Ivo Ihrke

Reconstruction and Rendering of Time-Varying Natural Phenomena

– Ph.D. Thesis –

Dissertation zur Erlangung des Grades
Doktor der Naturwissenschaften (Dr. rer. nat.)
der Naturwissenschaftlich-Technischen Fakultät I
der Universität des Saarlandes

29. Mai 2007

Max-Planck-Institut für Informatik

Stuhlsatzenhausweg 85
66123 Saarbrücken
Germany

Eingereicht am 29. März 2007 in Saarbrücken durch

Ivo Ihrke
MPI Informatik
Stuhlsatzenhausweg 85
66 123 Saarbrücken

`ihrke@mpi-sb.mpg.de`

Betreuender Hochschullehrer – Supervisor
Prof. Dr. Marcus A. Magnor, Technische Universität Braunschweig, Germany

Gutachter – Reviewers
Prof. Dr. Hans-Peter Seidel, Max-Planck-Institut für Informatik, Germany
Prof. Dr. Marcus A. Magnor, Technische Universität Braunschweig, Germany
Prof. Dr. Wolfgang Heidrich, University of British Columbia, Canada

Dekan – Dean
Prof. Dr. Thorsten Herfet, Universität des Saarlandes, Saarbrücken, Germany

Promovierter akademischer Mitarbeiter –
Academic Member of the Faculty having a Doctorate
Dr. Rafał, Max-Planck-Institut für Informatik, Saarbrücken, Germany

Datum des Kolloquiums – Date of Defense
21. Mai 2007 – May 21st, 2007

Abstract

While computer performance increases and computer generated images get ever more realistic, the need for modeling computer graphics content is becoming stronger. To achieve photo-realism detailed scenes have to be modeled often with a significant amount of manual labour. Interdisciplinary research combining the fields of Computer Graphics, Computer Vision and Scientific Computing has led to the development of (semi-)automatic modeling tools freeing the user of labour-intensive modeling tasks. The modeling of animated content is especially challenging. Realistic motion is necessary to convince the audience of computer games, movies with mixed reality content and augmented reality applications. The goal of this thesis is to investigate automated modeling techniques for time-varying natural phenomena. The results of the presented methods are animated, three-dimensional computer models of fire, smoke and fluid flows.

Kurzfassung

Durch die steigende Rechenkapazität moderner Computer besteht die Möglichkeit immer realistischere Bilder virtuell zu erzeugen. Dadurch entsteht ein größerer Bedarf an Modellierungsarbeit um die nötigen Objekte virtuell zu beschreiben. Um photorealistische Bilder erzeugen zu können müssen sehr detaillierte Szenen, oft in mühsamer Handarbeit, modelliert werden. Ein interdisziplinärer Forschungszweig, der Computergrafik, Bildverarbeitung und Wissenschaftliches Rechnen verbindet, hat in den letzten Jahren die Entwicklung von (semi-)automatischen Methoden zur Modellierung von Computergrafikinhalten vorangetrieben. Die Modellierung dynamischer Inhalte ist dabei eine besonders anspruchsvolle Aufgabe, da realistische Bewegungsabläufe sehr wichtig für eine überzeugende Darstellung von Computergrafikinhalten in Filmen, Computerspielen oder Augmented-Reality Anwendungen sind. Das Ziel dieser Arbeit ist es automatische Modellierungsmethoden für dynamische Naturerscheinungen wie Wasserfluß, Feuer, Rauch und die Bewegung erhitzter Luft zu entwickeln. Das Resultat der entwickelten Methoden sind dabei dynamische, dreidimensionale Computergrafikmodelle.

Summary

The generation of images and animations that are virtually indistinguishable from real-world photographs is one of the primary goals of computer graphics. The existence of adequate scene descriptions suitable for processing by a computer is a prerequisite for the achievment of this goal.

Traditionally computer models are generated by a human modeler. Recently researchers have started to use real-world images taken by conventional photographic or video cameras to automatically generate digitized descriptions of objects and material characteristics that are difficult to model manually. The work so far concentrates on static or dynamic opaque objects. Limited research has been done in the area of static transparent object acquisition. A large class of difficult to model, optically complex effects that cannot be captured by current image-based techniques, is comprised of natural phenomena. This dissertation is a first step towards automatic modeling of dynamic, transparent phenomena like fire, smoke, and fluid flows from video content.

The work presented in this thesis utilizes recent advances in camera hardware. Today it is possible to record scenes from different view points simultaneously using a number of synchronized cameras. The ability to record this imagery is of paramount importance for the task of automated 3D-modeling. All acquisition techniques described here use multi-video footage as their input data.

The 3D acquisition techniques developed in this work are based on the principle of tomographic reconstruction, very similar to its application in medical imaging. The challenging aspect, however, is the small number of available viewpoints for our work since the video data must be captured simultaneously to enable the acquistion of dynamic effects. Whereas the medical imaging community can rely on hundreds of views of a quasi-static object to reconstruct its interior we have to be content with a very limited number of cameras due to the cost of the recording systems. We are interested in the automatic reconstruction of a sufficiently accurate computer model to enable photo-realistic view extrapolation. This turns out to be possible, and we present algorithms for the acquisition of dynamic, three-dimensional models of flames, thin smoke, free-flowing water columns, and heated air flows. Finally, since the goal is photo-realistic image synthesis, we present a real-time rendering approach for a large class of transparent objects that enables the display of all effects that have been reconstructed by the methods presented in this thesis and more.

Zusammenfassung

Eines der Hauptziele der Computergrafik ist die Erzeugung photorealistischer Bilder und Animationen. Photorealismus beschreibt in der Computergrafik die Berechnung künstlicher Bilder, die für das menschliche Auge nahezu ununterscheidbar von realen Photos sind. Die rechnergestützte Erzeugung solcher Bilder erfordert geeignete digitale Modelle der darzustellenden Objekte.

Herkömmlicherweise werden diese Modelle von Menschen, am Rechner, mittels geeigneter Software, in mühevoller Handarbeit erstellt. In den letzten Jahren gab es jedoch Bemühungen seitens der Forschung, diese Arbeit durch sogenannte bildbasierte Modellierungsverfahren zu vereinfachen. Dabei werden herkömmliche Photographien oder Videos als Eingabedaten für rechnergestützte automatische Modellierungsverfahren verwendet, um komplexe, schwierig in Handarbeit zu modellierende Objekte zu digitalisieren. Der Schwerpunkt der Forschung lag dabei bisher auf der Rekonstruktion statischer oder dynamischer, lichtundurchlässiger Objekte. In begrenztem Umfang wurde auch an der Digitalisierung von statischen, transparenten Objekten geforscht. Einen großen Bereich von Objekten, die mit heutzutage bekannten Methoden nicht digitalisiert werden können, bilden Naturerscheinungen wie Feuer, Rauch, Wasser und durch Hitze hervorgerufene Luftbewegungen. Diese Dissertation stellt einen ersten Schritt in Richtung automatischer Modellierung dieser Phänomene dar.

Die in dieser Arbeit vorgestellten Verfahren bedienen sich neuester Entwicklungen in der Kameratechnik. Es ist heutzutage, unter Zuhilfenahme einer Anzahl synchronisierter Kameras, möglich, eine Szene gleichzeitig aus verschiedenen Blickwinkeln aufzunehmen. Diese Informationen ermöglichen eine automatische Modellierung des Szeneninhalts durch rechnergestützte Verfahren. Alle in dieser Arbeit vorgestellten Ansätze zur Rekonstruktion von Naturerscheinungen bedienen sich dieser Aufnahmetechnik.

Der Grundansatz aller hier vorgestellten Rekonstruktionsverfahren ist die Computertomographie. Das tomographische Bildgebungsverfahren basiert auf der Aufnahme von sogenannten Projektionen des Objekts aus verschiedenen Blickrichtungen. Dabei durchdringen elektromagnetische Wellen das Objekt und werden abgeschwächt. Diese Abschwächungen ermöglichen, wie aus der medizinischen Bildverarbeitung bekannt, die Rekonstruktion des Objektinneren. Der anspruchsvolle Aspekt dieser Arbeit besteht in der geringen Anzahl der zur Verfügung stehenden Blickpunkte, aus denen das Geschehen aufgenommen wird. Diese Beschränkung ist durch die hohen Kosten eines großen Kamerasystems gegeben. Während in der medizinischen Bildverarbeitung hunderte von Aufnahmen eines quasi-statischen Objektes zur Verfügung stehen, muß unsere Anwendung mit einer sehr geringen Anzahl von Blickpunkten auskommen. Der Grund hierfür liegt in der praktischen Anordnung eines Versuchsaufbaus sowie in der dynamischen Natur der zu rekonstruierenden Phänomene.

Die in dieser Arbeit vorgestellten Algorithmen zeigen die Durchführbarkeit dieser Art von Rekonstruktion für verschiedene Naturerscheinungen. Es werden Methoden zur bildbasierten Modellierung von Feuer, Rauch, Wasser und heißen Luftströmungen aufgezeigt. Abschließend betrachten wir auch eine Methode zur Echtzeitdarstellung der mit Hilfe dieser Algorithmen digitalisierten Modelle.

Contents

Part I

Introduction

1 Introduction

1.1 Motivation

People have always been fascinated by the looks of the phenomena of nature. This led computer graphics researchers very early to the attempt of modeling these effects on the computer. The simulation methods to model the appearance of natural phenomena have become very sophisticated to the degree that the laws of physics governing the burning of fire, the rise of smoke columns, the behavior of fluid flows, the look of the sky and the stars and nearly all other effects imaginable have been simulated on computing machines to imitate these effects on a computer screen, decoupled from reality, observable for everybody in the possession of these machines to view them under arbitrary self-chosen conditions. However, although the physical processes can be simulated very realistically on today's computing hardware, there is still an artificial feel to the animations generated in this way. The computations are predictable, they lack the chaotic behavior of real-world environments where e.g. a small motion of air can drastically change the appearance of a rising smoke column or a burning flame. The motion of these phenomena can be simulated realistically as long as the physical boundary conditions are modeled appropriately. The difficulty of modeling these 'imperfections' mathematically has led to the development of image-based acquisition techniques, where real-world images are used as an input for automatic modeling techniques that try to capture the real-world appearance of objects and make them appear more life-like.

In the context of natural phenomena, surprisingly, this approach has not been followed to a great extent. Our goal in this thesis is to close the gap and capture three-dimensional, time-varying models of transparent natural phenomena such as fire, optically thin smoke, and fluid flows to provide computer graphics models not only for direct rendering but also for analysis purposes that might lead to improved future models of these effects. A promising research direction is the automatic analysis of example data and the extraction of key-features defining the phenomenon. The techniques presented in this thesis are a prerequisite to attempt this kind of research in the context of natural phenomena, and I hope they will find their application in the future.

1.2 Major Contributions

Parts of the techniques discussed in this dissertation have already been presented at various conferences and journals [85, 87, 88, 86, 89, 57, 90]. These publications form the core of the thesis and are presented here in a revised and extended form. Chapter 8 contains yet unpublished work that has been submitted for publication. The major contributions of this dissertation can be stated as follows:

- The development of a sparse view tomographic reconstruction technique that performs well with a very restricted number of camera views. It is based on a reduction of the degrees of freedom of the reconstruction problem by employing conservative information about the shape of the object.
- An adaptive grid tomographic reconstruction technique that yields higher quality reconstructions in terms of effective resolution and RMS error compared to ground truth.
- The development of a pixel accurate visual hull algorithm for arbitrarily shaped basis functions covering the reconstruction volume.
- An error projection method between the codomain and the domain of a linear operator.
- A reconstruction technique for fully three-dimensional, dynamic water surfaces.
- A new formulation of refractive index tomography, properly taking curved light paths into account.
- A versatile real-time rendering method for refractive objects enabling the use of anisotropic, volumetric material properties. The method can be used to simultaneously render a superset of the effects acquired by the reconstruction methods presented in this thesis.
- An efficient light propagation scheme for refractive objects enabling a fast pre-computation of volumetric light distributions, including light directions, caused by inhomogeneous refractive index fields.

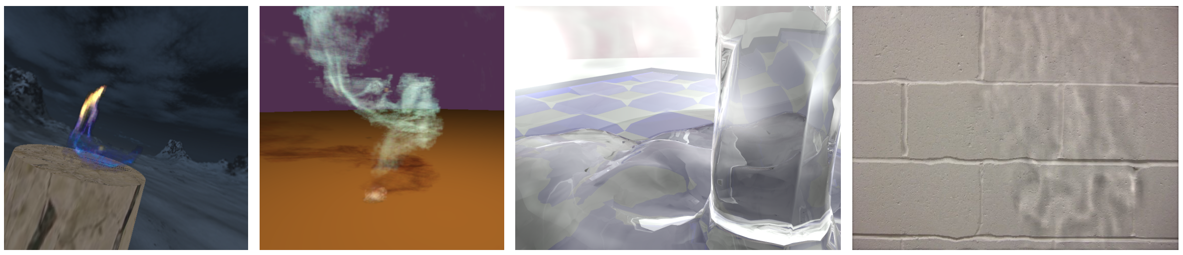


Fig. 1.1. Examples of computer models of natural phenomena acquired using the algorithms described in this thesis. From left to right: flames, thin smoke, water and heated air flow.

1.3 Outline of the Thesis

The thesis is structured as follows: In the remainder of Part I we discuss needed concepts and terminology, Chapter 2. In Chapter 3 we review the work related to our goal of reconstructing and rendering three-dimensional, dynamic models of natural phenomena. The part concludes with a description of our acquisition setup and the necessary pre-processing steps for the raw data, Chapter 4.

In Part II we develop reconstruction methods for non-refractive natural phenomena. It starts with the presentation of the basic visual hull-restricted tomography algorithm with an application to the reconstruction of time-varying flame models, Chapter 5. This algorithm is used and extended in the following chapters. Chapter 6 introduces an adaptive version of the basic algorithm with an application to smoke reconstruction. It turns out that the increased peak resolution of an adaptive representation comes at the price of a less stable reconstruction algorithm. We investigate this issue and present a remedy for this behavior.

In Part III we turn our attention to natural phenomena exhibiting refractive properties. Chapter 7 introduces a reconstruction method for free-flowing bodies of water. We present a new experimental setup that allows for the measurement of the optical path length in refractive objects. Based on these measurements we introduce a photo-consistency constraint that is used to optimize a weighted minimal surface representing the boundary between air and water. The method is suited for the reconstruction of volumes with a constant refractive index different from air. Chapter 8 eases this constraint by considering the acquisition of time-varying, inhomogeneous refractive index fields. However, the maximum refractive index magnitude that can be reconstructed is smaller than that of the water reconstruction approach. Examples of computer models acquired by the methods presented in this thesis are shown in Fig. 1.1.

Part IV consists of Chapter 9 only and discusses real-time rendering of the acquired models. The method presented in this chapter is much more general than previous approaches and can render refractive objects with highly complex material properties. A fast light propagation technique is presented as well that lends itself to the pre-computation of volumetric lighting effects like volume caustics, volumetric shadows and the like.

Finally we conclude with Chapter 10, summarizing the thesis and presenting directions for future work.

2

Background

In this thesis we are concerned with the acquisition of computer models of three-dimensional, time-varying natural phenomena such as fire, smoke, and fluid flows. This chapter provides some background information on the optical characteristics of these phenomena, Sec 2.1. The reconstruction of these time-varying models from a sparse set of images is a so called *inverse problem*. These types of problems are often not well-posed. A short introduction to inverse problems is given in Sec. 2.2. The acquisition of these phenomena is performed using multi-view video setups. The details of our experimental setups as well as preliminary steps for the preparation of the raw data are given in Chapter 4. In Sec. 2.3 we review fundamental concepts of multi-view geometry and their application to the type of problems considered in this thesis. Finally, in Sec. 2.4, we introduce the concept of photo-consistency which is the fundamental error measure for the optimization methods employed for the reconstruction techniques presented in this text.

2.1 Optical Characteristics of Natural Phenomena

Natural phenomena such as fire, smoke and fluid flows exhibit a wide variety of optical effects when interacting with light. Fire is often a self-emitting effect[1]. Fuel particles become luminous due to the heat produced by the combustion reaction. The particles approximate black-body radiators [72, 209] that also emit light in the visible wavelengths. Simultaneously, absorption takes place. When cooling off, the fire reaction products stop glowing and become visible as smoke. Here the major effects influencing the visible image are absorption and scattering. For smoke particle size is of the order of the wavelength of

[1] There are also fires that burn without a visible flame.

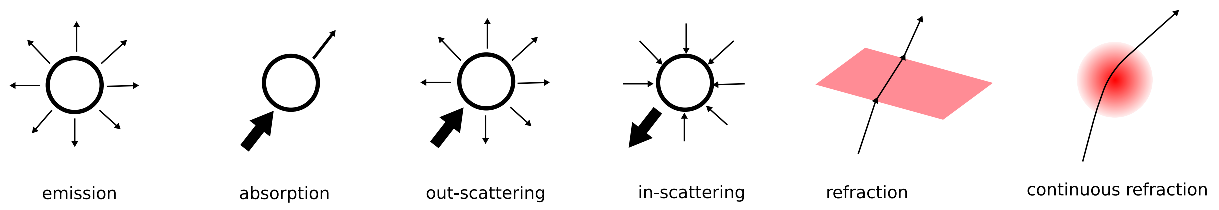


Fig. 2.1. The major optical effects occurring in natural phenomena are self-emission like in visible flames, absorption, (anisotropic) scattering most prominently visible in smoke columns, refraction at object boundaries occurring, e.g., at water surfaces, and continuous refraction often visible above heated surfaces or fires.

visible light, therefore Mie scattering is the determining factor for smoke appearance [129]. If that the smoke is optically thin, single scattering can be assumed to be sufficient to describe the scattering properties of the medium. For thick smoke multiple-scattering has to be taken into account. Furthermore the heat generated by the combustion process influences the density of the surrounding air. Changes in air density result in refractive index variations. Under normal real-world conditions these refractive index changes can be related to the change in density of the medium by the Gladstone-Dale equation [55]. The inhomogeneous refractive index distribution in hot gases leads to bent light rays as appearing, e.g., in mirages. The same effect is observable above heated surfaces or fires. When light from a strong light source is refracted and cast onto a surface the effect is called a shadow graph [180]. In case of objects with a higher refractive index like water or glass, the images caused by the convergent and divergent light rays are referred to as caustics. An illustration of the major optical effects present in natural phenomena is shown in Fig. 2.1.

The reconstruction methods presented in this thesis deal with a single effect at a time. Three-dimensional reconstruction of flames, Chapter 5, considers self-emissive phenomena. For the acquisition of thin smoke, Chapter 6, we neglect absorption and scattering effects, resorting to a trick while recording the phenomenon. The smoke is recorded under uniform diffuse illumination and thus the scattering can be assumed to be uniform. Furthermore since we are dealing with thin smoke we consider the absorption to be negligible. These two assumptions let us treat the smoke in the same way as a self-emissive phenomenon. However, the smoke columns exhibit slowly evolving, fine, turbulent structures. To resolve these in a volumetric reconstruction we develop an adaptive version of the fire reconstruction method. Refraction at well defined interfaces is treated in Chapter 7 in the context of reconstruction of free-flowing water surfaces. A technique for the acquisition of continuously varying refractive index fields is presented in Chapter 8. We reconstruct time-varying models of heated air-flows above candles and more turbulent flames.

2.2 Inverse Problems

In forward problems the task is to simulate a physical phenomenon given a number of parameters of a model describing its behavior. Inverse problems, on the other hand, consist of inferring model parameters from actual measurements of a physical process. These problems are often much more difficult to solve than the corresponding forward problem. In computer graphics these types of problems are known as inverse rendering [123, 158] or image-based acquisition techniques, e.g. [112]. Inverse problems are encountered in many areas of science such as geophysics, medical imaging, astronomy, oceanography, combustion analysis and tele-communications to name just a few. Usually a forward mathematical model for the physical effect is developed and then inverted. There is a whole class of algorithms that use the forward model directly to infer model parameters. These are known as Analysis-by-Synthesis approaches. Unfortunately, the inversion process is often *ill-posed* and unstable. A problem is ill-posed if it does not satisfy the following three criteria:

- A solution exists,
- The solution is unique, and
- The solution depends continuously on the data.

These properties are often referred to as existence, uniqueness and continuity requirements. This definition of a well-posed problem is due to Hadamard [67]. In inverse problems often the continuity requirement is violated. The model parameters, also called the solution of the problem, may be varied hugely while causing only slight changes in the measurements. This is of course undesirable since measurement noise which cannot be controlled gives rise to exactly these kinds of changes in the measurement data. The counter-strategy is to impose additional constraints on the solution, e.g., smoothness constraints or other a-priori information. These *regularization* strategies are usually problem-specific but there exist methods that are broadly applicable. Regularization details are discussed in Sect. 6.3. A good introduction to inverse problems based on a probabilistic approach is given in [194]. Regularization issues are discussed in [69] and numerical methods for performing the inversion are treated, e.g., in [23, 12].

Inverse problems can be divided into two classes: linear inverse problems and non-linear inverse problems. Linear problems are inherently simpler to solve than non-linear ones. They are characterized by a linear mapping between the model parameters and the measurement quantities. This linearity allows to model the forward problem as a linear mapping between metric spaces. After discretization, the linear operator describing the mapping can be represented by a linear system of equations, and the mapping takes place between the model vector space and the measurement vector space. For all reconstruction methods presented in this thesis the model vector space is the discretized 3D world space, and the measurement space is the union of the camera image planes of all cameras observing the scene. Unfortunately, the ill-posedness of inverse problems results in ill-conditioned linear operators, i.e., they exhibit a (numerical) null space; thus, families of solutions exist that produce very similar measurements, i.e., images. Ill-conditioned linear systems can be identified by their *condition number*, i.e., the ratio between the largest and the smallest eigenvalue of the linear system. A high condition number indicates instability of the inversion. The ill-posedness can be alleviated slightly by taking more measurements, but for inherently ill-posed problems it cannot be removed completely. The methods presented in Part II deal with linear inverse problems.

Non-linear inverse problems, on the other hand, are defined by non-linear operators between model space and measurement space. The refractive index reconstructions performed in Part III are an example of this. The non-linear nature of light transport in the case of varying refractive indices makes these problems much harder to solve than the linear ones considered in Part II. We resort to iterative, ray-tracing based approaches to deal with these phenomena, i.e., free-flowing water and refractive air flows.

Three of the approaches presented in this thesis are *tomographic reconstruction* methods. The computerized form is known as computed tomography (CT). Computed tomography is a classical inverse problem. The measurements for CT techniques are line integrals of some function or operator of the quantity to be reconstructed:

$$m = \int_c f(\mathbf{p}, \mathbf{x}) ds. \tag{2.1}$$

Here m denotes the measured quantity, $\mathbf{p}$ are the model parameters, $\mathbf{x}$ is a positions in space, f is a function(al) of these quantities, and the integral is taken along a curve c which is usually the line-of-sight. Curve c is not required to be a straight line. If p is a function instead of a discrete set of parameters, f is an operator. The quantity m is commonly referred to as the *projection* of $f(p)$. Distributing m or some function $g(m)$ evenly over the curve c is called a *back-projection* of m. Back-projection forms the basis of the most commonly used CT reconstruction algorithm, Filtered Back-Projection (FBP). Implicitly, projection and back-projection are the underlying principles of all tomographic reconstruction algorithms. Note that projection and back-projection have different meanings in the computer graphics and computer vision literature from the one just introduced for the tomography problem. To differentiate

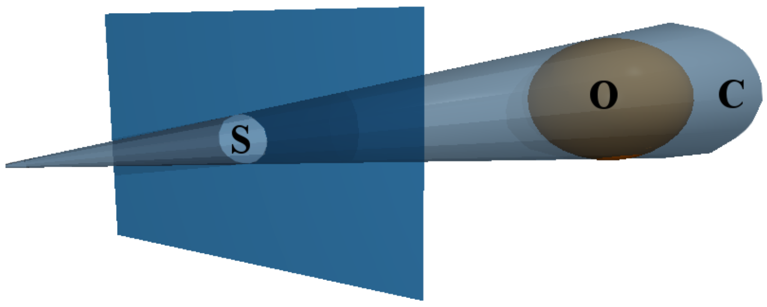


Fig. 2.2. Projection of an object O onto the image plane of a camera, forming silhouette S. The back-projected cone C is depicted as well.

between the two, we will use the terms *tomographic projection* and *tomographic back-projection* whenever we refer to line-integrals as in Eq. (2.1).

2.3 Multi-View Basics

A projection in computer graphics and computer vision typically specifies a mapping $\pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ from world space to image space. Determining this mapping for a number of cameras $C_i, i > 0$ is the task of *camera calibration.* For real cameras π_i is usually non-linear due to lens distortions. The major lens distortions are due to radially symmetric imperfections around the optical axis of the camera. Algorithms estimating these distortions and upgrading the non-linear projections π_i to linear projective mappings $\mathbf{P}_i$ are known as *radial undistortion* methods. We discuss practical camera calibration issues in Chapter 4. With a linear relationship between world and image space in place, projections of points can be written as matrix-vector multiplications:

$$x^i = \mathbf{P}_i X. \tag{2.2}$$

The projections x^i of the 3D point X into the different camera's image planes are performed using homogeneous coordinates introduced by August Ferdinand Moebius [135]. In homogeneous coordinates points in $\mathbb{R}^n$ are represented as a family of points in $\mathbb{R}^{n+1}$: $(\mathbf{x_1}, \ldots, \mathbf{x_n}) \rightarrow (wx_1, \ldots, wx_n, w)$, for all $w \neq 0$. To obtain the Euclidean image coordinates $\mathbf{x}^i$ of the projected points, we have to perform the division $\mathbf{x}^i = x^i / {x^i}_{n+1}$. Homogeneous coordinates

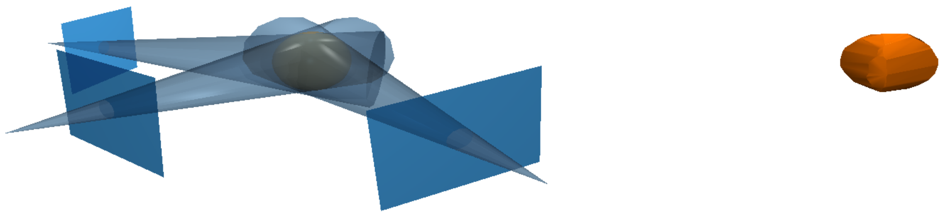


Fig. 2.3. *left:* multiple view geometry of a simple sphere and 3 cameras, *right:* the corresponding visual hull V is the intersection of the generalized cones of the sphere's back-projected silhouettes.

enable the formulation of projective mappings in terms of linear transformations.

In computer vision back-projection is the process of computing the subset C of $\mathbb{R}^3$ that projects to a subset S_i of $\mathbb{R}^2$, i.e. the image plane of camera C_i. Conceptually it can be written as

$$\mathrm{C} = \pi_i^{-1}\mathrm{S}_i. \tag{2.3}$$

C is a generalized cone with its apex at the camera center, Fig. 2.2. In case S_i contains only a single point, C is a ray passing through the camera center and the point in the image plane. We refer to this ray as a pixel's back-projected ray. The ray is given by a position in space $\mathbf{T}$ and a ray direction $\mathbf{D}$. All points $\mathbf{X} = s\mathbf{D} + \mathbf{T}$ project to the pixel at $\mathbf{x}$. To achieve this, $\mathbf{T}$ is typically chosen as the camera center and $\mathbf{D}$ is computed by

$$\mathbf{D} = \mathbf{P}^{+}x. \tag{2.4}$$

We will use a pixel's back-projected ray as curve c in Eq. (2.1) for the linear inverse problems considered in Part II and as an initialization for the bent light rays in the non-linear problems, Part III.

2.3.1 Visual Hull

Previously, we defined the generalized cone C as the set of 3D points projecting to a subset S_i of the image plane without actually specifying S_i, Eq. (2.3). Of particular interest is the choice of S_i as the set of points contained in the *silhouette* of an imaged object. Let O denote the subset of $\mathbb{R}^3$ that is occupied by the object, then $\mathrm{S}_i = \pi_i(\mathrm{O})$ is the silhouette of the object in the image plane, Fig. 2.2. Now the image-based visual hull [124] of the object is defined as the intersection of the generalized cones generated by the back-projections of the silhouettes S_i, see also Fig. 2.3,

$$\mathrm{V} := \bigcap_i \pi_i^{-1} \mathrm{S}_i. \tag{2.5}$$

The image-based visual hull is an approximation to the visual hull which was introduced by Laurentini [111] as the limit of V for $i \to \infty$. The visual hull as defined in [111] is the intersection of the infinite number of back-projected cones C_i generated by infinitely many cameras placed in the space outside the convex hull of the object O.

In general, $\mathrm{O} \subset \mathrm{V}$, i.e., the object is fully contained in the visual hull which therefore serves as a conservative approximation of the object. We use the visual hull for all reconstruction algorithms presented in this thesis either as an initialization of the true shape, Chapter 7, or as a restriction of the solution space, Chapters 5, 6 and 8.

In practice, we discretize the solution space and compute either a voxelized version of the visual hull by projecting all voxels into the recorded views, checking whether they fall into the silhouette of the object, or, for more general basis functions we identify whether they are contained in the visual hull. The latter is especially important in the adaptive tomography algorithm presented in Chapter 6.

2.3.2 Discretization of Space

As was mentioned before, all reconstruction algorithms presented in this text use measurements of line integrals of some quantity of interest for input data. For computations in a computer with finite memory, the measurements as well as the reconstructed model have to be discretized. Our strategy in Chapters 5, 6 and 8 will be to discretize the function f of Eq. (2.1) using a linear combination of basis functions

$$f = \sum_i \mathbf{f_i} \phi_i. \tag{2.6}$$

This discretization of f is then inserted into the forward equation, Eq. (2.1),

$$m = \int_c f ds = \int_c \sum_i \mathbf{f_i} \phi_i ds = \sum_i \mathbf{f_i} \int_c \phi_i ds. \tag{2.7}$$

The resulting equation is a linear combination of the coefficients $\mathbf{f_i}$ and the factors$\int_c \phi_i ds$. Function f can be vector-valued, e.g a tuple of RGB-values, Chapters 5 and 6, or describe a vector field, Chapter 8. In the case of vector-valued functions f, the coefficients $\mathbf{f_i}$ are vectors and the basis functions ϕ_i remain scalar functions.

The linear equation (2.7) describes one measurement, i.e., one pixel in terms of the unknown discretized function f. Since we measure a large number of pixels simultaneously, we arrive at a linear system of equations that must be satisfied by the unknown function f. We will compute estimates of this function by inverting the linear system(s) of equations obtained from Eq. (2.7) in a least-squares sense.

The discretization of the problem decouples the unknowns $\mathbf{f_i}$ from the tomographic projection, Eq. (2.1). Instead, we can compute the tomographic projections of the basis functions and invert a linear system to compute a discretized version of f. To obtain the model parameters $\mathbf{p}$ we still have to invert f. In Chapters 5 and 6, f will simply be unity, whereas in Chapter 8 it will be the gradient operator. Inversion is carried out by integrating the reconstructed function.

2.3.3 Computation of Tomographic Projections of the Basis Functions

An important task in this framework is the efficient computation of the tomographic projection of the basis functions introduced in Sect. 2.3.2 over potentially curved rays c. These quantities form the matrix entries of our linear systems. In theory, we have to compute a large number of these matrix entries, one for every combination of basis function ϕ_i and measurement m_j. As this is computationally expensive and the storage of the resulting matrix is not feasible, we restrict ourselves to discretizations using basis functions with local support. The simplest such choice is a basis function that is constant in one voxel and zero outside. However, we will also use slightly more complex basis functions to obtain improved reconstruction results.

To compute the tomographic projections of the basis functions we will resort either to analytical integration on axis-aligned grids, Fig. 2.4 left, or use a discretized version of the line integrals, Fig. 2.4 right. Analytic integration is usually more efficient but can only be used in case of curves c that are straight rays, restricting its use to non-refractive tomography, Chapters 5 and 6. Quadrature-based evaluation of the line integrals, on the other hand, is more flexible but less accurate and computationally more expensive, Chapter 8. In any case, the curve c along which the line integral is computed must be known before computing the tomographic projection value.

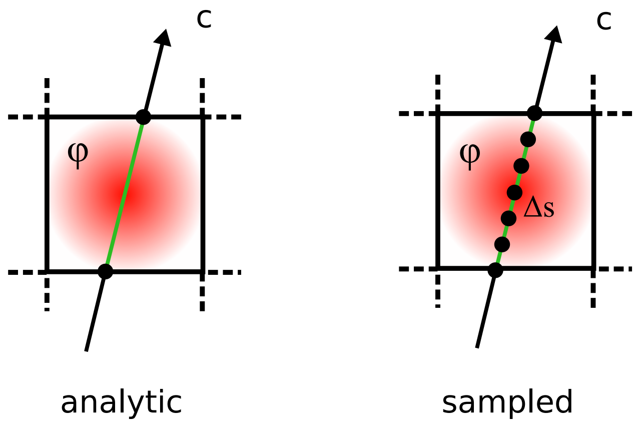


Fig. 2.4. Computing the tomographic projection of a basis function - *left:* The ray is intersected with the boundary of the axis-aligned basis function and an analytic expression for the line integral between entry and exit point is computed. Curve c must be a line. *right:* The basis function is sampled between the entry and the exit point of the ray and the line integral is evaluated using a quadrature rule with stepping Δs. Curve c can be arbitrarily shaped.

2.3.4 Computing the Curve

In the case of non-refractive phenomena, the computation of curve c is straightforward. It is simply the back-projected ray of a certain pixel as introduced in Sect. 2.3. For refractive phenomena directional changes of the ray have to be taken into account. An elegant way of doing this is the formulation of the rays' trajectory as the solution of a system of ordinary differential equations (system of ODE's). The curved ray is described as the trajectory of a particle moving in the refractive index field. The differential change of its position and direction can conceptually be written in the following way:

$$\frac{d\mathbf{x}}{ds} = \mathbf{d} \tag{2.8}$$

$$\frac{d\mathbf{d}}{ds} = f(n, \mathbf{d}, \ldots). \tag{2.9}$$

Here $\mathbf{x}$ is the position and $\mathbf{d}$ the direction of the particle, ds denotes an infinitesimal change in the tangential curve direction. Thus the position $\mathbf{x}$ changes according to direction $\mathbf{d}$, i.e., upon integration of the system of ODE's we perform a step in the current ray direction. However, direction $\mathbf{d}$ can change at every position in space, expressed by the generic function f. It depends on the refractive index, the direction $\mathbf{d}$ itself and potentially other parameters. Function f will be chosen differently for the reconstruction of free-flowing water surfaces, Chapter 7 and air flows, Chapter 8, respectively. The reason is the implementation as a level set evolution in the former and as a tomographic reconstruction problem in the latter case. Additionally, this formulation is employed for viewing ray and light propagation simulation in the real-time rendering algorithm presented in Chapter 9.

Occasionally, we will add equations to this framework to facilitate the integration of certain variables along the ray. An application is the computation of the tomographic projections, Sect. 2.3.3. These additional equations are of the following form:

$$\frac{dm}{ds} = \phi_i. \tag{2.10}$$

As can be seen by integrating Eq. (2.10),

$$m = \int_c \phi_i ds, \tag{2.11}$$

we obtain the tomographic projection of the basis function along the curved ray c by solving the system of ODE's (2.8)-(2.10) with appropriate initial conditions. These are the camera center for the position $\mathbf{x}$, the back-projected ray direction $\mathbf{D}$ for the ray direction $\mathbf{d}$ and $m = 0$.

Similarly, other quantities of interest can be integrated along the curved ray c by using equations of the same form as Eq. (2.10).

2.4 Photo-Consistency

The main goal of this thesis is to reconstruct computer models suitable for photo-realistic rendering of natural phenomena from real-world measurements. Since we are using purely optical methods with conventional camera hardware to perform our measurements, one obvious requirement for the reconstructed computer models is that they are in agreement with the acquired images when rendered from the same perspective as the originally acquired views. This requirement is known as *photo consistency.* However, this is just a necessary condition for photo-realistic view synthesis. Additionally we require good *view extrapolation* properties, i.e., if we change the virtual viewpoint in the rendering algorithm such that it does not match any of the input views, the image quality should not degrade considerably. This can happen as a result of *over-fitting* the data [72]. In Chapter 7, in the context of water reconstruction, we specifically use an experimental setup that allows for a photo-consistency measure to be defined. This measure is then used to minimize the discrepancy between the reconstructed model and the recorded images. In Chapters 5, 6 and 8 photo-consistency is optimized in a least-squares sense.

3

Related Work

The modeling of transparent, especially natural phenomena for computer graphics purposes is a challenging and computationally expensive problem that, nevertheless, has fascinated researchers for a long time. This led to the development of an abundance of methods for the realistic modeling of these phenomena or objects, both for dynamic as well as for static scenes. We present a taxonomy of transparent object/phenomena modeling techniques in Fig. 3.1. This taxonomy is mainly intended to help classifying our own work and not as a general-purpose classification scheme. E.g., the intentional omission of a distinction between static and dynamic scene modeling techniques is a source of major differences especially for simulation based methods - techniques or models that produce realistic static imagery are not necessarily suited to obtain convincing animations. However, in the context of the reconstruction of these phenomena the techniques used in dynamic settings are mostly similar to the static case. The main difference is the amount of hardware that is necessary to capture appropriate data. In some cases, though, the reconstruction methods are relying on data that is not acquirable in one time instant. We will point this out and mention that the particular method is only suitable for the reconstruction of static objects. If not mentioned explicitly, the discussed methods can be used for the reconstruction of dynamic phenomena by applying them on a per-frame basis.

3.1 Simulation

We now review simulation-based methods for the modeling of transparent phenomena. There are mainly two categories of approaches: procedural modeling like particle systems or dynamic systems and physics-based simulations.

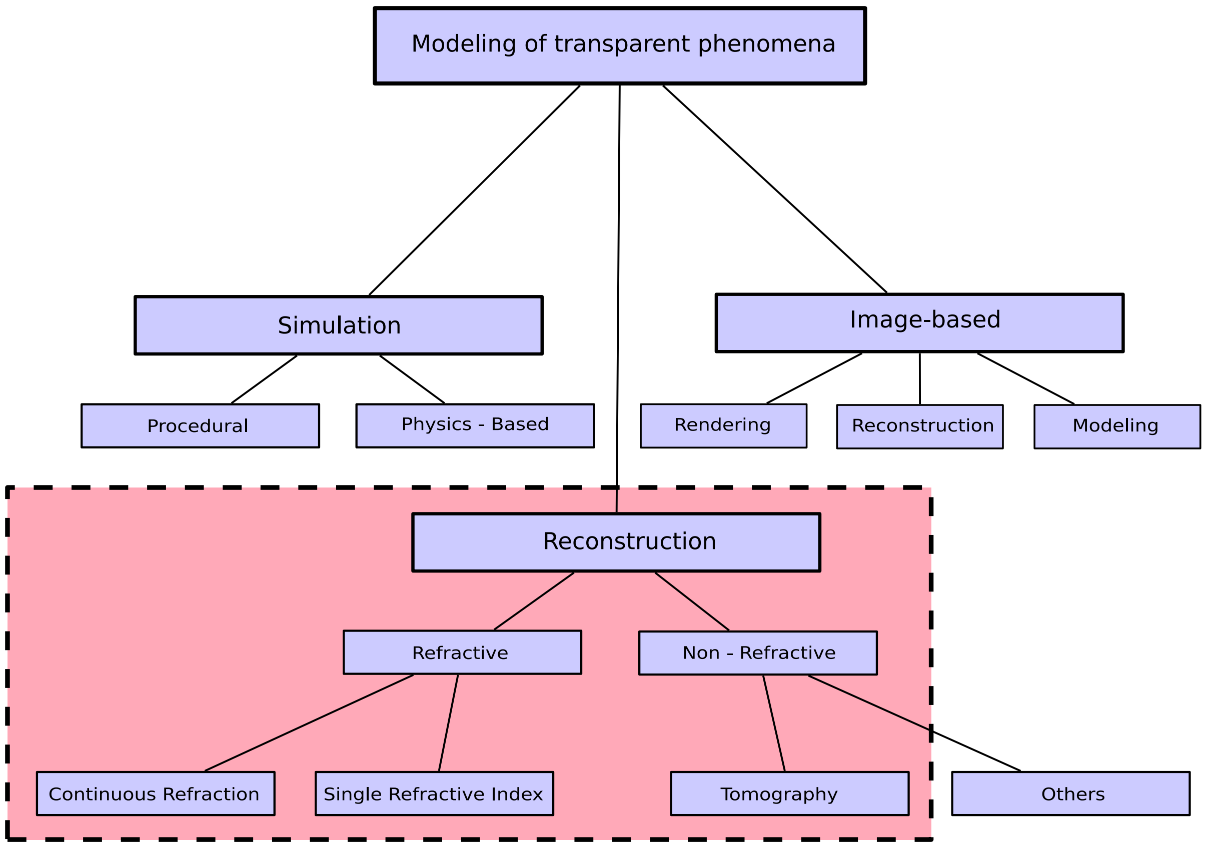


Fig. 3.1. A taxonomy of transparent object modeling techniques in computer graphics. The red box indicates the modeling approaches covered in this thesis.

3.1.1 Procedural Modeling

The first class of simulation techniques consists of mostly ad-hoc or intuitive descriptions of the qualitative behavior for phenomena like fire, water, and smoke. There is a strong tradition of procedural modeling in computer graphics reaching back to the time when computers were not powerful enough to perform the complex simulations required for physically correct modeling of the underlying phenomena. Nevertheless, procedural models still enjoy wide popularity in the graphics community. The advantages of procedural models are fast computation times and the possibility to include intuitive control for the animator. On the other hand, the burden of creating a physically plausible look is put on the operator. Of course, this also results in more freedom in creating special effects that are physically impossible.

Particle systems were the first method to be employed for the simulation of natural phenomena. Reeves' 1983 paper [163] is a classic in this respect. The method was used to animate the Genesis Demo sequence in Star Trek II: The Wrath of Khan. He used a two-layer hierarchy of particle systems, where the first layer was a particle system whose particles contained the second layer, i.e., again a particle system. In this way the complex explosion could be modeled.

Takai et al. [193] and Takahashi et al. [192] use cellular automata in two and three dimensions respectively to simulate the spread of fire. Temperatures are exchanged between neighboring cells, and when the ignition temperature is reached, a cell starts participating in a particle system that is used to render the fire. A large number of particles is necessary to achieve realistic looking results.

Cellular automata are also used to model other natural phenomena. Dobashi et al. [41] simulate cloud formation, a hybrid physics-based, procedural method is proposed by Kim et al. [98] to model ice formation. A special hexagonal grid is used to avoid procedural modeling and interpolation artifacts when switching to the physics-based simulation.

Beaudoin et al. [17] use chains of particles for fire simulation. They simulate fire propagation on polygonal meshes, building on the work of Perry and Picard [35]. The boundary between parts of the burning object that are burning and those that have not been reached by the fire is modeled on the object's surface. Then chains of particles that are animated by ad-hoc vector fields defined by the animator are released from the surface. The vector fields model effects of buoyancy, wind blows, flame flickering, and other properties. To render the fire, a combination of potential fields around each of the chains of particles is defined and rendered in a volumetric fashion. Lamorlette and Foster [108] present a similar, much more sophisticated version of this approach for a movie production environment. They also use statistical properties of real flames to guide the creation of vector fields and heuristics for splitting of the flame structures.

To enhance the visual quality of particle based fire simulations, Wei et al. [208] use texture splats [38], i.e., small rectangular textures, to render the particles. The texture images are extracted from photographs of real flames.

A popular choice in the above-mentioned methods is to add Perlin Noise [151] to the vector fields describing motion in order to model turbulence in flames, smoke, and other effects. Perlin used his noise functions to directly generate images of fire, water, and soap bubbles. Ebert and Parent [43] animate smoke by translating the evaluation region of a volumetric perlin noise texture in space, yielding a smooth transition for the modeling of volumetric smoke.

3.1.2 Physics-Based Simulation

The other major category of modeling algorithms for transparent phenomena is physics-based simulation. These techniques produce physically correct images and usually provide superior visual quality compared to procedurally modeled natural phenomena. This comes at the cost of computational expense and high memory demands which makes it difficult to achieve high-resolution simulations. The computations scale at least with $O(n^3)$, where n is the discretization in one spatial dimension [51]. It is also difficult to devise intuitive means of controlling the boundary conditions and parameters guiding the simulation. Moreover, re-runs of the simulations with slightly changed parameter values can produce dramatically different results because of the chaotic behavior exhibited by the governing equations. Thus, although the visual quality of the algorithms described in this section is very good, it is still desirable to design hybrid methods like [108] to include better artistic control over the modeling process.

Stam and Fiume [185] seem to be the first authors investigating the use of advection-diffusion equations for modeling transparent phenomena. Although they stress that they are not actually solving the equations of fluid dynamics, their equations exhibit a similar structure to the Navier-Stokes equations except for the coupling of the different physical parameters like density, temperature, pressure, and the spatial velocities of the fluid. In the tradition of particle systems, Stam and Fiume simulate the transport and diffusion of quantities based on *externally* specified velocity fields. The full Navier-Stokes equations of fluid dynamics differ in that the transported and diffused properties of the fluid themselves influence the velocity fields and, thus, the evolution of the phenomenon.

Foster and Metaxas [50, 51] introduce the solution of the equations of fluid dynamics into the computer graphics community. They show that the simulations can in fact result in high quality renderings and also point out the computational expense of these simulations. They use an explicit time stepping scheme for the solution of the equations, thus requiring many small time steps to be taken to obtain a stable numerical simulation[1].

A major break-through for physics-based simulations of natural phenomena was Stam's paper [184]. Stam introduces an unconditionally stable and fast fluid solver which found wide-spread use in later work. The main problem of his solver is numerical dissipation, i.e., excessive smoothing of the solution. Fedkiw et al. [48] introduce the method of vortex confinement to the graphics

[1] In general transport equations are similar to the wave equation and require step sizes of $O(\Delta t)$ for explicit integration schemes. In fluid dynamics, this is known as the Courant-Friedrich-Levy (CFL) condition and yields an upper bound for Δt. Diffusion equations, on the other hand, require much more restrictive step sizes of $O(\Delta t^2)$ for stability. These restrictions can be overcome by implicit time stepping which is unconditionally stable. Note that stability does not equate with accuracy, thus taking larger simulation steps typically decreases simulation accuracy.

community where small detail is added back to the solution in appropriate places, i.e., in places of strong vortices in the flow. In [48] the focus was on smoke simulation. Since then, the methods have been adapted to a wide range of natural phenomena including fire [144], explosions [218, 49], chemical reactant flows [84], water [116], and more.

3.2 Image-Based Methods

In this section we review the literature concerning image-based techniques for the modeling of transparent phenomena. We distinguish between image-based rendering, i.e., methods that use the available imagery as is, image-based reconstruction where real images are combined with coarse geometry models, and learning-based techniques. The latter class of algorithms uses the data to learn model parameters from the images or aims to recombine them to generate new photo-realistic images that have not been recorded in the acquisition step.

3.2.1 Image Based Rendering

Light field [114] or Lumigraph rendering [60] was first introduced by Levoy and Hanrahan, and Gortler et al., respectively. Light fields are a sampled representation of (parts of) the plenoptic function [1], which describes the light distribution in time and space. Light field rendering permits the generation of images of complex scenes without modeling its underlying geometry. A prerequisite for the application of this method is a large number of known views of the scene. The intermediate views are then obtained by interpolating the original views. The two methods differ in the placement the input viewpoints. While Levoy and Hanrahan require a regular sampling of the viewpoints, Gortler et al. use unstructured viewpoints acquired by a hand-held camera and resample them into a structured representation. Additionally, Gortler et al. use depth-assisted warping to generate new views while Levoy and Hanrahan rely on pure image interpolation. Although these methods have not been applied to model transparent phenomena, when combined with a large camera array [211, 212] these methods can be used to render novel view points of dynamic transparent phenomena. The depth-assisted warping is not applicable in this case, though. There have been extensions to this scheme, mapping these algorithms to graphics hardware for dynamic light fields [56] and interpolating directly from the unstructured data [29].

Another line of research deals with the recombination of the acquired images in the temporal domain, thus trading the ability to change viewpoints for the generation of new dynamics of the recorded scene. The methods in this category can work with a single video sequence of repetitive nature.

With video textures [175] complex phenomena can be animated in a non-repetitive fashion from input videos but are restricted to a fixed viewpoint unless the texture is mapped to synthetic geometry. The input videos are analyzed for loops and smooth transition points. This information is then used to re-play the video frames in a different order. A similar method specifically geared towards natural phenomena is presented by Stich and Magnor [189]. In [189] manifold analysis and warping are used to find appropriate transition points in the video and to smoothen transitions when looping the video. Kwatra et al. [107] extend this scheme to spatially varying temporal transition points in order to lessen transition artifacts in Schödl et al.'s [175] method. For phenomena that exhibit a major evolution direction, they suggest adding spatial tracking of features to obtain spatio-temporal transition effects.

3.2.2 Image-Based Reconstruction

The limitations of image-based rendering methods are due to the huge amount of data necessary to facilitate the combination of dynamic content, the ability to change the viewpoint, and possibly altering the dynamics of the phenomenon. This means that an enormous amount of hardware has to be employed to achieve this goal.

The desire to alleviate these limitations of purely image-based rendering led to the development of image-based reconstruction techniques. Image-based reconstruction techniques acquire coarse geometric data in addition to the images that the rendering is based on. A key attribute of these methods is that the reconstruction is not carried out in three-dimensional space but rather implicitly in the image plane of the newly generated views.

Schirmacher et al. [173] used per image depth maps to warp images into new views using a reasonable number of views. Matusik et al. [124] developed image-based visual hulls, an implicit reconstruction computed in the image plane only. These techniques are only applicable to opaque objects. Extending their earlier approach, Matusik et al. developed algorithms to acquire fuzzy [125] and transparent, refractive objects [126]. These approaches are based on environment matting techniques [221, 2] which lend themselves to rendering static images of transparent, refractive objects against new backgrounds. Matusik et al. eases the fixed-viewpoint constraint by acquiring environment mattes on the visual hull surface of the object, effectively augmenting a low-quality approximation of the objects' geometry with view-dependent environment mattes. Unfortunately, this requires the acquisition of a large number of images per viewpoint for establishing the background-image plane relation. Therefore, although yielding high quality renderings from arbitrary viewpoints, this method is not suitable to capture dynamic, transparent phenomena.

3.2.3 Image-Based Modeling

We refer to methods that use real world data to infer a model, i.e., a low-dimensional representation of the underlying phenomenon and estimate the model's coefficients from the available data as image-based or data-driven modeling methods. These methods try to capture the characteristic dynamics of a phenomenon. They trade the ability to change the viewpoint for the ability to generate new images that have not been captured in the original sequences. Interestingly, most of the research on image-based modeling of natural phenomena for static viewpoints has been done in the context of texture generation. A texture, in comparison to a general image, is one where every sub-image is perceived to be similar whereas this is not true for a general image [207].

The earliest work on the analysis and synthesis of temporally varying textures is by Szummer and Picard [191]. They introduce the notion of a three-dimensional video space, i.e., two spatial dimensions and one temporal dimension. An extended autoregressive model is fitted to the data, and new sequences of effects like water and steam are generated. A similar goal is pursued by Bar-Joseph et al. [10]. They employ statistical learning to obtain a statistical model which is used to generate new random samples of the underlying phenomenon. Results for fire, clouds, and water are presented. Wei and Levoy [207] use vector quantization to perform temporal texture synthesis. All these previous methods can only synthesize so-called stationary regions, i.e., inner regions of the phenomenon that exhibit a repetitive temporal behavior per pixel.

There has only been limited work in the context of full models of transparent phenomena so far. Bhat et al. [21] present a method that models natural phenomena as a flow of particles that are extracted from input video. The paths for the particles are specified by an animator, then particle motions are extracted by tracking a textured patch for each of them through the video sequence. Later, the particle paths can be edited by the animator, and the phenomenon can be changed to generate new video sequences. Results for fire, water, and smoke are reported.

Stich and Magnor [188] present a two-dimensional morphable flame model the parameters of which are determined from input video data. The parameters are used to learn an auto-regressive process for the flame dynamics that can be used to sample new instances of flame animations. The advantage of this approach is the possibility to include effects that where not present in the acquired video data. For example, external forces such as wind can be used to change the trajectory of the flame.

The fact that only little work exists for modeling of natural phenomena is surprising, given the fact that data-driven models have been applied to a wide range of modeling problems in computer graphics. To cite just a few, data-driven models have been applied to shape synthesis [181], face modeling [25, 24], example-based synthesis of motion [109], BRDF synthesis [128] and modeling by recycling parts of models, recombining them into a new geometric model [53]. Often a generic model is derived and fitted to the data. A general description of this approach can be found in [104]. A notable exception is the data-driven BRDF model [128] where the model itself is derived from the data automatically using a manifold learning technique like [195, 168, 171].

3.3 Reconstruction

The goal of reconstruction techniques is the acquisition of computer models from real world objects. Different methods focus on different physical parameters of the objects or phenomena, but the main interest is in getting a good description of the actual object in terms representable by a computer. The methods reviewed in this section come from a wide range of scientific areas and focus on different aspects of the same problem. In computer graphics the main objective is the generation of convincing imagery - the physical parameters of the objects are not of primary importance. Computer vision techniques concentrate on finding better models for scene descriptions that make computer models consistent with acquired imagery. Combustion scientists, experimental fluids researchers, and the applied optics community are typically interested in the exact physical parameters of the underlying processes. These different goals led to a wide range of approaches for the reconstruction of transparent objects/phenomena which are difficult to classify. A major difference is whether refraction is taken into account. Approaches ignoring refraction can work with a simple perspective image formation model, whereas modeling refraction requires a more complex model. When refraction is taken into account, light rays typically arrive at the camera via curved paths that are often only C^0 continuous.

3.3.1 Non-Refractive Phenomena

Related work concerned with non-refractive, transparent phenomena can be coarsely divided into tomographic approaches and others, where the other approaches differ widely and cannot be categorized easily. Methods suitable for photo-realistic image synthesis are based on laser scanning [75, 52] or are similar to tomographic reconstruction [72, 73, 74, 87, 89]. Hawkins et al. [75] use a laser plane that is quickly swept through a column of smoke. A high-speed camera captures slices of the volumetric density distribution while the laser plane is moving through the volume. Additionally, the scattering phase function is measured and the albedo of the smoke is determined to facilitate realistic rendering results. However, the setup uses expensive equipment, e.g., a powerful laser and a high-speed camera. Furthermore, the capture is not instantaneous because the laser plane needs to sweep through the smoke, thus restricting this method to the acquisition of slowly varying smoke columns. Fuchs et al. [52] alleviate this problem by trading spatial sampling resolution for instantaneous capture of the data set. They use a set of laser lines that sample the reconstruction volume sparsely and interpolate the remaining values. Although this method allows for capturing of rapidly changing flow patterns, spatial resolution is limited.

The tomographic approaches [72, 73, 74, 87, 89] use imagery from multiple conventional cameras and do not require other specialized equipment. Hasinoff and Kutulakos [73, 74] base their derivation on a photo-consistency constraint. They show that a sheet-like structure can always be made photo-consistent with two views. Furthermore, convex combinations of several sheet structures from different camera pairs are also shown to be photo-consistent. The so-called *flame sheets* can be interpreted as a special basis for the tomography problem. This basis is shown to be the spatially most restricted basis that yields photo-consistent reconstructions. In this thesis, on the other hand, we use standard bases to cover the volumetric phenomenon and estimate the emission density inside its visual hull, Chapters 5 and 6. The visual hull restriction of the solution is indispensable for good-quality reconstructions from a sparse number of views. Both methods do not exhibit the temporal aliasing problems inherent in [75] and can be applied to capture dynamic, volumetric models of fire and thin smoke.

The tomographic reconstruction problem has been studied extensively. A good overview of classical techniques to solve the problem of finding a function from measurements of its tomographic projections is given in [96]. They are mostly based on the Radon transform [157]. The most widely used method is *filtered back projection.* Graphics hardware-accelerated implementations are available [31]. In the medical imaging community, multi-resolution methods have been developed to improve robustness against measurement noise. They facilitate spot-light tomography[2] or limited-angle tomography.

[2] a tomography problem where only a small region of interest is reconstructed with good quality, whereas the remaining volume is only approximated coarsely

Some approaches are based on non-rectilinear 'optimal' grids [115, 202, 77], others on wavelet expansions of the solution or the input data [15, 161, 170]. However, these methods have not found wide-spread practical use. Another way of limiting the acquisition time or spatial resolution is an adaptive scanning process, where the mechanical setup is changed according to the region of interest [79].

Additionally, a couple of methods have been developed in the computer vision literature. However, their main goal is not photo-realistic image synthesis, but an improvement of opaque scene reconstructions by using more sophisticated models of image formation. Bhotika et al. [22] include an occupancy probability in a space carving [105] framework. This measure is taken to improve opaque surface reconstructions and is not intended to allow for transparent object reconstruction. De Bonet et al. [27] and Szeliski and Golland [190] explicitly include the possibility of transparent objects in a volumetric scene description. A photo-consistency measure is employed to automatically differentiate between opaque and transparent regions in space. The results are, however, not suitable for realistic rendering.

Computerized tomography has also been (mis-)used by applying it to conventional photographs of an opaque object [54, 165], the results are, however, no improvement over visual hull reconstructions [111] with the same number of views available. A plausible application of computerized tomography for the reconstruction of opaque objects has been the work of Reche et al. [162]. They reconstruct a pseudo-density distribution of trees and use billboard textures to render photo-realistic images of real trees from arbitrary viewpoints.

3.3.2 Refractive Phenomena

Work on the reconstruction of refractive phenomena can be divided into methods that treat objects with a, potentially unknown, single refractive index and methods that try to recover a continuous field of varying refractive indices. The former methods usually deal with objects with a high refractive index, whereas the latter can only be used in case of a relatively small maximum refractive index such as refractive index variations in hot air flows. The goal of constant refractive index reconstruction methods is usually the recovery of time-varying water surfaces or static glass objects. Variable refractive index reconstruction methods, on the other hand, are mostly used in the applied optics and experimental fluids literature to extract secondary information: the refractive index of gas flows is under certain conditions coupled to other physical quantities like density and temperature of the air flow.

Constant Refractive Index Reconstruction

Historically, the first reconstruction methods dealing with refractive phenomena are from the photogrammetry literature [80, 119]. The focus is on reconstructing underwater imagery from outside the water. A parametric surface description, i.e., a plane equation, is assumed, but the refractive index can be computed alongside the undistorted image. Computer vision techniques improved on this scheme by computing the water surface from texture distortions [138, 139, 176] detected by optical flow measurements [81, 118, 14, 9]. The refractive index is assumed to be known. A similar technique is presented by Morris and Kutulakos [136], extending the previous work by estimating the refractive index in addition to the surface position and its normals. All these methods consider a time-varying two-dimensional water surface. [136] is a special case of a theoretical analysis of the geometry of light paths presented by Kutulakos and Steger [106]. The latter leads to algorithms allowing for the reconstruction of fully three-dimensional refractive objects. However, the multi-pass measurements restrict this method to static objects. A different method presented in this thesis, Chapter 7, allows the reconstruction of time-varying, three-dimensional water surfaces [86]. The water is dyed with a chemiluminescent chemical, making the water self-emitting. The technique is based on measuring the optical path length by means of intensity measurements. Finally, Trifonov et al. [198] describe an elegant way of removing refraction effects. The object to be reconstructed is submerged into a fluid of the same refractive index, thus straightening the light paths. The object is placed into a tank containing the fluid and the ray directions are calibrated prior to object acquisition.

Variable Refractive Index Reconstruction

Variable refractive index reconstruction methods are usually formulated as tomography problems since tomographically projected refractive index variations can be measured in different ways, i.e., line integrals of some function of the refractive index field can be measured with appropriate measurement setups. The line integrals are measured using ultra-sonic waves [152], bi-focal optical coherence tomography [222], or Schlieren imaging [180]. Schlieren imaging is a purely optical measurement method. It was used predominantly for qualitative imaging of fluid flows and uses a sophisticated setup of lenses and filters in its original incarnation [172, 180]. Recently, quantitative measurements have become possible [82]. Quantitative Schlieren imaging is based on ray deflection measurements and has been simplified considerably by the advent of digital video cameras. The ray deflections in the image plane correspond to line integrals over the gradient of the refractive index field under observation. They can be measured using the color-based Rainbow Schlieren method [61] or the Background Oriented Schlieren technique [166]. Background Oriented Schlieren techniques [39, 166, 132, 44] measure the ray deflections by computing the optical flow [81, 118, 14, 9] between a reference view and the distorted view due to refractive index changes. They are therefore suitable for acquiring time-varying volumetric models of refractive index fields due to small refractive index changes. 3D reconstructions using traditional Schlieren imaging has been shown by Schwarz [177]. The acquisition setup is very sophisticated, requiring twenty traditional Schlieren setups arranged in a circle and multiple measurement passes. For this reason only quasi-static flows can be reconstructed. The same restriction applies to one-view acquisition techniques [3, 47, 40]. These methods can only reconstruct rotationally symmetric phenomena. McMackin et al. [131] and Venkatakrishnan et al. [203] demonstrate the acquisition of two-dimensional slices of a three-dimensional flow. These methods also use only one view and are thus restricted to two-dimensional measurements. In Chapter 8 we present an approach that enables a fully three-dimensional reconstruction of time-varying, continuous refractive index fields, thus improving on the state of the art.

3.4 Rendering

In Sect. 2.1 we discussed the complex optical characteristics exhibited by natural phenomena. Real-time rendering of the majority of the effects - emission, absorption, in-scatter, out-scatter, and refraction simultaneously is a complex task. Many modeling techniques discussed in the previous sections include rendering issues as well. This section is concerned with previous work regarding the rendering aspect of these optical characteristics without discussing modeling issues.

The earliest work on rendering volumes of varying refractive indices appears in the literature in context with modeling of atmospheric effects. Berger et al. [19] ray-trace mirages by repeated application of Snell's law in an off-line renderer. Musgrave [142] include total reflection which was ignored in the previous paper to render the same phenomenon.

Several approaches were published in the literature that approximate refractive effects in real-time on the GPU [147, 214], on a special signal processor [146], or in a CPU-based real-time ray-tracer [204]. Hakura and Snyder [68] propose a hybrid ray-tracing based approach that produces appealing results but does not run in real-time. Most of these algorithms achieve good results by recursively evaluating Snell's law at material boundaries. An interesting approach for displaying gemstones that handles refraction and polarization effects was presented by Guy and Soler [66].

Refraction rendering is related to the problem of rendering realistic caustics. Popular off-line approaches for high-quality caustic rendering are backward ray-tracing [7], and photon mapping, which was introduced by Jensen [93]. It is also possible to generate volume caustics [92] with the latter approach. Either of them stores photon energies in spatial storage data structures and gathers their contributions during image formation. Real-time ray-tracing systems [150, 32, 204] enable the execution of these methods at interactive frame rates [216], but typically a cluster of PCs is needed [65] to handle the computational load. Recently, these algorithms have been ported to graphics hardware to achieve real-time performance. Wand and Strasser [205] compute reflective caustics by approximating surfaces with uniformly sampled light sources. Wyman and Davis [215] propose an interactive image space technique for approximate caustic rendering on the GPU that is related to photon mapping. They also suggest a light shaft structure similar to the illumination volumes of Nishita and Nakamae [145] that approximates the intensity distribution of the flux of light in a beam in space. A similar concept is employed by Ernst et al. [46] to generate surface and volume caustics.

Volumetric scattering of light was considered first in computer graphics by Blinn [26]. Kajiya and von Herzen [95] derive a general formulation of scattering in terms of volume densities. They present general equations for single and multiple scattering. Light interaction between surfaces and volumes is treated by Rushmeier and Torrance [169] in a radiosity style algorithm. Stam [183] explores the limit of multiple scattering and presents a diffusion approximation to this limit. Mertens et al. [133] approximate single subsurface-scattering by using a dipole approximation to multiple scattering in real-time.

Recently, real-time scattering implementations for volumetric data sets have been presented. Magnor et al. [120] use a GPU ray-casting implementation to render reflection nebulae which includes emission and absorption effects. A discussion of combined emission, absorption and scattering rendering can be found in [129].

The majority of the approaches discussed so far imply ray geometry for the light paths and assume well-defined boundaries between layers of materials with differing refractive indices. An alternative approach based on wavefront tracking has been proposed by Mitchell and Hanrahan [134]. They use an analytical description of light transport and integrate the wavefront curvature along piecewise straight light paths. Stam and Languénou [186] introduce a framework based on the description of a light path as the solution to a system of ordinary differential equations. This enables tracing curved ray paths in optically inhomogeneous media with continuously varying refractive index. They use a ray-tracer to compute bent eye-rays. Seron et al. [178] use a related ODE-based framework for the simulation of atmospheric effects. The

ODE-based description of light paths can handle arbitrarily curved light rays in a similar way as non-linear ray-tracing that has, for instance, been used to simulate gravitational lenses [63, 210]. The solution of these differential equations is similar to particle tracing. Krüger et al. [103] show that particle systems in large voxel volumes can be simulated on the GPU for the purpose of flow visualization.

3.5 Discussion

The modeling techniques reviewed in this chapter allow the, more or less realistic, display of time-varying natural phenomena. Image-based methods are either restricted to a fixed viewpoint or static objects due to the multi-pass acquisition setups required for capturing these models. Simulation approaches allow for interactive manipulation of the results but are restricted in their physical accuracy either due to the modeling paradigm, numerical accuracy issues or unknown boundary conditions.

This thesis aims to provide three-dimensional real-world measurements of phenomena such as fire, smoke, and fluid flows. These data could help verify numerical simulations and serve as input data to data-driven modeling techniques. The output of the techniques presented in this thesis can be rendered directly to provide photo-realistic, dynamic free-viewpoint video of natural phenomena. This enables using common effects of free-viewpoint video like stop motion and the bullet time effect for natural phenomena. Hopefully, these techniques will provide a way to derive more realistic models of natural phenomena for image generation purposes.

On the rendering side we present a real-time rendering technique for volumes of continuously varying refractive index. This method improves on the state of the art by combining the majority of the optical effects exhibited by natural phenomena which has only been partially shown before. The method is flexible enough to allow for the simultaneous rendering of all effects considered separately in the reconstruction methods presented in this thesis. Additional effects that have been neglected can be rendered as well. Thus, it is suitable to render even phenomena for which no acquisition techniques exist today.

4

Data Acquisition

This chapter deals with the steps involved in setting up and using a multi-camera studio. We discuss recording options, calibration issues and pre-processing of the acquired video footage. The data acquired and pre-processed as described in this chapter serves as the input data for all reconstruction algorithms presented in this thesis. We start with the description of the recording setups that were used to record the experimental data for our reconstruction algorithms.

4.1 Recording

For the purpose of the reconstruction algorithms discussed in this thesis it is a prerequisite to acquire *synchronized* video data from multiple vantage points simultaneously. Synchronized multi-video data is acquired by multiple cameras that record the scene all at the same time but from different positions. Synchronization is achieved either by hardware triggering - this is the most common and accurate way - or by software triggering, where a start pulse is sent and the cameras keep their frame-rate, running on an internal clock. While still somewhat expensive, today several research labs have small scale multi-video studios available featuring 6 - 16 cameras [124, 137, 174, 196, 187, 64, 127, 220, 206]. Studios with a larger number of cameras (up to 128) have been built [160, 211, 212] but due to their large cost and the high level of technical sophistication necessary to stream the data to disk these large studios have not become too popular. Another issue is the large amount of work that is necessary to calibrate these studios.

A good overview of many multi-camera setups, the choice of cameras, lens selection, camera placement for different reconstruction tasks and the necessary hardware can be found in [121]. In the following we concentrate on our

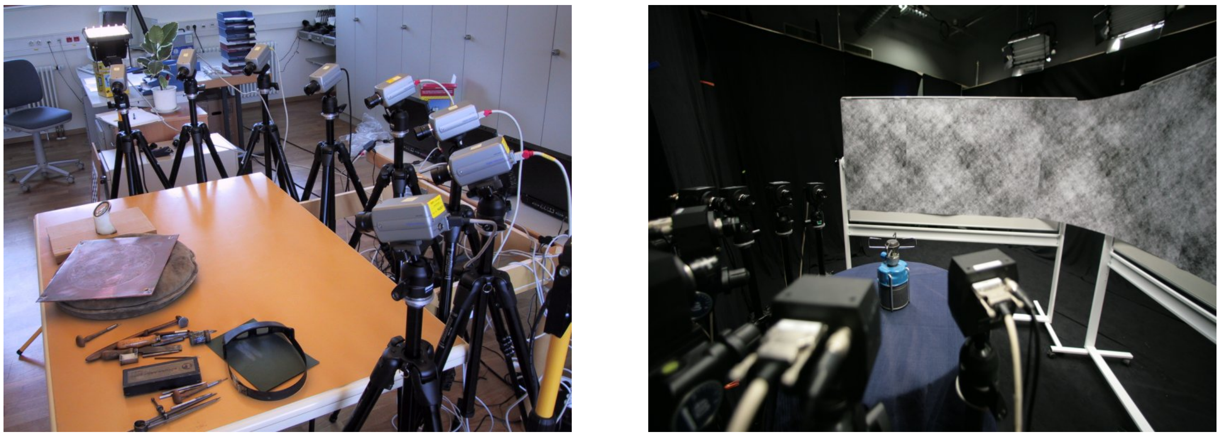


Fig. 4.1. The two different camera systems that were used to acquire multi-video data for the projects in this thesis, *left:* a mobile multi-camera studio, *right:* our in-house system.

particular application, the acquisition of multi-view video footage for the reconstruction of time-varying natural phenomena and the necessary calibration and pre-processing steps.

4.1.1 Acquisition Setup

For the research presented in this thesis we used two different acquisition systems. Chronologically the first was a mobile multi-video acquisition setup that was developed in joint work with Lukas Ahrenberg [4]. The goal was to enable the acquisition of scenes in their natural context without being restricted to record in carefully set up studio environments. The system uses 8 Sony DFW-500 FireWire cameras and 4 controlling laptops equipped with 1GB of RAM. The cameras are synchronized via wireless network and the images are captured to the main memory of the laptops. The cameras run at 15 fps, at a resolution of 640×480, 8 bit per pixel, which allows for capturing of up to 40 seconds of video footage. Afterwards the data is compressed and written to the hard-disk. If only one computer is used to control each camera the recording time is doubled. The system is shown in the field in Fig. 4.1, left. It is modular and can be extended by additional laptops and cameras. One of these modules is shown in Fig. 4.2. This camera system was used to acquire the data for the algorithms presented in Chapters 5 and 6.

The other camera system used in this thesis is a commercial system. It also consists of 8 cameras but streams directly to disk using a RAID system of SCSI hard-drives. The camera model is the Imperx MDC-1004. It can capture at a resolution of 1004×1004, 12 bit per pixel, with up to 48 frames per second. The cameras can be hardware triggered or started by a common pulse, keeping their frame-rate by running on an internal clock. The system

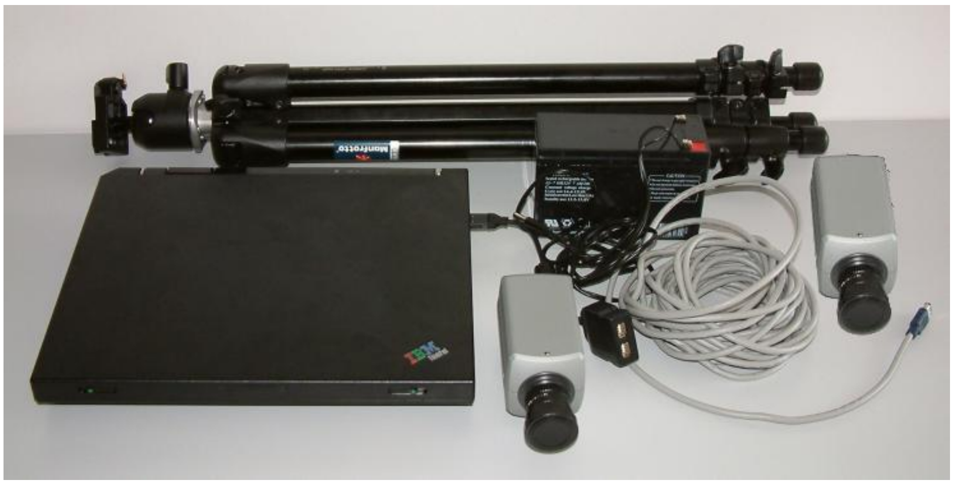


Fig. 4.2. One module of the mobile recording system [4]: it consists of two cameras, a laptop for controlling the acquisition, a rechargeable battery to power the cameras, a tripod and some wires.

is shown in Fig. 4.1, right hand side. Due to its size it can only be used in our in-house studio. The raw data for Chapters 7 and 8 was acquired using this system. A slight complication is the fact that these cameras use two A/D converters for half of the image each when recording at 48 frames per second. This results in slight color differences between the two halves of the image. A correction scheme is presented in Sec. 4.4.

After describing the general characteristics of our recording systems, we now turn to practical issues regarding the acquisition process. We use either a circular or a semi-circular camera setup. The circular camera setup is to be preferred even though we record transparent phenomena. The reason is the better quality of the visual hull that can be reached when the cameras surround the scene completely. However, due to the need for a well defined background pattern in Chapter 8 we resort to a semi-circular setup in this case. An important point to consider using a circular setup is to distribute the cameras in a way that the optical axes of the cameras are as far from parallel as possible. This happens if the cameras directly face each other and has to be avoided as far as possible because it introduces redundancy into the measurements. Since the phenomena we are recording are transparent and the camera rays for a particular camera are only slightly divergent, we would measure nearly the same datawith cameras placed opposite to each other. This is most important for the non-refractive reconstruction methods considered in Chapters 5 and 6.

Another practical issue arising in the context of fire and water reconstruction, Chapters 5 and 7, respectively is the acquisition of very low intensity levels. Since the recording takes place in a dark surrounding and the effect itself[1] produces very little light for the cameras we need to trade-off between different camera parameters. These trade-offs are

- aperture setting: light sensitivity vs. depth of field,
- exposure time: camera noise vs. motion blur, and related,
- frame rate: exposure time vs. dynamics of the phenomenon.

It is not always possible to find a good trade-off. However, we found that recording with software triggering, i.e. setting a start pulse and letting the cameras keep their pace autonomously, freed up a lot of exposure time that is simply used for waiting for the hardware trigger pulse otherwise. We performed tests to ensure that the cameras were still reasonably synchronized using a LED binary counter visible to all cameras simultaneously. The test resulted in good synchronization up to a switching rate of 200 Hz. A second observation is that in the case of recording fire, the shape of the flame is heavily determined by the exposure time. Different exposure times result in differently realistic shapes of the flames. This effect should be investigated further with respect to the human visual system[2].

Using our in-house camera system we recorded our sequences in raw mode to avoid pre-processing of the images on-board the camera. Unfortunately this was not possible with the Sony cameras. However, when using digital cameras as measurement devices all calibration and processing should be performed by the experimenter in order to ensure that the data is not modified unintentionally.

4.1.2 Bayer Interpolation

Most CCD cameras today use a monochrome chip and place an array of red, green and blue filters in front of the single sensor elements. The arrangement of the filters is known as *Bayer pattern* [16], see Fig. 4.3. This raises the problem of color interpolation, also referred to as *demosaicing*. Since the final image is supposed to have red, green and blue values at every pixel some sort of interpolation has to be applied to the shifted grids. This problem is however non-trivial since we are sampling a vector valued function at spatially different positions for the single vector components. A large number of methods, e.g [110, 99, 140, 141] have been suggested to solve this problem but it is still an active area of research. A performance evaluation of different demosaicing algorithms is presented in [159]. Following the results of this study we mostly use [110] and [99] to perform the Bayer interpolation. We use a representative

[1] alcohol flame and chemiluminescence, respectively
[2] i.e. what is the 'exposure time' of the human visual system ?

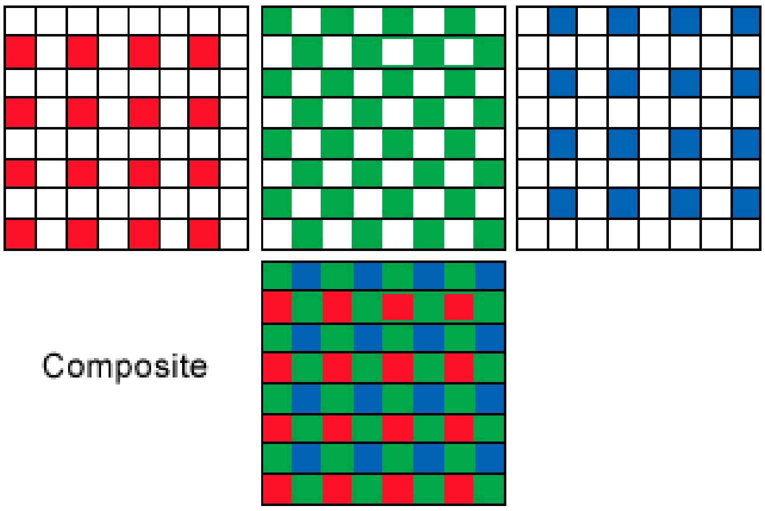


Fig. 4.3. The Bayer pattern consists of red, green and blue filters placed in front of the single sensor elements of a CCD chip. There are twice as many green filters as blue and red filters respectively. This is due to the frequency response of the human visual system, the red and blue color channels cannot be differentiated as well as the green color channel.

test image from the multi-view video sequences and run both methods. We then choose the method that produces the least artifacts.

4.2 Calibration

Camera calibration is the process of determining a camera's projection parameters and its photometric and colorimetric imaging characteristics. For multi-camera systems the goal is to find a common coordinate system for all cameras, i.e. the internal parameters of each camera which are mainly governed by the camera lens, the cameras' spatial positioning with respect to each other and a common color space such that the measurements of the single cameras can be related to one another.

4.3 Photometric Calibration

Often it is also desirable that the cameras exhibit a linear response to incoming radiance since forward models of light transport are often expressed in this radiometric quantity. An introduction to radiometric concepts can be found in [42]. Charge-Coupled Device (CCD) cameras exhibit a linear response to radiance except for both high and low ends of the dynamic range of the camera. If it proofs necessary, as e.g in Chapter 7, the camera response can be estimated by performing multiple exposures with different aperture settings [167]

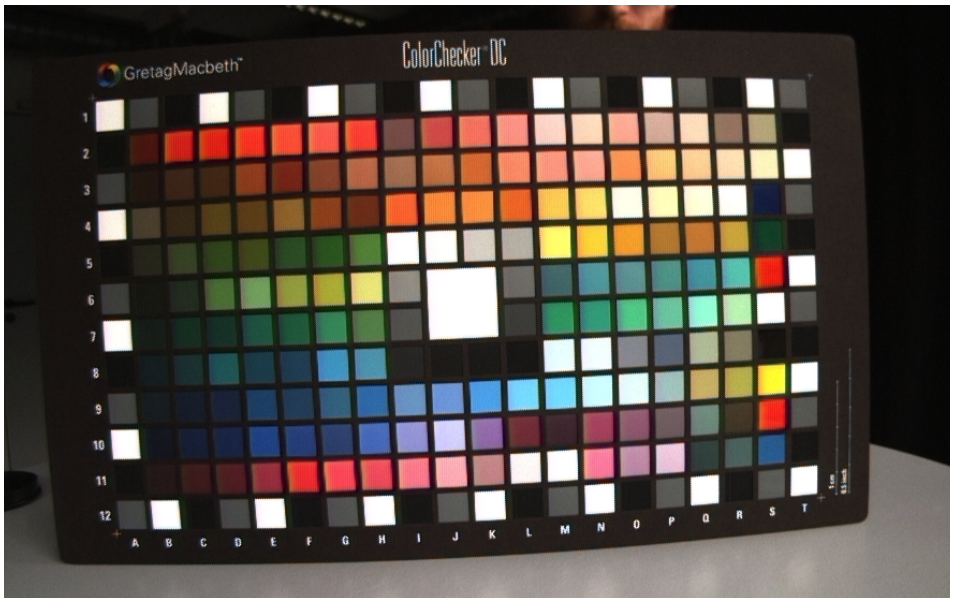


Fig. 4.4. The GretagMacBeth ColorChecker DC $^{\mathrm{TM}}$.

or using optical neutral density filters [102] where the latter is preferable for multi-camera systems since the aperture setting does not have to be restored to a common value for all cameras after the calibration step.

4.3.1 Color Calibration

Determining a common color space for all cameras of the multi-video system is called colorimetric calibration. Background information on color models, color perception etc. can be found in [217]. We differentiate between absolute and relative colorimetric calibration. Both types of methods use a color calibration chart with a number of lambertian reflectors of different color, see Fig. 4.4. When performing absolute color calibration, the colors on the chart are known under special illumination conditions [3]. Using this equipment a so called ICC profile [83] can be generated. These profiles allow for the transformation between the color spaces of the single cameras using a common profile connection space (PCS). The implementation of the transformations and measurement processes is vendor specific and typically proprietary information is involved.

Relative colorimetric calibration methods on the other hand do not try to establish a relationship between colors in some physical sense and their digital representations. The common color space is often the color space of one of the cameras. The simplest such method is known as white balancing. An

[3] Usually a D65 illuminator is required which produces a well defined spectrum similar to daylight. The illumination can be produced using specialized illuminaires, e.g. the GretagMacBeth SpectraLight III $^{\mathrm{TM}}$.

area of the image that is known to be white is selected and the multiplicative factors for each color channel are determined from this area. Different methods exist for adjusting the image statistics of two images such that they appear similar. Different color spaces like $l\alpha\beta$ are employed for this purpose [164]. In [143] multiple cameras are automatically adjusted in the camera hardware to obtain uniform color response of a 5 camera system. Joshi et al. [94] use a mixture between automatic camera control and a linear warp of the RGB-color space of each of their 100 cameras towards the reference color space of a randomly chosen camera. For our setups we use a similar approach. We record the color checker, Fig. 4.4 and extract the color patches semi-automatically using a homography based patch generation algorithm. We then estimate the coefficients of a 3×4 color correction matrix $\mathbf{C}_j^i$ for the mapping from camera j to camera i.

$$\begin{pmatrix} R_i \\ G_i \\ B_i \end{pmatrix} = \mathbf{C}_j^i \begin{pmatrix} R_j \\ G_j \\ B_j \\ 1 \end{pmatrix} \tag{4.1}$$

This color correction scheme yields acceptable results in practice. Although the mapping between color spaces could be computed using higher order polynomial models of the color transformation, in practice the restricted number of color samples often results in over-fitting and colors especially at the dark and bright end of the dynamic range of the color channels tend to overshoot.

4.3.2 Radial Undistortion

Lens distortions are found in almost all lenses, especially in lenses with short focal length. An advantage of higher price lenses is that, additionally to exhibiting less distortions than their cheap counterparts, their distortions are systematic in nature and can be described by low parameter non-linear models [130]. The major types of distortions are radial and tangential distortions [130], where the tangential part is often neglected in the computer vision literature [76]. An example of radial distortion in industrial grade lenses is shown in Fig. 4.5, left hand side. The task of radial undistortion is to recover the parameters of the lens distortion model.

There are linear radial lens distortion models [11], but these are usually not sufficient to capture the lens distortion entirely. Most radial undistortion methods rely on a polynomial model of the distortion [76, 200, 71]:

$$x = \hat{x} + (\hat{x} - c_x)(\kappa_1 r^2 + \kappa_2 r^4 + \ldots) \tag{4.2}$$

$$y = \hat{y} + (\hat{y} - c_y)(\kappa_1 r^2 + \kappa_2 r^4 + \ldots), \tag{4.3}$$

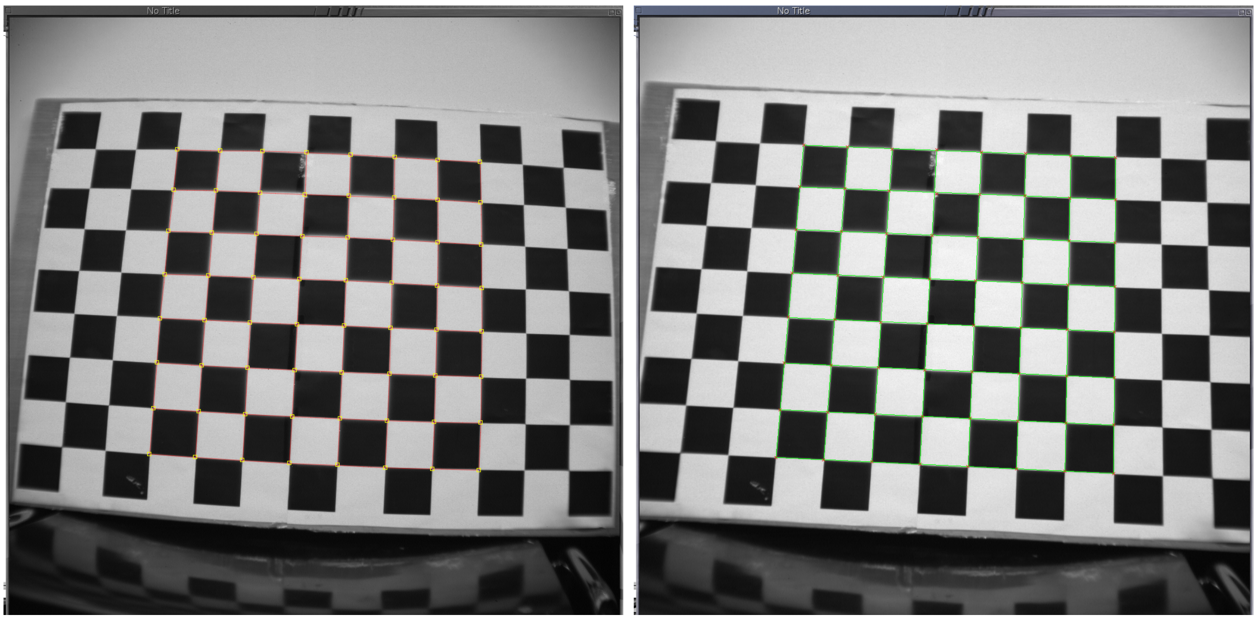


Fig. 4.5. *left:* original image with radial distortion as recorded by our cameras, *right:* after radial undistortion.

where $(\hat{x}, \hat{y})$ are the distorted and (x, y) the undistorted image coordinates. (c_x, c_y) denotes the center of the image [4] and

$$r = \sqrt{(\hat{x} - c_x)^2 + (\hat{y} - c_y)^2} \tag{4.4}$$

is the distance of the distorted image coordinates from the optical image center. κ_1 and κ_2 are called the radial distortion parameters. We developed a method similar to [97] to estimate these coefficients. We record a checkerboard pattern that covers most of the image and extract its corner points using standard image processing techniques. We use a user-defined quadrilateral to automatically generate checkerboard corner positions in absence of radial distortion. These initial points are matched to the nearest feature points extracted from the recorded image. We then impose the grid structure of the synthetically generated points onto the distorted points and generate Kochanek-Bartels splines [100] to fit the curved checkerboard lines. We introduce a measure for the curvedness of the splines

$$e = \sum_i \int_{c_i} (\kappa(s))^2 ds. \tag{4.5}$$

Here c_i denotes the i'th spline and $\kappa(s)$ its curvature at a position s along the curve. The measure e thus evaluates to the sum of the squared curvatures

[4] Actually these are the coordinates of the optical center of the camera which is not necessarily the same as the image center. Methods for characterizing and estimating the optical camera center are given in [213].

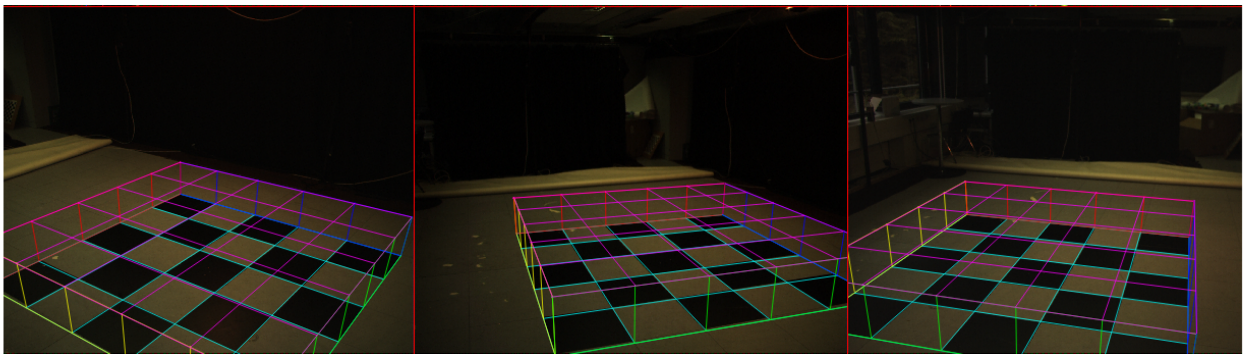


Fig. 4.6. 3 of 8 calibrated views with super-imposed world coordinates shown as a grid.

of all splines. Since the lines of the checkerboard pattern are supposed to be straight we can minimize e with respect to κ_1, κ_2, c_x and c_y

$$\min_{\kappa_1,\kappa_2,c_x,c_y} e \tag{4.6}$$

by warping the splines according to the image deformation model. The minimization is performed using the non-linear conjugate gradient method [154]. The resulting parameters are minimizing e in a least squares sense. Finally we compute a warping vector field for the corrected image. This step lets us process video sequences efficiently. A result of radial undistortion is shown on the right hand side of Fig. 4.5.

4.3.3 Geometric Calibration

After radial undistortion the non-linear effects of the projections π_i, Sec. 2.3, are removed and a linear, projective relationship remains to be determined. The projection matrix $\mathbf{P}$ is commonly [71] written as

$$\mathbf{P} = \mathbf{KR}[\mathbf{I}| - \mathbf{t}]. \tag{4.7}$$

Matrix $\mathbf{K}$ contains the internal camera parameters like pixel size and focal length, $\mathbf{R}$ is the rotation of the camera in world coordinates and $\mathbf{t}$ is the translation. There exist numerous methods to compute these calibration matrices. Some use a known calibration object (known 3D coordinates), e.g. [200, 76, 219], whereas others work from point correspondences in the image planes of the cameras only, e.g. [8, 153, 33]. The former are typically more robust whereas the latter are more flexible in that they do not require a common view of a calibration object from all cameras simultaneously.

For our mobile multi-camera system we developed a calibration technique [85] based on a virtual calibration object [8, 33] and a viewing graph [113] of the cameras. This method is joint work with Lukas Ahrenberg. It allows for the calibration of multi camera setups where not all cameras share a common viewing volume.

However, for the practical purpose of calibrating the camera setups used for the experiments in this thesis we use a different approach. It is based on the calibration technique by Zhengyou Zhang [219]. Zhang's method estimates the camera parameters using at least three homographies of a plane in general orientations. It can be used to calibrate a single camera. We typically use a much larger number of homographies for increased robustness. We record a multi-video sequence of a checkerboard moved in front of the cameras and track the checkerboard features throughout the scene. Then, for each camera separately, we estimate the plane-to-plane homographies relating the checkerboard to the camera's image plane. From these homographies we compute the camera's internal parameters and the plane positions relative to the camera. A subsequent bundle adjustment [199] increases the accuracy of the single camera calibration. After having calibrated each camera separately, we identify the camera image pairs with a shared view of the checkerboard pattern and use the estimated plane equations to compute a rigid body transform [6] for this camera pair. Ideally we can establish a relationship between one camera and all the others in the multi-camera setup. If this fails we have to resort to the graph-based approach described in [85]. An additional bundle adjustment for the final multi-camera calibration fixes minor misregistrations.

Because we work with a whole video sequence, the bundle adjustment for single camera calibration is computationally expensive when carried out in a naïve way. This is because it relies on a non-linear optimization, using e.g. the Levenberg-Marquardt [154] method, of all camera parameters simultaneously [5]. Therefore we implemented a sparse bundle adjustment procedure [117]. This is possible because the camera orientation parameters are independent for each plane position. A result of the camera calibration step is shown in Fig. 4.6 where we draw a grid defined in the world coordinate system into 3 camera views.

4.4 Image Pre-Processing

In addition to calibrating the cameras in the multi-video setup, all images acquired during recording have to be pre-processed. While for CMOS-cameras it

[5] In Zhang's algorithm [219] the moving plane is regarded as fixed and the camera is correspondingly treated as moving. Therefore 6 camera orientation parameters (translation and rotation) per plane position are estimated.

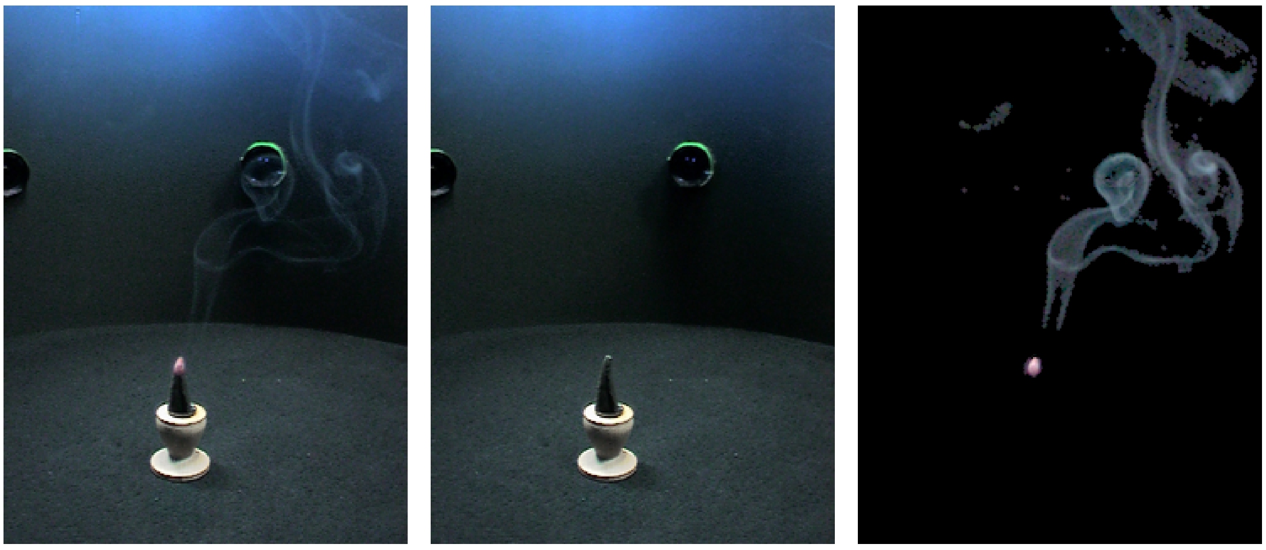

(a) Original (b) Background (c) Foreground

Fig. 4.7. Background subtraction for thin smoke: The original image (a) contains a burning incense that produces thin smoke. It is diffusely lit with daylight. (b) is a background image recorded beforehand and (c) is the background subtracted image that contains only the smoke column.

is mandatory to record a black image with the cap on the lens to capture broken 'hot' pixels, this is not strictly necessary for the CCD-cameras employed in our camera systems. A small video sequence can be recorded to estimate the mean background noise of the cameras [20] which is later subtracted from the recorded images. Further techniques for compensating non-gaussian noise characteristics of imaging sensors are described in [20] as well.

As mentioned in Sec. 4.1.1 when using the Imperx MDC-1004 cameras at full frame rate we have to deal with two A/D converters processing the two halves of an image separately. This leads to different image characteristics within the image. We adjust one of the half images' image characteristics by estimating the statistics of the two half images in a band near the splitting between the two converters. Afterwards we adjust the pixel values of the second half image to meet the statistical distribution of intensity values of the first half image.

4.4.1 Background Subtraction

In Sec. 2.3.1 we described the image based visual hull that we will use frequently throughout this thesis. A prerequisite-requisite for computing it is the automatic identification of object silhouettes. For this purpose we record a background sequence of 100 frames for each shot. We then proceed to compute the median background image and the per pixel standard deviation of the noise present in the background sequences. This enables us to classify pixels into foreground and background in the image sequences containing the recorded phenomenon. The foreground classification is stored in a silhouette image and the video frames are processed by subtracting the median background image, see Fig. 4.7.

4.4.2 Silhouette Processing

The silhouettes acquired in this way usually have small holes due to imperfections in the simple classification scheme. We therefore post-process them by applying standard image processing techniques like median filtering and morphological operators [182] to the silhouette images. This step sometimes increases the size of the silhouettes but since we are reconstructing volumetric models of transparent phenomena and use the silhouette data for initialization purposes only, this does not have adverse effects on the reconstruction results.

Part II

Reconstruction of Non-Refractive Phenomena

Overview

This part of the thesis deals with the reconstruction of dynamic natural phenomena that can be described as emissive density fields. We consider two classes of phenomena. These are fire and thin smoke. Although refraction is present in flames we ignore this effect in this part of the thesis. This is possible because the ray deflections due to heat produced in the flame are typically small. An alternative to assuming straight rays will be presented in Part III, Chapter 8.

In the following Chapters 5 and 6 we develop methods for the reconstruction of these phenomena. The methods are validated by tests with synthetic test data and the behavior of our algorithms with respect to important parameters of the problem is investigated. In Chapter 5 we present the basic approach, visual hull restricted computerized tomography, that enables the reconstruction of these phenomena from a sparse number of views.

Chapter 6 then extends this scheme to an adaptive version of the algorithm. The adaptive grid is shown to benefit the reconstruction quality while lowering memory demands in exchange for processing time. A new class of error measures for the splitting heuristic is introduced and evaluated with respect to reconstruction performance. Regularization issues are shown to play a major role in the adaptive version of the algorithm. The uniform grid exhibits a superior numerical conditioning and is straight forward to implement. The adaptive version of the algorithm on the other hand demands a more detailed analysis of the problem and different regularization strategies.

5

Fire Reconstruction

In this chapter we introduce the basic tomographic reconstruction algorithm for transparent natural phenomena that we will use with variations, with exception of Chapter 7, throughout the thesis to reconstruct volumetric computer models of fire, smoke and fluid flows. The basic ingredients are an algebraic formulation of the reconstruction problem, the restriction of the solution space by using the visual hull of the phenomenon and iterative methods for solving large, sparse linear systems of equations. The application of these techniques to the reconstruction of flames is presented in this chapter. Next we introduce the forward image formation model that our reconstruction algorithm is based on.

5.1 Image Formation Model and Basic Equations

We use a simplified image formation model for fire which was introduced by Hasinoff et. al [73]. The fire is modeled as a 3D density field ϕ of fire reaction products i.e. soot particles. Image intensity is related to the density of luminous particles in the fire. The model has the form

$$I_p = \int_c \phi \, ds + I_{bg}. \tag{5.1}$$

Here I_p is pixel p's intensity, c a curve through 3D space, ϕ is the density field of the soot particles and I_{bg} is the background intensity. Curve c is the back-projected ray of pixel p in our case. This model assumes infinitely many pixels. We approximate every pixel by one ray through the density field. As will be seen later this assumes that the back-projection cone of one pixel is smaller than the local support of the basis functions we will use to approximate the density field ϕ. Additional assumptions of the model are

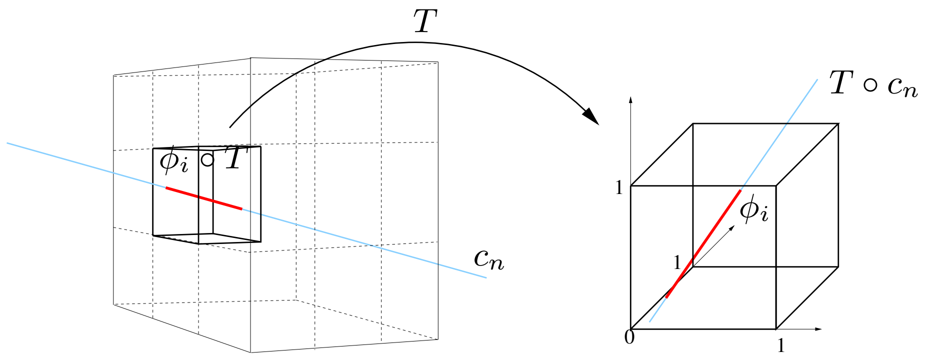


Fig. 5.1. Relationship between basis function ϕ_i, defined on the unit cube, and curve c defined in world coordinates.

- Negligible absorption/scattering - this assumption is valid for fire not substantially obscured by smoke, and
- Proportional self-emission - the brightness depends on the density of the soot particles only

In order to invert Eq. (5.1) we have to make an assumption on the structure of ϕ. We do this by assuming that ϕ can be represented as a linear combination of basis functions ϕ_i:

$$I_p = \int_c \left(\sum_i a_i \phi_i \right) ds + I_{bg} \tag{5.2}$$

The sum and the coefficients a_i can be moved out of the integral and we get

$$I_p = \sum_i a_i \left(\int_c \phi_i \, ds \right) + I_{bg}. \tag{5.3}$$

Eq. (5.3) describes a linear system of equations,

$$\mathbf{p} = \mathbf{Sa} + \mathbf{b}. \tag{5.4}$$

The rows represent the equations for one pixel and the columns contain the integrals of the pixel's back-projected rays over the basis function ϕ_i. The choice of the basis functions ϕ_i is essential for the tractability of the problem. Vector $\mathbf{b}$ will be zero for the purposes of this chapter since we record our fire sequences in the dark. In Chapter 6 where we consider smoke reconstruction this is no longer true and we discuss additional processing steps there.

Eq. (5.4) has to be solved for the coefficients $\mathbf{a_i}$ contained in vector $\mathbf{a}$ to reconstruct the density field of the flame. Unfortunately, matrix $\mathbf{S}$ is not well behaved. It has, in general, a large condition number (the quotient between the largest and the smallest singular value), making the inversion of

Eq. (5.4) an ill-conditioned problem. This inversion is exactly the computerized tomography (CT) problem. The CT problem is usually solved using the Radon transform [157]. The Radon transform uses the Fourier slice theorem to obtain the reconstruction by applying an inverse Fourier transform. This method has the drawbacks that a camera setup is required where all cameras principal axes meet in one point, and the basis functions must be of the type

$$\phi_i^{Fourier}(x, y, z) = e^{i(\alpha^i x + \beta^i y + \gamma^i z)}. \tag{5.5}$$

These basis functions have infinite support and thus give rise to a full matrix **S** (i.e. every pixel is influenced by every basis function). To be able to solve the linear system in Eq. (5.4) we would like to have a sparse matrix **S**. This is obtained by choosing basis functions with local support. The simplest of these basis functions is the box function:

$$\phi_i^{Box}(x, y, z) = \begin{cases} 1 & \begin{array}{l} x^i_{min} < x \leq x^i_{max} \\ y^i_{min} < y \leq y^i_{max} \\ z^i_{min} < z \leq z^i_{max} \end{array} \\ 0 & \text{else} \end{cases} \tag{5.6}$$

Most algebraic reconstruction techniques (ART) [96] use this basis function and approximate Eq. (5.3) in some way.

The advantage of algebraic reconstruction techniques that use basis functions with local support is the ability to constrain the problem and restrict the solution space by keeping basis function coefficients from being estimated that are known to be zero. We will use this fact to perform a sparse-view tomographic reconstruction of good quality using images of fire.

5.2 Implementation

The following section presents implementation details, how to efficiently set up and solve the linear system (5.3). We split the process into two steps and describe them separately. The matrix generation process determines the entries of **S**, regardless of additional knowledge about the solution. This knowledge is used in the process of solving the linear system. Regarding the reconstruction problem as two separate parts allows for the efficient processing of whole sequences of video data.

5.2.1 Setting up the Linear System

As can be seen in Eq. (5.3), the entries of matrix **S** consist of integrals over the basis functions ϕ_i. Since these are chosen to have local support, they are zero over a wide range of the volume. Therefore the integral is zero for a large number of entries of matrix $\mathbf{S}$. Determining the entries $\mathbf{S_{ji}}$ amounts to intersecting the back-projected rays of all pixels with the support of all basis functions. This is essentially a volume ray-tracing process. To simplify matters, we choose voxel-aligned basis functions. This choice decreases the amount of computation needed for the intersections from $O(n^3)$ (intersect all basis functions) to $O(3n)$ (intersect $3n$ planes and perform a suitable lookup).

We now consider specific types of basis functions and the resulting structure of matrix $\mathbf{S}$ of Eq. (5.4). We present a unified approach to the integration problem for different kinds of basis functions. It is not the most efficient implementation for the box basis function but serves as an example for more complicated cases. We define the basis function on the unit cube, transform the curve c of Eq. (5.3) from world coordinates to the unit cube, perform the integration and adjust the result so it is valid in world coordinates (see Fig. 5.1). This is similar to the approach taken in Finite Element Methods (FEM) for computing the stiffness matrix in the numerical analysis literature. We need to compute

$$\mathbf{S_{ji}} = \int_{c_j} \phi_i \circ T \, ds \tag{5.7}$$

where c_j is the ray back-projected from pixel j, $j = 1 \dots j_{max}$, and j_{max} is the number of pixels that are influenced by any of the basis functions in all camera images. $A \circ B(x) = A(B(x))$ denotes the concatenation of functions. We want to perform the integration in unit coordinates. Therefore, the integral has to be transformed in the following way:

$$\int_{c_j} \phi_i \circ T \, ds = \frac{||\mathbf{d}_j||}{||T\mathbf{d}_j||} \int_{T \circ c_j} \phi_i \, dt \tag{5.8}$$

The factor $\frac{||\mathbf{d}_j||}{||T\mathbf{d}_j||}$ relates the integral in unit coordinates to the integral in world coordinates, $\mathbf{d}_j$ is the direction vector of the back-projected ray c_j. This factor is only valid for a linear curve c and a linear transform T. The proof is given in Appendix A.1.1.

Box Basis Function

In case of the box basis function ϕ_i^{Box}, the whole computation simplifies considerably. $\phi_i^{Box} \circ T$ is unity in exactly one voxel i and zero elsewhere. $\mathbf{x}^1$ and $\mathbf{x}^2$ are the points of intersection of ray c_j with voxel i. The integral on the unit cube can then be transformed in the following way: Let us consider the curve c_j in world coordinates as

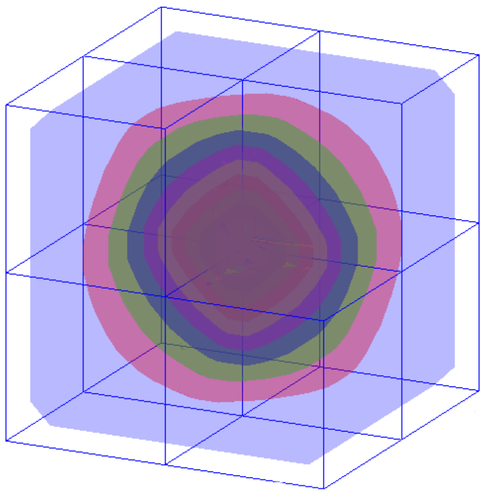


Fig. 5.2. Visualization of the trilinear basis function. This basis function has a support of 8 voxels (1 removed for better visibility). The values of the function are shown as transparent iso-surfaces. At the meeting point of all voxels, the function is one, on the borders it falls off to zero.

$$c_j(t) = (1-t)\mathbf{x}^1 + t\mathbf{x}^2 \tag{5.9}$$

and denote the transformed curve $T \circ c_j$ as

$$F(t) = T \circ c_j(t) = T\big((1-t)\mathbf{x}^1 + t\mathbf{x}^2\big). \tag{5.10}$$

Applying Eq. (A.4) to compute the integral in unit coordinates yields

$$\mathbf{S_{ji}} = \frac{||\mathbf{x}^2 - \mathbf{x}^1||}{||T(\mathbf{x}^2 - \mathbf{x}^1)||} \int_0^1 \phi_i^{Box}(F(t)) \; ||T(\mathbf{x}^2 - \mathbf{x}^1)|| \; dt = \tag{5.11}$$

$$||\mathbf{x}^2 - \mathbf{x}^1|| \int_0^1 \phi_i^{Box}(F(t)) \; dt. \tag{5.12}$$

Because $F(t)$, $t \in [0,1]$ is completely contained in the support of ϕ_i^{Box} and $\phi_i^{Box} = const. = 1$ we arrive at

$$\mathbf{S_{ji}} = ||\mathbf{x}^2 - \mathbf{x}^1||. \tag{5.13}$$

This is simply the distance that the back-projected ray travels in the voxel corresponding to basis function ϕ_i^{Box}.

Trilinear Basis Function

The trilinear basis function has a support of 8 voxels that are arranged in a cube. A visualization is shown in Fig. 5.2. In the center of this cube the function is one and on its borders it falls off to zero. The values in between are trilinearly interpolated. This results in a cubic polynomial in three dimensions for each voxel. The intersection of the back-projected ray $T \circ c_j$ in unit coordinates and the basis function ϕ_i^{TriLin} is a cubic polynomial in every voxel as well, and can thus be integrated analytically. The coefficients of the polynomial can be found by computing an approximation which is exact because polynomials approximate polynomials of the same degree perfectly. Another option is to compute it using a computer software like Maple. The polynomial that has to be integrated is given in Appendix A.1.2.

The trilinear basis functions are arranged on the voxel grid such that there is a one-voxel overlap in every dimension. This ensures smooth blending when rendering the fire. Furthermore, it is well suited for visualization using graphics hardware (see Sect. 5.3).

5.2.2 Solution

After having set up the linear system (5.4) we face a number of difficulties:

- the matrix $\mathbf{S}$ is large,
- the linear system is ill-conditioned, and
- we want to obtain a physically plausible, i.e. non-negative density field ϕ.

We wish to compute a least squares solution to Eq. (5.4):

$$\mathbf{a} = (\mathbf{S}^T\mathbf{S})^{-1}\mathbf{S}^T(\mathbf{p} - \mathbf{b}) \tag{5.14}$$

The size of the matrix $\mathbf{S}^T\mathbf{S}$ that is to be inverted is the overall number of basis functions squared. For a reasonably resolved voxel model i.e. more than 64^3 voxels, this is a large system which can only be solved using iterative methods.

5.2.3 Conjugate Gradients for a Regularized Solution

Fortunately there exist iterative solution methods for linear systems of equations with regularizing properties which is especially useful for our application. The conjugate gradient method (CG, e.g. [23, 69]), developed to solve large symmetric positive definite (SPD) matrices, is suitable for our task. The normal equations, Eq. (5.14), are by construction symmetric and positive definite.

Hansen [69] discusses the regularizing properties of the CG method in detail. Despite the incomplete theoretical understanding of the convergence properties it is experimentally and partially theoretically shown that the CG method behaves quite similar to the truncated singular value decomposition.

```
Initialization:
    $\mathbf{a}_0 = 0, \quad \mathbf{r}_0 = \mathbf{p} - \mathbf{S}\mathbf{a}_0, \quad \mathbf{d}_0 = \mathbf{S}^T\mathbf{r}_0, \quad k = 1$
Iteration:
    $\alpha_k = ||\mathbf{S}^T\mathbf{r}_{k-1}||_2^2 / ||\mathbf{S}\mathbf{d}_{k-1}||_2^2$
    $\mathbf{a}_k = \mathbf{a}_{k-1} + \alpha_k \mathbf{d}_{k-1}$
    $\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_k \mathbf{S}\mathbf{d}_{k-1}$
    $\beta_k = ||\mathbf{S}^T\mathbf{r}_k||_2^2 / ||\mathbf{S}^T\mathbf{r}_{k-1}||_2^2$
    $\mathbf{d}_k = \mathbf{S}^T\mathbf{r}_k + \beta_k \mathbf{d}_{k-1}$
Termination: L-curve criterion
```

Fig. 5.3. The CGLS algorithm.

The singular values are captured in their natural order starting with the largest [69].

This leaves us with finding a non-negative solution. We choose basis functions that are non-negative everywhere. This ensures that non-negative coefficients $\mathbf{a}$ yield a non-negative density field. Therefore we have to find a non-negative solution vector to Eq. (5.14). We do this by projecting the current solution $\mathbf{a}_k$ to the subspace of non-negative solutions in every iteration k of the CG method. This is done by setting the negative entries of $\mathbf{a}_k$ to zero. We apply the CGLS variant [69] of conjugate gradient methods to our problem. This variant was developed for solving the normal equations without explicitly computing $\mathbf{S}^T\mathbf{S}$. This saves memory because the explicit representation of the product is usually dense. The CGLS algorithm is given in Fig. 5.3 for completeness.

As the termination criterion we adopt a variant of the L-curve criterion [23, 69]. The norm of the solution $||\mathbf{a}_k||_2$ is plotted against the norm of the residual error $||\mathbf{S}\mathbf{a}_k - \mathbf{p}||_2$ on a log-log scale. The point of highest curvature on this curve is the best trade-off between a smooth solution and accuracy in the fit [69]. The number of iterations of the CG method plays the role of the regularization parameter in our case.

The results of applying this methodology (using the box basis function) to an input multi-video sequence of 8 camera streams is shown in the upper row of Fig. 5.4. As can be expected, the number of views is not sufficient to restrict the solution to the real density field and ghosting artifacts are clearly visible.

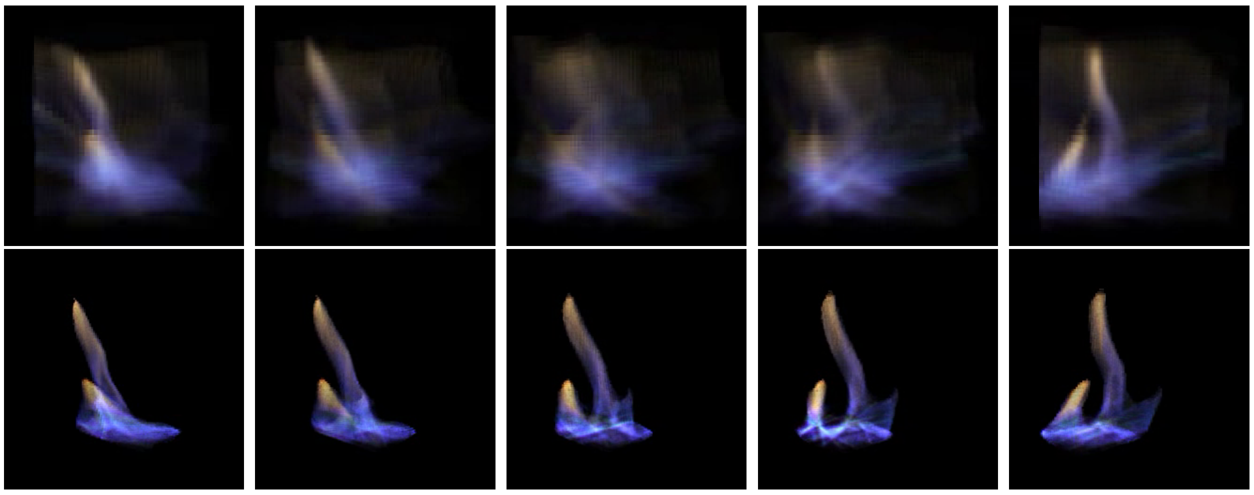


Fig. 5.4. Synthesized views of the reconstructed volume based on the full equation system, Eq. (5.4) (upper row) and the visual hull restricted system (lower row). The left and rightmost images correspond to views near original input views, whereas the middle views are in between views. Ghosting artifacts are clearly visible in the full system case (upper row), the density field suffers from over-fitting. These problems are resolved in the visual hull-restricted solution (lower row). The images cover approximately 90° and the viewpoints are equally spaced.

5.2.4 Visual Hull Restricted Solution

We circumvent this problem by exploiting additional information. We know that each basis function whose support projects outside the silhouette of the fire in any source image must have a coefficient of zero because the density field vanishes outside the visual hull. We can use this information to determine the basis functions with possibly nonzero coefficients. Fig. 5.5 shows the area of discretization and the basis functions whose support is inside/outside the visual hull in case of the box basis function, and a discretization of 64^3 basis functions. As can be seen from the figure, most of the 64^3 basis functions are situated in 'empty' space. Only about one tenth of them contribute to the density field that is to be reconstructed.

Since the contribution of each basis function to each pixel in the source images is stored in the columns of the linear system (5.4) it is simple to adjust it such that the coefficients of those basis functions whose support lies outside the visual hull are not estimated: we simply have to remove the corresponding columns from the linear system before solving the normal equations.

5.2.5 Animated Fire

The restriction of the linear system at solving time is an efficient way to deal with multi-video sequences, i.e. the reconstruction of a time-varying volumetric model of the flame. Given a static camera setup, the only things that

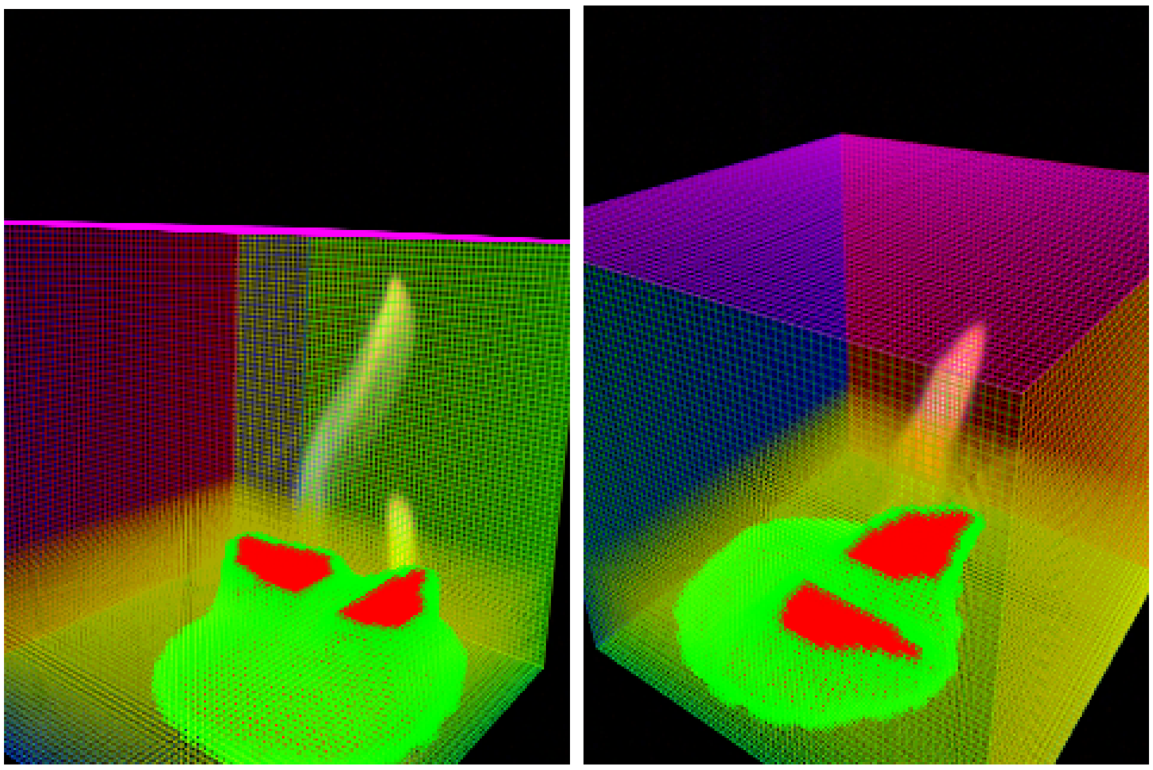


Fig. 5.5. The image shows the basis functions inside the visual hull in red, partially inside and partially outside green and outside yellow. The box depicts the area of discretization seen from two of the recording camera's viewpoints.

change from frame to frame are the affected basis functions and the right hand side of the linear system (5.4). Since setting up the linear system is much more expensive, computationally, than solving it, it is advisable to compute the full linear system first and restrict the problem while solving for every frame of the multi-video sequence.

Some results when using this procedure are shown in the lower row of Fig. 5.4. The images are taken from the same virtual viewpoints as in the row above. By constraining the reconstruction to the volume of the visual hull, photo-realistic rendering results can be obtained.

5.3 Connection to Volume Rendering

Given the coefficients $\mathbf{a_i}$ of the basis functions ϕ_i, rendering corresponds to the direct application of the forward image formation model, Eq. (5.3). Therefore, it is sufficient to create the matrix $\mathbf{S}$ for the new view and perform the matrix multiplication, Eq. (5.4), to obtain the pixel values. Matrix $\mathbf{S}$ is a pre-computed direct volume rendering [129]. The similarity between volume rendering and computerized tomography is also pointed out in Ref. [122], where the Fourier Slice Theorem is used to speed up direct volume rendering.

An animation from a static viewpoint can be obtained very efficiently since it amounts to just one matrix-vector multiplication per frame.

We reproduce the color information by computing the reconstruction for each color channel separately. The rendering is performed using three different voxel models that are rendered to the three color channels, respectively.

In case of the trilinear basis function it is more convenient (and faster) to perform a volume slice rendering approach using modern graphics hardware [31]. Modern graphics cards perform the trilinear interpolation automatically, so the coefficients $\mathbf{a_i}$ can be used as a volume texture to perform the rendering. Hardware accelerated rendering allows for interactive frame rates. The frame rate is restricted by the transfer of the volumetric flame models to the memory of graphics hardware. On modern PC's around 150 volumetric flame models can be fit into the main memory of the computer. One volumetric model uses about 25 MB of memory. We have suggested a wavelet based compression scheme to reduce this size and enable real-time rendering of the sequences reconstructed with the methods presented in this chapter [5]. Due to space restrictions we will not discuss the method in this thesis.

5.4 Experimental validation

We recorded a multi video sequence with the mobile camera setup presented in Sec. 4.1.1. This system uses 8 Sony DFW-500 cameras, recording at 640x480 pixels and 15 frames per second. An approximately circular camera setup was used to acquire the images. The recording was performed in a darkened room with the fire being the only source of light. It was therefore possible to circumvent the step of background subtraction. However, we had to use very high gain settings, introducing noise in the images, notable in the blue channel, Fig. 5.7(a).

5.4.1 Experiments

We experimentally analyzed the dependency of our method on the discretization resolution. Since we aim at creating photo-realistic images from arbitrary viewpoints using the reconstructed volumetric model, we perform a reconstruction, followed by a rendering of the model. We reconstruct a volumetric model using all views except for one, which in turn is used to validate the rendered image of the model. Seven out of the eight views are used to reconstruct a sequence of 100 frames. The reconstructed model is then projected into the eighth view and the difference is computed for each color channel.

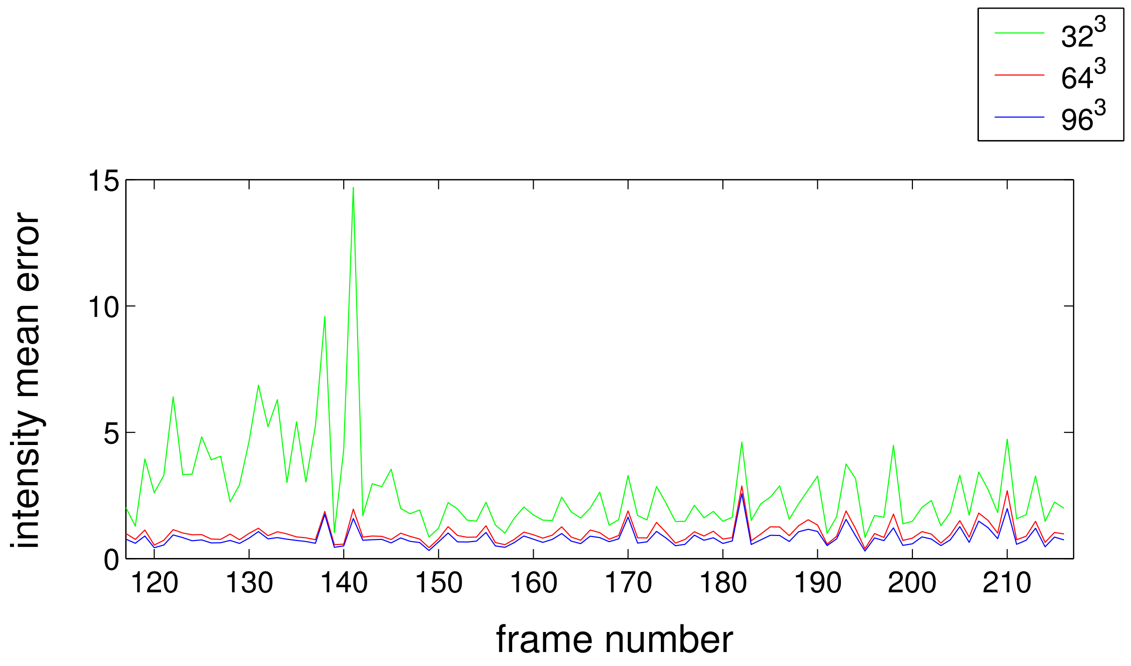


Fig. 5.6. Plot of the mean reconstruction error for all frames of a sequence. The different curves show results for a discretization of space into 32^3 (green), 64^3 (red) and 96^3(blue) voxels.

The average difference in intensity is shown in Fig. 5.6 for all 100 frames and different levels of discretization. The pixel values range from zero to 255.

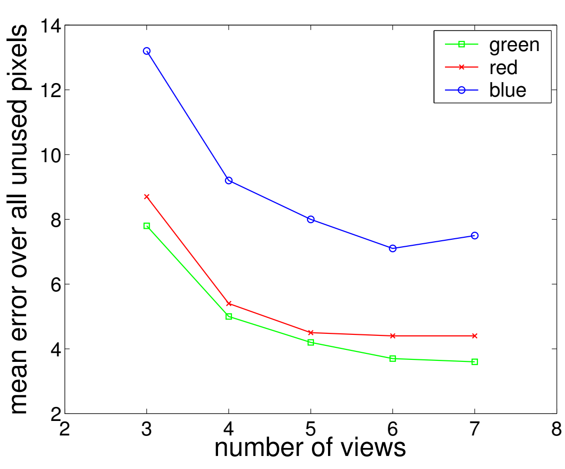
Fig. 5.6 shows that approximation quality becomes better with higher level of discretization. We also performed experiments with a discretization level of 48^3 and 72^3 voxels. The results of these are not shown for clarity, but are included in Fig. 5.7(b). These experiments show the tendency to converge towards the correct unused view, suggesting that the 3D structure of the density field is indeed captured accurately.

Another experiment was performed to evaluate the dependency on the number of views that are used for the reconstruction. We reconstructed one frame of the sequence using 3, 4, 5, 6 and 7 views of the flame. The comparison is performed against the left-out views. The results are shown in Fig. 5.7(a). Here as well, the convergence of the solution can clearly be observed. Visually acceptable reconstruction results, especially when used in animated renderings, are obtainable with as few as 4 to 5 views. 8 views are sufficient for photo-realistic rendering.

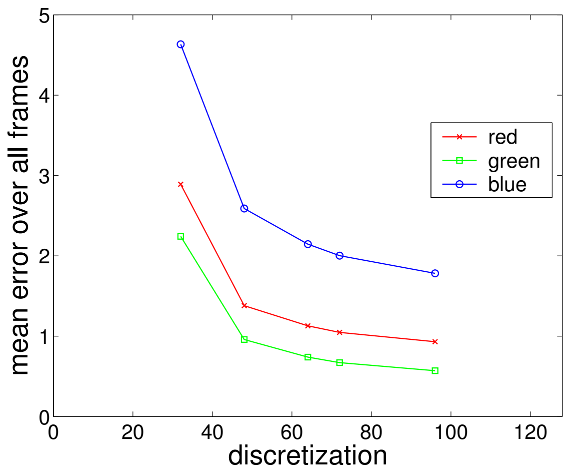
5.4.2 Discussion

Potential sources of error of our method are

- camera calibration errors,



(a) error vs. number of views



(b) error vs. discretization

Fig. 5.7. Mean error plotted against the number of views used for reconstruction (left) and the mean error over the whole sequence of Fig. 5.6 against the level of discretization. The three curves denote the three color channels red, green and blue.

- color calibration errors,
- 2D image processing (rescaling), and
- discretization (number of views and spatial discretization).

Calibration errors are inevitable and tend to create too small visual hulls. This means that the actual silhouette is a bit larger than the silhouette of the reconstructed model. Since this does not happen with synthetic test images, such errors can be attributed to camera calibration inaccuracies. We exclude these pixels from the error measurement to prevent biasing the error measure.

Rescaling the images is necessary to fit the matrix in memory (2 GB RAM). It should be noted that 2D image processing might influence the result of the reconstruction because it can introduce effects not described by our model. Projecting the result of the reconstruction back to the original views and referring to Eq. (5.14), the reconstruction/rendering loop $\hat{\mathbf{p}} = \mathbf{S}(\mathbf{S}^T\mathbf{S})^{-1}\mathbf{S}^T\mathbf{p}$ can be interpreted as a filter on the 2D pixel data $\mathbf{p}$ where $\hat{\mathbf{p}}$ is the solution projected to the original views. If the pixel data $\mathbf{p}$ is filtered in 2D prior to the reconstruction procedure, it might not correspond to a filter in the 3D domain and therefore introduce artifacts. Finally, the discretization of the density field and the assumption on its special structure, i.e. its composition from basis functions, introduce errors of their own.

While one has to be aware of these error sources, our validation experiments demonstrate that it is possible to compute density fields that enable photo-realistic image synthesis from arbitrary viewpoints.

5.5 Summary

We have presented a method that is capable of reconstructing dynamic, volumetric models of fire for animation purposes. Our approach is applicable in case of

- negligible scattering (fire not obscured by smoke)
- no sensor saturation in the input images
- no opaque objects inside the flame
- no part of the flame is seen by less than 2 cameras

We obtain photo-realistic results. Validation shows that our approach reconstructs the actual 3D distribution of flame intensity. Our results are obtained using 8 cameras, demonstrating that the method is applicable in a sparse view scenario. The reconstruction is optimized for processing of multi-video-sequences, making it a suitable tool to model fire for animation purposes.

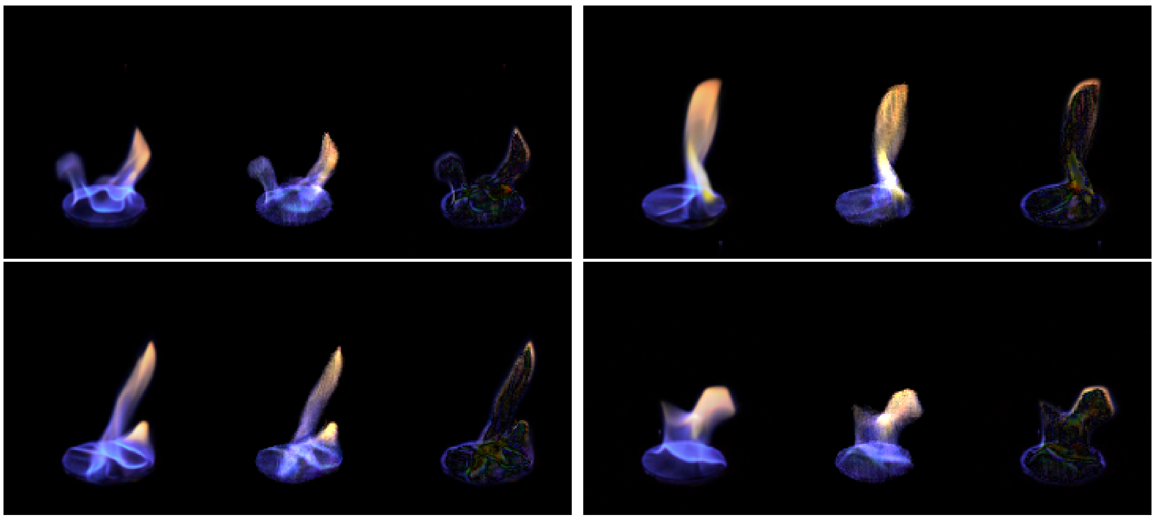


Fig. 5.8. Difference between original, unused views and reconstruction rendered from the same viewpoint. Left: Original, Middle: Reconstruction, Right: Difference.

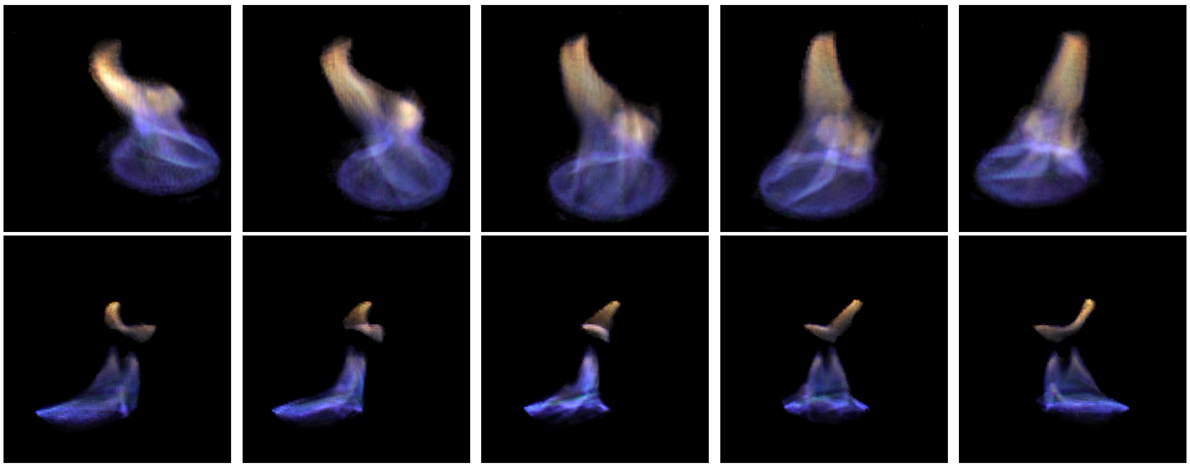


Fig. 5.9. Synthesized views of two different flames with black background, the 5 images of each row cover approx. 120°.

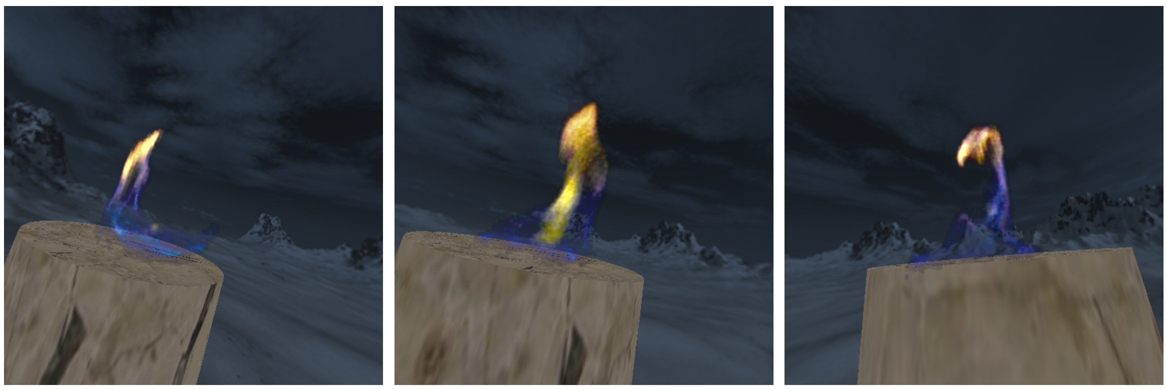


Fig. 5.10. Synthesized views of animated fire in a virtual environment.

6

Adaptive Grid Tomography - Smoke Reconstruction

In the previous chapter we introduced the sparse view tomographic reconstruction of emissive phenomena like fire. This chapter deals with an extension of the basic approach. We demonstrate that under carefully chosen recording conditions the basic method, Chapter 5, can be applied to the reconstruction of thin smoke columns. To capture the fine details present in whirling smoke we extend the previous method to an adaptive grid scenario. The idea is to concentrate the resolution of the reconstruction where fine detail is visible and to approximate homogeneous regions by larger basis functions. Since we still rely on the visual hull of the phenomenon to restrict the solution space, we present a technique to compute this information directly from matrix **S**, introduced in Sec. 5.1. The non-uniform basis functions introduced in this chapter lead to stability problems in the solution of the linear system presented earlier. We will discuss the reasons for this and present a remedy for this behavior.

6.1 Application of the Image Formation Model to Smoke

To apply the method presented in the previous chapter to smoke we have to make sure the image formation model is more or less valid. Both assumptions stated in section 5.1 are obviously not true for smoke in general. We tackle the problem by making the following assumptions

- The smoke is uniformly and diffusely lit, and
- Scattering takes place in a uniform manner.

These assumptions make it possible to treat the smoke as a self-emissive medium. We found this model to be applicable for thin smoke reconstruction.

In the case of flame reconstruction it is possible to record in a dark setting, avoiding the complication of background subtraction for transparent phenomena, i.e. vector $\mathbf{b}$ of Eq. (5.4) is zero. Since the smoke has to be lit uniformly we have to perform background subtraction which was described in Sec. 4.4.1. This step computes the difference $(\mathbf{p}-\mathbf{b})$ in advance and is depicted in Fig. 4.7. In the following we regard vector $\mathbf{p}$ as the background-subtracted pixel vector.

Reconstruction and re-rendering of the reconstructed model into the original views can be seen as a 2D image filtering operation as discussed in Sec. 5.4.2. This filter is defined by the image formation model and the reconstructed density field. Pre-processing of the input images might not correspond to a valid filter of this form and thus affect the reconstruction process negatively. Fortunately, background subtraction and noise reduction are not observed to have adverse effects on the visual quality of the reconstruction.

6.2 Adaptive Reconstruction

An adaptive reconstruction technique for three-dimensional computerized tomography is motivated by the better memory efficiency that can be achieved. This in turn yields higher effective resolutions for the reconstructed models.

Using a uniform grid, the memory limit of 2 GB on a 32-bit machine is reached relatively fast. In our experiments we have found that using 8 cameras with images taken at 320x240 pixel resolution, we can achieve a reconstruction resolution of 128^3 voxels.

In general we have n_p rows with $O(\sqrt[3]{n_b})$ non-zero elements each in matrix $\mathbf{S}$, where n_b is the number of basis functions needed to approximate the density field and n_p the number of pixels in a frame (a frame is one time frame of a multi-video sequence and contains n_c images, where n_c is the number of cameras). A uniform discretization of the reconstruction space is assumed. We use an index-stored sparse matrix as a representation for $\mathbf{S}$. We store the two indices and the matrix value for each non-zero entry. Even though this is the most memory efficient storage scheme for an unstructured sparse matrix, the available memory fills up quite fast when using a uniform grid.

6.2.1 Basic Iteration

We now turn to the modifications necessary to convert the algorithm of Chapter 5 into an adaptive reconstruction scheme. An adaptive reconstruction algorithm iteratively estimates the solution and refines it in appropriate places. This is done by performing the following iteration:

1. Estimate the coefficients $\mathbf{a}$ of Eq. (5.4), then

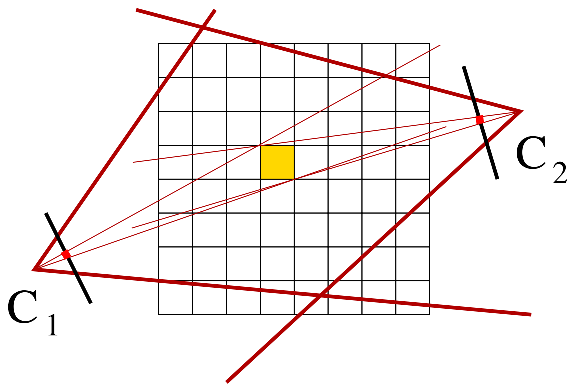


Fig. 6.1. The error heuristic measures the accumulated residual error in the image plane for each basis function.

2. Compute an error measure for each basis function, followed by
3. Splitting of the k basis functions that are responsible for the largest error, and
4. optionally, lifting of the current solution to the new discretization[1].
5. Until convergence go to step 1.

The following subsections cover the single steps in detail. The estimation process (step 1) has already been described in Section 5.1. It is independent of the shape of the basis functions and therefore directly applicable to the adaptive algorithm.

6.2.2 Error Measure

The main difficulty in the adaptive estimation process is to relate the residual error

$$\mathbf{r} = \mathbf{p} - \mathbf{S}\mathbf{a} \tag{6.1}$$

to the interpolation error

$$|u - P_{\phi_i} u| \tag{6.2}$$

Here u is the perfectly reconstructed function and $P_{\phi_i} u$ its projection onto the subspace of functions representable by the basis functions ϕ_i. A relation between the two errors allows for the identification of the coefficients $\mathbf{a_i}$ that contribute most to the residual error. We present a heuristic for this projection step and show the feasibility of an adaptive computerized tomography reconstruction.

[1] This can speed up convergence of the algorithm but might also lead to a local minimum.

The basic idea of our heuristic is based on the projection of the basis function's regions of support into the camera images, and on the accumulation of the residual error of the affected pixels, see Fig. 6.1. This yields an intuitive way of relating the error caused by a particular basis function to the residual error in the image plane. In addition the error measure is efficiently implemented using sparse matrix - vector multiplications.

Our main observation is that the complete geometry of the problem is encoded in the matrix $\mathbf{S}$. The system of equations (5.4) has the following structure:

$$\begin{pmatrix} \mathbf{p_1} \\ \mathbf{p_2} \\ \vdots \\ \mathbf{p_{n_p}} \end{pmatrix} = \begin{pmatrix} \int_{c_1} \phi_1 ds & \dots \int_{c_1} \phi_{n_b} ds \\ \int_{c_2} \phi_1 ds & \dots \int_{c_2} \phi_{n_b} ds \\ \vdots & \vdots \;\; \vdots \\ \int_{c_{n_p}} \phi_1 ds & \dots \int_{c_{n_p}} \phi_{n_b} ds \end{pmatrix} \mathbf{a} \tag{6.3}$$

Since our basis functions ϕ_i have local support the matrix $\mathbf{S}$ is sparsely populated. Note that every column of the matrix corresponds to a particular basis function. The rows are the equations for one particular pixel. Therefore an entry $\mathbf{S_{ji}}$ is non-zero only if the support of basis function ϕ_i projects onto pixel $\mathbf{p_j}$. We use this observation to formulate the projection of the basis functions into the images and the accumulation of residual errors per basis function in matrix notation:

$$\mathbf{e}_{\phi_i} = \frac{1}{n_{\phi_i}} \sum_j \delta^S_{ji} |\mathbf{r_j}| \tag{6.4}$$

$$\delta^S_{ji} = \begin{cases} 1 & : \mathbf{S_{ji}} \neq 0 \\ 0 & : \text{else} \end{cases} \tag{6.5}$$

The basis function ϕ_i is visible in n_{ϕ_i} cameras, and vector $\mathbf{e}$ contains the error measure for all basis functions.

Careful observation of Eq. (6.4) shows a close similarity to the L_1 norm $||\mathbf{r}||_1 = \sum_j |\mathbf{r_j}|$ for each of the basis functions' associated error values. This motivates the introduction of a new class of error norms which we call the class of *projected codomain L_p norms.* We define it in the following way

$$\mathbf{e}_{\phi_i} = \frac{1}{n_{\phi_i}} \left(\sum_j \delta^S_{ji} |\mathbf{r_j}|^p \right)^{\frac{1}{p}}, \tag{6.6}$$

where δ^S_{ji} is defined as before.

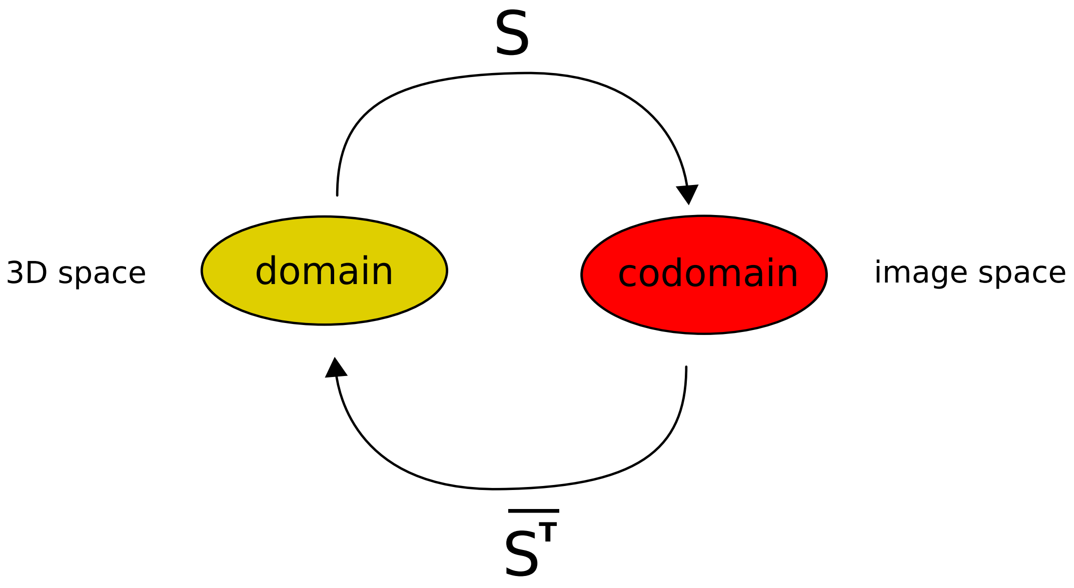


Fig. 6.2. The linear operator $\mathbf{S}$ maps the domain to the codomain and its adjoint operator $\overline{\mathbf{S}^T}$ maps the codomain back to the domain. In the case of real matrix entries $\mathbf{S_{ji}}$ the complex conjugate matrix $\overline{\mathbf{S}^T}$ is simply the matrix transpose $\mathbf{S}^T$.

Now consider the linear matrix operator $\mathbf{S}$ which is a discretized version of the linear operator defined in Eq. (5.1). $\mathbf{S}$ maps the domain, i.e. the emission densities in three-dimensional space, to the codomain of intensity values in the image plane, see Fig. 6.2. This implies that we cannot measure the reconstruction error in the domain of the operator. Instead the residual error $\mathbf{r} = \mathbf{S}\left(\mathbf{S}^T\mathbf{S}\right)^{-1}\mathbf{S}^T\mathbf{p} - \mathbf{p}$ resides in the codomain of the operator, i.e. in the image plane. The proposed error measure performs a tomographic backprojection, see Sec. 2.2, of the L_p norm of the residual in the codomain to the three-dimensional domain of the density values.

Now we prove that our error measure is indeed a norm in the domain of the linear operator $\mathbf{S}$. The norm axioms

$$||\mathbf{x}|| \geq 0 \tag{6.7}$$

$$||\mathbf{x}|| = 0 \Leftrightarrow \mathbf{x} = \mathbf{0} \tag{6.8}$$

$$||\alpha\mathbf{x}|| = |\alpha|||\mathbf{x}|| \tag{6.9}$$

$$||\mathbf{x}+\mathbf{y}|| \leq ||\mathbf{x}|| + ||\mathbf{y}|| \tag{6.10}$$

are shown to hold for the class of projected codomain L_p norms. Essentially this follows from the linearity of operator $\mathbf{S}$ and the reasonable assumption that no basis functions are present that do not influence any of the data. The latter requirement is enforced by the visual hull restriction of our operator.

Inequality (6.7) follows from the norm properties of the regular L_p norm and a positivity condition on the matrix elements δ^S_{ji} which are fulfilled be-

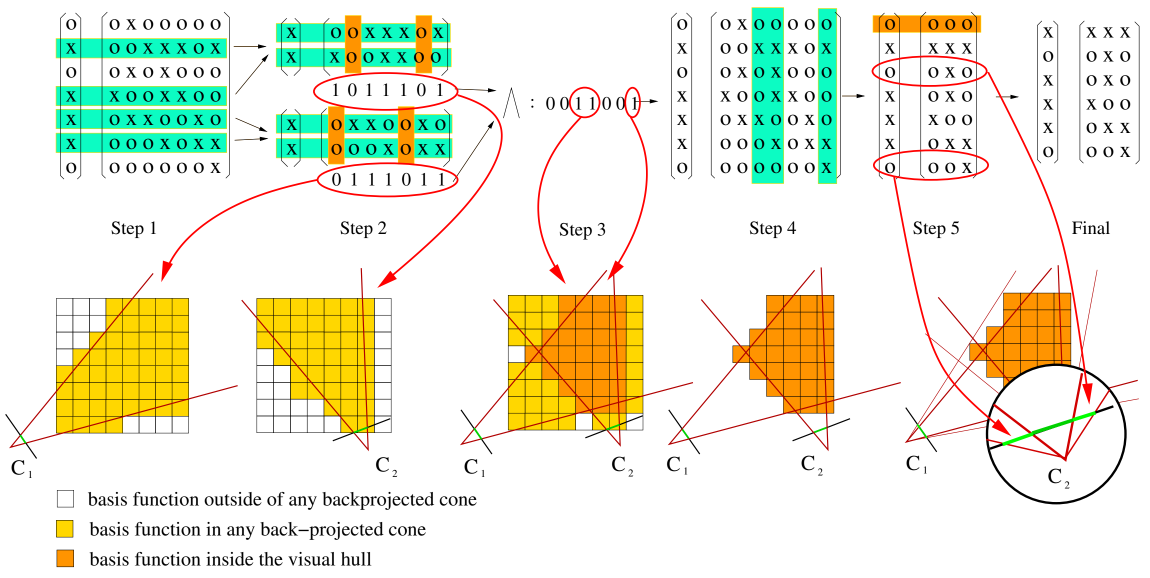

Step 1
Step 2
Step 3
Step 4
Step 5
Final
1 0 1 1 1 0 1
0 1 1 1 0 1 1
⋀ : 0 0 1 1 0 0 1
C₁
C₂
basis function outside of any backprojected cone
basis function in any back–projected cone
basis function inside the visual hull

Fig. 6.3. Visual hull restriction of matrix **S**.

cause they are either zero or one. Condition (6.8) is true iff there does not exist a column in δ^S_{ji} that is zero for all j, i.e. there is no basis function that does not project to any pixel. This condition is enforced by construction of the matrix **S** which is described in Sec 6.2.3.[2]. Eq. (6.9) holds because of the linearity of δ^S_{ji} and inequality (6.10) is a modified version of the Minkowski inequality:

$$\left(\sum_j \delta^S_{ji}|x_i + y_i|^p\right)^{\frac{1}{p}} \leq \left(\sum_j \delta^S_{ji}|x_i|^p\right)^{\frac{1}{p}} + \left(\sum_j \delta^S_{ji}|y_i|^p\right)^{\frac{1}{p}} \tag{6.11}$$

of which a more general form is proved in Appendix A.2. The performance of the adaptive reconstruction algorithm using the projected codomain L_1, L_2 and L_∞ norms is assessed in Sect. 6.4.1.

6.2.3 Splitting and Basis Function Independent Visual Hull Restriction

Using the error heuristic from the previous section, we iteratively split those k basis functions that cause the largest residual error. k is an arbitrary number that influences the convergence of the adaptive scheme. It should not be set too large because our error measure is just a heuristic and unnecessary basis functions might be introduced. We perform a uniform splitting on the box basis functions. For different types of basis functions the splitting process

[2] see step 4 of the visual hull restriction process

becomes more complicated. E.g. for the linear basis function, asymmetric basis functions have to be introduced where basis functions of different splitting level overlap.

We do not have to recompute matrix **S** in every iteration. Instead the columns corresponding to the split basis functions are removed and replaced by columns corresponding to the new basis functions, see Fig. 6.4. This, together with the error heuristic expressed in matrix form results in an efficient implementation of the iterative method described in Section 6.2.1.

The visual hull restriction of matrix **S** can be performed in an efficient and accurate way. The following discussion refers to Fig. 6.3. In step 1 we extract all rows that have non-zero entries in pixel vector **p**. These represent the rays that are inside the silhouette of camera C. In step 2 we identify the columns that have zero entries only, i.e. the basis functions that do not affect the silhouette in camera C. Therefore they cannot be part of the visual hull. This step has to be performed per camera. Therefore it is necessary to keep track where the pixels in vector **p** originated.

We compute a binary vector for each camera, marking all basis functions that are potentially contained in the visual hull with one. These correspond to columns containing non-zeros in the sub-matrices extracted in step 1. The basis functions marked with one have non-zero support in the generalized cone back-projected from the silhouette of camera C.

By taking the element-wise logical AND of all binary vectors, we compute the intersection of the generalized cones of all cameras and thus the visual hull (step 3). This computation is accurate up to the discretization in the image plane, i.e. up to the pixel level. Step 4 restricts the original matrix **S** to columns corresponding to basis functions that have non-zero support in the visual hull. The resulting linear system has zero rows for some of the pixels in vector **p** that are outside the silhouette. Note that not all rows that have a zero right hand side get removed. This is because the basis functions might not be completely contained in the visual hull. Therefore the zero pixels have to be accurately included in the estimation process (step 5). Basis functions on the boundary of the visual hull are very likely to be split, so the boundary of the visual hull gets represented accurately after some iterations of the adaptive scheme. This can be seen in the middle plot of Fig. 6.13.

The splitting process cannot be performed infinitely. It is advisable to set a maximum split level. A suitable criterion is the Nyquist limit, i.e. if a basis function projects to less than two pixels in all images the splitting can be stopped. A useful number for k is the square root of the number of basis functions currently used. This choice results in a sub-exponential growth in the number of basis functions but converges faster than a constant number k.

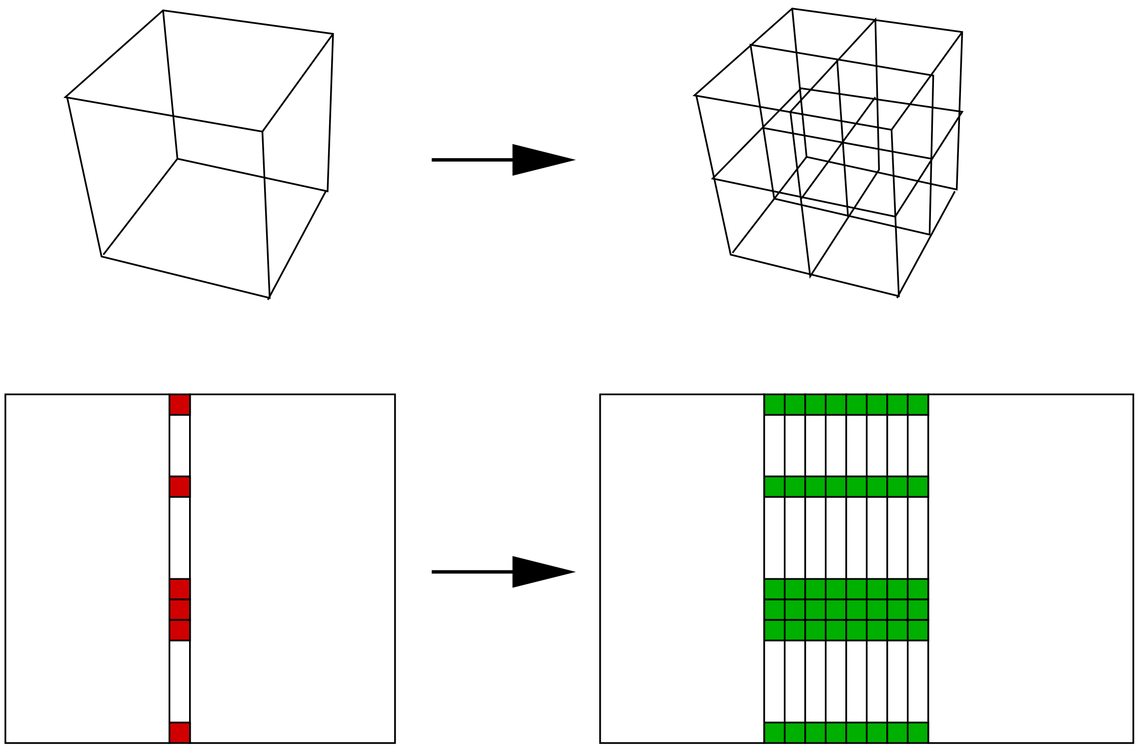


Fig. 6.4. Augmentation of matrix **S** to accommodate adaptive splitting. The column corresponding to a split basis function (red) is removed and its non-zero entries are recomputed for the smaller basis functions taking its place (green).

6.2.4 Initialization of the New Discretization

The solution of the previous iteration in the adaptive process can be used as an initial guess for the updated discretization of the domain. This ensures a faster convergence of the conjugate gradient method in each iteration of the adaptive process, but might lead to an undesirable local minimum of the solution. We initialize the coefficients of the unsplit basis functions with their value from the previous iteration and the newly introduced ones with the value of their parent in the octree data structure.

6.2.5 Implementation

The whole adaptive process can be efficiently implemented using basic matrix - vector operations. A simple indexed, unordered sparse matrix representation has been found to be suitable for the purposes of this algorithm. This allows for a straight-forward implementation and is also easily parallelizable. The memory requirements are typically only 20 to 25% of the uniform grid case while achieving comparable reconstruction accuracy. This allows for higher resolution input images and a higher resolution of the reconstructed model.

We use a minimalist octree data structure to keep track of the splitting process. For each column index of matrix **S** we store the split level from the

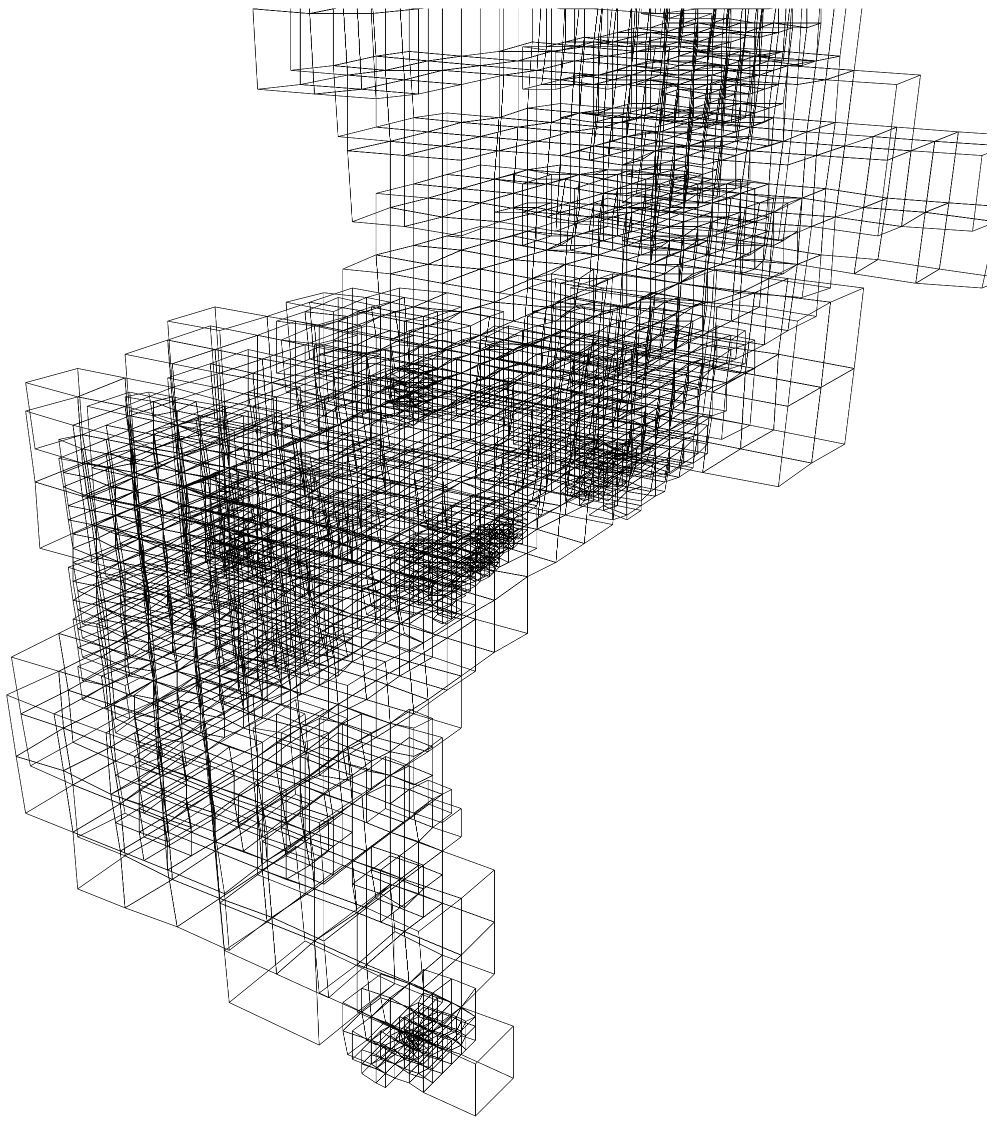


Fig. 6.5. A visual hull restricted adaptive grid (intermediate step in the iteration).

root of the tree and the index that the corresponding leaf would have in a uniformly split octree under a fixed order of traversal. This enables us to identify the positions and sizes of the newly inserted basis functions and to generate the integral values for the new columns of matrix **S** in every iteration.

To minimize the overhead introduced by re-generating the matrix values for the newly inserted columns we keep track of the cameras and pixels that are affected by a particular basis function. For each row of matrix **S** we store the camera and the pixel corresponding to the linear equation. Thus by identifying the non-zero entries of the removed column we can efficiently generate the new non-zero entries of the columns replacing the original one. This is possible

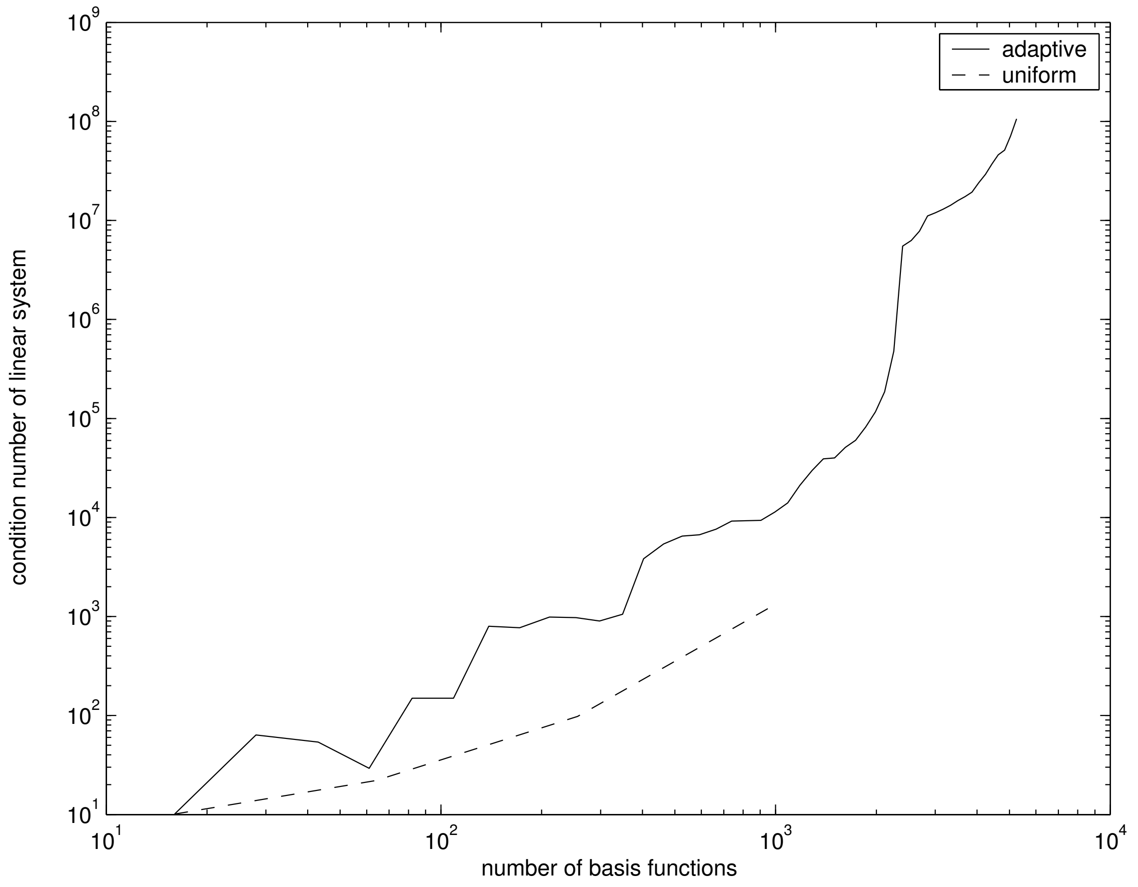


Fig. 6.6. The condition number of matrix **S** plotted against the number of basis functions in the adaptive and the uniform grid version of the reconstruction algorithm. Note that this is a double logarithmic plot.

because the refined basis functions' support is completely contained in the support of the original basis function. Fig. 6.4 shows the changes applied to matrix **S** during the refinement of the adaptive grid.

6.3 Regularization Issues

The basic adaptive reconstruction method suffers from problems which cannot intuitively be seen. It turns out that the introduction of differently sized basis functions changes the numerical conditioning of the inverse problem. Fig. 6.6 shows the condition number of matrix **S** defining the least squares problem for the algorithm based on the uniform grid and for the adaptive grid case. As can be seen from the Figure, the numerical conditioning of the problem is orders of magnitude worse for the adaptive grid than for the uniform grid case. We can also observe jumps in the graph of the adaptive grid case. These

```
Initialization:

    $\mathbf{a}_0 = 0, \quad \mathbf{r}_0 = \mathbf{p} - \mathbf{S}\mathbf{a}_0, \quad \mathbf{d}_0 = \mathbf{M}^{-1}\mathbf{S}^T\mathbf{r}_0, \quad \gamma_0 = (\mathbf{S}^T\mathbf{r}_0)^T(\mathbf{M}^{-1}\mathbf{S}^T\mathbf{r}_0), \quad k = 1$

Iteration:

    $\alpha_k = \gamma_{k-1}/||\mathbf{S}\mathbf{d}_{k-1}||_2^2$
    $\mathbf{a}_k = \mathbf{a}_{k-1} + \alpha_k\mathbf{d}_{k-1}$
    $\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_k\mathbf{S}\mathbf{d}_{k-1}$
    $\gamma_k = (\mathbf{S}^T\mathbf{r}_k)^T(\mathbf{M}^{-1}\mathbf{S}^T\mathbf{r}_k)$
    $\beta_k = \gamma_k/\gamma_{k-1}$
    $\mathbf{d}_k = \mathbf{M}^{-1}\mathbf{S}^T\mathbf{r}_k + \beta_k\mathbf{d}_{k-1}$

Termination: L-curve criterion
```

Fig. 6.7. The preconditioned CGLS algorithm.

jumps correspond to the introduction of new split levels of the basis functions into the grid.

This effect negates the positive character of the smaller degree of freedom caused by the differently sized basis functions and decreases the convergence rate of the CGLS method considerably. To solve this problem we need two ingredients. First we have to switch to a preconditioned version of the CGLS method and second we introduce explicit regularization of the linear least squares problem.

6.3.1 Preconditioned CGLS

Preconditioning is the process of modifying a linear system to improve its condition number without changing its solution. This is done by multiplying the original system of equations with an approximation to the inverse of the original matrix, i.e. for the normal equations, Eq.(5.14)

$$\mathbf{M}^{-1}\mathbf{S}^T\mathbf{S}\mathbf{a} = \mathbf{M}^{-1}\mathbf{S}^T\mathbf{b}. \tag{6.12}$$

The matrix $\mathbf{M}^{-1}$ is called a *preconditioner*. Eq. (6.12) has obviously the same solution as Eq. (5.14) as long as the inverse of $\mathbf{M}$ exists, however the condition number of the resulting linear system can be improved considerably and thus the convergence of iterative solution methods can be accelerated. The drawback is that matrix $\mathbf{M}$ must be computed and inverted. This must be possible at a low computational expense such that the improved convergence rate amortizes the additional costs. The preconditioned CGLS algorithm is given in Fig. 6.7.

We use the Jacobi preconditioner which is

$$\mathbf{M} = \mathrm{diag}(\mathbf{S}^T\mathbf{S}). \tag{6.13}$$

This simple preconditioner is already effective in increasing the convergence rate of the CGLS method to acceptable levels and can be computed efficiently directly from matrix $\mathbf{S}$..

6.3.2 Explicit Regularization

The conjugate gradient method exhibits a regularizing behavior as it behaves similarly to the truncated singular value decomposition (TSVD). The TSVD basically removes the numerical null space of the ill-posed problem and therefore smoothes the solution.

However we can also take explicit knowledge of the solution of the ill-posed problem into account and thus select an appropriate solution from the numerical null space of the linear system. This is done by modifying the original least squares problem with another quadratic, positive definite form. Ideally the solution space of this quadratic form covers the null space of the original linear system and has itself a null space in the solution space of the original problem. However since the problems considered here are too large for the null space of the original problem to be computed explicitly, the modifying quadratic form usually does not exhibit these properties and trade-offs between noise reduction and solution accuracy have to be made. Now consider the minimization problem

$$\min_{\mathbf{x}} ||(\mathbf{Ax}-\mathbf{b})||_2^2 = \min_{\mathbf{x}} (\mathbf{Ax}-\mathbf{b})^T(\mathbf{Ax}-\mathbf{b}) = \min_{\mathbf{x}} f(\mathbf{x})$$

which leads to the normal equations via

$$\nabla f = 2\mathbf{A}^T\mathbf{Ax} - 2\mathbf{A}^T\mathbf{b} = 0.$$

We can incorporate prior knowledge about the solution vector $\mathbf{x}$ by adding a second quadratic, positive definite form to the original minimization problem:

$$\begin{aligned}\min_{\mathbf{x}}(1-\alpha)||(\mathbf{Ax}-\mathbf{b})||_2^2 + \alpha||\mathbf{Rx}||_2^2 &=\\ \min_{\mathbf{x}}(1-\alpha)(\mathbf{Ax}-\mathbf{b})^T(\mathbf{Ax}-\mathbf{b}) + \alpha\mathbf{x}^T\mathbf{R}^T\mathbf{Rx} &=\\ \min_{\mathbf{x}} \hat{f}(\mathbf{x})&\end{aligned}$$

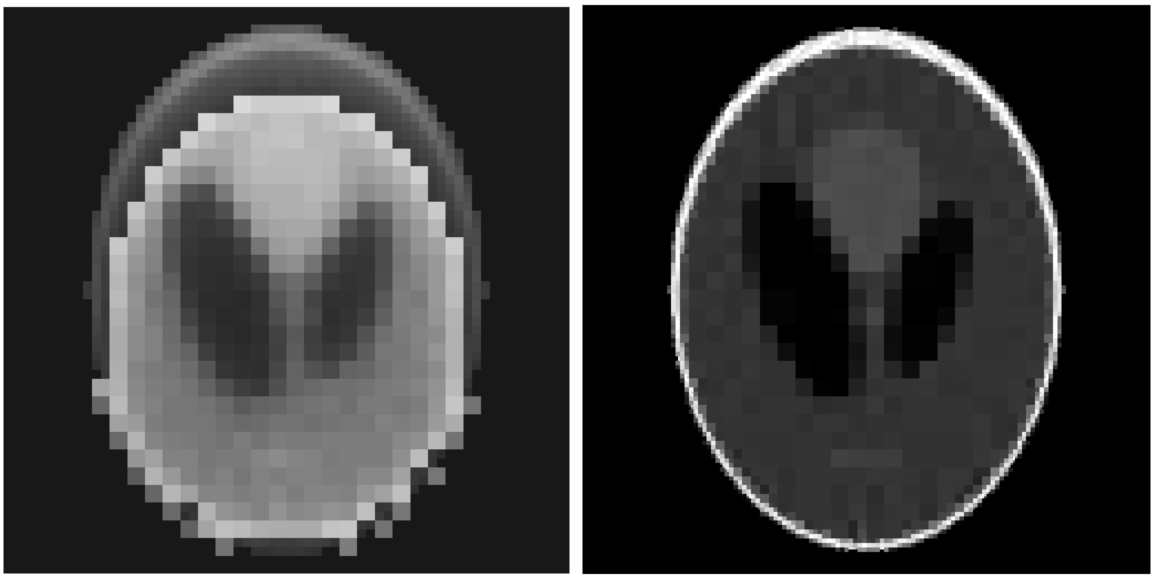


Fig. 6.8. Left: standard Tikhonov regularization leads to differently penalized coefficients, Right: adaptive Tikhonov regularization overcomes this problem.

Setting the gradient of $\hat{f}$ to zero yields the modified normal equations

$$\nabla \hat{f} = 2(1-\alpha)(\mathbf{A}^T\mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{b}) + 2\alpha\mathbf{R}^T\mathbf{R}\mathbf{x} = 0.$$

Here α allows the user to weigh prior knowledge against data fitting capabilities. We will refer to the term involving $\mathbf{A}$ as the *data term* and call the second term involving $\mathbf{R}$ the *smoothing term*. Rearranging the terms and setting $\lambda = \frac{\alpha}{1-\alpha}$ yields the regularized normal equations

$$\left[\mathbf{A}^T\mathbf{A} + \lambda\mathbf{R}^T\mathbf{R}\right]\mathbf{x} = \mathbf{A}^T\mathbf{b}. \tag{6.14}$$

The parameter λ is called regularization parameter and $\mathbf{R}$ is the regularization matrix. For use with the preconditioned CGLS method we write the regularized normal equations as

$$\begin{bmatrix}\mathbf{A}\\ \sqrt{\lambda}\mathbf{R}\end{bmatrix}^T \begin{bmatrix}\mathbf{A}\\ \sqrt{\lambda}\mathbf{R}\end{bmatrix}\mathbf{x} = \begin{bmatrix}\mathbf{A}\\ \sqrt{\lambda}\mathbf{R}\end{bmatrix}^T \begin{bmatrix}\mathbf{b}\\ \mathbf{0}\end{bmatrix}. \tag{6.15}$$

6.3.3 Tikhonov Regularization

The simplest choice for the regularization matrix is the unit matrix. The resulting regularization method is called Tikhonov regularization. The choice of $\mathbf{R} = \mathbf{I}$ yields an objective function of

$$\hat{f} = (\mathbf{A}\mathbf{x} - \mathbf{b})^T(\mathbf{A}\mathbf{x} - \mathbf{b}) + \lambda\mathbf{x}^T\mathbf{x}$$

for the least squares minimization. This shows the well known fact that Tikhonov regularization in addition to the data term minimizes the norm of the solution $\mathbf{x}$. However in the case of adaptive grid reconstructions this is not a viable way to go. Artifacts like the ones shown in Fig. 6.8 appear due to different scales in the size of the basis functions.

To understand this phenomenon we have to consider the smoothing term $\mathbf{x}^T\mathbf{x}$. Each coefficient $\mathbf{x_i}$ of the solution contributes equally to the smoothing term. On the other hand the influence of each corresponding basis function ϕ_i on the residual error is different because of their differing area of support. Therefore basis functions with a smaller support are penalized more heavily by Tikhonov regularization than basis functions with a larger support, i.e. the smoothing term can be minimized without changing the residual error too heavily. This suggests altering the regularization matrix to reflect the contribution of the differently sized basis functions. Setting the regularization matrix $\mathbf{R}$ to the diagonal matrix whose elements are

$$\mathbf{R_{ii}}^{at} = \sqrt{\frac{1}{v(\phi_i)}}, \tag{6.16}$$

where $v(\phi_i)$ measures the volume covered by the non-zero support of basis function ϕ_i yields the adaptive Tikhonov regularization matrix $\mathbf{R}^{at}$. The norm of the solution vector is weighed by the size of the support of the basis functions in the adaptive grid. This takes into account the inhomogeneous nature of the grid and results in a stable solution for the adaptive computerized tomography problem. Other choices of $\mathbf{R}$ like e.g. discretized derivative operators are possible though [69]. A result of applying the adaptive Tikhonov regularization scheme is shown in Fig. 6.8, right hand side. The artifacts introduced by standard Tikhonov regularization have been successfully removed.

6.4 Results

In this section we present results for synthetic and real-world data reconstructed with the proposed method. We study the convergence properties of the adaptive scheme with respect to the numbers of cameras used and the error measure employed. For real-world data we show results of a reconstruction of a 300 frames smoke sequence.

6.4.1 Synthetic Tests

To validate the proposed adaptive tomographic reconstruction scheme we performed tests on synthetic data. We use the Shepp-Logan phantom head model,

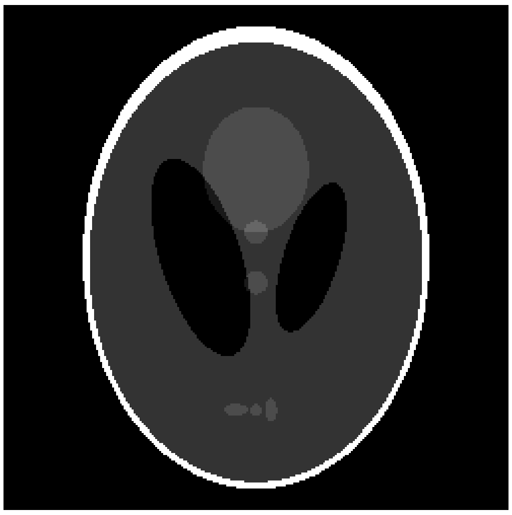


Fig. 6.9. The Shepp-Logan phantom head model used as ground truth in our synthetic tests.

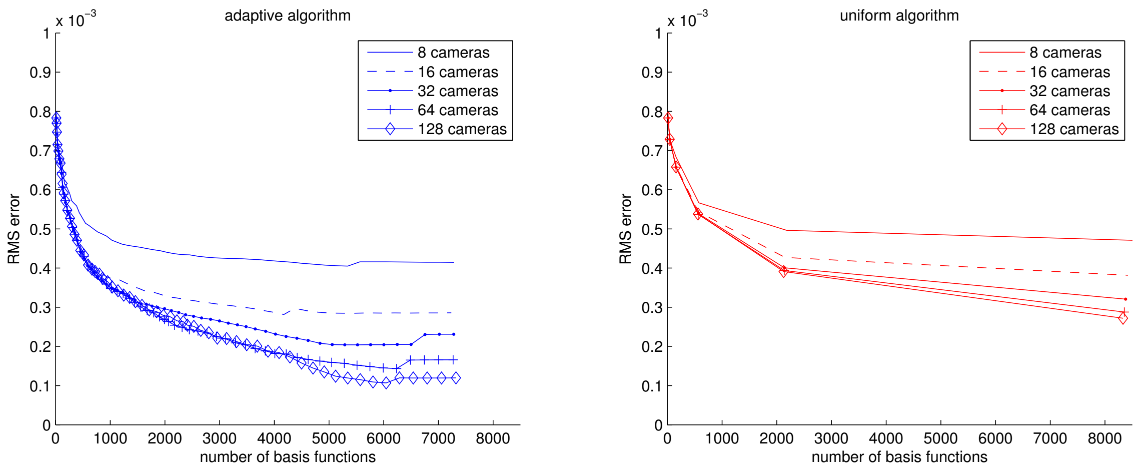


Fig. 6.10. RMS reconstruction error vs. number of basis functions for different numbers of cameras, *left:* the adaptive algorithm, *right:* using uniform discretization as in Chapter 5. The scale of the ground truth data is 1.0.

Fig. 6.9, as is common in the computerized tomography literature [96] to assess the accuracy of tomographic reconstruction methods. The model includes stylized, typical features of a human head.

We use the head model as ground truth and compute the RMS error for different numbers of simulated cameras. We analyze the convergence properties of the adaptive scheme with respect to the number of basis functions used in the reconstruction process, Fig. 6.10. The plot shows the behavior of the RMS error over 80 iterations of the adaptive scheme for different numbers of cameras. As expected the RMS error reduces with larger numbers of iterations and it converges on a lower level when using more cameras. The visual quality

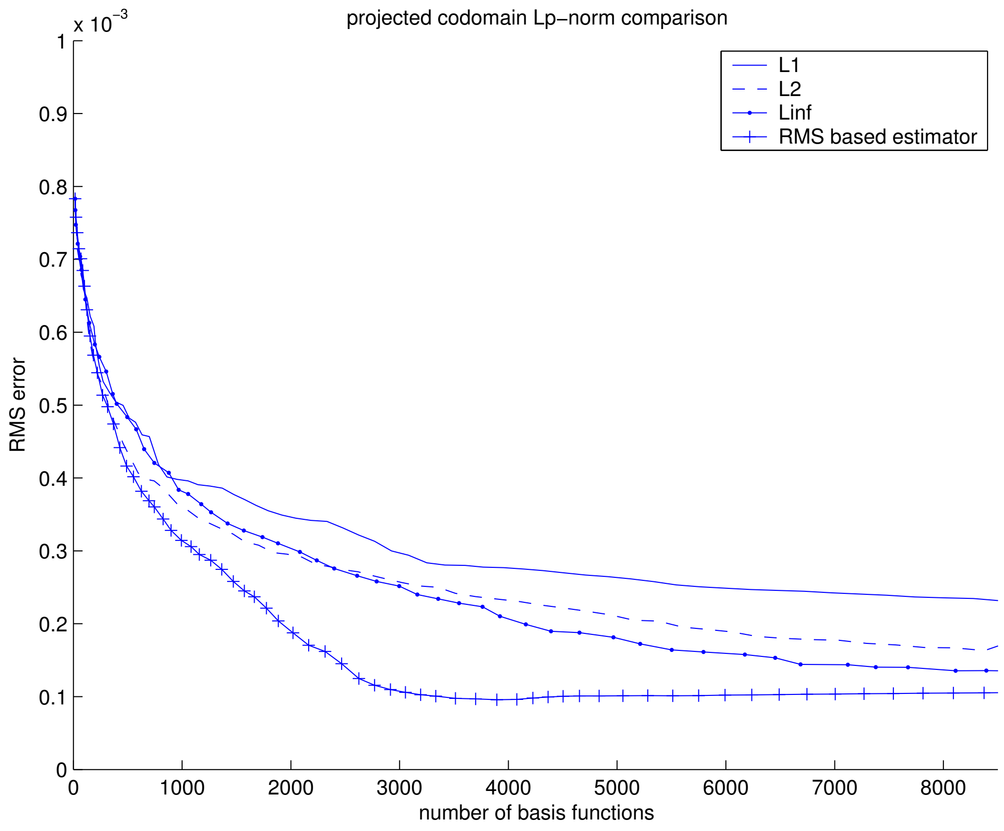


Fig. 6.11. RMS reconstruction error vs. number of basis functions for different error measures. The plot shows the convergence behavior of the adaptive algorithm using the projected codomain L_1, L_2 and L_∞ norms. The RMS error achievable by using the perfect error measure (comparison against the known solution) is included for comparison. The scale of the ground truth data is 1.0.

of the reconstructions is shown in Fig. 6.12. Compared with the uniform grid case a smaller RMS error is achieved using less basis functions.

Fig. 6.11 shows a comparison of the convergence behavior of the adaptive algorithm using different error measures for predicting the insufficiently approximated regions. We compare the projected codomain L_1, L_2 and L_∞ norms. We also include a plot for the perfect estimator, using the known solution to identify parts of the solution that need to be improved. According to Fig. 6.11, the projected codomain L_2 and L_∞ norms perform similar, where in practice we prefer the projected codomain L_2 norm because of a more homogeneous splitting of the space. However, there is still room for improvements as can be seen by the significantly lower graph of the perfect estimator. The number of cameras used for this synthetic test was 64. The results are similar for different numbers of cameras.

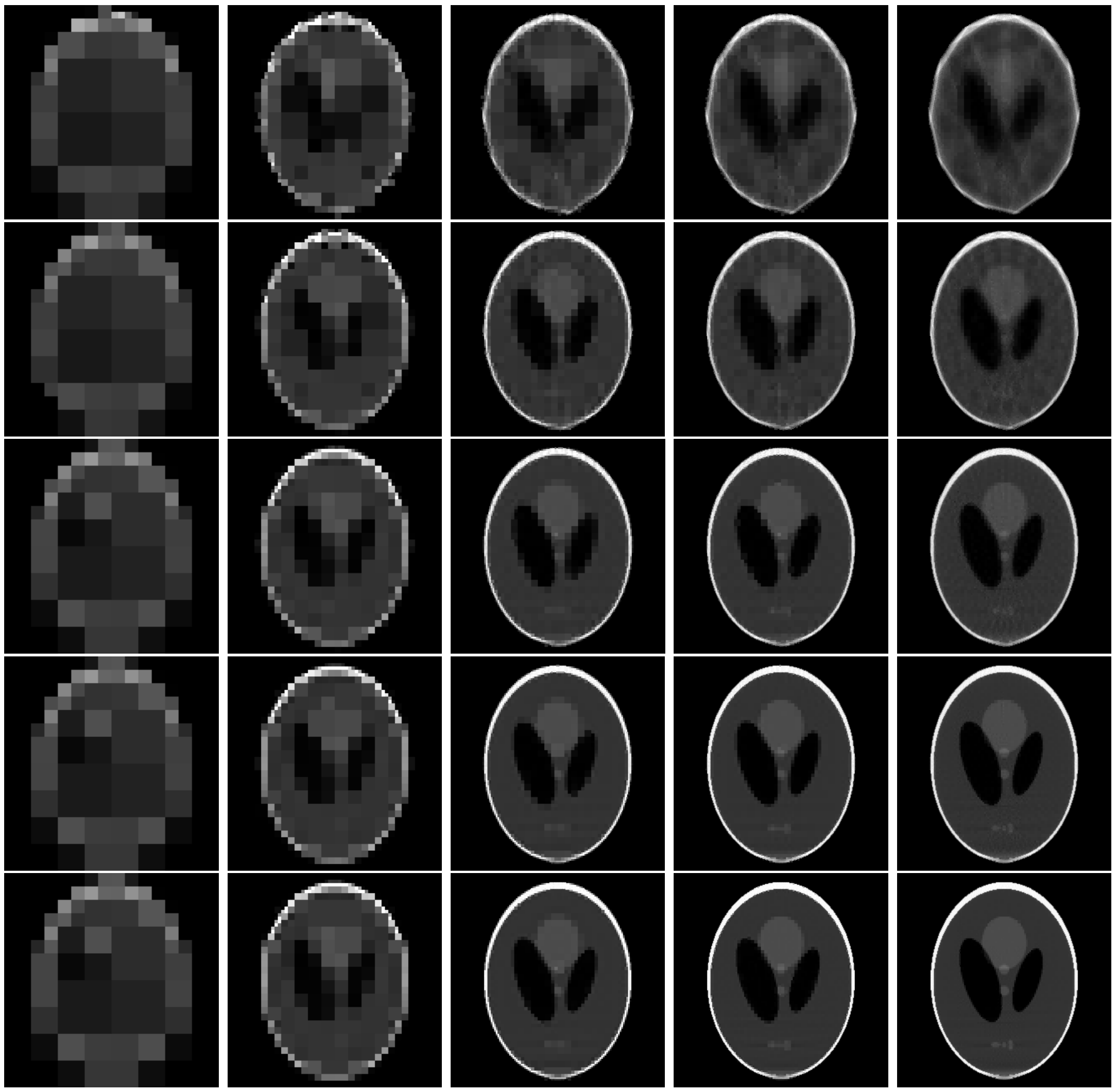


Fig. 6.12. Reconstruction results for different number of cameras (rows from top to bottom 8, 16, 32, 64, 128 cameras respectively) and different numbers of iterations of the adaptive reconstruction scheme (from left to right 4, 12, 32, 48, 64 iterations).

6.4.2 Real-World Experiments

For our real-world experiments we used again the mobile multi-camera system, Sec. 4.1.1. We perform background subtraction as a preprocessing step. Because the background subtraction is not perfect, we use an alpha-matte in step 1 of the visual hull restriction process Fig. 6.3. The matte is created using morphological operators on the thresholded foreground images.

We performed a convergence study of the adaptive algorithm in a real-world setting. The results are shown in Fig. 6.13. Since we do not have ground truth data for this case we use the residual error for the analysis. The residual error decreases as expected with the number of iterations. The decrease is not monotonic though. This is because our error measure is based on a heuristic. An interesting graph is the plot of the number of rows in matrix **S** versus the number of iterations. It shows that the basis functions adapt to the pixel perfect visual hull. A visualization of the results after different numbers of iterations and the convergence of the solution is shown in Fig. 6.14. Along with Figs. 6.16 and 6.15 it shows reconstructions we have obtained by applying our algorithm to different multi-video sequences.

6.5 Discussion

Compared to the uniform discretization case our new method offers some advantages which are reduced memory requirements allowing for an increased resolution of the reconstruction. The adaptive grid on the other hand is a disadvantage for the reconstruction of video sequences in terms of computation time. Since the adaptive grid cannot be re-used for the different frames of the video sequence and the majority of the computation time is spent in the setup of the linear system, the reconstruction times for one frame with the adaptive method are comparable to the time spent for the reconstruction of a whole video sequence in the uniform grid case.

Thus the adaptive grid tomographic technique is primarily suited for the reconstruction of single frames where high resolution of the reconstructed model is of major concern.

6.6 Summary

We have presented an adaptive algorithm for optical tomography. The algorithm is based on an octree hierarchy of piecewise constant basis functions. We propose a heuristic that enables the projection of errors in the image plane into the domain of the basis functions. This heuristic is the transfer of a norm from image space to the three-dimensional solution space. This allows us to iteratively split basis functions that cause large residual errors in the image plane. Using this algorithm we are able to reconstruct dynamic, volumetric models of flames and thin smoke. Additionally we presented an efficient scheme for the accurate computation of the visual hull. This scheme is independent of the choice of basis functions and accurate up to the pixel level. We believe that our error projection method can be used in different contexts as well where the domain and the codomain of a linear operator live in different spaces.

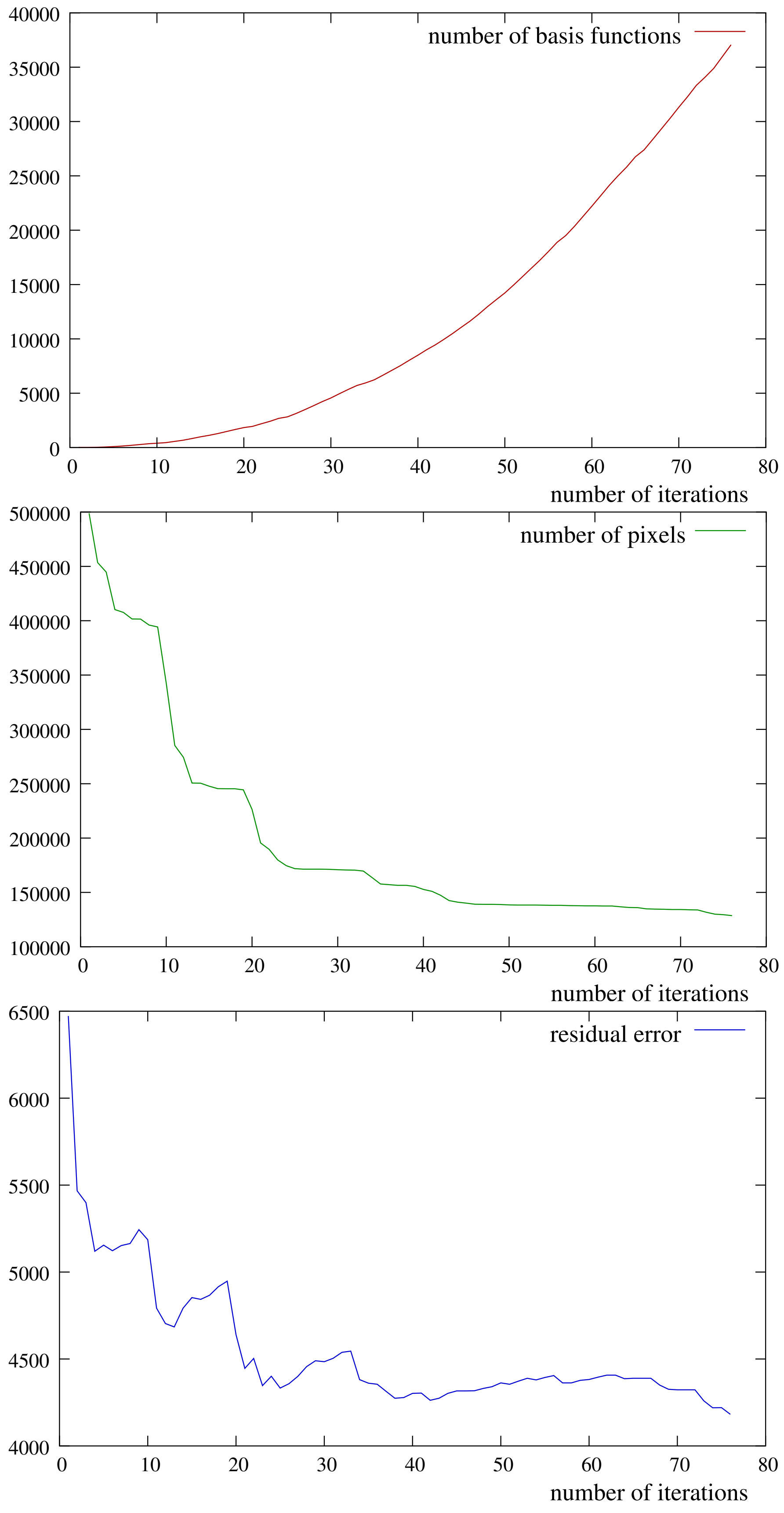


Fig. 6.13. Behavior of n_b the number of visual hull consistent basis functions (columns) in matrix $\mathbf{S}$ (top), n_p the number of pixels (rows) in matrix $\mathbf{S}$ (middle) and the residual error (bottom) versus the number of iterations of the adaptive algorithm run on real-world data.

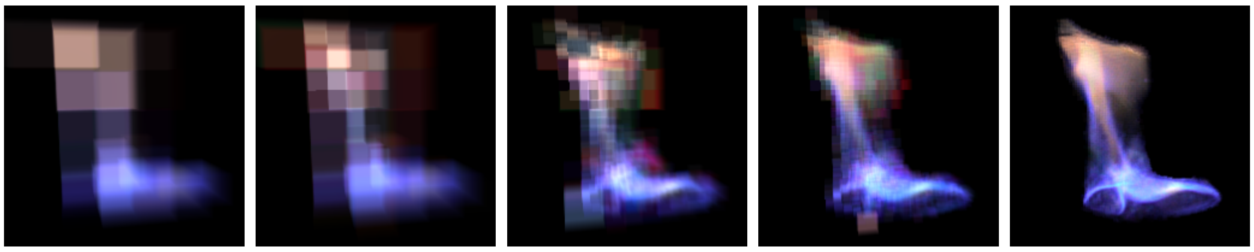


Fig. 6.14. Visualization of reconstruction results after 1, 2, 14, 28 and 100 iterations.

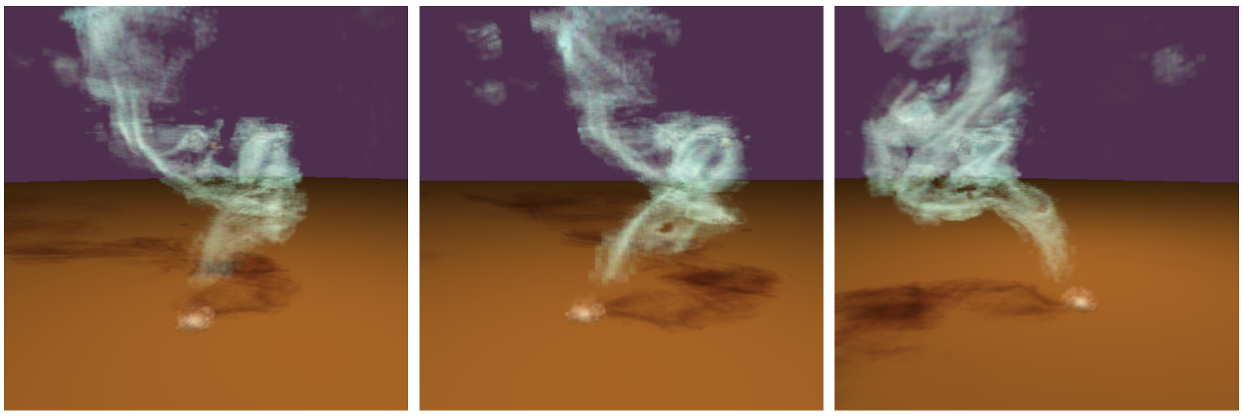


Fig. 6.15. A volumetric model of smoke rendered from different viewpoints.

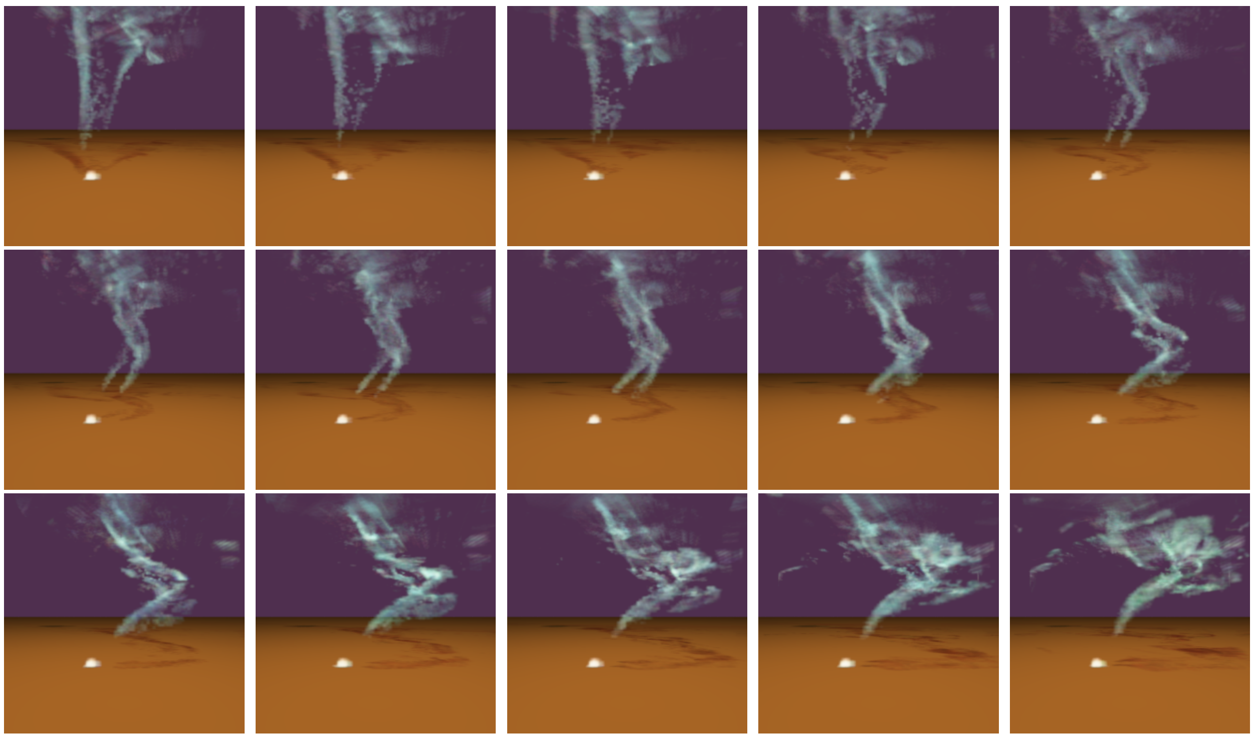


Fig. 6.16. Reconstructions of 15 consecutive frames of a smoke sequence.

Part III

Reconstruction of Refractive Phenomena

Overview

In this part of the thesis we turn our attention to phenomena that include refraction as a major effect. These are free-flowing surfaces of water in Chapter 7 and heated air flows in Chapter 8. The first phenomenon is an example of a homogenous refractive index distribution inside the object. Thus the reconstruction problem is ideally formulated as a surface reconstruction problem. We want to find the boundary between air and water, separating two different refractive indices. Differing from the other reconstruction methods presented in this thesis we use a level set formulation to find a surface minimizing a photo-consistency based error measure as introduced in Sec. 2.4. We are essentially using the a-priori information that only two refractive indices, separated by a well defined boundary, are present in the scene. This changes in Chapter 8. Our goal will be to reconstruct a spatially inhomogenous field of refractive indices. This is again realized using a tomographic reconstruction algorithm similar to Chapters 5 and 6. We introduce a new formulation of the tomographic reconstruction problem allowing for (in theory) arbitrarily curved light paths. The solution of the tomographic problem will be a vector field - the gradient of the refractive index distribution - which has to be integrated to obtain the final reconstruction. As in Part II, the reconstruction methods presented in this part of the thesis are applicable to the reconstruction of dynamic phenomena, again because a single measurement pass is sufficent to acquire the information needed.

7 Water Reconstruction

In this chapter we introduce refraction into the image formation model. We develop a method for the reconstruction of fully three-dimensional bodies of water. We deal with a single index of refraction which is required to be known. The light rays can be refracted an arbitrary number of times at well defined boundaries between the water volume and the surrounding air. This is a binary tomography problem. We approach it by modeling the interface between water and air as a dynamic surface. We develop experimental conditions that allow for a photo-consistency error measure as introduced in Sec. 2.4 to be defined. The surface is then deformed such that it minimizes the difference between synthetically generated and recorded views of the water column.

7.1 Level Set Basics

Since this chapter uses a different solution method for the tomography problem than the ones employed in the rest of the thesis we shortly introduce some basic terminology and concepts.

7.1.1 Level Set Representation

We model the water as an implicit surface separating air from water

$$u(\mathbf{x}) = 0 \qquad \Leftrightarrow \qquad \mathbf{x} \in \Sigma, \tag{7.1}$$

where $\mathbf{x} \in \mathbb{R}^3$ is part of the surface Σ if it is part of the zero level set of function u. The surface can be deformed by adding a flow field, the so called *flux* to u, see Fig. 7.1. In theory function u can have an arbitrary form, however for practical computations it is advantageous to choose u as a signed

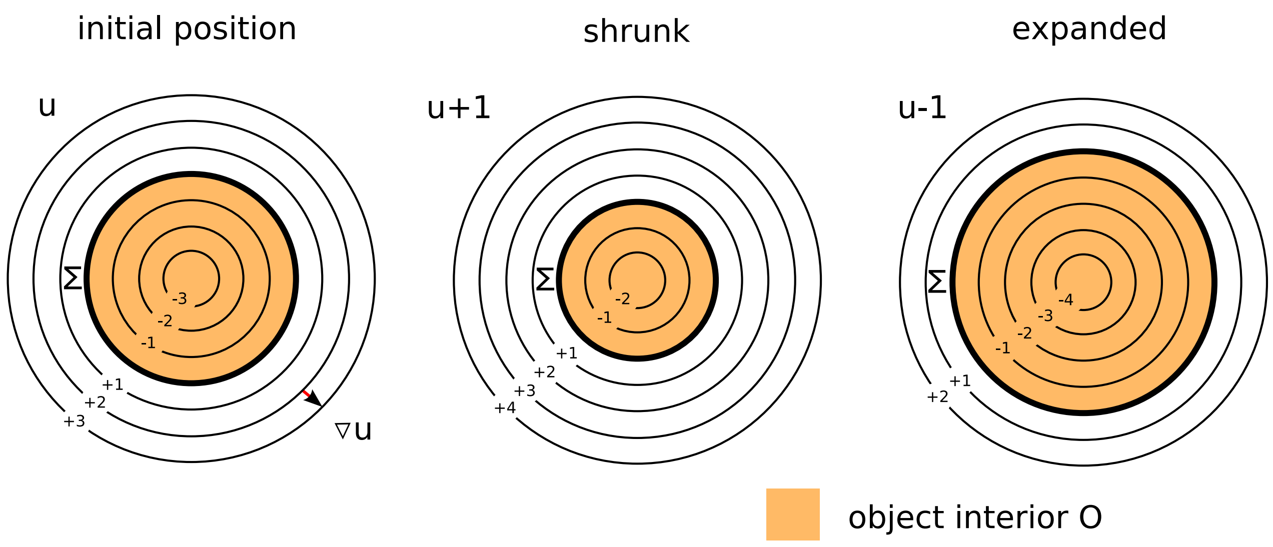


Fig. 7.1. Surfaces are represented by the zero level set of an implicit function u. The surface can be shrunk or expanded by adding and subtracting values from u. If the term being added is inhomogeneous the surface deforms (not shown in the figure).

distance function to the zero level set [148]. With this description only closed surfaces Σ can be described. For open surfaces see [179, 148]. Summarizing, function u has the following properties:

$$\begin{aligned}
u(\mathbf{x}) &= 0 \Leftrightarrow \quad \mathbf{x} \in \Sigma \\
u(\mathbf{x}) &< 0 \Leftrightarrow \quad \mathbf{x} \in \mathrm{O} \\
u(\mathbf{x}) &> 0 \Leftrightarrow \quad \mathbf{x} \notin (\mathrm{O} \bigcap \Sigma) \\
|\nabla u| &= 1 \\
\mathbf{n}_\Sigma &= \nabla u \\
d_\Sigma(\mathbf{x}) &= |u(\mathbf{x})|
\end{aligned} \tag{7.2}$$

i.e. it is negative inside the object O and positive on the outside. The surface is described by the zero-crossing of function u and its gradient points in the normal direction $\mathbf{n}_\Sigma$ of the surface everywhere. The distance $d_\Sigma(\mathbf{x})$ of a point $\mathbf{x}$ to surface Σ is given by the absolute value of the function u. This enables a very simple computation of the closest point on the surface for every point in space:

$$\mathbf{x}_\Sigma = \mathbf{x} - u \nabla u. \tag{7.3}$$

The convenience of using the signed distance function for function u comes at the price of having to adjust it to exhibit these properties every time the surface Σ is deformed, which is done via a partial differential *evolution equation*. The evolution equation is problem specific and describes the flux modifying function u such that it exhibits the properties required by the specific task. An evolution parameter τ is introduced and the flow described by the level set evolution equation is applied until convergence to a stationary solution, where the flow becomes zero and the surface stops moving. The final position of the surface is often heavily dependent on the initial level set u_0. It can easily stop moving in undesirable places. Therefore a good initial guess is very important for the convergence of the surface. Convergence properties of level set methods are still not fully understood, for a discussion see e.g. [34].

To re-initialize u to signed distance after a deformation, the evolution equation

$$\frac{du}{d\tau} = |\nabla u| - 1 \tag{7.4}$$

is solved for a stationary solution, details can be found in [148]. An initial guess for the signed distance function can be computed using the sweeping algorithm described in [62].

7.2 Error Minimization using Weighted Minimal Surfaces

A minimal surface is a surface that minimizes an error functional $\mathcal{A}(\Sigma)$ defined for a surface Σ

$$\mathcal{A}(\Sigma) = \int_{\Sigma} \Phi(\mathbf{x}, \mathbf{n}) dA. \tag{7.5}$$

Here Φ is the weight function or error measure defined on the surface. It may depend on a point $\mathbf{x}$ on the surface Σ and the normal of the surface in that point $\mathbf{n}(\mathbf{x})$. Goldluecke and Magnor [59] showed that a surface minimizing Eq. (7.5) fulfills the Euler-Lagrange equation[1]

$$< \Phi_{\mathbf{x}}, \mathbf{n} > -\mathrm{Tr}(\mathbf{S})\Phi + \mathrm{div}_{\Sigma}(\Phi_{\mathbf{n}}) = 0. \tag{7.6}$$

They present a level set evolution equation, the solution of which is a stationary point of Eq. (7.6). Defining the flux

[1] $\Phi_{\mathbf{x}}$ is the differentiation of Φ with respect to its spatial coordinates $\mathbf{x}$ and $\Phi_{\mathbf{n}}$ with respect to the normal $\mathbf{n}$ respectively. $\mathrm{Tr}(\mathbf{S})$ is the trace of the second fundamental form of the surface Σ and related to the mean curvature of the surface. div_{Σ} denotes the divergence operator for vector fields on the manifold Σ. We refer the interested reader to [58] and the references therein for a detailed discussion.

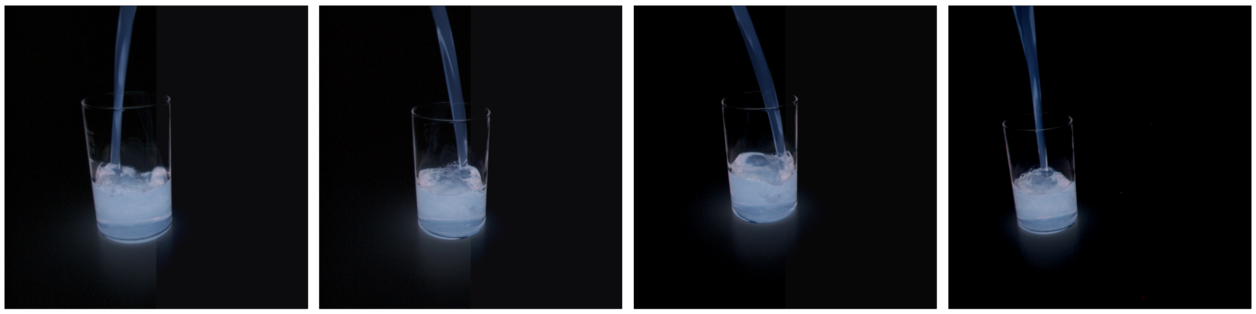


Fig. 7.2. Four of eight source images from our test video sequence. The images were taken at the same point in time.

$$\Psi = -\mathrm{div}(\Phi \mathbf{n}) + \mathrm{div}_{\Sigma}(\Phi_{\mathbf{n}}), \tag{7.7}$$

[59] write the level set evolution equation for minimizing Eq. (7.5):

$$\frac{du}{d\tau} = \Psi |\nabla u|. \tag{7.8}$$

This is a flow in the normal direction of the surface [148], optimizing Σ with respect to $\mathcal{A}(\Sigma)$. It should be noted that the integral in Eq. (7.5) is never evaluated explicitly. Evolving Eq. (7.8) to a steady state implicitly minimizes the integral by deforming surface Σ using only local computations. A disadvantage caused by this is that we must be able to evaluate Φ away from surface Σ. We will discuss this issue in Sec. 7.3.1.

7.3 Experimental Setup

In order to define an error measure for surface Σ in the context of this chapter, we use a special experimental setup which enables us to formulate a photo-consistency constraint on the surface. The error measure Φ will measure the agreement between synthetically generated views using the reconstructed surface Σ and the recorded images.

Our experimental setup consists of chemically dyed water that is recorded using a multi-camera setup. Two chemicals are mixed into the water and their chemical reaction produces light, making the water self-emissive. This effect is called chemiluminescence. Commercially available products using this effect are glow sticks. They are optimized for brightness and longevity of the chemical reaction.

Recording the self-emissive water in a dark setting and assuming constant self-emission in a well mixed solution of the chemicals allows us to measure the optical path length that a ray travels in the fluid. We ignore absorption and scattering. Some example images of our input data are shown in Fig. 7.2.

Equating the image intensity with the optical path length within the water column requires the photometric calibration of all cameras, see Sec. 4.3, such that they exhibit a linear response to incoming radiance.

7.3.1 Error Measure

Since intensity in our experimental setup corresponds to optical path length within the refractive medium we can set up a photo-consistency measure to measure the goodness of fit of a surface model Σ to the acquired data. Defining the quantity

$$r = \int_c \rho H(-u) ds \tag{7.9}$$

we have a tool to measure the optical path length within the volume enclosed by surface Σ. We make use of the Heaviside function

$$H(x) = \begin{cases} 0 & x < 0 \\ \frac{1}{2} & x = 0 \\ 1 & x > 0 \end{cases} \tag{7.10}$$

and the property that the level set function u is negative inside the enclosed volume. c is the curved ray traversing the volume. The Heaviside function selects the interior of the volume depending on the level set function u used for its description. ρ is a constant scale factor that combines camera response and emissivity value of the chemical reaction.[2] The practical computation of c and r is performed simultaneously using a description of the ray trajectory as the solution to the initial value problem

$$\frac{d\mathbf{x}}{ds} = \mathbf{d} \tag{7.11}$$

$$\frac{d\mathbf{d}}{ds} = \delta(u) f(\mathbf{d}, u) \tag{7.12}$$

$$\frac{dr}{ds} = H(-u). \tag{7.13}$$

as described in Sec. 2.3.4. The function f models Snell's Law, see Appendix A.3 for the exact definition. $\delta(u)$ is the Dirac delta function, serving as a boundary indicator such that a change in ray direction, i.e. refraction or

[2] In practice we normalize the camera images with the mean value of all pixels within the silhouettes of the water column and divide synthetically generated images by the mean value of all ray-traced pixels. The resulting values are independent of ρ.

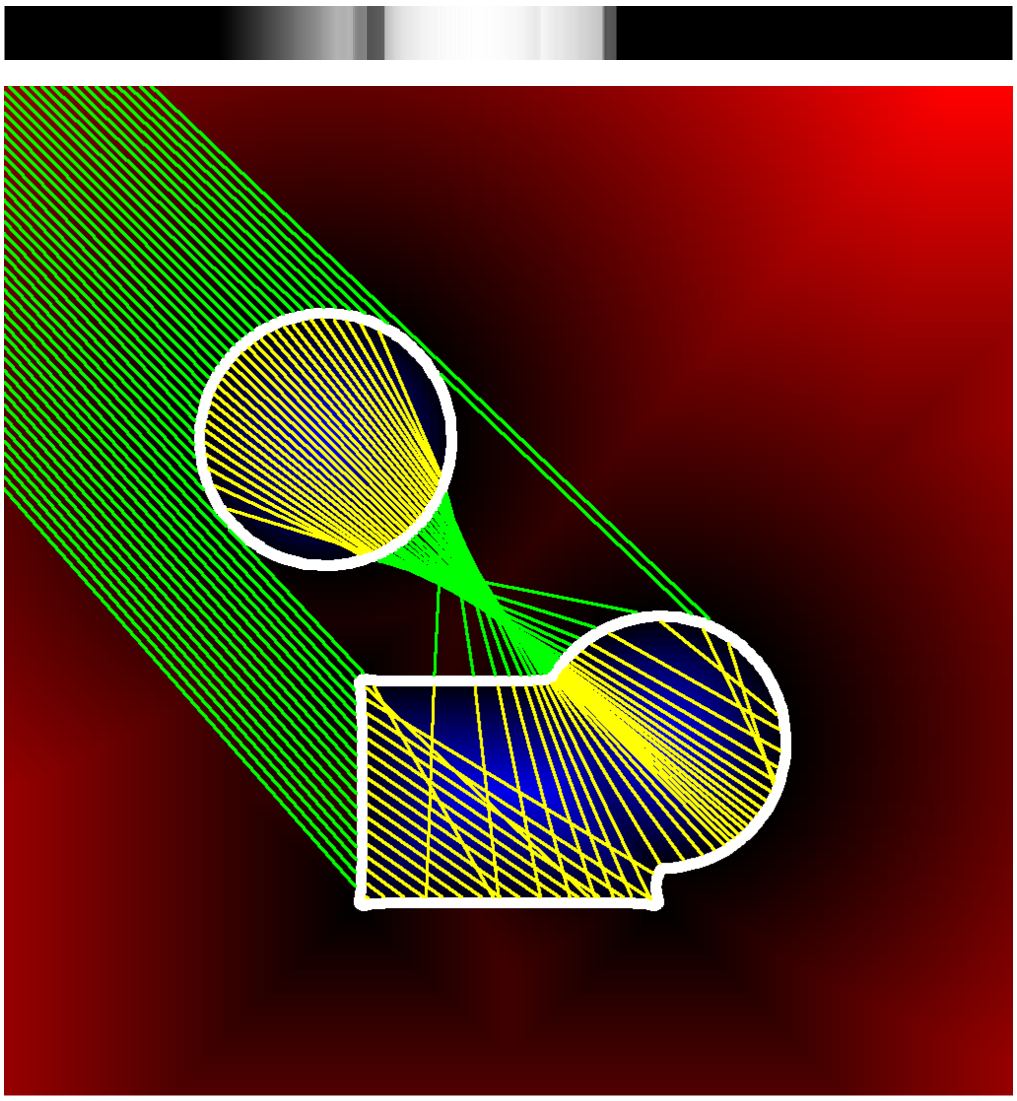


Fig. 7.3. An illustration of the complexity of image formation: The light rays enter from the top left corner and are refracted multiple times while traversing the scene. The parts of the light paths that the light travels in the water volume are shown in yellow. Since they add nothing to the optical path length travelled in the refractive volume, the last part of the rays, i.e. the last refraction and the path to the boundary of the scene volume, are omitted. On the top a 1D view of the resulting intensities is shown.

total reflection, only takes place at material boundaries. Fig. 7.3 shows some ray paths traversing a synthetic test volume consisting of two separate bodies of water. A synthetic 1D-view of ray-traced intensities r is shown on the top of Fig. 7.3.

Conceptually the error measure Φ can now be written

$$\Phi(\mathbf{x}, \mathbf{n}) = \frac{1}{k} \sum_{i=1}^{k} (I_i \circ \pi_i(\mathbf{x}) - r(\mathbf{x}, \mathbf{n}))^2, \tag{7.14}$$

where $r(\mathbf{x}, \mathbf{n})$ is the ray-traced intensity along curve c_i, computed using Eqs. (7.11)-(7.13) with initial conditions set to $r_0 = 0$, $\mathbf{x}_0 = \mathbf{x}$ and $\mathbf{d}_0 = \mathbf{d}^o$, see Fig. 7.4. We start to ray-trace curve c_i at a point $\mathbf{x}$ on a distorted surface with a normal $\mathbf{n}$ independent of the level set normal $\mathbf{n}_\Sigma$. The initial ray direction is determined using the modified normal. This is necessary to enable normal optimization of the surface Σ. $I_i \circ \pi_i(\mathbf{x})$ is the intensity recorded by camera C_i for the projection of point $\mathbf{x}$. Summation is performed over a set of cameras $C_1 \dots C_k$ that have a reasonably unobscured view of $\mathbf{x}$. In the following we discuss the details of the implementation.

Of particular difficulty is the evaluation of the error function $\Phi(\mathbf{x}, \mathbf{n})$ for a given point $\mathbf{x}$ and corresponding normal $\mathbf{n}$. The problem is that this term has to be evaluated away from the current surface Σ in order to numerically compute the derivatives in Eq. (7.8), i.e. for points that do not lie directly on the surface, and with a normal which may be different from the current surface normal. In fact, the question is what local error would arise *if the surface was distorted* such that it lies in $\mathbf{x}$ with normal $\mathbf{n}$. For this reason, ray tracing in order to evaluate the error function has to be performed for a distorted surface Σ'. The computation of $\Phi(\mathbf{x}, \mathbf{n})$ is thus performed in three steps.

In the first step, we construct the distorted surface Σ' through which rays are traced. We have to change Σ locally in a reasonably smooth manner such that the new surface passes through $\mathbf{x}$. At this moment, we do not yet care about the normal. Assume for now that $\mathbf{x}$ lies outside the volume O enclosed by Σ. The desired result can then be achieved by uniting O with a sphere B centered in the point $\mathbf{x}_\Sigma$ closest to $\mathbf{x}$ on Σ, with radius d_Σ (the distance of $\mathbf{x}$ to surface Σ), see Fig. 7.4. Vice versa, if $\mathbf{x}$ lies inside O, we can achieve the result by subtracting B from O.

The second step is to define the set of cameras $\mathcal{C} = \{C_1, \dots, C_k\}$ which contribute to the error measure. Ideally, since the medium is transparent, we would like to consider all cameras we have available. Unfortunately, this requires to find for each camera the ray passing from the camera center to $\mathbf{x}$, possibly refracted multiple times on the way. This computation definitely is too time-consuming. Instead, we only consider those cameras which have a reasonable unobscured view of $\mathbf{x}_\Sigma$ with regard to the original surface. More precisely, each camera C_i belonging to $\mathcal{C}$ must meet the following two criteria:

- The straight line segment from $\mathbf{x}_\Sigma$ to the center of projection C_i must not intersect Σ, and
- The ray starting from $\mathbf{x}$ in the refracted direction $\mathbf{d}^o$ must travel inside O in the beginning. $\mathbf{d}^o$ is computed using Snell's law, using the index of refraction of water for inside the volume, and of vacuum for outside.

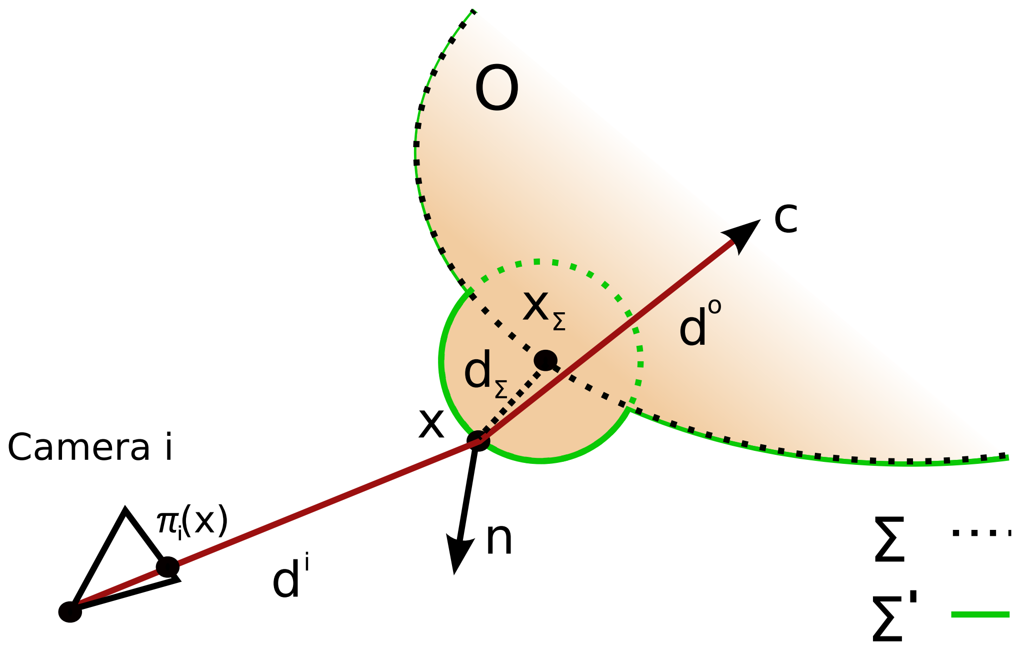


Fig. 7.4. *Modifying surface Σ to compute the partial error function ϕ_i for a single camera.*. The surface Σ is united with a sphere of radius d_Σ centered in point $\mathbf{x}_\Sigma$ closest to point $\mathbf{x}$ on the surface to yield a new surface Σ' that passes through $\mathbf{x}$. Surface Σ' permits to evaluate changes in the error measure Φ due to a deforming surface. Additionally $\mathbf{n}$ can be chosen close to the exact surface normal, in order to evaluate the derivative of Φ with respect to the normal. Ray-tracing of curve c is started in $\mathbf{x}$ with direction $\mathbf{d}^o$.

In the third step, we finally compute the photo-consistency error ϕ_i for each contributing camera C_i and average those to get the total error Φ. Each individual error is computed as follows: Let $I_i \circ \pi_i(\mathbf{x})$ be the intensity of the projection of $\mathbf{x}$ in image I_i, and $r_i(\mathbf{x}, \mathbf{n})$ be the accumulated intensity along a ray traced from $\mathbf{x}$ into the refracted direction $\mathbf{d}^o$. Then

$$\phi_i(\mathbf{x}, \mathbf{n}) \ := \ \left(I_i \circ \pi_i(\mathbf{x}) - r_i(\mathbf{x}, \mathbf{n})\right)^2 . \tag{7.15}$$

This corresponds to comparing the image intensity to the ray-traced intensity of a ray cast from the camera to $\mathbf{x}$, refracted by a surface located in $\mathbf{x}$ with normal $\mathbf{n}$. Thus, the desired normal $\mathbf{n}$ is also correctly taken into account.

Unfortunately the resulting weight function Φ is not locally dependent on $\mathbf{x}$ and $\mathbf{n}$ because the distortion of Σ changes Φ globally. The silhouette constraint introduced in the next subsection counters this shortcoming and experiments on synthetic test data suggest the feasibility of the reconstruction approach, cf. Fig. 7.6 for a qualitative analysis.

Silhouette Constraints

An additional constraint on the photo-consistency of the reconstruction result is that the projection of the reconstruction in each camera image must match the silhouette of the object to be reconstructed [105]. This constraint yields both a stopping term in our evolution equation, as well as an initial surface for the evolution in form of the visual hull [111]. To this end, let σ_i be the signed distance to the silhouette, defined in the image plane, negative inside the object silhouette.

We prohibit the projections of surface Σ to the cameras' image planes to ever shrink inside any of the recorded silhouettes. A stopping term is therefore added to the surface evolution, which grows very large if a point on the projected boundary of the surface lies inside a silhouette. When computing the visibility of a point $\mathbf{x}_\Sigma$, we can extract from the set of unobscured views $\mathcal{C}$ the set of cameras $\mathcal{B} \subset \mathcal{C}$ in which $\mathbf{x}_\Sigma$ lies on or very close to the boundary of the projection. The two criteria for camera C_i in $\mathcal{C}$ to lie in $\mathcal{B}$ as well is that

- the angle between viewing direction $\mathbf{d}^i$ from $\mathbf{x}_\Sigma$ to the center of projection C_i and the level set surface normal $\mathbf{n}_\Sigma$ must be close to ninety degrees, and
- the straight line from $\mathbf{x}_\Sigma$ in the direction $\mathbf{d}^i$ away from the camera must not intersect the surface.

Then the boundary stopping term is defined as

$$B(\mathbf{x}) := \sum_{C_i \in \mathcal{B}} \left[\exp\left(-\beta(\sigma_i \circ \pi_i)(\mathbf{x}_\Sigma)\right) - 1 \right], \tag{7.16}$$

where $\mathbf{x}_\Sigma$ is again the point closest to $\mathbf{x}$ on Σ, and $\beta > 0$ a user-defined weight, which should be set reasonably high. We use $\beta = 10$ throughout all of our tests, where the images are defined to lie in $[0,1]^2$, and the signed distance is normalized accordingly.

PDE Discretization

In order to implement the level set evolution equation, Eq. (7.8), the volume surrounding the surface Σ has to be discretized. We use a regular three-dimensional grid of evenly distributed cells with variable spatial resolution of usually 64^3 or 128^3 cells. Denoting as u_i^{xyz} the value of the discretized function u at grid point (x, y, z) at an iteration i of the evolution, a first order Euler forward step in the evolution parameter is given by

$$u_{i+1}^{xyz} = u_i^{xyz} + \Psi(u_i^{xyz})|\nabla u_i^{xyz}|\Delta\tau. \tag{7.17}$$

$|\nabla u_i^{xyz}|$ is computed using the Godunov upwind scheme[3] [148]. Since an explicit time stepping scheme is used, a step size restriction for $\Delta\tau$ has to be enforced to ensure stability. A necessary condition for stability is the so called CFL-condition[4] from the computational fluids literature:

$$\Delta\tau \leq \frac{\text{diam cell(x,y,z)}}{\max_{x,y,z}|\Psi(u_i^{xyz})||\nabla u_i^{xyz}|}. \tag{7.18}$$

It states that the evolution pseudo time step must be sufficiently small that the level sets of the discretized function u cannot cross more than one grid cell during one update.

We now turn to the discretization of the differential operator Ψ, Eq. (7.7). It consists of two parts div($\Phi\mathbf{n}$) and div$_\Sigma$($\Phi_\mathbf{n}$). To compute these the surface normals $\mathbf{n}$ must be computed. This is done by evaluating ∇u_i^{xyz} using central differences. The discretized differential operator div($\Phi\mathbf{n}$) is then computed as

$$\sum_{i=1}^{3} \frac{\Phi^{p+\Delta x_i e_i}\mathbf{n_i}^{p+\Delta x_i e_i} - \Phi^{p-\Delta x_i e_i}\mathbf{n_i}^{p-\Delta x_i e_i}}{2\Delta x_i} \tag{7.19}$$

where $p = (x, y, z)$ is the position of a grid cell, e_i a Cartesian unit vector and Δx_i the grid spacing in direction e_i. Please keep in mind that Φ has to be re-evaluated using locally distorted surfaces Σ' for every point of the discretization stencil, i.e. to evaluate div($\Phi\mathbf{n}$) it is necessary to trace six rays through six differently distorted surfaces.

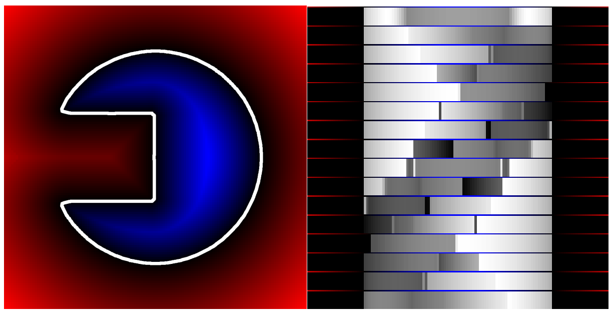
The evaluation of the normal optimization term div$_\Sigma$($\Phi_\mathbf{n}$) is more complex. It involves computing the matrix of second order derivatives $\Phi_\mathbf{xn}$:

$$\Phi_\mathbf{xn} = \begin{pmatrix} \frac{\partial^2\Phi}{\partial\mathbf{x_1}\partial\mathbf{n_1}} & \frac{\partial^2\Phi}{\partial\mathbf{x_1}\partial\mathbf{n_2}} & \frac{\partial^2\Phi}{\partial\mathbf{x_1}\partial\mathbf{n_3}} \\ \frac{\partial^2\Phi}{\partial\mathbf{x_2}\partial\mathbf{n_1}} & \frac{\partial^2\Phi}{\partial\mathbf{x_2}\partial\mathbf{n_2}} & \frac{\partial^2\Phi}{\partial\mathbf{x_2}\partial\mathbf{n_3}} \\ \frac{\partial^2\Phi}{\partial\mathbf{x_3}\partial\mathbf{n_1}} & \frac{\partial^2\Phi}{\partial\mathbf{x_3}\partial\mathbf{n_2}} & \frac{\partial^2\Phi}{\partial\mathbf{x_3}\partial\mathbf{n_3}} \end{pmatrix} \tag{7.20}$$

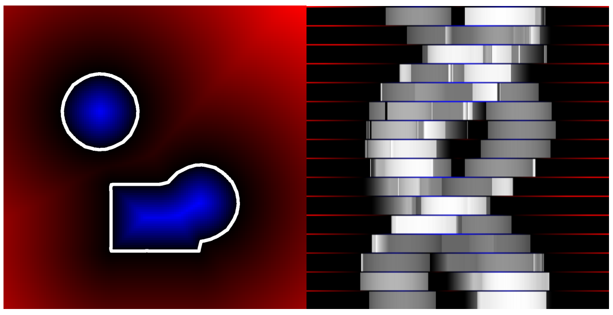
The trace Tr($\Phi_\mathbf{xn}$) of this matrix is the common divergence operator div($\Phi_\mathbf{n}$), however it has to be mapped to the tangent plane of Σ to compute the divergence operator on the manifold. This is done by choosing an arbitrary orthonormal base $\{\mathbf{t}_0, \mathbf{t}_1\}$ for the tangent space at the center point

[3] This is necessary because the computed flows cause shocks in the solution, i.e. there are places in the solution where the characteristics collide and a particular solution has to be picked to avoid multi-valuedness. This solution is referred to as *viscosity solution*. Using standard central differencing discretization schemes results in unstable algorithms.

[4] Courant-Friedrich-Levy condition



(a) The first synthetic volume (left) with 16 input views (stacked,right). Below each view is shown the signed distance transform σ of the silhouette.



(b) The second synthetic volume, also with 16 input views and signed distance transform of the silhouette.

Fig. 7.5. Synthetic test volumes and ray-traced views. Red color denotes positive values of signed distance, blue color negative values.

$\mathbf{x}$. Setting $\mathbf{U} := \Phi_{\mathbf{xn}}$, the entries of the 2×2 partial derivative matrix $\mathbf{V}$ in tangent space are then computed as

$$\mathbf{V_{ij}} = \mathbf{t}_i{}^T \mathbf{U} \mathbf{t}_j, \qquad i, j \in \{0, 1\}. \tag{7.21}$$

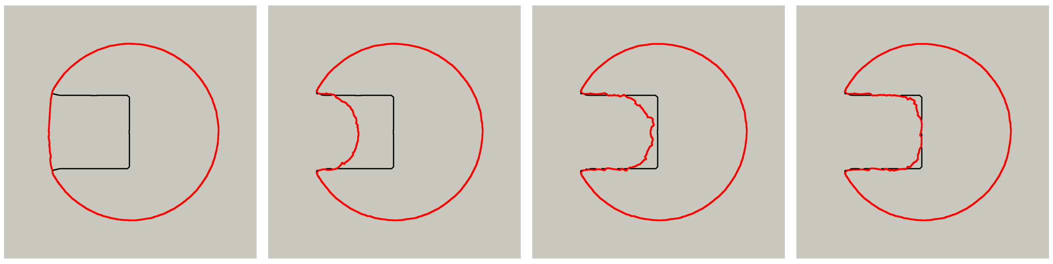
Finally $\mathrm{div}_\Sigma(\Phi_\mathbf{n})$ is computed as the trace $\mathrm{Tr}(\mathbf{V})$. The second order partial derivatives are computed similarly to Eq. (7.19) using central differences. However to compute matrix $\mathbf{U}$, 24 evaluations of Φ with 6 differently distorted surfaces become necessary.

For computational efficiency the surface is evolved according to the narrow band level set method [179], starting the evolution with the visual hull surface Σ_0 and the values u_0^{xyz} of the corresponding signed distance level set function u_0 in the centers of the grid cells. However, there are two optimization terms which are added to the values in the cells after each update step, Eq. (7.17).

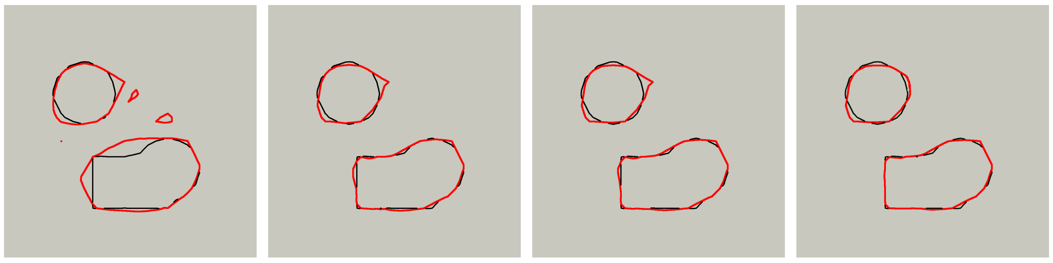
The first one is the boundary term $B(x, y, z)$. The second term is designed to speed up convergence and avoid local minima. It accelerates the shrinking process in regions where the error is excessively high. We add to u_{i+1}^{xyz} the value

$$\epsilon_1 B(x, y, z) - \epsilon_2 L_{\sigma(\Phi)}(\Phi(x, y, z) - \bar{\Phi}),$$

where $L_{\sigma(\Phi)}$ is the stable Leclerc M-estimator for the standard deviation of the error values of all cells, and $\bar{\Phi}$ the mean value of the error. $\epsilon_1, \epsilon_2 > 0$ are two user-defined weights. Good choices and their influence on convergence behavior are discussed in the next section.



(a) Convergence towards the first test volume, after 0, 100, 200, and 300 iterations.



(b) Convergence towards the second test volume, after 0, 15, 30, and 45 iterations.

Fig. 7.6. We achieved the best results using 24 input views. Several in-between stages of the iteration are shown for the two test volumes.

7.4 Results

In this section we present results for synthetic tests as well as reconstructions performed on real-world data.

Synthetic 2D Experiments

In order to verify that our surface evolution is capable of producing correct results despite the complex problem we want to solve, we first tested it on synthetic 2D data. For this purpose, we ray-traced several views of two different test volumes using the image formation model presented. The first volume is designed to test how well the algorithm can recover concavities, while the second volume is not connected and has a mixture of straight and round edges. Both test volumes and resulting 1D views are shown in Fig. 7.5.

We ran our algorithm with different numbers of input views in order to test the dependence of convergence on this critical parameter. Convergence becomes stable if eight or more cameras are available, with twelve views required in the more complex second test case. We also note that there is a quick

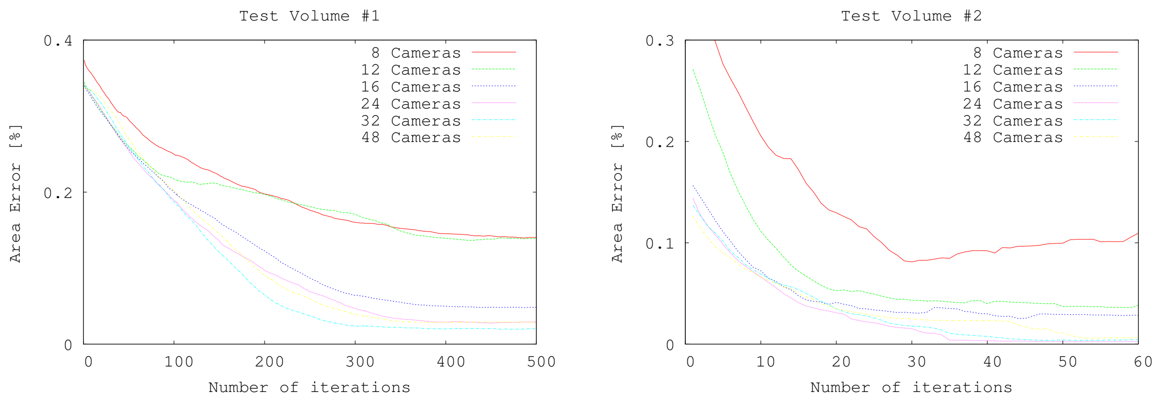


Fig. 7.7. Convergence of the results depending on the number of input views.

saturation of reconstruction quality with respect to the number of cameras because the visual hull does not improve further if more than 16 cameras are used, in accordance with earlier results [124]. In addition, the quality of the reconstruction levels out at around 24 cameras for both test volumes.

In all cases, the algorithm runs with the same parameter values of $\epsilon_1 = 0.1$ and $\epsilon_2 = 100$. These values give stable behavior against parameter changes using 24 cameras to estimate the first test volume. As a rule of thumb, there is a certain threshold value for the speedup term above which it accelerates the evolution above a stable limit, causing the surface to shrink inside the silhouettes. Too low a choice of ϵ_1 has no ill effects on stability, but slows down convergence. ϵ_2 can safely be chosen somewhere between 10 and 100 without much effect, but may cause the surface to be stuck at an undesirable spot if set too high.

Table 7.1 shows the reconstruction error, i.e., the difference between the ground truth area (Fig. 7.6) and the area enclosed by the reconstructed surface, after 200 iterations for the first test volume and different choices of ϵ_1 and ϵ_2.

Table 7.1. Error in the reconstruction of the volume shown in Fig. 7.6(a) after 200 iterations, depending on different choices of ϵ_1 and ϵ_2. An entry of "U" indicates instability and "S" indicates a stopped evolution.

		ϵ_1				
		0.01	0.1	0.5	1	5
ϵ_2	1	0.07	U	U	U	U
	10	0.05	0.04	0.06	U	U
	50	0.16	0.07	0.03	0.04	U
	100	0.04	0.05	0.04	0.06	U
	1000	S	S	S	S	0.03

Real-world Water Videos

For the real-world tests, we used the Imperx-MDC 1004 multi-video recording setup, Sec. 4.1.1.

The cameras were geometrically and photometrically calibrated. We acquired our test sequences in the dark, the chemiluminescent water being the only source of light. The images were pre-processed as discussed in Sec. 4.4. We performed a clean-up of the foreground masks using morphological operators. The reconstruction was performed on an equidistant, uniform grid of 128^3 voxels. The geometry of two of the reconstructed surface models as well as computer graphics renderings in virtual environments using modified material properties for the water volumes are shown in Fig. 7.8.

7.5 Summary

In this chapter we have presented a method for the reconstruction of flowing water surfaces. A novel recording methodology and a corresponding image formation model enable us to define a photo-consistency constraint on the reconstructed surface taking refraction into account. We utilize weighted minimal surfaces to refine the visual hull of the water using constraints based on optical path length measurements of the real surface. Real-world experiments demonstrate the suitability of our method for the reconstruction of free-flowing water.

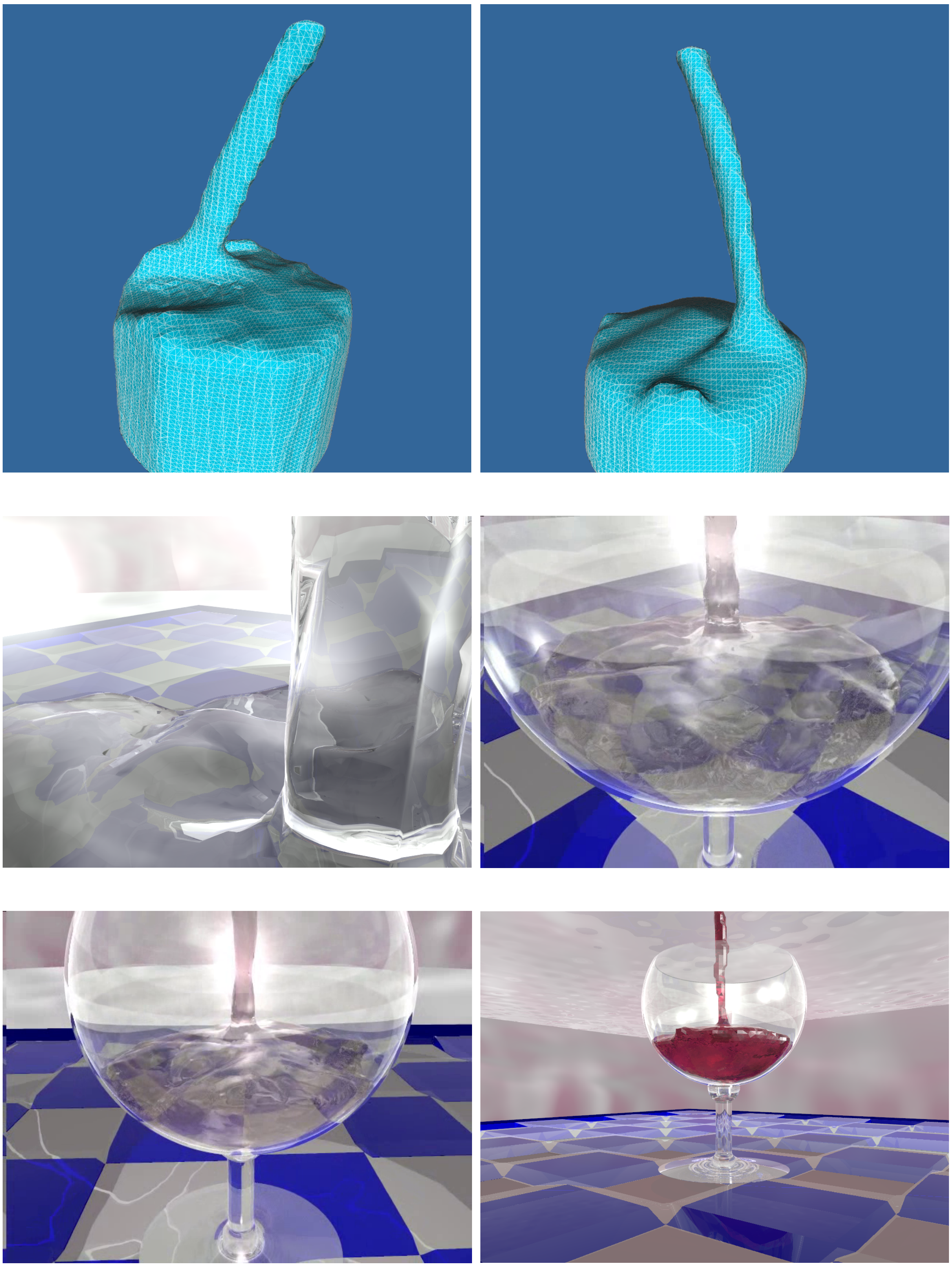


Fig. 7.8. In the top row the surface geometries of two reconstructed water columns are shown. The middle and lower rows show still frames of a flight around a reconstructed water column placed into a virtual environment. The material properties are gradually changed. All ray-tracing was performed with the *Persistence of Vision Ray Tracer*, available at `www.povray.org`.

8 Reconstruction of Continuous Refractive Index Fields

In the previous chapter we investigated a reconstruction method for refractive phenomena with a well defined surface and a single refractive index. In this chapter we develop a method to reconstruct continuously varying, three-dimensional refractive index fields. This problem can again be formulated as a tomographic reconstruction problem. The basic approach is similar to Chapter 5, however inhomogeneous refractive index fields cause the bending of light. Therefore it is necessary to introduce a description of the bent light rays for the forward image formation model. The basis of the reconstruction will be measurements of ray deflections in the image plane of the cameras. These displacements of imaged world points due to refraction can be described in terms of the gradients of the refractive index field. The gradient field is reconstructed using the tomographic approach described in Chapter 5. Afterwards an integration step is performed to recover the refractive indices. The approach presented in this chapter can be used to recover refractive index fields of comparatively low maximum refractive index magnitude as found in heated air flows or fluid mixtures.

8.1 Overview

Our method for capturing dynamic, spatially-varying refractive index fields consists of two primary components: the 2D imaging of ray deflections due to a 3D refractive index field, and the tomographic reconstruction of that 3D field from a number of deflection images captured from different positions.

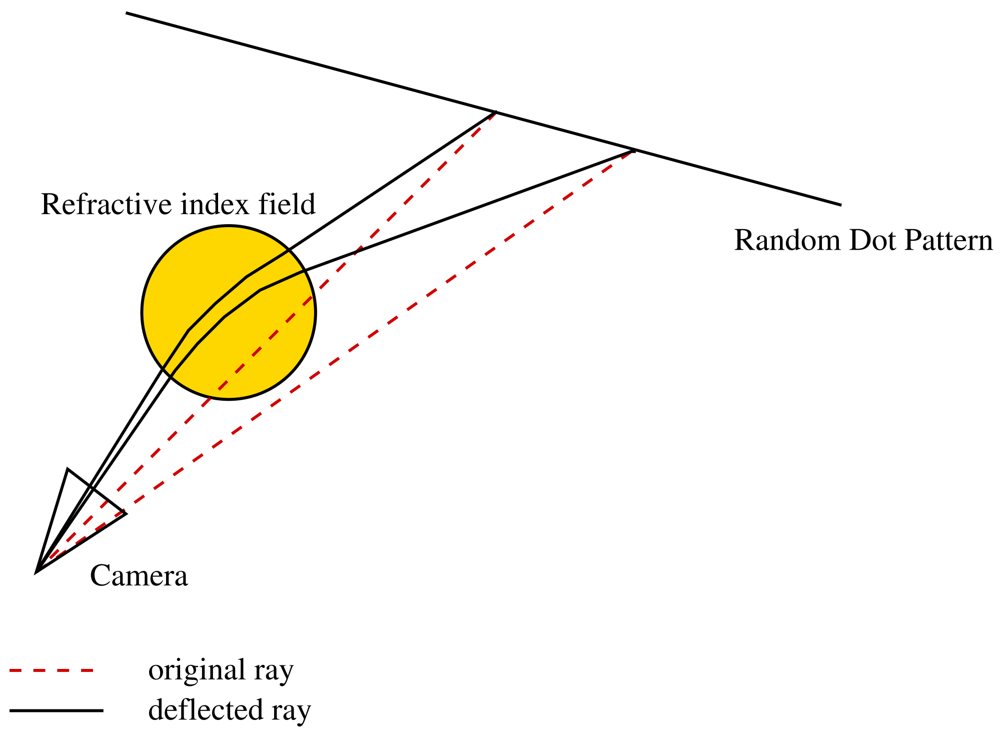


Fig. 8.1. Principle of the BOS deflection sensor: A plane with a high-frequency dot pattern is placed behind the scene of interest and an image is recorded without the object (dashed red lines). Then the inhomogeneous refractive index field is inserted between the camera and the background plane. Another image is taken and the deflection of the light rays in the image plane is computed using optical flow.

8.1.1 2D Deflection Sensing

We use a Background Oriented Schlieren (BOS) imaging setup [39, 166, 132, 44], using digital video cameras observing a high-frequency background through the refractive index field under investigation. The distortions caused by the refraction (Figure 8.1) are captured with an optical flow algorithm, which can be directly used to visualize these distortions.

Our contribution to BOS imaging is the development of a robust optical flow method that finds matches in the background pattern even if the refraction is strong enough to locally cause large isotropic and anisotropic scaling of the background pattern. It should be noted, however, that there are limits to what is possible with optical flow methods. If the refractive index differences are strong enough to cause total internal reflection or drastic changes in the focus of the optical setup, then optical flow methods may not be the best choice. Our method is therefore not well suited for recovering solids made of glass or other transparent materials. It may, however, be possible to use our method if such solids are immersed in fluids of comparable refractive index, similar to the work of Trifonov et al. [197], but without the need for a very precise match of refractive indices. We leave this application for future work.

8.1.2 3D Tomographic Reconstruction

The reconstruction is based on a set of deflection images taken from different viewpoints. The existing methods in the literature are, without exception, based on the *paraxial approximation* [47, 203], i.e. the assumption that the deflections do not cause significant changes in the ray path through the reconstruction volume, even though the deflection angle is large enough to be measured. Under this approximation, consider a camera ray along the z-axis. We can write the angular deflection in the horizontal (i.e. x) direction as a line integral of differential horizontal changes in index of refraction along the ray

$$\phi_x = \frac{1}{n_0} \int \frac{\partial n}{\partial x} dz, \tag{8.1}$$

where n_0 is the refractive index of the surrounding environment. A similar equation holds for the vertical deflection angle ϕ_y.

Under this paraxial view, the pixel measurements correspond directly to line integrals of refractive index gradients. Consequently, a volume of refractive index gradients can be reconstructed with standard tomographic methods, such as Fourier slice reconstruction or algebraic reconstruction (ART) [96]. Finally, this gradient volume can be integrated into a refractive index field by solving a Poisson equation. For more detail on this approach, please refer to the work by Faris and Byer [47].

Unfortunately, this method breaks down if inhomogeneities in the refractive index field are strong enough to cause ray deflections comparable to or larger than the voxel spacing used in the reconstruction. For this reason, we derive a novel theory for tomographic reconstruction that does not neglect changes in the geometry of the ray path. This theory is derived from the Eikonal equation and gives rise to an efficient reconstruction algorithm.

8.2 Background Oriented Schlieren Imaging

As mentioned before, the recently developed Background Oriented Schlieren technique [132, 166, 44, 203] measures the per-pixel ray deflection in the image plane caused by refraction in a volume by observing a high-frequency background through that volume, and computing the optical flow relative to an undistorted image (Figure 8.2). This method represents a significant reduction in cost and complexity compared to traditional Schlieren imaging.

However, some challenges remain, especially if the object to be measured exhibits comparatively large differences in refractive index. In this case, the background distortions become too severe for simple optical flow algorithms

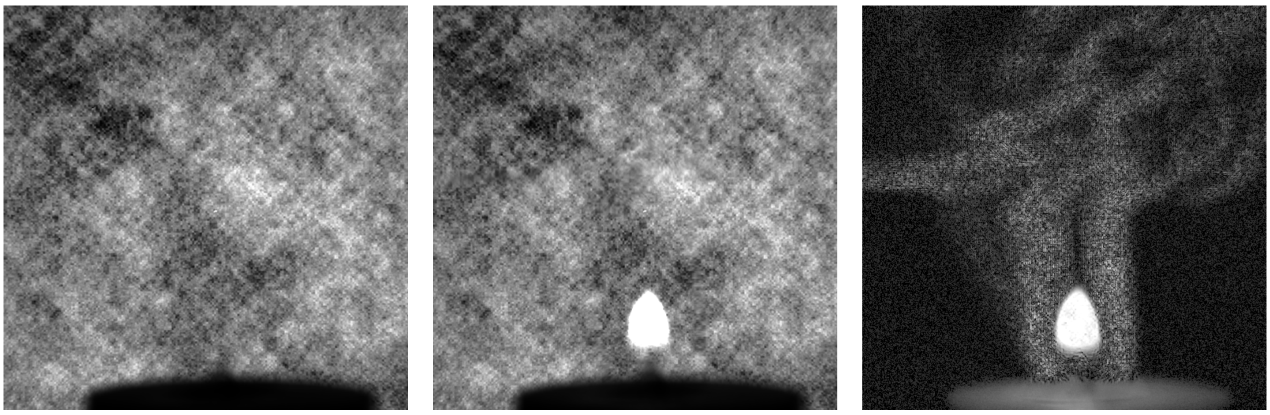


Fig. 8.2. Left: reference image, middle: distorted image, right: absolute difference (contrast enhanced). The candle plume is being blown from the side with a can of compressed air.

to find matching regions. One challenge is that the frequency characteristics of the random dot patterns used in most BOS implementations [166] degenerate quickly with even uniform scaling: magnification of such patterns results in larger regions of uniform black or white color, while shrinking quickly results in a uniform medium grey color. In both cases, it is difficult for the optical flow algorithm to pick up matching regions. For this reason, we use Wavelet Noise [37] as our background pattern, since it contains details in a wide range of frequency bands, and therefore degrades much more gracefully.

Our optical flow algorithm is designed to use this high-frequency information to find matches under distortions. We use window-based normalized cross-correlation to compute the flow, because it can be adapted to handle these distortions by searching for matches in isotropically or anisotropically scaled versions of the images. The basic optical flow is computed using spatial convolution on a per-window basis using the standard normalized cross-correlation formula

$$C(m,n) = \frac{\sum_{i,j=1}^{N} \left(f_{i,j} - \bar{f}\right)\left(g_{i-m,j-n} - \bar{g}\right)}{\sqrt{\sum_{i,j=1}^{N} \left(f_{i,j} - \bar{f}\right)^2 \sum_{i,j=1}^{N} \left(g_{i,j} - \bar{g}\right)^2}}, \tag{8.2}$$

where $f_{i,j}$ and $g_{i,j}$ are pixel values from the distorted and the background image, respectively, and $\bar{f}$ and $\bar{g}$ are the mean intensities over the comparison window.

The cross-correlation results in a matrix $C(m,n)$ of correlation scores for each translation $m,n \in [-(N-1)\dots N-1]$. In the neighborhood of the maximum of these correlation scores we fit 3-point Gaussians in the horizontal and vertical dimensions to the scores, which helps us locate the match with sub-pixel accuracy. We also compute a signal-to-noise ratio for each window as the ratio between the peak correlation value and the average correlation score of all other pixels. We use this ratio as a reliability metric for filtering in a post-processing stage.

This basic algorithm is adapted to handle significant distortions by iterative refinement. The optical flow field estimated in one iteration is used to determine an affine transformation for each neighborhood. In the next iteration, each extracted window is warped with the corresponding affine transformation before computing the correlation scores. In addition, we adjust the window size from iteration to iteration, starting with a large window size for robustness, and ending with a small window size that better captures detail in the data.

The final vector field is improved by filtering. Spurious vectors that differ by more than a fixed threshold from the global mean or their local median vectors are removed and filled in by linear interpolation. Bilateral filtering is used to smooth the resultant field while preserving the reasonably sharp boundaries that do occur in practice. A Gaussian spatial smoothing kernel is modulated by another Gaussian in the vector magnitude range. The width of this second Gaussian is set on a per-vector basis according to the signal-to-noise ratio, so that highly reliable vectors are not smoothed over, while poor quality ones have their neighbors weighted heavily.

All stages of the optical flow algorithm are designed such that they can be implemented on a GPU. Processing times are on the order of a few minutes per frame for multiple iterations on 512×512 images with window sizes up to 64×64. Figure 8.3 shows the optical flow for a plume of hot air and a 75 mm lens computed in this fashion.

The optical flow fields recovered with BOS can be used directly in rendering, for example to distort camera rays in environment matting applications, or for deflecting light rays to compute caustics (see Section 8.5.2).

8.2.1 Computing Three-Dimensional Deflection Directions

The tomographic reconstruction algorithm described in the following section requires three-dimensional deflection directions rather than optical flow vectors as its input, see Fig. 8.4. We therefore have to convert the 2D vectors obtained with BOS into estimates of world-space ray directions. In a calibrated setup with known spacing between object and background pattern, these directions could be computed from the optical flow and the exact location of the point where the deflected ray exits the volume under consideration. Unfortunately, the latter information is not readily available. A simple estimate used in the literature is to approximate the precise location with the point where the original (undistorted) camera ray would exit the volume (Figure 8.4). This

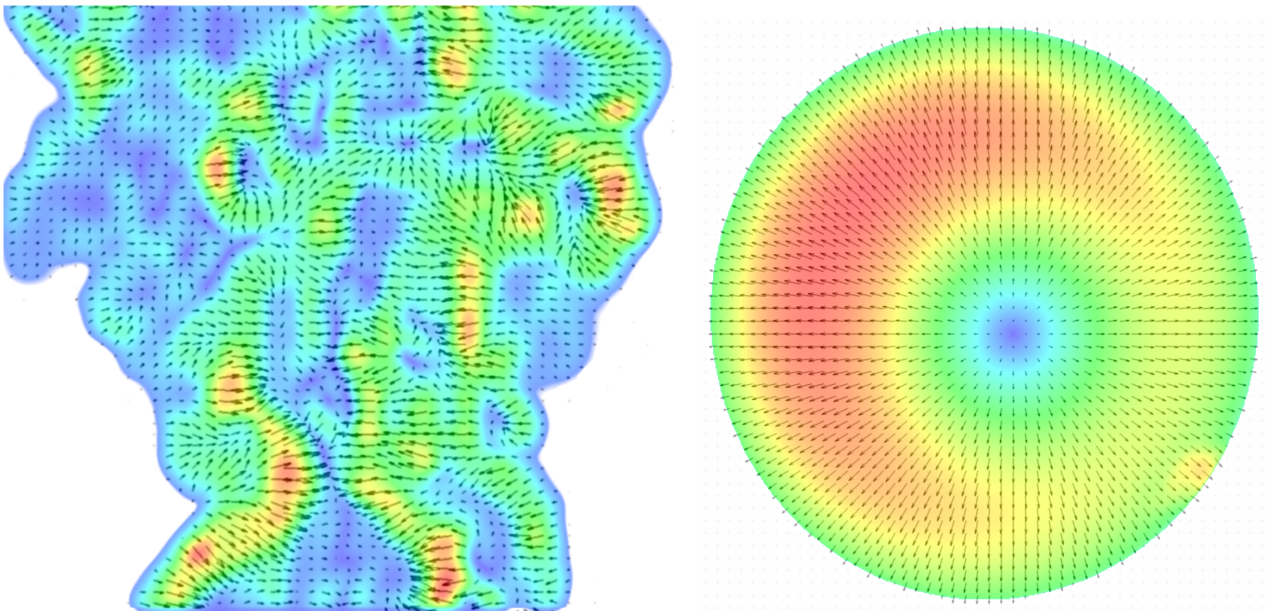


Fig. 8.3. Results from the BOS algorithm. Left: deflection field caused by a plume of hot air. Right: extreme ray deflections created by a (chipped) 75 *mm* lens, viewed from slightly off-axis.

approximation is valid if the object diameter is significantly smaller than the distance between object and background, which is the case for all our measurements.

8.3 Tomographic Reconstruction

In the following, we discuss the tomographic reconstruction of a 3D refractive index field from the three-dimensional deflected ray directions measured with the BOS algorithm and converted as described in the previous section. We first derive the theory for a tomographic reconstruction of the gradient field that does not require the use of the paraxial approximation explained in Section 8.1. We then describe a practical implementation of this theory (Section 8.3.2), and finally describe how to integrate the gradient field to obtain the refractive index field (Section 8.3.3).

8.3.1 Gradient Field Tomography

The derivation of our reconstruction method starts from the Eikonal equation

$$(\nabla S)^2 = n^2, \tag{8.3}$$

which can itself be derived from Fermat's principle of least time or alternatively from the Maxwell equations. In this equation, S describes the time it

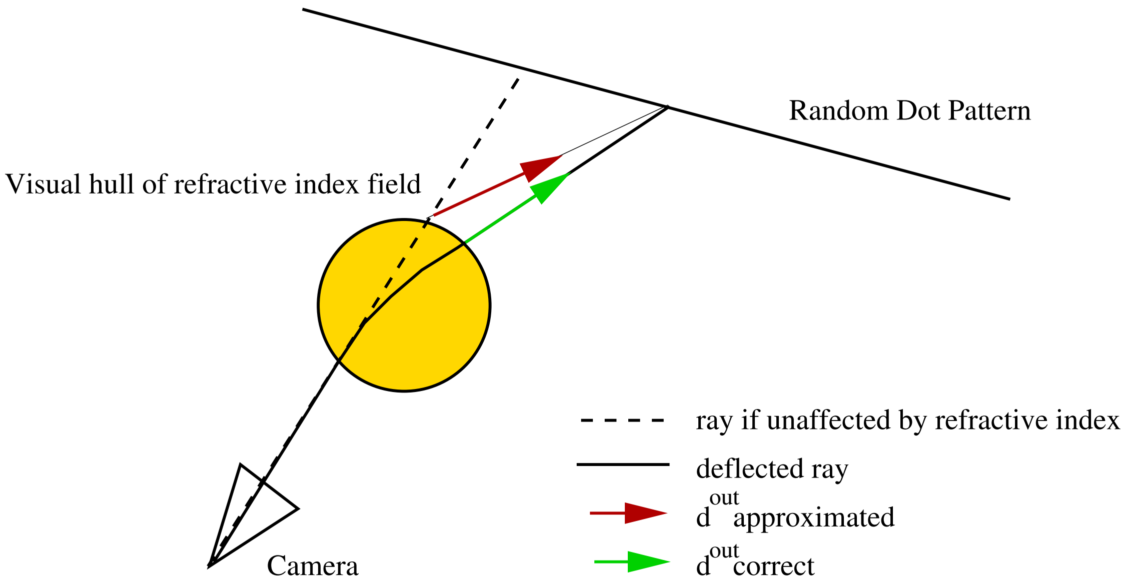


Fig. 8.4. The deflection direction can only be measured with two planes behind the object. An approximation is the direction from the intersection of the undeflected ray with the visual hull of the object to the position on the background plane that was measured. This approximation is valid as long as the extent of the object is considerably smaller than the distance to the background plane.

takes for light to arrive at a particular point in space, and n is the refractive index as before. Unfortunately S is not a function since it is multi-valued: light can reach a point in space via multiple paths of varying length, causing the solution to branch at various loci in space. These places are known as caustics. To solve Eq. (8.3) uniquely, the so-called viscosity solution is usually considered. This solution of the Eikonal equation describes the time of *first arrival* via any path at a particular point in space. The solution of Eq. (8.3) depends on the initial and boundary conditions that are used to set the positions of light sources, or, via Helmholtz reciprocity, the location of cameras.

Since iso-surfaces of S describe regions of constant time of first arrival, and light rays are described by the path of least time, light 'particles' travel normal to the iso-surfaces of S, along ∇S. We therefore have

$$n\frac{d\mathbf{x}}{ds} = \nabla S, \tag{8.4}$$

where the factor n enters the equation because ∇S is not normalized. From Eq. (8.3) it is clear that the magnitude of ∇S is n, the reciprocal of the relative speed of light in the medium.

We do not have to solve Eq. (8.3) explicitly for S and then integrate using Eq. (8.4). Instead, Eqs. (8.3) and (8.4) can be combined to obtain a differential equation for the particle position in terms of the refractive index n. For this we take the gradient of Eq. (8.3):

$$\nabla\left(\nabla S \cdot \nabla S\right) = \nabla\left(n \cdot n\right) \tag{8.5}$$

end thus

$$2\left(\nabla \cdot \nabla S\right)\nabla S = 2n\ \nabla n. \tag{8.6}$$

Inserting Eq. (8.4) together with $\nabla \cdot \frac{d\mathbf{x}}{ds} = \frac{d}{ds}$ yields

$$\frac{d}{ds}\left(n\frac{d\mathbf{x}}{ds}\right) = \nabla n. \tag{8.7}$$

Eq. (8.7) is known as the *ray equation of geometric optics*. It is a second order ordinary differential equation that can be written as a set of first order ordinary differential equations. As in Chapter 7, this is an application of our ODE ray-tracing framework, introduced in Sect. 2.3.4.

Starting from Eq. (8.7) and setting

$$n\frac{d\mathbf{x}}{ds} = \mathbf{d}, \tag{8.8}$$

we obtain

$$\frac{d\mathbf{d}}{ds} = \nabla n. \tag{8.9}$$

Observing that

$$\frac{dn}{ds} = \frac{dn}{d\mathbf{x}} \cdot \frac{d\mathbf{x}}{ds} = \nabla n \cdot \frac{d\mathbf{x}}{ds}, \tag{8.10}$$

we find that Eqs. (8.8) - (8.10) define a coupled system of ordinary differential equations for the path of a bent ray in a medium of non-uniform refractive index in terms of ∇n. Eq. (8.10) is another example of information that can be integrated along a curved ray using a system of ordinary differential equations.

We integrate Eq. (8.9) to obtain an equation that relates the unknown gradient of the refractive index field to our measurements, i.e. the outgoing ray directions

$$\mathbf{d}^{out} = \int_c \nabla n\ ds + \mathbf{d}^{in}. \tag{8.11}$$

Eq. (8.11) forms the basis of our reconstruction scheme. Similar to Chapters 5 and 6 we discretize the unknown vector function ∇n using a set of basis functions ϕ_i with unknown coefficient vectors $\mathbf{n}_i$,

$$\int_c \sum_i \mathbf{n}_i \phi_i \; ds = \sum_i \mathbf{n}_i \int_c \phi_i \; ds = \mathbf{S}\mathbf{n}_i = \mathbf{d}^{out} - \mathbf{d}^{in}. \tag{8.12}$$

Solving this linear system of equations for the coefficient vectors $\mathbf{n}_i$ allows us to recover the gradient of the refractive index field ∇n. Observe that Eq. (8.12) is vector valued and describes three linear systems with three sets of coefficients, one for each coordinate of the gradient field.

Since the curved rays c are initially unknown, we develop an iterative solution that updates the ray geometry along with the volume estimate as described in the following.

8.3.2 Reconstruction Algorithm

Our reconstruction algorithm for the gradient of the refractive index field thus takes the following form:

1. compute curved rays c, Eq. (8.7),
2. set up linear system, Eq. (8.12),
3. solve unconstrained linear system for ∇n,
4. until convergence go to step 1.

The initial guess for the gradient field is $\nabla n = 0$, i.e. we initially start from the straight ray geometry also assumed in the paraxial approximation, cf. Eq. (8.9). The rays are computed by a discretization of Eqs. (8.8) - (8.10). The ray integrals can be performed using any standard ODE integration scheme like the Runge-Kutta family. After recomputing the curved rays for the current volume estimate, we can also recompute the estimate of three-dimensional deflection directions from the optical flow data (Section 8.2.1).

The structure of the linear systems, Eq. (8.12), is very similar to the one for the fire and smoke reconstruction case, cf. Eq. (6.3):

$$\begin{pmatrix} \mathbf{d_1}^{out} - \mathbf{d_1}^{in} \\ \mathbf{d_2}^{out} - \mathbf{d_2}^{in} \\ \vdots \\ \mathbf{d_{n_m}}^{out} - \mathbf{d_{n_m}}^{in} \end{pmatrix}^j = \begin{pmatrix} \int_{c_1} \phi_1 ds & \dots \int_{c_1} \phi_{n_b} ds \\ \int_{c_2} \phi_1 ds & \dots \int_{c_2} \phi_{n_b} ds \\ \vdots & \vdots \;\; \vdots \\ \int_{c_{n_m}} \phi_1 ds & \dots \int_{c_{n_m}} \phi_{n_b} ds \end{pmatrix} \mathbf{n}^j. \tag{8.13}$$

The index j refers to the coordinates x, y and z. We have one equation for every measurement $\mathbf{d}^{out}$; $\mathbf{d}^{in}$ is obtained from the camera calibration by computing a pixel's back-projected ray. The values n_m and n_b denote the number of measurements and the number of basis functions respectively. To set up the linear system we have to determine the line integrals for the line

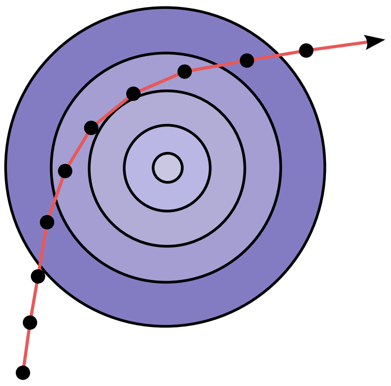


Fig. 8.5. The matrix entries are computed while performing the ray integration to determine the ray path with a fixed spatial step size. The values at the ray sampling positions (black dots) approximate the ray integral over the radially symmetric basis functions (blue).

of sight of each pixel over the basis functions ϕ_i. We use radially symmetric Kaiser-Bessel basis functions for a high-quality reconstruction. The matrix entries are computed while performing the ray integration, see Figure 8.5. We are using the visual hull restricted tomography algorithm of Chapter 5 to solve the linear system in Eq. (8.12) independently for each coordinate. Therefore, those columns of the matrix containing integrals over basis functions outside the visual hull [111] are removed from the computations. However, in this application this cannot be done as a post-processing step as introduced in Chapter 6, because of the non-linear nature of the rays c. For this reason we have to compute the visual hull of the object before setting up the linear systems.

Similar to Chapters 5 and 6, the linear systems can be solved using CGLS [69] if the number of cameras recording the scene is moderate. In the case of more than 8 cameras it is necessary to resort to out-of-core algorithms because of the memory requirements for storing matrix $\mathbf{S}$. For larger numbers of viewpoints we use a re-formulation of the simultaneous algebraic reconstruction technique (SART) [96, 197] in terms of matrix-vector products. The SART iteration can be performed on partial matrices $\mathbf{S}_k$, e.g. corresponding to equations generated by the measurements of one view. It takes the following form:

$$\mathbf{n}^j_{k+1} = \mathbf{n}^j_k + \mathbf{S}_k{}^T \frac{(\mathbf{d}^{out} - \mathbf{d}^{in})^j - \mathbf{S}_k \mathbf{n}^j_k}{m \operatorname{diag}(\mathbf{S}_k \mathbf{S}_k{}^T)}, \tag{8.14}$$

where m is the number of equations in matrix $\mathbf{S}_k$, j is the coordinate index, and k is the iteration count for the tomographic reconstruction. The SART iteration has an intuitive explanation as the tomographically back-projected, rescaled residual error in the image plane, that is, the solution is updated with a rescaled version of the tomographic back-projection of the residual error. The matrix-vector formulation of SART readily accounts for the visual hull restriction.

In general it is preferable to use the conjugate gradient method to solve the linear systems, Eq. (8.12), since it exhibits better convergence properties, possesses implicit regularization properties and allows for the application of explicit regularization methods like e.g. Tikhonov regularization as discussed in Sec. 6.3.

8.3.3 Integration of the Gradient Field

After the computation of ∇n, the final step is to find the refractive index field n from the gradient information. Computing n from ∇n is similar to computing a surface from its normals. We use the definition of the Laplacian operator

$$\triangle n = \nabla \cdot \nabla n \tag{8.15}$$

to compute n. The left hand side of Eq. (8.15) is discretized and the right hand side of it is computed using our recovered ∇n. The resulting Poisson equation is solved for n. We use central differences for the Laplacian operator and Dirichlet boundary conditions $n = 1$ at the boundary of the computational domain.

The Poisson equation can be solved most efficiently with multi-grid solvers. Since we have to perform the integration only once per frame we use a less efficient but easier to implement Jacobi-preconditioned Conjugate Gradient method [13] to solve the linear system, Eq. (8.15). It is not advisable to use the (preconditioned) CGLS method in this case because the convergence is very slow due to the squared condition number of the matrix.

8.4 Physical Measurement Setup

We use two different camera setups for our experiments with real-world measurements (Figure 8.6):

- a single video camera setup that can be used to acquire 2D data for use as environment mattes, or for the reconstruction of rotationally symmetric flows. In this setup, we use a Prosilica black-and-white 1.5 megapixel C-MOUNT camera, and

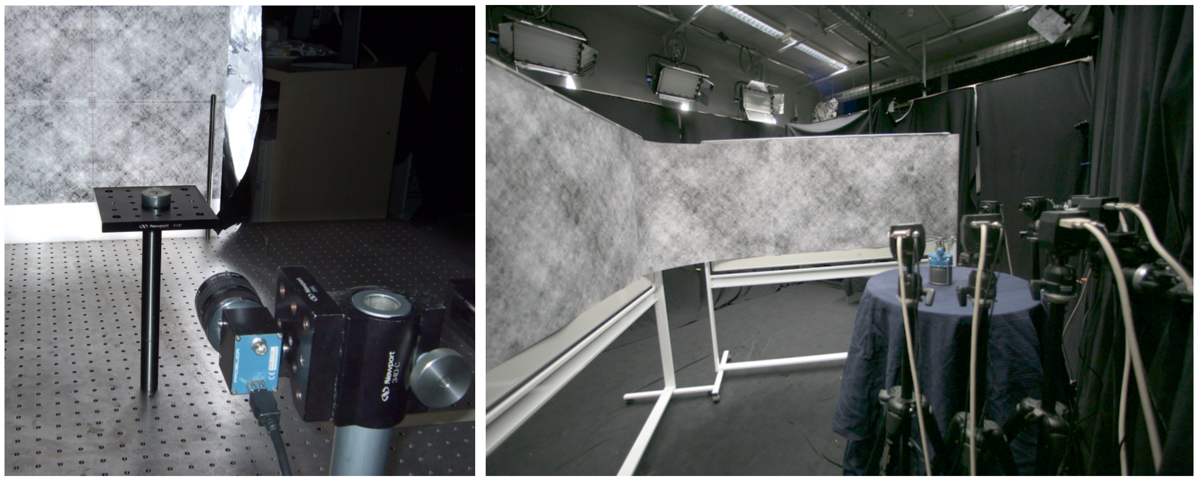


Fig. 8.6. The single camera setup (left) and the camera array (right) that we use for acquisition.

- the Imperx MDC-1004 multi-camera system, described in Sec. 4.1.1, that can be used to capture non-symmetric flow for 3D reconstruction.

The Wavelet Noise patterns that we use as a background are printed either on paper, or on overhead transparencies with a laser printer. Since short camera exposure times are required to capture fast or very turbulent flows, it helps to use transparencies that are backlit by a light box or another bright and well-diffused light source. For slow or laminar flow, the patterns can be printed on paper and used in a reflective setting.

The cameras are geometrically calibrated as discussed in Sec. 4.2. Photometric and colorimetric calibration is not necessary in this case.

For fluid imaging, we use a rectangular water tank with clear plastic walls of homogeneous thickness. The cameras are located outside the tank, observing it through one of the walls. The ray bundle emerging from each perspective camera is refracted on a planar interface between water and tank, yielding a different perspective view with a virtual center of projection behind the location of the camera. This virtual center of projection and the associated intrinsic and extrinsic (virtual) camera parameters can again be found with standard calibration techniques.

8.5 Results and Applications

Our evaluation of the BOS algorithm and the tomographic reconstruction is based on both simulations with synthetic data, and real-world measurements.

8.5.1 BOS Imaging on Synthetic Datasets

For ground truth experiments with the BOS algorithm, we applied known 2D flow fields to a Wavelet Noise pattern, and recovered the original flow from these images. We found that we can obtain very good results (relative RMS error below 1%) for high isotropic scaling factors and anisotropic scaling around 4 : 1. This is more than sufficient accuracy for the flow fields we are considering here, as well as for low-curvature solids such as the 75 *mm* lens from Figure 8.3. High curvature solids could result in larger distortions, which would require different methods.

8.5.2 BOS Imaging on Real Measurements

We acquired BOS data for a large number of different flows. Figure 8.7 shows the displacement magnitude plots of a small selection from that set, including both laminar flows and more turbulent ones.

These kinds of datasets can be used directly in rendering, for example as environment mattes to distort the camera rays. Similarly, the data can be used to distort light rays, for example in a photon mapping algorithm. This will cast caustics on the receiving surface, which are also known as *shadowgraphs*. Real-world shadowgraphs are related to Schlieren imaging, and can also be used to image flows [180]. Both the environment matting and the shadowgraph rendering are depicted in Figure 8.8.

8.5.3 Tomographic Reconstruction of Simulated Data

To obtain quantitative results for the robustness of our tomographic reconstruction method, we ran the algorithm on synthetic data. We tested both smooth datasets and ones with high spatial frequencies, and analyzed the relative RMS error of the reconstruction depending on the number of views and the differences in refractive index. For the smooth datasets, we used Gaussian blobs of different variance. Publicly available volume datasets with volume densities linearly mapped to refractive indices were used for the high-frequency analysis.

Table 8.1 shows the relative RMS error for the HIPIP dataset from the UNC CHVRTD volume collection for various configurations. We define relative RMS error as RMS/Δn, with $\Delta n = n_{max} - n_{min}$. In all tests we set $n_{min} = 1$. As expected, the RMS error drops with the number of views used for reconstruction. However, even with only 8 views and a comparatively large Δn of 0.1, the RMS error across the volume is below 3% for refractive index fields, which indicates that our camera array is adequate for measurements of gases with reasonable precision.

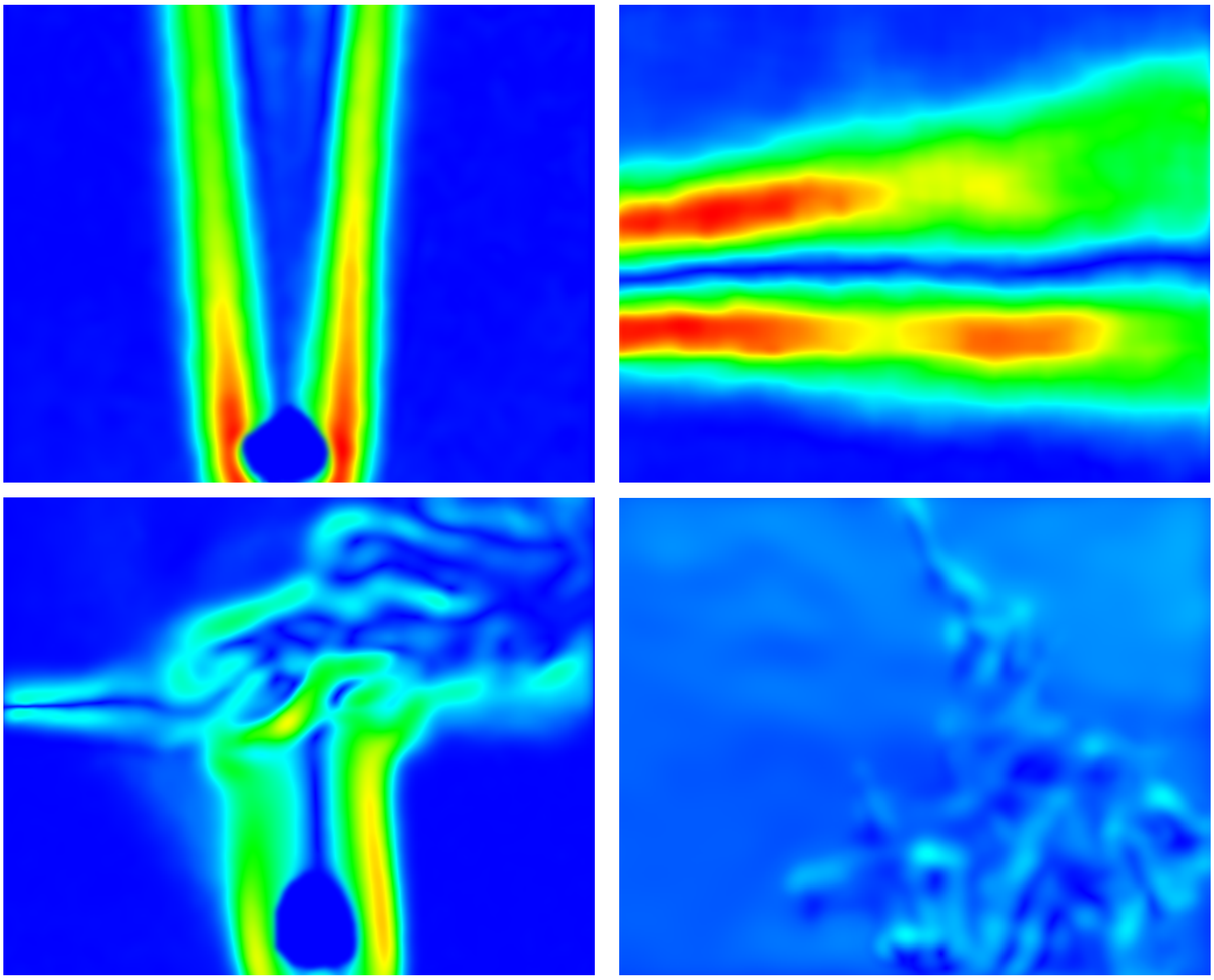
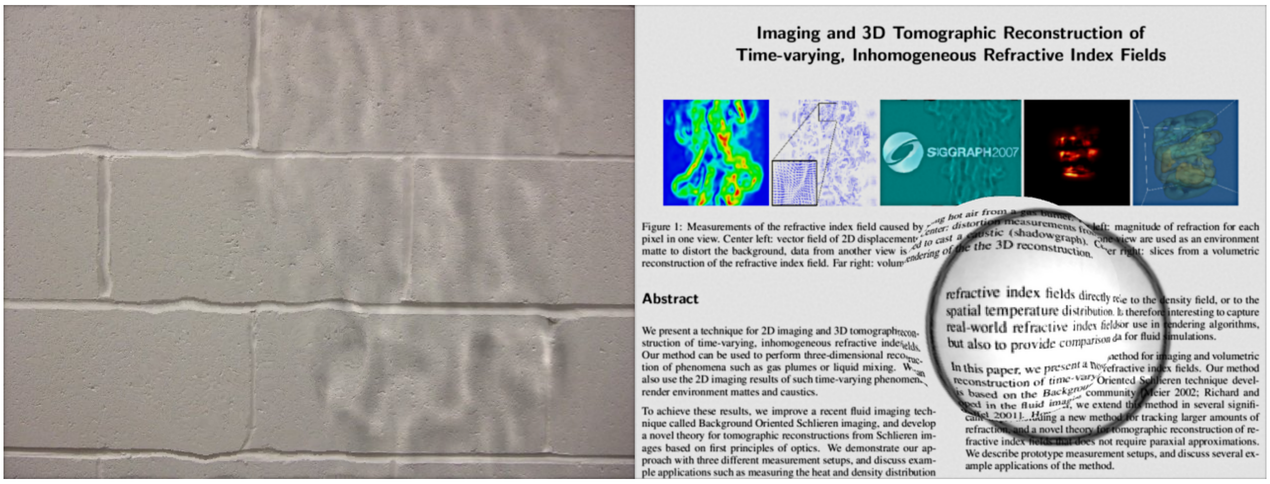


Fig. 8.7. Displacement magnitude images for various flows. Top left: laminar flow of heat rising from a candle. Top right: low turbulence gas jet from a spray can. Bottom left: turbulent interaction of a hot air plume above a candle with a jet of compressed air. Bottom right: turbulent mixture of water with corn syrup.

It is worth noting that most of the error is concentrated in regions of strong gradients, which appear blurred in the reconstruction. A larger number of views reduces this blur, and therefore the reconstruction error.

8.5.4 Tomographic Reconstruction of Real Measurements

On the right of Figure 8.9, we show a 3D reconstruction of a laminar candle plume from a single BOS image, assuming rotational symmetry. The BOS data (left) was projected from a total of 16 virtual camera positions. In the process, the slight asymmetries present in the data BOS image are averaged out, so that the reconstructed volume is fully symmetric. The closing of the iso-surfaces near the top is an artifact of the limited reconstruction volume since the top of the plume was outside the camera field of view.



Imaging and 3D Tomographic Reconstruction of Time-varying, Inhomogeneous Refractive Index Fields

Figure 1: Measurements of the refractive index field caused by ...ng hot air from a gas burner. ...enter: distortion measurements from ...left: magnitude of refraction for each pixel in one view. Center left: vector field of 2D displacement... ...ed to cast a caustic (shadowgraph). ...one view are used as an environment matte to distort the background, data from another view is ...endering of the the 3D reconstruction. C...er right: slices from a volumetric reconstruction of the refractive index field. Far right: volum...

Abstract

We present a technique for 2D imaging and 3D tomographic recon- struction of time-varying, inhomogeneous refractive index fields. Our method can be used to perform three-dimensional reconstruc- tion of phenomena such as gas plumes or liquid mixing. We can also use the 2D imaging results of such time-varying phenomena to render environment mattes and caustics.

To achieve these results, we improve a recent fluid imaging tech- nique called Background Oriented Schlieren imaging, and develop a novel theory for tomographic reconstructions from Schlieren im- ages based on first principles of optics. We demonstrate our ap- proach with three different measurement setups, and discuss exam- ple applications such as measuring the heat and density distribution

refractive index fields directly re...e to the density field, or to the spatial temperature distribution. It therefore interesting to capture real-world refractive index fields for use in rendering algorithms, but also to provide comparison da... for fluid simulations.

...method for imaging and volumetric

In this paper, we present a ne... refractive index fields. Our method reconstruction of time-vary... Oriented Schlieren technique devel- is based on the Backgrou... community [Meier 2002; Richard and ...oped in the fluid imag..., we extend this method in several signifi- cant 2001]. ...ding a new method for tracking larger amounts of refraction, and a novel theory for tomographic reconstruction of re- fractive index fields that does not require paraxial approximations. We describe prototype measurement setups, and discuss several ex- ample applications of the method.

Fig. 8.8. Environment matte and shadowgraph rendering of two BOS measurements. Left: hot air, Right: the chipped lens from Figure 8.3. Also see video.

#Views	RMS Error
8	2.89%
16	2.69%
32	2.38%
64	1.97%
128	1.54%

Table 8.1. RMS error for the tomographic reconstruction of the synthetic HIPIP volume from optical flow images.

Figure 8.10 shows a few frames from some of the real-world footage we captured with the camera array. The top four rows are different renderings of the same dataset: turbulent hot air rising from a gas burner. The first two rows show a maximum intensity projection rendering of the refractive indices, and the third and fourth rows an iso-surface representation of the same information. Both sequences clearly show the advection of the hot air.

The last two rows of Figure 8.10 show a direct volume rendering of the *temperature distribution* in a plume of hot air above a candle. This flow is more laminar than the flow above the burner. These images are an example of the secondary information that can be extracted from the refractive index fields, depending on the dataset.

For gases, the refractive index is directly related to the volume density ρ through the Gladstone-Dale Equation:

$$\rho = \frac{n-1}{k(\lambda)},$$

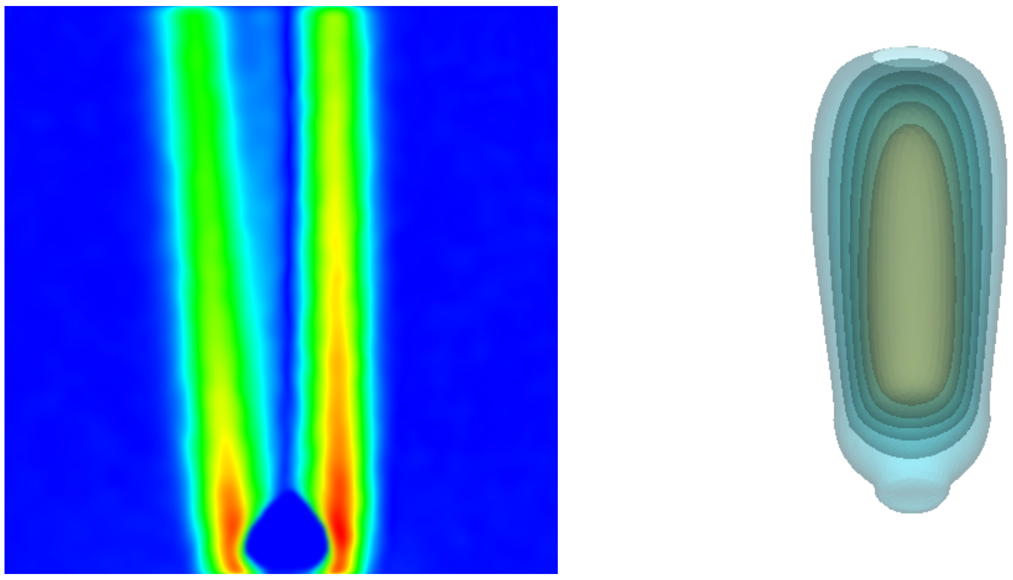


Fig. 8.9. Reconstruction of a rotationally symmetric flow from a single BOS image.

where k is the Gladstone-Dale constant, which depends on the material and weakly on the wavelength of light. For air and visible light, k has an approximate value of 0.23 cm^3/g.

Under certain assumptions, one can derive further quantities from the volumetric densities. For example, under constant pressure such as in the examples of hot air, the ideal gas law (see, e.g. [201]) can be used to infer the temperature distribution in the volume as

$$T = \frac{p \cdot M}{R \cdot \rho},$$

where p is the pressure, M is the molecular mass of the gas ($\approx 29\ g/mol$ for air), and R is the gas constant ($\approx 8.31\ J/(K \cdot mol)$).

In a similar fashion other secondary information can be inferred from the refractive index field. In the case of liquid mixtures, for example, the refractive index is an indicator of the local fluid concentrations. This kind of information could be useful for verifying fluid simulators, and furthering the understanding of certain types of flow in general.

8.6 Summary

We have presented a novel technique for capturing time-varying, inhomogeneous refractive index fields, such as the ones created by gas and liquid flows. Our major contributions are an improvement of the Background Oriented Schlieren imaging method developed in fluid imaging, and a novel theory for tomographic reconstruction of 3D volumes from Schlieren images, which gives rise to a practical algorithm. With these methods, it is now possible to capture complex flows with very moderate hardware requirements. The data captured with this approach can be directly used in computer graphics for rendering camera distortions or caustics.

Maybe even more interesting in the long run is the fact that the recovered refractive index fields are indicators of other physical properties, such as the density or temperature distribution in gases, or concentrations of fluids in the scenario of liquid mixing. These and similar derived quantities could be excellent tools for furthering the understanding of fluids, and comparison of fluid simulations against reference data.

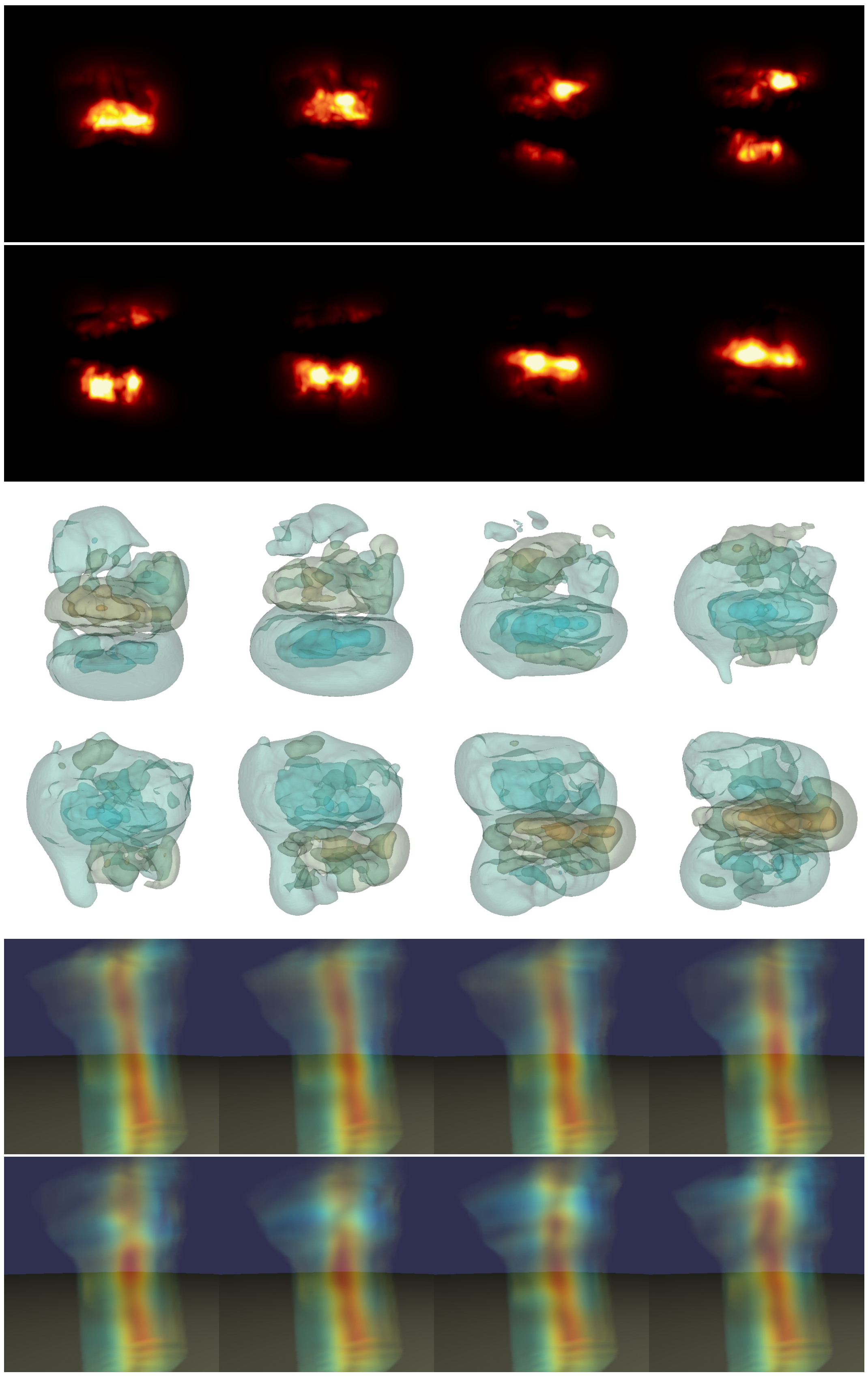


Fig. 8.10. Results from the tomographic reconstruction process. Top: refractive index volume of turbulent hot air rising from a gas burner, rendered with maximum intensity volume rendering. Center: the same volumes rendered as iso-surfaces. Bottom: direct volume rendering of the temperature distribution in a more laminar flow of hot air rising above a candle (for clarity, the volume rendering is restricted to the visual hull).

Part IV

Rendering

9

Real-time Rendering of Optically Complex Refractive Objects

In this chapter we develop a real-time rendering method that can display the natural phenomena that are reconstructed using the techniques described in Chapters 5 - 8. However the rendering method presented here is much more powerful and can render complex anisotropic effects such as single scattering effects and surface reflectance including the Fresnel effect. The majority of the optical characteristics of natural phenomena as discussed in Sec. 2.1 can be reproduced in real-time for static objects. For dynamic effects interactive frame-rates are possible. The rendering method is once again based on the ODE ray-tracing scheme, Sec. 2.3.4. The differential equations can be integrated very efficiently on todays programmable graphics hardware. Additionally we describe a technique to simulate light propagation in optically inhomogeneous media. In this way shadowgraphs and caustics caused by refractive objects or phenomena can be simulated efficiently. The light propagation scheme is based on the same mathematical framework that is used throughout the thesis to compute curved light rays. The two ingredients fast viewing ray computation and efficient light propagation are joined by an expressive image formation model that allows the use of emission, absorption, reflection, refraction and single scattering characteristics to describe optically complex refractive objects.

In the following we first introduce our image formation model, describe the approximations that facilitate real-time rendering using this model, derive the mathematical framework for viewing ray and light propagation in the scene and finally describe the implementation on a consumer grade graphics board.

9.1 Image Formation Model

9.1.1 General Image Formation

We are concerned with the realistic and efficient rendering of transparent objects with spatially varying optical characteristics. To this end, we assume that the parameters of the model are stored in a 3D volume that can be generated by a human modeler or captured using one of the methods described in Chapters 5 - 8. Our general model of image formation accounts for emission, absorption, reflection, refraction and single scattering. A mathematical formulation for a particular, potentially curved ray that passes through the volume is given by

$$I(c) = \int_c I_c(\mathbf{x}, \mathbf{d}) A(t, c) dt + I_{bg} A(t_\infty, c) \ , \tag{9.1}$$

where I_c denotes light intensity on the ray c that is scattered, emitted or reflected into the direction of the eye. I_{bg} is the background intensity and $A(t, c)$ the absorption of light at position t along the ray. I_c is composed of different components contributing to the light intensity on a given ray. Function I_c depends on the position in space $\mathbf{x} = c(t)$ and the local ray direction $\mathbf{d} = \frac{dc}{dt}$. In general it is wavelength-dependent and can be computed using different parameters for each wavelength λ. We can express I_c in terms of these variables:

$$I_c(\mathbf{x}, \mathbf{d}) = I_s(\mathbf{x}, \mathbf{d}) + \delta(\mathbf{x}) \rho I_r(\mathbf{x}, \mathbf{d}) + I_e(\mathbf{x}, \mathbf{d}) \ . \tag{9.2}$$

Here I_s denotes the intensity due to in-scatter, I_r the intensity caused by reflections and I_e the local emission intensity. The Dirac delta function $\delta(\mathbf{x})$ serves as a boundary indicator, i.e. it equals one if $\mathbf{x}$ is on a boundary between two different objects and zero elsewhere. This accounts for the fact that reflections occur on boundaries between different materials. ρ is the Fresnel reflection factor for unpolarized light [28]. The Fresnel transmission factor τ enters the absorption equation (9.5) through factor $T(t)$, as we will describe later.

I_s, I_r and I_e all depend on the position in space $\mathbf{x}$ and on the local ray direction $\mathbf{d}$ and can be evaluated locally given volumetric descriptions of their distributions. The last point is important. The locality of I_c, given appropriate pre-computations, allows us to parallelize its computation in an efficient way.

We formulate in-scatter in terms of the scattering phase function ϕ. It may vary in space and depends further on the local ray direction $\mathbf{d}$ and the local light directions ω

$$I_s(\mathbf{x}, \mathbf{d}) = \int_\Omega I_i(\omega) \phi(\mathbf{x}, \mathbf{d}, \omega) d\omega \ . \tag{9.3}$$

$I_i(\omega)$ denotes the incoming light intensity from direction ω and the light contributions due to in-scatter are integrated over the sphere of incoming directions to yield I_s. Similarly we write

$$I_r(\mathbf{x}, \mathbf{d}) = \int_{\Omega_+} I_i(\omega)\psi(\mathbf{x}, \mathbf{d}, \omega)d\omega \ , \tag{9.4}$$

where ψ describes a BRDF including the cosine factor. The normal of the surface can either be provided as an additional function or be derived from the refractive index field n. ψ thus gives us the light contribution due to reflection on a boundary between two different materials. Please keep in mind that this term is only valid on the boundary of objects and its contribution is triggered by $\delta(\mathbf{x})$.

I_e is just a function $I_e(\mathbf{x}, \mathbf{d})$ in its most general form. In conjunction with the light source definitions, it can be used to model multiple scattering effects or self-emission due to fluorescence or phosphorescence.

Finally, we have a closer look at the absorption function A in Eq. (9.1). If arbitrary absorption distributions are considered, it depends on the distance along the ray and the ray's shape, and thus it evaluates to

$$A(t, c) = T(t)e^{-\int_0^t a(c(s))ds} \ , \tag{9.5}$$

i.e. the absorption function describes the exponential attenuation of an intensity at position $\mathbf{x} = c(t)$ due to a spatially varying attenuation function a. Out-scatter can be included as an additive term into the absorption integral, see [120], but we omit it here for reasons of clarity. $T(t)$ is the product of all Fresnel transmission factors τ encountered along the ray up to position t.

9.1.2 Simplified Image Formation

In its general form, our image formation model is too complex to be evaluated in real-time. Therefore, we make two simplifying assumptions:

1. The light in the scene originates from a discrete number of light sources, and
2. for each point in the scene, there is only one incoming light ray from each of the light sources.

The second point is important. We model only one ray from a light source to a particular point in space. This ray can be chosen arbitrarily from the set of rays passing through a particular point in space, e.g. the first arrival ray or the ray carrying the highest energy. This restriction allows us to develop

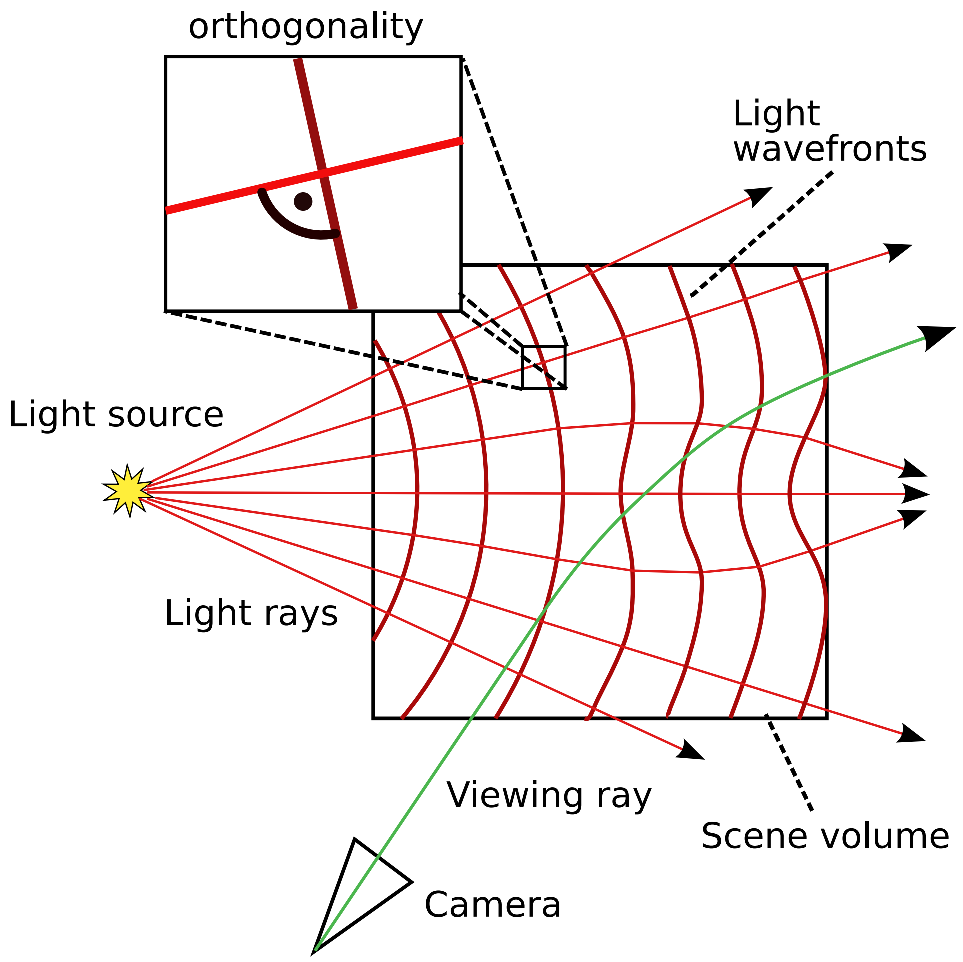


Fig. 9.1. 2D illustration of our complex image formation scenario – due to inhomogeneous material distribution, light rays and viewing rays are bent on their way through the scene volume. Light rays always travel orthogonal to the light wavefronts, i.e. the iso-surfaces of constant travel time.

an efficient rendering algorithm for a fairly complex image formation model, since we can convert the integrals of Eqs. (9.3) and (9.4) into discrete sums:

$$I_s(\mathbf{x},\mathbf{d}) = \sum_j I_{i_j}(\mathbf{x})\phi(\mathbf{x},\mathbf{d},\mathbf{l}_j) \tag{9.6}$$

$$I_r(\mathbf{x},\mathbf{d}) = \sum_j I_{i_j}(\mathbf{x})\psi(\mathbf{x},\mathbf{d},\mathbf{l}_j) \ . \tag{9.7}$$

Now, at a particular point in space we have only *one* incoming intensity I_{i_j} and *one* incoming local light direction $\mathbf{l}_j$ per light source j. Thus, if we can pre-compute these, and if we are able to quickly traverse the light ray c, we can evaluate Eq. (9.1) with local operations only. In the following section we derive the mathematical recipes for viewing ray traversal and light propagation.

9.2 Light Transport

In this section, we develop the equations for the transport of light in the scene. The propagation of viewing rays is described in Sect. 9.2.1 and light transport is discussed in Sect. 9.2.2. Viewing rays and light rays, Fig. 9.1, behave very similarly and the governing equations are derived from the same basic equation, the *ray equation of geometric optics* [28]. However, we use different parameterizations to account for specifics in the two processes. Please note that for light rays, we have to take the intensity fall-off into account whereas viewing rays carry radiance.

9.2.1 Viewing Ray Propagation

The ray equation of geometric optics has been previously used in computer graphics by [186] and [178]. The equation describes the motion of a light 'particle' in a field n of inhomogeneous refractive indices:

$$\frac{d}{ds}\left(n\frac{d\mathbf{x}}{ds}\right) = \nabla n \; . \tag{9.8}$$

It is derived from the eikonal equation and the motion of a massless particle along the gradient of the eikonal solution as described in Sec. 8.3.1. ds denotes an infinitesimal step in the direction tangential to the curved ray. Eq. (9.8) can be re-written as a system of first order ordinary differential equations

$$\frac{d\mathbf{x}}{ds} = \frac{\mathbf{d}}{n} \tag{9.9}$$

$$\frac{d\mathbf{d}}{ds} = \nabla n \tag{9.10}$$

which can be discretized using a simple Euler forward scheme

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \frac{\Delta s}{n}\mathbf{d}_i \tag{9.11}$$

$$\mathbf{d}_{i+1} = \mathbf{d}_i + \Delta s \nabla n \tag{9.12}$$

or some higher order integration method like the Runge-Kutta family [155]. This is again an application of the ODE ray-tracing framework introduced in Sec. 2.3.4. The equations (9.9) and (9.10) have the nice property that the spatial step size is equal for all ray trajectories, see Appendix A.4 for a proof. This proves advantageous for rendering, Sect. 9.3, where the number of iterations for each particle trace should be approximately equal to ensure optimal performance. Conveniently, ray bending and total reflection are naturally supported by the ray equation of geometric optics.

9.2.2 Modeling Light Sources

We model a light source with a three-dimensional vector field of local light directions $\mathbf{l}(\mathbf{x})$ and a scalar field of intensities $I_i(\mathbf{x})$ (cf. Sect. 9.1.2). These fields can be computed in several ways. A popular choice among computer graphics researchers is photon mapping [91] of which GPU implementations are available [156]. In the computational physics and numerical analysis literature a huge range of methods have been proposed to solve this problem. Choices range from purely Eulerian formulations using the eikonal and transport equations [30], phase space methods [149] and hybrid Lagrangian-Eulerian approaches [18] to adaptive wavefront tracing [45]. All methods except for the purely Eulerian approach deal with the inherent multi-valuedness of the solution of the underlying equations.

We use adaptive wavefront tracing [45, 36] for the computation of the local light directions and intensities because it offers the best trade-off between computation time and accuracy of the solution. A wavefront is an iso-surface of constant travel time of light originating from a light source, see Fig. 9.1. In accordance with Fermat's Principle light rays travel always normal to these wavefronts. The wavefront is discretized by a set of connected particles. These are propagated through the inhomogeneous refractive index field. In case the wavefront becomes under-resolved new particles are inserted to preserve a minimum sampling rate, Fig. 9.2. The local light directions are represented by the traveling directions of the particles and the intensities can be computed from the areas of wavefront patches. The pre-computation of the three-dimensional light distribution takes the following subsequent steps:

- wavefront propagation,
- intensity computation,
- wavefront refinement,
- voxelization of the wavefront normals and intensities.

This process is repeated until the wavefront leaves the volume of interest. The individual steps are detailed in the following.

Wavefront Propagation

We discretize the wavefront into a set of inter-connected particles which are propagated independently. This way, the wavefront is subdivided into so-called wavefront patches whose corners are defined by light particles, Fig. 9.2 (right). The connectivity information is needed for the intensity computation. The propagation of the particles is performed according to Eq. (9.8) similar to the ray propagation, Sec. 9.2.1. We re-parameterize it to yield equi-temporal discretization steps:

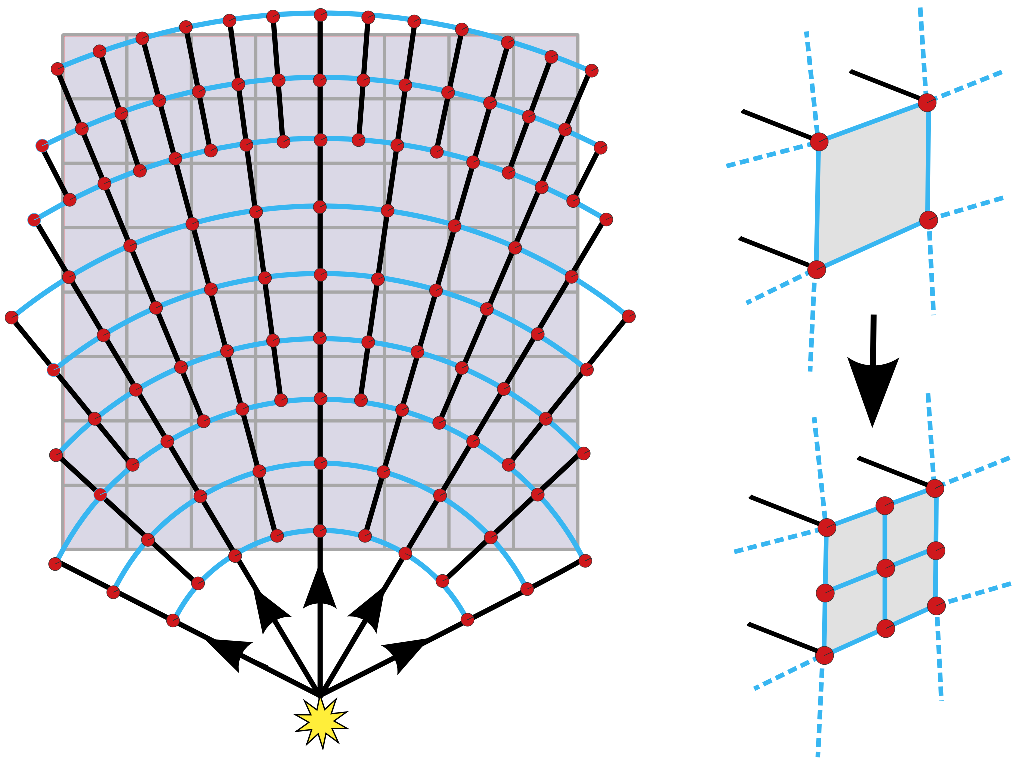


Fig. 9.2. Adaptive wavefront refinement – (left) 2D illustration: the wavefront is represented by particles (red dots) that are connected to form a wavefront (blue lines). While advancing through the voxel volume (shown in gray) the wavefront is tessellated such that its patches span less than a voxel. – (right) 3D illustration of the tessellation for one wavefront patch.

$$n\frac{d}{dt}\left(n^2\frac{d\mathbf{x}}{dt}\right) = \nabla n \; . \tag{9.13}$$

A proof of this property is given in Appendix A.4. The re-parameterization is necessary to enable a simple formulation of the intensity computation described in Sect. 9.2.2. It ensures that all particles stay on a common wavefront over time. Similar to Eqs. (9.9) and (9.10) we can write Eq. (9.13) as a system of first order ordinary differential equations

$$\frac{d\mathbf{x}}{dt} = \frac{\mathbf{d}}{n^2} \tag{9.14}$$

$$\frac{d\mathbf{d}}{dt} = \frac{\nabla n}{n} \; . \tag{9.15}$$

This formulation enables a fast GPU implementation of the wavefront propagation scheme as a particle tracer. Once the wavefront can be tracked over time we can compute the intensity of light from the area of the wavefront patches that connect the particles.

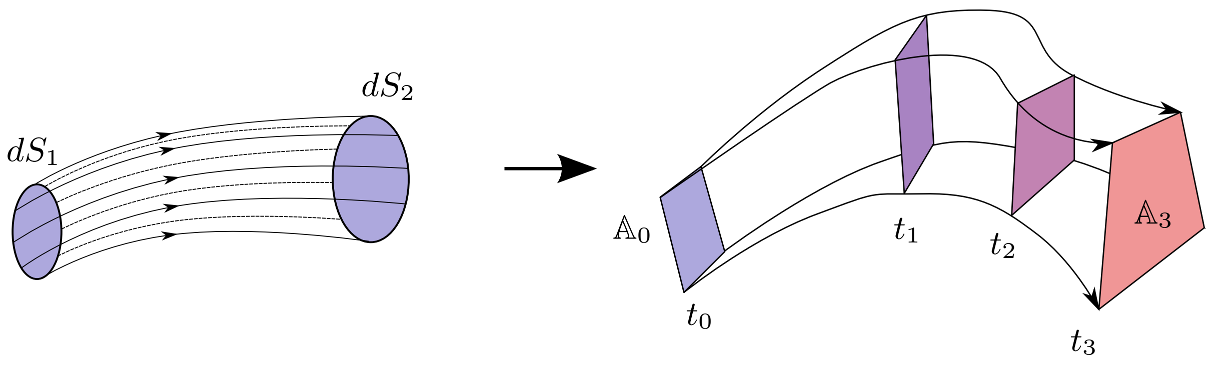


Fig. 9.3. The intensity law of geometric optics (left) and its discretized version (right) in the form of a *stream tube*. The product of area and intensity is constant along a tube of rays.

Intensity Computation

The intensity computation is based on *the intensity law of geometric optics* [28], see Fig. 9.3 (left). The law states that in an infinitesimal tube of rays the product of intensity and area stays constant:

$$I_1 dS_1 = I_2 dS_2 \ . \tag{9.16}$$

We use a discretized version of the intensity law to update the energy contribution of wavefront patches during propagation. The motion of each patch through the scene describes a so-called stream-tube, Fig. 9.3 (right). Eq. (9.16) then reads

$$I(t) = \frac{I(0)\mathbb{A}(0)}{\mathbb{A}(t)} \ . \tag{9.17}$$

Here $\mathbb{A}(t)$ denotes the area of a wavefront patch at time t and $I(t)$ its intensity per area. Since we are modeling absorption in our image formation model this effect has to be included in the intensity computation as well. Thus the final intensity for a wavefront patch is given by

$$I_i(t) = \frac{I(0)\mathbb{A}(0)}{\mathbb{A}(t)} e^{-\int_0^t \frac{a(c(\hat{t}))}{n} d\hat{t}} \ . \tag{9.18}$$

Wavefront Refinement and Voxelization

In order to obtain a continuous volumetric representation of the light distribution the wavefront patches have to be voxelized. However, due to divergent corners the patches can in general become arbitrarily large while they are propagated.

If a wavefront patch becomes larger than one voxel the wavefront becomes under-resolved and sampling problems in the computational grid occur.

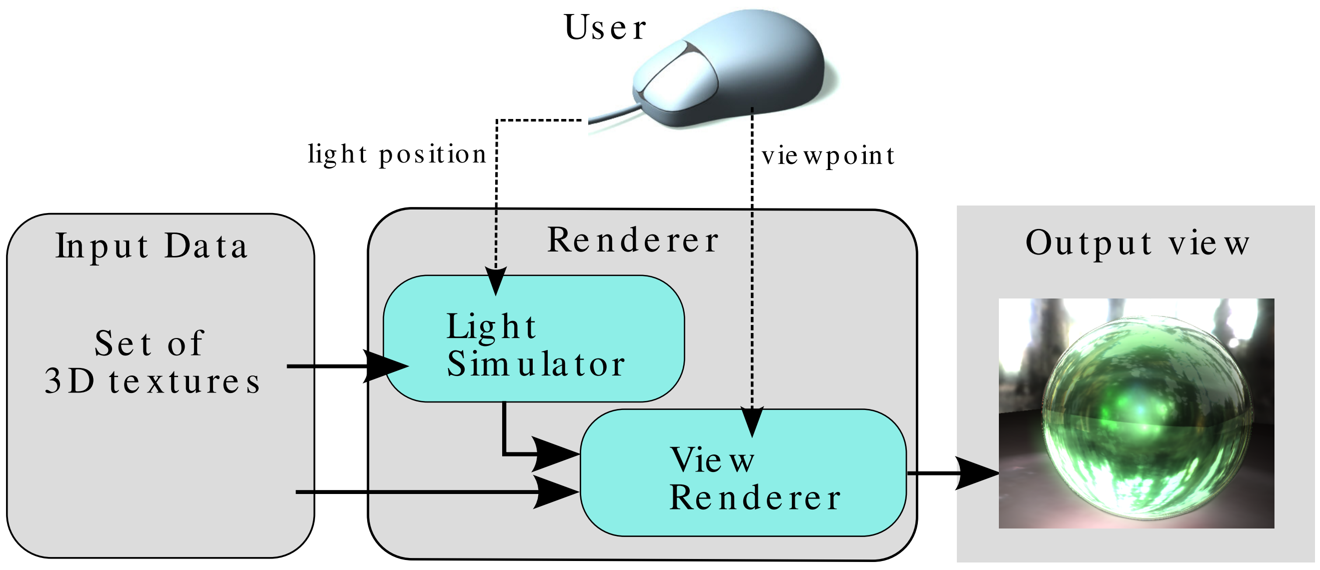


Fig. 9.4. Work-flow of our rendering system.

To alleviate this, we adaptively split the wavefront patches once they grow larger than one voxel, see Fig. 9.2. By this means, we also adapt to the changing curvature of the continuous solution.

Since at the same time, graphics hardware is not able to rasterize arbitrarily sized quads into 3D volumes, we use the adaptive sampling and equate wavefront patches with their midpoints, storing intensity and directional information as a single voxel sample. Under-sampling of the wavefront is thus solved in conjunction with implementing GPU-based voxelization.

9.3 Implementation Issues

After the theoretical foundation has been set, we now have a closer look at how to map the outlined concepts onto the GPU. Fig. 9.4 illustrates the work-flow of our renderer. In the following, we detail its most important components, the employed data format, Sect. 9.3.1, the light simulator, Sect. 9.3.2 and the view renderer, Sect. 9.3.3.

9.3.1 Input Data Format

Input scenes are stored as a set of 3D volume textures. In a first set of volumes, the spatially-varying refractive index field, as well as its gradient field are stored. The objects in our test scenes were created as solids of revolution, implicit surfaces, or triangle meshes that we rasterized into the 3D volume. Refractive index distributions can be reconstructed as described in Chapters 7 and 8, derived directly from the implicit functions of the objects or they can be defined interactively. In case the gradient information is not already known we smooth the volumes prior to gradient computation.

Other 3D textures contain the spatially-varying attenuation function (separate values for the three color channels), the material boundary indicator, as well as BRDF parameters and emission descriptions. For approximating anisotropic scattering effects, we employ the Henyey-Greenstein scattering-phase function [78]. Its parameters are also stored in volumetric textures.

9.3.2 Light Simulator

Our implementation follows the adaptive wavefront propagation described in Sect. 9.2.

Basically, after initialization at the light source, the wavefront is represented as a *particle system*. The difference to a standard particle system is that the particles are bound into packets of four and thus span a *wavefront patch*, Fig. 9.2 (right). This allows us to simulate the stream tube concept on graphics hardware, where each tube in the wavefront is represented by its front-most patch. All the patches are stored in floating point textures, which hold the four corners' positions, their propagation directions (see Fig. 9.3 (right)) and a RGB energy value (see Sec. 9.2.2).

During initialization, we use the 2D-parameterization of the patch list texture to either produce a planar wavefront (directional light source) or a spherical wavefront (point light source). The initialization ensures that the wavefront is large enough to cover the simulation volume. Other light source types (as multi-directional light) can be implemented, as the wavefront patches are independent and thus can be stacked on top of each other. The propagation of the wavefronts through the scene and the logging into the output 3D volume is performed in four subsequent steps described in the following.

Patch List Update

For every time step, we update the patches' corner positions and directions according to Eqs. (9.14) and (9.15). We further update the patches' held RGB energies according to Eq. (9.18) in case a material with attenuation is traversed.

Patch List Voxelization

After each update step, we need to protocol the wavefront patches into the 3D output volumes for energy and direction. On graphics hardware, this is accomplished using point primitives and the concept of Flat 3D textures introduced by Harris et al. [70]. Before we commit a patch to the 3D volume, we check if it is allowed to overwrite the one already stored there (if any), e.g. based on the first arrival or the highest energy criterion.

Patch List Tessellation

After the wavefront patches have been propagated we check whether their area is larger than one voxel cross-section. In case the patch is too large it is split as shown in Fig. 9.2 (right). We currently do not unite converging wavefront patches to form larger ones because the merging of the patches' propagation directions is non-trivial in the general case (i.e. when they are not co-planar).

Patch Termination

If a wavefront patch holds too little energy, we apply an energy threshold to eliminate it. We assume it will not contract again and thus yield a noteworthy energy contribution. Termination typically happens after too many tessellations or loss of energy due to repeated attenuation. We also eliminate patches that leave the simulation volume.

After the new patch list has been generated, it is forwarded to the patch list update to advance the simulation. This repeats until no patches remain in the simulation volume. In Fig. 9.5, we show a wavefront propagating through a wine glass. The computed intensities are used as colors, resulting in a preview of the caustic patterns in and around the object.

9.3.3 View renderer

Given the output from the light simulator, we can render arbitrary user views of a complex refractive object. The view renderer implements a fast ray-caster for arbitrarily bent viewing rays based on Eqs. (9.11) and (9.12). Intensities along viewing rays are computed according to Eq. (9.1), using the simplified image formation model and the scene parameters stored in the input textures.

In theory, we can handle arbitrary BRDF models, including parametric or tabulated representations. However, since our glass objects come close to perfect reflectors and a good approximation of the first reflection is already visually pleasing, we use simple dynamic environment mapping. The Fresnel effect and the anisotropic scattering phase function are computed on-the-fly in the fragment shader.

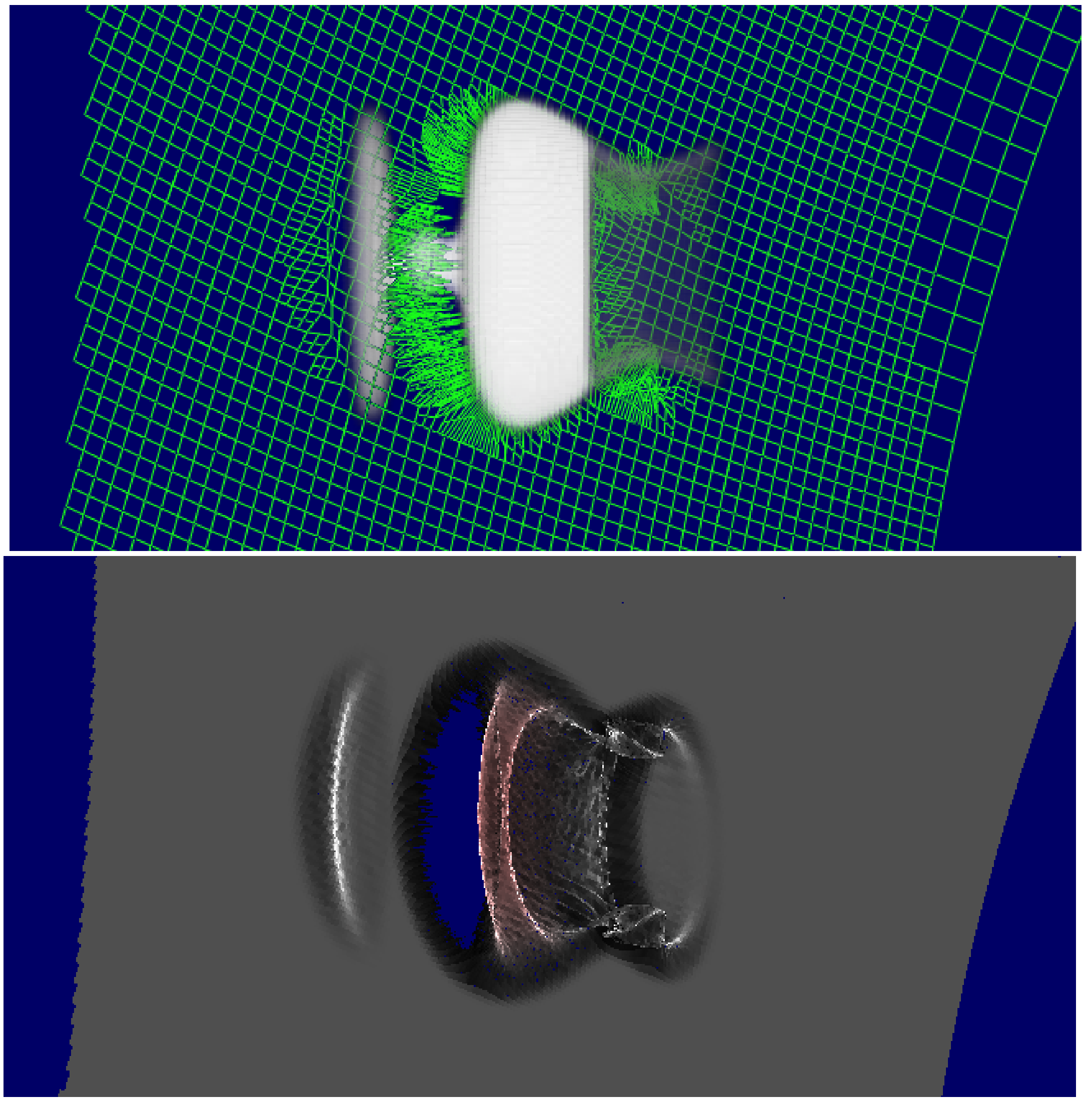


Fig. 9.5. (top) The refractive index volume of the glass is approached by a spherical wavefront from the right. The adaptive tessellation of the wavefront is also visible. – (bottom) When it passes through the object, caustic patterns appear in its intensity distribution.

After the viewing ray has finished volume traversal, we use its exit direction to conduct a lookup into a dynamic environment map to approximate the background intensity. All lighting computations are performed in high dynamic range and an adaptive tone-mapping based on [101] is applied prior to display.

9.4 Results and Discussion

We rendered result sequences with five different objects in several surroundings, thereby visualizing different combinations of effects. The results are shown in Figs. 9.6 and 9.7. For light simulation within the objects we used the highest energy ray criterion to compute the three-dimensional light distribution.

Fig. 9.6 shows a variety of objects that were modeled by a human modeler. The objects exhibit various combinations of the optical effects that can be described by our image formation model. Fig. 9.7 shows renderings of a manipulated version of the Burner data set of Chapter 8. The volumetric real-world data acquired in the previous chapter was computationally modified to exhibit scattering and absorption characteristics. We identified regions with high refractive index gradients and added scattering and absorption terms depending on the gradient strength. This purely ad-hoc method is an example of the types of modifications that can be performed with computer models of real-world data.

Our test data were stored in 128^3 voxel volumes. On an AMD Dual Core Athlon with 2.6 GHz equipped with an NVidia GeForce 8800 GTX and 768 MB of video RAM, we obtain a sustained rendering frame rate of around 25 fps if one object is displayed and if the light source remains in a fixed position. Mind that the frame rate decreases when zooming in closely on the object, since then more rays need to be cast from the viewpoint into the volume. After moving a light source to a new position, the lighting simulation has to be rerun once. This typically takes around 5 to 7 seconds if the simulator is initialized with 50.000 wavefront patches. Due to the adaptive tessellation, the patch count in the simulation typically peaks at 450.000.

For our particular application, the ODE-based ray propagation and adaptive wavefront tracing formulation has a couple of intriguing advantages. The voxel representation allows for fast non-linear ray casting. Expensive ray/geometry intersections, like in [156], would lead to performance bottlenecks on complex curved light paths.

Adaptive wavefront tracing also enables us to simulate non-linear light transport with a fast particle tracer while simultaneously avoiding under-sampling problems. Our update times after light position changes are comparable to other state-of-the art GPU methods reproducing fewer effects, e.g. only caustics in isotropic media [46]. We see an advantage over alternative methods like photon-mapping [91] because we only insert particles when needed, i.e. when the wavefront is under-sampled. We also obtain densely sampled light intensity and direction information throughout 3D space, such that we can cater for anisotropic visual effects at any point in the scene. Also, no special reconstruction kernels are required. Furthermore, we obtain a physically plausible[1] light distribution with significantly reduced sampling problems. An advantage over PDE approaches is the fast simulation and the ability to pick a particular solution in case of multi-valuedness of the light dis-

[1] within the limits of geometrical optics, see [28] for details

tribution. For a particular point in space, we can choose the first arrival ray or the one carrying the highest energy. With additional memory consumption and higher algorithmic complexity multi-valued solutions could be computed as well.

Despite these advantages for refractive object rendering, on general scenes our algorithm does not match the power of related approaches like photon mapping, which can efficiently produce full global illumination solutions.

The required level of discretization makes our method only suitable for the simulation of spatially confined refractive objects. These objects may appear as part of larger scenes, as shown in our renderings. Due to the volumetric representation, the scene's size is mainly limited by the available video memory. Octree representations can help to further reduce memory consumption. Besides, with future generations of graphics boards, memory limits will become less of an issue. We also have to properly match triangle-based lighting to our volume-based lighting to insert our volumetric scene models into larger triangle-based scenes rendered with standard surface shading.

Furthermore, we are dependent on decent gradient fields to yield visually pleasing results. To this end, we pre-smooth the refractive index volumes prior to gradient evaluation. Here, one needs to take care to not over-smooth which leads to halo-effects around material boundaries. This effect can be prevented by using a suitable proxy geometry to restrict the tracing of rays to the inside of the silhouette of a refractive object. This measure additionally yields a speed-up over the rendering of the complete discretization volume, however volumetric light effects in the space surrounding the object cannot be rendered anymore. A sufficiently high voxelization level is needed for extreme close-up renderings. Otherwise, discretization artifacts in the lighting effects may occur.

9.5 Summary

In this chapter we presented a fast and versatile method to render a variety of sophisticated lighting effects in and around refractive objects in real-time. It is based on a sophisticated model of light transport in volumetric scene representations that accounts for a variety of effects, including refraction, reflection, anisotropic scattering, emission and attenuation. It employs a fast particle tracing method derived from the eikonal equation that enables us to efficiently simulate non-linear viewing rays and complex propagating light wavefronts on graphics hardware. Since our method maps efficiently to a standard GPU we see many applications in computer games and real-time visualization.

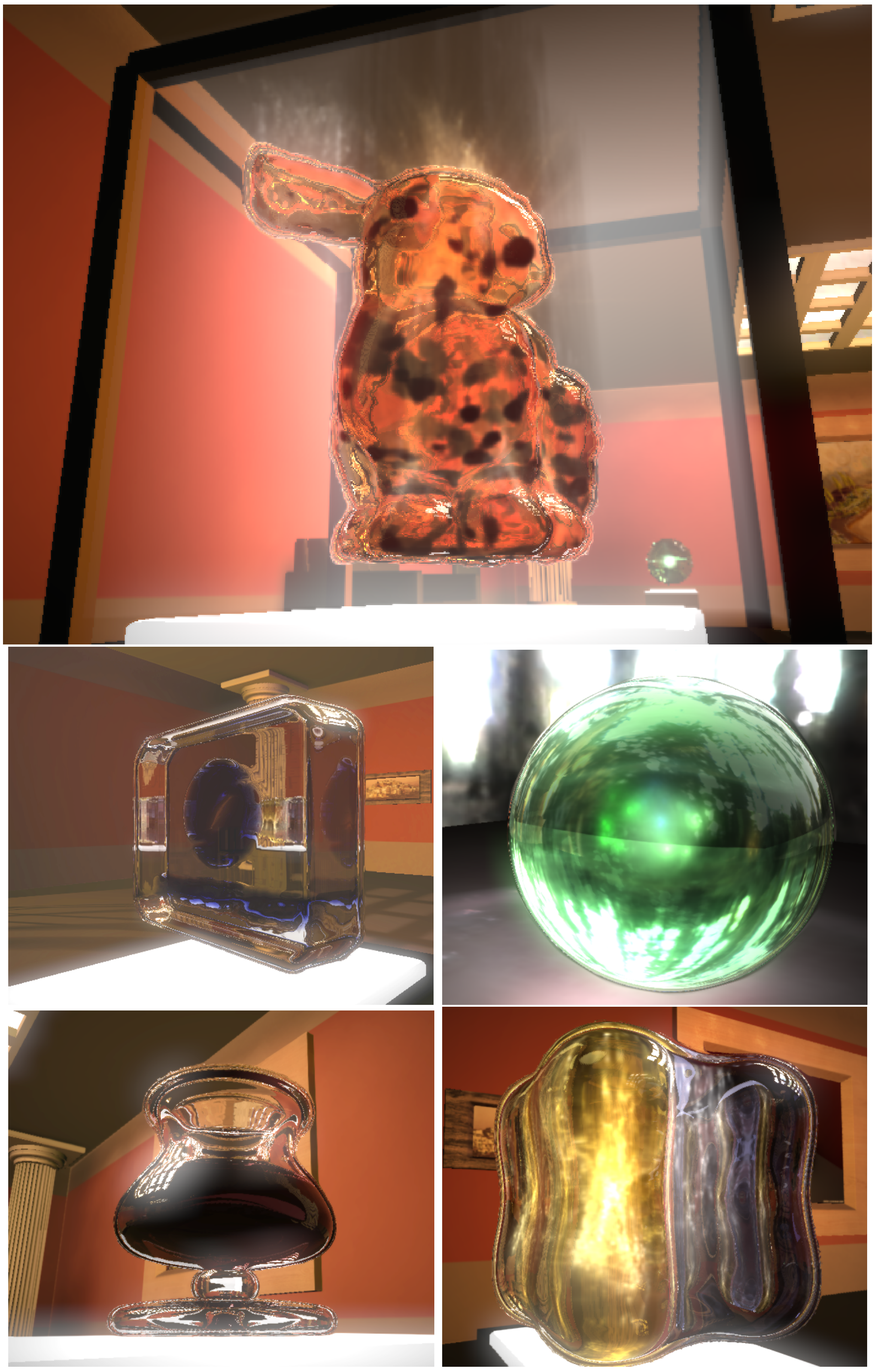


Fig. 9.6. A gallery of different objects rendered with the proposed algorithm.

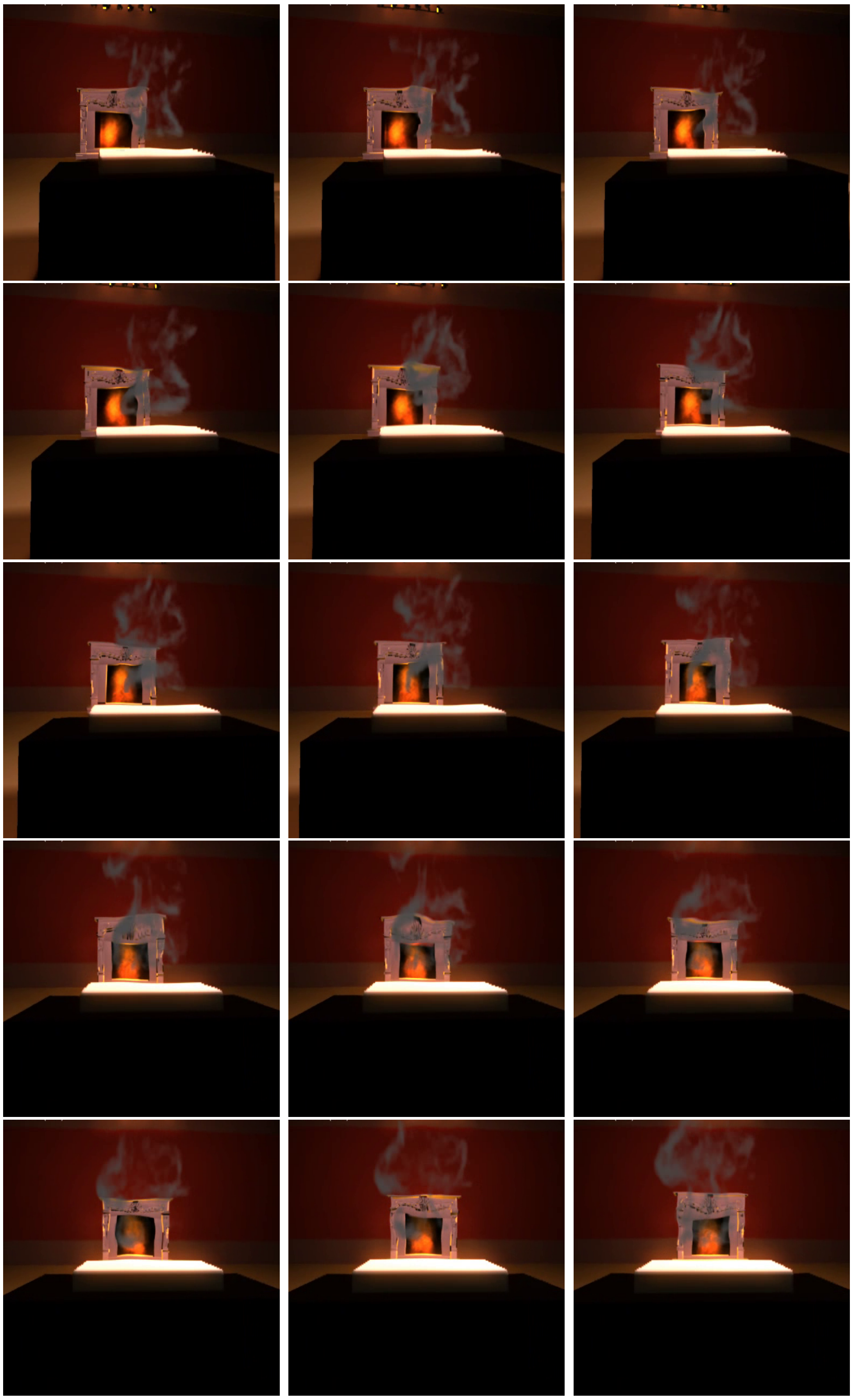


Fig. 9.7. Frames from an animation of heated airflow captured with the method described in Chapter 8 and modified to exhibit scattering and absorption properties.

10 Discussion and Conclusions

In the following we summarize our work, discussing contributions and drawbacks of the presented methods. Following that we draw conclusions and present an outlook on future work.

10.1 Summary

We first introduced a sparse-view tomographic reconstruction technique for the acquisition of dynamic, volumetric models of flames suitable for photo-realistic image synthesis. The basic tool for the achievement of this goal using only a handful of cameras is the restriction of the solution space to the visual hull of the flames. The reduced degrees of freedom of the inversion problem and the coarse proxy geometry provided by the visual hull result in good view extrapolation properties of the acquired models. However, the method has only been tested under laboratory conditions because of the amount of hardware necessary for the acquisition of the data processed by the reconstruction algorithm. Necessarily we were restricted to small-scale flames. It is to be expected that the application of the proposed method to large-scale fires like a camp fire will make it necessary to use more cameras than employed in our experimental setup. Another factor not considered in this work is the presence of opaque objects like firewood in the flames. The presence of these will have a similar effect as metal implants in medical computed tomography applications, resulting in spurious projection artifacts.

Next, we discussed an extension of the basic visual hull-restricted tomography algorithm that uses an adaptive representation of the underlying computational grid. This measure increases the effective resolution of the reconstruction. We apply the modified method to the reconstruction of thin smoke columns that are recorded under carefully chosen experimental conditions. The higher effective resolution enables reconstructing fine details like swirls in the smoke. However, it also leads to stability problems during reconstruction. The causes are analyzed and a solution is presented. In due course we develop an error projection method from the image plane of the cameras to the three-dimensional reconstruction volume. In a mathematical context, this constitutes a method for the transfer of an error measure from the codomain of a linear operator to its domain which might be applicable in more general settings as well. The adaptive reconstruction method suffers from the same shortcomings as the basic algorithm. Additionally, the scattering and absorption properties of the smoke are ignored.

The water reconstruction technique enables acquiring the surface of free-flowing bodies of water. We developed experimental conditions under which the optical path length of a refracted light ray in the water volume can be measured. This is achieved by using the effect of chemiluminescence. Based on this approach we develop a photo-consistency measure that is optimized using a weighted minimal surface. The surface evolution is implemented using the well-known level set technique. Drawbacks of this method are that the refractive index of the fluid must be known in advance, and that the result is dependent on a good initial guess, where the latter requirement is more restrictive. Additionally, the chemicals added to the fluid may change its viscosity and thus the overall behavior of the fluid.

The restriction to a single, known refractive index throughout the volume is alleviated by our reconstruction technique for inhomogeneous refractive index fields based on the tomographic reconstruction of the refractive index gradient field. We introduce a new formulation of the tomographic problem that does not require the paraxial approximation to be used, i.e., we can handle arbitrary curved light paths. The prerequisite-requisite for computing the tomographic reconstruction is an estimate of image space ray deflections caused by the refractive volume. According to synthetic tests this seems to be the limiting factor in the application of this method since the deflections have to be computed from image space measurements of distorted background information using optical flow techniques. The maximum magnitude of the refractive index fields is thus limited by the measurement setup.

The proposed rendering method for refractive objects is very general. Its capabilities exceed the data acquirable by the reconstruction methods presented in this thesis. We can render spatially varying, inhomogeneous refractive index fields exhibiting emission, absorption, single scattering, and reflection on material boundaries simultaneously. An efficient light propagation technique enables rapidly computing three-dimensional light distributions, including local light directions. This allows for rendering of anisotropic volumetric lighting effects like volume caustics in smoke, volumetric shadows and similar effects. The advantage, and simultaneously the greatest restriction, of this rendering algorithm is the volumetric representation of the objects. The power lies in the possibility for truly spatially varying material properties, allowing for a combination of visual effects that has not been demonstrated before. However, the memory consumption of the models restricts the rendering of *dynamic* models to about 3 frames per second. This number could probably be improved by a factor of two by utilizing appropriate caching schemes. We investigated wavelet compression schemes for emissive volume rendering of dynamic models [5]. However this method is not suitable for the rendering of refractive objects and thus has not been included in this thesis.

10.2 Conclusions

In this dissertation we introduced the tomographic reconstruction of natural phenomena for computer graphics purposes. The acquisition of fully three-dimensional, dynamic models of fire, smoke, and fluid flows enables free-viewpoint video applications for natural phenomena. The quality of the reconstructed, dynamic computer models is suitable for photo-realistic view synthesis which was the primary goal of this work. However, the utility of the acquired models is beyond pure rendering of the phenomena considered here. I believe that the subsequent analysis of the reconstructed models can make the development of example-based modeling algorithms possible. Furthermore, the data, especially the refractive index fields, lend themselves to the analysis for secondary information like fluid density or temperature. The color information present in the flame models might be used for pyrometry investigations. This, in turn, could lead to estimates of boundary conditions for the equations of fluid dynamics, leading in turn to better computer animations.

10.3 Future Work

The work presented so far just amounts to a first step in image-based modeling of natural phenomena. To summarize, the basic tomographic algorithm could be improved by adding support for opaque objects present in the reconstruction volume. Another goal would be to measure absorption and potentially scattering properties in addition to emissivity. Refractive index tomography and emission tomography could be combined to allow for the simultaneous measurement of the two effects.

Further research should be conducted in the direction of compressing the acquired volumetric models to facilitate real-time rendering of whole sequences of the reconstructed models. A connection to the reconstruction methods presented in this thesis is formed by wavelet bases. The feasibility of directly reconstructing using a wavelet basis should be explored. This is complicated by the fact that it is difficult to ensure a physically plausible, non-negative reconstruction result since wavelet bases contain negative components and our projection to the space of non-negative solutions cannot be employed.

On the rendering side the wavefront tracking scheme for the computation of three-dimensional light distributions with directional information seems to be a promising direction of future work. The equations describing the light propagation in their current parameterization do not lend themselves to the simulation of light-matter interactions with reflective objects. However, the viewing ray parameterization is suitable for this task. Unfortunately, the intensity computations become much more involved when using this parameterization since the simple intensity rule cannot be used anymore. Instead, the curvature of the wavefronts has to be integrated along the ray. On the other hand this would lead to a unification of refractive and reflective light transport.

A

Appendix

A.1 Integration of Rays over Basis Functions

A.1.1 Transformation of Curve Integrals

Here we give the proof for Eq. (5.8).

$$\int_{c_n} \phi_i \circ T \, dt \tag{A.1}$$

$$= \int_0^1 \phi_i \circ T \circ c_n(t) \, \Big|\Big|\frac{dc_n(t)}{dt}\Big|\Big| \, dt \tag{A.2}$$

$$\overset{(*)}{=} \frac{||\mathbf{d}_n||}{||T\mathbf{d}_n||} \int_0^1 \phi_i \circ T \circ c_n(t) ||T\mathbf{d}_n|| \, dt \tag{A.3}$$

$$\overset{(**)}{=} \frac{||\mathbf{d}_n||}{||T\mathbf{d}_n||} \int_0^1 \phi_i \circ T \circ c_n(t) \Big|\Big|\frac{d(T \circ c_n)}{dt}\Big|\Big| \, dt \tag{A.4}$$

$$= \frac{||\mathbf{d}_n||}{||T\mathbf{d}_n||} \int_{T \circ c_n} \phi_i \, dt \tag{A.5}$$

$$(*) \qquad \frac{dc_n(t)}{dt} \overset{c_n(t) linear}{=} \mathbf{d}_n, const. \tag{A.6}$$

$$(**) \qquad \frac{d(T \circ c_n)(t)}{dt} = \frac{dT}{dc_n} \frac{dc_n}{dt} \overset{\overset{T linear}{c_n linear}}{=} T\mathbf{d}_n \tag{A.7}$$

Note that this proof is only valid for a linear curve c_n and linear transformation T.

A.1.2 Polynomial to be Integrated for the Trilinear Basis Function

In the following we assume the curve c_n to be in the form $T \circ c_n(s) = p + sr$. The direction vector r is normalized. The cubic polynomial that is to be integrated for the trilinear basis function is given by:

$$\begin{aligned}
&s^3 * \ (a_1 r_x r_y r_z) + \\
&s^2 * \ (a_1(r_x r_y p_z + p_x r_y r_z + p_y r_x r_z) + \\
&\qquad a_2 r_x r_y + a_3 r_x r_z + a_4 r_y r_z) + \\
&s \ * \ (a_1(p_z p_x r_y + p_z p_y r_x + p_x p_y r_z) + \\
&\qquad a_2(p_x r_z + p_y r_x) + a_3(p_x r_z + p_z r_x) + \\
&\qquad a_4(p_y r_z + p_z r_y) + a_5 r_x + a_6 r_y + a_7 r_z) + \\
&1 \ * \ (a_1 p_x p_y p_z + a_2 p_x p_y + a_3 p_x p_y + a_4 p_y p_z + \\
&\qquad a_5 p_x + a_6 p_y + a_7 p_z + a_8)
\end{aligned} \tag{A.8}$$

The constants $a_1 \dots a_8$ are defined as

$$\begin{aligned}
a_1 &= -p_{000} - p_{001} + p_{010} + p_{011} \\
&\quad -p_{100} - p_{101} + p_{110} + p_{111} \\
a_2 &= -p_{000} + p_{001} + p_{010} - p_{011} \\
a_3 &= -p_{000} + p_{001} - p_{100} + p_{101} \\
a_4 &= -p_{000} + p_{010} - p_{100} + p_{110} \\
a_5 &= -p_{000} - p_{001} \\
a_6 &= p_{000} - p_{010} \\
a_7 &= p_{000} + p_{100} \\
a_8 &= -p_{000}
\end{aligned} \tag{A.9}$$

$p_{000} \dots p_{111}$ are the values at the corner points of the unit cube in the order of p_{zyx}, respectively. The integration has to be performed from s_1 to s_2 which are the intersections of c_n with the voxel in question in unit coordinates.

A.2 Projected Codomain L_p norm related proofs

A.2.1 Proof of the Minkowski Inequality for Projected Codomain L_p Norms

The Minkowski inequality states that

$$\left(\sum_i |x_i + y_i|^p\right)^{\frac{1}{p}} \le \left(\sum_i |x_i|^p\right)^{\frac{1}{p}} + \left(\sum_i |y_i|^p\right)^{\frac{1}{p}}.$$

In the context of proving the triangle inequality for the class of projected codomain L_p norms it reads

$$\left(\sum_i |s_i||x_i + y_i|^p\right)^{\frac{1}{p}} \le \left(\sum_i |s_i||x_i|^p\right)^{\frac{1}{p}} + \left(\sum_i |s_i||y_i|^p\right)^{\frac{1}{p}}. \tag{A.10}$$

We start with rewriting the p'th power of the left hand side of Eq. A.10:

$$\left(\sum_i |s_i||x_i+y_i||x_i+y_i|^{p-1}\right)^{\frac{1}{p}p} \leq \tag{A.11}$$

We use the standard triangle inequality to obtain

$$\sum_i |s_i| \left(|x_i||x_i+y_i|^{p-1} + |y_i||x_i+y_i|^{p-1}\right) =$$
$$\sum_i |s_i||x_i||x_i+y_i|^{p-1} + \sum_i |s_i||y_i||x_i+y_i|^{p-1} \leq$$

which is less or equal than

$$\left(\sum_i |s_i||x_i|^p\right)^{\frac{1}{p}} \left(\sum_i |s_i||x_i+y_i|^{q(p-1)}\right)^{\frac{1}{q}} + \left(\sum_i |s_i||y_i|^p\right)^{\frac{1}{p}} \left(\sum_i |s_i||x_i+y_i|^{q(p-1)}\right)^{\frac{1}{q}}$$

because of Hölders inequality which is proofed in the next subsection. Rearranging terms we obtain

$$\left[\left(\sum_i |s_i||x_i|^p\right)^{\frac{1}{p}} + \left(\sum_i |s_i||y_i|^p\right)^{\frac{1}{p}}\right] \left(\sum_i |s_i||x_i+y_i|^{qp-q}\right)^{\frac{1}{q}} \leq$$

Hölders inequality requires $\frac{1}{p}+\frac{1}{q}=1,\ \ p,q \geq 0$, therefore $qp-q=p$ and the equation becomes

$$\left[\left(\sum_i |s_i||x_i|^p\right)^{\frac{1}{p}} + \left(\sum_i |s_i||y_i|^p\right)^{\frac{1}{p}}\right] \left(\sum_i |s_i||x_i+y_i|^{p}\right)^{\frac{1}{p}\frac{p}{q}}. \tag{A.12}$$

Dividing Eqs. A.11 and A.12 by $\left(\sum_i |s_i||x_i+y_i|^p\right)^{\frac{1}{p}\frac{p}{q}}$ yields the desired result

$$\left(\sum_i |s_i||x_i+y_i|^p\right)^{\frac{1}{p}} \leq \left(\sum_i |s_i||x_i|^p\right)^{\frac{1}{p}} + \left(\sum_i |s_i||y_i|^p\right)^{\frac{1}{p}} \square.$$

A.2.2 Proof of the Hölder Inequality for Projected Codomain L_p Norms

The original Hölder inequality reads

$$\sum_i |x_i y_i| \leq \left(\sum_i |x_i|^p\right)^{\frac{1}{p}} \left(\sum_i |y_i|^q\right)^{\frac{1}{q}} .$$

Here we prove the Hölder inequality for projected codomain L_p norms

$$\sum_i |s_i||x_i y_i| \leq \left(\sum_i |s_i||x_i|^p\right)^{\frac{1}{p}} \left(\sum_i |s_i||y_i|^q\right)^{\frac{1}{q}}$$

to complete the proof of the Minkowski inequality for projected codomain L_p norms. We use a result from differential calculus as our main tool:

$$a^{\frac{1}{p}} b^{\frac{1}{q}} \leq \frac{a}{p} + \frac{b}{q} \tag{A.13}$$

Setting

$$\begin{aligned} A &= \left(\sum_i |s_i||x_i|^p\right)^{\frac{1}{p}} \\ B &= \left(\sum_i |s_i||y_i|^q\right)^{\frac{1}{q}} \\ a &= \frac{|s_i||x_i|^p}{A^p} \\ b &= \frac{|s_i||y_i|^q}{B^q} \end{aligned}$$

and inserting in Eq. A.13, we obtain

$$\frac{|s_i||x_i y_i|}{AB} \leq \frac{|s_i||x_i||y_i|}{AB} \leq \frac{|s_i||x_i|^p}{pA^p} + \frac{|s_i||y_i|^q}{qB^q},$$

where the first inequality stems from part of the proof of the triangle inequality for real numbers. Summing over i and expanding A and B yields

$$\frac{\sum_i |s_i||x_i y_i|}{\left(\sum_i |s_i||x_i|^p\right)^{\frac{1}{p}} \left(\sum_i |s_i||y_i|^q\right)^{\frac{1}{q}}} \leq$$
$$\frac{\sum_i |s_i||x_i|^p}{p\sum_i |s_i||x_i|^p} + \frac{\sum_i |s_i||y_i|^q}{q\sum_i |s_i||y_i|^q} =$$
$$\frac{pq}{p+q}\,\frac{\left(\sum_i |s_i||x_i y_i|\right)\left(\sum_i |s_i||x_i|^p\right)\left(\sum_i |s_i||y_i|^q\right)}{\left(\sum_i |s_i||x_i|^p\right)^{\frac{1}{p}} \left(\sum_i |s_i||y_i|^q\right)^{\frac{1}{q}}} \leq$$
$$\left(\sum_i |s_i||x_i|^p\right)\left(\sum_i |s_i||y_i|^q\right).$$

With $\frac{pq}{p+q} = 1$ (again because of $\frac{1}{p} + \frac{1}{q} = 1$) we obtain

$$\left(\sum_i |s_i||x_i y_i|\right)\left(\sum_i |s_i||x_i|^p\right)^{1-\frac{1}{p}}\left(\sum_i |s_i||y_i|^q\right)^{1-\frac{1}{q}} \leq$$
$$\left(\sum_i |s_i||x_i|^p\right)\left(\sum_i |s_i||y_i|^q\right).$$

Dividing by $\left(\sum_i |s_i||x_i|^p\right)^{1-\frac{1}{p}}\left(\sum_i |s_i||y_i|^q\right)^{1-\frac{1}{q}}$ completes the proof:

$$\sum_i |s_i||x_i y_i| \leq \left(\sum_i |s_i||x_i|^p\right)^{\frac{1}{p}}\left(\sum_i |s_i||y_i|^q\right)^{\frac{1}{q}} \square.$$

A.3 Refraction Computation at a Boundary

The following formulas describe refraction at a well defined boundary, refraction from both sides and total reflection are included. Fig. A.1 shows the quantities involved in the computation. The direction vector $\mathbf{d}^i$ is a unit vector incident on the surface and the resulting direction vector $\mathbf{d}^o$ is also a unit vector, exiting the surface.

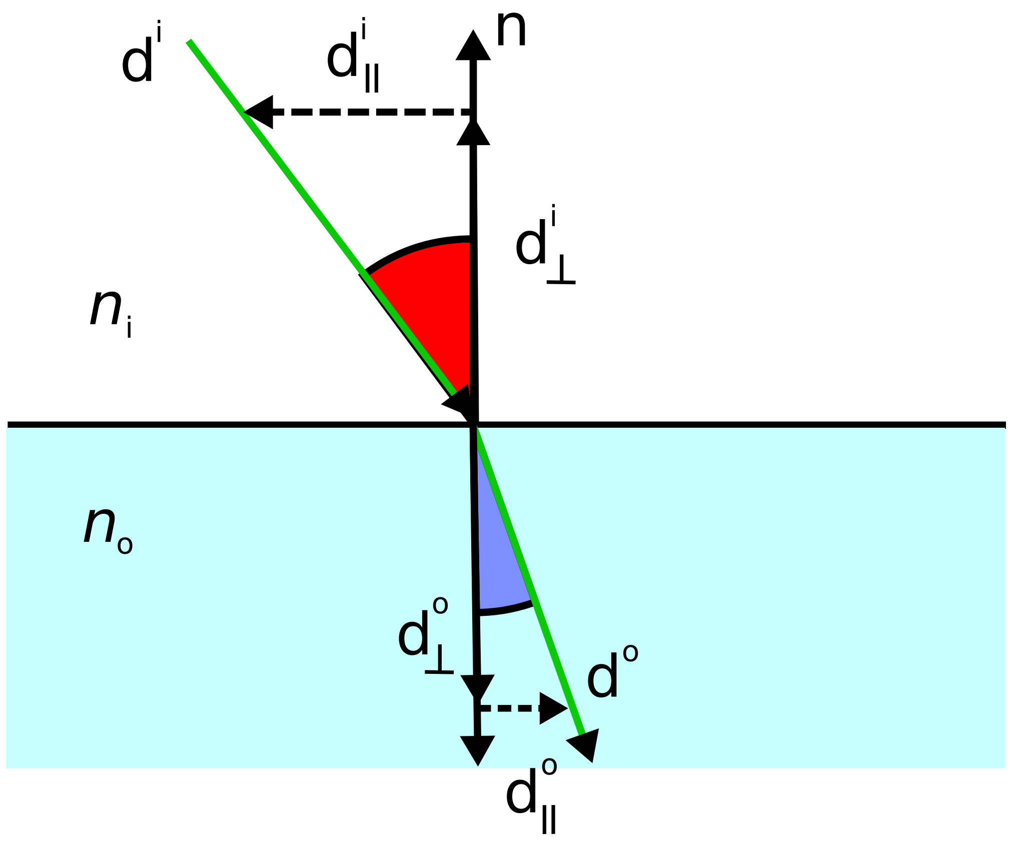


Fig. A.1. The figure shows the geometric quantities used in the refraction formula.

$$\begin{aligned}
\mathbf{n} &= \frac{\nabla u}{|\nabla u|} \\
\mathbf{d}^i_\perp &= (-\mathbf{d}^i \cdot \mathbf{n})\mathbf{n} \\
\mathbf{d}^i_\parallel &= -(\mathbf{d}^i + \mathbf{d}^i_\perp) \\
\sin\theta_o &= \begin{cases} |\mathbf{d}^i_\parallel|\frac{n_i}{n_o} & (\mathbf{n}\cdot\mathbf{d}^i) > 0 \\ |\mathbf{d}^i_\parallel|\frac{n_o}{n_i} & (\mathbf{n}\cdot\mathbf{d}^i) < 0, |\mathbf{d}^i_\parallel| < \frac{n_i}{n_o} \\ |\mathbf{d}^i_\parallel| & (\mathbf{n}\cdot\mathbf{d}^i) < 0, |\mathbf{d}^i_\parallel| \geq \frac{n_i}{n_o} \end{cases} \\
\cos\theta_o &= \sqrt{1 + \sin{\theta_o}^2} \\
s &= \begin{cases} -1 & (\mathbf{n}\cdot\mathbf{d}^i) < 0, |\mathbf{d}^i_\parallel| \geq \frac{n_i}{n_o} \\ 1 & \text{else} \end{cases}
\end{aligned}$$

$$f(\mathbf{d}^i, u) = \mathbf{d}^o = -s\frac{\mathbf{d}^i_\perp}{|\mathbf{d}^i_\perp|}\sin\theta_o - \frac{\mathbf{d}^i_\parallel}{|\mathbf{d}^i_\parallel|}\cos\theta_o \tag{A.14}$$

A.4 Ray parameterizations for Rendering

We derive a constant spatial and a constant temporal step size parameterization of the ray equation of geometric optics. Eq. (9.8) is derived by combining the eikonal equation

$$|\nabla S| = n \tag{A.15}$$

and the equation of a particle moving normal to the wavefronts $S = const.$

$$\frac{d\mathbf{x}}{ds} = \frac{\nabla S}{|\nabla S|}. \tag{A.16}$$

S is a solution of the eikonal equation and iso-surfaces of this function are called wavefronts. They are surfaces of constant travel time from the light source. The derivation of Eq. (9.8) can be found in [28].

A.4.1 Parameterization with constant spatial step size

Using Eq. (A.16) we immediately have

$$|\frac{d\mathbf{x}}{ds}|^2 = \frac{d\mathbf{x}}{ds} \cdot \frac{d\mathbf{x}}{ds} = 1. \tag{A.17}$$

Inserting Eq. (A.15) into Eq. (A.16) yields

$$n\frac{d\mathbf{x}}{ds} = \nabla S. \tag{A.18}$$

Setting $\mathbf{d} = \nabla S$ we obtain a parameterization with constant spatial step size ds, Eqs. (9.9) and (9.10).

A.4.2 Parameterization with constant temporal step size

We are looking for a parameterization where

$$\frac{dS}{dt} = \nabla S \cdot \frac{d\mathbf{x}}{dt} = 1, \tag{A.19}$$

i.e. the infinitesimal change of the eikonal function S with respect to the parameter t is constant. Inserting Eq. (A.18) into Eq. (A.19) yields

$$\frac{1}{n} = \frac{d\mathbf{x}}{ds} \cdot \frac{d\mathbf{x}}{dt} = \frac{d\mathbf{x}}{ds} \cdot \frac{d\mathbf{x}}{ds}\frac{ds}{dt} = \frac{ds}{dt}, \tag{A.20}$$

where the last identity is due to Eq. (A.17). Using this result we perform a change of parameters using the chain rule and obtain Eqs. (9.14) and (9.15) from Eqs. (9.9) and (9.10).

References

1. Edward H. Adelson and James R. Bergen. The Plenoptic Function and the Elements of Early Vision. *Computational Models of Visual Processing*, pages 3–20, 1991.
2. S. Agarwal, S.P. Mallick, D.J. Kriegman, and S. Belongie. On refractive optical flow. In *ECCV04*, pages Vol II: 483–494, 2004.
3. A.K. Agrawal, B.W. Albers, and D.W. Griffin. Abel inversion of deflectometric measurements in dynamic flows. *Applied Optics*, 38(15):3394–3398, May 1999.
4. Lukas Ahrenberg, Ivo Ihrke, and Marcus Magnor. A mobile system for multi-video recording. In *1st European Conference on Visual Media Production (CVMP)*, pages 127–132, Savoy Place, London, UK, March 2004. Institution of Electrical Engineers.
5. Lukas Ahrenberg, Ivo Ihrke, and Marcus Magnor. Volumetric reconstruction, compression and rendering of natural phenomena from multi-video data. In E Gröller and I Fujishiro, editors, *International Workshop on Volume Graphics 2005*, pages 83–90, Stony Brook, New York, USA, 2005. Eurographics.
6. K. S. Arun, T. S. Huang, and S. D. Blostein. Least Squares Fitting of Two 3-D Point Sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 9(5), 1987.
7. James R. Arvo. Backward Ray Tracing. In *ACM SIGGRAPH '86 Course Notes - Developments in Ray Tracing*, volume 12, 1986.
8. Ali Azarbayejani and Alex Pentland. Camera self-calibration from one point correspondence. Technical Report 341, MIT Media Lab, 1995.
9. S. Baker and I. Matthews. Lucas-Kanade 20 Years On: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, February 2004.
10. Ziv Bar-Joseph, Ran El-Yaniv, Dani Lischinski, and Michael Werman. Texture Mixing and Texture Movie Synthesis Using Statistical Learning. *IEEE Transactions on Visualization and Computer Graphics*, 7(2):120–135, April-June 2001.
11. João P. Barreto and Kostas Daniilidis. Wide Area Multiple Camera Calibration and Estimation of Radial Distortion. In *Proc. of OMNIVIS*, May 2004.

12. R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition.* SIAM, Philadelphia, PA, 1994.
13. R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition.* SIAM, Philadelphia, PA, 1994.
14. J. Barron, D. Fleet, and S. Beauchemin. Performance of Optical Flow Techniques. *International Journal of Computer Vision*, 12(1):43–77, February 1994.
15. M. Bathia, W. C. Karl, and A. S. Willsky. A Wavelet-Based Method for Multiscale Tomographic Reconstruction. *Transactions on Medical Imaging*, 15(1):92–101, February 1996.
16. Bryce E. Bayer. Color imaging array, 1976. U.S. Patent No. 3,971,065.
17. Philippe Beaudoin, Sébastien Paquet, and Pierre Poulin. Realistic and Controllable Fire Simulation. In *Graphics Interface 2001*, pages 159–166, June 2001.
18. Jean-David Benamou. Big ray tracing: Multivalued travel time field computation using viscosity solutions of the eikonal equation. *Journal of Computational Physics*, 128(2):463–474, 1996.
19. Marc Berger, Terry Trout, and Nancy Levit. Ray tracing mirages. *IEEE CGAA*, 10(3):36–41, 1990.
20. Richard Berry and James Burnell. *The Handbook of Astronomical Image Processing.* Willmann-Bell, Inc., 2000.
21. Kiran S. Bhat, Steven M. Seitz, Jessica K. Hodgins, and Pradeep K. Khosla. Flow-based Video Synthesis and Editing. *Proceedings of ACM SIGGRAPH*, pages 360–363, August 2004.
22. Rahul Bhotika, David J. Fleet, and Kiriakos N. Kutulakos. A Probabilistic Theory of Occupancy and Emptiness. *Proceedings of European Conference on Computer Vision*, 3:112–132, 2002.
23. Åke Björck. *Numerical Methods for Least Squares Problems.* Society of Industrial and Applied Mathematics, 1996.
24. Volker Blanz, C.Basso, T.Poggio, and T.Vetter. Reanimating Faces in Images and Video. *Proceedings of Eurographics '03*, 22(3):641–650, 2003.
25. Volker Blanz and Thomas Vetter. A Morphable Model For The Synthesis Of 3D Faces. *Proceedings of ACM SIGGRAPH*, pages 187–194, 1999.
26. J.F. Blinn. Light reflection functions for simulation of clouds and dusty surfaces. In *Proc. of SIGGRAPH'82*, pages 21–29, 1982.
27. Jeremy S. De Bonet and Paul A. Viola. Roxels: Responsibility Weighted 3D Volume Reconstruction. In *Proc. International Conference on Computer Vision (ICCV '99)*, pages 418–425, 1999.
28. Max Born and Emil Wolf. *Principles of Optics, seventh edition.* Cambridge University Press, 1999.
29. Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured Lumigraph Rendering. In *SIGGRAPH '01: Proceedings*

of the 28th annual conference on Computer Graphics and Interactive Techniques, pages 425 – 432, 2001.
30. S. Buske and U. Kästner. Efficient and Accurate Computation of Seismic Traveltimes and Amplitudes . *Geophysical Prospecting*, 52:313–322, 2004.
31. B. Cabral, N. Cam, and J. Foran. Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. *Proc. Symposium on Volume Visualization (VolVis'94)*, pages 91–98, 1994.
32. Nathan A. Carr, Jesse D. Hall, and John C. Hart. The ray engine. In *Proc. of Graphics Hardware*, pages 37–46, 2002.
33. Xing Chen, James Davis, and Philipp Slusallek. Wide Area Camera Calibration Using Virtual Calibration Objects. In *Proceedings of CVPR*, volume 2, pages 520–527, 2000.
34. Y. G. Chen, Y. Giga, and S. Goto. Uniqueness and existence of viscosity solutions of generalized mean curvature flow. *Journal of Differential Geometry*, 33:749–786, 1991.
35. C.H.Perry and R.W.Picard. Synthesizing flames and their spreading. *Fifth Eurographics Workshop on Animation and Simulation*, pages 105–117, September 1994.
36. Steven Collins. *Wavefront Tracking for Global Illumination Solutions.* PhD thesis, 1997.
37. R. L. Cook and T. DeRose. Wavelet Noise. *ACM Trans. Graphics (Proc. ACM SIGGRAPH)*, 24(3):803–811, 2005.
38. R. A. Crawfis and Nelson Max. Texture Splats for 3D scalar and vector field visualization. *Proceedings of the conference on Visualization*, pages 91–98, October 1993.
39. S.B. Dalziel, G.O. Hughes, and B.R. Sutherland. Whole-field density measurements by'synthetic schlieren'. *Experiments in Fluids*, 28:322–335, 2000.
40. Kyle J. Daun, Kevin A. Thomson, Fengshan Liu, and Greg J. Smallwood. Deconvolution of axisymmetric flame properties using tikhonov regularization. *Applied Optics*, 45(19):4638–4646, 2006.
41. Y. Dobashi, K. Kaneda, H. Yamashita, T. Okita, and T. Nishita. A Simple, Efficient Method for Realistic Animation of Clouds. *Proceedings of ACM SIGGRAPH*, pages 19–28, August 2000.
42. Phil Dutre, Kavita Bala, and Philippe Bekaert. *Advanced Global Illumination, 2nd Edition.* A K Peters, Natick, MA, 2006.
43. David S. Ebert and Richard E. Parent. Rendering and Animation of Gaseous Phenomena by Combining Fast Volume and Scanline A-buffer Techniques. *Computer Graphics*, 24(4):357–366, August 1990.
44. G.E. Elsinga, B.W. Oudheusden, F. Scarano, and D.W. Watt. Assessment and application of quantitative schlieren methods: Calibrated color schlieren and background oriented schlieren. *Experiments in Fluids*, 36:309–325, 2004.
45. Björn Enquist and Olof Runborg. Computational High Frequency Wave Propagation. *Acta Numerica*, 12:181–266, 2003.
46. M. Ernst, T. Akenine Moeller, and H. Wann Jensen. Interactive rendering of caustics using interpolated warped volumes. In *Proc. of GI*, pages 87–96, 2005.

47. G.W. Faris and R.L. Byer. Three-dimensional beam-deflection optical tomography of a supersonic jet. *Applied Optics*, 27(24):5202–5215, December 1988.
48. Ron Fedkiw, Jos Stam, and Henrik Wann Jensen. Visual Simulation of Smoke. *Proceedings of ACM SIGGRAPH*, pages 15–22, August 2001.
49. Bryan E. Feldman, James F. O'Brien, and Okan Arikan. Animating Suspended Particle Explosions. *Proceedings of ACM SIGGRAPH*, 22(3):708–715, 2003.
50. Nick Foster and Dimitris Metaxas. Realistic Animation of Liquids. *Graphical Models and Image Processing*, 85(5):471–483, 1996.
51. Nick Foster and Dimitris Metaxas. Modeling the motion of a hot, turbulent gas. *Proceedings of ACM SIGGRAPH*, pages 181–188, August 1997.
52. Christian Fuchs, Tongbo Chen, Michael Goesele, Holger Theisel, and Hans-Peter Seidel. Volumetric Density Capture From a Single Image. In *Proceedings of International Workshop on Volume Graphics*, pages 17–22, 2006.
53. Thomas Funkhouser, Michael Kazhdan, Philip Shilane, Patrick Min, William Kiefer, Ayellet Tal, Szymon Rusinkiewicz, and David Dobkin. Modeling by Example. *ACM Transactions on Graphics (TOG)* , 23(3):652–663, August 2004.
54. David T. Gering and William M. Wells III. Object Modeling using Tomography and Photography. In *Proc. of IEEE Workshop on Multi-View Modeling and Analysis of Visual Scenes*, pages 11–18, June 1999.
55. J. H. Gladstone and T. P. Dale. Researches on the refraction, dispersion and sensitiveness of liquids. *Phil. Trans. Roy. Soc. London*, 153, 1863.
56. B. Goldlücke and M. Magnor. Hardware-accelerated dynamic light field rendering. In *Vision, Modeling and Visualisation*, 2002.
57. Bastian Goldlücke, Ivo Ihrke, Christian Linz, and Marcus Magnor. Weighted minimal hypersurface reconstruction. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, accepted for publication.
58. Bastian Goldluecke. *Multi-Camera Reconstruction and Rendering for Free-Viewpoint Video*. PhD thesis, 2005.
59. Bastian Goldluecke and Marcus Magnor. Weighted Minimal Hypersurfaces and Their Applications in Computer Vision. In *Proceedings of ECCV (2)*, volume 3022 of *Lecture Notes in Computer Science*, pages 366–378, Prague, Czech Republic, May 2004. Springer.
60. S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen. The Lumigraph. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer Graphics and Interactive Techniques*, pages 43–54, 1996.
61. P. S. Greenberg, R. B. Klimek, and D. R. Buchele. Quantitative rainbow schlieren deflectometry. *Applied Optics*, 34(19):3810–3822, 1995.
62. George J. Grevera. The Dead Reckoning Signed Distance Transform. In *Computer Vision and Image Understanding*, volume 95, pages 317–333, 2004.
63. Eduard Gröller. Nonlinear ray tracing: visualizing strange worlds. *The Visual Computer*, 11(5):263–274, 1995.
64. Markus Gross, Stephan Würmlin, Martin Naef, Edouard Lambouray, Christian Spagno, Andreas Kunz, Esther Koller-Meier, Tomas Svoboda, Luc Van Gool, Silke Lang, Kai Strehlke, Andrew Vande Moere, and Oliver Staadt. blue-c: A Spatially Immersive Display and 3D Video Portal for Telepresence. In *Proceedings of ACM SIGGRAPH*, pages 819–827, 2003.

65. J. Günther, I. Wald, and P. Slusallek. Realtime caustics using distributed photon mapping. In *Proc. of EGSR*, pages 111–121, 2004.
66. S. Guy and C. Soler. Graphics gems revisited: fast and physically-based rendering of gemstones. In *Proc. of ACM SIGGRAPH*, pages 231–238, 2004.
67. Jacques Hadamard. Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton University Bulletin*, pages 49–52, 1902.
68. Z.S. Hakura and Snyder. J.M. Realistic reflections and refractions on graphics hardware with hybrid rendering and layered environment maps. In *Proc. of EGSR*, pages 289–300, 2001.
69. Per Christian Hansen. *Rank-Deficient and Discrete Ill-Posed Problems.* Society of Industrial and Applied Mathematics, 1998.
70. M.J. Harris, W.V. Baxter, T. Scheuermann, and A. Lastra. Simulation of cloud dynamics on graphics hardware. In *Proc. of Graphics Hardware*, pages 92–101, 2003.
71. Richard Hartley and Andrew Zisserman. *Multiple View Geometry.* Cambridge University Press, 2000.
72. Samuel W. Hasinoff. Three-Dimensional Reconstruction of Fire from Images. MSc Thesis, University of Toronto, Department of Computer Science, 2002.
73. Samuel W. Hasinoff and Kiriakos N. Kutulakos. Photo-Consistent 3D Fire by Flame-Sheet Decomposition. In *In Proc. 9th IEEE International Conference on Computer Vision (ICCV '03)*, pages 1184–1191, 2003.
74. Samuel W. Hasinoff and Kiriakos N. Kutulakos. Photo-consistent reconstruction of semi-transparent scenes by density sheet decomposition (to appear). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9), May 2007.
75. Tim Hawkins, Per Einarsson, and Paul Debevec. Acquisition of time-varying participating media. In *Proceedings of ACM SIGGRAPH*, pages 812–815, 2005.
76. J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. In *CVPR 97*, pages 1106–1112, 1997.
77. V. Henson, M. Limber, S. McCormick, and B. Robinson. Multilevel image reconstruction with natural pixels. *SIAM J. Sci. Comp.*, 17:193–216, 1996.
78. Louis G. Henyey and Jesse L. Greenstein. Diffuse Radiation in the Galaxy. *Astrophysical Journal*, 93:70–83, 1941.
79. Jürgen Hofmann, Alexander Flisch, and Andreas Obrist. Adaptive CT scanning - mesh based optimisation methods for industrial X-ray computed tomography applications. *NDT&E International*, 37:271–278, 2004.
80. J. Höhle. Reconstruction of the underwater object. *Photogrammetric Engineering*, pages 948–954, 1971.
81. B. Horn and B. Schunck. Determining Optical Flow. *Artificial Intelligence*, 17:185–203, 1981.
82. W. L. Howes. Rainbow schlieren and its applications. *Applied Optics*, 23(4):2449–2460, 1984.
83. ICC. Specification icc.1:2003-09, file format for color profiles (version 4.1.0), international color consortium, 2003.
84. Insung Ihm, Byungkwon Kang, and Deukhyun Cha. Animation of reactive gaseous fluids through chemical kinetics. *ACM Siggraph / Eurographics Symposium Proceedings, Symposium on Computer Animation*, pages 203–212, 2004.

85. Ivo Ihrke, Lukas Ahrenberg, and Marcus Magnor. External camera calibration for synchronized multi-video systems. *Journal of WSCG*, 12(1-3):537–544, January 2004.
86. Ivo Ihrke, Bastian Goldluecke, and Marcus Magnor. Reconstructing the geometry of flowing water. In *International Conference on Computer Vision 2005*, pages 1055–1060, Beijing, PRC, 2005. IEEE.
87. Ivo Ihrke and Marcus Magnor. Image-Based Tomographic Reconstruction of Flames. *ACM Siggraph / Eurographics Symposium Proceedings, Symposium on Computer Animation*, pages 367–375, June 2004.
88. Ivo Ihrke and Marcus Magnor. Adaptive grid optical tomography. In Emanuele Trucco and Mike Chantler, editors, *Vision, Video, and Graphics 2005*, pages 141–148, Edinburgh, UK, 2005. Eurographics.
89. Ivo Ihrke and Marcus Magnor. Adaptice grid optical tomography. *Graphical Models*, 68:484–495, 2006.
90. Ivo Ihrke, Gernot Ziegler, Art Tevs, Christian Theobalt, Marcus Magnor, and Hans-Peter Seidel. Eikonal rendering: Efficient light transport in refractive objects. *ACM Trans. on Graphics (SIGGRAPH'07)*, page to appear, 2007.
91. Henrik Wann Jensen. *Realistic Image Synthesis Using Photon Mapping.* AK Peters, 2001.
92. Henrik Wann Jensen and Per H. Christensen. Efficient simulation of light transport in scences with participating media using photon maps. In *Proc. of SIGGRAPH'98*, pages 311–320. ACM Press, 1998.
93. Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan. A practical model for subsurface light transport. In *Proc. of SIGGRAPH'01*, pages 511–518. ACM Press, 2001.
94. Neel Joshi, Bennet Wilburn, V. Vaish, Marc Levoy, and M. Horowitz. Automatic color calibration for large camera arrays. Technical Report CS2005-0821, UCSD CSE, 2005.
95. J.T. Kajiya and B.P. Von Herzen. Ray tracing volume densities. In *Proc. of SIGGRAPH'84*, pages 165–174, 1984.
96. A. C. Kak and Malcolm Slaney. *Principles of Computerized Tomographic Imaging.* Society of Industrial and Applied Mathematics, 2001.
97. Sing Bing Kang. Radial Distortion Snakes. In *IAPR Workshop on Machine Vision Applications (MVA2000)*, pages 603–606, Nov. 2000.
98. Theodore Kim, Michael Henson, and Ming C. Lin. A Hybrid Algorithm for Modeling Ice Formation. *ACM Siggraph / Eurographics Symposium Proceedings, Symposium on Computer Animation*, pages 305–314, 2004.
99. Ron Kimmel. Demosaicing: Image Reconstruction from Color CCD Samples. *IEEE Trans. on Image Processing*, 8(9):1221–1228, 2000.
100. Doris H. U. Kochanek and Richard H. Bartels. Interpolating Splines with Local Tension, Continuity, and Bias Control. In *Proceedings of ACM SIGGRAPH*, volume 18, pages 33–41, 1984.
101. G. Krawczyk, K. Myszkowski, and H.-P. Seidel. Perceptual effects in real-time tone mapping. In *Proc. of Spring Conference on Computer Graphics*. ACM, 2005.

102. Grzegorz Krawczyk, Michael Goesele, and Hans-Peter Seidel. Photometric Calibration of High Dynamic Range Cameras. Research Report MPI-I-2005-4-005, Max-Planck-Institut für Informatik, May 2005.
103. Jens Krüger, Peter Kipfer, Polina Kondratieva, and Rüdiger Westermann. A particle system for interactive visualization of 3d flows. *IEEE TVCG*, 11(6):744–756, 2005.
104. V. Kumar and T. Poggio. Learning based approach to estimation of morphable model parameters, 2000.
105. Kiriakos N. Kutulakos and Steven M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, July-August 2000.
106. Kiriakos N. Kutulakos and Eron Steger. A Theory of Refractive and Specular 3D Shape by Light-Path Triangulation. In *In Proc.10th IEEE International Conference on Computer Vision (ICCV)*, pages 1448–1455, 2005.
107. Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaaron Bobick. Graphcut Textures: Image and Video Synthesis Using Graph Cuts. *ACM Transactions on Graphics (TOG)*, 22(3):277–286, July 2003.
108. Arnauld Lamorlette and Nick Foster. Structural Modeling of Flames for a Production Environment. *Proceedings of ACM SIGGRAPH*, pages 729–735, August 2002.
109. Alexis Lamouret and Michiel van de Panne. Motion Synthesis by Example. *Proceedings of the Eurographics Workshop on Computer Animation and Simulation '96*, pages 199–212, 1996.
110. C. A. Laroche and M. A. Prescott. Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients, 1994. U.S. Patent No. 5,373,322.
111. A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Recognition*, 16(2):150–162, February 1994.
112. Hendrik P. A. Lensch. *Efficient, Image-Based Appearance Acquisition of Real-World Objects.* Cuvillier Verlag, Göttingen, Germany, December 2003.
113. N. Levi and M. Werman. The Viewing Graph. In *Proceedings of CVPR*, volume 1, pages 518–522, 2003.
114. Marc Levoy and Pat Hanrahan. Light Field Rendering. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer Graphics and Interactive Techniques*, pages 31–42, 1996.
115. Mark A. Limber, Tom A. Manteuffel, and Stephen F. McCormick. Optimal Resolution in Maximum Entropy Image Reconstruction from Projections with Multigrid Acceleration. In *Proceedings of the Sixth Annual Copper Mountain Conference on Multigrid Methods*, pages 361–375, 1993.
116. Frank Losasso, Frederic Gibou, and Ron Fedkiw. Simulation of Water and Smoke with an Octree Data Structure. *ACM Transactions on Graphics*, 23(3):457–462, August 2004.
117. M.I.A. Lourakis and A.A. Argyros. The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm. Technical Report 340, Institute of Computer Sci-

ence - FORTH, Heraklion, Crete, Greece, Aug. 2004. Available from `http://www.ics.forth.gr/~lourakis/sba`.
118. B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. Seventh International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.
119. Hans-Gerd Maas. New Developments in Multimedia Photogrammetry. In A. Grün and H. Kahmen, editors, *Optical 3D Measurement Techniques III.* Wichmann Verlag, 1995.
120. M. Magnor, K. Hildebrand, A. Linţu, and A. Hanson. Reflection Nebula Visualization. In *Proc.of IEEE Visualization*, pages 255–262, 2005.
121. Marcus A. Magnor. *Video-Based Rendering.* A K Peters, Ltd., 2005.
122. Tom Malzbender. Fourier Volume Rendering. *ACM Transaction on Graphics*, 12(3):233–250, July 1993.
123. Stephen R. Marschner. *Inverse Rendering for Computer Graphics.* PhD thesis, August 1998.
124. W. Matusik, C. Buehler, R. Raskar, S. Gortler, and L. McMillan. Image-based visual hulls. In *Proceedings of ACM SIGGRAPH*, pages 369–374, 2000.
125. W. Matusik, H. Pfister, A. Ngan, P. Beardsley, R. Ziegler, and L. McMillan. Image-based 3D photography using opacity hulls. In *Proceedings of ACM SIGGRAPH*, pages 427–436, 2002.
126. W. Matusik, Hanspeter Pfister, Remo Ziegler, Addy Ngan, and Leonard McMillan. Acquisition and rendering of transparent and refractive objects. In *Proceedings of the 13th Eurographics Workshop on Rendering*, pages 267–278, 2002.
127. Wojciech Matusik. 3D TV: A Scalable System for Real-Time Acquisition, Transmission, and Autostereoscopic Display of Dynamic Scenes. *ACM Transactions on Graphics*, 23(3):814–824, 2004.
128. Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. A Data-Driven Reflectance Model. *ACM Transactions on Graphics (TOG)* , 22(3):759–769, July 2003.
129. Nelson Max. Optical Models for Direct Volume Rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, June 1995.
130. Chris McGlone, Edward Mikhael, and James Bethel. *Manual of Photogrammetry, 5th ed.* American Society of Photogrammetry, 2004.
131. L. McMackin, R.J. Hugo, K.P. Bishop, E.Y. Chen, R.E. Pierson, and C.R. Truman. High speed optical tomography system for quantitative measurement and visualization of dynamic features in a round jet. *Experiments in Fluids*, 26:249–256, 1999.
132. G.E.A. Meier. Computerized Background-Oriented Schlieren. *Experiments in Fluids*, 33:181–187, 2002.
133. Tom Mertens, Jan Kautz, Philippe Bekaert, Hans-Peter Seidelz, and Frank Van Reeth. Interactive rendering of translucent deformable objects. In *Proc. of EGRW'03*, pages 130–140, 2003.
134. Don Mitchell and Pat Hanrahan. Illumination from curved reflectors. In *Proc. of SIGGRAPH '92*, pages 283–291, 1992.
135. A. F. Moebius. *Der Barycentrische Calcul.* Leipzig, 1827.

136. Nigel J. W. Morris and Kiriakos N. Kutulakos. Dynamic Refraction Stereo. In *In Proc.10th IEEE International Conference on Computer Vision (ICCV)*, pages 1573–1580, 2005.
137. Jane Mulligan and Kostas Daniilidis. View-independent Scene Acquisition for Tele-Presence . Technical report, CIS Technical Report, Computer and Information Science, University of Pennsylvania, July 2000.
138. H. Murase. Surface shape reconstruction of an undulating transparent object. In *ICCV90*, pages 313–317, 1990.
139. H. Murase. Surface Shape Reconstruction of a Nonrigid Transparent Object Using Refraction and Motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10):1045–1052, October 1992.
140. D. Darian Muresan and Thomas W. Parks. Optimal recovery demosaicing. In *SIP '02: Proceedings of the IASTED Conference on Signal and Image Processing*. IASTED, 2002.
141. D. Darian Muresan and Thomas W. Parks. Demosaicing using Optimal Recovery. *IEEE Trans. on Image Processing*, 14(2):267–278, 2005.
142. F. Kenton Musgrave. Ray tracing mirages. *IEEE CGAA*, 10(6):10–12, 1990.
143. Harsh Nanda and Ross cutler. Practical calibrations for a real-time digital omnidirectional camera. Proceedings of CVPR, Technical Sketch, 2001.
144. Duc Quang Nguyen, Ronald Fedkiw, and Henrik Wann Jensen. Physically Based Modelling and Animation of Fire. *ACM Transactions on Graphics*, 21(3):721–728, July 2002.
145. T. Nishita and E. Nakamae. Method of displaying optical effects within water using accumulation buffer. In *Proc. of SIGGRAPH*, pages 373–379. ACM Press, 1994.
146. E. Ohbuchi. A real-time refraction renderer for volume objects using a polygon-rendering scheme. In *Proc. of CGI*, pages 190–195, 2003.
147. G. Oliveira. Refractive texture mapping, part two. www.gamasutra.com/features/20001117/oliveira_01.htm, 2000.
148. S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*, volume 153 of *Applied Mathematical Sciences*. Springer-Verlag New York, 2003.
149. Stanley Osher, Li-Tien Cheng, Myungjoo Kang, Hyeseon Shim, and Yen-Hsi Tsai. Geometric Optics in a Phase-Space-Based Level Set and Eulerian Framework. *Journal of Computational Physics*, 179(2):622–648, 2002.
150. S. Parker, W. Martin, P.P.J. Sloan, P. Shirley, B. Smits, and C. Hansen. Interactive ray tracing. In *Proc. of I3D*, pages 119–126. ACM Press, 1999.
151. Ken Perlin. An Image Synthesizer. *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, 19(3):287–296, July 1985.
152. C. Pintavirooj, A. Romputtal, A. Ngamlamiad, W. Withayachumnankul, and K. Hamamoto. Ultrasonic Refractive Index Tomography. *Journal of WSCG*, 12(2):333–339, February 2004.
153. Marc Pollefeys. *Self-calibration and metric 3D reconstruction from uncalibrated image sequences*. PhD thesis, 1999.
154. Press. *Numerical Recipes in C*. Cambridge University Press, 1988.
155. W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes*. Cambridge University Press, 1992.

156. Timothy J. Purcell, Craig Donner, Mike Cammarano, Henrik Wann Jensen, and Pat Hanrahan. Photon mapping on programmable graphics hardware. In *Proc. of Graphics Hardware*, pages 41–50, 2003.
157. J. Radon. Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten. Ber. Ver. Sachs. Akad. Wiss. Leipzig, MathPhys. Kl., 69:262 277, April 1917.
158. Ravi Ramamoorthi. *A Signal-Processing Framework for Forward and Inverse Rendering*. PhD thesis, August 2002.
159. Rajeev Ramanath, Wesley E. Snyder, and Griff L. Bilbro. Demosaicking methods for Bayer color arrays. *Journal of Electronic Imaging*, 11(3):306–315, 2002.
160. Peter Rander, P. J. Narayanan, and Takeo Kanade. Virtualized Reality: Constructing Time-Varying Virtual Worlds From Real World Events. In *Proc. of IEEE Visualization '97*, pages 277–283, Oct. 1997.
161. Farrokh Rashid-Farrokhi, K. J. Ray Liu, Carlos A. Berenstein, and David Walnut. Wavelet-Based Multiresolution Local Tomography. *Transactions on Image Processing*, 10(6):1412–1430, October 1997.
162. Alex Reche, Ignacio Martin, and George Drettakis. Volumetric Reconstruction and Interactive Rendering of Trees from Photographs. *Proceedings of ACM SIGGRAPH*, pages 720–727, August 2004.
163. William T. Reeves. PARTICLE Systems - A Technique for Modeling a Class of Fuzzy Objects. In *ACM Transactions on Graphics*, volume 2, pages 91–108, April 1983.
164. Erik Reinhard, Michael Ashikhmin, Bruce Gooch, and Peter Shirley. Color Transfer between Images. *IEEE Computer Graphics and Applications*, 21(5):34–41, 2001.
165. S. Reruang, Withawat Withayachumnankul, Chuchart Pintavirooj, and Surapan Airphaiboon. 3d shape extraction using photographic tomography with its applications. In *WSCG (Posters)*, pages 153–156, 2004.
166. H. Richard and M. Raffel. Principle and applications of the background oriented schlieren (BOS) method. *Measurement Science Technology*, 12:1576–1585, 2001.
167. Mark A. Robertson, Sean Borman, and Robert L. Stevenson. Estimation-theoretic approach to dynamic range enhancement using mutliple exposures. *SPIE Journal of Electronic Imaging*, 12(2):219–228, April 2003.
168. S. T. Roweis and L. K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. In *Science*, volume 290, pages 2323–2326, 2000.
169. H.E. Rushmeier and K.E. Torrance. The zonal method for calculating light intensities in the presence of a participating medium. In *Proc. of SIGGRAPH'87*, pages 293–302, 1987.
170. Berkman Sahiner and Andrew E. Yagle. Image Reconstruction from Projections under Wavelet Constraints. *Transactions on Signal Processing*, 41(12):3579–3584, December 1993.
171. L. K. Saul and S. T. Roweis. Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds. In *Journal of Machine Learning Research*, volume 4, pages 119–155, 2003.

172. H. Schardin. Die Schlierenverfahren und ihre Anwendungen. *Ergebnisse der Exakten Naturwissenschaften*, 20:303–439, 1942. English translation available as NASA TT F-12731, April 1970 (N70-25586).
173. H. Schirmacher, W. Heidrich, and H.-P. Seidel. High-quality interactive lumigraph rendering through warping. In *Proceedings of Graphics Interface 2000*, pages 87–94, 2000.
174. Hartmut Schirmacher, Ming Li, and Hans-Peter Seidel. On-the-Fly Processing of Generalized Lumigraphs. *Computer Graphics Forum*, 20(3):165–174, 2001.
175. Arno Schödl, Richard Szeliski, David H. Salesin, and Irfan Essa. Video Textures. *Proceedings of ACM SIGGRAPH*, pages 489–498, August 2000.
176. H. Schultz. Retrieving Shape Information from Multiple Images of a Specular Surface. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):195–201, February 1994.
177. A. Schwarz. Multi-tomographic flame analysis with a schlieren apparatus. *Meas. Sci. Technol.*, 7:406–413, 1996.
178. F. J. Seron, D. Guiterrez, G. Guiterrez, and E. Cerezo. Visualizing sunsets through inhomogenous atmospheres. In *Proc. of CGI*, pages 349–356, 2004.
179. J. A. Sethian. *Level Set Methods and Fast Marching Methods.* Monographs on Applied and Computational Mathematics. Cambridge University Press, 2nd edition, 1999.
180. G. S. Settles. *Schlieren and Shadowgraph Techniques.* Experimental Fluid Mechanics. Springer, 2001.
181. Peter-Pike J. Sloan, Charles F. Rose III, and Micheal F. Cohen. Shape by Example. *Proceedings of the 2001 symposium on Interactive 3D Graphics*, pages 135–143, 2001.
182. Milan Sonka, Vaclav Hlavac, and Roger Boye. *Image Processing, Analysis and Machine Vision.* PWS Publishing, second edition, 1999.
183. Jos Stam. Multiple Scattering as a Diffusion Process. In *Proc. of EGSR*, pages 41–50, 1995.
184. Jos Stam. Stable Fluids. *Proceedings of ACM SIGGRAPH*, pages 121–128, August 1999.
185. Jos Stam and Eugene Fiume. Depicting Fire and Other Gaseous Phenomena Using Diffusion Processes. *Proceedings of ACM SIGGRAPH*, pages 129–136, August 1995.
186. Jos Stam and Eric Languénou. Ray-tracing in non-constant media. In *Proc. of EGSR*, pages 225–ff, 1996.
187. J. Starck and A. Hilton. Towards a 3d virtual studio for human appearance capture. In *Proceedings of Video, Vision and Graphics (VVG) '03,*, pages 17–24. IMA, 2003.
188. Timo Stich and Marcus Magnor. Learning Flames. *Procc. 10th International Fall Workshop - Vision, Modeling and Visualization (VMV)*, pages 65–70, 2005.
189. Timo Stich and Marcus Magnor. Keyframe Animation From Video. *International Conference on Image Processing (ICIP)*, pages ??–??, 2006.
190. Richard Szeliski and Polina Golland. Stereo Matching with Transparency and Matting. In *Proceedings Sixth International Converence on Computer Vision*, pages 517–524, 1998.

191. Martin Szummer and Rosalind W. Picard. Temporal Texture Modeling. *Proceedings of ACM SIGGRAPH*, pages 479–488, July 1996.
192. Jun-ya Takahashi, Hiromichi Takahashi, and Norishige Chiba. Image Synthesis of Flickering Scenes Including Simulated Flames. *IEICE Transactions on Information Systems*, E80-D(11):1102–1108, November 1997.
193. Y. Takai, K. Ecchu, and N.K.Takai. A cellular automaton model of particle motions and its applications. *The Visual Computer*, 11(5):240–252, 1995.
194. Albert Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation.* SIAM, 2005.
195. J. Tenenbaum. A Global Geometric Framework for Nonlinear Dimensionality Reduction. In *Science*, volume 290, pages 2319–2323, 2000.
196. Christian Theobalt, Ming Li, Marcus Magnor, and Hans-Peter Seidel. A flexible and versatile studio for synchronized multi-view video recording. In *Proceedings of Video, Vision and Graphics (VVG) '03,*, pages 9–16. IMA, 2003.
197. B. Trifonov, D. Bradley, and Wolfgang Heidrich. Tomographic Reconstruction of Transparent Objects. In *Proc. of Eurographics Symposium on Rendering 2006*, pages 51–60, 2006.
198. Borislav Trifonov, Derek Bradley, and Wolfgang Heidrich. Tomographic Reconstruction of Transparent Objects. In *In Proc. Eurographics Symposium on Rendering (EGSR)*, pages 51–60, 2006.
199. Bill Triggs, Philip McLauchlan, Richard Hartley, and Andrew Fitzgibbon. Bundle adjustment – A modern synthesis. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, LNCS, pages 298–375. Springer Verlag, 2000.
200. Roger Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4), August 1987.
201. G.J. VanWylen and R.E. Sonntag. *Fundamentals of Classical Thermodynamics.* John Wyley & Sons, 1976.
202. Henson V.E., M.A. Limber, S.F. McCormick, and B.T. Robinson. Spotlight Computed Tomography with Natural Pixels. *Proc. Fifth SIAM Conference on Applied Linear Algebra*, pages 97–101, 1994.
203. L. Venkatakrishnan and E. Meier. Density measurements using the background oriented schlieren technique. *Experiments in Fluids*, 37(237–247), 2004.
204. I. Wald, C. Benthin, P. Slusalek, T. Kollig, and A. Keller. Interactive global illumination using fast ray tracing. In *Proc. of EGSR*, pages 15–24, 2002.
205. M. Wand and W. Strasser. Real-time caustics. *Computer Graphics Forum (Eurographics 2003)*, page 611ff, 2003.
206. Michael Waschbuesch, Stephan Würmlin, Daniel Cotting, Filip Sadlo, and Markus Gross. Scalable 3D Video of Dynamic Scenes. *The Visual Computer*, 21((8-10)):629–638, 2005.
207. Li-Yi Wei and Marc Levoy. Fast Texture Synthesis using Tree-structured Vector Quantization. *Proceedings of ACM SIGGRAPH*, pages 479–488, July 2000.
208. Xiaoming Wei, Wei Li, Klaus Mueller, and Arie Kaufman. Simulating Fire With Texture Splats. *Proceedings of the conference on Visualization*, pages 227–235, 2002.

209. F. J. Weinberg. *Optics of Flames: Including methods for the study of refractive index fields in combustion and aerodynamics.* Butterworths, London, 1963.
210. D. Weiskopf, T. Schafhitzel, and T. Ertl. Gpu-based nonlinear ray tracing. *Computer Graphics Forum (Eurographics 2004)*, 23(3):625–633, 2004.
211. B. Wilburn, M. Smulski, K. Lee, and M. Horowitz. The light field video camera. *SPIE Proc. Media Processors 2002*, 4674, January 2002.
212. Bennett Wilburn, Neel Joshi, Vaibhav Vaish, Eino-Ville Talvala, Emilio Antunez, Adam Barth, Andrew Adams, Mark Horowitz, and Marc Levoy. High Performance Imaging Using Large Camera Arrays. In *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2005)*, volume 24, pages 765–776, July 2005.
213. Reg Willson and Steven Shafer. What is the center of the image? *Journal of the Optical Society of America A*, 11(11):2946 – 2955, November 1994.
214. C. Wyman. An approximate image-space approach for interactive refraction. pages 1050–1053, 2005.
215. C. Wyman and S. Davis. Interactive image-space techniques for approximating caustics. In *Proceedings of ACM I3D*, pages 153–160, 2006.
216. C. Wyman, C. Hansen, and P. Shirley. Interactive caustics using local precomputed irradiance. pages 143–151, 2004.
217. Günther Wyszecki and W. S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae, 2nd ed.* Wiley & Sons, Ltd., August 2000.
218. Gary D. Yngve, James F. O'Brien, and Jessica K. Hodgins. Animating Explosions. *Proceedings of ACM SIGGRAPH*, pages 29–36, 2000.
219. Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(11), 2000.
220. C. Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High-quality video view interpolation using a layered represntation. *ACM Transactions on Graphics*, 23(3):600–608, 2004.
221. Douglas E. Zongker, Dawn M. Werner, Brian Curless, and David H. Salesin. Environment matting and compositing. In *Proceedings of ACM SIGGRAPH*, pages 205–214, 1999.
222. Andrei V. Zvyagin, K. K. M. B. Dilusha Silva, Sergey A. Alexandrov, Timothy R. Hillman, and Julian J. Armstrong. Refractive index tomography of turbid media by bifocal optical coherence refractometry. *Optics Express*, 11(25):3503–3517, December 2003.

Curriculum Vitæ - Lebenslauf

Curriculum Vitæ

December 1977		born in Dresden, Germany
September 1984 -	June 1997	Highschool degree, main subjects english and mathematics
		Alexander von Humboldt Gymnasium, Eberswalde, Germany
November 1998 -	July 2000	*Vordiplom* in Computer Science
		University of Erlangen-Nürnberg, Erlangen, Germany
September 2000 -	June 2002	*Master of Science* in Scientific Computing
		Royal Institute of Technology (KTH), Stockholm, Sweden
February 2003 -	May 2007	Ph.D. in Computer Science
		Max-Planck-Institut für Informatik, Saarbrücken, Germany
March 2005 -	June 2005	Research Stay
		Microsoft Research Asia, Beijing, PRC
October 2006 -	November 2006	Research Stay
		University of British Columbia, Vancouver, Canada

Lebenslauf

Dezember 1977		geboren in Dresden
September 1984 -	Juni 1997	Abitur
		Alexander von Humboldt Gymnasium, Eberswalde
November 1998 -	Juli 2000	*Vordiplom* Informatik
		Friedrich-Alexander Universität Erlangen-Nürnberg
September 2000 -	Juni 2002	*Master of Science* in Scientific Computing
		Royal Institute of Technology (KTH), Stockholm, Schweden
Februar 2003 -	Mai 2007	Promotion Informatik
		Max-Planck-Institut für Informatik, Saarbrücken, Germany
März 2005 -	Juni 2005	Forschungsaufenthalt
		Microsoft Research Asia, Peking, VR China
Oktober 2006 -	November 2006	Forschungsaufenthalt
		University of British Columbia, Vancouver, Kanada