implementation is fairly simple. Besides doing the rendering steps for each resolution we need additional Pbuffer or FBOs and simple adding and difference computations in the shader to accomplish the results that are presented here.

2.5 Rendering several stars

So far we only discussed the visualization model for a single star which is located in the center of the volume. However, the proposed visualization system is not

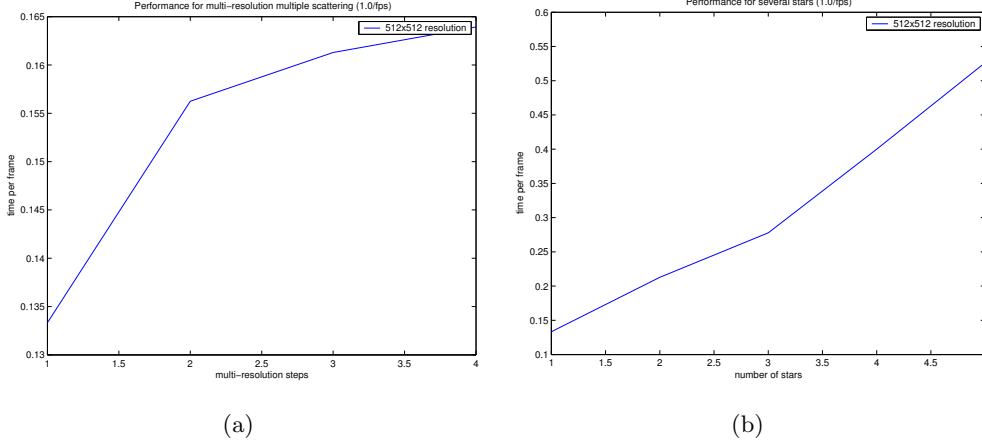


Figure 2.15: (a) Multi-resolution multiple scattering rendering performance. Volume resolutions 128^3 , 64^3 , 32^3 and 16^3 are rendered and combined to a final multiple-scattering result for a 512x512 pixel resolution. (b) A short overview of the performance for 1-5 stars for a 512x512 pixel resolution.

restricted to one star.

Figure 2.16 shows the influence of several stars on one voxel. An important change in our algorithm is the light that is received by a voxel due to optical depth. Since we are generating 3D volume textures including to scattering depth π_{sct} and the received radiance at a voxel L_{ill} , we need to upload the information of how much light arrived at the volume element for every star. In theory it is possible to place an arbitrary number of stars in our model. Nevertheless, due to texture memory constraints we only experimented with up to five stars.

To obtain correct result we just update the shader to lookup how much light is received at a certain voxel for every star, which is just a loop over all stars. The final rendition is just the linear superposition for all given stars for the scattered radiance, which is simply summed before computing the observed radiance. As one can see in Figure 2.15(b), adding more stars in the volume rendering speed decreases linearly due to the additional loops in the shader. The performance drops from 7.5 fps for one star to 4.7, 3.6, 2.5 and 1.9 for up to 5 stars at a resolution of 512x512 pixel on a nVidia GeForce 6800 Ultra graphics card. Figure 2.17 shows rendering results for the proposed method.

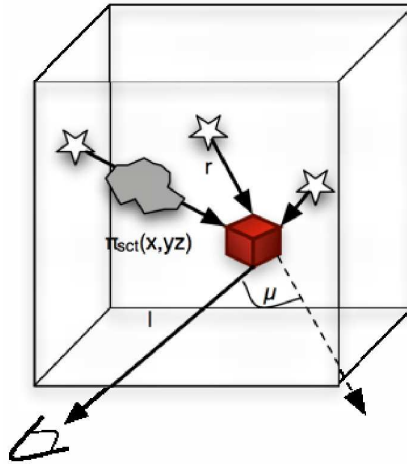


Figure 2.16: Every star has due to the dust density between star and voxel a different influence on the voxel. Therefore the light received at a voxel is store separately for every star when uploaded to the shader.

2.6 Nebula Generation

Our visualization tool can render every kind of three-dimensional dust distribution including an arbitrary number of stars which is essentially constraint by the graphics board texture memory. For educational purposes or planetarium shows an esthetic appearance is eligible. However, in astrophysical settings it might be necessary to reproduce measured experimental brightness distribution of some real nebula. Since we are not trying to reconstruct existing nebulae, we need to generate a 3D dust dataset which is realistic or natural looking or physically plausible.

According to the physical process of star formation we can assume that stellar winds from the illuminating stars have swept clean the immediate surroundings, so that the star is locate in a sphere of low dust concentration. Outside the sphere the dust concentration increases fast and forms optical dense clouds of interstellar dust which are not homogeneous due to hydrodynamic winds. To mimic the effects of clouds and varying dust densities over the volume we use a constraint 3D noise function [53] depending on the distance from the star. The noise modulates the scattering depth $\pi(x, y, z)$, which gives us the possibility to generate a wide range of different-looking yet realistically appealing nebulae. Perlin Noise [53] is widely used in computer graphics to create effects like smoke, fire and clouds. It creates these

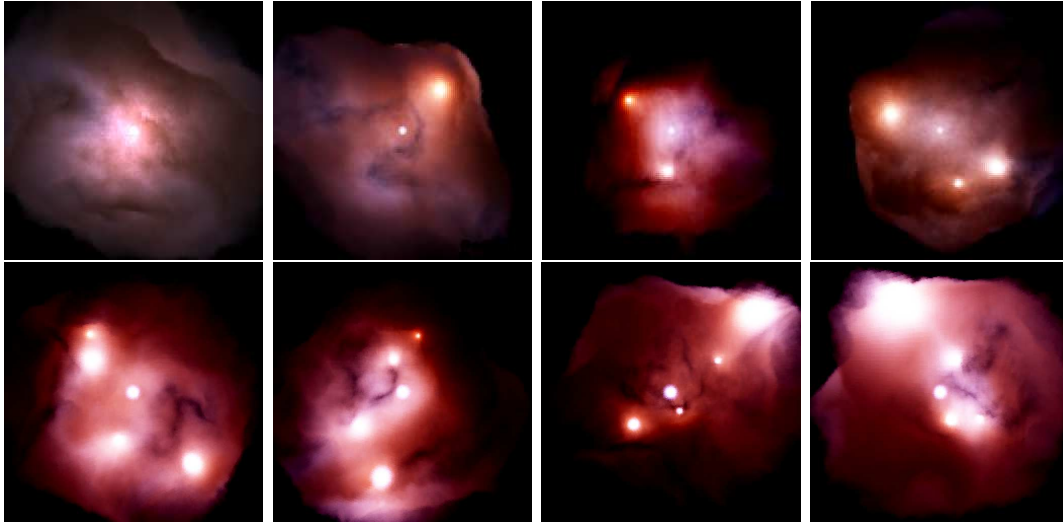


Figure 2.17: From left to right rendering the same dust density for 1-4 stars. Lower row: different views on a nebula with 5 stars.

phenomena with large and small variations simply by adding up noise functions at a range of different scales, as shown in Figure 2.18 for the 2D case. To generate a Perlin Noise we need two things, a noise function and an interpolation function. A noise function is essentially a seeded random noise generator. It takes an integer parameter and returns a random number based on that parameter. If one takes several of these functions, with various frequencies and amplitudes, one can add them all together to create a natural looking noisy function. Figure 2.18 shows several 2D noise functions in the upper row. By combining their amplitudes one obtains the resulting natural looking noise image seen in the second row.

One problem of Perlin Noise is the loss of high frequency noise due to the fact that frequency regions overlap and smooth out. A very recently published work from Cook et al. [11] gives a solution for that problem using Wavelet Noise. However, in the current implementation we rely on Perlin Noise.

2.7 Results and Discussion

We have presented a visualization algorithm for physically correct rendering of reflection nebulae. Figure 2.19 shows some more rendering results for the proposed method. We have outlined the basic astrophysical background and described our vi-

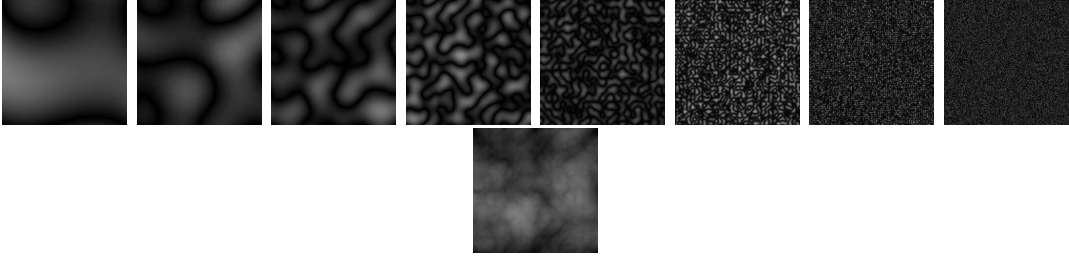


Figure 2.18: Upper row shows smooth 2D noise functions. The intermediate images are scaled in contrast for displaying purposes. The combined 2D Perlin Noise image looks natural, e.g. clouds or dust

sualization model in detail including the implementation in GPU-hardware. We have also shown a novel approach for approximating global multiple scattering effects. However, local per-voxel multiple scattering effects are simulated by a Monte-Carlo method which creates a probability scattering table for an arbitrary single-particle phase function. We propose to use the Henyey-Greenstein phase function, since it is widely accepted in astrophysical research.

Our interactive rendering tool can visualize any synthetic or reconstructed interstellar dust distribution. The user can change the viewing position and direction interactively and adjust several astrophysical parameters. We can create virtual fly-through renderings for interactive desktop visualization and scientifically accurate renderings for educational purposes. Furthermore the tool can be used to visualize and validate astrophysical models for observed or measured data and helps studying effects of dust albedo and scattering anisotropy to understand the optical properties of interstellar dust. It runs on conventional hardware.

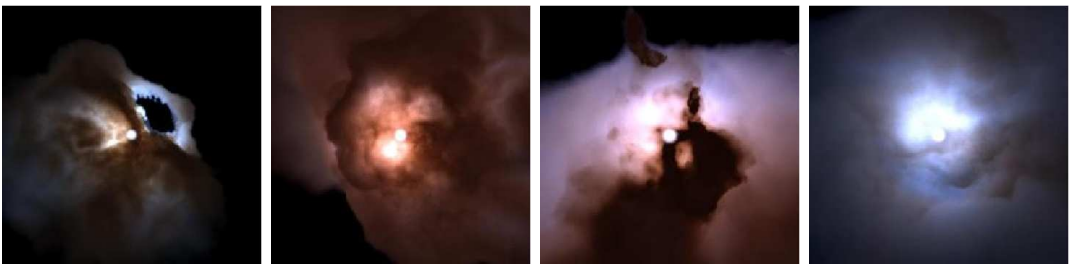


Figure 2.19: Exemplary rendering results of several synthetic datasets including one star in the center of the nebulae

We achieve interactive frame rates of up to 7.5fps for a single star without our

proposed multi-resolution multiple scattering. However, including several stars, Figure 2.15(b) shows that the time spend per frame increases linearly due to bandwidth limitations and caching problems on the GPU-hardware especially after including several stars.

One interpretation would be that the GLSL compiler unrolls the inner loop of up to five stars in the shader code. It seems, that the texture access is coherent so that the fetch for additional stars comes from the cache. The outer loop through the volume can't be unrolled due to it's size (128). Therefore, it dominates the performance linearly, which matches the results one can see in Figure 2.15(b). The rendering performance for the proposed multi-resolution multiple scattering falls off less steep with increasing number of resolutions for a single star. The additional summation and difference operations seem to have little influence on the performance. The several ray-casting passes for one frame seem to dominate the time spend per frame, as Figure 2.15(a).

For future work one could imagine a reconstruction of the 3D shape of a real reflection nebula from calibrated telescope images. We would have to use additional observations in the infrared and emission line images to recover dust and gas. To obtain realistic results one could try to find optimal values for a dust concentration for one pixel by some kind of diffusion process. The biggest problem here is how to determine the high-frequency cloud structure.

3 Reconstruction of Galaxies from Telescope Images

In this part of the thesis we focus on reconstruction techniques to recover the spatial structure of a galaxy from conventional 2D images. The challenge here is to find a volumetric representation of a galaxy given a single viewpoint. To solve the problem we present two approaches. The first method takes similar galaxy images, from galaxies of the same type, into account. The second approach relies on several band-filtered images of one galaxy that provide additional information for the reconstruction process. In Section 3.1 we survey some astrophysical background of galaxies, while explaining our image calibration approach in Section 3.2. After a short overview of our reconstruction techniques in Section 3.3, we go into more detail in Section 3.4, Section 3.5, Section 3.6, and Section 3.7. We conclude and validate our results in Section 3.8.

3.1 Galaxy Background

We want to give an introductory overview of general properties of galaxies that are used in this work. A galaxy is a system of stars, interstellar gas and dust, dark matter in the center and possibly dark energy. Galaxies usually contain 10 million to one trillion stars orbiting a center of gravity. It consists of tenuous interstellar material, star clusters, single stars and various types of nebulae, such as emission-, dark-, planetary- and reflection nebulae. The generic shape in Figure 3.1 can be divided into a center bulge embedding very old stars, a circular disk of younger stars and a surrounding spherical halo [3].

Astronomers classify galaxies based on their overall shape (elliptical, spiral or barred spiral) and further by the specific properties of the individual galaxy, like the number of spiral arms, the degree of the ellipse or the pitch angle of the spiral. The system of galaxy classification is known as the Hubble sequence or Hubble tuning fork which is shown in Figure 3.2. This classification scheme starts at the left with

elliptical galaxies (E0-E6 types) divided by the factor of oval-shape. Then the diagram splits into two branches. The upper branch shows spiral galaxies (Sa-Sc types) which are basically split into different spiral pitch angles. The lower branch (SBa-SBc types) covers barred-spiral galaxies that differ in their characteristic formed bar in contrast to the spherical shaped bulge of Sa-Sc types.

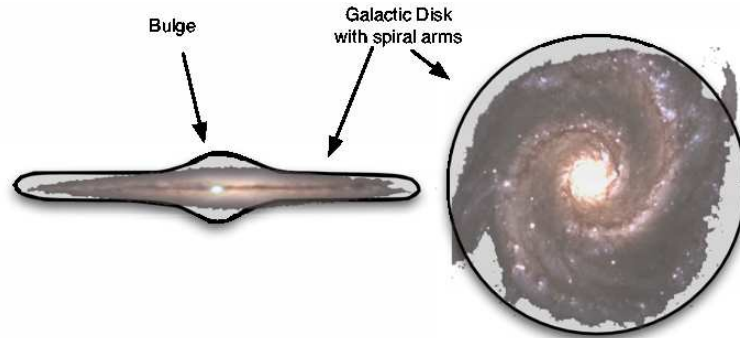


Figure 3.1: Generic Galaxy Shape. On the left is the so-called edge-on view (90°). One can see the dust is concentrated around the central axis. Compared to the circular face-on view on the right, which is often referred as front (0°), where the dust is concentrated along the spiral arms.

In spiral galaxies, the arms have approximately the shape of a logarithmic spiral, which is the result of disturbance in a uniformly rotating mass of stars. The arms are areas of high density or density waves. They can be observed at visible wavelength because the high concentration of gas and dust facilitates star formation of very bright stars (see Section 2). Massive stars and spiral arms usually shine in blue or white light. A galaxy viewed from the front, the so-called face-on view, shows dark spiral stripes containing interstellar dust, dimming and reddening the view as described in detail in Section 2.1.1 [3]. Figure 3.1 also shows a general distribution of dust in the galaxy.

We will focus on recovering galaxies of type Sb and Sc using data from Spitzer telescope [64], the so called Frei Sample [50] and images from the National Optical Astronomy Observatory [52]. One should note here, that the available amount of observed data is overwhelming. Usually the images are stored in FITS file format [49] and one has to calibrate the raw image data using the standard astronomical image processing software IRAF [34]. However, for our reconstruction process we rely on already calibrated RGB images since the IRAF calibration process is non-trivial and

it is not the focus of our work.

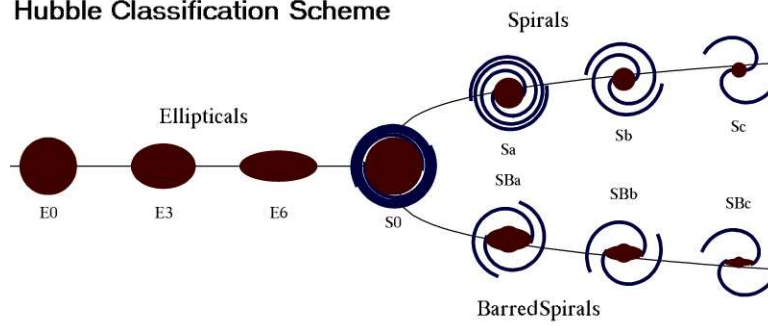


Figure 3.2: Hubble Classification Scheme ©S.D. Cohen (from [9]). Starts from the left with elliptical galaxies as a base. Splits into two branches of spiral galaxies and barred spiral galaxies.

There are other properties of galaxies like surface photometry, luminosity distribution and properties of star forming regions, that show interesting insights about galactic research. However, we do not rely on these properties and do not go into detail here.

Before we start the reconstruction process we have to preprocess the galaxy images to correct for intensity, centroid and rotation in order to work with different image data in a convenient and faithful way. Image calibration and normalization is often a necessary procedure in computer vision problems to analyze the data and to find additional information. We need this process to find the inclination angle of a galaxy which describes the tilt of the galaxy disk towards earth.

3.2 Image Calibration and Normalization

A good way to normalize images in intensity, find the centroid shape of the image as well as the major principal axis, is *geometric moments*. A set of moments computed from an image, generally represents global characteristics of the image shape, and provides a lot of information about different types of geometrical features of the image [46]. A general definition of the moment functions Φ_{pq} of order $(p + q)$ of the image function $I(x, y)$ is given as follows

$$\Phi_{pq} = \int \int M_{pq} I(x, y) dx dy \quad (3.1)$$

where M describes the moment of order $(p + q)$, with $M_{pq} = x^p \cdot y^q$. In order to use several images to analyze the generic shape, we have to normalize the images to the same orientation and overall intensity.

The normalization of an image using *geometric moments* involves transforming it to a standard image having the following properties.

- $\Phi_{00} = \beta$
The total area is β and the intensity values are 1.
- $\Phi_{10} = \Phi_{01} = 0$
The centroid of the shape of the image is at the origin.
- $\Phi_{20} \geq \Phi_{02}$
Major principal axis is the x-axis.

The first two conditions introduce scale and translation normalization. The second order moments yield rotation normalization. Figure 3.3 illustrates the image normalization process. Image (a) is the original data from [50] of galaxy NGC 2683. Image (b) shows the major and minor axis derived from the second order geometric moment and figure (c) exhibits the final corrected and centered grey scale image.

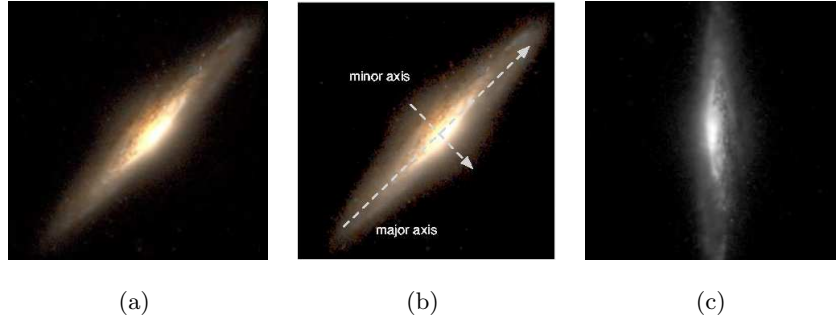


Figure 3.3: (a) Galaxy NGC 2683 (b) Major and minor axis in the galaxy image give us the galaxies inclination. (c) Normalized image computed using geometric moments up to the second order and center and rotate the image. The resulting image shows the galaxy at its defined inclination angle, which is close to edge-on view for NGC 2683.

Observational research has shown that spiral galaxies have a circular shape when seen face-on, as the right image in Figure 3.1. The inclination angle θ of a galaxy can

be described as a tilt of the disk towards earth. The angle of the shift can be easily computed by assuming that spiral galaxies are round when seen face-on, which is widely accepted in astrophysical research, and a orthographic projection is given.

To find the galaxy's orientation with respect to observer's line of sight one can compute (3.2):

$$\theta = \arccos \sqrt{\frac{q^2 - q_0^2}{1.0 - q_0^2}} \quad (3.2)$$

where θ is the inclination and $q = [0, 1]$ is the ratio between the two major axis of an ellipse that can be fitted onto the galaxy. q_0 is commonly accepted as 0.2, which describes the general thickness of the galaxy. In other words, if a galaxy's disk appears more than five times longer than it is wide, we assume that the galaxy's inclination is 90° . Most astronomers adopt a correction factor of 3° that takes into account inaccuracy in measuring a galaxy's axial ratio, that can be discovered by using radio mapping techniques [3]. That means, we have to add the 3° to our computed angle to derive the correct inclination.

For the proposed reconstruction methods it is necessary do the normalization procedure with all images before we are using them in the process. For the second approach we also 'de-project' the galaxy image to a face-on view. Deprojection is commonly referred to the process of rotating the galaxy image as vertical as possible and then stretching it horizontally to approximate a face-on orientation. The assumption of an orthographic projection makes it valid to perform this affine transformation. Figure 3.13 shows results of the image normalization and de-projection on the galaxy M81.

3.3 Reconstruction from Images to 3D Volume

From our terrestrially confined viewpoint, recovering the actual three-dimensional shape of distant astronomical objects is, in general very challenging. In the next sections we want to present our approaches to determine the spatial structure for one class of astronomical objects, i.e. spiral galaxies, from a conventional 2D image alone. For the so-called disk galaxies (see Figure 3.2) we can make reasonable assumptions to obtain a volumetric model from an image. First, we assume a generic shape of the disk galaxy, shown in Figure 3.1. Second, we infer additional information by image analysis techniques, as presented in Section 3.2.

Lets summarize the problem in the following way. The general question is, how to adjust a representative volumetric model by projecting the volume into an image plane, i.e. using volume rendering, trying to match it with an original galaxy image as closely as possible. However, this a hard problem and there are only very few methods that can be used for this approach. Imagine a 128^3 sized volume of the galaxy, which expands to 2.097.152 unknown voxel values. In contrast, standard telescope images are no larger than a few hundred pixels along each dimension. After preprocessing we have an 128^2 image to match the volume size and that provides only 16.384 pixel values for the reconstruction. That is a strongly under-determined problem, which cannot be expected to be solved correctly.

One way of reconstructing a galaxy model is to use computerized tomography. The difficulty we encounter here is that ~ 100 views from different sides of the object are necessary to recover the actual shape, which is the constraint for computerized tomography methods (see Section 3.5). There are general image based methods to reconstruct natural phenomena such as fire, as for example, presented by Ihrke et al. [32], which rely on a sparse set of viewing directions. However, they depend on image data from different views of one object. Our approach, on the other hand, tries to mimic different views with similar looking but different oriented galaxy images to constrain the tomography problem. Therefore we need many views to restrict our solution space.

One can artificially generate a series of images around the galaxy by applying an image morphing technique and warping between several galaxy images. Therefore, the images are normalized beforehand and sorted by their inclination angle. We choose manually similar looking galaxies, that are tilted at increasing inclination angles and morph between them to generate images around the galaxy every 1° from face-on to edge-on view. Additionally one can generated a face-on views by de-projection. The problem one encounters here is that the tomography results are often diffuse due to the reconstruction method and the inconsistent data. We compare two different tomography techniques, Algebraic Reconstruction Tomography (ART) and Filtered-Backprojection (FBP).

Another general approach is to back-project the image and fit it to a generic shape (Figure 3.1) by additionally using noise for disturbance to give it a more natural look. The problem is that back-projecting an RGB image and rendering it as an emissive volume might look convincing from one viewpoint but wouldn't look realistic at all viewed from the side because the emissive volume rendering doesn't

account for extinction effects caused by dust.

The appearance of galaxies as we see them can be described by similar effect as we have seen for *reflection nebulae*. There is illuminating star light emission, extinction and scattering by dust, as can be seen in Figure 3.4 from [3].

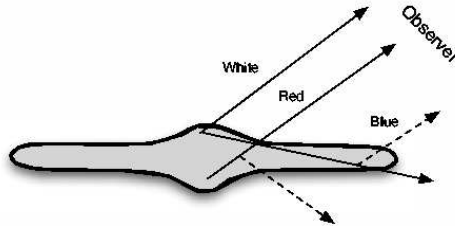


Figure 3.4: Effects of scattering and absorption of light by dust. Light from the top reaches the observer without obstruction. Light from the lower part is partially absorbed and scattered into the path of the observer.

Using the *reflection nebulae* rendering is still an approximation, since the effects of scattering and absorption are only evaluated locally for a few stars and not globally for the entire galaxy. However, we can assume a dust density or scattering depth π_{sct} and a radiant power of star light L_{ill} for our galaxy visualization model at a voxel. By reconstructing a dust density map from several band-filtered infrared images it is possible to find a valid representation for an original image. As mentioned in Section 2.1.1 and Section 3.6 mid- and far-infrared data provide necessary information about the dust distribution of the galaxy. We can also expand this approach by using the reconstructed dust density map as an initialization parameter for an analysis-by-synthesis algorithm to approximate the original shape faster and more closely.

3.4 Image Morphing

Our first reconstruction approach relies on *Computerized Tomography (CT)* which is discussed in more detail in Section 3.5. However, in order to perform the reconstruction by tomography we need to generate more corresponding views around the galaxy. For CT we have an under-determined system of linear equations, generating more views constrains the solution space. One way to do this, if there is only one viewpoint but several objects seen from different sides, is morphing between these images and interpreting every morphing step as a new view of the galaxy.

Image morphing is commonly referred to as the animated transition of digital im-

ages from one image to another. It is a powerful tool used in special effects industry, e.g. for Micheal Jackson's famous music video 'Black or White' and in Terminator 2. Morphing is achieved by coupling image warping with color interpolation. Before the development of morphing, image transitions were achieved by cross-dissolves, like linear interpolation (fading from one image to another). However, the results are poor and introduce double-exposure effects and noise in the misaligned regions. A better solution is to transform each image, so that pixels that belong to the same object parts are aligned in a common coordinate system and then use image warping to achieve a smooth transition.

In the morphing process a source image is gradually distorted and faded out, while a target image starts out and is faded in. The first images in the morphing sequence are much like the source image, while the middle sequence gives an average halfway distorted view on both images. The last images belong by the majority to the target one. One can imagine the approach as warping two images to match the same 'shape' and then cross-dissolving the resulting images.

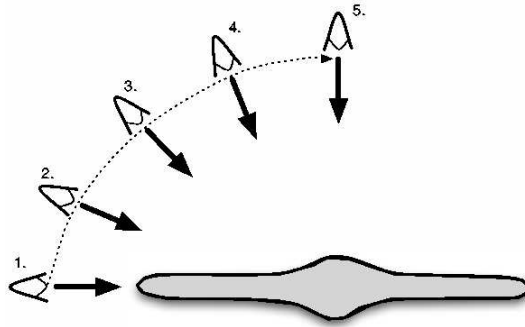


Figure 3.5: Image warping technique. Generating new 'viewing' positions around the galaxy to feed the computerized tomography process. We can create a continuous view from edge-on to face-on galaxies.

For a warping process in general, one needs correspondences between two images. This can be established in two ways: a landmark-based approach or an image-based approach. Landmark-based approaches generally require pairs of corresponding points in source and target images, which are referred to as landmarks and are normally specified manually to register the images. Image-based techniques on the other hand use automatically detected image features to establish the warping.

For simplification purposes we use the landmark-based approach for image registration to manually generate correspondences required for the transformation pro-

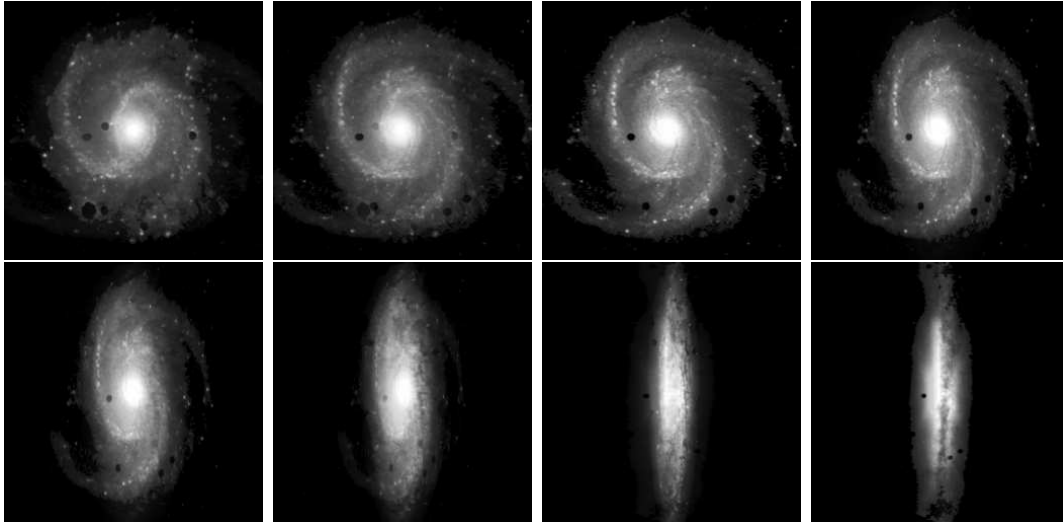


Figure 3.6: From upper left to lower right we morph between four original but normalized galaxy images to generate new plausible views from different inclination angles from a face-on to an edge-on view. First, an original galaxy image with 0° (upper left image) inclination angle, the second original image has 35° (3rd image upper row), followed by 60° (1st image lower row) and 90° (4th image lower row).

cess. The general algorithm for image morphing for an animation series can be summarized in the following steps:

- Identify corresponding points in the source and target image, e.g. center of the galaxy, maximum extent from the center to left, right, top and bottom.
- Estimate a parametric transformation from the source to the target image and the inverse transformation from the target to the source.
- Consider a time parameter $t = [0, 1]$. Warp the source image towards the target image for an amount of t . Warp the target image towards the source image for an amount of $1 - t$.
- Interpolate between aligned images at every time step.

This process generates a series of new plausible views of our galaxy which can be used in our reconstruction methods. One can imagine the process like shown in

Figure 3.5 and Figure 3.6. We choose 3-5 galaxy images that are normalized and sorted by inclination angle and morph between them. Figure 3.6 shows a series from 0° to 90° of warped images from front-view to edge-view created out of four input images: a face-on view onto the galaxy (about 0° inclination angle), two intermediate views (about 35° and 60° inclination angle) and an edge-on view (about 90° inclination angle).

To find the correct geometric transformation from the source to the target images we propose a model called thin plate spline model [6].

3.4.1 Thin-Plate Splines

A thin plate spline is a smooth function $f(x, y)$ which interpolates a surface that has fixed corresponding points p_i , i.e landmarks. Given a set of data points, a weighted combination of thin plate splines centered about each data point gives the interpolation function that passes through the points exactly while minimizing the so-called 'bending energy'. One can imagine the surface as a thin metal plate which bends least in its optimal - bending energy minimized - shape [6]. Thus, we can minimize the bending energy by

$$E[f(x, y)] = \int \int_{R^2} \left(\frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 f}{\partial y^2} \right)^2 dx dy. \quad (3.3)$$

with R^2 describing the image plane. The name 'thin plate spline' refers to a physical analogy involving the bending of a thin sheet of metal, that lies at some distance above a ground plane at the landmarks. In order to apply this idea to the problem of coordinate transformation, one interprets the lifting of the plate as a displacement of the x or y coordinates within the plane. Thus, in general, two thin plate splines are needed to specify a two-dimensional coordinate transformation.

By using two separate thin-plate spline functions f_x and f_y which model the displacement of the landmarks in the x and y direction we obtain a vector-valued function which maps each point of the image into a new point in the image plane:

$$(x, y) \rightarrow (f_x(x, y), f_y(x, y)) \quad (3.4)$$

To create a transformation structure from the source to the target image for a given set of point correspondences we define:

Let $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$, \dots , $P_n = (x_n, y_n)$ be n points in the ordinary Euclidean plane according to a Cartesian coordinate system. Bookstein [6] describes

the steps of the algorithm by first computing the so-called fundamental solution for the biharmonic equation ($\Delta^2 U = 0$) that satisfies the condition of bending energy minimization.

$$U(r) = r^2 \log(r^2) \quad (3.5)$$

where r is the distance between two points. Write $r_{ij} = |P_i - P_j|$ for the distance between points i and j . Then one can define matrices

$$\mathbf{K} = \begin{bmatrix} 0 & U(r_{12}) & \dots & U(r_{1n}) \\ U(r_{21}) & 0 & \dots & U(r_{2n}) \\ \dots & \dots & \dots & \dots \\ U(r_{n1}) & U(r_{n2}) & \dots & 0 \end{bmatrix}, n \times n$$

$$\mathbf{P} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ \dots & \dots & \dots \\ 1 & x_n & y_n \end{bmatrix}, 3 \times n$$

and

$$\mathbf{L} = \begin{bmatrix} K & P \\ P^T & O \end{bmatrix}, (n+3) \times (n+3)$$

where T is the transpose operator and O is a 3×3 zero matrix. Furthermore one defines $Y = (V|000)^T$, a column vector of length $n+3$, where

$$\mathbf{V} = \begin{bmatrix} x'_1 & x'_2 & \dots & x'_n \\ y'_1 & y'_2 & \dots & y'_n \end{bmatrix}$$

are the landmarks in the target image, i.e. which correspond to the points (x_i, y_i) in the source image. This leads to $W = (w_1, \dots, w_n)$ where $W = L^{-1}Y$ or $L^{-1}Y = (W|a_1 a_x a_y)^T$. Use the elements of $L^{-1}Y$ to define a thin-plate spline interpolation function $f(x, y)$ in the plane

$$f(x, y) = a_1 + a_x x + a_y y + \sum_{i=1}^n w_i U(|P_i - (x, y)|). \quad (3.6)$$

The interpolation function is evaluated for every pixel to compute the transformed pixel position at every time step of the image morphing. The transformation is then

applied to every pixel. We use linear interpolation to fade between the two images I_{step0} and I_{step1} .

$$I_{res} = (1 - t)I_{step0} + tI_{step1} \quad (3.7)$$

A result of the morphing process is illustrated in Figure 3.6. One can see that morphing between four different but similar galaxy images gives a plausible view of the galaxy for all side from face-on to edge-on view. That way, we artificially created a set of new plausible views on a galaxy we want to reconstruct.

3.5 Computerized Tomography

Basis of the reconstruction process is Computerized Tomography and described in detail in Kak et al. [37]. The basic problem of tomography, which can be simplified to a one- or two-dimensional problem, is: Given a set of 1D projections taken at different angles μ , how can we reconstruct the 2D image from which these projections were taken. Figure 3.7 shows the basic principle of these projections. We define $P(\mu, d)$ as a 1D projection at an angle μ . P is the line integral of the image intensity $f(x, y)$ along the line l that is at distance d from the origin and at angle μ off the axis.

$$P(\mu, d) = \int_l f(x, y) dl \quad (3.8)$$

All points on this line satisfy the equation $x \sin(\mu) - y \cos(\mu) = d$. Therefore we can rewrite the projection function $P(\mu, d)$ as

$$P(\mu, d) = \int \int f(x, y) \delta(x \sin \mu - y \cos \mu - d) dx dy \quad (3.9)$$

where $\delta(t)$ is the Dirac delta function. The obtained pixel intensities can be interpreted as the mass along the corresponding rays between the source of radiation and the detector. One can express tomographic imaging of P at all μ by a parallel projection of the image $f(x, y)$.

Applications for computerized tomography are typically in the medical imaging and security industry. Lately, modified computerized tomography methods have been used to reconstruct natural phenomena like fire and smoke [32], [29], [33]. They use sparse-view computerized tomography which is applicable to static camera setups. Using Algebraic Reconstruction Tomography (see Section 3.5.1) it is possible

to restrict the solution space to obtain a high-quality volumetric model. Other applied tomographic methods have been investigated to faithfully construct the 3D shape of solid and transparent objects in [24], [5]. A general drawback of these methods is that typically projections for many directions are needed to achieve meaningful reconstruction results, since the process is very sensitive to noise.

One important assumption we have to make for the tomographic reconstruction process is an orthographic projection which is valid due to the distance between the earth and the galaxy object. To apply the proposed reconstruction method to our images, we also have to make sure, that the image formation model is, under some assumption, valid. These assumptions are: the galaxy is an emission volume and scattering takes place only in a uniform manner [33].

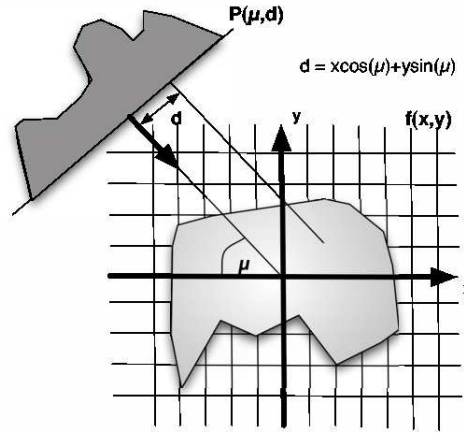


Figure 3.7: Image intensity $f(x, y)$ and the 1D projection $P(\mu, d)$ at a given angle μ .

3.5.1 Algebraic Reconstruction Tomography

One reconstruction method for computerized tomography reformulates the problem in an algebraic framework. Algebraic reconstruction tomography (ART) assumes the cross-sections of an scanned object consists of an array of unknown mass densities. Each ray imposes a new linear constraint relating an image intensity with a weighted sum of grid elements [37]. Figure 3.7 shows a square grid of the image $f(x, y)$. Let the value f_j for each j th cell be constant under the total number of N cells. Imagine now a ray travelling through the medium in the (x, y) -plane. The integral line over the ray gives an emission intensity at a photo receiver collecting all the light over

the ray as a 1D intensity measure. This 1D projection is generated for several consecutive viewing directions.

The relationship between the cell values f_j and p_i can be expressed as

$$\sum_{j=1}^N w_{ij} f_j = p_i \quad i = 1, 2, \dots, M \quad (3.10)$$

where M is the total number of rays in all projections and w_{ij} is the weighting factor that contributes to every j th cell and i th ray integral. One can interpret the weighting factor w_{ij} as a binary factor to express if the i th ray is contributing to the j th cell at all. Note that a lot of w_{ij} are zero since they are not contributing to any ray-sum. This leads to a system of linear equations which is usually very big, but sparse. It contains M by N entries, where M is the number of projections times the number of rays and N describes the number of pixels. For large M and N exist iterative methods for solving the system of equations. This approach was first proposed by Kaczmarz [35]. For pixel $i = 1, \dots, N$ do

$$f_i = f_{i-1} - \frac{f_{i-1} w_i - p_i}{\sum_{i=1}^N w_i^2} w_i \quad (3.11)$$

where f_i is the new pixel value, f_{i-1} is the old pixel value, $f_{i-1} w_i - p_i$ is the difference between the result and back-projected measurement which is normalized by the denominator and weighted by w_i . The iterative algorithm can be summarized by the following steps seen in Algorithm 3.5.1.

Algorithm 1 Algebraic Reconstruction Tomography

```

Make initial guess
while Convergence not reached do
  for all Projection  $p_i$  do
    for all Ray in projection  $p_i$  do
      Compute back-projection
      Compute difference to measured projection
      Distribute difference over projection
    end for
  end for
end while

```

As pointed out in [26], [29] and [1], algebraic methods are inefficient and lack of accuracy of filtered back-projection, but they are more capable of handling noisy and

sparse data. Algebraic methods have also been useful handling more complicated detector geometries that involve non-parallel or curved rays [37]. We implemented ART for testing reasons using a 2D image of the galaxy NGC4321, because it contains high-frequencies noise, which is interesting for reconstruction purposes. However, the results for our test images are not as convincing as compared to the filtered-backprojection as one can see in Figure 3.8. The latter approach gives less noise in the reconstruction and is more accurate for high-frequencies.

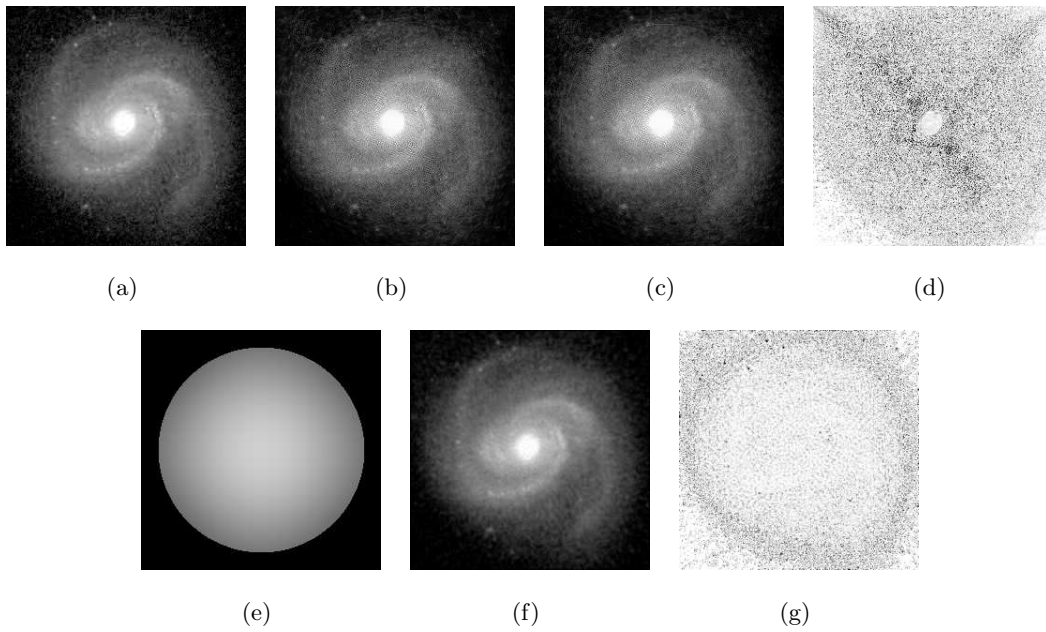


Figure 3.8: (a) Original NGC4321 image. (b) Algebraic Reconstruction Tomography using 90 projections. (c) Algebraic Reconstruction Tomography using 90 projections with circular initial guess (e). (d) Difference between ART and Original (f) Filtered Backprojection from 90 projections. (g) Difference between Filtered Backprojection to Original

Figure 3.8 shows an original example image (a) on the left. Using ART we reconstructed images (b) and (c) using 90 1D-projections obtained by 'scanning' the image and 10 steps for the iterative method (until the difference of reconstructed images between iteration steps was marginal). Image (c) uses an initial guess from image (e). The difference images (d) (for the ART method) and (g) (for filtered back-projection) show our results are close to the original, but ART is (image (b))

not as accurate as the filter-backprojection method in image (f). That is the reason why we rely on the latter method for the 3D galaxy reconstruction, which is discussed in the following section.

3.5.2 Filtered Back-projection

In order to reconstruct the galaxy images we use the general idea of filtered back-projection (FBP). A number of projections of an object are measured. By running the projections back into the image one can obtain a rough approximation of the original image. The back-projection causes blurring artifacts which can be reduced by using filters, like high-pass- or ramp-filters. The basic mathematical concept relies on the Fourier Slice theorem that gives a simple relationship in the Fourier domain between an object and its projection.

The slice theorem basically shows that the 1D Fourier transform of the projection function $P(\mu, d)$ (Figure 3.7) is equal to the 2D Fourier transform of the image evaluated on the line that the projection was taken from. That basically tells us how the 2D Fourier Transform of the image looks like, i.e on certain lines. This line will pass through the origin and lie in the direction perpendicular to the direction of the projection.

We can show the Fourier Slice Theorem in the following way: The 1D Fourier Transform is given by

$$F(\mu, \omega) = \int e^{-j\omega d} P(\mu, d) dd \quad (3.12)$$

When we substitute our expression for $P(\mu, d)$ into (3.12)

$$F(\mu, \omega) = \int \int \int f(x, y) e^{-j\omega d} \delta(x \sin \mu - y \cos \mu - d) dx dy \quad (3.13)$$

we can use the selecting property of the Dirac delta function to simplify

$$F(\mu, \omega) = \int \int f(x, y) e^{-j\omega(x \sin \mu - y \cos \mu)} dx dy \quad (3.14)$$

If we recall the definition of the 2D Fourier Transform of f

$$F(u, v) = \int \int f(x, y) e^{-j(ux+vy)} dx dy \quad (3.15)$$

one can see that (3.14) is just $F(u, v)$ evaluated at $u = \omega \sin(\mu)$ and $v = -\omega \cos(\mu)$, which is the line that the projection $P(\mu, d)$ was taken on.

By interpolating this information and afterwards taking the 2D inverse Fourier Transform we can recover the original image. In the case of continuous images and an unlimited number of views, the Fourier slice theorem can, in theory, be applied directly to obtain a perfect reconstruction.

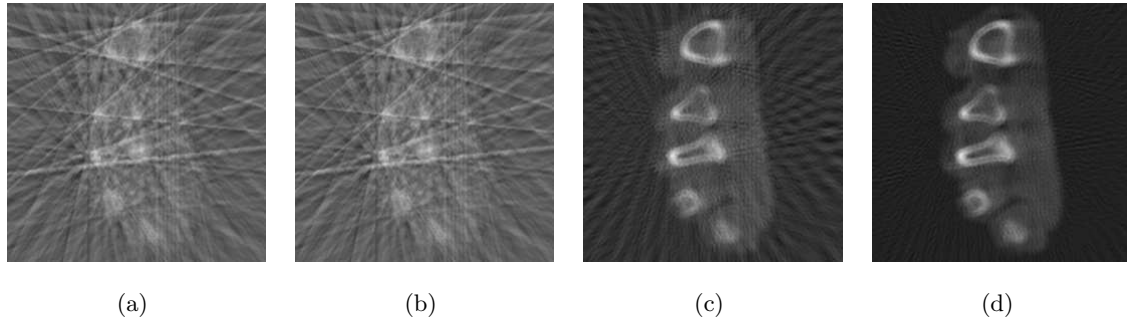


Figure 3.9: Filtered Back-projection for different numbers of views for a publicly available volume dataset from [66]. We reconstruct one slice in the middle of the data set. (a)-(d) filtered backprojection with 15, 30, 60, and 120 views.

Figure 3.9 illustrates results for a filtered-backprojection algorithm using MATLAB's inverse Radon transform with a ramp filter for a different number of projections. It is strongly noticeable that with increasing number of projections the quality of the results improves.

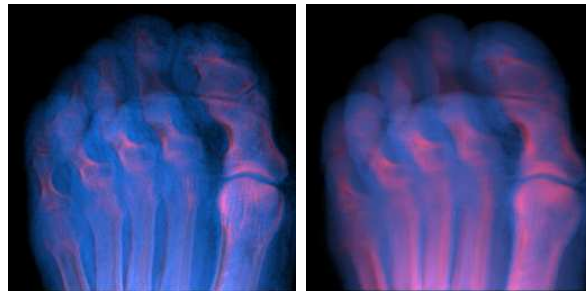


Figure 3.10: Left: Original volume dataset used to generate arbitrary views for reconstruction. Right: Reconstructed volume from snapshots of the original dataset. One can see that high frequency is lost in the reconstruction. Both images are rendered using the same transfer function.

Figure 3.10 illustrates the used filtered-backprojection method on a publicly avail-

able volume dataset from [66]. On the left side one can see the original volume dataset rendered using ray-casting and an arbitrary color mapping transfer function. By volume rendering the data set from different viewing directions (about 90 different views) and applying the inverse radon transform one can recover the three-dimensional shape as seen on the right. We can use the same approach for our galaxy images and their warped intermediate views. The results are shown in Figure 3.11

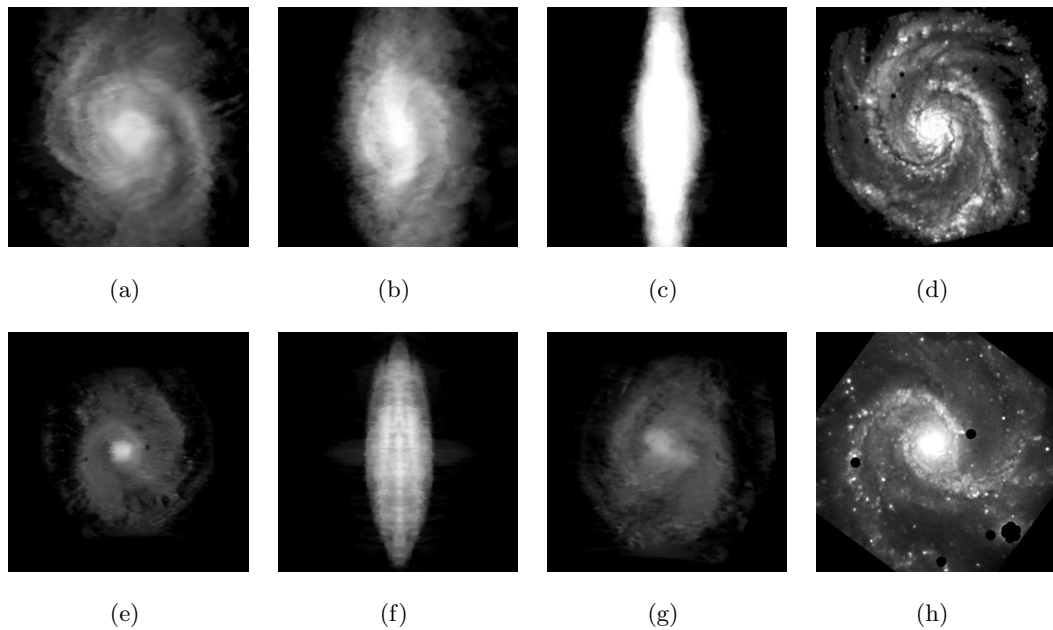


Figure 3.11: Reconstruction results from filtered-backprojection method. Disregard noise by thresholding. (a)-(c) Reconstructed galaxy using CT. (d) Galaxy M51. (e)-(g) Another reconstruction example. (h) Galaxy NGC3184.

Figure 3.11(d) shows one of the original images, galaxy M51 and its reconstructed volumetric model from different sides (a)-(c). Figure 3.11(h) illustrates galaxy NGC3184 and its reconstruction results from several viewpoints (e)-(g). It is noticeable, that we can recover the galaxy's shape really well. However, the reconstruction lacks detail and contains strong amounts of noise. Furthermore, the high-frequencies in the image are completely blurred out. The reason for the rather diffuse results is obviously due to the inconsistency between the different morphed views of the

galaxy that are used for the reconstruction. Another problem with this approach is the visualization for the galaxy. In order to retrieve physical plausible results we have to take a physically plausible visualization into account to represent the high amount of dust in the galaxy that forms their overall appearance. That actually leads us to a rather different idea of dust distribution reconstruction.

3.6 Dust Distribution Reconstruction

Another way of reconstructing a galaxy volume relies on infrared information. The infrared images were obtained by Spitzer's infrared array camera, a space telescope to obtain images and spectra in infrared at wavelengths between 3 to 180 microns (*micron* = μm), that cannot be detected from Earth [64]. Figure 3.12 shows a three-color composite of invisible light, showing emissions from wavelengths of 3.6 microns (blue), 8 microns (green), and 24.0 microns (red). Images in near-infrared collected at 3.6 micron trace the distribution of older and redder stars and are virtually unaffected by obscuring dust. As one moves to longer wavelengths, the spiral arms become the dominant feature of the galaxy. The 8 micron emission is dominated by infrared light radiated by hot dust that has been heated by nearby luminous stars. Dust in the galaxy is bathed by ultraviolet and visible light from nearby stars. Upon absorbing an ultraviolet or visible-light photon, a dust grain re-emits the energy at longer infrared wavelengths, see Section 2.1.1. The dust particles are composed of silicates (chemically similar to beach sand), carbonaceous grains and polycyclic aromatic hydrocarbons and trace the gas distribution in the galaxy. The dust also provides a reservoir of raw materials for future star formation. The 24-micron image shows emission of warm dust heated by the most luminous young stars. The bright knots show where massive stars are being born. These star forming regions are of great astrophysical interest because they help identifying the conditions and processes of star formation [64].

Before analyzing the images we compute the geometric moments up to the second order and center, rotate and de-project the images. The resulting images in Figure 3.13 show the galaxy de-project to a face-on view. Analyzing the images, we can recover approximately the density of dust by adding up images (c) and (d) in Figure 3.12. Adding up the dust images is essentially like accumulating different sorts of interstellar material, that radiate at different infrared wavelengths because of their size and temperature. The image intensity is then interpreted as a dust density

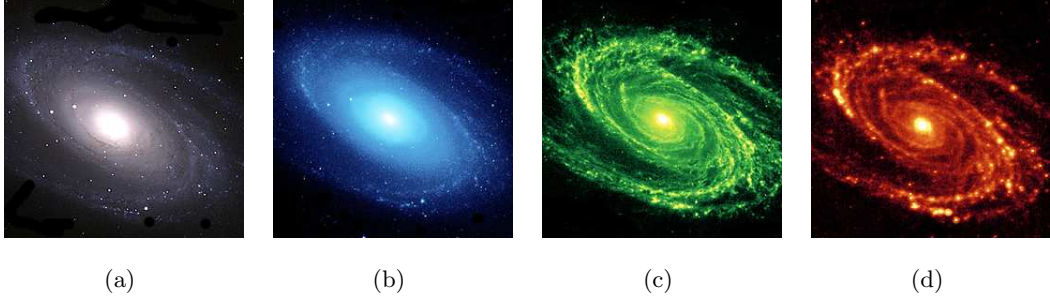


Figure 3.12: M81 as photographed from the Spitzer Space Telescope in the infrared light ©NASA/Spitzer [64]. M81 is located at a distance of 12 million light-years from Earth. (a) Galaxy M81 in visible light. (b), (c) and (d) M81 in 3.6, 8 and 24 micron infrared light.

which is proportional to the scattering depth π_{sct} (see Section 2). The radiance L_{ill} received at a voxel can be taken from Figure 3.12(b), which shows the star light unaffected by obscuring dust. That means, we simplify our visualization model of *reflection nebulae* from Section 2.2 in the following way: The light received at a voxel is the radiance L_{ill} provided by the stars in Figure 3.12(b). Since there are so many stars in the galaxy, it is reasonable to assume that the light is not attenuated by optical depth from the star to the voxel in this large scale structure. However, we still have to take the amount of light into account that gets scattered in the observer's direction. Also we account for the attenuation by optical depth when light travels through the galaxy to the observer as seen in (2.9) and (2.10). Furthermore we use a scattering look-up table for isotropic scattering, because the effects of strong forward- or backward scattering can be neglected in the large scale structure of a galaxy. Nevertheless, this can only be an approximation of the real visualization model of a galaxy in contrast to a physically accurate model of *reflection nebulae* as described in Section 2.2.

L_{ill} and π_{sct} are sufficient to define our volume data structure. We now can try to fit our 2D image in the generic galaxy shape using a simple back-projection approach. By back-projecting the image we smear it through the volume constraint by the generic galaxy shape which can be described by a gaussian function. We discard all values outside the model and weight their contribution according to the distance from the center. However, just back-projecting the dust distribution into the generic shape creates stripes that make the image look unnatural. Using procedural noise

[53] we can break up these stripes in a more natural way. Figure 3.13 shows, that the original image (a) can be implicitly reconstructed by using the dust distribution (b) and a radiance profile in image (c). Image (d) shows the result. It is interesting to see that without any given color, just by using dust and luminosity we can achieve a similar appearance of the galaxy.

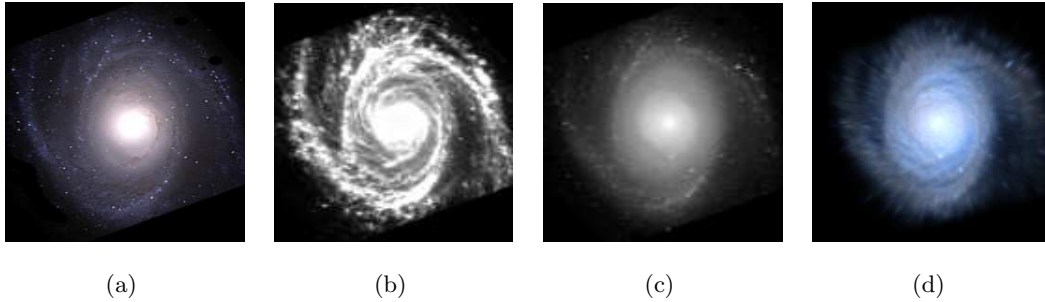


Figure 3.13: Reconstruction results from infrared image data, see Figure 3.12. We can achieve similar rendering results by extracting a dust distribution from the infrared images. (a) Original image in visible light, deprojected to face-on view and centered. (b) Dust distribution. (c) and (d) Resulting renditions.

Obviously this is still a coarse approximation of the real dust density and radiance, but can be used as a first initial guess for an analysis-by-synthesis step as suggested by Magnor et al. [43].

3.7 Analysis by Synthesis Reconstruction

3D image analysis-by-synthesis is the general concept of inverting the image formation process by solving the forward problem repeatedly while adjusting the parameters of the reconstruction until the differences between the original and the synthesized image are minimized. The approach can basically be divided in three steps.

- Starting from a configuration of model parameters we solve the forward problem, i.e. the image formation or rendering, of the scene from a given model.
- Compare the rendition from step one with the original image and measure the error.

- Use an optimization procedure to correct the model parameters and iterate until convergence.

The model restricts the solution to plausible results, but one can enforce complex constraints and exploit a-priori scene knowledge as shown in [14], [44] and [43].

Figure 3.14 shows a small example of the method to illustrate the general capabilities of the procedure. One can see a simple OpenGL rendered scene with static lighting on the left. In this example we reconstruct the light color and position, as well as material properties like shininess of the material for RGB values separately. The series of images illustrates, how the randomly initialized model parameters for light and material $m_{1...N}$ are optimized by iteratively comparing the results of the renderings and minimizing the error functional. We use the sum-of-squared-differences (SSD) to measure the error for the pixel in the original and the reconstructed image which yields

$$\arg \min_{m_{1...N}} \sum (p(x, y) - p_r(x, y))^2 \quad (3.16)$$

where $p(x, y)$ is the intensity of the original pixel and $p_r(x, y)$ is the rendered pixel intensity. The difference image on the right illustrates that the difference between the original image and the optimized reconstruction result is marginal.

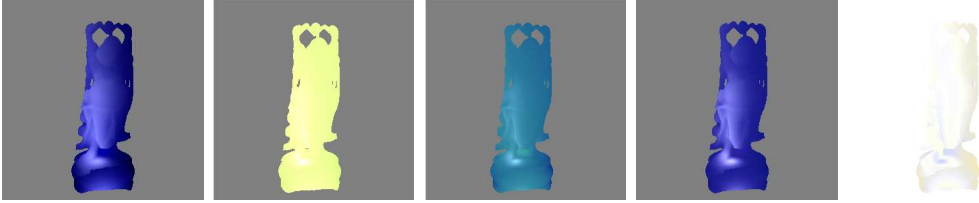


Figure 3.14: Left: Image shows the original rendered scene using one light and OpenGL gouraud shading model. Middle: Reconstructed light color and position in the scene by analyzing the image and minimizing the error between original and model parameters. Right: Difference image between original and reconstruction. Inverted for displaying purposes.

The galaxy reconstruction approach relies on constraint inverse volume rendering (CIVR), Magnor et al. [43]. However, our CIVR approach is based on the generic shape of the galaxy and the *reflection nebulae* rendering, which reduces the 3D volumetric problem to a 2D dust density function, we derived in Section 3.6. The model

we want to minimize is the dust density map which is proportional with the scattering depth π_{sct} . The approximated images can be used as prior knowledge and as an initial guess for the optimization. Additionally, we optimize the overall star color that is a multiplier with the radiance L_{ill} . The *reflection nebulae* rendering provides the basis of our approach, since we assume that it is a plausible way to realistically visualize galaxy volumes. It is important to understand that a rendering which cannot provide plausible renderings, cannot be used for this approach, since we rely on evaluating the error functional based on the rendition and the image difference. As described before, the only difference to the *reflection nebulae* rendering is that we apply a scattering lookup table that was generated for isotropic scattering, due to the fact that strong forward or backward scattering can be neglected for a large-scale structure such as a galaxy. We also simplify the visualization model such that the light received by a voxel as approximated by the radiance L_{ill} in Figure 3.13(c). Given that the *reflection nebulae* rendering is a non-linear process and extinction and scattering effects have influence on other volume elements in a non-linear way, we employ non-linear optimization.

For our optimization procedure we use a standard implementation of Powell's non-linear optimization method [56]. *Powell's direction set* numerically evaluates the error function's local slope along all dimensions of the model $m_{1...N}$ from which it determines the *conjugate-gradient* direction, which is shown in Figure 3.15, where the shortest path from start point x_0 to the global minimum in N -dimensions is illustrated.

Powell's direction set follows the general scheme of successive line minimization, i.e. a minimization in one dimension, but it does not involve explicit computation of the function's gradient. The method produces N *conjugate directions*. Usually one needs derivatives to generate a set of *conjugates*. However, if, due to the problem, no derivatives are available Powell's direction set can be used to generate a *conjugate set*, which is a sequence of line minimizations. It is the method of choice if one has a reasonable starting approximation, cannot easily obtain derivatives and the function isn't too noisy [56].

If we start at a point x_0 in N -dimensional space, and proceed from there in some vector direction \mathbf{u} to find the minimum, we can use line minimization. However, we want to perform a line minimization in direction \mathbf{u} and then choose a new direction \mathbf{v} so that minimizing along that new direction \mathbf{v} will not 'spoil' the minimization along the former direction \mathbf{u} . In order not to interfere with our \mathbf{u} minimization we

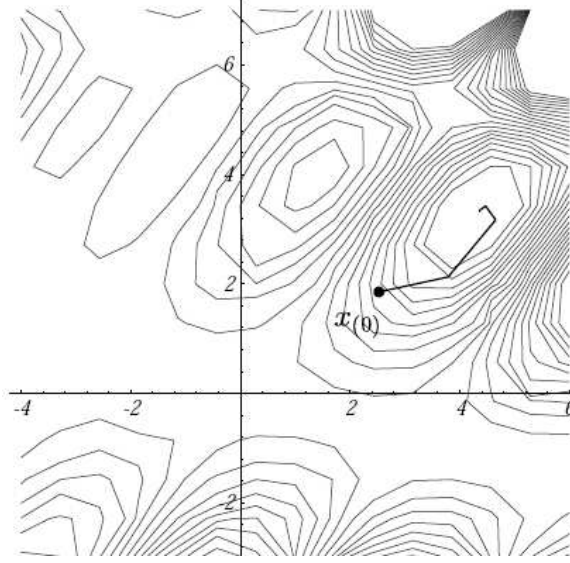


Figure 3.15: Successive minimizations along coordinate directions. Powell’s direction set evaluates the error function’s local slope along all dimensions and determines the conjugate-gradient direction. Image taken from [27]

require that the gradient remains perpendicular to \mathbf{u} . If two vectors \mathbf{u} and \mathbf{v} have this property, they are called *conjugate*. A set of vectors for which this is valid for all pairs, is a *conjugate set*. That means, we have a set of n directions in N -dimensional space and compute successive line minimization of a function along a conjugate set of directions. That has the advantage that we don’t have to redo those directions and just repeat cycles of N line minimizations to converge to the minimum.

The aim of the optimization is to determine the solution for the 2D projection, i.e. the volume rendering result matches as closely as possible with the original galaxy image at visible wavelengths. Each optimization iteration step entails a modification in the volume data set, uploading the modified data onto the graphics card, rendering the model again and reevaluating the error measure. To qualify the difference between both images we compare the corresponding pixel and the sum-of-squared-differences (SSD) over all pixels (3.17).

$$\arg \min_{d_{1...N}} \sum (p(x, y) - p_r(x, y))^2 \quad (3.17)$$

where $d_{1...N}$ denotes the parameters in the dust density map and the color parameters. Additionally, the error functional penalizes negative values and scattering

depth π_{sct} values that reach outside the scattering table for values $\pi_{sct} > 10.0$. The color values are penalized, if they fall outside the RGB range. That allows us to constrain our optimization to physically reasonable values. Magnor et al. [43] employed several error functionals of which the SSD error measure yielded the fastest convergence. Algorithm 3.7 summarizes the steps again:

Algorithm 2 Analysis-by-Synthesis

```

Back-project volume into generic shape
Render galaxy volume
Initialize optimization parameters, i.e. 2D scattering depth
while Convergence not reached do
    Render volume
    Optimize parameters using Powell's Direction Set
    Compute SSD to evaluate error
    Penalize parameters that are out of range
    Update optimization parameters, i.e. 2D scattering depth  $\sigma_{sct}$  and color values
    Back-project and update 3D volume on the graphics card
end while

```

The algorithm stops, when the difference between optimization steps is lower than a certain tolerance value. From the optimization point of view this approach underlies a high-dimensional parameter space. Each iteration step we modify our parameters until the algorithm converges to a minimum of the error function. Since we are dealing with a non-linear optimization problem, a global convergence to the global minimum can not be guaranteed.

If the initial guess is not close to the global minimum, or the parameters are not reasonable constraint, the algorithm does not converge to a physically plausible solution. Also, if our rendering procedure does not map the values closely to the original projection a convergence cannot be expected, because we are not able to produce the desired values.

Figure 3.16 and Figure 3.17 illustrate the results before and after the *analysis-by-synthesis* step. Figure 3.16 shows that the difference between the original image and the not optimized rendition of the volumetric back-projection in the inner part, near the bulge, of the galaxy is very high. This effect is much reduced in Figure 3.17 after optimization, but also dimmed very much in the bulge, which increases the difference in the center. Also, the spiral arms fade out and their bluish appearance vanishes

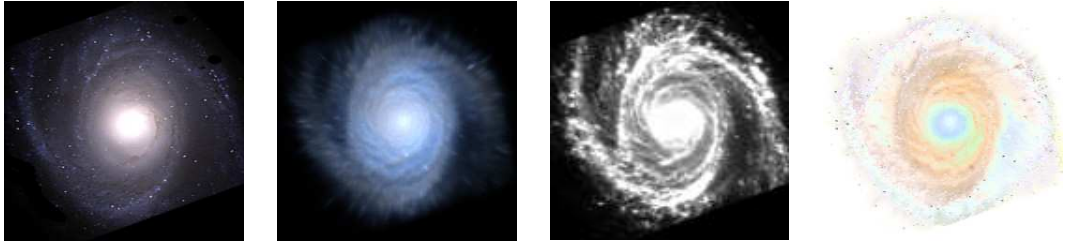


Figure 3.16: Left: Image shows the original M81 image deprojected to face-on view. Middle-Left: Reconstructed rendering result using infrared image data. Before optimization. Middle-Right: Initial dust density image. Right: Difference image between original and reconstruction. Inverted for displaying purposes.

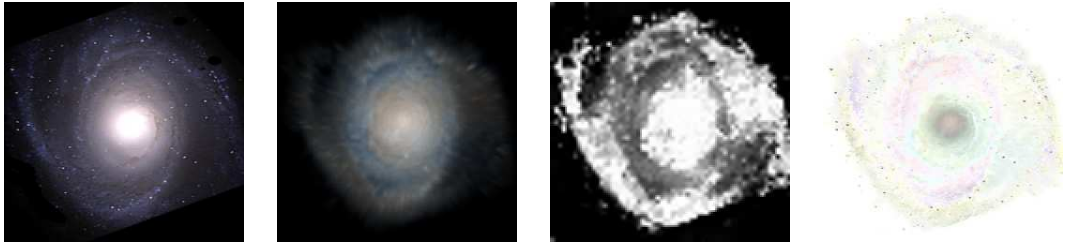


Figure 3.17: Left: Image shows the original M81 image deprojected to face-on view. Middle-Left: M81 after analysis-by-synthesis optimization. Middle-Right: Reconstructed dust density map. Right: Difference image between original and optimized image. Inverted for displaying purposes.

slightly. The dust densities in both figures still show similarities in shape but are very different in detail. One can see that the optimization procedure compensates reddish part in the original image with more dust, which is a reasonable and physical plausible reconstruction.

However, the procedure also shows that there are still difference between the original and the reconstruction. Nevertheless, the presented method seems to be a promising approach, that should undergo further investigation. Optimizing the rendering parameters and taking the radiance map into account during the optimization will be a first step to improve the results even more.

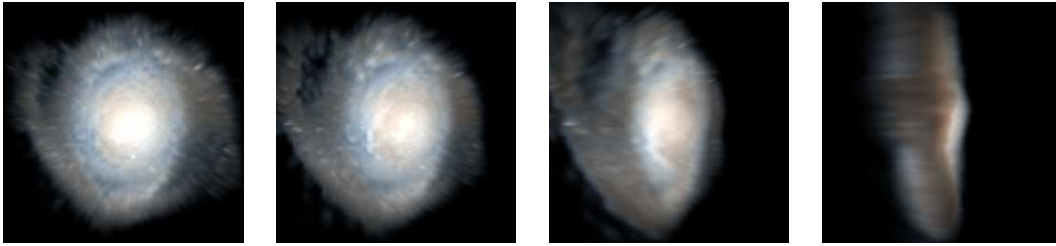


Figure 3.18: Reconstructed volume of the synthesis-by-analysis procedure. Fly-by of galaxy M81.

3.8 Results, Validation and Discussion

The proposed reconstruction methods show different results. The computerized tomography approach is able to reconstruct the general shape of a galaxy, including the different parts such as the bulge and the spirals, without relying on any explicit 3D shape information, like a generic shape model. But the volumes are very diffuse and are barely similar to the original images. Figure 3.11 shows grey-valued reconstruction results. Also, if one wants to reconstruct a galaxy image where the galaxy has an inclination angle towards earth of about $\sim 45^\circ$ the recovered shape is totally diffuse and details vanish. The reason for that is, that the reconstruction process blurs out the features in the structure so much, that one can distinct these features, like the spirals, only viewed face-on. This is also due to the rendering. In general, that approach leads to physically non-plausible results and cannot be used for reconstruction of galaxy images, by using these specific band-filtered information. One could imagine better results by trying to use CT methods on infrared data to recover a dust distribution and use it as an input parameter for the proposed *reflection nebulae* rendering. The problem here is that one depends on the available far-infrared data. The Spitzer Telescope database [64] is due to the capabilities of their telescope the only site to retrieve these observations. Unfortunately, there are currently only a couple data sets publicly available.

However, the presented results for our *Dust Density Reconstruction* are physically more plausible. The reconstructed colors match with the original image and approximate the overall appearance. Furthermore, general properties can be recovered much better then with the tomography technique, such that dust is concentrated along the spiral arms and is noticeable in more detail and the overall appearance matches more closely when rotating around the galaxy, as shown in Figure 3.18.

Even the start configuration before optimization exhibits convincing and physically plausible results. Of course, that is because we rely on the generic shape of our galaxy. That way we introduce 'controlled' noise only to implement a more natural look and to bind the values not strictly to the shape. This is essentially a simple but efficient approach for the reconstruction of volumetric galaxy models from conventional 2D images.

Since this reconstruction problem is non-linear, there is no proof for a general convergence to a physically plausible minimum. To validate the results of the reconstruction derived for the galaxy, we would have to know the actual shape. Unfortunately there does not exist a 3D density structure of a galaxy to generate test images that can be recovered using our approach. Therefore we propose a simple validation test by using synthetic data. Figure 3.19 shows the original rendering in (a). The original volume is constructed by back-projection radiance map (c) and the dust density (d) through an 16^3 volume. We then let our algorithm recover the original image by adjusting the dust density, back-projecting it again and minimizing the error in the resulting rendition. Figure 3.19(b) shows the result of the reconstruction and its difference image in Figure 3.19(f), which shows only slight difference between both. However, the dust density is noisy and not as bright as the original, which suggests that the reconstructed dust density in it's recovered shape is already bright enough to recover the original image and an increase in brightness would not have made a difference for the overall appearance. That results in a slightly different dust density volume with the same shape as the original one, but slightly darker.

One can see that the proposed method recovers an approximated, plausible shape of the volume using *reflection nebula* rendering, which leads to the conclusion, that the proposed method is a promising approach to attack the problem.

However, there are still a lot of problems, mostly related to approximations in the visualization model. The radiance map, as seen in Figure 3.13(c), doesn't have enough influence on the rendering to display the bluish areas along the spiral arms. That should be taken into account during optimization. Furthermore, we should take other physical parameters into account to represent the galaxies appearance more closely.

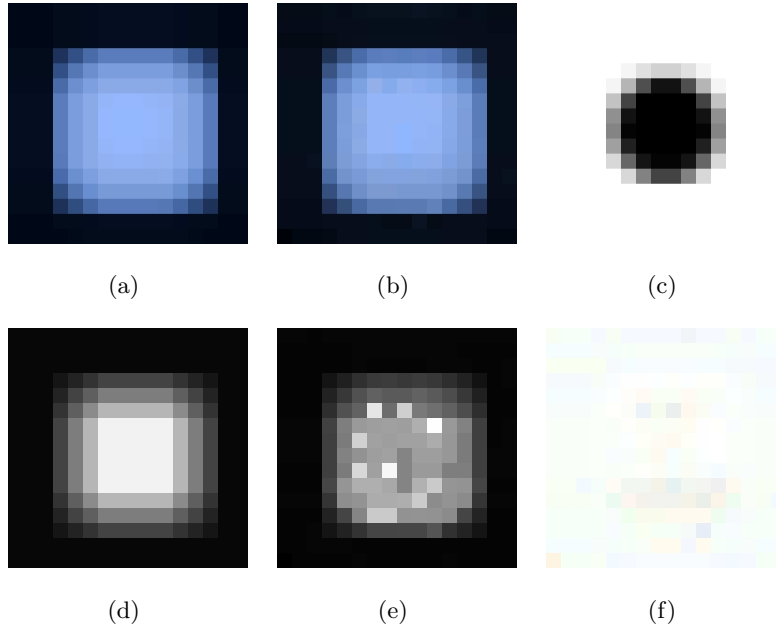


Figure 3.19: (a) Original synthetic 16^3 volume data set. (b) Reconstructed result using the analysis-by-synthesis approach (c) Map that is back-projected through the volume denoting the light received at a voxel. Black is low value, white shows high radiance. (d) Original dust distribution 2D image that is back-projected through the volume. (e) Reconstructed slightly noisy dust distribution. Recovered from the optimization. (f) Difference between reconstruction and original.

4 Summary and Outlook

We have presented a new visualization technique for rendering *Reflection Nebulae* as well as a reconstruction approach for recovering a three-dimensional *Spiral Galaxy* model from conventional 2D images.

Our visualization tool renders physically correct interstellar dust surrounding luminous stars. It allows to interactively vary viewing position as well as to change several physical parameters of the nebula. We have seen how scattering and absorption of light by dust influences the overall appearance of the astronomical object and illustrated how the effects of global multiple-scattering can be approximated with a multi-resolution approach. Figure 4.1 shows some more rendering results for the proposed methods. One can see a star surrounded by synthetically generated clouds of interstellar dust. We have shown how this visualization method can be used for several applications, for example for scientifically accurate fly-through animations for planetarium shows, for augmented telescope applications and for validation of hypothetical derived models from observed data.



Figure 4.1: Rendering results for physically correct reflection nebulae visualization.

We have also presented how we adopt the rendering approach to reconstruct a plausible shape for the spiral galaxy M81. Its inherent generic shape and additional observations in far-infrared enable us to use a model to describe the galaxy's three-dimensional dust distribution in space, thereby constraining the reconstruction

problem. By rendering realistic images of our model and comparing the rendering results to the original image data, we employ an optimization approach that helps converge towards a dust density distribution. It closely resembles the true shape of the galaxy by fitting it to a generic shape. Figure 4.2 shows a series of images from different view points around galaxy M81 recovered from far-infrared image data.



Figure 4.2: Results for the presented dust distribution reconstruction of Spiral Galaxy M81.

We have also used other reconstruction techniques, like computerized tomography, to recover the shape of the galaxy. We pointed out the advantages and disadvantages of these methods, but the results are not convincing and lack in accuracy and blur the data, so that it is not possible to determine a realistic looking volume. There are several reasons for that: Firstly, the available data that is used to generate new views by image morphing techniques is not consistent enough to be able to faithfully reproduce the shape. Secondly, we have seen that it is not possible to match a reconstructed volume with the original 2D image without physically plausible rendering of galaxies. Approaches to use the reflection nebulae rendering gave more reasonable results that match closely with the original image.

Nevertheless, all approaches presented here still lack in accuracy and essentially give an overview of what is possible to reconstruct from the available observational data. There are a lot of ways to improve the results. One way for improvement is to clean the available data for background noise more carefully and use directly observed data in FITS file format for calibration, so that the images exhibit a linear response curve, i.e., twice the number of photons falling on a pixel must double the pixel's value. We can use the standard astronomical image processing software IRAF [34] to automatically mask out regions of no interest, like foreground stars more carefully to avoid bias in our reconstruction. One can think of incorporating

additional astrophysical research data from the galaxy’s morphology and photometry, such as luminosity distributions for single galaxies [16]. Furthermore, there is way of improvement by optimizing the rendering parameters, such as adding more galaxy specific photometry constants in the visualization model or take the radiance map also into account during optimization. To accelerate the analysis-by-synthesis approach it is possible to pursue a multi-resolution approach, which can speed up the convergence of our optimization by several factors, as pointed out in Magnor et al. [43]. One can also imagine to connect the presented reconstruction procedure and use computerized tomography to recover the distribution of dust from far-infrared data. Unfortunately there is currently not enough data publicly available. Of course, to make the rendering of galaxies and reflection nebulae more faithful and interesting for fly-through applications, we need to render stars around and into the volume. Also, incorporating more sophisticated high-dynamic range rendering would make the visualization look more realistic.

Finally, we want to conclude that rendering and reconstruction of astronomical objects, like *Reflection Nebulae* and *Spiral Galaxies* is an interesting field of research, which could not been covered in the whole complexity. However, we hope to see more physically correct rendering and reconstruction approaches of astronomical objects in future to create scientifically accurate fly-through animations and to illustrate the beauty of space.

A Johnson System and Star Types

Table A.1: Star colors of different star types without intervening atmosphere or interstellar medium and their RGB values for visualization [10]. It only shows chromaticity (hue and saturation), but not brightness to be able to display it on computer screen.

Star Type	R	G	B
O	155	176	255
B	170	191	255
A	202	215	255
F	248	247	255
G	255	244	234
K	255	210	161
M	255	204	111

Table A.2: Table of bands and associated wavelengths for the UBVRI Johnson color system. Astronomical images are recorded using these standardized bands. The UBVRI system is one among several different systems that is used in today's astronomy observations.

System	Band	Wavelength λ in nm
UBVRI	U	365
	B	445
	V	551
	R	658
	I	806
	J	1220
	H	1630
	K	2190
	L	3450

Bibliography

- [1] A. Andersen and A. Kak. Simultaneous algebraic reconstruction technique (sart). a superior implmentatoin of the art algorithm. *Ultrasonic imaging*, 6:81–94, 1984.
- [2] C. Barbera, E. Athanassoula, and Garcia-Gomez C. Deprojecting spiral galaxies using fourier analysis. application to the frei sample. *A&A*, 415:849–861, 2004.
- [3] J. Binney and M.Merrifi. Galactic astronomy. *Princeton University Press*, 1998.
- [4] J. Blinn. Voyager fly-by animations. *Animations depicting space missions to Jupiter, Saturn and Uranus*, 1977-87.
- [5] J.S.D. Bonet and P.A. Viola. Roxels: Responsibility weighted 3d volume reconstruction. *Proceedings of International Conference on Computer Vision (ICCV)*, pages 418–425, 1999.
- [6] Fred L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:567–585, 1989.
- [7] B. Cabral, N. Cam, and J.Foran. Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. *Proceedings of ACM Symposium on Volume Visualization*, pages 91–98, 1994.
- [8] J. Cardelli, G. Clayton, and J. Mathis. The relationship between infrared, optical and ultraviolet extinction. *Astrophysical Journal*, pages 245–256, 1989.
- [9] S. D. Cohen. Hubble classification scheme. *website: <http://en.wikipedia.org/wiki/Image:Hubblescheme.png>. visited August 2005*, 2005.
- [10] What colors are the stars? *website: <http://www.vendian.org/mncharity/dir3/starcolor/>. visited April 2005*.

- [11] R.L. Cook and T. DeRoseK. Wavelet noise. *Proceedings of ACM SIGGRAPH*, 24:803–811, 2005.
- [12] W.M. Cornette and J.G. Shanks. Physically reasonable analytic expression for the single-scattering phase function. *Applied Optics*, 32:3052–3060, 1992.
- [13] T. Cullip and U. Neumann. Accelerated volume reconstruction with 3d texture mapping hardware. *Technical Report TR93-027, Department of Computer Science, UNC Chapel Hill*, 1993.
- [14] B. Curless and M. Levoy. A volumetric method for building complex models from range images. *Proceedings of ACM Siggraph*, 30:303–312, 1996.
- [15] B.T. Draine. Scattering by interstellar dust grains. i. optical and ultraviolet. *Astrophysical Journal*, 598:1017–1025, 2003.
- [16] R. Drimmel and D. Spergel. Distribution of dust and stars in the galaxy. *Astrophysical Journal*, 556, 2001.
- [17] K. Engel, M. Kraus, and T. Ertl. High-quality pre-integrated volume rendering using hardware-accelerated pixel shading. *Proceedings of Graphics Hardware*, pages 9–16, 2001.
- [18] Evans and Sutherland Wonders of the Universe planetarium show. *website: http://www.ccsn.nevada.edu/planetarium/Wonders_Trailer.mov. visited August 2005*, 2000.
- [19] Chi-Wing Fu and A. Hanson. The powers-of-ten visualization architecture for exploring the physical universe. *submitted for publication*, 2005.
- [20] NASA/ESA Hubble Heritage Image Gallery. Ngc 1999. *website: <http://heritage.stsci.edu/gallery/galindex.html>. visited June 2005.*, 2005.
- [21] A. Van Gelder and K. Kim. Direct volume rendering with shading via three-dimensional textures. *Proceedings of ACM Symposium on Volume Visualization*, pages 23–ff., 1996.
- [22] J. Genetti. Volume-rendered galactic animations. *Communications of the ACM*, 45:62–66, 2002.

- [23] O. Gerhard and J. Blinney. On the deprojection of axisymmetric bodies. *AstroSoc*, 279:993–1004, 1996.
- [24] D.T. Gering and M.W. Wells III. Object modelling using tomography and photography. *Proceedings of IEEE Workshop on Multiview Modeling and Analysis of Visual Scenes*, pages 11–18, 1999.
- [25] K. Gordon. Interstellar dust scattering properties. *Astrophysics of Dust*, 2004.
- [26] R. Gordon, R. Bender, and G. Herman. Algebraic reconstruction techniques (art) for three-dimensional electron microscopy and x-ray photography. *Journal of theoretical biology*, 29:471–481, 1970.
- [27] Painless Conjugate Gradient. *website: <http://www.cs.cmu.edu/~quake-papers/painless-conjugate-gradient.pdf>. visited August 2005.*
- [28] A.J. Hanson, Chi-Wing Fu, and E.A. Wernert. Very large scale visualization methods for astrophysical data. *Proceedings of the Joint Eurographics and IEEE TVCG Symposium on Visualization*, pages 115–124, 2000.
- [29] Samuel W. Hasinoff. Three-dimensional reconstruction of fire from images. *MSc Thesis, University of Toronto, Department of Computer Science*, 2002.
- [30] J. Henyey and J. Greenstein. The theory of the colors of reflection nebulae. *Astrophysical Journal*, 88:580–604, 1938.
- [31] J. Henyey and J. Greenstein. Diffuse radiation in the galaxy. *Astrophysical Journal*, 93:70–83, 1941.
- [32] Ivo Ihrke and Marcus Magnor. Image-based tomographic reconstruction of flames. *ACM Siggraph / Eurographics Symposium Proceedings, Symposium on Computer Animation*, pages 367–375, 2004.
- [33] Ivo Ihrke and Marcus Magnor. Adaptive Grid Optical Tomography. *IMA Vision, Video, and Graphics (VVG'05)*, pages 141–148, 2005.
- [34] IRAF. *website: <http://iraf.tuc.noao.edu>. visited February 2005.*
- [35] S. Kaczmarz. Angenaehrte aufloesung von systemen linearer gleichungen. *Bull. Acad. Pol. Sci. Lett. A*, 6-8A:355–357, 1937.

- [36] J. Kajiya and B. von Herzen. Ray tracing volume densities. *Proceedings ACM SIGGRAPH*, pages 165–174, 1984.
- [37] A.C. Kak and M. Stanley. Principles of computerized tomographic imaging. *Society of Industrial and Applied Mathematics*, 2001.
- [38] J.B. Kaler. Kosmische wolken. *Materie-Kreislufe in der Milchstrasse*, Spektrum Akademischer Verlag, 1998.
- [39] J. Krueger and R. Westermann. Acceleration technique for gpu-based volume rendering. *Proceedings of IEEE Visualization*, pages 287–292, 2003.
- [40] Philippe Lacroute and Marc Levoy. Fast volume rendering using a shear-warp factorization of the viewing transformation. *Proceedings of ACM SIGGRAPH*, pages 451–458, 1994.
- [41] A. Lințu and M. Magnor. Augmented astronomical telescope. *Second GI-Workshop VR/AR Proceedings*, 2005.
- [42] M. Magnor, K. Hildebrand, A. Lințu, and A. Hanson. Reflection nebula visualization. *Proceedings of IEEE Visualization*, 2005.
- [43] M. Magnor, G. Kindlmann, and H. Hansen. Constrained inverse volume rendering for planetary nebulae. *Proceedings of IEEE Visualization*, 2004.
- [44] S. Marschner. Inverse rendering for computer graphics. *Ph.D. dissertation, Cornell University*, 1998.
- [45] N. Max. Efficient light propagation for multiple anisotropic volume scattering. *Proceedings of ACM Eurographics Symposium on Rendering*, pages 87–104, 1994.
- [46] R. Mukudan and K.R. Ramakrishnan. Moment functions on image analysis. *World Scientific*, 1998.
- [47] D. Nadeau and E. Engquist. Volume visualization of the evolution of an emission nebula. website: <http://vis.sdsc.edu/research/hayden2.html>. visited January 2005., 2002.
- [48] D. Nadeau, J. Genetti, S. Napear, B. Pailthorpe, C. Emmart, E. Wesselak, and D. Davidson. Visualizing stars and emission nebulae. *Computer Graphics Forum*, 20:27–33, 2001.

- [49] NASA/GSFC. *Flexible image transport system (FITS)*, website: <http://fits.gsfc.nasa.gov>. visited April 2005.
- [50] Frei Bright Galaxy Catalog 113 nearby galaxies. website: <http://www.astro.princeton.edu/~frei/catalog.htm>. visited April 2005.
- [51] T. Nishita, Y. Dobashi, and E. Nakamae. Display of clouds taking into account multiple anisotropic scattering and sky light. *Proceedings of ACM SIGGRAPH*, pages 379–386, 1996.
- [52] National Optical Astronomy Observatory. Noao/aura/nsf. website: <http://www.noao.edu>. visited June 2005.
- [53] K. Perlin. An image synthesizer. *Proceedings of ACM SIGGRAPH*, pages 287–296, 1985.
- [54] M. Pharr and R. Fernando. *Gpu-gems2*. Addison Wesley, 2004.
- [55] S. Premoze, M. Ashikhmin, and P. Shirley. Path integration for light transport in volumes. *Proceedings of ACM Eurographics Symposium on Rendering*, pages 52–63, 2003.
- [56] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. Numerical recipes in c. *Cambridge University Press*, 1992.
- [57] I. Puerari, D.L. Block, B.G. Elmegreen, J.A. Frogel, and P.B. Eskridge. The detection of spiral arm modulation in the stellar disk of an optically flocculent and an optically grand design galaxy. *A&A*, 359:932, 2000.
- [58] I. Puerari and H.A. Dottori. Fourier analysis of structure in spiral galaxies. *A&A*, 93:469–493, 1992.
- [59] K. Riley, D.S. Ebert, M. Kraus, J. Tessendorf, and C. Hansen. Efficient rendering of atmospheric phenomena. *Proceedings of ACM Eurographics Symposium on Rendering*, 11:375–386, 2004.
- [60] H. Scharsach. Advanced gpu raycasting. *Central European Seminar on Computer Graphics*, 2005.
- [61] M.F.S. Schroeder, M.G. Pastoriza, S.O. Kepler, and I. Puerari. The distribution of light in the spiral galaxy ngc 7412. *A&A*, 108:41–54, 1994.

- [62] K. Sellgren, M. Werner, and H. Dinerstein. Scattering of infrared radiation by dust in ngc 7023 and ngc 2023. *Astrophysical Journal*, 400:238–247, 1992.
- [63] J. Stam. Multiple scattering as a diffusion process. *Proceedings of ACM Eurographics Symposium on Rendering*, pages 51–58, 1995.
- [64] Spitzer Space Telescope. website: <http://www.spitzer.caltech.edu/>. visited June 2005.
- [65] H. van de Hulst. Light scattering by small particles. *Dover Publications Inc., New York*, 1982.
- [66] volvis.org. website: <http://www.volvis.org/>. visited January 2005., 2005.
- [67] R. Westermann and T. Ertl. Efficiently using graphics hardware in volume rendering applications. *Proceedings of ACM SIGGRAPH*, pages 169–178, 1998.
- [68] L. Westover. Interactive volume rendering. *Volume Visualization Workshop*, pages 9–16, 1989.
- [69] K. Whitehouse. Comet explodes on jupiter and the web. *IEEE Computer Graphics and Applications*, 1994.
- [70] A. Witt, G. Walker, R. Bohlin, and T. Stecher. The scattering phase function of interstellar grains: The case of the reflection nebula ngc7023. *Astrophysical Journal*, 261:492–509, 1982.
- [71] W. Zhen and C. ODell. A three-dimensional model of the orion nebula. *Astrophysical Journal*, 438:784–793, 1995.

Ehrenwörtliche Erklärung

Ich versichere, daß ich die vorliegende Diplomarbeit selbständig verfaßt und nur mit den angegebenen Quellen und Hilfsmitteln gearbeitet habe. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Saarbrücken, 28.August 2005,