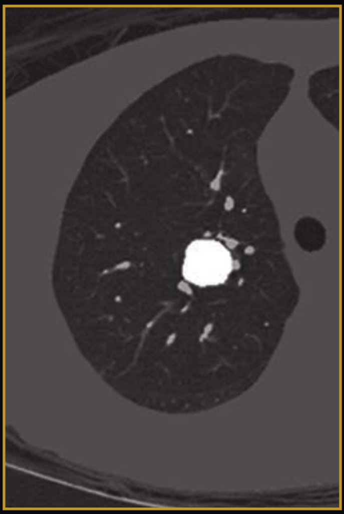
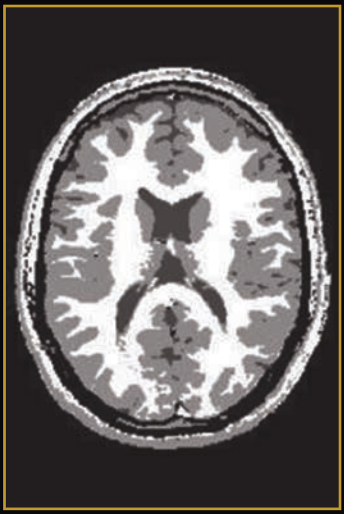
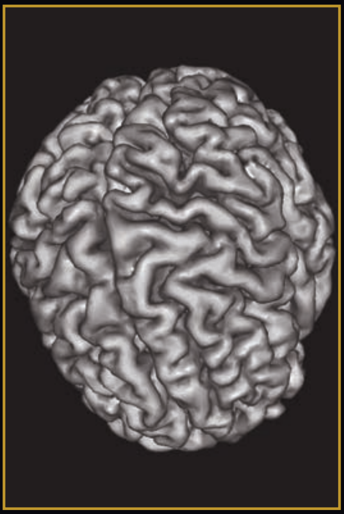
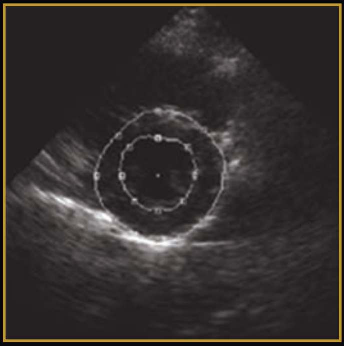
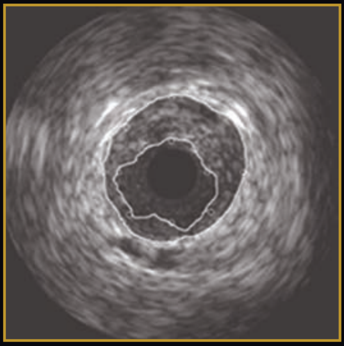
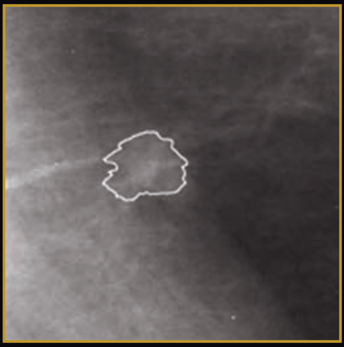
HANDBOOK OF

# Medical Imaging

## Volume 2. Medical Image Processing and Analysis

**Milan Sonka**
**J. Michael Fitzpatrick**
*Editors*

HANDBOOK OF

# Medical Imaging

## Volume 2. Medical Image Processing and Analysis

# Editorial Board



The images on the front cover are taken from figures in the text. *Top row:* left, Fig. 7.13(c), p. 430; middle, Fig. 2.14, p. 104; right, Fig. 6.10(b), p. 363 and color plate 1. *Middle row:* left, Fig. 11.27, p. 659; right, Fig. 13.28(d), p. 774, *Bottom:* Fig. 15.10(d), p. 937.

HANDBOOK OF

# Medical Imaging

## Volume 2. Medical Image Processing and Analysis

Milan Sonka
J. Michael Fitzpatrick

*Editors*

SPIE PRESS
A Publication of SPIE—The International Society for Optical Engineering
Bellingham, Washington USA

Library of Congress Cataloging-in-Publication Data

Handbook of medical imaging / [edited by] Jacob Beutel, Harold L. Kundel,
and Richard L. Van Metter.
p. cm.
Includes bibliographical references and index.
Contents: v. 1. Progress in medical physics and psychophysics.
ISBN 0-8194-3621-6 (hardcover: alk. paper)
1. Diagnostic imaging—Handbooks, manuals, etc. I. Beutel, Jacob. II. Kundel, Harold L.
III. Van Metter, Richard L.
[DNLM: 1. Diagnostic Imaging—Handbooks. 2. Health Physics—Handbooks.
3. Image Processing, Computer-Assisted—Handbooks. 4. Psychophysics—
Handbooks. 5. Technology, Radiologic—Handbooks. WN 39 H2363 2000]
RC78.7.D53 H36 2000
616.07'54—dc21 99-054487
CIP

Vol. 2 ISBN 0-8194-3622-4
*Medical Image Processing and Analysis*, edited by Milan Sonka and J. Michael Fitzpatrick.

Published by

SPIE
P.O. Box 10
Bellingham, Washington 98227-0010 USA
Phone: +1 360 676 3290
Fax: +1 360 647 1445
Email: Books@SPIE.org
Web: http://spie.org



Printed in the United States of America.

# Contents

# Extended Contents

# Preface to the Handbook of Medical Imaging

During the last few decades of the twentieth century, partly in concert with the increasing availability of relatively inexpensive computational resources, medical imaging technology, which had for nearly 80 years been almost exclusively concerned with conventional film/screen x-ray imaging, experienced the development and commercialization of a plethora of new imaging technologies. Computed tomography, MRI imaging, digital subtraction angiography, Doppler ultrasound-imaging, and various imaging techniques based on nuclear emission (PET, SPECT, etc.) have all been valuable additions to the radiologist's arsenal of imaging tools toward ever more reliable detection and diagnosis of disease. More recently, conventional x-ray imaging technology itself is being challenged by the emerging possibilities offered by flat panel x-ray detectors. In addition to the concurrent development of rapid and relatively inexpensive computational resources, this era of rapid change owes much of its success to an improved understanding of the information theoretic principles on which the development and maturation of these new technologies is based. A further important corollary of these developments in medical imaging technology has been the relatively rapid development and deployment of methods for archiving and transmitting digital images. Much of this engineering development continues to make use of the ongoing revolution in rapid communications technology offered by increasing bandwidth.

A little more than 100 years after the discovery of x rays, this three-volume *Handbook of Medical Imaging* is intended to provide a comprehensive overview of the theory and current practice of Medical Imaging as we enter the twenty-first century. Volume 1, which concerns the physics and the psychophysics of medical imaging, begins with a fundamental description of x-ray imaging physics and progresses to a review of linear systems theory and its application to an understanding of signal and noise propagation in such systems. The subsequent chapters concern the physics of the important individual imaging modalities currently in use: ultrasound, CT, MRI, the recently emerging technology of flat-panel x-ray detectors and, in particular, their application to mammography. The second half of this volume, which covers topics in psychophysics, describes the current understanding of the relationship between image quality metrics and visual perception of the diagnostic information carried by medical images. In addition, various models of perception in the presence of noise or "unwanted" signal are described. Lastly, the statistical methods used in determining the efficacy of medical imaging tasks, and

ROC analysis and its variants, are discussed.

Volume 2, which concerns Medical Image Processing and Image Analysis, provides descriptions of the methods currently being used or developed for enhancing the visual perception of digital medical images obtained by a wide variety of imaging modalities and for image analysis as a possible aid to detection and diagnosis. Image analysis may be of particular significance in future developments, since, aside from the inherent efficiencies of digital imaging, the possibility of performing analytic computation on digital information offers exciting prospects for improved detection and diagnostic accuracy.

Lastly, Volume 3 describes the concurrent engineering developments that in some instances have actually enabled further developments in digital diagnostic imaging. Among the latter, the ongoing development of bright, high-resolution monitors for viewing high-resolution digital radiographs, particularly for mammography, stands out. Other efforts in this field offer exciting, previously inconceivable possibilities, e.g., the use of 3D (virtual reality) visualization for surgical planning and for image-guided surgery. Another important area of ongoing research in this field involves image compression, which in concert with increasing bandwidth enables rapid image communication and increases storage efficiency. The latter will be particularly important with the expected increase in the acceptance of digital radiography as a replacement for conventional film/screen imaging, which is expected to generate data volumes far in excess of currently available capacity. The second half of this volume describes current developments in Picture Archiving and Communications System (PACS) technology, with particular emphasis on integration of the new and emerging imaging technologies into the hospital environment and the provision of means for rapid retrieval and transmission of imaging data. Developments in rapid transmission are of particular importance since they will enable access via telemedicine to remote or underdeveloped areas.

As evidenced by the variety of the research described in these volumes, medical imaging is still undergoing very rapid change. The editors hope that this publication will provide at least some of the information required by students, researchers, and practitioners in this exciting field to make their own contributions to its ever increasing usefulness.

**Jacob Beutel**
**J. Michael Fitzpatrick**
**Steven C. Horii**
**Yongmin Kim**
**Harold L. Kundel**
**Milan Sonka**
**Richard L. Van Metter**

# Introduction to Volume 2: Medical Image Processing and Analysis

The subject matter of this volume, which is well described by its name, *Medical Image Processing and Analysis*, has not until now been the focus of a rigorous, detailed, and comprehensive book. While there are many such books available for the more general subjects of image processing and image analysis, their broader scope does not allow for a thorough examination of the problems and approaches to their solution that are specific to medical applications. It is the purpose of this work to present the ideas and the methods of image processing and analysis that are at work in the field of medical imaging.

There is much common ground, of course. Image processing, whether it be applied to robotics, computer vision, or medicine, will treat imaging geometry, linear transforms, shift-invariance, the frequency domain, digital versus continuous domains, segmentation, histogram analysis, morphology, and other topics that apply to any imaging modality and any application. Image analysis, regardless of its application area, encompasses the incorporation of prior knowledge, the classification of features, the matching of models to subimages, the description of shape, and many of the generic problems and approaches of artificial intelligence. However, while these classic approaches to general images and to general applications are important, the special nature of medical images and medical applications requires special treatment. This volume emphasizes those approaches that are appropriate when medical images are the subjects of processing and analysis. With the emphasis placed firmly on medical applications and with the accomplishments of the more general field used as a starting point, the chapters that follow are able, individually, to treat their respective topics thoroughly, and they serve, collectively, to describe the current state of the field of medical image processing and analysis in great depth.

The special nature of *medical* images derives as much from their method of acquisition as it does from the subjects whose images are being acquired. While surface imaging is used in some applications, for example for the examination of the properties of skin in Chapter 19, medical imaging has been distinguished primarily by its ability to provide information about the volume beneath the surface, a capability that sprang first from the discovery of x radiation some one hundred years ago. Indeed, images are obtained for medical purposes almost exclusively to probe the otherwise invisible anatomy below the skin. This information may be in the form of the two-dimensional projections acquired by traditional radiography, the two-dimensional slices of B-mode ultrasound, or full three-dimensional mappings, such as those provided by computed tomography (CT), magnetic resonance (MR) imaging, single photon emission computed tomography (SPECT), positron emission tomography (PET), and 3D ultrasound. Volume 1 in this series provides a detailed look at the current state of these modalities.

In the case of radiography, perspective projection maps physical points into image space in the same way as photography, but the detection and classification of objects is confounded by the presence of overlying or underlying tissue, a problem rarely considered in general works on image analysis. In the case of tomography, three-dimensional images bring both complications and simplifications to processing and analysis relative to two-dimensional ones: The topology of three dimensions is more complex than that of two dimensions, but the problems associated with perspective projection and occlusion are gone. In addition to these geometrical differences, medical images typically suffer more from the problems of discretization, where larger pixels (voxels in three dimensions) and lower resolution combine to reduce fidelity. Additional limitations to image quality arise from the distortions and blurring associated with relatively long acquisition times in the face of inevitable anatomical motion – primarily cardiac and pulmonary, and reconstruction errors associated with noise, beam hardening, etc. These and other differences between medical and nonmedical techniques of image acquisition account for many of the differences between medical and nonmedical approaches to processing and analysis.

The fact that medical image processing and analysis deal mostly with living bodies brings other major differences in comparison to computer or robot vision. The objects of interest are soft and deformable with three-dimensional shapes whose surfaces are rarely rectangular, cylindrical, or spherical and whose features rarely include the planes or straight lines that are so frequent in technical vision applications. There are however major advantages in dealing with medical images that contribute in a substantial way to the analysis design. The available knowledge of what is and what is not normal human anatomy is one of them. Recent advances in selective enhancement of specific organs or other objects of interest via the injection of contrast-enhancing material represent others. All these differences affect the way in which images are effectively processed and analyzed.

Validation of the developed medical image processing and analysis techniques is major part of any medical imaging application. While validating the results of any methodology is always important, the scarcity of accurate and reliable independent standards creates yet another challenge for the medical imaging field.

Medical image processing deals with the development of problem-specific approaches to the enhancement of raw medical image data for the purposes of selective visualization as well as further analysis. Medical image analysis then concentrates on the development of techniques to supplement the mostly qualitative and frequently subjective assessment of medical images by human experts with a variety of new information that is quantitative, objective, and reproducible. Of course, a treatment of medical image processing and analysis without a treatment of the methods by which images are acquired, displayed, transmitted, and stored would provide only a limited view of the field. The two accompanying volumes of this handbook complete the picture with complementary information on all of these topics. Image acquisition approaches are presented in Volume 1, and image visualization, virtual reality, image transmission, compression, and archiving are dealt with in Volume 3.

The volume you hold in your hands is a result of the work of a dedicated team of researchers in medical image processing and analysis. The editors have worked very closely with the authors of individual chapters to produce a coherent volume with a uniformly deep treatment of all its topics, as well as to provide a comprehensive coverage of the field. Its pages include many cross references that further enhance the usability of the nineteen chapters, which treat separate but frequently interrelated topics. The book is loosely divided into two parts: Generally applicable theory is provided in Chapters 1–10 with the remaining chapters devoted more specifically to separate application areas. Nevertheless, many general approaches are presented in the latter group of chapters in synergy with information that is pertinent to specific applications. Each of the chapters is accompanied by numerous figures, example images, and abundant references to the literature for further reading.

The first part of this volume, which emphasizes general theory, begins with a rigorous treatment of **statistical image reconstruction** in Chapter 1. Author, *J. Fessler*, deals with the problems of tomographic reconstruction when the number of detected photons is so low that Poisson statistics must be taken into account. In this regime standard back-projection methods fail, but maximum likelihood methods, if properly applied, can still produce good images. Fessler gives a rigorous presentation of optimization methods for this problem with assessments of their practical implementation, telling us what works and what does not. The focus is on attenuation images, but the simpler problem of emission tomography is treated as well.

**Image segmentation**, which is the partitioning of an image into regions that are meaningful for a specific task, is one of the first steps leading to image analysis and interpretation. Chapter 2, which presents this subject, was written by *B. Dawant* and *A. Zijdenbos* and deals with the detection of organs such as the heart, the liver, the brain, or the lungs in images acquired by various imaging modalities.

Image **segmentation using deformable models** is the topic of Chapter 3, written by *C. Xu, D. L. Pham*, and *J. L. Prince*. Parametric and geometric deformable models are treated in a unifying way, and an explicit mathematical relationship between them is presented. The chapter also provides a comprehensive overview of many extensions to deformable models including deformable Fourier models, deformable superquadrics, point distribution models, and active appearance models.

Chapter 4, prepared by *J. Goutsias* and *S. Batman*, provides a comprehensive treatment of binary as well as gray-scale **mathematical morphology**. The theoretical concepts are illustrated on examples demonstrating their direct applicability to problems in medical image processing and analysis.

Chapter 5 is devoted to the **extraction of description features** from medical images and was written by *M. Loew*. The text summarizes the need for image features, categorizes them in several ways, presents the constraints that may determine which features to employ in a given application, defines them mathematically, and gives examples of their use in research and in clinical settings.

*A. Guéziec* authored Chapter 6, which describes methods for extracting **surface models of the anatomy** from medical images, choosing an appropriate surface representation, and optimizing surface models. This chapter provides detailed algorithms for surface representation and evaluates and compares their performance on real-life examples.

**Image interpretation** is one of the ultimate goals of medical imaging and utilizes techniques of image segmentation, feature description, and surface representation. It is heavily dependent on a priori knowledge and available approaches for pattern recognition, general interpretation, and understanding. It is also a frequent prerequisite for highly automated quantitative analysis. This topic is treated in Chapter 7 by *M. Brown* and *M. McNitt-Gray*.

Chapter 8 was authored by *J. M. Fitzpatrick*, *D. L. G. Hill*, and *C. R. Maurer, Jr.* and presents the field of **image registration**. The goal of registration, which is simply to map points in one view into corresponding points in a second view, is important when information from two images is to be combined for diagnosis or when images are used to guide surgery. This chapter presents the theoretical as well as experimental aspects of the problem and describes many approaches to its solution, emphasizing the highly successful application to rigid objects.

One of the promising directions of medical image analysis is the potential ability to perform **soft tissue characterization** from image-based information. This topic is treated in Chapter 9 by *M. Insana*, *K. Myers*, and *L. Grossman*. Tissue characterization is approached as a signal processing problem of extracting and presenting diagnostic information obtained from medical image data to improve classification performance, or to more accurately describe biophysical mechanisms. This chapter discusses in detail the difficulties resulting from the lack of accurate models of image signals and provides an insight into tissue modeling strategies.

**Validation** of the image analysis techniques is a necessity in medical imaging. In Chapter 10, *K. Bowyer* focuses on the problem of measuring the performance of medical image processing and analysis techniques. In this context, performance relates to the frequency with which an algorithm results in a correct decision. The chapter provides an overview of basic performance metrics, training and testing methodologies, and methods for statistical testing, as well as it draws attention to commonly occurring flaws of validation.

Chapter 11 opens the second part of this volume, which emphasizes applications, by providing detailed information about cardiac anatomy in the context of **echocardiographic imaging and analysis**. The chapter was written by *F. Sheehan, D. Wildon, D. Shavelle,* and *E. C. Geiser* and consists of sections devoted to echocardiographic imaging including 3-D echocardiography, echocardiographic assessment of ventricular volume, mass, and function, and their clinical consequences. Separate sections are devoted to imaging and analysis of valvular morphology and function as well as to an overview of available automated analysis approaches and their validation.

Ventricular motion and function is a topic of Chapter 12, which was contributed by *X. Papademetris* and *J. Duncan*. This chapter further explores the diagnostic utility of estimating **cardiac motion and deformation** from medical images. The authors focus primarily on the use of 3D MR image sequences, while also discussing the applications to ultrafast CT and 3D ultrasound. Description of magnetic resonance tagging, tag detection, and phase-contrast methods are all included and motion assessment is discussed in the context of the corresponding image data.

The following chapter, Chapter 13, is authored by *J. H. C. Reiber, G. Koning, J. Dijkstra, A. Wahle, B. Goedhart, F. Sheehan,* and *M. Sonka* and deals with minimally invasive approaches to **imaging the heart and coronary arteries** using contrast angiography and intravascular ultrasound. This chapter summarizes the preprocessing of angiography images, geometric correction techniques, and analysis approaches leading to quantitative coronary angiography as well as quantitative left ventriculography. Approaches for three-dimensional reconstruction from biplane angiography projections are discussed. Later sections deal with quantitative intravascular ultrasound techniques and introduce methodology for image data fusion of biplane angiography and intravascular ultrasound to achieve a geometrically correct representation of coronary lumen and wall morphology.

Chapter 14 treats ultrasound, MR, and CT approaches to non-invasive **vascular imaging** and subsequent image analysis. The chapter, written by *M. Sonka, A. Stolpen, W. Liang,* and *R. Stefancik*, covers the determination of intima–media thickness using carotid ultrasound, assessment of brachial artery endothelial function, as well as the imaging of peripheral and brain vasculature via MR angiography and x-ray CT angiography. The chapter contains methods for determining the topology and morphology of vascular structures and demonstrates how X-ray CT can be used to determine coronary calcification. Overall, facilitating early diagnosis of cardiovascular disease is one of the main goals of the methodologies presented.

**Mammography** accounts for one of the most challenging, as well as most promising, recent additions to the set of highly automated applications of medical imaging. In Chapter 15 authors, *M. L. Giger, Z. Huo, M. A. Kupinski,* and *C.*

*J. Vyborny*, provide a comprehensive treatment of computer-aided diagnosis as a second opinion for the mammographer. The entire process of image acquisition, segmentation, lesion extraction, and classification is treated in this comprehensive chapter along with a careful look at the important problem of clinical validation.

**Pulmonary imaging and analysis** is the topic of Chapter 16, prepared by *J. Reinhardt, R. Uppaluri, W. Higgins,* and *E. Hoffman*. After a brief overview of pulmonary anatomy and a survey of methods and clinical applications of pulmonary imaging, the authors discuss pulmonary image analysis leading to the segmentation of lungs and lobes, vascular and airway tree segmentation, as well as the role of virtual bronchoscopy. Approaches for the characterization of pulmonary tissue are discussed followed by sections devoted to pulmonary mechanics, image-based perfusion and ventilation, and multi-modality data fusion.

Chapter 17, authored by *P. M. Thompson, M. S. Mega, K. L. Narr, E. R. Sowell, R. E. Blanton,* and *A. W. Toga*, presents the subjects of **brain imaging, analysis, and atlas construction**. The authors describe brain atlases that fuse data across subjects, imaging modalities, and time, storing information on variations in brain structure and function in large populations. The chapter then reviews the main types of algorithms used in brain image analysis, including approaches for non-rigid image registration, anatomical modeling, tissue classification, cortical surface mapping, and shape analysis. Applications include the use of atlases to uncover disease-specific patterns of brain structure and function, and to analyze the dynamic processes of brain development and degeneration.

Chapter 18, by *M. W. Vannier*, is devoted to **tumor imaging, analysis, and cancer treatment planning**. The chapter summarizes the use of imaging in diagnosis and treatment of solid tumors. It emphasizes current imaging technologies and image processing methods used to extract information that can guide and monitor interventions after cancer has been detected leading to initial diagnosis and staging.

Chapter 19, the concluding chapter of this volume, deals with light imaging of **soft tissue movement and its finite element modeling**. It was contributed by *L. V. Tsap, D. B. Goldgof,* and *S. Sarkar*. The main topic is the analysis of soft tissue motion descriptors not easily recoverable from visual observations. The descriptors include strain and initially unknown (or hard to observe) local material properties. New methods for human tissue motion analysis from range image sequences using the nonlinear finite element method are provided, and their practical utility is demonstrated, using assessment of burn scar tissue severity and the extent of repetitive motion injury.

Medical image processing and analysis has, over the last thirty years or so, evolved from an assortment of medical applications into an established discipline. The transition has been achieved through the cooperation of a large and growing number of talented scientists, engineers, physicians, and surgeons, many of whose ideas and accomplishments are detailed by the authors of these chapters. We have produced this volume in order to make these achievements accessible to researchers both inside and outside the medical imaging field. It is our hope that its publication will encourage others to join us in the common goal of improving the diagnosis and treatment of disease and injury by means of medical imaging.

## Acknowledgments

This volume was prepared using the $\LaTeX\,2_\varepsilon$ publishing tool. All communication between the editors and authors was carried out over the Internet via E-mail, FTP, and the Web. All the authors, who contributed the individual chapters and agreed to adhere to strict deadlines and the demanding requirements of common formatting and cross referencing across chapter boundaries, share in the credit for the final product as you see it. Their enthusiasm, patience, and cooperation helped us to overcome all the obstacles that we encountered during preparation of this publication.

Several other people contributed enthusiasm and expertise during the fourteen months devoted to preparation of this volume. We acknowledge Juerg Tschirren, graduate student of the Department of Electrical and Computer Engineering at the University of Iowa, for sharing his deep knowledge of the $\LaTeX\,2_\varepsilon$ environment, solving hundreds of small problems that we all faced as authors, as well as for designing procedures for the exchange of information that served us all so well and that lead to the successful publication of this camera-ready volume.

Ken Hanson of Los Alamos National Laboratory, was present at the beginning of this project and was in fact responsible for getting it underway. We wish to thank him for editorial help and for his advice and encouragement. We acknowledge Rick Hermann, SPIE Press Manager, for his visionary support of this project, and we acknowledge the support that we received from our respective home universities – The University of Iowa at Iowa City and Vanderbilt University, Nashville, Tennessee.

Finally, we wish to thank those who gave us time to spend on this book that otherwise would have been spent with them – Jitka Sonkova, Marketa Sonkova, Pavlina Sonkova, Patricia F. Robinson, Katherine L. Fitzpatrick, John E. Fitzpatrick, and Dorothy M. Fitzpatrick. Without their patience, understanding, and support, this work would not have been possible.

Milan Sonka
*milan-sonka@uiowa.edu*

J. Michael Fitzpatrick
*j.michael.fitzpatrick@vanderbilt.edu*

# CHAPTER 1
# Statistical Image Reconstruction Methods for Transmission Tomography

Jeffrey A. Fessler
*University of Michigan*

## Contents

## 1.1 Introduction

The problem of forming cross-sectional or *tomographic* images of the attenuation characteristics of objects arises in a variety of contexts, including medical x-ray computed tomography (CT) and nondestructive evaluation of objects in industrial inspection. In the context of emission imaging, such as positron emission tomography (PET) [1, 2], single photon emission computed tomography (SPECT) [3], and related methods used in the assay of containers of radioactive waste [4], it is useful to be able to form "attenuation maps," tomographic images of attenuation coefficients, from which one can compute attenuation correction factors for use in emission image reconstruction. One can measure the attenuating characteristics of an object by transmitting a collection of photons through the object along various paths or "rays" and observing the fraction that pass unabsorbed. From measurements collected over a large set of rays, one can reconstruct tomographic images of the object. Such image reconstruction is the subject of this chapter.

In all the above applications, the number of photons one can measure in a transmission scan is limited. In medical x-ray CT, source strength, patient motion, and absorbed dose considerations limit the total x-ray exposure. Implanted objects such as pacemakers also significantly reduce transmissivity and cause severe artifacts [5]. In industrial applications, source strength limitations, combined with the very large attenuation coefficients of metallic objects, often result in a small fraction of photons passing to the detector unabsorbed. In PET and SPECT imaging, the transmission scan only determines a "nuisance" parameter of secondary interest relative to the object's emission properties, so one would like to minimize the transmission scan duration. All the above considerations lead to "low-count" transmission scans. This chapter discusses algorithms for reconstructing attenuation images from low-count transmission scans. In this context, we define low-count to mean that the mean number of photons per ray is small enough that traditional filtered-backprojection (FBP) images, or even methods based on the Gaussian approximation to the distribution of the Poisson measurements (or logarithm thereof), are inadequate. We focus the presentation in the context of PET and SPECT transmission scans, but the methods are generally applicable to all low-count transmission studies. See [6] for an excellent survey of statistical approaches for the emission reconstruction problem.

Statistical methods for reconstructing attenuation images from transmission scans have increased in importance recently for several reasons. Factors include the necessity of reconstructing 2D attenuation maps for reprojection to form 3D attenuation correction factors in septaless PET [7, 8], the widening availability of SPECT systems equipped with transmission sources [9], and the potential for reducing transmission noise in whole body PET images and in other protocols requiring short transmission scans [10]. An additional advantage of reconstructing attenuation maps in PET is that if the patient moves between the transmission and emission scan, and if one can estimate this motion, then one can calculate appropriate attenuation correction factors by reprojecting the attenuation map at the proper angles.

The traditional approach to tomographic image reconstruction is based on the nonstatistical filtered backprojection method [11, 12]. The FBP method and the data-weighted least-squares method [13, 14] for transmission image reconstruction both lead to systematic biases for low-count scans [14–16]. These biases are due to the nonlinearity of the logarithm applied to the transmission data. To eliminate these biases, one can use statistical methods based on the Poisson measurement statistics. These method use the *raw measurements* rather than the logarithms thereof [14, 17–19]. Statistical methods also produce images with lower variance than FBP [14, 16, 20]. Thus, in this chapter we focus on statistical methods.

The organization of this chapter is as follows. Section 1.2 first reviews the low-count tomographic reconstruction problem. Section 1.3 gives an overview of the principles underlying optimization algorithms for image reconstruction. Section 1.4 through Section 1.7 describe in detail four categories of reconstruction algorithms: expectation maximization, coordinate ascent, paraboloidal surrogates, and direct algorithms. All these algorithms are presented for the Poisson statistical model; Section 1.8 summarizes alternatives to that approach. Section 1.9 briefly summarizes application of the algorithms to emission reconstruction. Section 1.10 gives an overview of some advanced topics. Section 1.11 presents illustrative results for real PET transmission scans.

A few of the algorithms presented are “new” in the sense that they are derived here under more realistic assumptions than were made in some of the original papers. And we have provided simple extensions to some algorithms that were not intrinsically monotonic as previously published, but can be made monotonic by suitable modifications.

## 1.2 The problem

In transmission tomography, the quantity of interest is the spatial distribution of the linear attenuation coefficient, denoted $\mu(\vec{x}, \mathcal{E})$, where $\vec{x} = (x_1, x_2, x_3)$ denotes spatial location in 3-space, and the argument $\mathcal{E}$ parameterizes the dependence of the attenuation coefficient on incident photon energy [12]. The units of $\mu$ are typically inverse centimeters (cm$^{-1}$). If the object (patient) is moving, or if there are variations due to, e.g., flowing contrast agent, then we could also denote the temporal dependence. For simplicity we assume the object is static in this chapter. The ideal transmission imaging modality would provide a complete description of $\mu(\vec{x}, \mathcal{E})$ for a wide range of energies, at infinitesimal spatial and temporal resolutions, at a modest price, and with no harm to the subject. In practice we settle for much less.

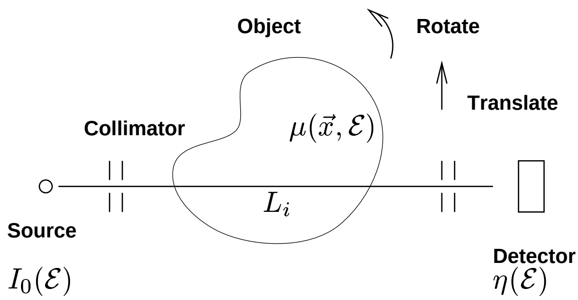



**Figure 1.1**: Transmission scanning geometry for a 1st-generation CT scanner.

### *1.2.1 Transmission measurements*

The methods described in this chapter are applicable to general transmission geometries. However, the problem is simplest to describe in the context of a first-generation CT scanner as illustrated in Fig. 1.1. A collimated source of photons with intensity $I_0(\mathcal{E})$ is transmitted through the attenuating object and the transmitted photons are recorded by a detector with detector efficiency $\eta(\mathcal{E})$. The source and detector are translated and rotated around the object. We use the letter $i$ to index the source/detector locations, where $i = 1, \ldots, N_Y$. Typically $N_Y$ is the product of the number of radial positions assumed by the source for each angular position times the number of angular positions. For 2D acquisitions, in SPECT transmission scans $N_Y \approx 10^4$; in PET transmission scans, $N_Y \approx 10^5$; and in modern x-ray CT systems $N_Y \approx 10^6$.

For simplicity, we assume that the collimation eliminates scattered photons, which is called the "narrow beam" geometry. In PET and SPECT transmission scans, the source is usually a monoenergetic radioisotope[1] that emits gamma photons with a single energy $\mathcal{E}_0$, i.e.,

$$I_0(\mathcal{E}) = I_0 \delta(\mathcal{E} - \mathcal{E}_0)\,,$$

where $\delta(\cdot)$ is the Dirac delta function. For simplicity, we assume this monoenergetic case hereafter. (In the polyenergetic case, one must consider effects such as *beam hardening* [21].)

The absorption and Compton scattering of photons by the object is governed by Beer's law. Let $b_i$ denote the mean number of photons that would be recorded by the detector (for the $i$th source-detector position, hereafter referred to as a "ray") if the object were absent. This $b_i$ depends on the scan duration, the source strength, and the detector efficiency at the source photon energy $\mathcal{E}_0$. The dependence on $i$ reflects the fact that in modern systems there are multiple detectors, each of which

[1] Some gamma emitting radioisotopes produce photons at two or more distinct energies. If the detector has adequate energy resolution, then it can separate photons at the energy of interest from other photons, or bin the various energies separately.

can have its own efficiency. By Beer's law, the mean number of photons recorded for the $i$th ray ideally would be [12]

$$b_i \exp\left(-\int_{L_i} \mu_0(\vec{x})\, dl\right), \tag{1.1}$$

where $L_i$ is the line or strip between the source and detector for the $i$th ray, and where

$$\mu_0(\vec{x}) \stackrel{\triangle}{=} \mu(\vec{x}, \mathcal{E}_0)$$

is the linear attenuation coefficient at the source photon energy. The number of photons actually recorded in practice differs from the ideal expression (1.1) in several ways. First, for a photon-counting detector[2], the number of recorded photons is a Poisson random variable [12]. Second, there will usually be additional "background" counts recorded due to Compton scatter [22], room background, random coincidences in PET [23,24], or emission crosstalk in SPECT [9,25–27]. Third, the detectors have finite width, so the infinitesimal line integral in (1.1) is an approximation. For accurate image reconstruction, one must incorporate these effects into the statistical model for the measurements, rather than simply using the idealized model (1.1).

Let $Y_i$ denote the random variable representing the number of photons counted for the $i$th ray. A reasonable statistical model[3] for these transmission measurements is that they are independent Poisson random variables with means given by

$$E[Y_i] = b_i \exp\left(-\int_{L_i} \mu_0(\vec{x})\, dl\right) + r_i\,, \tag{1.2}$$

where $r_i$ denotes the mean number of background events (such as random coincidences, scatter, and crosstalk). In many papers, the $r_i$'s are ignored or assumed to be zero. In this chapter, we assume that the $r_i$'s are known, which in practice means that they are determined separately by some other means (such as smoothing a delayed-window sinogram in PET transmission scans [31]). The noise in these estimated $r_i$'s is not considered here, and is a subject requiring further investigation and analysis. In some PET transmission scans, the random coincidences are subtracted from the measurements in real time. Statistical methods for treating this problem have been developed [32–34], and require fairly simple modifications of the algorithms presented in this chapter.

We assume the $b_i$'s are known. In PET and SPECT centers, these are determined by periodic "blank scans": transmission scans with nothing but air in the

[2]For a current integrating detector, such as those used in commercial x-ray CT scanners, the measurement noise is a mixture of Poisson photon statistics and gaussian electronic noise.

[3]Due to the effects of detector deadtime in PET and SPECT, the measurement distributions are not exactly Poisson [28–30], but the Poisson approximation seems adequate in practice.

scanner portal. Since no patient is present, these scans can have fairly long durations (typically a couple of hours, run automatically in the middle of the night). Thus the estimated $b_i$'s computed from such a long scan have much less variability than the transmission measurements ($Y_i$'s). Therefore, we ignore the variability in these estimated $b_i$'s. Accounting for the small variability in the $b_i$ estimates is another open problem (but one likely to be of limited practical significance).

### *1.2.2 Reconstruction problem*

After acquiring a transmission scan, the tomographic reconstruction problem is to estimate $\mu_0(\vec{x})$ from a realization $\{y_i = Y_i\}_{i=1}^{N_Y}$ of the measurements. This collection of measurements is usually called a *sinogram*[4] [35]. The conventional approach to this problem is to first estimate the $i$th line integral from the model (1.2) and then to apply the FBP algorithm to the collection of line-integral estimates. Specifically, let

$$l_i^{\text{true}} \triangleq \int_{L_i} \mu_0(\vec{x})\, dl$$

denote the true line integral along the $i$th ray. Conventionally, one forms an estimate $\hat{l}_i$ of $l_i^{\text{true}}$ by computing the logarithm of the measured sinogram as follows:

$$\hat{l}_i = \begin{cases} \log\left(\frac{b_i}{Y_i - r_i}\right), & Y_i > r_i \\ ?, & Y_i \leq r_i\,. \end{cases} \tag{1.3}$$

One then reconstructs an estimate $\hat{\mu}^{\text{FBP}}$ from $\{\hat{l}_i\}_{i=1}^{N_Y}$ using FBP [35]. There are several problems with this approach. First, the logarithm is not defined when $Y_i \leq r_i$, which can happen frequently in low-count transmission scans. (Typically one must substitute some artificial value (denoted "?" above) for such rays, or interpolate neighboring rays [36], which can lead to biases.) Second, the above procedure yields *biased* estimates of the line-integral. By Jensen's inequality, since $-\log x$ is a concave function, for any random variable $X$,

$$E[-\log X] \geq -\log E[X]$$

(see [37], p. 50), so when applied to (1.2) and (1.3)

$$\begin{aligned} E[\hat{l}_i] &= E\left[\log\left(\frac{b_i}{Y_i - r_i}\right)\right] = E\left[-\log\left(\frac{Y_i - r_i}{b_i}\right)\right] \\ &\geq -\log\left(E\left[\frac{Y_i - r_i}{b_i}\right]\right) = \int_{L_i} \mu_0(\vec{x})\, dl = l_i^{\text{true}}\,. \end{aligned} \tag{1.4}$$

Thus, the logarithm in (1.3) *systematically over-estimates* the line integral on average. This over-estimation has been verified empirically [14, 15]. One can show

[4] When the ray measurements are organized as a 2D array according their radial and angular coordinates, the projection of a point object appears approximately as a sinusoidal trace in the array.

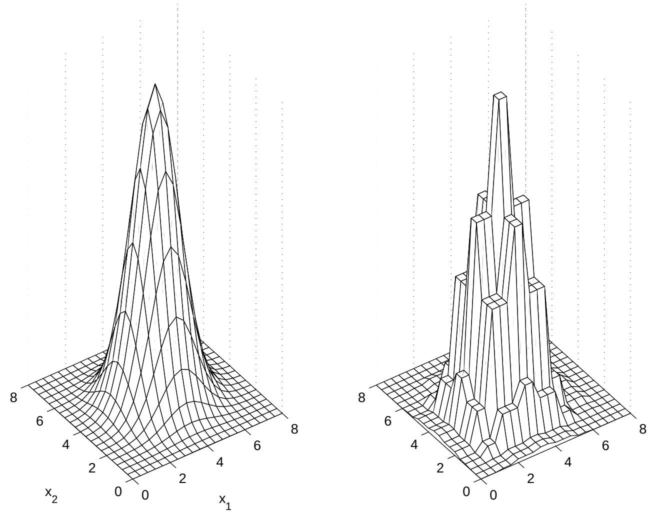



**Figure 1.2:** Illustration of 2D function $\mu_0(x_1, x_2)$ parameterized using the pixel basis (1.6).

analytically that the bias increases as the counts decrease [14], so the logarithm is particularly unsuitable for low-count scans. A third problem with (1.3) is that the variances of the $\hat{l}_i$'s can be quite nonuniform, so some rays are much more informative than other rays. The FBP method treats all rays equally, even those for which $Y_i - r_i$ is non-positive, which leads to noisy images corrupted by streaks originating from high variance $\hat{l}_i$'s. Noise is considered only as an afterthought by apodizing the ramp filter, which is equivalent to space-invariant smoothing. (There are a few exceptions where space-variant sinogram filtering has been applied, e.g., [38–40].) Fourth, the FBP method is poorly suited to nonstandard imaging geometries, such as truncated fan-beam or cone-beam scans, e.g., [41–46].

Since noise is a primary concern, the image reconstruction problem is naturally treated as a statistical estimation problem. Since we only have a finite number $N_Y$ of measurements, it is natural to also represent $\mu_0(\vec{x})$ with a finite parameterization. Such parameterizations are reasonable in practice since ultimately the estimate of $\mu_0$ will be viewed on a digital display with a finite number of pixels. After one has parameterized $\mu_0$, the reconstruction problem becomes a statistical problem: estimate the parameters from the noisy measurements $\{Y_i\}_{i=1}^{N_Y}$.

A general approach to parameterizing the attenuation map is to expand it in terms of a finite basis expansion [47, 48]:

$$\mu_0(\vec{x}) = \sum_{j=1}^{N_p} \mu_j \chi_j(\vec{x}) , \tag{1.5}$$

where $N_p$ is the number of coefficients $\mu_j$ and basis functions $\chi_j(\vec{x})$. There are many possible choices for the basis functions. We would like to choose basis functions that naturally represent nonnegative functions since $\mu_0 \geq 0$. We would also like basis functions that have compact support, since such a basis yields a very sparse system matrix $\boldsymbol{A}$ in (1.9) below. The conventional basis is just the "pixel" or "voxel" basis, which satisfies both of these requirements. The voxel basis $\chi_j(\vec{x})$ is 1 inside the $j$th voxel, and is 0 everywhere else. In two-space, one can express the pixel basis by

$$\chi_j(x, y) = \operatorname{rect}\left(\frac{x - x_j}{\Delta}\right) \operatorname{rect}\left(\frac{y - y_j}{\Delta}\right), \tag{1.6}$$

where $(x_j, y_j)$ is the center of the $j$th pixel and $\Delta$ is the pixel width. This basis gives a piecewise-constant approximation to $\mu_0$, as illustrated in Fig. 1.2. With any parameterization of the form (1.5), the problem of estimating $\mu_0(\vec{x})$ is reduced to the simpler problem of estimating the parameter vector $\underline{\mu} = [\mu_1, \ldots, \mu_{N_p}]'$ from the measurement vector $\underline{Y} = [Y_1, \ldots, Y_{N_Y}]'$, where "$'$" denotes vector and matrix transpose. Under the parameterization (1.5), the line integral in (1.2) becomes the following summation:

$$\int_{L_i} \mu_0(\vec{x})\, dl = \int_{L_i} \sum_{j=1}^{N_p} \mu_j \chi_j(\vec{x})\, dl = \sum_{j=1}^{N_p} \mu_j \int_{L_i} \chi_j(\vec{x})\, dl = \sum_{j=1}^{N_p} a_{ij} \mu_j,$$

where

$$a_{ij} \triangleq \int_{L_i} \chi_j(\vec{x})\, dl$$

is the line integral[5] along the $i$th ray through the $j$th basis function. This simplification yields the following discrete-discrete measurement model:

$$Y_i \sim \operatorname{Poisson}\left\{\bar{y}_i(\underline{\mu}_{\text{true}})\right\}, \; i = 1, \ldots, N_Y, \tag{1.7}$$

where the ensemble mean of the $i$th measurement is denoted

$$\bar{y}_i(\underline{\mu}) \triangleq b_i e^{-[\boldsymbol{A}\underline{\mu}]_i} + r_i \tag{1.8}$$

$$[\boldsymbol{A}\underline{\mu}]_i \triangleq \sum_{j=1}^{N_p} a_{ij} \mu_j, \tag{1.9}$$

where $\boldsymbol{A} = \{a_{ij}\}$. The remainder of this chapter will be based on the Poisson measurement model (1.7). (See Appendix 1.14 for a review of Poisson statistics.)

[5] In practice, we use normalized strip integrals [49, 50] rather than line integrals to account for finite detector width [51]. Regardless, the units of $a_{ij}$ are length units (mm or cm), whereas the units of the $\mu_j$'s are inverse length.

### 1.2.3 Likelihood-based estimation

*Maximum-likelihood (ML) estimation* is a natural approach[6] for finding $\underline{\mu}$ from a particular measurement realization $\underline{Y} = \underline{y}$ when a statistical model such as (1.7) is available. The ML estimate $\hat{\underline{\mu}}$ is defined as follows:

$$\hat{\underline{\mu}} = \arg \max_{\underline{\mu} \geq \underline{0}} L(\underline{\mu}) \;, \quad L(\underline{\mu}) \stackrel{\triangle}{=} \log P[\underline{Y} = \underline{y}; \underline{\mu}] \,.$$

For the Poisson model (1.7), the measurement joint probability mass function is

$$P[\underline{Y} = \underline{y}; \underline{\mu}] = \prod_{i=1}^{N_Y} P[Y_i = y_i; \underline{\mu}] = \prod_{i=1}^{N_Y} e^{-\bar{y}_i(\underline{\mu})} [\bar{y}_i(\underline{\mu})]^{y_i} / y_i! \,. \tag{1.10}$$

The ML method seeks the object (as described by the parameter vector $\underline{\mu}$) that maximizes the probability of having observed the particular measurements that were recorded. The first paper to propose a ML approach for transmission tomography appears to be due to Rockmore and Macovski in 1977 [47]. However, the pseudoinverse method described in [47] in general does not find the maximizer of the likelihood $L(\underline{\mu})$.

For independent transmission measurements, we can use (1.8) and (1.10) to express the log-likelihood in the following convenient form:

$$L(\underline{\mu}) \equiv \sum_{i=1}^{N_Y} h_i([\boldsymbol{A}\underline{\mu}]_i) \tag{1.11}$$

where we use "$\equiv$" hereafter for expressions that are equal up to irrelevant constants independent of $\underline{\mu}$, and where the marginal log-likelihood of the $i$th measurement is

$$h_i(l) \stackrel{\triangle}{=} y_i \log(b_i e^{-l} + r_i) - (b_i e^{-l} + r_i) \,. \tag{1.12}$$

A typical $h_i$ is shown in Fig. 1.4 on page 17. For convenience later, we also list the derivatives of $h_i$ here:

$$\dot{h}_i(l) \stackrel{\triangle}{=} \frac{d}{dl} h_i(l) = \left[1 - \frac{y_i}{b_i e^{-l} + r_i}\right] b_i e^{-l} \tag{1.13}$$

$$\ddot{h}_i(l) \stackrel{\triangle}{=} \frac{d^2}{dl^2} h_i(l) = -\left[1 - \frac{y_i r_i}{(b_i e^{-l} + r_i)^2}\right] b_i e^{-l} \,. \tag{1.14}$$

[6] The usual rationale for the ML approach is that ML estimators are asymptotically unbiased and asymptotically efficient (minimum variance) under very general conditions [37]. Such asymptotic properties alone would be a questionable justification for the ML approach in the case of low-count transmission scans. However, ML estimators often perform well even in the "non-asymptotic" regime. We are unaware of any data-fit measure for low-count transmission scans that outperforms the log-likelihood, but there is no known proof of optimality of the log-likelihood in this case, so the question is an open one.

The algorithms described in the following sections are based on various strategies for finding the maximizer of $L(\underline{\mu})$. Several of the algorithms are quite general in the sense that one can easily modify them to apply to many objective functions of the form (1.11), even when $h_i$ has a functional form different from the form (1.12) that is specific to transmission measurements. Thus, even though the focus of this chapter is transmission imaging, many of the algorithms and comments apply equally to emission reconstruction and to other inverse problems.

Maximizing the log-likelihood $L(\cdot)$ alone leads to unacceptably noisy images, because tomographic image reconstruction is an ill-conditioned problem. Roughly speaking, this means that there are many choices of attenuation maps $\mu_0(\vec{x})$ that fit the measurements $\{Y_i\}_{i=1}^{N_Y}$ reasonably well. Even when the problem is parameterized, there are many choices of the vector $\underline{\mu}$ that fit the measurements $\{Y_i\}_{i=1}^{N_Y}$ reasonably well, where the fit is quantified by the log-likelihood $L(\underline{\mu})$. Not all of those images are useful or physically plausible. Thus, the likelihood alone does not adequately identify the "best" image. One effective remedy to this problem is to modify the objective function by including a *penalty function* that favors reconstructed images that are piecewise smooth. This process is called *regularization* since the penalty function improves the conditioning of the problem[7]. In this chapter we focus on methods that form an estimate $\underline{\hat{\mu}}$ of the true attenuation map $\underline{\mu}_{\mathrm{true}}$ by maximizing a *penalized-likelihood* objective function of the following form:

$$\underline{\hat{\mu}} \stackrel{\triangle}{=} \arg\max_{\underline{\mu} \geq \underline{0}} \Phi(\underline{\mu}), \quad \Phi(\underline{\mu}) \stackrel{\triangle}{=} L(\underline{\mu}) - \beta R(\underline{\mu}), \tag{1.15}$$

where the objective function $\Phi$ includes a roughness penalty $R(\underline{\mu})$ discussed in more detail below. The parameter $\beta$ controls the tradeoff between spatial resolution and noise: larger values of $\beta$ generally lead to reduced noise at the price of reduced spatial resolution. Solving (1.15) is the primary subject of this chapter.

One benefit of using methods that are based on objective functions such as (1.15) is that for such methods, image quality is determined by the objective function rather than by the particular iterative algorithm, provided the iterative algorithm converges to the maximizer of the objective function. In particular, from the objective function one can analyze spatial resolution properties and bias, variance, and autocorrelation properties [16, 20, 54–59].

#### *1.2.3.1 Connection to Bayesian perspective*

By letting $\beta = 0$ in (1.15) and in the algorithms presented in the following sections, one has ML algorithms as a special case. Bayesian image reconstruc-

---

[7] For emission tomography, a popular alternative approach to "regularization" is simply to post-smooth the ML reconstruction image with a Gaussian filter. In the emission case, under the somewhat idealized assumption of a shift-invariant Gaussian blur model for the system, a certain commutability condition ((12) of [52] holds, which ensures that Gaussian post-filtering is equivalent to Gaussian sieves. It is unclear whether this equivalence holds in the transmission case, although some authors have implied that it does without proof, e.g. [53].

tion formulations also lead to objective functions of the form (1.15). Suppose one considers $\underline{\mu}$ to be a random vector drawn from a prior distribution $f(\underline{\mu})$ that is proportional to $e^{-\beta R(\underline{\mu})}$. (Such priors arise naturally in the context of Markov random field models for images [60].) One computes the maximum *a posteriori* (MAP) estimate of $\underline{\mu}$ by maximizing the posterior distribution $f(\underline{\mu}|\underline{Y})$. By Bayes rule:

$$f(\underline{\mu}|\underline{Y}) = f(\underline{Y}|\underline{\mu}) f(\underline{\mu}) / f(\underline{Y})$$

so the log posterior is

$$\log f(\underline{\mu}|\underline{Y}) \equiv \log f(\underline{Y}|\underline{\mu}) + \log f(\underline{\mu}) \equiv L(\underline{\mu}) - \beta R(\underline{\mu}) .$$

Thus MAP estimation is computationally equivalent to (1.15).

### *1.2.4 Penalty function*

It has been considered by many authors to be reasonable to assume that the attenuation maps of interest are piecewise smooth functions. An extreme example of such assumptions is the common use of attenuation map segmentation in PET imaging to reduce the noise due to attenuation correction factors [61–63]. Under the piecewise smooth attenuation map assumption, it is reasonable for the penalty function $R(\underline{\mu})$ to discourage images that are too "rough." The simplest penalty function that discourages roughness considers the discrepancies between neighboring pixel values:

$$R(\underline{\mu}) = \sum_{j=1}^{N_p} \frac{1}{2} \sum_{k=1}^{N_p} w_{jk} \psi(\mu_j - \mu_k) , \tag{1.16}$$

where $w_{jk} = w_{kj}$. Ordinarily $w_{jk} = 1$ for the four horizontal and vertical neighboring pixels, $w_{jk} = 1/\sqrt{2}$ for diagonal neighboring pixels, and $w_{jk} = 0$ otherwise. One can also adopt the modifications described in [20, 54–57] to provide more uniform spatial resolution. The *potential function* $\psi$ assigns a cost to $\mu_j - \mu_k$.

For the results presented in Section 1.11, we used a penalty function of the form (1.16). However, the methods we present all apply to much more general penalty functions. Such generality is needed for penalty functions such as the weak-plate prior of [64] or the local averaging function considered in [65]. One can express most[8] of the penalty functions that have been used in tomographic reconstruction in the following very general form:

$$R(\underline{\mu}) = \sum_{k=1}^{K} \psi_k([\boldsymbol{C}\underline{\mu}]_k) \tag{1.17}$$

[8] One exception is the median root prior [66].

where $\boldsymbol{C}$ is a $K \times N_p$ penalty matrix and

$$[\boldsymbol{C}\underline{\mu}]_k = \sum_{j=1}^{N_p} c_{kj}\mu_j \,.$$

We assume throughout that the functions $\psi_k$ are symmetric and differentiable.

The pairwise penalty (1.16) is the special case of (1.17) where $K \approx 2N_p$ and each row of $\boldsymbol{C}$ has one $+1$ and one $-1$ entry corresponding to some pair of pixels.

In this chapter we focus on quadratic penalty functions where $\psi_k(t) = \omega_k t^2/2$ for $\omega_k \geq 0$, so

$$R(\underline{\mu}) = \sum_{k=1}^{K} \omega_k \frac{1}{2}([\boldsymbol{C}\underline{\mu}]_k)^2 = \frac{1}{2}\underline{\mu}'\boldsymbol{C}'\boldsymbol{\Omega}\boldsymbol{C}\underline{\mu}\,, \tag{1.18}$$

where $\boldsymbol{\Omega} \stackrel{\triangle}{=} \operatorname{diag}\{\omega_k\}$. The second derivative of such a penalty is given by

$$\frac{\partial^2}{\partial\mu_j^2} R(\underline{\mu}) = \sum_{k=1}^{K} c_{kj}^2 \omega_k \,. \tag{1.19}$$

This focus is for simplifying the presentation; in practice nonquadratic penalty functions are often preferable for transmission image reconstruction e.g. [67, 68].

### 1.2.5 Concavity

From the second derivative expression (1.14), when $r_i = 0$, $\ddot{h}_i(l) = -b_i e^{-l}$, which is always nonpositive, so $h_i$ is concave over all of $\mathbb{R}$ (and strictly concave if $b_i > 0$). From (1.11) one can easily verify that the Hessian matrix (the $N_p \times N_p$ matrix of second partial derivatives) of $L(\cdot)$ is:

$$\nabla^2 L(\underline{\mu}) = \boldsymbol{A}' \operatorname{diag}\Big\{\ddot{h}_i([\boldsymbol{A}\underline{\mu}]_i)\Big\}\, \boldsymbol{A}\,, \tag{1.20}$$

where $\operatorname{diag}\{d_i\}$ is a $N_Y \times N_Y$ diagonal matrix with $i$th diagonal element $d_i$. Thus the log-likelihood is concave over all of $\mathbb{R}^{N_p}$ when $r_i = 0\ \forall i$. If the $\psi_k$'s are all strictly convex and $L(\cdot)$ is concave, then the objective $\Phi$ is strictly concave under mild conditions on $\boldsymbol{A}$ [69]. Such concavity is central to the convergence proofs of the algorithms described below. In the case $r_i \neq 0$, the likelihood $L(\cdot)$ is not necessarily concave. Nevertheless, in our experience it seems to be unimodal. (Initializing monotonic iterative algorithms with different starting images seems to lead to the same final image.) In the non-concave case, we cannot guarantee global convergence to the global maximum for any of the algorithms described below. For the monotonic algorithms we usually can prove convergence to a local maximum [70]; if in addition the objective function is unimodal, then the only local maximum will be the global maximum, but proving that $\Phi$ is unimodal is an open problem.

## 1.3 Optimization algorithms

Ignoring the nonnegativity constraint, one could attempt to find $\underline{\hat{\mu}}$ analytically by zeroing the gradient of $\Phi$. The partial derivatives of $\Phi$ are

$$\frac{\partial}{\partial \mu_j}\Phi(\underline{\mu}) = \sum_{i=1}^{N_Y} a_{ij}\dot{h}_i([\boldsymbol{A}\underline{\mu}]_i) - \beta\frac{\partial}{\partial \mu_j}R(\underline{\mu}) \ , \ \ j = 1, \ldots, N_p \, , \tag{1.21}$$

where $\dot{h}_i$ was defined in (1.13). Unfortunately, even disregarding both the nonnegativity constraint and the penalty function, there are no closed-form solutions to the set of equations (1.21), except in the trivial case when $\boldsymbol{A} = \boldsymbol{I}$. Even when $\boldsymbol{A} = \boldsymbol{I}$ there are no closed-form solutions for nonseparable penalty functions. Thus iterative methods are required to find the maximizer $\underline{\hat{\mu}}$ of such objective functions.

### *1.3.1 Why so many algorithms?*

Analytical solutions for the maximizer of (1.15) appear intractable, so one must use iterative algorithms. An *iterative algorithm* is a procedure that is initialized with an initial guess $\underline{\mu}^{(0)}$ of $\underline{\mu}$, and then recursively generates a sequence $\underline{\mu}^{(1)}, \underline{\mu}^{(2)}, \ldots$, also denoted $\{\underline{\mu}^{(n)}\}$. Ideally, the iterates $\{\underline{\mu}^{(n)}\}$ should rapidly approach the maximizer $\underline{\hat{\mu}}$. When developing algorithms for image reconstruction based on penalized-likelihood objective functions, there are many design considerations, most of which are common to any problem involving iterative methods. In particular, an algorithm designer should consider the impact of design choices on the following characteristics.

- Monotonicity ($\Phi(\underline{\mu}^{(n)})$ increases every iteration)
- Nonnegativity constraint ($\underline{\mu} \geq \underline{0}$)
- Parallelization
- Sensitivity to numerical errors
- Convergence rate (as few iterations as possible)
- Computation time per iteration (as few floating point operations as possible)
- Storage requirements (as little memory as possible)
- Memory bandwidth (data access)

Generic numerical methods such as steepest ascent do not exploit the specific structure of the objective function $\Phi$, nor do they easily accommodate the nonnegativity constraint. Thus for fastest convergence, one must seek algorithms tailored to this type of problem. Some of the relevant properties of $L$ include:

- $L(\underline{\theta})$ is a sum of scalar functions $h_i(\cdot)$.
- The $h_i$'s have bounded curvature, and are concave when $r_i = 0$.
- The arguments of the functions $h_i$ are inner products.
- The inner product coefficients are all nonnegative.

The cornucopia of algorithms that have been proposed in the image reconstruction literature exploit these properties (implicitly or explicitly) in different ways.

### *1.3.2 Optimization transfer principle*

Before delving into the details of the many algorithms that have been proposed for maximizing $\Phi$, we first describe a very useful and intuitive general principle that underlies almost all the methods. The principle is called *optimization transfer*. This idea was described briefly as a *majorization principle* in the limited context of 1D line searches in the classic text by Ortega and Rheinbolt [71, p. 253]. It was rediscovered and generalized to inverse problems in the recent work of De Pierro [72, 73] and Lange [69, 74]. Since the concept applies more generally than just to transmission tomography, we use $\underline{\theta}$ as the generic unknown parameter here.

The basic idea is illustrated in Fig. 1.3. Since $\Phi$ is difficult to maximize, at the $n$th iteration we can replace $\Phi$ with a *surrogate function* $\phi(\underline{\theta}; \underline{\theta}^{(n)})$ that is easier to maximize, i.e., the next iterate is defined as:

$$\underline{\theta}^{(n+1)} \triangleq \arg\max_{\underline{\theta}} \phi(\underline{\theta}; \underline{\theta}^{(n)}) \,. \tag{1.22}$$

The maximization is restricted to the valid parameter space (e.g. $\underline{\theta} \geq \underline{0}$ for problems with nonnegative constraints). Maximizing $\phi(\cdot; \underline{\theta}^{(n)})$ will usually not lead directly to the global maximizer $\underline{\hat{\mu}}$. Thus one repeats the process iteratively, finding a new surrogate function $\phi$ at each iteration and then maximizing that surrogate function. If we choose the surrogate functions appropriately, then the sequence $\{\underline{\theta}^{(n)}\}$ should eventually converge to the maximizer $\underline{\hat{\mu}}$ [75].

Fig. 1.3 does not do full justice to the problem, since 1D functions are usually fairly easy to maximize. The optimization transfer principle is particularly compelling for problems where the dimension of $\underline{\theta}$ is large, such as in inverse problems like tomography.

It is very desirable to use algorithms that monotonically increase $\Phi$ each iteration, i.e., for which $\Phi(\underline{\theta}^{(n+1)}) \geq \Phi(\underline{\theta}^{(n)})$. Such algorithms are guaranteed to be stable, i.e., the sequence $\{\underline{\theta}^{(n)}\}$ will not diverge if $\Phi$ is concave. And generally such algorithms will converge to the maximizer $\underline{\hat{\mu}}$ if it is unique [70]. If we choose surrogate functions that satisfy

$$\Phi(\underline{\theta}) - \Phi(\underline{\theta}^{(n)}) \geq \phi(\underline{\theta}; \underline{\theta}^{(n)}) - \phi(\underline{\theta}^{(n)}; \underline{\theta}^{(n)}), \ \forall \underline{\theta}, \underline{\theta}^{(n)} \,, \tag{1.23}$$

then one can see immediately that the algorithm (1.22) monotonically increases $\Phi$. To ensure monotonicity, it is not essential to find the exact maximizer in (1.22). It suffices to find a value $\underline{\theta}^{(n+1)}$ such that $\phi(\underline{\theta}^{(n+1)}; \underline{\theta}^{(n)}) \geq \phi(\underline{\theta}^{(n)}; \underline{\theta}^{(n)})$, since that alone will ensure $\Phi(\underline{\theta}^{(n+1)}) \geq \Phi(\underline{\theta}^{(n)})$ by (1.23). The various algorithms described in the sections that follow are all based on different choices of the surrogate function $\phi$, and on different procedures for the maximization in (1.22).

Rather than working with (1.23), all the surrogate functions we present satisfy

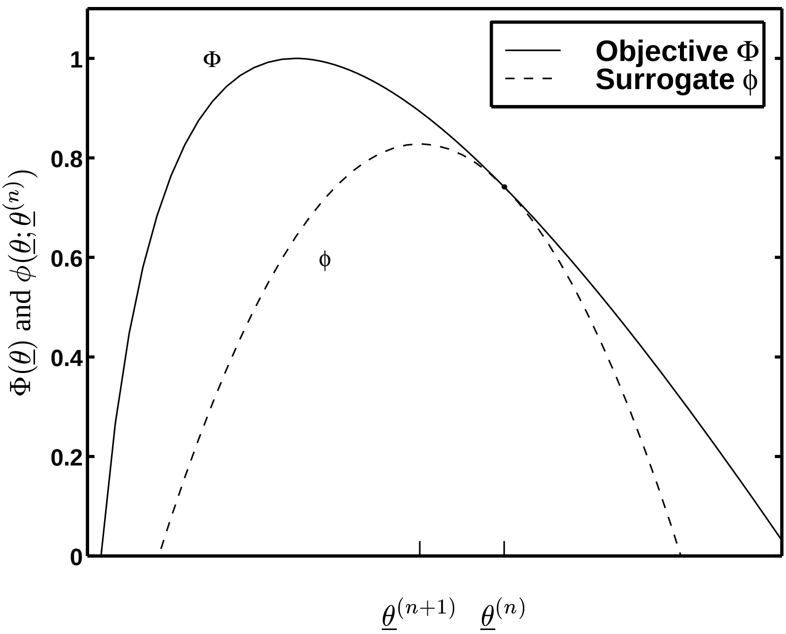



**Figure 1.3:** Illustration of optimization transfer in 1D.

the following conditions:

$$\begin{aligned} \phi(\underline{\theta}^{(n)}; \underline{\theta}^{(n)}) &= \Phi(\underline{\theta}^{(n)}) && (1.24) \\ \nabla_{\underline{\theta}} \phi(\underline{\theta}; \underline{\theta}^{(n)}) \Big|_{\underline{\theta}=\underline{\theta}^{(n)}} &= \nabla \Phi(\underline{\theta}) |_{\underline{\theta}=\underline{\theta}^{(n)}} && (1.25) \\ \phi(\underline{\theta}; \underline{\theta}^{(n)}) &\leq \Phi(\underline{\theta}) \ \forall \underline{\theta} \geq \underline{0} \, . && (1.26) \end{aligned}$$

Any surrogate function that satisfies these conditions will satisfy (1.23). (The middle condition follows from the outer two conditions when $\Phi$ and $\phi$ are differentiable.)

### *1.3.3 Convergence rate*

The convergence rate of an iterative algorithm based on the optimization transfer principle can be analyzed qualitatively by considering Fig. 1.3. If the surrogate function $\phi$ has low curvature, then it appears as a "broad" graph in Fig. 1.3, which means that the algorithm can take large steps ($\underline{\theta}^{(n+1)} - \underline{\theta}^{(n)}$ can be large) which means that it reaches the maximizer faster. Conversely, if the surrogate function has high curvature, then it appears as a "skinny" graph, the steps are small, and many steps are required for convergence. So in general we would like to find low curvature surrogate functions, with the caveat that we want to maintain $\phi \leq \Phi$ to ensure monotonicity [76]. And of course we would also like the surrogate $\phi$ to be

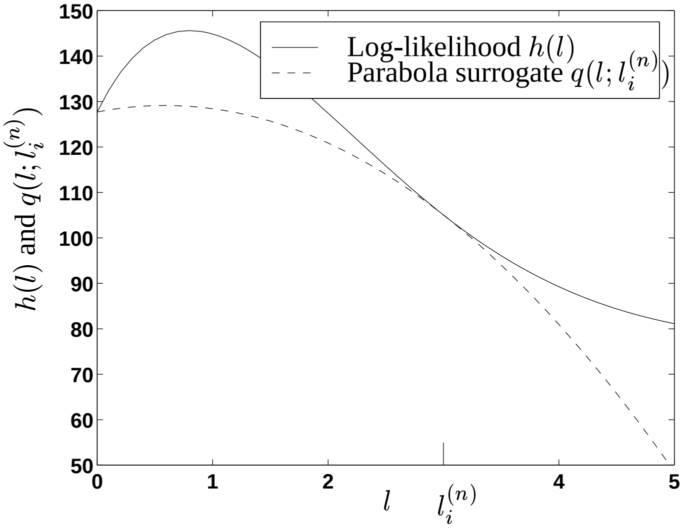



**Figure 1.4**: Ray log-likelihood $h_i(l)$ and parabola surrogate $q_i(l; l_i^{(n)})$ for $y_i = 50$, $b_i = 100$, $r_i = 5$, and $l_i^{(n)} = 3$.

easy to maximize for (1.22). Unfortunately, the criteria "low curvature" and "easy to maximize" are often incompatible, so we must compromise.

### *1.3.4 Parabola surrogate*

Throughout this chapter we struggle with the ray log-likelihood function $h_i$ defined in (1.12), which we rewrite here without the $i$'s for simplicity:

$$h(l) = y \log(b\, e^{-l} + r) - (b\, e^{-l} + r)\,,$$

where $y, b, r \geq 0$ are known constants. Of the many possible surrogate functions that could replace $h(\cdot)$ in an optimization transfer approach, a choice that is particularly convenient is a parabola surrogate:

$$q(l; l_i^{(n)}) \triangleq h(l_i^{(n)}) + \dot{h}(l_i^{(n)})(l - l_i^{(n)}) - \frac{c}{2}(l - l_i^{(n)})^2\,, \tag{1.27}$$

for some choice of the curvature $c \geq 0$, where

$$l_i^{(n)} = [\boldsymbol{A}\underline{\mu}^{(n)}]_i \tag{1.28}$$

is the $i$th line integral through the estimated attenuation map at the $n$th iteration. The choice (1.27) clearly satisfies conditions (1.24) and (1.25), but we must carefully choose $c = c(l_i^{(n)}, y, b, r)$ to ensure that $q(l; l_i^{(n)}) \leq h(l)\ \forall l \geq 0$ so that (1.26) is satisfied. On the other hand, from the convergence rate description in the preceding section, we would like the curvature $c$ to be as small as possible. In other words, we would like to find

$$c(l_i^{(n)}, y, b, r) \triangleq$$

$$\min \left\{ c \,:\, h(l_i^{(n)}) + \dot{h}(l_i^{(n)})(l - l_i^{(n)}) - \frac{c}{2}(l - l_i^{(n)})^2 \leq h(l),\ \forall l \geq 0 \right\} .$$

In [77], we showed that the optimal curvature is as follows:

$$c(l_i^{(n)}, y, b, r) = \begin{cases} \left[ -2 \dfrac{h(0) - h(l_i^{(n)}) + \dot{h}(l_i^{(n)}) l_i^{(n)}}{(l_i^{(n)})^2} \right]_+ , & l_i^{(n)} > 0 \\ \left[ -\ddot{h}(l_i^{(n)}) \right]_+ , & l_i^{(n)} = 0 , \end{cases} \tag{1.29}$$

where $[x]_+$ is $x$ for positive $x$ and zero otherwise. Fig. 1.4 illustrates the surrogate parabola $q$ in (1.27) with the optimal curvature (1.29).

One small inconvenience with (1.29) is that it changes every iteration since it depends on $l_i^{(n)}$. An alternative choice of the curvature that ensures $q \leq h$ is the maximum second derivative of $-h(l)$ over $[0, \infty)$. In [77] we show that

$$\max_{l \geq 0} \left[ -\ddot{h}(l) \right]_+ = \left[ -\ddot{h}(0) \right]_+ = \left[ \left( 1 - \frac{y}{b + r} \right) b \right]_+ . \tag{1.30}$$

We can precompute this curvature before iterating since it is independent of $l_i^{(n)}$. However, typically this curvature is much larger than the optimal choice (1.29), so the floating point operations (flops) saved by precomputing may be lost in increased number of iterations due to a slower convergence rate.

The surrogate parabola (1.27) with curvature (1.29) will be used repeatedly in this chapter, both for the derivation of recently developed algorithms, as well as for making minor improvements to older algorithms that were not monotonic as originally proposed. A similar approach applies to the emission reconstruction problem [78].

## 1.4 EM algorithms

The emission reconstruction algorithm derived by Shepp and Vardi in [79] and by Lange and Carson in [17] is often referred to as "the" EM algorithm in the nuclear imaging community. In fact, the expectation-maximization (EM) framework is a general method for developing many different algorithms [80]. The appeal of the EM framework is that it leads to iterative algorithms that in principle yield sequences of iterates that monotonically increase the objective function. Furthermore, in many statistical problems one can derive EM algorithms that are quite simple to implement. Unfortunately, the Poisson transmission reconstruction problem does not seem to be such a problem. Only one basic type of EM algorithm has been proposed for the transmission problem, and that algorithm converges *very* slowly and has other difficulties described below. We include the description of the EM algorithm for completeness, but the reader who is not interested in the historical perspective could safely skip this section since we present much more efficient algorithms in subsequent sections.

We describe the general EM framework in the context of problems where one observes a realization $\underline{y}$ of a measurement vector $\underline{Y}$, and wishes to estimate a parameter vector $\underline{\theta}$ by maximizing the likelihood or penalized log-likelihood. To develop an EM algorithm, one must first postulate a hypothetical collection of random variables called the "complete data space" $\underline{X}$. These are random variables that, in general, were not observed during the experiment, but that might have simplified the estimation procedure had they been observed. The only requirement that the complete data space must satisfy is that one must be able to extract the observed data from $\underline{X}$, i.e. there must exist a function $h(\cdot)$ such that

$$\underline{Y} = h(\underline{X}) \,. \tag{1.31}$$

This is a trivial requirement since one can always include the random vector $\underline{Y}$ itself in the collection $\underline{X}$ of random variables.

Having judiciously chosen $\underline{X}$, an essential ingredient of any EM algorithm is the following conditional expectation of the log-likelihood of $\underline{X}$:

$$Q(\underline{\theta};\underline{\theta}^{(n)}) = E[\log f(\underline{X};\underline{\theta}) \mid \underline{Y} = \underline{y}; \underline{\theta}^{(n)}] = \int f(\underline{x} \mid \underline{Y} = \underline{y}; \underline{\theta}^{(n)}) \log f(\underline{x};\underline{\theta})\, d\underline{x}, \tag{1.32}$$

and, in the context of penalized-likelihood problems, the following function

$$\phi(\underline{\theta};\underline{\theta}^{(n)}) \stackrel{\triangle}{=} Q(\underline{\theta};\underline{\theta}^{(n)}) - \beta R(\underline{\theta}) \,, \tag{1.33}$$

where $R(\underline{\theta})$ is a penalty function. We refer to $\phi$ as an *EM-based surrogate function*, since one replaces the difficult problem of maximizing $\Phi$ with a sequence of (hopefully) simpler maximizations of $\phi$. There are often alternative surrogate functions that have advantages over the EM-based functions, as described in Section 1.6.1. The surrogate function concept is illustrated in Fig. 1.3 on page 16.

An EM algorithm is initialized at an arbitrary point $\underline{\theta}^0$ and generates a sequence of iterates $\underline{\theta}^1, \underline{\theta}^2, \ldots$. Under fairly general conditions [81], the sequence $\{\underline{\theta}^{(n)}\}$ converges to the maximizer of the objective function $\Phi(\underline{\theta}) = L(\underline{\theta}) - \beta R(\underline{\theta})$. The EM recursion is as follows:

E-step: find $Q(\underline{\theta};\underline{\theta}^{(n)})$ using (1.32) and $\phi(\underline{\theta};\underline{\theta}^{(n)})$ using (1.33)
M-step: $\underline{\theta}^{(n+1)} = \arg\max_{\underline{\theta}} \phi(\underline{\theta};\underline{\theta}^{(n)})\,,$

where the maximization is restricted to the set of valid parameters, as in (1.22). Many generalizations of this basic framework have been proposed, see for example [82–86] and a recent review paper [87].

By applying Jensen's inequality, one can show [80] that the EM-based surrogate function satisfies the monotonicity inequality (1.23) for all $\underline{\theta}$ and $\underline{\theta}^{(n)}$. Since the M-step ensures that

$$\phi(\underline{\theta}^{(n+1)}; \underline{\theta}^{(n)}) \geq \phi(\underline{\theta}^{(n)}; \underline{\theta}^{(n)}) ,$$

it follows from (1.23) that the above EM recursion is guaranteed to lead to monotone increases in the objective function $\Phi$. Thus, the EM framework is a special case of the optimization transfer approach (1.22), where the surrogate function derives from statistical principles (1.32).

### *1.4.1 Transmission EM algorithm*

Although Rockmore and Macovski proposed ML transmission reconstruction in 1977 [47], it took until 1984 for the first practical algorithm to appear, when Lange and Carson proposed a complete data space for a transmission EM algorithm [17]. Lange and Carson considered the case where $r_i = 0$. In this section we derive a transmission EM algorithm that generalizes that of [17]; we allow $r_i \neq 0$, and we consider arbitrary pixel orderings, as described below. The algorithm of [17] is a special case of what follows.

The "complete" data space that we consider for the case $r_i \neq 0$ is the following collection of random variables, all of which have Poisson marginal distributions:

$$\underline{X} = \{Y_{ik} \,:\, i = 0, 1, \ldots, N_Y,\ k = 1, \ldots, N_p\} \cup \{R_i \,:\, i = 1, \ldots, N_Y\} .$$

We assume that

$$R_i \sim \text{Poisson}\{r_i\}$$

and

$$Y_{i0} \sim \text{Poisson}\{b_i\} ,\ i = 1, \ldots, N_Y ,$$

and that the $R_i$'s are all mutually independent and statistically independent of all the $Y_{ik}$'s. Furthermore, $Y_{ik}$ and $Y_{lk}$ are independent for $i \neq l$. However, $Y_{ik}$ and $Y_{ij}$ are not independent. (The distributions of $R_i$ and $Y_{i0}$ do not depend on $\underline{\mu}$, so are of less importance in what follows.)

For each $i$, let $(j_{i,1}, \ldots, j_{i,N_p})$ be *any* permutation of the set of pixel indices $j = 1, \ldots, N_p$. Notationally, the simplest case is just when $j_{i,k} = k$, which corresponds to the algorithm considered in [88]. Lange and Carson [17] assign $j_{i,k}$ to the physical ordering corresponding to the ray connecting the source to the detector. Statistically, any ordering suffices, and it is an open (and probably academic) question whether certain orderings lead to faster convergence. Given such an ordering, we define the remaining $Y_{ik}$'s recursively by the following conditional distributions:

$$Y_{ik} \mid Y_{i,k-1}, Y_{i,k-2}, \ldots, Y_{i0} \sim \text{Binomial}\Big(Y_{i,k-1}, e^{-\lambda_{i,k}}\Big) ,$$

where

$$\lambda_{i,k} \stackrel{\triangle}{=} a_{ij_{i,k}} \mu_{j_{i,k}} .$$

Thus the conditional probability mass function (PMF) of $Y_{ik}$ for $k = 1, \ldots, N_p$ is given by:

$$p(y_{ik} \,|y_{i,k-1}, y_{i,k-2}, \ldots, y_{i0}; \underline{\mu}) = p(y_{ik} \,|y_{i,k-1}; \underline{\mu})$$

$$= \begin{pmatrix} y_{i,k-1} \\ y_{ik} \end{pmatrix} \left(e^{-\lambda_{i,k}}\right)^{y_{ik}} \left(1 - e^{-\lambda_{i,k}}\right)^{y_{i,k-1} - y_{ik}} , \tag{1.34}$$

where $\begin{pmatrix} n \\ m \end{pmatrix}$ is the binomial coefficient. An alternative way of writing this recursion is

$$Y_{ik} = \sum_{l=1}^{Y_{i,k-1}} Z_{ikl}, \;\; k = 1, \ldots, N_p, \;\; i = 1, \ldots, N_Y ,$$

where $Z_{ikl}$ is a collection of independent 0-1 Bernoulli random variables with

$$P[Z_{ikl} = 1] = e^{-\lambda_{i,k}} .$$

Since a Bernoulli-thinned Poisson process remains Poisson (see Appendix 1.14), it follows that each $Y_{ik}$ has a Poisson distribution:

$$Y_{ik} \sim \text{ Poisson}\left\{ b_i \prod_{l=1}^{k} e^{-\lambda_{i,l}} \right\} \equiv \text{ Poisson}\left\{ b_i e^{-\sum_{l=1}^{k} \lambda_{i,l}} \right\} . \tag{1.35}$$

The special case $k = N_p$ in (1.35) yields

$$Y_{i,N_p} \sim \text{ Poisson}\left\{ b_i e^{-\sum_{l=1}^{N_p} \lambda_{i,l}} \right\} \equiv \text{ Poisson}\left\{ b_i e^{-[\boldsymbol{A}\underline{\mu}]_i} \right\} .$$

Therefore, the observed measurements are related to the complete data space by

$$Y_i = Y_{i,N_p} + R_i ,$$

so the condition (1.31) is satisfied. As noted in [89], there are multiple orderings of the $Y_{ik}$'s that can be considered, each of which would lead to a different update, but which would leave unchanged the limit if there is a unique maximizer (and provided this EM algorithm is globally convergent, which has never been established for the case $r_i \neq 0$).

Figure 1.5 illustrates a loose physical interpretation of the above complete data space. For the $i$th ordering, imagine a sequence of layers of material with attenuation coefficients $\mu_{j_{i,1}}, \ldots, \mu_{j_{i,N_p}}$ and thicknesses $a_{ij_{i,1}}, \ldots, a_{ij_{i,N_p}}$. Suppose a

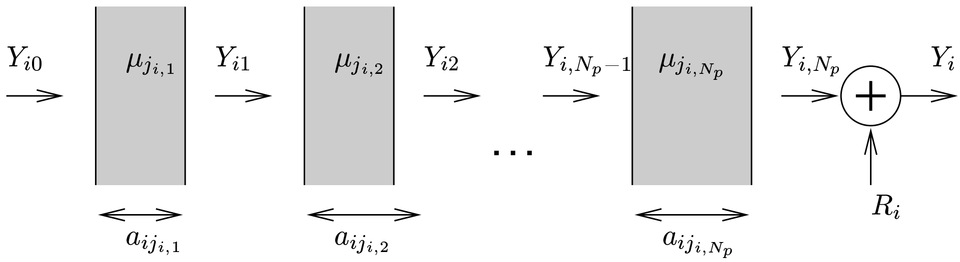



**Figure 1.5**: Pseudo-physical interpretation of transmission EM complete-data space.

Poisson number $Y_{i0}$ of photons is transmitted into the first layer. The number that survive passage through that layer is $Y_{i1}$, which then proceed through the second layer and so on. The final number of photons exiting the sequence of layers is $Y_{i,N_p}$, and this number is added to the random coincidences $R_i$ to form the observed counts $Y_i$. This interpretation is most intuitive when the pixels are ordered according to the actual passage of photons from source to detector (as in [17]), but a physical interpretation is not essential for EM algorithm development.

It follows from (1.35) and Appendix 1.14 that

$$Y_{i,N_p} \,|Y_{ik} \sim \text{Binomial}\left( Y_{ik}, \prod_{l=k+1}^{N_p} e^{-\lambda_{i,l}} \right),$$

since a cascade of independent Binomials is Binomial with the product of the success probabilities.

Having specified the complete-data space, the next step in developing an EM algorithm is to find the surrogate function $Q(\underline{\mu}; \underline{\mu}^{(n)})$ of (1.32). It follows from the above specifications that the joint probability mass function (PMF) of $\underline{X}$ is given by

$$p(\underline{X}; \underline{\mu}) = \prod_{i=1}^{N_Y} p(\{Y_{ik}\}_{k=0}^{N_p}; \underline{\mu}) \prod_{i=1}^{N_Y} p(R_i) \,.$$

By applying the chain rule for conditional probabilities [90] and using (1.34):

$$p(\{Y_{ik}\}_{k=0}^{N_p}; \underline{\mu}) = p(Y_{i0}) \prod_{k=1}^{N_p} p(Y_{ik} \,|Y_{i,k-1}; \underline{\mu})$$

so

$$\begin{aligned}\log p(\underline{X};\underline{\mu}) &\equiv \sum_{i=1}^{N_Y}\sum_{k=1}^{N_p} \log p(Y_{ik}\,|Y_{i,k-1};\underline{\mu}) \\ &\equiv \sum_{i=1}^{N_Y}\sum_{k=1}^{N_p} Y_{ik}\log\Big(e^{-\lambda_{i,k}}\Big) + (Y_{i,k-1}-Y_{ik})\log\Big(1-e^{-\lambda_{i,k}}\Big)\ .\end{aligned}$$

Thus, following [17, 88], the EM-based surrogate function for the above complete data space has the following form:

$$\begin{aligned}Q(\underline{\mu};\underline{\mu}^{(n)}) &= \sum_{i=1}^{N_Y}\sum_{j=1}^{N_p} \bar{N}_{ij}^{(n)}\log\big(e^{-a_{ij}\mu_j}\big) + (\bar{M}_{ij}^{(n)} - \bar{N}_{ij}^{(n)})\log(1-e^{-a_{ij}\mu_j}) \\ &= \sum_{j=1}^{N_p} Q_j(\mu_j;\underline{\mu}^{(n)})\,, \end{aligned} \tag{1.36}$$

where

$$Q_j(\mu_j;\underline{\mu}^{(n)}) \triangleq \sum_{i=1}^{N_Y} \bar{N}_{ij}^{(n)}\log\big(e^{-a_{ij}\mu_j}\big) + (\bar{M}_{ij}^{(n)} - \bar{N}_{ij}^{(n)})\log(1-e^{-a_{ij}\mu_j}) \tag{1.37}$$

$$\bar{N}_{ij}^{(n)} \triangleq E[Y_{ik}\,|Y_i=y_i;\underline{\mu}^{(n)}]\Big|_{k\,:\,j_{i,k}=j} \tag{1.38}$$

$$\bar{M}_{ij}^{(n)} \triangleq E[Y_{i,k-1}\,|Y_i=y_i;\underline{\mu}^{(n)}]\Big|_{k\,:\,j_{i,k}=j}\ . \tag{1.39}$$

To complete the E-step of the EM algorithm, we must find the preceding conditional expectations. Using the law of iterated expectation [90]:

$$\begin{aligned}E[Y_{ik}\,|Y_i=y_i] &= E\left[E[Y_{ik}\,|R_i, Y_i=y_i]\mid Y_i=y_i\right] \\ &= E\left[E[Y_{ik}\,|Y_{i,N_p}=y_i-R_i]\mid Y_i=y_i\right] \\ &= E\left[E[Y_{ik}]-E[Y_{i,N_p}]+y_i-R_i\mid Y_i=y_i\right] \\ &= E[Y_{ik}]-E[Y_{i,N_p}]+y_i-y_i\frac{r_i}{r_i+E[Y_{i,N_p}]}\end{aligned} \tag{1.40}$$

using results from [17] and Appendix 1.14. From (1.35)

$$E[Y_{ij};\underline{\mu}^{(n)}] = \gamma_{ij}^{(n)} \triangleq b_i\prod_{l=1}^{k} e^{-a_{ij_{i,l}}\mu_{j_{i,l}}^{(n)}} = b_i e^{-\sum_{l=1}^{k} a_{ij_{i,l}}\mu_{j_{i,l}}^{(n)}}\Bigg|_{k:j_{i,k}=j}\,, \tag{1.41}$$

from which one can show using (1.40) that

$$\bar{M}_{ij}^{(n)} - \bar{N}_{ij}^{(n)} = \gamma_{ij}^{(n)} (e^{a_{ij}\mu_j^{(n)}} - 1) \geq 0 \,. \tag{1.42}$$

Combining (1.36), (1.37), (1.40), and (1.41) yields an explicit expression for the EM surrogate $Q(\underline{\mu}; \underline{\mu}^{(n)})$, completing the E-step.

For the M-step of the EM algorithm, we must find the maximizer of $Q(\cdot; \underline{\mu}^{(n)})$. The function $Q(\cdot; \underline{\mu}^{(n)})$ is a *separable function* of the $\mu_j$'s, as shown in (1.36), so it is easier to maximize than $\Phi$. Thus the M-step reduces to the $N_p$ separable 1D maximization problems:

$$\mu_j^{(n+1)} = \arg\max_{\mu_j \geq 0} Q_j(\mu_j; \underline{\mu}^{(n)}) \,. \tag{1.43}$$

Unfortunately however, due to the transcendental functions in (1.36), there is no closed-form expression for the maximizer of $Q_j(\cdot; \underline{\mu}^{(n)})$. In fact, finding the maximizer of $Q_j(\cdot; \underline{\mu}^{(n)})$ is no easier than maximizing $\Phi(\underline{\mu})$ with respect to $\mu_j$ while holding the other parameters fixed, which is the coordinate ascent algorithm described in Section 1.5. Nevertheless, the EM algorithm is parallelizable, unlike the coordinate ascent algorithm, so we proceed here with its description. Zeroing the derivative of $Q_j$ in (1.37) yields the following:

$$0 = \frac{d}{d\mu_j} Q_j(\mu_j; \underline{\mu}^{(n)}) = \sum_{i=1}^{N_Y} -\bar{N}_{ij}^{(n)} a_{ij} + (\bar{M}_{ij}^{(n)} - \bar{N}_{ij}^{(n)}) \frac{a_{ij} e^{-a_{ij}\mu_j}}{1 - e^{-a_{ij}\mu_j}} \,, \tag{1.44}$$

the solution to which is $\mu_j^{(n+1)}$. Unfortunately, (1.44) can only be solved for $\mu_j$ analytically when the $a_{ij}$'s are all equal. However, Lange and Carson [17] noted that typically $a_{ij}\mu_j$ will be small (*much* less than unity), so that the following Taylor-series expansion around zero should be a reasonable approximation:

$$\frac{x e^{-x}}{1 - e^{-x}} \approx 1 - \frac{x}{2} \text{ for } x \approx 0 \,.$$

Applying this approximation to (1.44) with $x = a_{ij}\mu_j$ yields:

$$\begin{aligned} \sum_{i=1}^{N_Y} \bar{N}_{ij}^{(n)} a_{ij} &\approx \sum_{i=1}^{N_Y} (\bar{M}_{ij}^{(n)} - \bar{N}_{ij}^{(n)}) \frac{1}{\mu_j} \left(1 - \frac{a_{ij}\mu_j}{2}\right) \\ &= \sum_{i=1}^{N_Y} (\bar{M}_{ij}^{(n)} - \bar{N}_{ij}^{(n)}) \left(\frac{1}{\mu_j} - \frac{a_{ij}}{2}\right) \,. \end{aligned}$$

Solving this equality for $\mu_j$ yields the final iterative form for the ML transmission EM algorithm [17]:

$$\boxed{\mu_j^{(n+1)} \approx \frac{\sum_{i=1}^{N_Y} (\bar{M}_{ij}^{(n)} - \bar{N}_{ij}^{(n)})}{\frac{1}{2} \sum_{i=1}^{N_Y} a_{ij} (\bar{M}_{ij}^{(n)} + \bar{N}_{ij}^{(n)})} \,.} \tag{1.45}$$

This algorithm is very slow to converge [69] and each iteration is very computationally expensive due to the large number of exponentiations required in (1.41). One exponentiation per nonzero $a_{ij}$ is required.

Lange and Carson [17] also describe an update based on a second-order Taylor series, and they note that one can use their expansion to find upper and lower bounds for the exact value of $\mu_j$ that maximizes $Q_j(\cdot; \underline{\mu}^{(n)})$.

### 1.4.2 *EM algorithms with approximate M-steps*

Since the M-step of the transmission EM algorithm of [17] did not yield a closed form for the maximizer, Browne and Holmes [91] proposed a modified EM algorithm that used an approximate M-step based on image rotations using bilinear interpolation. Kent and Wright made a similar approximation [89]. An advantage of these methods is that (after interpolation) the $a_{ij}$'s are all equal, which is the case where one can solve (1.44) analytically. Specifically, if $a_{ij} = a_0$ for all $i$ and $j$, then (1.44) simplifies to

$$\sum_{i=1}^{N_Y} \tilde{N}_{ij}^{(n)} = \sum_{i=1}^{N_Y} (\tilde{M}_{ij}^{(n)} - \tilde{N}_{ij}^{(n)}) \frac{1}{e^{a_0 \mu_j} - 1} \,,$$

where $\tilde{N}_{ij}^{(n)}$ and $\tilde{M}_{ij}^{(n)}$ replace $\bar{M}_{ij}^{(n)}$ and $\bar{N}_{ij}^{(n)}$ respectively, in rotated coordinates. When solved for $\mu_j$, this yields the iteration

$$\boxed{\mu_j^{(n+1)} = \frac{1}{a_0} \log \left( \frac{\sum_{i=1}^{N_Y} \tilde{M}_{ij}^{(n)}}{\sum_{i=1}^{N_Y} \tilde{N}_{ij}^{(n)}} \right) \,,} \tag{1.46}$$

which is the logarithm of the ratio of (conditional expectations of) the number of photons entering the $j$th pixel to the number of photons leaving the $j$th pixel, divided by the pixel size. However, the interpolations required to form $\tilde{N}_{ij}^{(n)}$ and $\tilde{M}_{ij}^{(n)}$ presumably destroy the monotonicity properties of the EM algorithm. Although bookkeeping is reduced, these methods require the same (very large) number of exponentiations as the original transmission EM algorithm, so they are also impractical algorithms.

### 1.4.3 *EM algorithm with Newton M-step*

Ollinger [92,93] reported that the M-step approximation (1.45) proposed in [17] led to convergence problems, and proposed a 1D Newton's method for maximizing $Q$ in the context of a GEM algorithm for the M-step. Since Newton's method is not guaranteed to converge, the step length was adjusted by a halving strategy to ensure a monotone increase in the surrogate function:

$$\mu_j^{(n+1)} = \mu_j^{(n)} + \alpha_{j,n} \frac{\left.\frac{d}{d\mu_j} Q_j(\mu_j; \underline{\mu}^{(n)})\right|_{\mu_j=\mu_j^{(n)}}}{\left.\frac{d^2}{d\mu_j^2} Q_j(\mu_j; \underline{\mu}^{(n)})\right|_{\mu_j=\mu_j^{(n)}}}, \tag{1.47}$$

where one ensures that $Q_j(\mu_j^{(n+1)}; \underline{\mu}^{(n)}) \geq Q_j(\mu_j^{(n)}; \underline{\mu}^{(n)})$ by choosing $\alpha_{j,n}$ via a line-search. This line-search can require multiple evaluations of $Q_j$ as each pixel is updated, which is relatively expensive. The large number of exponentiations required to compute $\bar{M}_{ij}^{(n)}$ and $\bar{N}_{ij}^{(n)}$ also remains a drawback.

From (1.44) and (1.42),

$$\begin{aligned}
\left.\frac{d}{d\mu_j} Q_j(\mu_j; \underline{\mu}^{(n)})\right|_{\mu_j=\mu_j^{(n)}} &= \sum_{i=1}^{N_Y} a_{ij} \left[ \frac{\bar{M}_{ij}^{(n)} - \bar{N}_{ij}^{(n)}}{1 - e^{-a_{ij}\mu_j^{(n)}}} e^{-a_{ij}\mu_j^{(n)}} - \bar{N}_{ij}^{(n)} \right] \\
&= \sum_{i=1}^{N_Y} a_{ij} \left( \gamma_{ij}^{(n)} - \left[ \gamma_{ij}^{(n)} - b_i e^{-l_i^{(n)}} + \frac{y_i b_i e^{-l_i^{(n)}}}{b_i e^{-l_i^{(n)}} + r_i} \right] \right) \\
&= \sum_{i=1}^{N_Y} a_{ij} \left( 1 - \frac{y_i}{b_i e^{-l_i^{(n)}} + r_i} \right) b_i e^{-l_i^{(n)}} = \left.\frac{d}{d\mu_j} L(\underline{\mu})\right|_{\underline{\mu}=\underline{\mu}^{(n)}},
\end{aligned}$$

so (1.25) is indeed satisfied. From (1.44) and (1.42),

$$-\frac{d^2}{d\mu_j^2} Q_j(\mu_j; \underline{\mu}^{(n)}) = \sum_{i=1}^{N_Y} (\bar{M}_{ij}^{(n)} - \bar{N}_{ij}^{(n)}) \frac{a_{ij}^2 e^{-a_{ij}\mu_j}}{(1 - e^{-a_{ij}\mu_j})^2}, \tag{1.48}$$

so

$$-\left.\frac{d^2}{d\mu_j^2} Q_j(\mu_j; \underline{\mu}^{(n)})\right|_{\mu_j=\mu_j^{(n)}} = \sum_{i=1}^{N_Y} a_{ij}^2 \gamma_{ij}^{(n)} / (1 - e^{-a_{ij}\mu_j^{(n)}}).$$

Thus, the ML EM Newton-Raphson (EM-NR) algorithm (1.47) becomes

$$\boxed{\mu_j^{(n+1)} = \mu_j^{(n)} + \alpha_{j,n} \frac{\sum_{i=1}^{N_Y} a_{ij} \left( 1 - \frac{y_i}{b_i e^{-l_i^{(n)}} + r_i} \right) b_i e^{-l_i^{(n)}}}{\sum_{i=1}^{N_Y} a_{ij}^2 \gamma_{ij}^{(n)} / (1 - e^{-a_{ij}\mu_j^{(n)}})}.} \tag{1.49}$$

From (1.48), the curvature of $Q_j$ becomes unbounded as $\mu_j \to 0$, which appears to preclude the use of parabola surrogates as described in Section 1.4.5.2 to form an intrinsically monotonic M-step (1.43).

Variations on the transmission EM algorithm continue to resurface at conferences, despite its many drawbacks. The endurance of the transmission EM algorithm can only be explained by its having “ridden on the coat tails” of the popular emission EM algorithm. The modern methods described in subsequent sections are entirely preferable to the transmission EM algorithm.

### *1.4.4 Diagonally-scaled gradient-ascent algorithms*

Several authors, e.g. [94], noted that the emission EM algorithm can be expressed as a diagonally-scaled, gradient-ascent algorithm, with a particular diagonal scaling matrix that (almost miraculously) ensures monotonicity and preserves nonnegativity. (The EM-NR algorithm (1.49) has a similar form.) Based on an analogy with that emission EM algorithm, Lange et al. proposed a diagonally-scaled gradient-ascent algorithm for transmission tomography [95]. The algorithm can be expressed as the following recursion:

$$\boxed{\underline{\mu}^{(n+1)} = \underline{\mu}^{(n)} + D(\underline{\mu}^{(n)})\nabla' L(\underline{\mu}^{(n)}) \, ,} \tag{1.50}$$

where $D(\cdot)$ is some iteration-dependent diagonal matrix and where $\nabla'$ denotes the column gradient operator (cf (1.21)):

$$\left[\nabla' L(\underline{\mu})\right]_j = \frac{\partial}{\partial \mu_j} L(\underline{\mu}) = \sum_{i=1}^{N_Y} a_{ij} \left[1 - \frac{y_i}{b_i e^{-[\boldsymbol{A}\underline{\mu}]_i} + r_i}\right] b_i e^{-[\boldsymbol{A}\underline{\mu}]_i} \, . \tag{1.51}$$

Since the gradient of the objective function is evaluated once per iteration, the number of exponentiations required is roughly $N_Y$, far fewer than required by the transmission EM algorithm (1.45).

The choice of the $N_p \times N_p$ diagonal scaling matrix $D(\underline{\mu})$ critically affects convergence rate, monotonicity, and nonnegativity. Considering the case $r_i = 0$, Lange et al. [95] suggested the following diagonal scaling matrix, chosen so that (1.50) could be expressed as a multiplicative update in the case $r_i = 0$:

$$D(\underline{\mu})_{jj} = \frac{\mu_j}{\sum_{i=1}^{N_Y} a_{ij} y_i} \, . \tag{1.52}$$

The natural generalization of this choice to the general case where $r_i \neq 0$ is the diagonal matrix with the following expression for the $j$th diagonal element:

$$D(\underline{\mu})_{jj} = \frac{\mu_j}{\sum_{i=1}^{N_Y} a_{ij} y_i b_i e^{-[\boldsymbol{A}\underline{\mu}]_i} / (b_i e^{-[\boldsymbol{A}\underline{\mu}]_i} + r_i)} \, . \tag{1.53}$$

Using (1.51) and (1.53) one can rewrite the diagonally-scaled gradient ascent (DS-GA) algorithm (1.50) as follows:

$$\boxed{\mu_j^{(n+1)} = \mu_j^{(n)} \frac{\sum_{i=1}^{N_Y} a_{ij} b_i e^{-l_i^{(n)}}}{\sum_{i=1}^{N_Y} a_{ij} y_i b_i e^{-l_i^{(n)}} / (b_i e^{-l_i^{(n)}} + r_i)} \, .} \tag{1.54}$$

This is a multiplicative update that preserves nonnegativity, and at least its positive fixed points are stationary points of the log-likelihood. However, the particular choice of diagonal scaling matrix (1.53) does not guarantee intrinsically monotone increases in the likelihood function. In Section 1.6.2 below, we present one form of a paraboloidal surrogates algorithm that has the same general form as (1.50) but overcomes the limitations of (1.54) by choosing $D$ appropriately.

Lange et al. also proposed modifications of the iteration (1.50) to include a separable penalty function and a line-search to enforce monotonicity [95] (for $r_i = 0$ case, but the ideas generalize easily to the $r_i \neq 0$ case).

Lange proposed another diagonally-scaled gradient-ascent algorithm in [96], based on the following diagonal scaling matrix:

$$D(\underline{\mu})_{jj} = \frac{\mu_j}{\sum_{i=1}^{N_Y} a_{ij} b_i e^{-[\boldsymbol{A}\underline{\mu}]_i} [\boldsymbol{A}\underline{\mu}]_i} . \tag{1.55}$$

Although the rationale for this choice was not given in [96], Lange was able to show that the algorithm has local convergence properties, but that it may not yield nonnegative estimates. Lange further modified the scaled-gradient algorithm in [97] to include nonseparable penalty functions, and a practical approximate line-search that ensures global convergence for $r_i = 0$.

Considering the case $r_i = 0$, Maniawski et al. [98] proposed the following over-relaxed unregularized version of the diagonally-scaled gradient-ascent algorithm (1.50):

$$\boxed{\mu_j^{(n+1)} = \mu_j^{(n)} \left[ \omega \frac{\sum_{i=1}^{N_Y} a_{ij} b_i e^{-l_i^{(n)}}}{\sum_{i=1}^{N_Y} a_{ij} y_i} + (1 - \omega) \right] ,} \tag{1.56}$$

where $\omega$ was selected empirically to be $4 \cdot 10^{-8}$ times the total number of measured counts in a SPECT transmission scan. Like (1.54), this is a multiplicative update that preserves nonnegativity. One can also express the above algorithm more generally as follows:

$$\underline{\mu}^{(n+1)} = (1 - \omega)\underline{\mu}^{(n)} + \omega D(\underline{\mu}^{(n)}) \nabla' L(\underline{\mu}^{(n)}) ,$$

where $D(\underline{\mu})$ is chosen as in (1.52). No convergence analysis was discussed for the algorithm (1.56), although "fast convergence" was reported.

### *1.4.5 Convex algorithm*

De Pierro [72] described a non-statistical derivation of the emission EM algorithm using the concavity properties of the log-likelihood for emission tomography. Lange and Fessler [69] applied a similar derivation to the transmission log-likelihood for the case $r_i = 0$, yielding a "convex[9] algorithm" that, like the transmission EM algorithm, is guaranteed to monotonically increase $L(\underline{\mu})$ each iteration. As discussed in Section 1.2.5, the transmission log-likelihood is concave

[9]The algorithm name is unfortunate, since the algorithm itself is not convex, but rather the algorithm is derived by exploiting the concavity of the log-likelihood.

when $r_i = 0$, so De Pierro's convexity method could be applied directly in [69]. In the case $r_i \neq 0$, the log-likelihood is not concave, so De Pierro's convexity argument does not directly apply. Fessler [14] noted that even when $r_i \neq 0$, the marginal log-likelihood functions (the $h_i$'s in (1.11)) are concave over a (typically) large interval of the real line, and thereby developed an "approximate" convex algorithm. However, the "convex algorithm" of [14] is not guaranteed to be globally monotonic.

Rather than presenting either the convex algorithm of [69], which is incomplete since it did not consider the case $r_i \neq 0$, or the algorithm of [14], which is non-monotone, we derive a new "convex" algorithm here. The algorithm of [69] falls out as a special case of this new algorithm by setting $r_i = 0$. The idea is to first use the EM algorithm to find a concave surrogate function $Q_1$ that "eliminates" the $r_i$ terms, but is still difficult to maximize directly; then we apply De Pierro's convexity argument to $Q_1$ to find another surrogate function $Q_2$ that is easily maximized. The same idea was developed independently by Kim [99].

Consider a "complete" data space that is the collection of the following statistically independent random variables:

$$\underline{X} = \{\{N_i\}_{i=1}^{N_Y},\ \{R_i\}_{i=1}^{N_Y}\},$$

where

$$\begin{aligned} N_i &\sim \text{Poisson}\left\{b_i e^{-[\boldsymbol{A}\underline{\mu}]_i}\right\} \\ R_i &\sim \text{Poisson}\{r_i\} \end{aligned}$$

and where the observed measurements are related to the elements of $\underline{X}$ by

$$Y_i = N_i + R_i\,,$$

so the condition (1.31) is satisfied. The complete-data log-likelihood is simply:

$$\log p(\underline{X}; \underline{\mu}) \equiv \sum_{i=1}^{N_Y} N_i \log\left(b_i e^{-[\boldsymbol{A}\underline{\mu}]_i}\right) - \left(b_i e^{-[\boldsymbol{A}\underline{\mu}]_i}\right)\ ,$$

since the distribution of the $R_i$'s is a constant independent of $\underline{\mu}$, so can be ignored. Since by Appendix 1.14

$$\bar{N}_i^{(n)} \triangleq E[N_i\,|Y_i = y_i; \underline{\mu}^{(n)}] = y_i \frac{b_i e^{-l_i^{(n)}}}{b_i e^{-l_i^{(n)}} + r_i}\,,$$

the EM surrogate function is the following concave function:

$$Q_1(\underline{\mu}; \underline{\mu}^{(n)}) = \sum_{i=1}^{N_Y} -\bar{N}_i^{(n)} [\boldsymbol{A}\underline{\mu}]_i - b_i e^{-[\boldsymbol{A}\underline{\mu}]_i} = \sum_{i=1}^{N_Y} g_i^{(n)}([\boldsymbol{A}\underline{\mu}]_i) \tag{1.57}$$

where

$$g_i^{(n)}(l) \triangleq -\bar{N}_i^{(n)} l - b_i e^{-l} . \tag{1.58}$$

The form of (1.57) is identical to the form that (1.11) would have if all the $r_i$'s were zero. Therefore, by this technique we can generalize any algorithm that has been derived for the $r_i = 0$ case to the realistic case where $r_i \neq 0$ simply by replacing $y_i$ in the algorithm with $\bar{N}_i^{(n)}$. However, in general the convergence of an algorithm derived this way may be slower than methods based on direct maximization of $L(\underline{\mu})$ since the curvatures of the $Q_1$ components are smaller than those of $L$ since $\bar{N}_i^{(n)} \leq y_i$. For rays where the random fraction is large, $\bar{N}_i^{(n)} \ll y_i$, leading to slow convergence rates. In statistical terms, the complete-data space $\underline{X}$ is much more informative than the observed data $\underline{Y}$ [100].

We could attempt to naively perform the M-step of the EM algorithm:

$$\underline{\mu}^{(n+1)} = \arg\max_{\underline{\mu} \geq \underline{0}} Q_1(\underline{\mu}; \underline{\mu}^{(n)}) ,$$

except that maximizing $Q_1$ is (almost[10]) as difficult as maximizing the original log-likelihood.

By differentiating twice, one can easily show that each $g_i^{(n)}$ is a concave function and that $Q_1(\cdot; \underline{\mu}^{(n)})$ is a concave function. Therefore, rather than maximizing $Q_1$ directly, we find a surrogate function for $Q_1$ by applying the convexity method of De Pierro [72]. The essence of De Pierro's method is the following clever expression for matrix-vector multiplication:

$$[\boldsymbol{A}\underline{\mu}]_i = \sum_{j=1}^{N_p} \alpha_{ij} \left[ \frac{a_{ij}}{\alpha_{ij}} (\mu_j - \mu_j^{(n)}) + l_i^{(n)} \right] \tag{1.59}$$

where the projection of the current attenuation map estimate is given by

$$l_i^{(n)} \triangleq [\boldsymbol{A}\underline{\mu}^{(n)}]_i .$$

The expression (1.59) holds for any collection of $\alpha_{ij}$'s, provided $\alpha_{ij} = 0$ only if $a_{ij} = 0$ and

$$\sum_{j=1}^{N_p} \alpha_{ij} = 1 .$$

If we choose nonnegative $\alpha_{ij}$'s, then because each $g_i^{(n)}$ is concave, by (1.59):

$$g_i^{(n)}([\boldsymbol{A}\underline{\mu}]_i) = g_i^{(n)} \left( \sum_{j=1}^{N_p} \alpha_{ij} \left[ \frac{a_{ij}}{\alpha_{ij}} (\mu_j - \mu_j^{(n)}) + l_i^{(n)} \right] \right) \geq \sum_{j=1}^{N_p} \alpha_{ij} \, g_{ij}^{(n)}(\mu_j; \underline{\mu}^{(n)}) , \tag{1.60}$$

[10] $Q_1$ is concave, unlike $L$.

where

$$g_{ij}^{(n)}(\mu_j; \underline{\mu}^{(n)}) \triangleq g_i^{(n)}\left(\frac{a_{ij}}{\alpha_{ij}}(\mu_j - \mu_j^{(n)}) + l_i^{(n)}\right).$$

Thus, a suitable surrogate function for $Q_1(\underline{\mu}; \underline{\mu}^{(n)})$ is

$$Q_2(\underline{\mu}; \underline{\mu}^{(n)}) \triangleq \sum_{i=1}^{N_Y} \sum_{j=1}^{N_p} \alpha_{ij}\, g_{ij}^{(n)}(\mu_j; \underline{\mu}^{(n)}) = \sum_{j=1}^{N_p} Q_{2,j}(\mu_j; \underline{\mu}^{(n)}) \tag{1.61}$$

where

$$Q_{2,j}(\mu_j; \underline{\mu}^{(n)}) \triangleq \sum_{i=1}^{N_Y} \alpha_{ij}\, g_{ij}^{(n)}(\mu_j; \underline{\mu}^{(n)}) = \sum_{i=1}^{N_Y} \alpha_{ij} g_i^{(n)}\left(\frac{a_{ij}}{\alpha_{ij}}(\mu_j - \mu_j^{(n)}) + l_i^{(n)}\right). \tag{1.62}$$

Since $Q_2$ is a *separable function*, its maximization reduces to $N_p$ simultaneous maximization problems:

$$\mu_j^{(n+1)} = \arg\max_{\mu_j \geq 0} Q_{2,j}(\mu_j; \underline{\mu}^{(n)}). \tag{1.63}$$

Unfortunately there is not a closed-form analytical solution for the maximizer, so we must apply approximations, line searches, or the optimization transfer principle.

### *1.4.5.1 Convex-NR algorithms*

A simple "solution" to (1.63) is to apply one or more Newton-Raphson steps, as in (1.47). Such an algorithm should be locally convergent, and can presumably be made globally convergent by a line-search modification of the type proposed by Lange [97]. From (1.62), (1.57), and (1.13):

$$\begin{aligned}
\frac{d}{d\mu_j} Q_{2,j}(\mu_j; \underline{\mu}^{(n)})\Big|_{\mu_j = \mu_j^{(n)}} &= \sum_{i=1}^{N_Y} a_{ij} \dot{g}_i^{(n)}(l_i^{(n)}) \\
= \sum_{i=1}^{N_Y} a_{ij} \left[b_i e^{-l_i^{(n)}} - \bar{N}_i^{(n)}\right] &= \sum_{i=1}^{N_Y} a_{ij} \left[b_i e^{-l_i^{(n)}} - y_i \frac{b_i e^{-l_i^{(n)}}}{b_i e^{-l_i^{(n)}} + r_i}\right] \\
= \sum_{i=1}^{N_Y} a_{ij} \left(1 - \frac{y_i}{b_i e^{-l_i^{(n)}} + r_i}\right) b_i e^{-l_i^{(n)}} &= \frac{\partial}{\partial \mu_j} L(\underline{\mu})\Big|_{\underline{\mu} = \underline{\mu}^{(n)}},
\end{aligned}$$

so

$$\nabla' Q_2(\underline{\mu}; \underline{\mu}^{(n)})\Big|_{\underline{\mu} = \underline{\mu}^{(n)}} = \nabla' L(\underline{\mu}; \underline{\mu}^{(n)})\Big|_{\underline{\mu} = \underline{\mu}^{(n)}}.$$

Thus the new convex algorithm (1.63), based on one 1D Newton-Raphson step for each pixel, has the same form as the diagonally-scaled gradient-ascent algorithm (1.50), except that it uses the following diagonal scaling matrix:

$$[D(\underline{\mu}^{(n)})]_{jj} = \frac{1}{-\frac{d^2}{d\mu_j^2} Q_{2,j}(\mu_j; \underline{\mu}^{(n)})\Big|_{\mu_j=\mu_j^{(n)}}},$$

where from (1.62) and (1.58)

$$-\frac{d^2}{d\mu_j^2} Q_{2,j}(\mu_j; \underline{\mu}^{(n)})\Bigg|_{\mu_j=\mu_j^{(n)}} = -\sum_{i=1}^{N_Y} \frac{a_{ij}^2}{\alpha_{ij}} \ddot{g}_i^{(n)}(\mu_j; \underline{\mu}^{(n)}) = \sum_{i=1}^{N_Y} \frac{a_{ij}^2}{\alpha_{ij}} b_i e^{-l_i^{(n)}}. \tag{1.64}$$

In [69] and [14], the following choice for the $\alpha_{ij}$'s was used, following [72]

$$\alpha_{ij} = \frac{a_{ij}\mu_j^{(n)}}{\sum_{k=1}^{N_p} a_{ik}\mu_k^{(n)}} = \frac{a_{ij}\mu_j^{(n)}}{l_i^{(n)}}. \tag{1.65}$$

Substituting into (1.64) etc. yields the *Convex-NR-1 algorithm*:

$$\boxed{\mu_j^{(n+1)} = \mu_j^{(n)} + \mu_j^{(n)} \frac{\sum_{i=1}^{N_Y} a_{ij}\left(1 - \frac{y_i}{b_i e^{-l_i^{(n)}} + r_i}\right) b_i e^{-l_i^{(n)}}}{\sum_{i=1}^{N_Y} a_{ij} l_i^{(n)} b_i e^{-l_i^{(n)}}}.} \tag{1.66}$$

The diagonal scaling matrix of this Convex-NR-1 algorithm is identical to (1.55), which is interesting since (1.55) was presented for the case $r_i = 0$. There are potential problems with the choice (1.65) when the $\mu_j$'s approach zero [69], so the following alternative choice, considered in [73] and [101], may be preferable:

$$\alpha_{ij} = \frac{a_{ij}}{\sum_{k=1}^{N_p} a_{ik}} = \frac{a_{ij}}{a_i}, \tag{1.67}$$

where $a_i \stackrel{\triangle}{=} \sum_{j=1}^{N_p} a_{ij}$, for which (1.64) leads to

$$[D(\underline{\mu})]_{jj} = \frac{1}{\sum_{i=1}^{N_Y} a_{ij} a_i \, b_i e^{-l_i^{(n)}}}.$$

A small advantage of the choice (1.67) over (1.65) is that the $a_i$ 's in the denominator of (1.67) are independent of $\underline{\mu}$ so they can be precomputed, unlike the denominator of (1.65).

Substituting the above into (1.50) yields the following *Convex-NR-2* algorithm:

$$\boxed{\mu_j^{(n+1)} = \mu_j^{(n)} + \frac{\sum_{i=1}^{N_Y} a_{ij} \left(1 - \frac{y_i}{b_i e^{-l_i^{(n)}} + r_i}\right) b_i e^{-l_i^{(n)}}}{\sum_{i=1}^{N_Y} a_{ij} a_i \, b_i e^{-l_i^{(n)}}}.} \tag{1.68}$$

Each iteration requires one forward projection (to compute the $l_i^{(n)}$'s) and two back-projections (one each for the numerator and denominator). In general a line search would be necessary with this algorithm to ensure monotonicity and convergence.

### *1.4.5.2 Convex-PS algorithm*

The function $Q_{2,j}$ in (1.62) cannot be maximized analytically, but we can apply the optimization transfer principle of Section 1.3.2 to derive the first intrinsically monotonic algorithm presented in this chapter. Using the surrogate parabola (1.27):

$$g_i^{(n)}(l) \geq q_i^{(n)}(l; l_i^{(n)}) \triangleq g_i^{(n)}(l_i^{(n)}) + \dot{g}_i^{(n)}(l_i^{(n)})(l - l_i^{(n)}) - \frac{c_i^{(n)}}{2}(l - l_i^{(n)})^2, \; \forall l \geq 0 \tag{1.69}$$

where from (1.29) the optimal curvature is

$$c_i^{(n)} = c(l_i^{(n)}; \bar{N}_i^{(n)}, b_i, 0) = \begin{cases} \left[2(b_i - l_i^{(n)} b_i e^{-l_i^{(n)}} - l_i^{(n)})/(l_i^{(n)})^2\right]_+, & l_i^{(n)} > 0 \\ b_i, & l_i^{(n)} = 0. \end{cases} \tag{1.70}$$

This suggests the following quadratic surrogate function

$$Q_{3,j}(\mu_j; \underline{\mu}^{(n)}) \triangleq \sum_{i=1}^{N_Y} q_i^{(n)}\left(\frac{a_{ij}}{\alpha_{ij}}(\mu_j - \mu_j^{(n)}) + l_i^{(n)}; l_i^{(n)}\right),$$

with corresponding algorithm

$$\mu_j^{(n+1)} = \arg\max_{\mu_j \geq 0} Q_{3,j}(\mu_j; \underline{\mu}^{(n)}). \tag{1.71}$$

Since $Q_{3,j}$ is quadratic, it is trivial to maximize analytically. Furthermore, since this is a 1D maximization problem for a concave function, to enforce the nonnegativity constraint we simply reset any negative pixels to zero. The derivatives of $Q_{3,j}$ are:

$$\begin{aligned}
\frac{d}{d\mu_j} Q_{3,j}(\mu_j; \underline{\mu}^{(n)})\bigg|_{\mu_j = \mu_j^{(n)}} &= \sum_{i=1}^{N_Y} a_{ij} \dot{q}_i^{(n)}(l_i^{(n)}; l_i^{(n)}) \\
&= \sum_{i=1}^{N_Y} a_{ij} \dot{g}_i^{(n)}(l_i^{(n)}; l_i^{(n)}) = \sum_{i=1}^{N_Y} a_{ij}(1 - \bar{N}_i^{(n)}/(b_i e^{-l_i^{(n)}}) b_i e^{-l_i^{(n)}} \\
&= \sum_{i=1}^{N_Y} a_{ij}(1 - y_i/(b_i e^{-l_i^{(n)}} + r_i) b_i e^{-l_i^{(n)}} = \frac{d}{d\mu_j} L(\underline{\mu})\bigg|_{\underline{\mu} = \underline{\mu}^{(n)}},
\end{aligned}$$

$$\left.\frac{d^2}{d\mu_j^2} Q_{3,j}(\mu_j; \underline{\mu}^{(n)})\right|_{\mu_j=\mu_j^{(n)}} = \sum_{i=1}^{N_Y} \frac{a_{ij}^2}{\alpha_{ij}} c_i^{(n)}.$$

Using the choice (1.67) for the $\alpha_{ij}$'s yields the following *Convex-PS algorithm*:

$$\boxed{\mu_j^{(n+1)} = \left[ \mu_j^{(n)} + \frac{\sum_{i=1}^{N_Y} a_{ij} \left(1 - \frac{y_i}{b_i e^{-l_i^{(n)}} + r_i}\right) b_i e^{-l_i^{(n)}}}{\sum_{i=1}^{N_Y} a_{ij} a_i \, c_i^{(n)}} \right]_+ .} \tag{1.72}$$

The $[\cdot]_+$ operation enforces the nonnegativity constraint. Since this is the first intrinsically monotonic algorithm presented in this chapter, we provide the following more detailed description of its implementation.

for $n = 0, 1, \ldots$ {

$$\begin{aligned} l_i^{(n)} &:= [\boldsymbol{A}\underline{\mu}^{(n)}]_i, \; i = 1, \ldots, N_Y \\ \dot{h}_i^{(n)} &:= \left(1 - \frac{y_i}{b_i e^{-l_i^{(n)}} + r_i}\right) b_i e^{-l_i^{(n)}}, \; i = 1, \ldots, N_Y \end{aligned} \tag{1.73}$$

compute $c_i^{(n)}$ using (1.70), $i = 1, \ldots, N_Y$
$e_i^{(n)} := a_i \, c_i^{(n)}, \; i = 1, \ldots, N_Y$
for $j = 1, \ldots, N_p$ {

$$\mu_j^{(n+1)} := \left[ \mu_j^{(n)} + \frac{\sum_{i=1}^{N_Y} a_{ij} \dot{h}_i^{(n)}}{\sum_{i=1}^{N_Y} a_{ij} e_i^{(n)}} \right]_+ \tag{1.74}$$

}

} .

This ML algorithm monotonically increases the log-likelihood function $L(\underline{\mu}^{(n)})$ each iteration.

To derive (1.72), we have used all three of the optimization transfer principles that are present in this chapter: the EM approach in (1.57), the convex separability approach in (1.61), and the parabola surrogate approach in (1.69). Undoubtably there are other algorithms awaiting discovery by using different combinations of these principles!

### *1.4.6 Ordered-subsets EM algorithm*

Hudson and Larkin [102] proposed an "ordered subsets" modification of the emission EM algorithm in which one updates the image estimate using a sequence of subsets of the measured data, subsampled by projection angle, rather than using all the measurements simultaneously. Manglos et al. [44] applied this concept to the transmission EM algorithm (1.45), yielding the iteration:

$$\boxed{\mu_j^{(n+1)} \triangleq \frac{\sum_{i \in \mathcal{S}_n} (\bar{M}_{ij}^{(n)} - \bar{N}_{ij}^{(n)})}{\frac{1}{2} \sum_{i \in \mathcal{S}_n} a_{ij} (\bar{M}_{ij}^{(n)} + \bar{N}_{ij}^{(n)})},} \tag{1.75}$$

where $\mathcal{S}_n$ is a subset of the ray indices $\{1, \ldots, N_Y\}$ selected for the $n$th subiteration. This type of modification destroys the monotonicity properties of the EM algorithm, and typically the sequence of images asymptotically approaches a limit cycle [103–105]. However, at least in the emission case, the OSEM algorithm seems to produce visually appealing images fairly quickly and hence has become very popular.

*Any* of the algorithms described in this chapter could be easily modified to have a block-iterative form akin to (1.75) simply by replacing any ray summations (those over $i$) with partial summations over $i \in \mathcal{S}_n$. Since (1.75) requires the same number of exponentiations per iteration as the transmission EM algorithm (1.45), it is still impractical. However, block-iterative forms of some of the other algorithms described in this chapter are practical. In particular, Nuyts et al. proposed a block-iterative modification of a gradient-based method [106] (only in the case $r_i = 0$). Kamphius and Beekman [107] proposed a block-iterative version of (1.66) (only in the case $r_i = 0$). Erdoğan and Fessler propose a block-iterative version of the separable paraboloidal surrogates algorithm of Section 1.6.2 in [108, 109].

### *1.4.7 EM algorithms with nonseparable penalty functions*

All the algorithms described above were given for the ML case (where $\beta = 0$). What happens if we want to include a nonseparable penalty function for regularization, for example in the Convex-PS algorithm? Considering (1.33), it appears that we should replace (1.71) with

$$\underline{\mu}^{(n+1)} = \arg\max_{\underline{\mu} \geq \underline{0}} Q_3(\underline{\mu}; \underline{\mu}^{(n)}) - \beta R(\underline{\mu}). \tag{1.76}$$

Unfortunately, this is a nontrivial maximization since a nonseparable penalty $R(\underline{\mu})$ leads to coupled equations. One approach to circumventing this problem is the *generalized EM* (GEM) method [80, 110–112], in which one replaces the maximization in (1.76) with a few cycles of, for example, the coordinate ascent algorithm.

A clever alternative is to replace $R(\underline{\mu})$ in (1.76) with a separable surrogate function using a similar trick as in (1.60), which was proposed by De Pierro [73]. We discuss this approach in more detail in Section 1.6.2: see (1.92).

## 1.5 Coordinate-ascent algorithms

A simple and natural approach to finding the maximizer of $\Phi(\underline{\mu})$ is to sequentially maximize $\Phi(\underline{\mu})$ over each element $\mu_j$ of $\underline{\mu}$ using *the most recent values* for all other elements of $\underline{\mu}$. A general coordinate ascent method has the following form:

for $n = 0, 1, \ldots$ {
  `for` $j = 1, \ldots, N_p$ {

$$\mu_j^{(n+1)} := \arg\max_{\mu_j \geq 0} \Phi(\mu_1^{(n+1)}, \ldots, \mu_{j-1}^{(n+1)}, \mu_j, \mu_{j+1}^{(n)}, \ldots, \mu_{N_p}^{(n)}) \quad (1.77)$$

  }
}

The operation in (1.77) is performed "in place," i.e., the new value of $\mu_j$ replaces the old value, so that the most recent values of all elements of $\underline{\mu}$ are always used. An early use of such a method for tomography was in [113].

Sauer and Bouman analyzed such algorithms using clever frequency domain arguments [13], and showed that sequential algorithms yield iterates whose high frequency components converge fastest. This is often ideal for tomography, since we can use a low-resolution FBP image as the initial guess, and then iterate to improve resolution and reduce noise, which is mostly high frequency errors. (Using a uniform or zero initial image for coordinate ascent is a very poor choice since low frequencies can converge very slowly.)

The long string of arguments in (1.77) is quite notationally cumbersome. For the remainder of this section, we use

$$\underline{\tilde{\mu}} \triangleq [\mu_1^{(n+1)} \; \ldots \; \mu_{j-1}^{(n+1)} \; \mu_j^{(n)} \; \mu_{j+1}^{(n)} \; \ldots \; \mu_{N_p}^{(n)}] \quad (1.78)$$

as shorthand for the vector of the most recent parameter values. For simplicity, this notation leaves implicit the dependence of $\underline{\tilde{\mu}}$ on iteration $n$ and pixel index $j$.

The general method described by (1.77) is not exactly an "algorithm," since the procedure for performing the 1D maximization is yet unspecified. In practice it is impractical to find the exact maximizer, even in the 1D problem (1.77), so we settle for methods that increase $\Phi$.

### *1.5.1 Coordinate-ascent Newton-Raphson*

If $\Phi(\underline{\mu})$ were a quadratic functional, then the natural approach to performing the maximization in (1.77) would be Newton's method. Since $\Phi(\underline{\mu})$ in (1.15) is nonquadratic, applying Newton's method to (1.77) does not guarantee monotonic increases in $\Phi$, but one might still try it anyway and hope for the best. In practice monotonicity usually does not seem to be a problem, as suggested by the success of Bouman et al. with this approach [18, 114]. For such a *coordinate-ascent Newton-*

*Raphson* (CA-NR) algorithm, we replace (1.77) with the following update:

$$\mu_j^{(n+1)} = \left[ \mu_j^{(n)} + \frac{\frac{\partial}{\partial \mu_j} \Phi(\underline{\mu}) \Big|_{\underline{\mu}=\underline{\tilde{\mu}}}}{-\frac{\partial^2}{\partial \mu_j^2} \Phi(\underline{\mu}) \Big|_{\underline{\mu}=\underline{\tilde{\mu}}}} \right]_+ , \tag{1.79}$$

The $[\cdot]_+$ operation enforces the nonnegativity constraint. The first partial derivative is given by (1.21), and the second is given by:

$$\frac{\partial^2}{\partial \mu_j^2} \Phi(\underline{\mu}) = \sum_{i=1}^{N_Y} a_{ij}^2 \ddot{h}_i([\boldsymbol{A}\underline{\mu}]_i) - \beta \frac{\partial^2}{\partial \mu_j^2} R(\underline{\mu}) \,, \tag{1.80}$$

where $\ddot{h}_i$ is given by (1.14).

Specifically, using (1.13), (1.14), (1.21), and (1.80), the update (1.79) of the CA-NR algorithm becomes

$$\mu_j^{(n+1)} = \left[ \mu_j^{(n)} + \frac{\sum_{i=1}^{N_Y} a_{ij} \left( 1 - \frac{y_i}{b_i e^{-[\boldsymbol{A}\underline{\tilde{\mu}}]_i} + r_i} \right) b_i e^{-[\boldsymbol{A}\underline{\tilde{\mu}}]_i} - \beta \frac{\partial}{\partial \mu_j} R(\underline{\tilde{\mu}})}{\sum_{i=1}^{N_Y} a_{ij}^2 \left( 1 - \frac{y_i r_i}{(b_i e^{-[\boldsymbol{A}\underline{\tilde{\mu}}]_i} + r_i)^2} \right) b_i e^{-[\boldsymbol{A}\underline{\tilde{\mu}}]_i} + \beta \frac{\partial^2}{\partial \mu_j^2} R(\underline{\tilde{\mu}})} \right]_+ . \tag{1.81}$$

Literally interpreted, this form of the CA-NR algorithm appears to be extremely inefficient computationally, because it appears to require that $\boldsymbol{A}\underline{\tilde{\mu}}$ be recomputed after *every* pixel is updated sequentially. This would lead to $O(N_p^2)$ flops per iteration, which is impractical.

In the following *efficient* implementation of CA-NR, we maintain a copy of $\underline{\tilde{l}} \triangleq \boldsymbol{A}\underline{\tilde{\mu}}$ as a "state vector," and update that vector after each pixel is updated.

Initialization: $\underline{\tilde{l}} := \boldsymbol{A}\underline{\mu}^0$
`for` $n = 0, 1, \ldots$ {
`for` $j = 1, \ldots, N_p$ {

$$\mu_j^{(n+1)} := \left[ \mu_j^{(n)} + \frac{\sum_{i=1}^{N_Y} a_{ij} \left(1 - \frac{y_i}{b_i e^{-\tilde{l}_i} + r_i}\right) b_i e^{-\tilde{l}_i} - \beta \frac{\partial}{\partial \mu_j} R(\underline{\tilde{\mu}})}{\sum_{i=1}^{N_Y} a_{ij}^2 \left(1 - \frac{y_i r_i}{(b_i e^{-\tilde{l}_i} + r_i)^2}\right) b_i e^{-\tilde{l}_i} + \beta \frac{\partial^2}{\partial \mu_j^2} R(\underline{\tilde{\mu}})} \right]_+ \tag{1.82}$$

$$\tilde{l}_i := \tilde{l}_i + a_{ij}(\mu_j^{(n+1)} - \mu_j^{(n)}), \ \forall i \ : \ a_{ij} \neq 0 \tag{1.83}$$

}
}

The computational requirements per iteration are summarized in [77].

Bouman et al. also present a clever method to search for a zero-crossing to avoid using Newton-Raphson for the penalty part of the objective function [19].

The numerator in (1.82) is essentially a backprojection, and appears to be quite similar to the backprojection in the numerator of (1.68). One might guess then that coordinate ascent and an algorithm like Convex-NR in (1.68) would have similar computational requirements, but they do not. We can precompute the entire expression $\left(1 - \frac{y_i}{b_i e^{-l_i^{(n)}} + r_i}\right) b_i e^{-l_i^{(n)}}$ for each $i$ in the numerator of (1.68) *before* starting the backprojection, which saves many flops and nonsequential memory accesses. In contrast, the numerator of (1.82) contains $\tilde{l}_i$'s that *change after each pixel is updated*, so that expression cannot be precomputed. During the "backprojection" step, one must access four arrays (nonsequentially): the $y_i$'s, $b_i$'s, $r_i$'s, and $\tilde{l}_i$'s, in addition to the system matrix elements $a_{ij}$. And one must compute an exponentiation and a handful of addition and multiplications for *each nonzero* $a_{ij}$. For these reasons, coordinate ascent is quite expensive computationally per iteration. On the other hand, experience shows that if one considers the number of *iterations* required for "convergence," then CA-NR is among the best of all algorithms. The PSCA algorithm described in Section 1.6.4 below is an attempt to capture the convergence rate properties of CA-NR, but yet guaranteeing monotonicity and greatly reducing the flop counts per iteration.

An alternative approach to ensuring monotonicity would be to evaluate the objective function $\Phi$ after updating each pixel, and impose an interval search in the (hopefully relatively rare) cases where the objective function decreases. Unfortunately, evaluating $\Phi$ after every pixel adds considerable computational overhead.

### *1.5.2 Variation 1: Hybrid Poisson/polynomial approach*

One approach to reducing the flops required by (1.82) is to replace some of the nonquadratic $h_i$'s in the log-likelihood (1.11) with quadratic functions. Specifically, any given measured sinogram is likely to contain a mixture of high and low count rays. For high-count rays, a quadratic approximation to $h_i$ should be adequate, e.g. a Gaussian approximation to the Poisson statistics. For low count rays, the Poisson $h_i$ function (1.12) can be retained to avoid biases. This "hybrid Poisson/polynomial" approach was proposed in [14], and was shown to significantly reduce CPU time. However, implementation is somewhat inelegant since the system matrix $\boldsymbol{A}$ must be stored by sparse columns, and those sparse columns must be regrouped according to the indices of low and high count rays, which is a programming nuisance.

### *1.5.3 Variation 2: 1D parabolic surrogates*

Besides CPU time, another potential problem with (1.82) is that it is not guaranteed to monotonically increase $\Phi$, so divergence is possible. One can ensure monotonicity by applying the optimization transfer principle to the maximization problem (1.77). One possible approach is to use a parabolic surrogate for the 1D function $f(\mu_j) = \Phi(\mu_1^{(n+1)}, \ldots, \mu_{j-1}^{(n+1)}, \mu_j, \mu_{j+1}^{(n)}, \ldots, \mu_{N_p}^{(n)})$. For the fastest convergence rate, the optimal parabolic surrogate would have the lowest possible curvature, as discussed in Section 1.6.2 below. The surrogate parabola (1.27) with optimal curvature (1.29) can be applied to (1.77) to yield an algorithm of the form (1.82) but with a different expression in the denominator. Ignoring the penalty function, the ML coordinate ascent parabola surrogate (CA-PS) algorithm is

$$\boxed{\mu_j^{(n+1)} := \left[\mu_j^{(n)} + \frac{\sum_{i=1}^{N_Y} a_{ij}\left(1 - \frac{y_i}{b_i e^{-\tilde{l}_i} + r_i}\right) b_i e^{-\tilde{l}_i}}{\sum_{i=1}^{N_Y} a_{ij}^2\, c(\tilde{l}_i, y_i, b_i, r_i)}\right]_+ ,} \tag{1.84}$$

where $c(\cdot)$ was defined in (1.29). Unfortunately, this algorithm suffers from the same high CPU demands as (1.82), so is impractical. To incorporate a penalty function, one could follow a similar procedure as in Section 1.6.2 below.

Another approach to applying optimization transfer to (1.77) was proposed by Saquib et al. [115] and Zheng et al. [116], called "functional substitution." That method also yields a monotonic algorithm, for the case $r_i = 0$ since concavity of $h_i$ is exploited in the derivation. The required flops are comparable to those of CA-NR. We can generalize the functional substitution algorithm of [116], to the case $r_i \neq 0$ by exploiting the EM surrogate described in Section 1.4.5 to derive a new monotic algorithm. Essentially one simply replaces $y_i$ with $y_i b_i e^{-\tilde{l}_i}/(b_i e^{-\tilde{l}_i} + r_i)$ in the curvature terms in [116], yielding an algorithm that is identical to (1.82) but with a different denominator.

## 1.6 Paraboloidal surrogates algorithms

Coordinate ascent algorithms are *sequential update* algorithms because the pixels are updated in sequence. This leads to fast convergence, but requires column access of the system matrix $\boldsymbol{A}$, and makes parallelization quite difficult. In contrast, *simultaneous update* algorithms can update all pixels independently in parallel, such as the EM algorithms (1.45), (1.46), and (1.49), the scaled gradient ascent algorithms (1.50), (1.54), and (1.56), and the Convex algorithms (1.66), (1.68), and (1.72). However, a serious problem with *all* the simultaneous algorithms described above, except Convex-PS (1.72), is that they are not intrinsically monotonic. (They can all be forced to be monotonic by adding line searches, but this is somewhat inconvenient.) In this section we describe an approach based on the optimization transfer principle of Section 1.3.2 that leads to a simultaneous update algorithm that is also intrinsically monotonic, as well as a sequential algorithm that is intrinsically monotonic like CA-PS (1.84), but much more computationally efficient.

As mentioned in Section 1.3.4, a principal difficulty with maximizing (1.15) is the fact that the $h_i$'s in (1.12) are nonquadratic. Maximization is much easier for quadratic functions, so it is natural to use the surrogate parabola described in (1.27) to construct a *paraboloidal surrogate function* for the log-likelihood $L(\underline{\mu})$ in (1.11).

Using (1.29), define

$$c_i^{(n)} = c(l_i^{(n)}, y_i, b_i, r_i)\,,$$

where $l_i^{(n)} = [\boldsymbol{A}\underline{\mu}^{(n)}]_i$ was defined in (1.28). For this choice of curvatures, the parabola

$$q_i(l; l_i^{(n)}) = h_i(l_i^{(n)}) + \dot{h}_i(l_i^{(n)})(l - l_i^{(n)}) - \frac{1}{2}c_i^{(n)}(l - l_i^{(n)})^2$$

is a surrogate for $h_i(\cdot)$ in the sense that $h_i(l) \geq q_i(l; l_i^{(n)})$ for all $l \geq 0$. Summing these 1D surrogate functions, as in (1.11), leads to the following surrogate function for the log-likelihood:

$$Q_1(\underline{\mu}; \underline{\mu}^{(n)}) \triangleq \sum_{i=1}^{N_Y} q_i([\boldsymbol{A}\underline{\mu}]_i; [\boldsymbol{A}\underline{\mu}^{(n)}]_i)\,. \tag{1.85}$$

This is a surrogate for the log-likelihood in the sense that if we define

$$\phi_1(\underline{\mu}; \underline{\mu}^{(n)}) = Q_1(\underline{\mu}; \underline{\mu}^{(n)}) - \beta R(\underline{\mu})\,, \tag{1.86}$$

then $\phi_1$ satisfies (1.24), (1.25), and (1.26).

When expanded, the paraboloidal surrogate function $Q_1$ in (1.85) has the following quadratic form:

$$Q_1(\underline{\mu}; \underline{\mu}^{(n)}) \equiv \nabla L(\underline{\mu}^{(n)})(\underline{\mu} - \underline{\mu}^{(n)}) - \frac{1}{2}(\underline{\mu} - \underline{\mu}^{(n)})' \boldsymbol{A}' \operatorname{diag}\left\{c_i^{(n)}\right\} \boldsymbol{A}(\underline{\mu} - \underline{\mu}^{(n)})\,. \tag{1.87}$$

Maximizing a quadratic form like (1.87) is potentially much easier than the log-likelihood (1.11).

### *1.6.1 Paraboloidal surrogate with Newton Raphson*

For a quadratic penalty $R(\underline{\mu}) = \underline{\mu}'\boldsymbol{R}\underline{\mu}/2$, as in (1.18), the surrogate function $\phi_1$ in (1.86) above is a quadratic form. Disregarding the nonnegativity constraint, in principle we can maximize $\phi_1$ (as in (1.22)) by zeroing the gradient of $\phi_1$:

$$\nabla'\phi_1(\underline{\mu};\underline{\mu}^{(n)}) = \nabla'L(\underline{\mu}^{(n)}) - \beta\boldsymbol{R}\underline{\mu}^{(n)} - [\boldsymbol{A}'\,\mathrm{diag}\left\{c_i^{(n)}\right\}\boldsymbol{A} + \beta\boldsymbol{R}]^{-1}(\underline{\mu} - \underline{\mu}^{(n)})\,.$$

This leads to the following paraboloidal surrogates Newton-Raphson (PS-NR) algorithm:

$$\underline{\mu}^{(n+1)} = \underline{\mu}^{(n)} + \left[-\nabla^2\phi_1(\underline{\mu};\underline{\mu}^{(n)})\Big|_{\underline{\mu}=\underline{\mu}^{(n)}}\right]^{-1} \nabla'_{\underline{\mu}}\phi_1(\underline{\mu};\underline{\mu}^{(n)})\Big|_{\underline{\mu}=\underline{\mu}^{(n)}}\,,$$

$$\boxed{\underline{\mu}^{(n+1)} = \underline{\mu}^{(n)} + [\boldsymbol{A}'\,\mathrm{diag}\left\{c_i^{(n)}\right\}\boldsymbol{A} + \beta\boldsymbol{R}]^{-1}[\nabla'L(\underline{\mu}^{(n)}) - \beta\boldsymbol{R}\underline{\mu}^{(n)}]\,.} \tag{1.88}$$

There are three problems with this algorithm. The matrix inverse is impractical, the method appears only to apply to quadratic penalty functions, and nonnegativity is not enforced. Fortunately all three of these limitations can be overcome, as we describe next.

### *1.6.2 Separable paraboloidal surrogates algorithm*

In this section we derive an intrinsically monotonic algorithm that we believe to be the current "method of choice" for cases where one desired a *simultaneous* update without any line searches.

A difficulty in maximizing (1.86) is that in general both $Q_1$ and $R$ are *nonseparable* functions of the elements of the parameter vector $\underline{\mu}$. However, since each $q_i$ in (1.85) is concave, we can apply precisely the same convexity trick of De Pierro used in (1.59) and (1.60) to form a second *separable* surrogate function. Since

$$\begin{aligned} q_i([\boldsymbol{A}\underline{\mu}]_i; l_i^{(n)}) &= q_i\left(\sum_{j=1}^{N_p}\alpha_{ij}\left[\frac{a_{ij}}{\alpha_{ij}}(\mu_j - \mu_j^{(n)}) + l_i^{(n)}\right]; l_i^{(n)}\right) \\ &\geq \sum_{j=1}^{N_p}\alpha_{ij}q_i\left(\frac{a_{ij}}{\alpha_{ij}}(\mu_j - \mu_j^{(n)}) + l_i^{(n)}; l_i^{(n)}\right)\,, \end{aligned}$$

the natural separable surrogate function for the log-likelihood is

$$L(\underline{\mu}) \geq Q_2(\underline{\mu};\underline{\mu}^{(n)}) \stackrel{\triangle}{=} \sum_{j=1}^{N_p} Q_{2,j}(\mu_j;\underline{\mu}^{(n)}) \tag{1.89}$$

where

$$Q_{2,j}(\mu_j;\underline{\mu}^{(n)}) \triangleq \sum_{i=1}^{N_Y} \alpha_{ij} q_i \left( \frac{a_{ij}}{\alpha_{ij}} (\mu_j - \mu_j^{(n)}) + l_i^{(n)}; l_i^{(n)} \right) . \tag{1.90}$$

Since $Q_2$ is a *separable function*, it is easily maximized.

We need to apply a similar trick to separate the penalty function. We assume that $R$ has the form (1.18). Similar to (1.59) we have

$$[\boldsymbol{C}\underline{\mu}]_k = \sum_{j=1}^{N_p} \gamma_{kj} \left[ \frac{c_{kj}}{\gamma_{kj}} (\mu_j - \mu_j^{(n)}) + [\boldsymbol{C}\underline{\mu}^{(n)}]_k \right] , \tag{1.91}$$

where $\gamma_{kj} \geq 0$ and $\sum_{j=1}^{N_p} \gamma_{kj} = 1$. So since $t^2/2$ is a convex function:

$$\begin{aligned} R(\underline{\mu}) &= \sum_{k=1}^{K} \omega_k \frac{1}{2} \left( [\boldsymbol{C}\underline{\mu}]_k \right)^2 \\ &= \sum_{k=1}^{K} \omega_k \frac{1}{2} \left( \sum_{j=1}^{N_p} \gamma_{kj} \left[ \frac{c_{kj}}{\gamma_{kj}} (\mu_j - \mu_j^{(n)}) + [\boldsymbol{C}\underline{\mu}^{(n)}]_k \right] \right)^2 \\ &\geq \sum_{k=1}^{K} \sum_{j=1}^{N_p} \gamma_{kj} \omega_k \frac{1}{2} \left( \frac{c_{kj}}{\gamma_{kj}} (\mu_j - \mu_j^{(n)}) + [\boldsymbol{C}\underline{\mu}^{(n)}]_k \right)^2 , \end{aligned} \tag{1.92}$$

so the natural surrogate function is

$$R(\underline{\mu};\underline{\mu}^{(n)}) \triangleq \sum_{j=1}^{N_p} R_j(\mu_j;\underline{\mu}^{(n)}) ,$$

where

$$R_j(\mu_j;\underline{\mu}^{(n)}) \triangleq \sum_{k=1}^{K} \gamma_{kj} \omega_k \frac{1}{2} \left( \frac{c_{kj}}{\gamma_{kj}} (\mu_j - \mu_j^{(n)}) + [\boldsymbol{C}\underline{\mu}^{(n)}]_k \right)^2 .$$

Combining $Q_2$ and $R(\cdot;\underline{\mu}^{(n)})$ yields the following separable quadratic surrogate function:

$$\phi_2(\underline{\mu};\underline{\mu}^{(n)}) \triangleq Q_2(\underline{\mu};\underline{\mu}^{(n)}) - \beta R(\underline{\mu};\underline{\mu}^{(n)}) = \sum_{j=1}^{N_p} \phi_{2,j}(\mu_j;\underline{\mu}^{(n)})$$

where

$$\phi_{2,j}(\mu_j;\underline{\mu}^{(n)}) \triangleq Q_{2,j}(\mu_j;\underline{\mu}^{(n)}) - \beta R_j(\mu_j;\underline{\mu}^{(n)}) .$$

Using the choice (1.67) for the $\alpha_{ij}$'s, the surrogate curvature is

$$\frac{d^2}{d\mu_j^2} Q_{2,j}(\mu_j; \underline{\mu}^{(n)}) = \sum_{i=1}^{N_Y} a_{ij}^2 / \alpha_{ij} c_i^{(n)} = \sum_{i=1}^{N_Y} a_{ij} a_i \, c_i^{(n)} .$$

Similarly, if we choose $\gamma_{kj} = |c_{kj}|/c_k$ , where $c_k \triangleq \sum_{j=1}^{N_p} |c_{kj}|$, then

$$r_j \triangleq \frac{d^2}{d\mu_j^2} R_j(\mu_j; \underline{\mu}^{(n)}) = \sum_{k=1}^{K} c_{kj}^2 \omega_k / \gamma_{kj} = \sum_{k=1}^{K} |c_{kj}| c_k \, \omega_k . \tag{1.93}$$

Since the surrogate is separable, the optimization transfer algorithm (1.22) becomes

$$\mu_j^{(n+1)} = \arg\max_{\mu_j \geq 0} \phi_{2,j}(\mu_j; \underline{\mu}^{(n)}), \; j = 1, \ldots, N_p .$$

Since $\phi_{2,j}(\mu_j; \underline{\mu}^{(n)})$ is quadratic, it is easily maximized by zeroing its derivative, leading to the following *separable paraboloidal surrogates* (SPS) algorithm:

$$\boxed{\mu_j^{(n+1)} = \left[ \mu_j^{(n)} + \frac{\sum_{i=1}^{N_Y} a_{ij} \dot{h}_i^{(n)} - \beta \sum_{k=1}^{K} c_{kj} \omega_k [\boldsymbol{C} \underline{\mu}^{(n)}]_k}{\sum_{i=1}^{N_Y} a_{ij} a_i \, c_i^{(n)} + \beta r_j} \right]_+ ,} \tag{1.94}$$

where $\dot{h}_i^{(n)}$ was defined in (1.73) and we precompute the $r_j$'s in (1.93) before iterating. This algorithm is highly parallelizable, and can be implemented efficiently using the same structure as the Convex-PS algorithm (1.74). It is also easy to form an ordered subsets version (cf. Section 1.4.6) of the SPS algorithm [109].

See [109, 117] for the extension to nonquadratic penalty functions, which is based on a parabola surrogate for $\psi_k$ proposed by Huber [118].

### *1.6.3 Ordered subsets revisited*

One can easily form an ordered subsets version (cf. Section 1.4.6) of the SPS algorithm (1.94) by replacing the sums over $i$ with sums over subsets of the rays, yielding the ordered subsets transmission (OSTR) algorithm described in [109]. Since ordered subsets algorithms are not guaranteed to converge, one may as well further abandon monotonicity and replace the denominator in the ordered subsets version of (1.94) with something that can be precomputed. Specifically, in [109] we recommend replacing the $c_i^{(n)}$'s in (1.94) with[11]

$$c_i = -\ddot{h}_i(\hat{l}_i) = \begin{cases} \frac{(y_i - r_i)^2}{y_i}, & y_i > 0 \\ 0, & y_i = 0 , \end{cases} \tag{1.95}$$

[11]This trick is somewhat similar in spirit to the method of Fisher scoring [119, 120], in which one replaces the Hessian with its expectation (the Fisher information matrix) to reduce computation in nonquadratic optimization problems.

where $\hat{l}_i \triangleq \log \frac{b_i}{y_i - r_i}$. For this "fast denominator" approximation, the OSTR-$M$ algorithm becomes:

$$\boxed{\mu_j^{\text{new}} = \left[\mu_j + \frac{M \sum_{i \in \mathcal{S}} a_{ij} \dot{h}_i^{(n)} - \beta \sum_{k=1}^K c_{kj} \omega_k [\boldsymbol{C} \underline{\mu}^{(n)}]_k}{d_j + w_j}\right]_+ ,} \tag{1.96}$$

where $\mathcal{S}$ is a cyclically chosen subset of the rays, formed by angular subsampling by a factor $M$, where $\dot{h}_i^{(n)}$ was defined in (1.73), and where we precompute

$$d_j \triangleq \sum_{i=1}^{N_Y} a_{ij} a_i \, c_i, \; j = 1, \ldots, N_p \, .$$

The results in [109] show that this algorithm does not quite find the maximizer of the objective function $\Phi$, but the images are nearly as good as those produced by convergent algorithms in terms of mean squared error and segmentation accuracy.

### *1.6.4 Paraboloidal surrogates coordinate-ascent (PSCA) algorithm*

A disadvantage of simultaneous updates like (1.94) is that they typically converge slowly since separable surrogate functions have high curvature and hence slow convergence rates (cf. Section 1.3.3). Thus, in [77, 121] we proposed to apply coordinate ascent to the quadratic surrogate function (1.86). (We focus on the quadratic penalty case here; the extension to the nonquadratic case is straightforward following a similar approach as in Section 1.6.2.) To apply CA to (1.86), we sequentially maximize $\phi_1(\underline{\mu}; \underline{\mu}^{(n)})$ over each element $\mu_j$, using the most recent values for all other elements of $\underline{\mu}$, as in Section 1.5. We again adopt the shorthand (1.78) here. In its simplest form, this leads to a *paraboloidal surrogates coordinate ascent* (PSCA) algorithm having a similar form as (1.82), but with the inner update being:

$$\boxed{\mu_j^{(n+1)} := \left[\mu_j^{(n)} + \frac{\sum_{i=1}^{N_Y} a_{ij} \dot{q}_i^{(n)} - \beta \frac{\partial}{\partial \mu_j} R(\underline{\tilde{\mu}})}{d_j + \beta \frac{\partial^2}{\partial \mu_j^2} R(\underline{\tilde{\mu}})}\right]_+ ,} \tag{1.97}$$

where, before looping over $j$ in each iteration, we precompute

$$d_j = \sum_{i=1}^{N_Y} a_{ij}^2 c_i^{(n)}, \; j = 1, \ldots, N_p \, , \tag{1.98}$$

and the following term is maintained as a "state vector" (analogous to the $\tilde{l}_i$'s in (1.83):

$$\dot{q}_i^{(n)} \triangleq \dot{h}_i(l_i^{(n)}) = \left(1 - \frac{y_i}{b_i e^{-l_i^{(n)}} + r_i}\right) b_i e^{-l_i^{(n)}} \, .$$

This precomputation saves many flops per iteration, yet still yields an intrinsically monotonic algorithm. Even greater computational savings are possibly by a "fast denominator" trick similar to (1.95), although one should then check for monotonicity after each iteration and redo the iteration using the monotonicity preserving denominators (1.98) in those rare cases where the objective function decreases. There are several "details" that are essential for efficient implementation; see [77].

### *1.6.5 Grouped coordinate ascent algorithm*

We have described algorithms that update a single pixel at a time, as in the PSCA algorithm (1.97) above, or update all pixels simultaneously, as in the SPS algorithm (1.94) above. A problem with the sequential algorithms is that they are difficult to parallelize, whereas a problem with simultaneous algorithms is their slow convergence rates. An alternative is to update a *group* of pixels simultaneously. If the pixels are well separated spatially, then they may be approximately uncorrelated[12], which leads to separable surrogate functions that have lower curvature [101, 122, 123]. We call such methods *grouped coordinate ascent* (GCA) algorithms. The statistics literature has work on GCA algorithms e.g. [124], which in turn cites related algorithms dating to 1964! In tomography, one can apply the GCA idea directly to the log-likelihood (1.11) [101,117,123], or to the paraboloidal surrogate (1.85) [77].

## 1.7 Direct algorithms

The algorithms described above have all been developed, to some degree, by considering the specific form of the log-likelihood (1.11). It is reasonable to hypothesize that algorithms that are "tailor made" for the form of the objective function (1.15) in tomography should outperform (converge faster) general purpose optimization methods that usually treat the objective function as a "black box" in the interest of greatest generality. Nevertheless, general purpose optimization is a very active research area, and it behooves developers of image reconstruction algorithms to keep abreast of progress in that field. General purpose algorithms that are natural candidates for image reconstruction include the conjugate gradient algorithm and the quasi-Newton algorithm, described next.

### *1.7.1 Conjugate gradient algorithm*

For unconstrained quadratic optimization problems, the preconditioned conjugate gradient (CG) algorithm [125] is particularly appealing because it converges rapidly[13] for suitably chosen preconditioners, e.g. [67]. For nonquadratic objective

[12] To be more precise: the submatrix of the Hessian matrix of $\Phi$ corresponding to a subset of spatially separated pixels is approximately diagonal.

[13] It is often noted that CG converges in $N_p$ iterations in exact arithmetic, but this fact is essentially irrelevant in tomography since $N_p$ is so large. More relevant is the fact that the convergence rate of CG is quite good with suitable preconditioners.

functions, or when constraints such as nonnegativity are desired, the CG method is somewhat less convenient due to the need to perform line searches. It may be possible to adopt the optimization transfer principles to simplify the line searches, cf. [67].

Mumcuoğlu et al. [31, 126] have been particularly successful in applying diagonally preconditioned conjugate gradients to both transmission and emission tomography. Their diagonal preconditioner was based on (1.52). They investigated both a penalty function approach to encourage nonnegativity [31, 126], as well as active set methods [127] for determining the set of nonzero pixels [128, 129].

An alternative approach to enforcing nonnegativity in gradient-based methods uses adaptive barriers [130].

### *1.7.2 Quasi-Newton algorithm*

The ideal preconditioner for the conjugate gradient algorithm would be the inverse of the Hessian matrix, which would lead to superlinear convergence [131]. Unfortunately, in tomography the Hessian matrix is a large non-sparse matrix, so its inverse is impractical to compute and store. The basic idea of the quasi-Newton family of algorithms is to form low-rank approximations to the inverse of the Hessian matrix as the iterations proceed [85]. This approach has been applied by Kaplan et al. [132] to simultaneous estimation of SPECT attenuation and emission distributions, using the public domain software for limited memory, bound-constrained minimization (L-BFGS-B) [133]. Preconditioning has been found to accelerate such algorithms [132].

## 1.8 Alternatives to Poisson models

Some of the algorithms described above are fairly complex, and this complexity derives from the nonconvex, nonquadratic form of the transmission Poisson log-likelihood (1.11) and (1.12). It is natural then to ask whether there are simpler approaches that would give adequate results in practice. Every simpler approach that we are aware of begins by using the logarithmic transformation (1.3), which compensates for the nonlinearity of Beer's law (1.2) and leads then to a linear problem

$$\hat{l}_i \approx [\boldsymbol{A}\underline{\mu}]_i, \; i = 1, \ldots, N_Y \,. \tag{1.99}$$

Unfortunately, for low-count transmission scans, especially those contaminated by background events of any type ($r_i \neq 0$), the logarithm (1.3) simply cannot be used since $Y_i - r_i$ can be nonpositive for many rays. In medium to high-count transmission scans, the bias described in (1.4) should be small, so one could work with the estimated line integrals (the $\hat{l}_i$'s) rather than the raw transmission measurements (the $Y_i$'s).

### 1.8.1 Algebraic reconstruction methods

A simple approach to estimating $\underline{\mu}$ is to treat (1.99) as a set of $N_Y$ equations in $N_p$ unknowns and try to "solve" for $\underline{\mu}$. This was the motivation for the algebraic reconstruction technique (ART) family of algorithms [11]. For noisy measurements the equations (1.99) are usually inconsistent, and ART "converges" to a limit cycle for inconsistent problems. One can force ART to converge by introducing appropriate strong underrelaxation [134]. However, the limit is the minimum-norm weighted least-squares solution for a particular norm that is unrelated to the measurement statistics. The Gauss-Markov theorem [90] states that estimator variance is minimized when the least-squares norm is chosen to be the inverse of the covariance matrix, so it seems preferable to approach (1.99) by first finding a statistically-motivated cost function, and then finding algorithms that minimize that cost function, rather than trying to "fix up" algorithms that were derived under the unrealistic assumption that (1.99) is a consistent system of equations.

### 1.8.2 Methods to avoid

A surprising number of investigators have applied the *emission* EM algorithm to "solve" (1.99), even though the statistics of $\hat{l}_i$ are *entirely different* from those of emission sinogram measurements, e.g. [15]. We strongly recommend avoiding this practice. Empirical results with simulated and phantom data show that this approach is inferior to methods such as OSTR which are based on the transmission statistical model (1.7).

Liang and Ye [135] present the following iteration for MAP reconstruction of attenuation maps without giving any derivation:

$$\mu_j^{(n+1)} = \mu_j \frac{\sum_{i=1}^{N_Y} a_{ij}}{\sum_{i=1}^{N_Y} a_{ij} [\boldsymbol{A}\underline{\mu}^{(n)}]_i / \log(b_i/y_i)} .$$

The iteration looks like an "upside down" emission EM algorithm. The convergence properties of this algorithm are unknown.

Zeng and Gullberg [136] proposed the following steepest ascent method with a fixed step-length parameter:

$$\mu_j^{(n+1)} = \mu_j^{(n)} + \alpha \left[ \sum_{i=1}^{N_Y} a_{ij} (y_i/b_i - e^{-l_i^{(n)}}) + \beta \frac{\partial}{\partial \mu_j} R(\underline{\mu}^{(n)}) \right] ,$$

for an interesting choice of penalty $R(\underline{\mu})$ that encourages attenuation values near those of air/lung, soft tissue, or bone. Without a line search of the type studied by Lange [95, 97], monotonicity is not guaranteed. Even with a line search it is unlikely that this algorithm maximizes the objective function since its fixed points are not stationary points of $\Phi$.

### 1.8.3 Weighted least-squares methods

Rather than simply treating (1.99) as a system of equations, we can use (1.99) as the rationale for a weighted least-squares cost function. There are several choices for the weights.

#### 1.8.3.1 Model-weighted LS

By a standard propagation-of-errors argument, one can show from (1.3) and (1.7) that

$$\mathrm{Var}\left\{\hat{l}_i\right\} \approx \frac{\bar{y}_i}{(\bar{y}_i - r_i)^2}\,, \tag{1.100}$$

where $\bar{y}_i$ was defined in (1.8). A natural "model-weighted" least-squares cost function is then

$$\Phi(\underline{\mu}) = \sum_{i=1}^{N_Y} (\hat{l}_i - [\boldsymbol{A}\underline{\mu}]_i)^2 \frac{(\bar{y}_i(\underline{\mu}) - r_i)^2}{\bar{y}_i(\underline{\mu})}\,. \tag{1.101}$$

This type of cost function has been considered in [137]. Unfortunately, the above cost function is nonquadratic, so finding its minimizer is virtually as difficult as maximizing (1.15).

#### 1.8.3.2 Data-weighted LS

A computationally simpler approach arises if we replace the estimate-dependent variance (1.100) with a data-based estimate by substituting the data $Y_i$ for $\bar{y}_i$. This leads naturally to the following *data-weighted least-squares* cost function:

$$\Phi(\underline{\mu}) = \sum_{i=1}^{N_Y} (\hat{l}_i - [\boldsymbol{A}\underline{\mu}]_i)^2 w_i\,, \tag{1.102}$$

where $w_i = \frac{(Y_i - r_i)^2}{Y_i}$ is a precomputed weight. Minimizing $\Phi$ is straightforward since it is quadratic, so one can apply, for example, conjugate gradient algorithms or coordinate descent algorithms [13]. This approach gives more weight to those measurements that have lower variance, and less weight to the noiser measurements. This type of weighting can significantly reduce the noise in the reconstructed image relative to unweighted least squares. In fact, unweighted least squares estimates are essentially equivalent to FBP images. (The shift-invariant FBP methods treat all data equally, since noise is ignored.) However, as mentioned below (1.4), data-weighting leads to a systematic negative bias that increases as counts decrease [14, 15]. So (1.102) is only appropriate for moderate to high SNR problems.

One can also derive (1.102) by making a second-order Taylor expansion of the log-likelihood (1.12) about $\hat{l}_i$ [13, 14, 138, 139].

#### 1.8.3.3 *Reweighted LS*

The two cost functions given above represent two extremes. In (1.102), the weights are fixed once-and-for-all prior to minimization, whereas in (1.101), the weights vary continuously as the estimate of $\underline{\mu}$ changes. A practical alternative is to first run any inexpensive algorithm (such as OSTR) for a few iterations and then reproject the estimated image $\underline{\mu}^{(n)}$ to form estimated line integrals $l_i^{(n)} = [\boldsymbol{A}\underline{\mu}^{(n)}]_i$. Then perform a second-order Taylor expansion of the log-likelihood (1.12) around $l_i^{(n)}$ to find a quadratic approximation that can be minimized easily. This approach should avoid the biases of data-weighted least-squares, and if iterated is known as *reweighted least squares* [140, 141].

## 1.9 Emission reconstruction

In emission tomography, the goal is to reconstruct an emission distribution $\lambda(\vec{x})$ from recorded counts of emitted photons. We again parameterize the emission distribution analogous to (1.5), letting $\lambda_j$ denote the mean number of emissions from the $j$th voxel. The goal is to estimate $\underline{\lambda} = [\lambda_1, \ldots, \lambda_{N_p}]'$ from projection measurements $\underline{Y} = [Y_1, \ldots, Y_{N_Y}]'$. The usual Poisson measurement model is identical to (1.7), except that the measurement means are given by

$$\bar{y}_i = \sum_{j=1}^{N_p} a_{ij}\lambda_j + r_i = [\boldsymbol{A}\underline{\lambda}]_i + r_i\,, \tag{1.103}$$

where $a_{ij}$ represents the probability that an emission from the $j$th voxel is recorded by the $i$th detector, and $r_i$ again denotes additive background counts such as random coincidences and scatter. (Accurate models for the $a_{ij}$'s can lead to significant improvements in image spatial resolution and accuracy, e.g. [142, 143]. The log-likelihood has a similar form to (1.11):

$$L(\underline{\lambda}) = \sum_{i=1}^{N_Y} h_i([\boldsymbol{A}\underline{\lambda}]_i)\,,$$

where

$$h_i(l) = y_i \log(l + r_i) - (l + r_i)\,. \tag{1.104}$$

This $h_i$ function is concave for $l \in (-r_i, \infty)$, and is strictly concave if $y_i > 0$. Since $\bar{y}_i$ is linearly related to the $\lambda_j$'s (in contrast to the nonlinear relationship in (1.8) in the transmission case), the emission reconstruction problem is considerably easier than the transmission problem. Many of the algorithms described above apply to the emission problem, as well as to other inverse problems having log-likelihood functions of the general form (1.11). We describe in this section a few algorithms for maximizing the emission log-likelihood $L(\underline{\lambda})$. Extensions to the regularized problem are similar to those described for the transmission case.

### *1.9.1 EM Algorithm*

One can derive the classical EM algorithm for the emission problem by a formal complete-data exposition [17], which is less complicated than the transmission case but still somewhat mysterious to many readers, or by fixed-point considerations [79] (which do not fully illustrate the monotonicity of the emission EM algorithm). Instead, we adopt the simple concavity-based derivation of De Pierro [72], which reinforces the surrogate function concepts woven throughout this chapter.

The key to the derivation is the following "multiplicative" trick, which applies if $\lambda_j^{(n)} > 0$:

$$[\boldsymbol{A}\underline{\lambda}]_i + r_i = \sum_{j=1}^{N_p} \left( \frac{a_{ij}\lambda_j^{(n)}}{\bar{y}_i^{(n)}} \right) \frac{\lambda_j}{\lambda_j^{(n)}} \bar{y}_i^{(n)} + \left( \frac{r_i}{\bar{y}_i^{(n)}} \right) \bar{y}_i^{(n)} . \tag{1.105}$$

The $N_p + 1$ terms in parentheses are nonnegative and sum to unity, so we can apply the concavity inequality. Since $g_i(l) \stackrel{\triangle}{=} y_i \log l - l$ is concave on $(0, \infty)$, it follows that

$$\begin{aligned} L(\underline{\lambda}) &= \sum_{i=1}^{N_Y} g_i([\boldsymbol{A}\underline{\lambda}]_i + r_i) \\ &= \sum_{i=1}^{N_Y} g_i \left( \sum_{j=1}^{N_p} \left( \frac{a_{ij}\lambda_j^{(n)}}{\bar{y}_i^{(n)}} \right) \frac{\lambda_j}{\lambda_j^{(n)}} \bar{y}_i^{(n)} + \left( \frac{r_i}{\bar{y}_i^{(n)}} \right) \bar{y}_i^{(n)} \right) \\ &\geq \sum_{i=1}^{N_Y} \sum_{j=1}^{N_p} \left( \frac{a_{ij}\lambda_j^{(n)}}{\bar{y}_i^{(n)}} \right) g_i \left( \frac{\lambda_j}{\lambda_j^{(n)}} \bar{y}_i^{(n)} \right) + \left( \frac{r_i}{\bar{y}_i^{(n)}} \right) g_i(\bar{y}_i^{(n)}) \stackrel{\triangle}{=} Q(\underline{\lambda}; \underline{\lambda}^{(n)}) . \end{aligned}$$

The surrogate function $Q$ is separable:

$$Q(\underline{\lambda}; \underline{\lambda}^{(n)}) = \sum_{j=1}^{N_p} Q_j(\lambda_j; \underline{\lambda}^{(n)}), \quad Q_j(\lambda_j; \underline{\lambda}^{(n)}) \equiv \sum_{i=1}^{N_Y} \frac{a_{ij}\lambda_j^{(n)}}{\bar{y}_i^{(n)}} g_i \left( \frac{\lambda_j}{\lambda_j^{(n)}} \bar{y}_i^{(n)} \right) . \tag{1.106}$$

Thus the the following parallelizable maximization step is guaranteed to monotonically increase the log-likelihood $L(\underline{\lambda})$ each iteration:

$$\lambda_j^{(n+1)} = \arg\max_{\lambda_j \geq 0} Q_j(\lambda_j; \underline{\lambda}^{(n)}) . \tag{1.107}$$

The maximization is trivial:

$$\frac{\partial}{\partial \lambda_j} Q_j(\lambda_j; \underline{\lambda}^{(n)}) = \sum_{i=1}^{N_Y} a_{ij} \dot{g}_i \left( \frac{\lambda_j}{\lambda_j^{(n)}} \bar{y}_i^{(n)} \right) = \sum_{i=1}^{N_Y} a_{ij} \left[ \frac{\lambda_j^{(n)}}{\lambda_j} \frac{y_i}{\bar{y}_i^{(n)}} - 1 \right] .$$

Equating to zero and solving for $\lambda_j$ yields the famous update:

$$\boxed{\lambda_j^{(n+1)} = \lambda_j^{(n)} \frac{\sum_{i=1}^{N_Y} a_{ij} y_i / \bar{y}_i^{(n)}}{\sum_{i=1}^{N_Y} a_{ij}}, \; j = 1, \ldots, N_p \,.} \tag{1.108}$$

Unfortunately, the emission EM algorithm (1.108) usually converges painfully slowly. To understand this, consider the curvatures of the surrogate functions $Q_j$:

$$-\frac{\partial^2}{\partial \lambda_j^2} Q_j(\lambda_j; \underline{\lambda}^{(n)}) = \frac{\lambda_j^{(n)}}{\lambda_j^2} \sum_{i=1}^{N_Y} a_{ij} \frac{y_i}{\bar{y}_i^{(n)}} \,. \tag{1.109}$$

For any pixels converging towards zero, these curvatures grow without bound. This leads to very slow convergence; even sublinear convergence rates are possible [76].

### *1.9.2 An improved EM algorithm*

One can choose a slightly better decomposition than (1.105) to get slightly faster converging EM algorithms [75]. First find any set of nonnegative constants $\{m_j\}_{j=1}^{N_p}$ that satisfy

$$r_i \geq \sum_{j=1}^{N_p} a_{ij} m_j, \forall i \,. \tag{1.110}$$

Then an alternative to (1.105) is:

$$[\boldsymbol{A}\underline{\lambda}]_i + r_i = \sum_{j=1}^{N_p} \left( \frac{a_{ij}(\lambda_j^{(n)} + m_j)}{\bar{y}_i^{(n)}} \right) \frac{\lambda_j}{\lambda_j^{(n)} + m_j} \bar{y}_i^{(n)} + \left( \frac{\hat{r}_i}{\bar{y}_i^{(n)}} \right) \bar{y}_i^{(n)} \,, \tag{1.111}$$

where $\hat{r}_i = r_i - \sum_{j=1}^{N_p} a_{ij} m_j \geq 0$. Again the terms in parentheses in (1.111) are nonnegative and sum to unity. So a similar derivation as that yielding (1.106) leads to a new surrogate function:

$$Q_j(\lambda_j; \underline{\lambda}^{(n)}) = \sum_{i=1}^{N_Y} \frac{a_{ij}(\lambda_j^{(n)} + m_j)}{\bar{y}_i^{(n)}} g_i \left( \frac{\lambda_j + m_j}{\lambda_j^{(n)}} \bar{y}_i^{(n)} \right) .$$

Maximizing as in (1.107) leads to the following algorithm

$$\boxed{\lambda_j^{(n+1)} = \left[ (\lambda_j^{(n)} + m_j) \frac{\sum_{i=1}^{N_Y} a_{ij} y_i / \bar{y}_i^{(n)}}{\sum_{i=1}^{N_Y} a_{ij}} - m_j \right]_+ , \; j = 1, \ldots, N_p \,.} \tag{1.112}$$

This algorithm was derived by a more complicated EM approach in [75], and called ML-EM-3. The surrogate function derivation is simpler to present and understand, and more readily generalizable to alternative surrogates.

The curvatures of the second $Q_j$'s just derived are smaller than those in (1.106), due to the $m_j$'s (replace $\lambda_j$ with $\lambda_j + m_j$ in the denominator of (1.109)). The convergence rate improves as the $m_j$'s increase, but of course (1.110) must be satisfied to ensure monotonicity. Since the EM algorithm updates all parameters simultaneously, the $m_j$ values must be "shared" among all pixels, and typically are fairly small due to (1.110). In contrast the SAGE algorithm [75] updates the pixels sequentially, which greatly relaxes the constraints on the $m_j$'s, allowing larger values and hence faster convergence rates.

### *1.9.3 Other emission algorithms*

Most of the other methods for developing reconstruction algorithms described in this chapter have counterparts for the emission problem. Monotonic acceleration is possible using line searches [94]. Replacing the sums over $i$ in (1.108) with sums over subsets of the projections yields the emission OSEM algorithm [102]; see also the related variants RAMLA [144] and RBBI [103, 104]. Although the OSEM algorithm fails to converge in general, it often gives reasonable looking images in a small number of iterations when initialized with a uniform image. Sequential updates rather than parallel updates leads to the fast converging SAGE algorithms [75] and coordinate ascent algorithms [19], including paraboloidal surrogate variations thereof [78]. The conjugate gradient algorithm has been applied extensively to the emission problem and is particularly effective provided one carefully treats the nonnegativity constraints [31].

## 1.10 Advanced topics

In this section we provide pointers to the literature for several additional topics, all of which are active research areas.

### *1.10.1 Choice of regularization parameters*

A common critique of penalized-likelihood and Bayesian methods is that one must choose (subjectively?) the regularization parameter $\beta$ in (1.15). (In unregularized methods there are also free parameters that one must select, such as the number of iterations or the amount of post-filtering, but fiddling these factors to get visually pleasing images is perhaps easier than adjusting $\beta$, since each new $\beta$ requires another run of the iterative algorithm.) A large variety of methods for automatically choosing $\beta$ have been proposed, based on principles such as maximum likelihood or *cross validation* e.g., [145–154], most of which have been evaluated in terms of mean-squared error performance, which equally weights bias (squared) and variance, even though resolution and noise may have unequal importance in imaging problems.

For quadratic regularization methods, one can choose both $\beta$ and $R(\underline{\mu})$ to control the spatial resolution properties and to relate the desired spatial resolution to an appropriate value of $\beta$ by a predetermined table [20, 54–57].

In addition to the variety of methods for choosing $\beta$, there is an even larger variety of possible choices for the "potential functions" $\psi_k$ in (1.17), ranging from quadratic to nonquadratic to nonconvex and even nondifferentiable. See [155] for a recent discussion.

The absolute value potential ($\psi_k(t) = |t|$) is particularly appealing in problems with piecewise constant attenuation maps. However, its nondifferentiability greatly complicates optimization [156–158].

### 1.10.2 Source-free attenuation reconstruction

In PET and SPECT imaging, the attenuation map is a nuisance parameter; the emission distribution is of greatest interest. This has spawned several attempts to estimate the attenuation map from the emission sinograms, without a separate transmission scan. See e.g., [132, 159, 160].

### 1.10.3 Dual energy imaging

We have focused on the case of monoenergetic imaging, by the assumption (1.23). For quantitative applications such as bone densitometry, one must account for the polyenergetic property of x-ray source spectra. A variety of methods have been proposed for dual energy image reconstruction, including (recently) statistical methods [161–163].

### 1.10.4 Overlapping beams

Some transmission scan geometries involve multiple transmission sources, and it is possible for a given detector element to record photons that originated in more than one of these sources, i.e., the beams of photons emitted from the various sources overlap on the detector. The transmission statistical model (1.8) must be generalized to account for such overlap, leading to new reconstruction algorithms [164–167].

### 1.10.5 Sinogram truncation and limited angles

In certain geometries, portions of the sinogram are missing due to geometric truncation (such as fan-beam geometries with a short focal length). In such cases, prior information plays an essential role in regularizing the reconstruction problem, e.g., [43]. Similarly, in *limited angle tomography* the sinograms are truncated due to missing angles. Nonquadratic regularization methods have shown considerable promise for such problems [68].

### 1.10.6 Parametric object priors

Throughout this chapter we have considered the image to be parameterized by the linear series expansion (1.5), and the associated regularization methods have used only fairly generic image properties, such as piecewise smoothness. In some applications, particularly when the counts are extremely low or the number of projection views is limited, it can be desirable (or even essential) to apply much stronger prior information to the reconstruction problem. Simple parametric object models such as circles and ellipses (with unknown location, shape, and intensity parameters) have been used for certain applications such as angiography [168–172] or for analysis of imaging system designs, e.g., [173, 174]. Polygonal models have been applied to cardiac image reconstruction, e.g., [175]. More general and flexible object models based on deformable templates have also shown considerable promise and comprise a very active research area, e.g., [176–180]. (See Chapter 3.)

## 1.11 Example results

This section presents representative results of applying penalized likelihood image reconstruction to real PET transmission scan data, following [109]. Many more examples can be found in the references cited throughout this chapter.

We collected a 12–minute transmission scan ($y_i$'s) on a Siemens/CTI ECAT EXACT 921 PET scanner with rotating rod sources of an anthropomorphic thorax phantom (Data Spectrum, Chapel Hill, NC). The sinogram size was 160 radial bins by 192 angles (over $180°$), with 3mm radial spacing. The reconstructed images were $128 \times 128$ pixels that were 4.5mm on each side.

For the penalized-likelihood reconstructions we used a second order penalty function of the form (1.16) with the following potential function proposed in [97]:

$$\psi(t) = \delta^2 \left[\, |t/\delta| - \log\left(1 + |t/\delta|\right) \,\right] , \tag{1.113}$$

where $\beta = 2^{10}$ and $\delta = 0.004\text{cm}^{-1}$ were chosen visually. This function approaches the quadratic $\psi(t) = t^2/2$ as $\delta \to \infty$, but provides a degree of edge preservation for finite $\delta$. The derivative of $\psi$ requires no transcendental functions, which is computationally desirable.

Fig. 1.6 presents a representative example of performance on real PET transmission data. The FBP image is noisy and blurry, since there are only 921K prompt coincidences in this scan [101]. In the upper right is the emission OSEM algorithm applied to the logarithm (1.3) and (1.99). As discussed in Section 1.8.2, this approach yields suboptimal images. The lower two images in Fig. 1.6 were reconstructed by penalized likelihood methods based on (1.15) and (1.16) with the penalty described above. The lower left image used 2 iterations of the OSTR-16 algorithm (1.96) (modified for a nonquadratic penalty as described in [77]). The lower right image used 10 iterations of the PSCA algorithm (1.97). Both penalized likelihood images have lower noise and better spatial resolution than FBP and ML-OSEM-8, as quantified in [77]. There are small differences between the nonconvergent OSTR image and the image reconstructed by the monotonic PSCA algorithm, but whether these differences are important in practice is an open question.

In PET the main purpose of the attenuation map is to form attenuation correction factors (ACFs) for the emission scan. Fig. 1.7 shows sagittal views (47 slices)

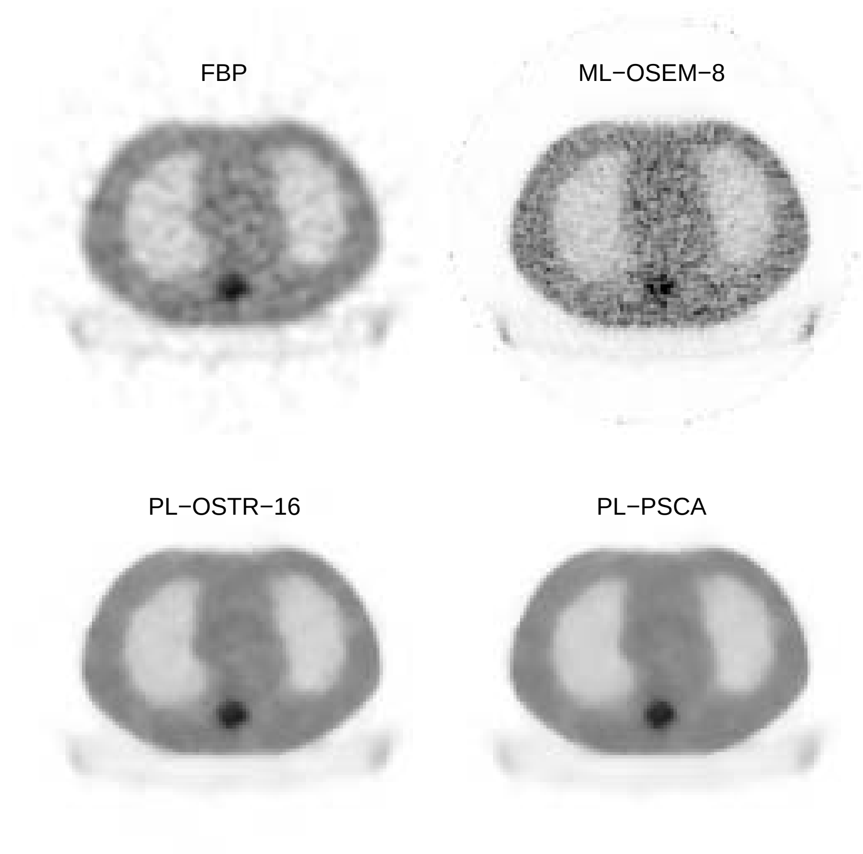



**Figure 1.6**: Reconstructed images of thorax phantom from 12–minute PET transmission scan.

of a patient injected with FDG and scanned with PET. In this case, a 2–minute transmission scan was emulated by binomial thinning of a 12–minute transmission scan [109]. For the subfigures labeled "T-PL" and "T-FBP" the ACFs were computed from attenuation maps reconstructed by penalized likelihood methods or by FBP respectively. For the subfigures labeled "E-PL" and "E-FBP," the emission data was reconstructed by penalized likelihood methods or by FBP respectively. The best image (upper left) is formed when both the emission and transmission images are reconstructed by statistical approaches. The second best image (upper right) is formed by using statistical reconstruction of the attenuation map, but ordinary FBP for the emission data. Clearly for such low-count transmission scans, reducing the noise in the ACFs is as important, if not more so, than how the emission images are reconstructed.

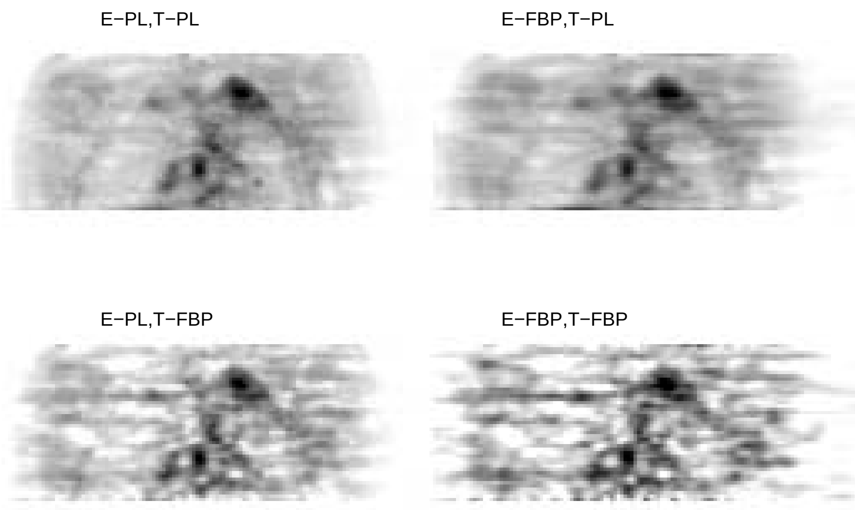



**Figure 1.7**: FDG PET emission images, reconstructed by both FBP (E-FBP) and penalized-likelihood (E-PL) methods. Attenuation correction was performed using attenuation maps generated either by transmission FBP (T-FBP) or transmission penalized-likelihood (T-PL) reconstructions. The use of statistical reconstruction methods significantly reduces image noise.

## 1.12 Summary

We have summarized a wide variety of algorithms for statistical image reconstruction from transmission measurements. Most of the ideas underlying these algorithms are applicable to emission tomography, as well as to image recovery problems in general.

There is a wide variety of algorithms in part because there is yet to have been found any algorithm that has all the desirable properties listed in Section 1.3.1. In cases where the system matrix $\boldsymbol{A}$ can easily be precomputed and stored, and a non-parallel computer is to be used, we recommend the PSCA algorithm of Section 1.6.4. For parallel computing, the conjugate gradient algorithm [31] is a reasonable choice, particularly if exact nonnegativity constraints can be relaxed. If an inexact maximum is acceptable, the OSTR algorithm of Section 1.6.3 is a very practical choice, and is likely to be widely applied given the popularity of the emission OSEM algorithm. Meanwhile, the search continues for an algorithm with the simplicity of OSTR that is parallelizable, monotone and fast converging, and can accommodate any form of system matrix.

## 1.13 Acknowledgements

The ideas in this chapter were greatly influenced by the dissertation research of Hakan Erdoğan [181], who also prepared Fig. 1.6 and 1.7. The author also gratefully acknowledges ongoing collaboration with Neal Clinthorne, Ed Ficaro, Ken Lange, and Les Rogers. The author thanks Ken Hanson for his careful reading of this chapter. This work was supported in part by NIH grants CA-60711 and CA-54362.

## 1.14 Appendix: Poisson properties

Suppose a source transmits $N$ photons of a certain energy along a ray passing through an object towards a specified pixel on the detector. We assume $N$ is a Poisson random variable with mean $N_0$:

$$P[N = k] = \frac{1}{k!} e^{-N_0} N_0^k \,.$$

Each of the $N$ transmitted photons may either pass unaffected ("survive" passage) or may interact with the object. These are Bernoulli trials since the photons interact independently. From Beer's law we know that the probability of surviving passage is given by

$$p = e^{-\int \mu(z)\, dz} \,.$$

The number of photons $M$ that pass unaffected through the object is a random variable, and from Beer's law:

$$P[M = m | N = n] = \binom{n}{m} p^m (1-p)^{n-m}, \; m = 0, \ldots, n \,.$$

Using total probability:

$$P[M = m] = \sum_n P[M = m | N = n] P[N = n]$$

$$= \sum_{n=m}^{\infty} \binom{n}{m} p^m (1-p)^{n-m} \frac{1}{n!} e^{-N_0} N_0^n = \frac{1}{m!} e^{-N_0 p} (N_0 p)^m, \; m = 0, 1, \ldots \,.$$

Therefore the distribution of photons that survive passage is also Poisson, with mean $E[M] = N_0 p$.

Furthermore, by applying Bayes' rule, for $n \geq m \geq 0$:

$$\begin{aligned} P[N = n | M = m] &= \frac{P[M = m | N = n] P[N = n]}{P[M = m]} \\ &= \frac{\begin{pmatrix} n \\ m \end{pmatrix} p^m (1-p)^{n-m} \ \frac{1}{n!} e^{-N_0} N_0^n}{\frac{1}{m!} e^{-N_0 p} (N_0 p)^m} \\ &= \frac{1}{(n-m)!} (N_0 - N_0 p)^{n-m} e^{-(N_0 - N_0 p)} \\ &= \frac{1}{(n-m)!} (E[N] - E[M])^{n-m} e^{-(E[N] - E[M])} . \end{aligned}$$

Thus, conditioned on $M$, the random variable $N - M$ has a Poisson distribution with mean $E[N] - E[M]$. In particular,

$$E[N - M | M] = E[N] - E[M] \, ,$$

which is useful in deriving the transmission EM algorithm proposed in [17].

## 1.15 References


[1] M. M. Ter-Pogossian, M. E. Raichle, and B. E. Sobel, "Positron-emission tomography," *Scientific American*, vol. 243, pp. 171–181, Oct. 1980.

[2] J. M. Ollinger and J. A. Fessler, "Positron emission tomography," *IEEE Sig. Proc. Mag.*, vol. 14, pp. 43–55, Jan. 1997.

[3] T. F. Budinger and G. T. Gullberg, "Three dimensional reconstruction in nuclear medicine emission imaging," *IEEE Tr. Nuc. Sci.*, vol. 21, no. 3, pp. 2–20, 1974.

[4] T. H. Prettyman, R. A. Cole, R. J. Estep, and G. A. Sheppard, "A maximum-likelihood reconstruction algorithm for tomographic gamma-ray nondestructive assay," *Nucl. Instr. Meth. Phys. Res. A.*, vol. 356, pp. 407–52, Mar. 1995.

[5] G. Wang, D. L. Snyder, J. A. O'Sullivan, and M. W. Vannier, "Iterative deblurring for CT metal artifact reduction," *IEEE Tr. Med. Im.*, vol. 15, p. 657, Oct. 1996.

[6] R. M. Leahy and J. Qi, "Statistical approaches in quantitative positron emission tomography," *Statistics and Computing*, 1998.

[7] K. Wienhard, L. Eriksson, S. Grootoonk, M. Casey, U. Pietrzyk, and W. D. Heiss, "Performance evaluation of a new generation positron scanner ECAT EXACT," *J. Comp. Assisted Tomo.*, vol. 16, pp. 804–813, Sept. 1992.

[8] S. R. Cherry, M. Dahlbom, and E. J. Hoffman, "High sensitivity, total body PET scanning using 3D data acquisition and reconstruction," *IEEE Tr. Nuc. Sci.*, vol. 39, pp. 1088–1092, Aug. 1992.

[9] E. P. Ficaro, J. A. Fessler, W. L. Rogers, and M. Schwaiger, "Comparison of Americium-241 and Technicium-99m as transmission sources for attenuation correction of Thallium-201 SPECT imaging of the heart," *J. Nuc. Med.*, vol. 35, pp. 652–63, Apr. 1994.

[10] S. R. Meikle, M. Dahlbom, and S. R. Cherry, "Attenuation correction using count-limited transmission data in positron emission tomography," *J. Nuc. Med.*, vol. 34, pp. 143–150, Jan. 1993.

[11] G. T. Herman, *Image reconstruction from projections: The fundamentals of computerized tomography*. New York: Academic Press, 1980.

[12] A. Macovski, *Medical imaging systems*. New Jersey: Prentice-Hall, 1983.

[13] K. Sauer and C. Bouman, "A local update strategy for iterative reconstruction from projections," *IEEE Tr. Sig. Proc.*, vol. 41, pp. 534–548, Feb. 1993.

[14] J. A. Fessler, "Hybrid Poisson/polynomial objective functions for tomographic image reconstruction from transmission scans," *IEEE Tr. Im. Proc.*, vol. 4, pp. 1439–50, Oct. 1995.

[15] D. S. Lalush and B. M. W. Tsui, "MAP-EM and WLS-MAP-CG reconstruction methods for transmission imaging in cardiac SPECT," in *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, vol. 2, pp. 1174–1178, 1993.

[16] J. A. Fessler, "Mean and variance of implicitly defined biased estimators (such as penalized maximum likelihood): Applications to tomography," *IEEE Tr. Im. Proc.*, vol. 5, pp. 493–506, Mar. 1996.

[17] K. Lange and R. Carson, "EM reconstruction algorithms for emission and transmission tomography," *J. Comp. Assisted Tomo.*, vol. 8, pp. 306–316, Apr. 1984.

[18] C. Bouman and K. Sauer, "Fast numerical methods for emission and transmission tomographic reconstruction," in *Proc. 27th Conf. Info. Sci. Sys., Johns Hopkins*, pp. 611–616, 1993.

[19] C. A. Bouman and K. Sauer, "A unified approach to statistical tomography using coordinate descent optimization," *IEEE Tr. Im. Proc.*, vol. 5, pp. 480–92, Mar. 1996.

[20] J. A. Fessler and W. L. Rogers, "Spatial resolution properties of penalized-likelihood image reconstruction methods: Space-invariant tomographs," *IEEE Tr. Im. Proc.*, vol. 5, pp. 1346–58, Sept. 1996.

[21] P. M. Joseph and R. D. Spital, "A method for correcting bone induced artifacts in computed tomography scanners," *J. Comp. Assisted Tomo.*, vol. 2, pp. 100–8, 1978.

[22] B. Chan, M. Bergström, M. R. Palmer, C. Sayre, and B. D. Pate, "Scatter distribution in transmission measurements with positron emission tomography," *J. Comp. Assisted Tomo.*, vol. 10, pp. 296–301, Mar. 1986.

[23] E. J. Hoffman, S. C. Huang, M. E. Phelps, and D. E. Kuhl, "Quantitation in positron emission computed tomography: 4 Effect of accidental coincidences," *J. Comp. Assisted Tomo.*, vol. 5, no. 3, pp. 391–400, 1981.

[24] M. E. Casey and E. J. Hoffman, "Quantitation in positron emission computed tomography: 7 a technique to reduce noise in accidental coincidence measurements and coincidence efficiency calibration," *J. Comp. Assisted Tomo.*, vol. 10, no. 5, pp. 845–850, 1986.

[25] E. P. Ficaro, W. L. Rogers, and M. Schwaiger, "Comparison of Am-241 and Tc-99m as transmission sources for the attenuation correction of Tl-201 cardiac SPECT studies," *J. Nuc. Med. (Abs. Book)*, vol. 34, p. 30, May 1993.

[26] E. P. Ficaro, J. A. Fessler, R. J. Ackerman, W. L. Rogers, J. R. Corbett, and M. Schwaiger, "Simultaneous transmission-emission Tl-201 cardiac SPECT: Effect of attenuation correction on myocardial tracer distribution," *J. Nuc. Med.*, vol. 36, pp. 921–31, June 1995.

[27] E. P. Ficaro, J. A. Fessler, P. D. Shreve, J. N. Kritzman, P. A. Rose, and J. R. Corbett, "Simultaneous transmission/emission myocardial perfusion tomography: Diagnostic accuracy of attenuation-corrected 99m-Tc-Sestamibi SPECT," *Circulation*, vol. 93, pp. 463–73, Feb. 1996.

[28] D. F. Yu and J. A. Fessler, "Mean and variance of photon counting with deadtime," in *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, 1999.

[29] D. F. Yu and J. A. Fessler, "Mean and variance of singles photon counting with deadtime," *Phys. Med. Biol.*, 1999. Submitted.

[30] D. F. Yu and J. A. Fessler, "Mean and variance of coincidence photon counting with deadtime," *Phys. Med. Biol.*, 1999. Submitted.

[31] E. U. Mumcuoglu, R. Leahy, S. R. Cherry, and Z. Zhou, "Fast gradient-based methods for Bayesian reconstruction of transmission and emission PET images," *IEEE Tr. Med. Im.*, vol. 13, pp. 687–701, Dec. 1994.

[32] M. Yavuz and J. A. Fessler, "New statistical models for randoms-precorrected PET scans," in *Information Processing in Medical Im.* (J. Duncan and G. Gindi, eds.), vol. 1230 of *Lecture Notes in Computer Science*, pp. 190–203, Berlin: Springer Verlag, 1997.

[33] M. Yavuz and J. A. Fessler, "Statistical image reconstruction methods for randoms-precorrected PET scans," *Med. Im. Anal.*, vol. 2, no. 4, pp. 369–378, 1998.

[34] M. Yavuz and J. A. Fessler, "Penalized-likelihood estimators and noise analysis for randoms-precorrected PET transmission scans," *IEEE Tr. Med. Im.*, vol. 18, pp. 665–74, Aug. 1999.

[35] A. C. Kak and M. Slaney, *Principles of computerized tomographic imaging*. New York: IEEE Press, 1988.

[36] A. Celler, A. Sitek, E. Stoub, P. Hawman, R. Harrop, and D. Lyster, "Multiple line source array for SPECT transmission scans: Simulation, phantom and patient studies," *J. Nuc. Med.*, vol. 39, pp. 2183–9, Dec. 1998.

[37] E. L. Lehmann, *Theory of point estimation*. New York: Wiley, 1983.

[38] K. Sauer and B. Liu, "Nonstationary filtering of transmission tomograms in high photon counting noise," *IEEE Tr. Med. Im.*, vol. 10, pp. 445–452, Sept. 1991.

[39] J. A. Fessler, "Tomographic reconstruction using information weighted smoothing splines," in *Information Processing in Medical Im.* (H. H. Barrett and A. F. Gmitro, eds.), vol. 687 of *Lecture Notes in Computer Science*, pp. 372–86, Berlin: Springer Verlag, 1993.

[40] M. N. Wernick and C. T. Chen, “Superresolved tomography by convex projections and detector motion,” *J. Opt. Soc. Am. A*, vol. 9, pp. 1547–1553, Sept. 1992.

[41] S. H. Manglos, “Truncation artifact suppression in cone-beam radionuclide transmission CT using maximum likelihood techniques: evaluation with human subjects,” *Phys. Med. Biol.*, vol. 37, pp. 549–562, Mar. 1992.

[42] E. P. Ficaro and J. A. Fessler, “Iterative reconstruction of truncated fan beam transmission data,” in *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, vol. 3, 1993.

[43] J. A. Case, T. S. Pan, M. A. King, D. S. Luo, B. C. Penney, and M. S. Z. Rabin, “Reduction of truncation artifacts in fan beam transmission imaging using a spatially varying gamma prior,” *IEEE Tr. Nuc. Sci.*, vol. 42, pp. 2260–5, Dec. 1995.

[44] S. H. Manglos, G. M. Gagne, A. Krol, F. D. Thomas, and R. Narayanaswamy, “Transmission maximum-likelihood reconstruction with ordered subsets for cone beam CT,” *Phys. Med. Biol.*, vol. 40, pp. 1225–41, July 1995.

[45] T.-S. Pan, B. M. W. Tsui, and C. L. Bryne, “Choice of initial conditions in the ML reconstruction of fan-beam transmission with truncated projection data,” *IEEE Tr. Med. Im.*, vol. 16, pp. 426–38, Aug. 1997.

[46] G. L. Zeng and G. T. Gullberg, “An SVD study of truncated transmission data in SPECT,” *IEEE Tr. Nuc. Sci.*, vol. 44, pp. 107–11, Feb. 1997.

[47] A. J. Rockmore and A. Macovski, “A maximum likelihood approach to transmission image reconstruction from projections,” *IEEE Tr. Nuc. Sci.*, vol. 24, pp. 1929–1935, June 1977.

[48] Y. Censor, “Finite series expansion reconstruction methods,” *Proc. IEEE*, vol. 71, pp. 409–419, Mar. 1983.

[49] R. Schwinger, S. Cool, and M. King, “Area weighted convolutional interpolation for data reprojection in single photon emission computed tomography,” *Med. Phys.*, vol. 13, pp. 350–355, May 1986.

[50] S. C. B. Lo, “Strip and line path integrals with a square pixel matrix: A unified theory for computational CT projections,” *IEEE Tr. Med. Im.*, vol. 7, pp. 355–363, Dec. 1988.

[51] J. A. Fessler, “ASPIRE 3.0 user’s guide: A sparse iterative reconstruction library,” Tech. Rep. 293, Comm. and Sign. Proc. Lab., Dept. of EECS, Univ. of Michigan, Ann Arbor, MI, 48109-2122, July 1995. Available from `http://www.eecs.umich.edu/~fessler`.

[52] D. L. Snyder, M. I. Miller, L. J. Thomas, and D. G. Politte, “Noise and edge artifacts in maximum-likelihood reconstructions for emission tomography,” *IEEE Tr. Med. Im.*, vol. 6, pp. 228–238, Sept. 1987.

[53] J. A. Browne and T. J. Holmes, “Maximum likelihood techniques in X-ray computed tomography,” in *Medical imaging systems techniques and applications: Diagnosis optimization techniques* (C. T. Leondes, ed.), vol. 3, pp. 117–46, Amsteldijk, Netherlands: Gordon and Breach, 1997.

[54] J. A. Fessler, "Resolution properties of regularized image reconstruction methods," Tech. Rep. 297, Comm. and Sign. Proc. Lab., Dept. of EECS, Univ. of Michigan, Ann Arbor, MI, 48109-2122, Aug. 1995.

[55] J. A. Fessler and W. L. Rogers, "Uniform quadratic penalties cause nonuniform image resolution (and sometimes vice versa)," in *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, vol. 4, pp. 1915–1919, 1994.

[56] J. W. Stayman and J. A. Fessler, "Spatially-variant roughness penalty design for uniform resolution in penalized-likelihood image reconstruction," in *Proc. IEEE Intl. Conf. on Image Processing*, vol. 2, pp. 685–9, 1998.

[57] J. W. Stayman and J. A. Fessler, "Regularization for uniform spatial resolution properties in penalized-likelihood image reconstruction," *IEEE Tr. Med. Im.*, 1998.

[58] J. Qi and R. M. Leahy, "A theoretical study of the contrast recovery and variance of MAP reconstructions with applications to the selection of smoothing parameters," *IEEE Tr. Med. Im.*, vol. 18, pp. 293–305, Apr. 1999.

[59] J. Qi and R. M. Leahy, "Resolution and noise properties of MAP reconstruction for fully 3D PET," in *Proc. of the 1999 Intl. Mtg. on Fully 3D Im. Recon. in Rad. Nuc. Med.*, pp. 35–9, 1999.

[60] J. Besag, "On the statistical analysis of dirty pictures," *J. Royal Stat. Soc. Ser. B*, vol. 48, no. 3, pp. 259–302, 1986.

[61] S. C. Huang, R. E. Carson, M. E. Phelps, E. J. Hoffman, H. R. Schelbert, and D. E. Kuhl, "A boundary method for attenuation correction in positron computed tomography," *J. Nuc. Med.*, vol. 22, no. 1, pp. 627–637, 1981.

[62] E. Z. Xu, N. A. Mullani, K. L. Gould, and W. L. Anderson, "A segmented attenuation correction for PET," *J. Nuc. Med.*, vol. 32, pp. 161–165, Jan. 1991.

[63] M. Xu, W. K. Luk, P. D. Cutler, and W. M. Digby, "Local threshold for segmented attenuation correction of PET imaging of the thorax," *IEEE Tr. Nuc. Sci.*, vol. 41, pp. 1532–1537, Aug. 1994.

[64] S.-J. Lee, A. Rangarajan, and G. Gindi, "Bayesian image reconstruction in SPECT using higher order mechanical models as priors," *IEEE Tr. Med. Im.*, vol. 14, pp. 669–80, Dec. 1995.

[65] G. T. Herman, D. Odhner, K. D. Toennies, and S. A. Zenios, "A parallelized algorithm for image reconstruction from noisy projections," in *Large-Scale Numerical Optimization* (T. F. Coleman and Y. Li, eds.), pp. 3–21, Philadelphia: SIAM, 1990.

[66] S. Alenius, U. Ruotsalainen, and J. Astola, "Using local median as the location of the prior distribution in iterative emission tomography image reconstruction," *IEEE Tr. Nuc. Sci.*, vol. 45, pp. 3097–104, Dec. 1998.

[67] J. A. Fessler and S. D. Booth, "Conjugate-gradient preconditioning methods for shift-variant PET image reconstruction," *IEEE Tr. Im. Proc.*, vol. 8, pp. 688–99, May 1999.

[68] A. H. Delaney and Y. Bresler, "Globally convergent edge-preserving regularized reconstruction: an application to limited-angle tomography," *IEEE Tr. Im. Proc.*, vol. 7, pp. 204–221, Feb. 1998.

[69] K. Lange and J. A. Fessler, “Globally convergent algorithms for maximum a posteriori transmission tomography,” *IEEE Tr. Im. Proc.*, vol. 4, pp. 1430–8, Oct. 1995.

[70] R. R. Meyer, “Sufficient conditions for the convergence of monotonic mathematical programming algorithms,” *J. Comput. System. Sci.*, vol. 12, pp. 108–21, 1976.

[71] J. M. Ortega and W. C. Rheinboldt, *Iterative solution of nonlinear equations in several variables*. New York: Academic Press, 1970.

[72] A. R. De Pierro, “On the relation between the ISRA and the EM algorithm for positron emission tomography,” *IEEE Tr. Med. Im.*, vol. 12, pp. 328–333, June 1993.

[73] A. R. De Pierro, “A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography,” *IEEE Tr. Med. Im.*, vol. 14, pp. 132–137, Mar. 1995.

[74] K. Lange, *Numerical analysis for statisticians*. New York: Springer-Verlag, 1999.

[75] J. A. Fessler and A. O. Hero, “Penalized maximum-likelihood image reconstruction using space-alternating generalized EM algorithms,” *IEEE Tr. Im. Proc.*, vol. 4, pp. 1417–29, Oct. 1995.

[76] J. A. Fessler, N. H. Clinthorne, and W. L. Rogers, “On complete data spaces for PET reconstruction algorithms,” *IEEE Tr. Nuc. Sci.*, vol. 40, pp. 1055–61, Aug. 1993.

[77] H. Erdoğan and J. A. Fessler, “Monotonic algorithms for transmission tomography,” *IEEE Tr. Med. Im.*, vol. 18, pp. 801–14, Sept. 1999.

[78] J. A. Fessler and H. Erdoğan, “A paraboloidal surrogates algorithm for convergent penalized-likelihood emission image reconstruction,” in *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, vol. 2, pp. 1132–5, 1998.

[79] L. A. Shepp and Y. Vardi, “Maximum likelihood reconstruction for emission tomography,” *IEEE Tr. Med. Im.*, vol. 1, pp. 113–122, Oct. 1982.

[80] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *J. Royal Stat. Soc. Ser. B*, vol. 39, no. 1, pp. 1–38, 1977.

[81] C. F. J. Wu, “On the convergence properties of the EM algorithm,” *Ann. Stat.*, vol. 11, no. 1, pp. 95–103, 1983.

[82] X. L. Meng and D. B. Rubin, “Maximum likelihood estimation via the ECM algorithm: A general framework,” *Biometrika*, vol. 80, no. 2, pp. 267–278, 1993.

[83] J. A. Fessler and A. O. Hero, “Space-alternating generalized expectation-maximization algorithm,” *IEEE Tr. Sig. Proc.*, vol. 42, pp. 2664–77, Oct. 1994.

[84] C. H. Liu and D. B. Rubin, “The ECME algorithm: a simple extension of EM and ECM with faster monotone convergence,” *Biometrika*, vol. 81, no. 4, pp. 633–48, 1994.

[85] K. Lange, “A Quasi-Newton acceleration of the EM Algorithm,” *Statistica Sinica*, vol. 5, pp. 1–18, Jan. 1995.

[86] D. A. van Dyk, X. L. Meng, and D. B. Rubin, “Maximum likelihood estimation via the ECM algorithm: computing the asymptotic variance,” *Statistica Sinica*, vol. 5, pp. 55–76, Jan. 1995.

[87] X. L. Meng and D. van Dyk, "The EM algorithm - An old folk song sung to a fast new tune," *J. Royal Stat. Soc. Ser. B*, vol. 59, no. 3, pp. 511–67, 1997.

[88] J. A. Fessler, "EM and gradient algorithms for transmission tomography with background contamination," Tech. Rep. UM-PET-JF-94-1, Cyclotron PET Facility, Univ. of Michigan, Ann Arbor, MI, 48109-2122, Dec. 1994. Available from `http://www.eecs.umich.edu/~fessler`.

[89] J. T. Kent and C. Wright, "Some suggestions for transmission tomography based on the EM algorithm," in *Stochastic Models, Statistical Methods, and Algorithms in Im. Analysis* (M. P. P Barone, A Frigessi, ed.), vol. 74 of *Lecture Notes in Statistics*, pp. 219–232, New York: Springer, 1992.

[90] H. Stark, *Image recovery, theory and application*. Orlando: Academic, 1987.

[91] J. A. Browne and T. J. Holmes, "Developments with maximum likelihood X-ray computed tomography," *IEEE Tr. Med. Im.*, vol. 12, pp. 40–52, Mar. 1992.

[92] J. M. Ollinger and G. C. Johns, "The use of maximum *a-posteriori* and maximum likelihood transmission images for attenuation correction PET," in *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, vol. 2, pp. 1185–1187, 1992.

[93] J. M. Ollinger, "Maximum likelihood reconstruction of transmission images in emission computed tomography via the EM algorithm," *IEEE Tr. Med. Im.*, vol. 13, pp. 89–101, Mar. 1994.

[94] L. Kaufman, "Implementing and accelerating the EM algorithm for positron emission tomography," *IEEE Tr. Med. Im.*, vol. 6, pp. 37–51, Mar. 1987.

[95] K. Lange, M. Bahn, and R. Little, "A theoretical study of some maximum likelihood algorithms for emission and transmission tomography," *IEEE Tr. Med. Im.*, vol. 6, pp. 106–114, June 1987.

[96] K. Lange, "An overview of Bayesian methods in image reconstruction," in *Proc. SPIE 1351, Dig. Im. Synth. and Inverse Optics*, pp. 270–287, 1990.

[97] K. Lange, "Convergence of EM image reconstruction algorithms with Gibbs smoothing," *IEEE Tr. Med. Im.*, vol. 9, pp. 439–446, Dec. 1990. Corrections, T-MI, 10:2(288), June 1991.

[98] P. J. Maniawski, H. T. Morgan, G. L. Gullberg, G. L. Zeng, A. E. Welch, and C. H. Tung, "Performance evaluation of a transmission reconstruction algorithm with simultaneous transmission-emission SPECT system in a presence of data truncation," in *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, vol. 4, pp. 1578–81, 1994.

[99] S. G. Kim, "Draft on some modifications of GCA algorithms for transmission CT," 1998. preprint Dec. 16, 1998.

[100] J. A. Fessler and A. O. Hero, "Complete-data spaces and generalized EM algorithms," in *Proc. IEEE Conf. Acoust. Speech Sig. Proc.*, vol. 4, pp. 1–4, 1993.

[101] J. A. Fessler, E. P. Ficaro, N. H. Clinthorne, and K. Lange, "Grouped-coordinate ascent algorithms for penalized-likelihood transmission image reconstruction," *IEEE Tr. Med. Im.*, vol. 16, pp. 166–75, Apr. 1997.

[102] H. M. Hudson and R. S. Larkin, “Accelerated image reconstruction using ordered subsets of projection data,” *IEEE Tr. Med. Im.*, vol. 13, pp. 601–609, Dec. 1994.

[103] C. L. Byrne, “Block-iterative methods for image reconstruction from projections,” *IEEE Tr. Im. Proc.*, vol. 5, pp. 792–3, May 1996.

[104] C. L. Byrne, “Convergent block-iterative algorithms for image reconstruction from inconsistent data,” *IEEE Tr. Im. Proc.*, vol. 6, pp. 1296–1304, Sept. 1997.

[105] C. L. Byrne, “Accelerating the EMML algorithm and related iterative algorithms by rescaled block-iterative methods,” *IEEE Tr. Im. Proc.*, vol. 7, pp. 100–9, Jan. 1998.

[106] J. Nuyts, B. D. Man, P. Dupont, M. Defrise, P. Suetens, and L. Mortelmans, “Iterative reconstruction for helical CT: A simulation study,” *Phys. Med. Biol.*, vol. 43, pp. 729–37, Apr. 1998.

[107] C. Kamphius and F. J. Beekman, “Accelerated iterative transmission CT reconstruction using an ordered subsets convex algorithm,” *IEEE Tr. Med. Im.*, vol. 17, pp. 1001–5, Dec. 1998.

[108] H. Erdoğan, G. Gualtieri, and J. A. Fessler, “An ordered subsets algorithm for transmission tomography,” in *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, 1998. Inadvertently omitted from proceedings. Available from web page.

[109] H. Erdoğan and J. A. Fessler, “Ordered subsets algorithms for transmission tomography,” *Phys. Med. Biol.*, vol. 44, pp. 2835–51, Nov. 1999.

[110] T. Hebert and R. Leahy, “A Bayesian reconstruction algorithm for emission tomography using a Markov random field prior,” in *Proc. SPIE 1092, Med. Im. III: Im. Proc.*, pp. 458–4662, 1989.

[111] T. Hebert and R. Leahy, “A generalized EM algorithm for 3-D Bayesian reconstruction from Poisson data using Gibbs priors,” *IEEE Tr. Med. Im.*, vol. 8, pp. 194–202, June 1989.

[112] T. J. Hebert and R. Leahy, “Statistic-based MAP image reconstruction from Poisson data using Gibbs priors,” *IEEE Tr. Sig. Proc.*, vol. 40, pp. 2290–2303, Sept. 1992.

[113] G. Gullberg and B. M. W. Tsui, “Maximum entropy reconstruction with constraints: iterative algorithms for solving the primal and dual programs,” in *Proc. Tenth Intl. Conf. on Information Processing in Medical Im.* (C. N. de Graaf and M. A. Viergever, eds.), pp. 181–200, New York: Plenum Press, 1987.

[114] C. A. Bouman, K. Sauer, and S. S. Saquib, “Tractable models and efficient algorithms for Bayesian tomography,” in *Proc. IEEE Conf. Acoust. Speech Sig. Proc.*, vol. 5, pp. 2907–10, 1995.

[115] S. Saquib, J. Zheng, C. A. Bouman, and K. D. Sauer, “Provably convergent coordinate descent in statistical tomographic reconstruction,” in *Proc. IEEE Intl. Conf. on Image Processing*, vol. 2, pp. 741–4, 1996.

[116] J. Zheng, S. Saquib, K. Sauer, and C. Bouman, “Functional substitution methods in optimization for Bayesian tomography,” *IEEE Tr. Im. Proc.*, Mar. 1997. IEEE Tr. Image Proc.

[117] J. A. Fessler, "Grouped coordinate descent algorithms for robust edge-preserving image restoration," in *Proc. SPIE 3071, Im. Recon. and Restor. II*, pp. 184–94, 1997.

[118] P. J. Huber, *Robust statistics*. New York: Wiley, 1981.

[119] H. M. Hudson, J. Ma, and P. Green, "Fisher's method of scoring in statistical image reconstruction: comparison of Jacobi and Gauss-Seidel iterative schemes," *Stat. Meth. Med. Res.*, vol. 3, no. 1, pp. 41–61, 1994.

[120] S. L. Hillis and C. S. Davis, "A simple justification of the iterative fitting procedure for generalized linear models," *American Statistician*, vol. 48, pp. 288–289, Nov. 1994.

[121] H. Erdoğan and J. A. Fessler, "Accelerated monotonic algorithms for transmission tomography," in *Proc. IEEE Intl. Conf. on Image Processing*, vol. 2, pp. 680–4, 1998.

[122] J. A. Fessler, E. P. Ficaro, N. H. Clinthorne, and K. Lange, "Fast parallelizable algorithms for transmission image reconstruction," in *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, vol. 3, pp. 1346–50, 1995.

[123] K. D. Sauer, S. Borman, and C. A. Bouman, "Parallel computation of sequential pixel updates in statistical tomographic reconstruction," in *Proc. IEEE Intl. Conf. on Image Processing*, vol. 3, pp. 93–6, 1995.

[124] S. T. Jensen, S. Johansen, and S. L. Lauritzen, "Globally convergent algorithms for maximizing a likelihood function," *Biometrika*, vol. 78, no. 4, pp. 867–77, 1991.

[125] G. H. Golub and C. F. Van Loan, *Matrix computations*. Johns Hopkins Univ. Press, 1989.

[126] E. Mumcuoglu, R. Leahy, and S. Cherry, "A statistical approach to transmission image reconstruction from ring source calibration measurements in PET," in *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, vol. 2, pp. 910–912, 1992.

[127] E. Ü. Mumcuoğlu and R. M. Leahy, "A gradient projection conjugate gradient algorithm for Bayesian PET reconstruction," in *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, vol. 3, pp. 1212–6, 1994.

[128] J. J. Moré and G. Toraldo, "On the solution of large quadratic programming problems with bound constraints," *SIAM J. Optim.*, vol. 1, pp. 93–113, Feb. 1991.

[129] M. Bierlaire, P. L. Toint, and D. Tuyttens, "On iterative algorithms for linear least squares problems with bound constraints," *Linear Algebra and its Applications*, vol. 143, pp. 111–43, 1991.

[130] K. Lange, "An adaptive barrier method for convex programming," *Methods and Appl. of Analysis*, vol. 1, no. 4, pp. 392–402, 1994.

[131] M. Al-Baali and R. Fletcher, "On the order of convergence of preconditioned nonlinear conjugate gradient methods," *SIAM J. Sci. Comp.*, vol. 17, pp. 658–65, May 1996.

[132] M. S. Kaplan, D. R. Haynor, and H. Vija, "A differential attenuation method for simultaneous estimation of SPECT activity and attenuation distributions," *IEEE Tr. Nuc. Sci.*, vol. 46, pp. 535–41, June 1999.

[133] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization," *SIAM J. Sci. Comp.*, vol. 16, pp. 1190–1208, 1995.

[134] Y. Censor, P. P. B. Eggermont, and D. Gordon, "Strong underrelaxation in Kaczmarz's method for inconsistent systems," *Numerische Mathematik*, vol. 41, pp. 83–92, 1983.

[135] Z. Liang and J. Ye, "Reconstruction of object-specific attenuation map for quantitative SPECT," in *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, vol. 2, pp. 1231–1235, 1993.

[136] G. L. Zeng and G. T. Gullberg, "A MAP algorithm for transmission computed tomography," in *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, vol. 2, pp. 1202–1204, 1993.

[137] J. M. M. Anderson, B. A. Mair, M. Rao, and C. H. Wu, "A weighted least-squares method for PET," in *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, vol. 2, pp. 1292–6, 1995.

[138] C. Bouman and K. Sauer, "Nonlinear multigrid methods of optimization in Bayesian tomographic image reconstruction," in *SPIE Neural and Stoch. Methods in Image and Signal Proc.*, 1992.

[139] C. Bouman and K. Sauer, "A generalized Gaussian image model for edge-preserving MAP estimation," *IEEE Tr. Im. Proc.*, vol. 2, pp. 296–310, July 1993.

[140] P. J. Green, "Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives," *J. Royal Stat. Soc. Ser. B*, vol. 46, no. 2, pp. 149–192, 1984.

[141] M. B. Dollinger and R. G. Staudte, "Influence functions of iteratively reweighted least squares estimators," *J. Am. Stat. Ass.*, vol. 86, pp. 709–716, Sept. 1991.

[142] J. Qi, R. M. Leahy, C. Hsu, T. H. Farquhar, and S. R. Cherry, "Fully 3D Bayesian image reconstruction for the ECAT EXACT HR+," *IEEE Tr. Nuc. Sci.*, vol. 45, pp. 1096–1103, June 1998.

[143] J. Qi, R. M. Leahy, S. R. Cherry, A. Chatziioannou, and T. H. Farquhar, "High resolution 3D Bayesian image reconstruction using the microPET small-animal scanner," *Phys. Med. Biol.*, vol. 43, pp. 1001–14, Apr. 1998.

[144] J. A. Browne and A. R. D. Pierro, "A row-action alternative to the EM algorithm for maximizing likelihoods in emission tomography," *IEEE Tr. Med. Im.*, vol. 15, pp. 687–99, Oct. 1996.

[145] A. M. Thompson, J. C. Brown, J. W. Kay, and D. M. Titterington, "A study of methods or choosing the smoothing parameter in image restoration by regularization," *IEEE Tr. Patt. Anal. Mach. Int.*, vol. 13, no. 4, pp. 326–339, 1991.

[146] J. W. Hilgers and W. R. Reynolds, "Instabilities in the optimal regularization parameter relating to image recovery problems," *J. Opt. Soc. Am. A*, vol. 9, pp. 1273–1279, Aug. 1992.

[147] Y. Pawitan and F. O'Sullivan, "Data-dependent bandwidth selection for emission computed tomography reconstruction," *IEEE Tr. Med. Im.*, vol. 12, pp. 167–172, June 1993.

[148] F. O'Sullivan and Y. Pawitan, "Bandwidth selection for indirect density estimation based on corrupted histogram data," *J. Am. Stat. Ass.*, vol. 91, pp. 610–26, June 1996.

[149] C. R. Vogel, "Non-convergence of the L-curve regularization parameter selection method," *Inverse Prob.*, vol. 12, pp. 535–47, Aug. 1996.

[150] P. P. B. Eggermont and V. N. LaRiccia, "Nonlinearly smoothed EM density estimation with automated smoothing parameter selection for nonparametric deconvolution problems," *J. Am. Stat. Ass.*, vol. 92, pp. 1451–8, Dec. 1997.

[151] D. M. Higdon, J. E. Bowsher, V. E. Johnson, T. G. Turkington, D. R. Gilland, and R. J. Jaszczak, "Fully Bayesian estimation of Gibbs hyperparameters for emission computed tomography data," *IEEE Tr. Med. Im.*, vol. 16, p. 516, Oct. 1997.

[152] G. Sebastiani and F. Godtliebsen, "On the use of Gibbs priors for Bayesian image restoration," *Signal Processing*, vol. 56, pp. 111–18, Jan. 1997.

[153] Z. Zhou, R. M. Leahy, and J. Qi, "Approximate maximum likelihood hyperparameter estimation for Gibbs priors," *IEEE Tr. Im. Proc.*, vol. 6, pp. 844–61, June 1997.

[154] S. S. Saquib, C. A. Bouman, and K. Sauer, "ML parameter estimation for Markov random fields, with applications to Bayesian tomography," *IEEE Tr. Im. Proc.*, vol. 7, pp. 1029–44, July 1998.

[155] M. Nikolova, J. Idier, and A. Mohammad-Djafari, "Inversion of large-support ill-posed linear operators using a piecewise Gaussian MRF," *IEEE Tr. Im. Proc.*, vol. 7, pp. 571–85, Apr. 1998.

[156] K. Sauer and C. Bouman, "Bayesian estimation of transmission tomograms using local optimization operations," in *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, vol. 3, pp. 2089–2093, 1991.

[157] K. Sauer and C. Bouman, "Bayesian estimation of transmission tomograms using segmentation based optimization," *IEEE Tr. Nuc. Sci.*, vol. 39, pp. 1144–1152, Aug. 1992.

[158] R. Mifflin, D. Sun, and L. Qi, "Quasi-Newton bundle-type methods for nondifferentiable convex optimization," *SIAM J. Optim.*, vol. 8, pp. 583–603, May 1998.

[159] Y. Censor, D. E. Gustafson, A. Lent, and H. Tuy, "A new approach to the emission computerized tomography problem: simultaneous calculation of attenuation and activity coefficients," *IEEE Tr. Nuc. Sci.*, vol. 26, Jan. 1979.

[160] A. Welch, R. Clack, F. Natterer, and G. T. Gullberg, "Toward accurate attenuation correction in SPECT without transmission measurements," *IEEE Tr. Med. Im.*, vol. 16, p. 532, Oct. 1997.

[161] R. E. Alvarez and A. Macovski, "Energy-selective reconstructions in X-ray computed tomography," *Phys. Med. Biol.*, vol. 21, pp. 733–44, 1976.

[162] N. H. Clinthorne, "A constrained dual-energy reconstruction method for material-selective transmission tomography," *Nucl. Instr. Meth. Phys. Res. A.*, vol. 352, pp. 347–8, Dec. 1994.

[163] P. Sukovic and N. H. Clinthorne, “Penalized weighted least-squares image reconstruction in single and dual energy X-ray computed tomography,” *IEEE Tr. Med. Im.*, 1999. Submitted.

[164] J. A. Fessler, D. F. Yu, and E. P. Ficaro, “Maximum likelihood transmission image reconstruction for overlapping transmission beams,” in *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, 1999.

[165] D. F. Yu, J. A. Fessler, and E. P. Ficaro, “Maximum likelihood transmission image reconstruction for overlapping transmission beams,” *IEEE Tr. Med. Im.*, 1999. Submitted.

[166] J. E. Bowsher, M. P. Tornai, D. R. Gilland, D. E. G. Trotter, and R. J. Jaszczak, “An EM algorithm for modeling multiple or extended TCT sources,” in *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, 1999.

[167] A. Krol, J. E. Bowsher, S. H. Manglos, D. H. Feiglin, and F. D. Thomas, “An EM algorithm for estimating SPECT emission and transmission parameters from emission data only,” *Phys. Med. Biol.*, 1999. Submitted.

[168] D. J. Rossi and A. S. Willsky, “Reconstruction from projections based on detection and estimation of objects—Parts i & II: Performance analysis and robustness analysis,” *IEEE Tr. Acoust. Sp. Sig. Proc.*, vol. 32, pp. 886–906, Aug. 1984.

[169] D. J. Rossi, A. S. Willsky, and D. M. Spielman, “Object shape estimation from tomographic measurements—a performance evaluation,” *Signal Processing*, vol. 18, pp. 63–88, Sept. 1989.

[170] Y. Bresler, J. A. Fessler, and A. Macovski, “Model based estimation techniques for 3-D reconstruction from projections,” *Machine Vision and Applications*, vol. 1, no. 2, pp. 115–26, 1988.

[171] Y. Bresler, J. A. Fessler, and A. Macovski, “A Bayesian approach to reconstruction from incomplete projections of a multiple object 3-D domain,” *IEEE Tr. Patt. Anal. Mach. Int.*, vol. 11, pp. 840–58, Aug. 1989.

[172] J. A. Fessler and A. Macovski, “Object-based 3-D reconstruction of arterial trees from magnetic resonance angiograms,” *IEEE Tr. Med. Im.*, vol. 10, pp. 25–39, Mar. 1991.

[173] S. P. Müller, M. F. Kijewski, S. C. Moore, and B. L. Holman, “Maximum-likelihood estimation: a mathematical model for quantitation in nuclear medicine,” *J. Nuc. Med.*, vol. 31, pp. 1693–1701, Oct. 1990.

[174] C. K. Abbey, E. Clarkson, H. H. Barrett, S. P. Müller, and F. J. Rybicki, “A method for approximating the density of maximum likelihood and maximum a posteriori estimates under a Gaussian noise model,” *Med. Im. Anal.*, vol. 2, no. 4, pp. 395–403, 1998.

[175] P. C. Chiao, W. L. Rogers, N. H. Clinthorne, J. A. Fessler, and A. O. Hero, “Model-based estimation for dynamic cardiac studies using ECT,” *IEEE Tr. Med. Im.*, vol. 13, pp. 217–26, June 1994.

[176] Y. Amit and K. Manbeck, “Deformable template models for emission tomography,” *IEEE Tr. Med. Im.*, vol. 12, pp. 260–268, June 1993.

[177] K. M. Hanson, "Bayesian reconstruction based on flexible priors," *J. Opt. Soc. Am. A*, vol. 10, pp. 997–1004, May 1993.

[178] X. L. Battle, G. S. Cunningham, and K. M. Hanson, "Tomographic reconstruction using 3D deformable models," *Phys. Med. Biol.*, vol. 43, pp. 983–90, 1998.

[179] G. S. Cunningham, K. M. Hanson, and X. L. Battle, "Three-dimensional reconstructions from low-count SPECT data using deformable models," *Optics Express*, vol. 2, pp. 227–36, 1998. http://www.osa.org.

[180] X. L. Battle and Y. Bizais, "3D attenuation map reconstruction using geometrical models and free form deformations," in *Proc. of the 1999 Intl. Mtg. on Fully 3D Im. Recon. in Rad. Nuc. Med.*, pp. 181–184, 1999.

[181] H. Erdoğan, *Statistical image reconstruction algorithms using paraboloidal surrogates for PET transmission scans*. Ph.D. thesis, Univ. of Michigan, Ann Arbor, MI, 48109-2122, Ann Arbor, MI., July 1999.

# CHAPTER 6
# Extracting Surface Models of the Anatomy from Medical Images

André Guéziec
Consultant

## Contents

## 6.1 Introduction

A large number of applications in medical imaging require geometric models of the surface of various anatomical structures. Although recent direct volume rendering techniques provide high visual quality [1] and speed, using cost-effective special purpose hardware [2], surfaces are very effective for showing the spatial relationship between different structures. (Radio-therapy planning visualization provides an excellent bio-medical example of this [3].) Surface extraction is necessary when surfaces (meshes) are used for further analysis and processing, independently of visualization. Such processes include finite element analysis (Chapter 19) and registration (see Chapter 8). Several methods have recently been developed to help guide surgery using computer-generated displays of anatomical structures registered with live imagery of the patient. Such methods often use surface models for registration and rendering [4–6].

This chapter is cross-disciplinary. One of its goals is to illustrate the flexibility in choosing a surface representation. Surface representations are reviewed in Section 6.2 and illustrated with various examples. Another goal is to describe recent techniques that have been developed for surface extraction from medical images in Sections 6.3, 6.4, and 6.5 and to provide references for further reading.

We also study some methods for optimizing surface models in Section 6.6, such as smoothing (fairing) and simplification. We introduce a few algorithms that operate on a surface and accompany them with real-world examples in Section 6.7.

## 6.2 Surface representations

Various geometric objects may be used to represent a surface. The first object, in Section 6.2.1, specifies a surface with a set of samples (and hence does not define a surface per se). A powerful piece-wise linear representation is introduced in Section 6.2.2. Smooth curved surface representations (Section 6.2.3) are (in theory) more compact and allow the full exploitation of curvature and normal information; however, their topology is less flexible.

### *6.2.1 Point set*

The simplest representation for a surface is probably a *point set*, each point, or *vertex*, being defined by three Cartesian coordinates $(x, y, z)$. Contrary to a curve, for which a set of sample points may be naturally ordered according to a progression along the curve, there is no such natural ordering for surface sample points.

Point sets have been used successfully for registration purposes, as in the "Head in the Hat" scheme [7]. More recent surface registration methods using sample points are described in [8, 9]. Registration between point sets or between a point set and a surface is generally performed by associating a corresponding point (respectively, surface location) to each point of a set and by determining a geometric transformation that best aligns each point with its correspondent using various optimization schemes. See Chapter 8 for an in-depth description of surface registration methods. Point sets have also been used for some visualization tasks where a representation of a surface using discrete points is suitable.

This representation has several limitations, due to the lack of information between sample points: area and volume measurements are not possible; for registration processes, misregistrations can occur due to the fact that a closest sample point and a closest surface point are not necessarily the same.

A number of strategies have been developed for constructing a surface from its samples and thus to bridge gaps between samples [10–13]. None of these techniques, to the best of the author's current knowledge, is guaranteed to produce a satisfactory result in all cases.

### 6.2.2 Triangular mesh

A *triangular mesh* comprises *vertices* and *triangles*, each triangle referring to an ordered triple of vertices. A triangular mesh modeling a portion of a human femur is illustrated in Fig. 6.1. For the type of applications that we are interested in, we consider a particular triangular mesh that is *oriented* and that is also a *manifold*. A triangular mesh is oriented if two triangles sharing an *edge* (an edge is a pair of vertices referred to by a triangle) are such that the two vertices (*endpoints*) of the edge are listed in a different order in both triangles. A triangular mesh is a manifold if each vertex is such that there is only one *triangle fan* incident to it. (A triangle fan of the mesh is a collection of triangles incident to a given vertex and connected by edges, such that each edge is shared by no more than two triangles of the mesh, i.e., is a *regular edge*. A *singular edge* is shared by more than two triangles—see Fig. 6.1(b)). A vertex with a single incident triangle fan is called a *regular vertex* (Fig. 6.1(c)). A vertex with more than one incident triangle fan is called a *singular vertex* (Fig. 6.1(d)).

A boundary edge is an edge that has only one incident triangle. A *boundary* of a manifold triangular mesh is obtained by linking boundary edges connected by vertices. If $n_v$ is the number of vertices of the triangular mesh, $n_e$ the number of edges and $n_t$ the number of triangles, the Euler Number is defined to be $n_v - n_e + n_t$. Supposing that the triangular mesh has no boundaries, then the Euler number equals $2(1 - n_h)$ where $n_h$ is the number of "handles" (exactly as the handle of a tea-cup) of the surface: for instance, it is equal to zero for a torus and to two for a sphere (or a tetrahedron, a cube, etc.). Figure 6.2 illustrates triangular meshes with zero, one, and two handles. When discussing the *topology* of a triangular mesh (or more generally, of a surface) in subsequent sections, we will be referring to the property of the surface to have handles or boundaries and to the number of handles and boundaries.

If the surface has no boundaries, we can also write that $3 * n_t = 2 * n_e$ (this is a standard result that the reader should be able to verify by him/herself when

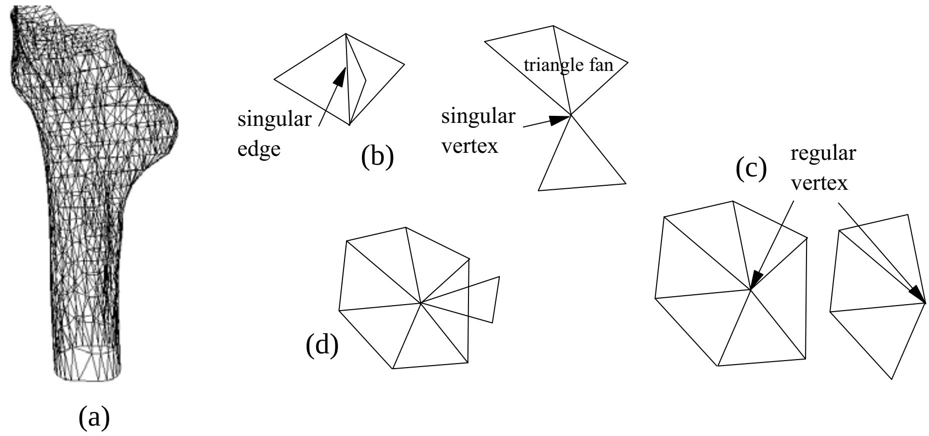



**Figure 6.1:** Triangular meshes: (a) triangular mesh modeling the boundary of a proximal portion of a human femur; (b) singular edge and vertex; (c) regular vertices. (d) a non-manifold mesh having a singular vertex.

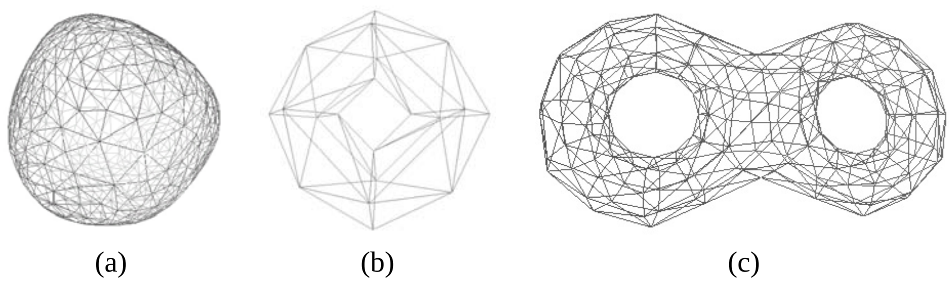



**Figure 6.2**: Triangular meshes with a different Euler Number: (a) 2; (b) 0 (one handle); (c) -2 (two handles).

enumerating the edges by visiting in turn each triangle). It can be seen that $2n_v - 4(1-n_h) = n_t$. Hence, the number of triangles is approximately twice the number of vertices, provided that $n_h/n_v << 1$.

Most of the applications listed in the present chapter utilize a triangular mesh. If the triangles are sufficiently small, it is possible to approximate surfaces closely enough in most cases. Also, as discussed further in Section 6.6.1, a signal processing framework may be applied to triangular meshes, allowing such operations as smoothing.

One drawback of a triangular mesh is the amount of data that is required to represent it. To address this issue, methods have been developed to approximate triangular meshes using fewer triangles (Section 6.6.2). Methods for compressing triangular meshes were also developed, initially in the computer graphics community [14, 15]; these methods facilitate storage, transmission, and rendering.

Delingette/Cotin use "simplex meshes" [16] for modeling the elastic properties of some tissues and simulating surgery. A simplex mesh is the dual of a triangular mesh: each triangle in a triangular mesh corresponds to a vertex in a simplex mesh, each pair of adjacent triangles in a triangular mesh corresponds to an edge in a simplex mesh.

### *6.2.3 Curved surfaces*

Several types of smooth curved surface representations have been developed in the past and used extensively in computer aided design, including Bézier rectangular patches, Bézier triangles, B-spline surfaces, and NURBS (Non-Uniform Rational B-splines). These representations are built by blending a set of *control vertices* using piecewise polynomial or rational functions. Generally, we will refer to them as *splines*. While it is outside the scope of this chapter to review the details of each type of spline, such details are available for instance in Farin [17]. In Section 6.5, however, we explain the details of tensor-product B-spline surfaces. A discussion of various representations is also available in Guéziec [18]. The process of fitting these representations to data is discussed in Dierckx [19].

The use of such representations in engineering is quite different from their use in medical imaging. While spline surfaces allow very effective designing by user-directed manipulation of control vertices, fitting spline surfaces to data is more complex and is still the focus of much research. In the computer aided design community this is in fact a process called "reverse engineering" [20].

Splines are explicit surface representations. *Implicit* representations, which characterize a surface as the zero set of some real-valued function of three-dimensional space, are also possible[1]. Implicit superquadrics are smooth curved surfaces that have been used in medical imaging [21, 22]. Superquadrics may be defined using fewer parameters than splines. Assuming that the fitting process is accurate, these parameters may be useful for tracking and characterizing the deformation of anatomical surfaces in live subjects. Such applications are covered more deeply in Chapter 3.

Another noteworthy implicit representation is Bajaj's implicit algebraic patches, or A-patches [23]. A-patches are embedded in a supporting tetrahedral mesh (modeling not only the surface, but also the enclosed volume) and can be used for dynamic simulations.

## 6.3 Iso-surface extraction

Iso-surface extraction is an essential technique for analyzing and visualizing medical images. The process of extracting an iso-surface that we develop in this section starts with regular volume data. Volume data may be formed with a se-

[1] A three-dimensional medical image, possibly segmented, may be viewed as a regular grid of samples of an implicit function. The process of extracting an iso-surface of Section 6.3 uses such an implicit representation.

ries of two-dimensional image slices (this is the case for CT-scan data) or may be readily available as in MRI-scan data. *Regular volume data* is defined with a set of vertices regularly positioned in space and a function value (e.g., Hounsfield number in CT data) provided at each vertex. Vertices are connected, forming a graph. More specifically, vertices are connected by a three-dimensional lattice, such that each vertex has 6 neighbors except at the boundary of the lattice (see Fig. 6.3). In what follows, we will call the vertices *voxels*. For simplicity, we will refer to the function values as *intensities*. Regular volume data is sometimes referred to as a "three-dimensional image," because one can resample the data on any plane chosen arbitrarily through the volume and obtain an (two dimensional) image as a result.

An *iso-surface* is a surface that will connect all the points of space that have the same function value associated; this function value is called the *iso-value*. A good model of the anatomy is rarely obtained by computing an iso-surface of the raw image data, except in particular cases, e.g., when dry bone is scanned, as illustrated in Section 6.3.6[2]. It is thus preferable to perform a segmentation of the image before extracting an iso-surface. Segmentation is studied in detail in Chapter 2.

There are two fundamental strategies for extracting an iso-surface corresponding to a given iso-value from regular volume data. The strategies differ in how they treat each *cell* of the three-dimensional lattice. The first strategy visits all hexahedral cells of the three-dimensional lattice and draws the (most often, empty) portion of the iso-surface intersecting each cell. Since a polygonal surface is built, this is often called *polygonizing* a cell. The second strategy first decomposes hexahedral cells into tetrahedra and polygonizes tetrahedral cells. There are a number of variants for either strategy. See [24, 25] for a discussion of various algorithms. The reader may also consult a text by the same author [26] for a detailed description of a variant of each strategy and for recent research on iso-surface extraction algorithms.

### *6.3.1 Hexahedral decomposition*

Marching Cubes (Lorensen & Cline, 1987) [27] is probably the best known variant of the hexahedral decomposition method. However, the resulting tiling may be inconsistent across hexahedra, resulting in non-manifold surfaces [24, 28, 29]. (This problem may be resolved in some implementations.) The Wyvill et al. [30] or Kalvin [28] methods for instance do not have this problem. For brevity, we concentrate on the tetrahedral decomposition method in the next Section 6.3.2. A complete description of an implementation of the Wyvill et al. method is also available in [26]. In [26] the results that are presented indicate that when subjected to the same simplification process the final output of the hexagonal and tetrahedral decompositions are virtually identical. It is also argued that implementing the tetrahedral decomposition is perhaps simpler. Additional perspectives on iso-surface

[2]Even for CT-scans of dry bone, iso-surfaces may yield models that are not anatomically faithful in regions of thin bone.

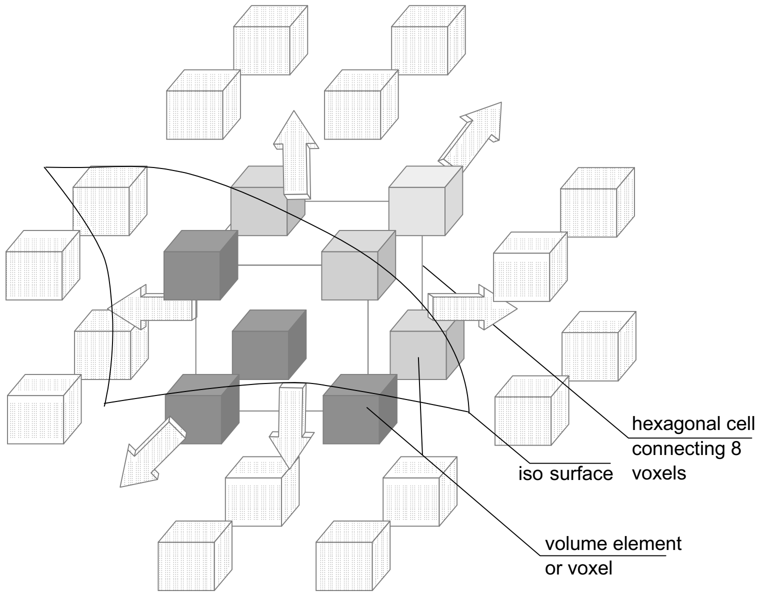



**Figure 6.3**: A hexahedral cell of a three-dimensional lattice of volume elements (voxels). A central cell is shown with 8 voxels of different intensities (function values), represented using different gray levels. The arrows indicate that the lattice extends a priori in all six directions. A portion of iso-surface is also shown. The iso-surface cuts through the lattice.

extraction can be found in [31–42].

### *6.3.2 Tetrahedral decomposition*

This section describes the method of Guéziec and Hummel [43], which is an extension of the method of Doi, Koide, et al. [44, 45]. The differences between the methods described in [43] and in [44, 45] are related to 1) the specifics of the decomposition into tetrahedra and the resulting orientation of the tetrahedra, 2) the look-up procedure for oriented triangles, and 3) the addition in [43] of a simplification process following the extraction process. The simplification process is described in detail in Section 6.6.2. The hexahedral lattice is decomposed into a tetrahedral lattice as shown in Fig. 6.4.

In addition to the examples provided in Section 6.3.6, Guéziec and Hummel's method has been applied to compare schizophrenic and normal ventricles by Dean et al. [46] and is used for surface extraction before registration in [9, 47].

The method has also been used during simulations studies by Gencer et al. on the optimal placement of electrodes for electric source imaging [48]. The effect of the simplification procedure on the accuracy of the detection of rib lines (which are discussed in detail in Section 6.3.5) has been studied by Guéziec and Dean [49].

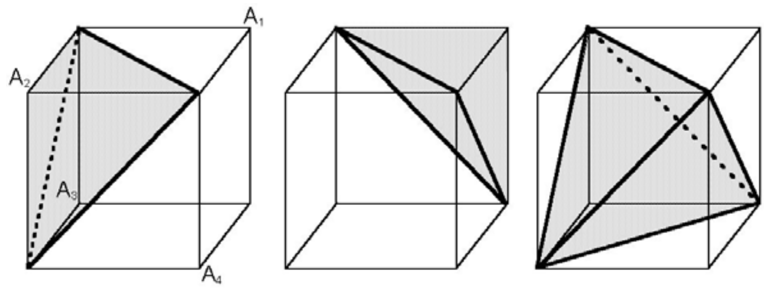



**Figure 6.4**: Triangulation of an hexahedral cell producing five tetrahedra. Four are isosceles and isomorphic to one another ($A_1, A_2, A_3$, and $A_4$ denote their apices). Two of the four are shown at the left and center. The fifth tetrahedron (right) is equilateral and occupies the center of the hexahedral cell.

### *6.3.2.1 Identifying tetrahedra vertices*

The following procedure is used to build a tetrahedron as an (oriented) quadruple of vertices. The actual implementation may either apply this procedure for every tetrahedron or apply it once and insert all possible tetrahedra (10) in a table. This table may also be built manually. The first two methods are probably less error-prone.

For any given hexahedral cell, two tetrahedral decompositions are possible. One of these is shown in Fig. 6.4; the other is mirror symmetric to the one shown. In order to be consistent between neighboring hexahedral cells, i.e., in order that faces and edges of tetrahedra in one cell match faces and edges of tetrahedra in the neighboring cells, we must alternate between the two decompositions from cell to cell, in a three-dimensional checkerboard fashion.

We next describe the use of binary operations to perform the decomposition easily. (This is not used in the Doi/Koide version.) Each of the vertices of the hexahedral cell is numbered from 0 to 7 (see Fig. 6.5). Moving along an edge of the hexahedral cell is performed by inverting one of the three bits of the vertex number. The three possible 1-bit inversions will be denoted by $\otimes$001, $\otimes$010, and $\otimes$100 (see Fig. 6.5).

Each hexahedral cell has an integer coordinate location $(i, j, k)$ for its local origin. We use $i$, $j$, and $k$ to measure the row, column, and height in the array of cells. In order to identify each tetrahedron within a cell, we perform the following steps:

1. If $i + j + k$ is even, we call the cell an *even cell*, and we say that the parity of the cell is even. If the sum is odd, the cell's parity is odd. To determine tetrahedron number 1 within the cell, we select apex $A_1$ as vertex 000 in an even cell, and as 001 in an odd cell. We then obtain three other apices from $A_1$ by applying the motion operators $\otimes$001, $\otimes$010, and $\otimes$100, resulting in $a_{11}$, $a_{12}$,

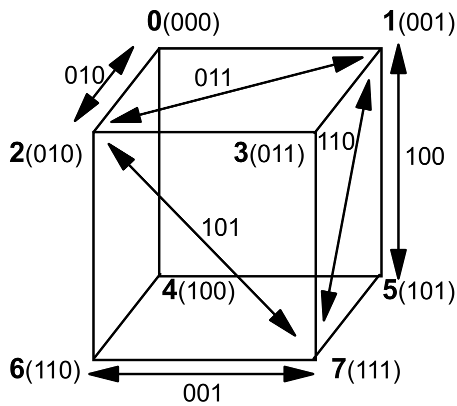



**Figure 6.5:** Some motion operators.

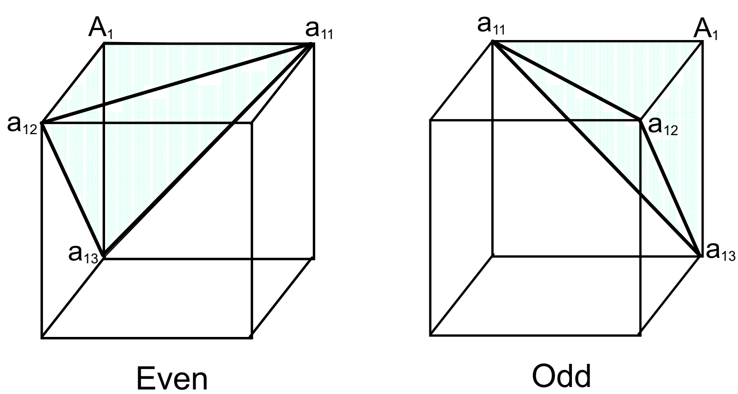



**Figure 6.6:** To determine tetrahedron number 1 within the cell, we select apex $A_1$ as vertex 000 in an even cell and as vertex 001 in an odd cell. We then obtain three other vertices from $A_1$ by applying the motion operators $\otimes$001, $\otimes$010, and $\otimes$100.

and $a_{13}$ respectively. Tetrahedron number 1 is spanned by $(A_1, a_{11}, a_{12}, a_{13})$, which we view as an ordered tuple of vertices (see Fig. 6.6).

2. Tetrahedron number 2 uses the second apex $A_2$, obtained from $A_1$ by the 2-bit inversion $\otimes$ 011. Three vertices, $a_{21}$, $a_{22}$, and $a_{23}$, are obtained from $A_2$ by the one-bit transitions $\otimes$001, $\otimes$010, and $\otimes$100.

3. Next, $A_3$ is obtained by applying $\otimes$ 101 to $A_1$. The span of $(A_3$, $a_{31}$, $a_{32}$, $a_{33})$ defines tetrahedron number 3, where the $a_{31}$, $a_{32}$, and $a_{33}$ once again were obtained from $A_3$ after applying $\otimes$001, $\otimes$010, $\otimes$100.

4. By applying $\otimes$ 110 to $A_1$, we define $A_4$, and the corresponding tetrahedron 4 is defined analogously to tetrahedra 1, 2, and 3.

5. The fifth tetrahedron is defined by shifting the four apices $(A_1$, $A_2$, $A_3$, $A_4)$ by $\otimes$001, resulting in $(A_1 \otimes$001, $A_2 \otimes$001, $A_3 \otimes$001, $A_4 \otimes$001).

### *6.3.2.2 Intersecting the iso-surface with a tetrahedron*

The next step consists in determining whether a portion of the iso-surface will intersect a given tetrahedron $(a_1, a_2, a_3, a_4)$. The voxel values $(v_1, v_2, v_3, v_4)$ corresponding to the four tetrahedron apices are retrieved from the three-dimensional lattice. For each tetrahedron edge that exhibits an intensity sign change, a vertex of the polygonal approximation of the iso-surface $v = 0$ is created. The exact position of the vertex is determined by the zero-crossing of a function interpolating voxel values along the edge. Because of the issue illustrated in Fig. 6.7, we use linear interpolation on an edge of the hexahedral cell and bilinear interpolation on a diagonal edge (which amounts to using bilinear interpolation overall). Specifically, if $(a, b, c, d)$ denote the four intensity values on the face of an 8-cell, then the intensity value along the $(a, d)$ diagonal edge is given by:

$$v(u) = (a + d - b - c)u^2 + (-2a + b + c)u + a\,.$$

Here, $u$ varies from 0 to 1 linearly along the diagonal. Provided $ad < 0$, $v$ will have exactly one zero in the range $0 < u < 1$, which can easily be determined from the quadratic formula (the other zero will fall outside this range).

### *6.3.2.3 Building a set of oriented triangles*

Consider a tetrahedron $(a_1, a_2, a_3, a_4)$ that is defined by an ordered tuple of apices. The order is as determined by one of the five steps from Section 6.3.2.1. The corresponding intensity values are given by a four-tuple $(v_1, v_2, v_3, v_4)$. If all four values have the same sign, then the surface does not intersect the tetrahedron. If the signs are mixed, however, we then have three major cases. Among $v_1, v_2, v_3, v_4$, there are either one, two, or three positive values. These cases, which we call Cases

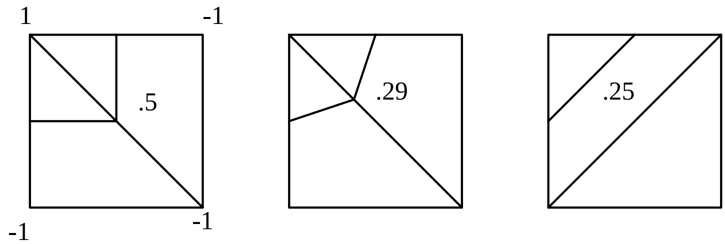



**Figure 6.7**: Using a linear interpolation along the diagonal edge results in a severe difference in position for the polygonal surface when the diagonal edge is swapped (left and right), assuming a square face. Instead, we use a *bilinear* interpolant (middle). The numbers .5, .29, and .25 indicate the relative position of the iso-surface, where it intersects the top-left-to-bottom-right diagonal.

I, II, and III, are illustrated in Fig. 6.8. For the Cases I and III, three of the values have the same sign. For Case II, two vertices are positive and two are negative. In this case, the surface will intersect all four faces, and we have a quadrilateral. By choosing arbitrarily a diagonal of the quadrilateral, we obtain two triangles within the tetrahedron. Combining all cases, we have a patch of the surface represented as either one or two triangles. The triangles can be oriented.

**Cases I and III** For Cases I and III, exactly one value among $(v_1,\ v_2,\ v_3,\ v_4)$ has sign opposite to the others. Using this value, we compute intersections along the edges connecting its vertex to the other three vertices, preserving order among $(a_1, a_2, a_3, a_4)$. So, for example, if $v_3$ has the different sign, we compute intersection along $(a_3, a_1)$, then $(a_3, a_2)$, and then $(a_3, a_4)$. These three intersections determine an ordering which is either the correct direction, or opposite to the correct direction. One way to determine if the orientation is correct is to check it by examining the sign of a determinant, as done by Doi/Koide, as follows.

First, we reorder the vertices $(a_1, a_2, a_3, a_4)$ to obtain $(\alpha, \beta, \gamma, \delta)$ such that the vertices with negative values precede the vertices with positive values, without otherwise disturbing relative order. Viewing the vertices as coordinates in three-space, we can then compute the determinant

$$\det(\beta - \alpha, \gamma - \beta, \delta - \gamma)\,.$$

If this determinant is negative, then the ordering of the triangle vertices is correct; otherwise, the ordering must be reversed. The procedure can be verified by carefully considering the cross product $(\beta - \alpha) \times (\gamma - \beta)$ (see Fig. 6.8).

**Case II** In this case, exactly two vertices are negative, and two are positive. We reorder the vertices $(a_1, a_2, a_3, a_4)$ as above, to give vertices $(\alpha, \beta, \gamma, \delta)$, where the

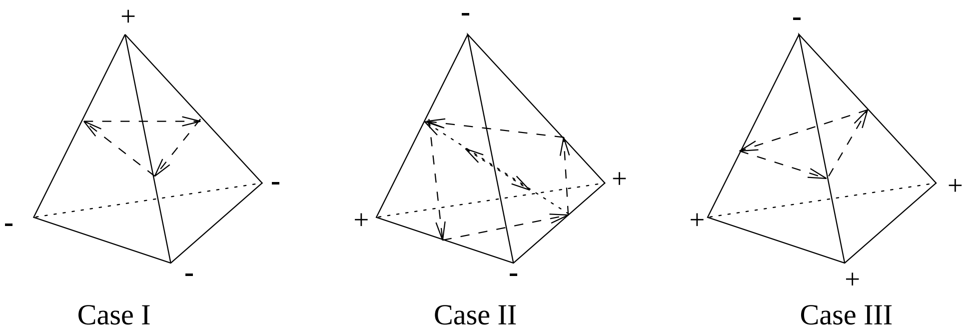



**Figure 6.8**: Defining one or two oriented triangles. Depending upon the number of vertices outside the solid (marked with the minus sign) we define three or four vertices of the surface approximation in the ordering specified in the text. The condition for having to reverse the ordering is $\det(\beta-\alpha,\gamma-\beta,\delta-\gamma)>0$. Note that this value is six times the *volume*, positive or negative, of the tetrahedron.

values associated with $\alpha$ and $\beta$ are negative, and the values associated with $\gamma$ and $\delta$ are positive. We compute the vertices of the quadrilateral as follows. We first find the zero along $(\alpha,\gamma)$. Next, we find the zero along $(\alpha,\delta)$, then $(\beta,\delta)$, and finally $(\beta,\gamma)$. This sequence of four points establishes a cycle, which is either the correct ordering, or the incorrect ordering, which can be easily checked.

In this case, the verification procedure is exactly the same as in the previous case. That is, viewing the vertices as vectors in $\Re^3$, we compute $\det(\beta-\alpha,\gamma-\beta,\delta-\gamma)$. The ordering of the interpolation points defined above is correct if the determinant is negative; if the determinant is positive, then the order should be reversed.

If the resulting ordering of the interpolation points is given by $(w_1,w_2,w_3,w_4)$, then there are two possible triangulations: one is given by the following pair of ordered triangles $\{(w_1,w_2,w_4),\ (w_2,w_3,w_4)\}$, and the other is given by the pair $\{(w_1,w_2,w_3),\ (w_3,w_4,w_1)\}$. The triangulations may be chosen arbitrarily. However, by always choosing the first one in an even cell, and the second one in an odd cell, the degree of vertices in the triangulation can be shown to be always less than or equal to nine [43].

### *6.3.3 A look-up procedure to replace the determinant test*

In fact, it is not necessary to compute the determinant value $\det(\beta-\alpha,\gamma-\beta,\delta-\gamma)$ to know its sign. By applying the procedure that follows instead of computing determinants, a 2.5-fold speed-up was recorded in [43].

Let us first suppose that for a given tetrahedron $(a_1,a_2,a_3,a_4)$ in an even cell, no reordering is required to obtain $(\alpha,\beta,\gamma,\delta)$. It then turns out, based on the construction of the ordered tuple representing the tetrahedron, that $\det(\beta-\alpha,\gamma-\beta,\delta-\gamma)$ will be positive, regardless of whether $(a_1,a_2,a_3,a_4)$ is a tetrahedron number 1, 2, 3, 4, or 5, as long as we are inside an even cell. (For an odd cell, the result is always negative.) Next, we suppose that some reordering is required in determining $(\alpha, \beta, \gamma, \delta)$. Recall that vertices with negative values are listed first, without changing other relative orderings. Suppose, for example, that $a_2$ and $a_1$ must be exchanged to obtain $(\alpha, \beta, \gamma, \delta)$. The result of this permutation is that the sign of the determinant is reversed. If instead $a_3$ must be brought to the front, then two transpositions are required, and the sign of the determinant stays the same.

In general, the sign of the determinant $\det(\beta - \alpha, \gamma - \beta, \delta - \gamma)$ is determined by the number of transpositions required to permute the negative vertices among $(a_1, a_2, a_3, a_4)$ to the beginning of the list. The number of transpositions required can be pretabulated, as follows. Representing a negative vertex in $(a_1, a_2, a_3, a_4)$ by a 0, and a positive vertex by a 1, we obtain a 4-bit code for the tetrahedron, yielding a number between 0 and 15. The resulting determinant will be positive or negative according to whether an even or odd number of transpositions are required, as tabulated in Table 6.1. Note that it is no longer required to physically reorder the vertices—only the bit code is needed. Accordingly, if the table yields a positive entry for the determinant, the oriented cycle representing the surface patch should be reversed.

**Table 6.1**: Mapping from the bit code representing the signs of the values at the tetrahedra vertices in an even cell to the sign of the determinant that determines whether a triangle is correctly oriented (negative determinant) or must be reversed (positive determinant). For an odd cell, the entries are opposites of those of an even cell.

| Bit code | Index | Sign of det() |
|---|---|---|
| 0000 | 0 | $+$ |
| 0001 | 1 | $+$ |
| 0010 | 2 | $-$ |
| 0011 | 3 | $+$ |
| 0100 | 4 | $+$ |
| 0101 | 5 | $-$ |
| 0110 | 6 | $+$ |
| 0111 | 7 | $+$ |
| 1000 | 8 | $-$ |
| 1001 | 9 | $+$ |
| 1010 | 10 | $-$ |
| 1011 | 11 | $-$ |
| 1100 | 12 | $+$ |
| 1101 | 13 | $+$ |
| 1110 | 14 | $-$ |
| 1111 | 15 | $+$ |

**Special Cases** Zero-area polygons can easily be avoided in the tetrahedral decomposition method as shown in this section. Such degenerate polygons occur when $v = 0$ at some or all of the positive vertices. (Recall that $v \geq 0$ corresponds to a positive vertex.) When $v = v_0 = 0$ occurs, a single vertex $v = 0$ within a tetrahedron is handled exactly as any positive vertex. Indeed, we treat a vertex with $v = 0$ exactly as any other positive vertex except for two special cases: (1) If there is one vertex with $v = 0$, and the other three are strictly negative, then we do not create a triangle, even though we are in a Case I situation; (2) Also, if there are two vertices with $v = 0$, and the other two vertices are strictly negative, then we do not create polygons, even though we have a Case II situation. In these two cases, the polygons are degenerate. On the other hand, if three values among $(v_1, v_2, v_3, v_4)$ are zero and the other is negative, then we create one triangle, exactly as a Case III situation would predict. Also, contrary to [44], when $v_1 = v_2 = v_3 = v_4 = 0$, we do not create any polygons, instead viewing this situation as four positive vertices lying inside the iso-surface.

This completes the description of the tetrahedral decomposition method. Once the surface has been extracted, it is possible to define memory-efficient data structures based on tetrahedra to store vertices and triangles. This is described in [43].

### *6.3.4 Computing surface curvatures*

Monga *et al.* [50] provide a method for computing principal curvature directions and curvature values of an iso-surface directly from voxel data and spatial derivatives of the voxel data [3]. Approaches based on triangular patches, rather than voxel data, are given in [52, 53] and [54]. If a differentiable explicit representation is available (e.g., using spline tensor products), the method of [18] may also be used.

In order to evaluate the surface curvatures using volume data, we consider a curve $C$, with tangent vector $\mathbf{t}$, along a *normal section* of the iso-surface $S$ [4]. Since $C$ is contained in a plane that contains the surface normal, $\mathbf{n}$, the normal of $C$ is also $\mathbf{n} = \nabla v / \|\nabla v\|$, where $v$ designates a continuous function modeling volume (voxel) data. The differentiation of $\nabla v \cdot \mathbf{t} = \mathbf{0}$ in the direction of $\mathbf{t}$ yields

$$\mathbf{t}^t H \mathbf{t} + \nabla v \cdot k\mathbf{n} = 0$$

where $k$ denotes the normal curvature of $C$, which is also by definition the surface curvature in direction $\mathbf{t}$, and $H(v)$ is the Hessian:

$$H(v) = \begin{pmatrix} v_{xx} & v_{xy} & v_{xz} \\ v_{xy} & v_{yy} & v_{yz} \\ v_{xz} & v_{yz} & v_{zz} \end{pmatrix} .$$

[3] See also [51].

[4] A normal section of a surface (through a vertex of the surface) is a section of the surface cut by a plane that contains the vertex normal. There are an infinity of normal sections at a given surface vertex.

Thus $k = -\mathbf{t}^t H\mathbf{t}/||\nabla v||$. The *principal curvatures* are obtained by maximizing and minimizing $k$ over possible tangent directions $\mathbf{t}$. Accordingly, the principal curvatures are the negatives of the eigenvalues of $P^tTHTP$ divided by the magnitude of the gradient $||\nabla v||$, where $P$ is the projection of $(x, y, z)$ onto $(x, y)$ and is a three-by-two matrix:

$$P = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$$

and $T$ is the Householder transformation that maps $\mathbf{n}$ to $\mathbf{z} = (0, 0, 1)$. We first define $\mathbf{u} = (\mathbf{n} - \mathbf{z})/||\mathbf{n} - \mathbf{z}||$, and then $T = \mathbf{I} - 2\mathbf{u}\mathbf{u}^t$, where $\mathbf{I}$ is the identity matrix. The *principal directions* $\mathbf{a}$ and $\mathbf{b}$ are the tangent directions for which the minimum and maximum curvatures are obtained.

All vertices of surface patches occur along edges of hexahedral cells, and derivatives of the data $v$ may be computed and interpolated at these locations by suitable methods. Solving the eigenproblem at the interpolated position of each surface vertex, we may attach curvatures (and also principal curvature directions, after applying the inverse Householder transformation) to each vertex of the iso-surface.

The above method was used to compute surface curvatures that were colorized with a suitable color map to produce some of the illustrations in Section 6.3.6.

### 6.3.5 Extracting rib (or ridge, or crest) lines

Rib (or ridge, or crest) lines of a surface are the loci of the surface where the principal curvatures reach a local extremum. Rib lines have been studied in medical imaging by Bookstein and Cutting [55, 56], the Epidaure team at INRIA [51, 57–61], as well as Dean et al. [46, 49]. While Epidaure's main application has been to extract (few) salient and robust features for registering 3D image data sets, Cutting, Bookstein, Dean, et al. have been pursuing the goal of using rib lines, combined with other networks of (geodesic, or minimum-distance) lines, to build templates of the anatomy, particularly of the face, skull, or ventricles.

Several characterizations of rib lines have been proposed. We refer to the above-referenced publications as well as textbooks on the geometry of surfaces [62–64]. Referring to Section 6.3.4 above, rib lines may be related to ribs (discontinuity curves) of the *focal surfaces* of a surface, which are the loci of the *centers of curvature* of the surface[5]. There is one center for the minimum principal curvature, another center for the maximum principal curvature, and hence two focal surfaces, except at an *umbilic point*. Ribs may thus be colored differently, depending on which focal surface they relate to [64].

In this section, we use and illustrate Markatis's characterization of a rib line

[5]A center of curvature is located along the normal, at an offset equal to the radius of curvature, which is the inverse of the curvature.

based upon focal surfaces [65][6]. Markatis's equation applies for a surface defined using an implicit equation, which is the case for an iso-surface (the implicit function being known at the voxels only). Markatis's characterization is also discussed on p.197 of the book by Porteous [64]. According to Markatis's equation, a surface point belongs to a rib if either of the following relations holds:

$$(F_1(v)\mathbf{n})(F_3(v)\mathbf{a}^3) - 3(F_2(v)\mathbf{an})(F_2(v)\mathbf{a}^2) \quad = \quad 0 \qquad (6.1)$$

$$(F_1(v)\mathbf{n})(F_3(v)\mathbf{b}^3) - 3(F_2(v)\mathbf{bn})(F_2(v)\mathbf{b}^2) \quad = \quad 0\,. \qquad (6.2)$$

The notations in Eqs. (6.1, 6.2) should be interpreted as follows: $F_1(v) = \nabla v$, $F_2(v) = H(v)$ and $F_3(v)$ is a matrix of third order derivatives of the intensity (with 10 different entries). $\mathbf{a}$ and $\mathbf{b}$ are the principal directions, as previously discussed in Section 6.3.4.

Equations (6.1) and (6.2) are implicit equations that define another set of iso-surfaces in the volume data. The intersection between these surfaces and the first surface corresponds in general to a set of lines. To extract these lines, one possibility is to use the Marching Lines method developed by Thirion and Gourdon [60]. Another possibility is to extend the method of Section 6.3.2 as follows. For each triangle of the iso-surface, the three vertices of the triangle are tested against Eqs. (6.1) and (6.2). If a change in sign is observed, then a segment of a rib line is created. Such segments may then be linked together, forming curves. We show a result of the latter approach in Fig. 6.9, where we illustrate ribs on an iso-surface of the cortex. The ribs appear to follow the centerlines of the sulci.

### *6.3.6 Iso-surface examples*

The first example in Fig. 6.10 is a cortex model composed of 362,000 triangles (courtesy of Henry Rusinek and Gregoire Malandain, who performed the segmentation [66]) that was constructed in 59 seconds on an IBM RS6000 computer using the tetrahedral decomposition method. Surface simplification (see [43] and Section 6.6.2) reduced the number of triangles from 362,000 to 52,000. The result is shown in Fig. 6.10(b). Brain sulci appear in red, and gyri appear in green and blue, as color coded relative to the magnitude of the largest principal curvature.

The next example is a surface model obtained from a CT scan of a cranium from the Cleveland Museum of Natural History Collection (Courtesy of Bruce Latimer, Court Cutting, and David Dean). The following surface model was obtained using the tetrahedral decomposition method followed by surface simplification. The model, shown in Fig. 6.11, comprises 66,000 triangles and 129,000 triangles, simplified from an original iso-surface containing 3,450,000 triangles.

[6] Markatis's ridge equation was brought to the author's attention through personal communication with I. Porteous, from the University of Liverpool.

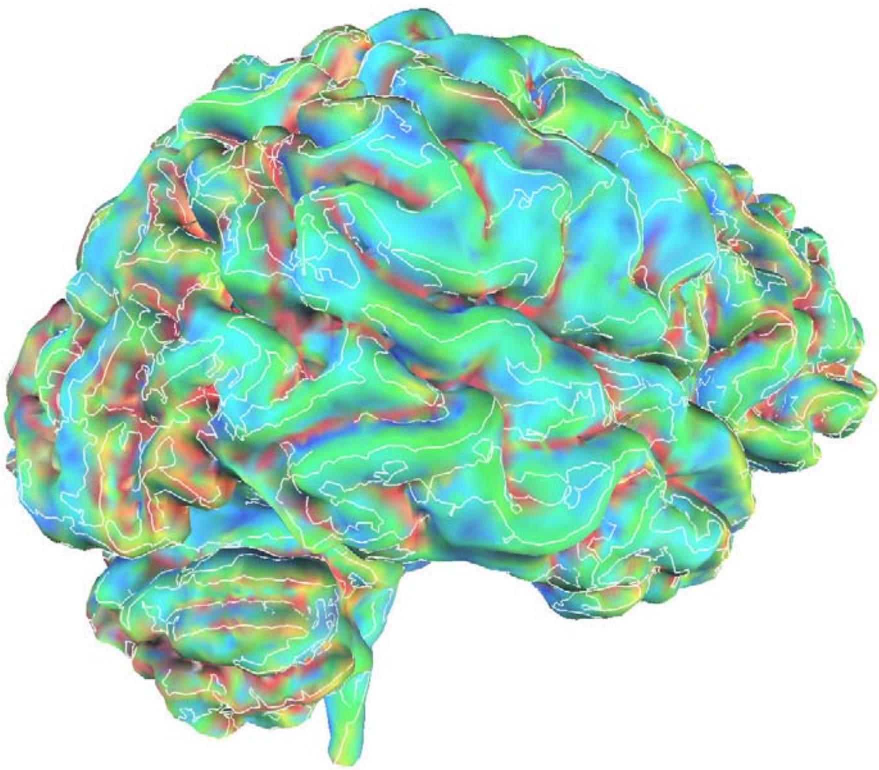


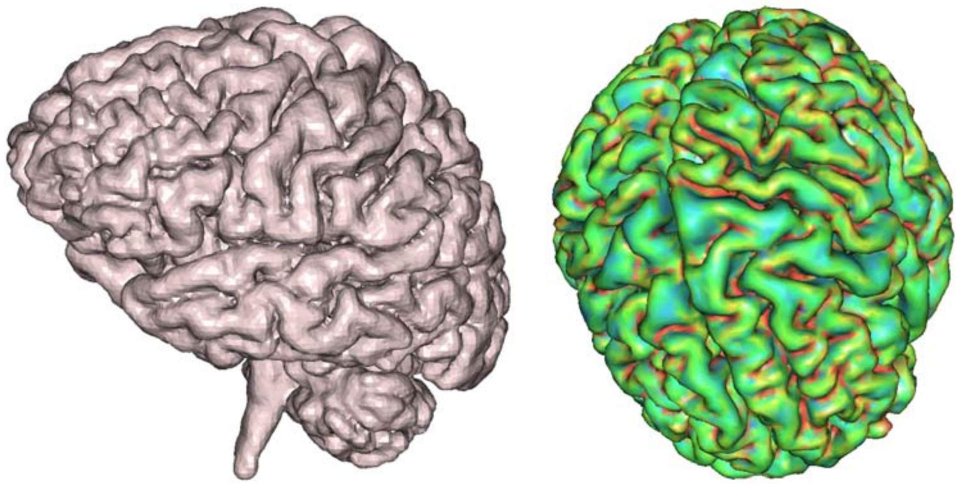
**Figure 6.9**: Ribs (in white) drawn on an iso-surface of a cortex segmented in an MRI image. The ribs were characterized using Markatis's equation. The ribs appear to follow the centerlines of the sulci.

## 6.4 Building surfaces from two-dimensional contours

As explained before, direct extraction of anatomical surfaces as iso-surfaces using the method explained above may fail to extract an accurate model of the anatomy. Another, perhaps more classical, approach for extracting a surface model from image slice data is to first extract two-dimensional contours from each relevant image slice and then tile the two-dimensional contours together to form a surface [67–71].

In the present section, we illustrate this situation with a concrete example: the extraction of a three-dimensional model of the femur from CT-scan data for surgical planning and intra operative registration in Total Hip Replacement (THR) surgery [72–74].

In the CT scan data that is clinically acquired for THR [75], the slice spacing typically varies significantly, from 1 mm in the vicinity of the fiducial markers to 6 mm or more in the femur shaft in order to maximize the detail in the critical areas



(a) (b)

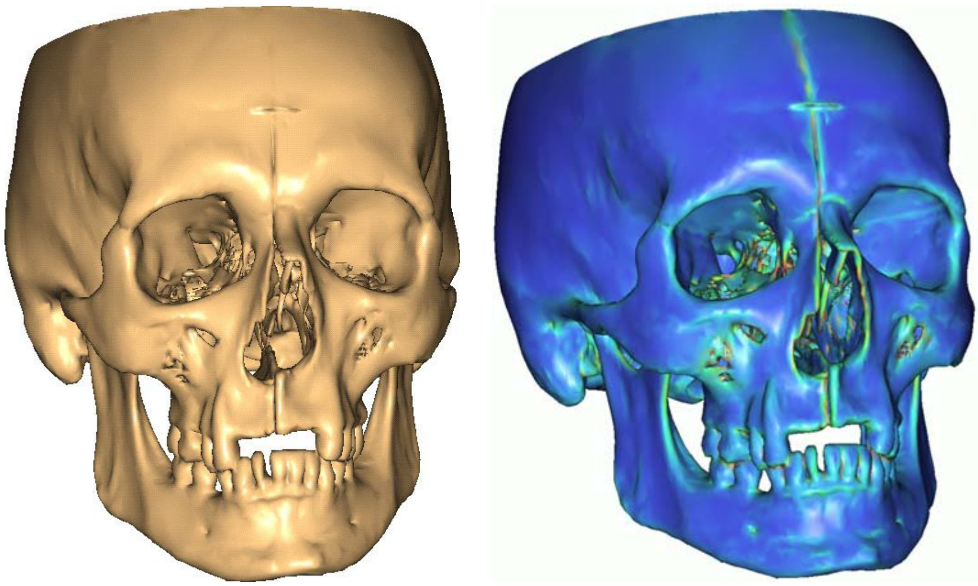
**Figure 6.10**: Example of iso-surface extraction and curvature computation. (a) cortex model represented as an iso-surface (362,000 triangles). (b) simplified cortex (52,000 triangles); the vertices are color-coded (with Gouraud shading) [8] according to the magnitude of the largest surface principal curvature. (For a color version of this Figure see Plate 1 in the color section of this book.)

while limiting the x-ray dosage. Direct extraction of bone surfaces from such CT data using software that builds an iso-surface produces "staircase" artifacts: bone contours are located several voxels apart from slice to slice, and the gap is modeled by the iso-surfacing method as a flat surface portion parallel to the slices. Aside from the problems created by the irregular spacing between slices, bone has very different Hounsfield numbers in the CT data because of varying bone densities (within the same femur bone). Selecting a different iso-value for each slice may not be sufficient to obtain a correct segmentation of each slice. This problem is illustrated in Fig. 6.12.

### *6.4.1 Extracting two-dimensional contours from an image*

The extraction of relevant two-dimensional contours from each slice may be automated or performed by an operator or semi automated (and operator assisted). This operation is part of a segmentation process and is described more completely in Chapters 2 and 3.

A solution that performs well in practice on the data of our THR example operates as follows [72]. For each slice, we use a deformable model technique to detect the two-dimensional contour of the bone. We implement the technique of Kass et al. [76] with some modifications as explained next. In [76] an "energy" is defined that incorporates the stretching and bending energies of the contour as well



(a) (b)

**Figure 6.11**: Iso-surface extracted from a CT-scan image. (a) model of cranium extracted as an iso-surface and simplified. (b) the surface is color-coded (with Gouraud shading) according to the curvature as in Fig. 6.10. (For a color version of this Figure see Plate 2 in the color section of this book.)

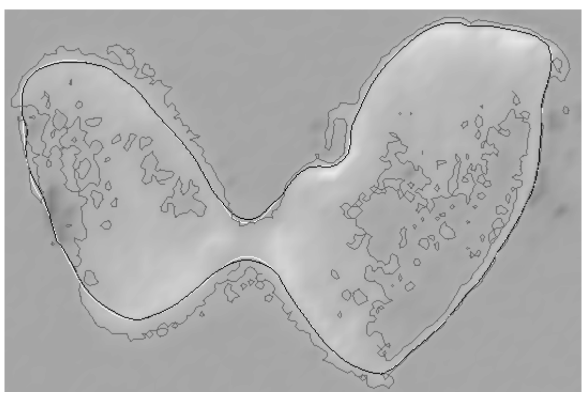


**Figure 6.12**: Because of bone density variations, iso-contouring methods may produce inadequate results, even if the iso-value is chosen differently for each slice. We show here a CT slice taken in the distal femur region (region of the condyles) along with a best attempt (yet widely unsuccessful) at segmenting the bone using iso-contours (gray almost-fractal-like curves). A segmentation using our deformable model implementation is shown using the white curve (another gray curve next to the white curve represents a preliminary result obtained using a smoothed low-resolution version of the image).

as a potential field that measures closeness to the data; a two-dimensional contour modeling of the bone is obtained at a minimum of the energy. The contour is then modeled with a series of points, and the energy is minimized by solving a partial differential equation using a finite difference method.

In our implementation we have chosen a smoothed image gradient norm for the potential field. The user is asked to select a few points in the vicinity of the structures of interest. The system then connects such points by a polygon, samples new points on that polygon, and uses such sampled points as an initial estimate for the deformable model (Fig. 6.13). As discussed in Chapter 3, a considerable research effort was recently devoted to extend [76] for attracting the contours from a long range and thus automating the process better[9]. To define a longer range potential, we have implemented a two-step process, whereby a first deformable model is attracted by a smoothed low-resolution version of the potential; the result is then used as an initialization for a second deformable model that is attracted by the full resolution potential. This process was particularly useful to reuse the same initial points from slice to slice and limit the user input. Such reuse of the input points is shown in Fig. 6.13. In this figure, the white curve model corresponds to the first iteration of the deformable model, operating at a coarse scale, while the black curve corresponds to the second iteration of the deformable model, operating at a fine scale.

### *6.4.2 Tiling contours into a surface portion*

The problem of tiling two-dimensional contours extracted from two parallel[10] slices can be quite complex. There are essentially two cases: (1) the correspondence between two contours is established and corresponding vertices for the beginning and end of contours are determined; (2) the correspondence must yet be determined.

In order to solve either problem both volume-based and surface-based approaches have been proposed. In volume-based approaches [3], contours are scan-converted to binary images. Intermediate (in-between) binary images may be interpolated to build a volume using various interpolation methods, including shape-based interpolation [77]. Shape-based interpolation methods use distance transforms [78] applied to binary images to assign grey levels to intermediate slices. Resulting grey-level images are then thresholded. A surface portion can then be extracted from the volume data that was built by the interpolation process using iso-surface extraction methods (see Section 6.3). While to some extent, the process of back converting to pixels/voxels a contour that was previously extracted from pixel/voxel data is wasteful, volume-based methods are guaranteed to produce a valid surface.

[9]Another avenue of research seeks to automate changes of topology. We briefly study some surface modeling aspects of this issue in Section 6.5.

[10]The methods could be extended to nonparallel slices, but they don't seem to occur in practice.

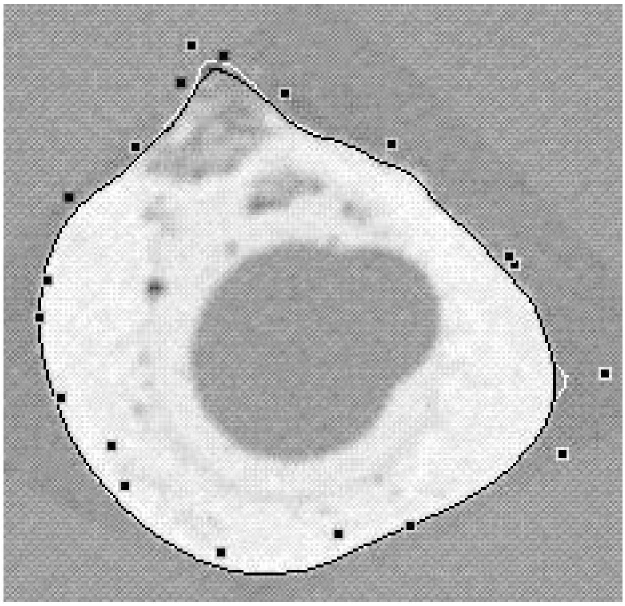


**Figure 6.13**: Using deformable contours for surface extraction. A hierarchical deformable contour can cope with incorrect user input (e.g., rightmost point marked with a square), or reuse such input for multiple slices.

Surface-based (or polygon-based) methods concentrate on determining a set of polygons or triangles that span the vertices of contours extracted from the slices. Several approaches address the general problem (2) associated with complex correspondence and branching situations between contours [69, 70, 79–82]. Boissonnat and Geiger's approaches [70, 79] use the notion of proximity, formalized using Voronoi diagrams, to decide which vertices to connect with triangles. Meyers et al. [69] represent (approximate) contours with ellipsoids and objects as cylinders connected with a graph; this representation is used to infer branchings. Bajaj et al. [82] impose further constraints on the problem of shape reconstruction and use them to derive precise correspondence and tiling rules.

In several respects, the correspondence and (particularly) branching problems are still not completely solved at the time this chapter is written: with some particular configurations of contours, even sophisticated algorithms may still fail to produce a suitable tiling.

One major argument in favor of surface-based models has traditionally been that these methods allow significant data reduction [70]: contours may be individually approximated using fewer polygonal edges, and triangles/polygons are built only to span such vertices. In contrast, when using volume based methods, contours are discretized using small voxels, and an iso-surface will likely produce a large collection of small triangles/polygons with a size comparable to a voxel. However, since [70] was published, there has been considerable interest and progress in polygonal surface simplification methods. This is studied in Section 6.6.2. Iso-surfaces can be simplified as a post process with bounds on the computational burden as well as the resulting surface deviation (degradation); iso-surface extraction followed by simplification has been demonstrated as an effective tool for real-world (clinical) applications [3]. The examples of Section 6.3.6 illustrate simplified iso-surfaces. For surfaces with complex topologies that represent a challenge for surface-based methods, volume-based methods, combined with data reduction, offer a guarantee of success, perhaps at a higher computational cost.

#### *6.4.2.1 Tesselating when the correspondence is known*

For many data-sets encountered in practice, the topological changes between adjacent contours can be resolved without too much difficulty, and the remaining challenge is to tesselate the contours. We next illustrate this process with a concrete example.

Returning to our example involving Total Hip Replacement surgery data, the full generality of the above methods is not necessary for most of the femur length, as contours are in one-to-one correspondence. Branching contours exist at the head of the femur and condyles.

When the correspondence between two contours is obvious, we may use a variation of a method described in [67, 83] to build a piece of surface connecting the two contours. We define a measure of surface quality as the sum of the measures of each individual triangle quality. As in Keppel [83], we can then build a set of triangles that connects the two contours and corresponds to a true optimum of the surface quality measure. Typical criteria that have been proposed so far include minimizing surface area or maximizing the volume enclosed by the surface portion between the two slices. The exact volume enclosed by a triangular mesh without boundary can be easily computed as a sum of volumes of tetrahedra spanned by an arbitrary origin and the triangles of the triangular mesh. After a coordinate transformation, we can assume that the origin has coordinates $(0, 0, 0)$. The volume $V$ is given by the following sum, wherein $\Delta(i_0, i_1, i_2)$ designates a triangle of the mesh ($i_0, i_1$, and $i_2$ are indices of the three vertices of the triangle having label $i$). Only the triangles of the surface connecting the contours contribute to the sum since the triangles capping the surface above and below cannot be modified.

$$V = \frac{1}{6} \sum_{\Delta(i_0,i_1,i_2)} \begin{vmatrix} x_{i_0} & y_{i_0} & z_{i_0} & 1 \\ x_{i_1} & y_{i_1} & z_{i_1} & 1 \\ x_{i_2} & y_{i_2} & z_{i_2} & 1 \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{6.3}$$

$$= \frac{1}{6} \sum_{\Delta(i_0,i_1,i_2)} \begin{vmatrix} x_{i_0} & y_{i_0} & z_{i_0} \\ x_{i_1} & y_{i_1} & z_{i_1} \\ x_{i_2} & y_{i_2} & z_{i_2} \end{vmatrix} \tag{6.4}$$

(a) (b)

**Figure 6.14**: Femur models obtained by tiling contours. The models have been simplified as explained in Section 6.6.2. (a) Proximal femur (1,664 triangles). (b) distal femur (4,199 triangles).

To produce the pictures shown in Fig. 6.14, we have maximized the sum of triangle *compactnesses* [84]. In [84], the compactness $c$ of a triangle is defined with the following formula:

$$c = \frac{4\sqrt{3}a}{l_0^2 + l_1^2 + l_2^2}\,. \tag{6.5}$$

where $a$ is the positive area of the triangle and $(l_0, l_1, l_2)$ are the lengths of the three sides. This formula defines a dimensionless measure similar to the area-to-perimeter ratio; this measure can be computed inexpensively, without evaluating square roots (except $\sqrt{3}$, which is precomputed or read from a table).

Following [67, 83], we consider a rectangular graph (grid) whose $n \times m$ nodes represent edges between $n$ and $m$ vertices in the respective adjacent contours. Edge $(1,1)$ (which is the same as $(m-1, n-1)$) is chosen as a closest point pair between the contours. A path in the graph from node $(0,0)$ to node $(m-1, n-1)$ represents a surface connecting the two contours. This graph is shown in Fig. 6.15(a). Starting with Edge 0–0, we may decide to construct a triangle by connecting the edge with with the next vertex on the top contour (with $m$ vertices), which is represented in the graph by drawing a horizontal arrow starting from vertex (0,0) and pointing left. Alternatively, we may decide to connect Edge 0–0 with the next vertex on the bottom contour (with $n$ vertices), which is represented in the graph by drawing a vertical arrow starting from vertex (0,0) and pointing down. Starting from the graph node pointed by the arrow (this node represents an edge in the tesselation,

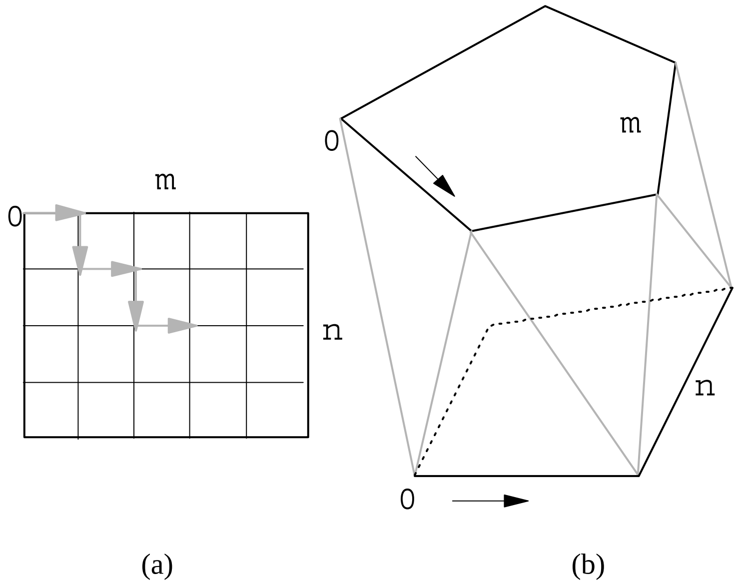


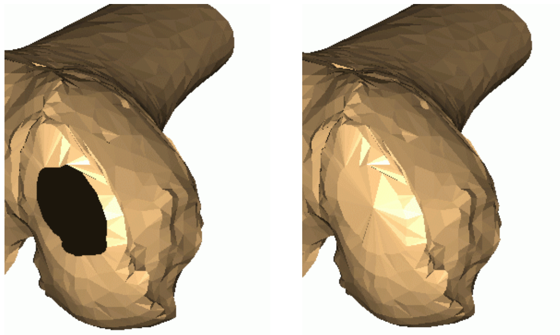

**Figure 6.15**: Tesselating two closed contours. When the correspondence between two contours comprising $m$ and $n$ vertices is known, and a first edge connecting the two contour is chosen (Edge 0–0), an optimum of a user-defined surface quality criterion is built using dynamic programming. (a) rectangular graph whose $n \times m$ nodes represent edges between $n$ and $m$ vertices in the respective adjacent contours. Each node has a cost associated, representative of an optimum path from (0,0) to that node; (b) edges drawn in gray map the gray path in A (see text).

either Edge 1–0 or Edge 0–1), we may again decide to connect with a vertex from either the top or the bottom contours. Both choices may be recorded in the graph. By examining carefully Fig. 6.15, the reader will be able to convince him/herself that all possible tesselations starting with Edge 0–0 may be represented as a path in the toroidal graph of Fig. 6.15(a).

For each triangle, we compute a cost function (area, volume, or else) as determined by the user. The cost of a particular tesselation is obtained by adding the costs of all triangles. The costs for all possible paths may be computed by filling an $m$ by $n$ table, wherein each table entry corresponds to a vertex of the graph of Fig. 6.15(a), and represents the cost of the optimum path leading to that vertex. The cost of Entry $(m-1, n-1)$ thus represents the cost of the best tesselation. This process follows a general method called *dynamic programming* [85].

### *6.4.2.2 Simple branching and "capping"*

We next illustrate with a concrete example how simple branching situations may be resolved. In our exemplary data-set, branching situations have been limited



(a) (b)

**Figure 6.16**: Capping of a distal femur model by extrapolating areas and centroids of a few contours: (a) before; (b) after.

to one "primary" contour branching to two "secondary" contours. We may treat this common situation using a particular case of the method of [70] as follows. We compute a Voronoi diagram [86] of the points of the two secondary contours, retaining only the edges of the diagram that bisect a pair of vertices from different contours. We then project this reduced diagram on the plane supporting the primary contour. It would generally intersect the primary contour in two locations that, together with segments falling inside the primary contour, may be used to split the contour into two portions; each portion may be tiled separately to one secondary contour.

CT slices generally fail to capture the exact location where an object starts or ends. This is the case for a CT scan of the femur in the vicinity of the condyles (the femur is visible in one slice and invisible in the next slice). In [87] we developed an original method to "cap" the surface: using a few contours, we fit an ellipsoid of revolution to the data formed by the contour areas and centroids. A better fit is generally obtained by allowing the ellipsoid's main axis to be tilted with respect to the scan's main axis. The fitted model then predicts that the equation of the area $A$ of the intersection between the ellipsoid and the slice is a parabola as a function of the slice's $z$ coordinate: $A = \alpha z^2 + \beta z + \gamma$, with $\alpha < 0$ and $\gamma > \frac{\beta^2}{4\alpha}$ (the area must decrease from a positive value). After fitting $\alpha, \beta, \gamma$ to the data, at the slice $z$ coordinate that predicts an area of zero, we create a new surface vertex ($x$ and $y$ coordinates may be predicted with the centroid data) and link it with triangles to all edges of the last contour: the surface is thus capped. This operation is illustrated in Fig. 6.16.

## 6.5 Some topological issues in deformable surfaces

In previous sections we provided the complete details of methods for extracting surfaces from segmented volume data (Section 6.3) and from a collection of contours that were segmented from two-dimensional image data (Section 6.4). We thus assumed that the processes of segmentation and surface extraction were decoupled. This is often the case, reflecting the organization of the present handbook comprising separate chapters for segmentation (Chapter 2) and surface extraction (present chapter).

In several situations, however, it is useful to combine the two processes. Three-dimensional deformable models, which are covered in a more comprehensive fashion in Chapter 3, allow us to do just that. With deformable surface models the characteristics of the surface (continuity, curvature) are used to guide the segmentation process by bridging areas where the image data is ambiguous, or by ignoring spurious data. In this section we study how the type of representation selected for the deformable surface (see Section 6.2) affects the result and particularly its topology.

The first method that we present uses tensor-product B-splines for representing the surface with a fixed topology. It is described with more detail in [18].

### 6.5.1 Tensor-product B-splines

B-spline *basis functions* are piecewise polynomials with a finite support and a recursive definition. The spline basis functions of degree zero are the characteristic functions of the variable $u \in \mathbb{R}$ for the intervals between real values $\{u_j\}$ called *knots*:

$$B_{j,1}(u) = \begin{cases} 1 & u_j \leq u < u_{j+1} \\ 0 & \text{otherwise} \end{cases} .$$

Higher-order splines are formed by convolving lower-order splines:

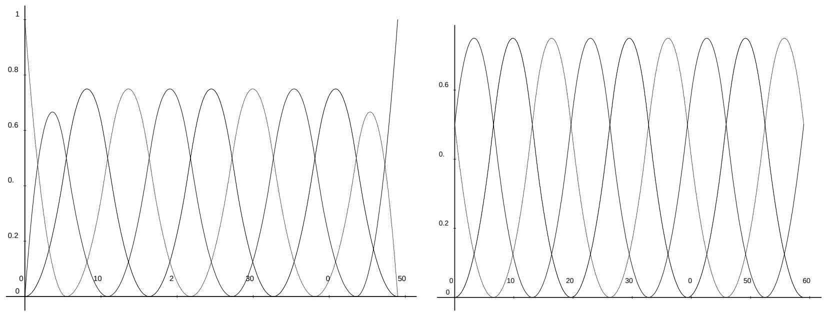
$$B_{j,K+1}(u) = \frac{u - u_j}{u_{j+K} - u_j} B_{j,K}(u) + \frac{u_{j+K+1} - u}{u_{j+K+1} - u_{j+1}} B_{j+1,K}(u) . \tag{6.6}$$

The $B_{j,K+1}$ functions have degree $K$. They are globally $\mathcal{C}^{K-1}$. The evaluation of (6.6) is especially efficient, i.e., can be computed with divided differences, and can be used to implement splines of all orders. We plot in Fig. 6.17 a quadratic B-spline function. In order to model a curve, we associate a control vertex $\mathbf{x}$ with each function. With the shape of the functions in Fig. 6.17(a), endpoints are interpolated. It is also possible to obtain a closed curve, with the help of functions as in Fig. 6.17(b).

*Cubic spline* functions (degree $K = 3$) are particularly useful because of their property of minimizing the bending energy in $1D$ (see for instance [88, 89]).

In order to model a surface patch $S$, we construct a tensor product of spline functions, i.e., a surface point $S(u, v)$ will be written as
$S(u, v) = \sum_{kl} B_k(u) B_l(v) \mathbf{x}_{(k,l)}$, where we have fixed and suppressed $K$.

When using the B-spline functions of the type of Fig. 6.17(a), if none of the control vertices is repeated, the surface has a planar topology. When using the B-spline functions of the type of Fig. 6.17(b) for one of the variables (say $u$) setting



(a) (b)

**Figure 6.17**: B-spline functions. (a) Uniform quadratic singular B-spline functions. If a control point is attached to each function, a $\mathcal{C}^1$ curve that interpolates endpoints is obtained. (b) Uniform quadratic periodic splines. The two last control points are set equal to the two first in order to obtain a closed $\mathcal{C}^1$ curve.

the last $K$ control points along $u$ equal to the $K$ first for each row of control points, we obtain a cylindrical topology. When using the B-spline functions of the type of Fig. 6.17(b) for the two variables $(u, v)$ and setting the first $K$ control points equal to the last $K$ for each row and each column of control points, a toroidal topology is obtained.

The spherical topology is slightly more difficult to obtain than the planar, cylindrical, and toroidal topologies. The control vertices may be organized along an even number of "meridians," aligned along one parametric direction. As in the toroidal case, B-spline functions of the type of Fig. 6.17(b) are used for both parametric directions. Control points should be repeated, and one must distinguish between odd order and even order splines. For instance, in the case of biquadratic B-splines illustrated in Fig. 6.19, each two terminal control vertices on a meridian are set equal to the two first control vertices on the opposite meridian (which is why the number of meridians must be even).

When using one B-spline surface patch, only the above mentioned four different topologies may be obtained (planar, cylindrical, toroidal, and spherical). This is a significant limitation for a large number of anatomical structures.

In order to deform the surfaces, the same mechanisms that were developed for "snakes" may be used [76]. An "energy," compounding surface tension and bending and "external" forces defined using the data to be segmented (often, depending upon the gradient of image data), is minimized. As the energy term is generally too complex to be minimized in closed form (it depends upon local voxel values and their derivatives), the minimization is generally performed using an iterative process, for instance an iterative solution of a partial differential equation, or a sequence of least-square approximations [18]. Each step of the minimization corresponds to a new position for the surface. Such an iterative deformation process is illustrated in Fig. 6.18. An initial B-spline surface approximating a cylinder is positioned inside a three-dimensional volume of MRI data. Selected slices of the MRI data set are shown in Fig. 6.18(a). The surface then evolves and converges to approximate the epidermis [Fig. 6.18(b)–(d)]. A more detailed study of deformable models and their convergence is available in Chapter 3.

Splines of order 3 and higher are particularly useful for computing surface curvatures. The ability to compute surface curvatures is one of the justifications for using smooth surface models (see also Cohen et al. [90]). Curvatures are visualized in Figs. 6.18 and 6.20. In order to perform these curvature computations, since the spline surfaces are of the form $S(u, v)$ with derivatives with respect to $u$ and $v$ easily available, it is possible to use standard formulae. These formulae can be found in textbooks of differential geometry [91] or in [92]. Recent work has demonstrated that high curvature features are very useful for registration purposes [93,94]. However, several methods exist for computing curvatures on triangular meshes [53,54].

### 6.5.2 *Dynamic changes of topology*

Few methods using curved surface models and allowing dynamic changes of topology have been proposed. The method of Leitner and Cinquin [95] does so using cubic B-spline surfaces. This method starts with a spherical topology. A hole is created (evolving to a torus) if the method detects a self-intersection of the surface at some point. To create more complex topologies, several smoothly connected B-spline patches are used.

The study of recent work indicates that when using a triangular mesh to represent a surface, there is more flexibility in changing the topology. Lachaud and Montanvert [96] propose a method in which the topology can be changed at each iteration. After the vertices have been moved using rules similar to other deformable models, simple proximity constraints determine whether vertices should be created or removed or whether a connection between surface bodies should be created or removed. Figure 6.21 illustrates the operation of the method of [96] for extracting brain vessels from a MRI scan.

McInerney and Terzopoulos [97] tessellate volume space using a tetrahedral mesh. After each evolution of the surface, the vertices that are crossed in the mesh are marked, and the surface is retiled (from the tetrahedral mesh data) as explained in Section 6.3. Owing to the mechanism used for marking the tetrahedral mesh vertices, the surface can only expand everywhere or retract everywhere.

## 6.6 Optimization

The surface models, either spline or polygonal, that were constructed in the previous sections may benefit from various optimizations. During iso-surface construction or tiling from contours for instance, the difference between intraslice and interslice sampling ratios may create artifacts, which can be purely visual (such as a staircase effect) or which can affect further processing such as registration.

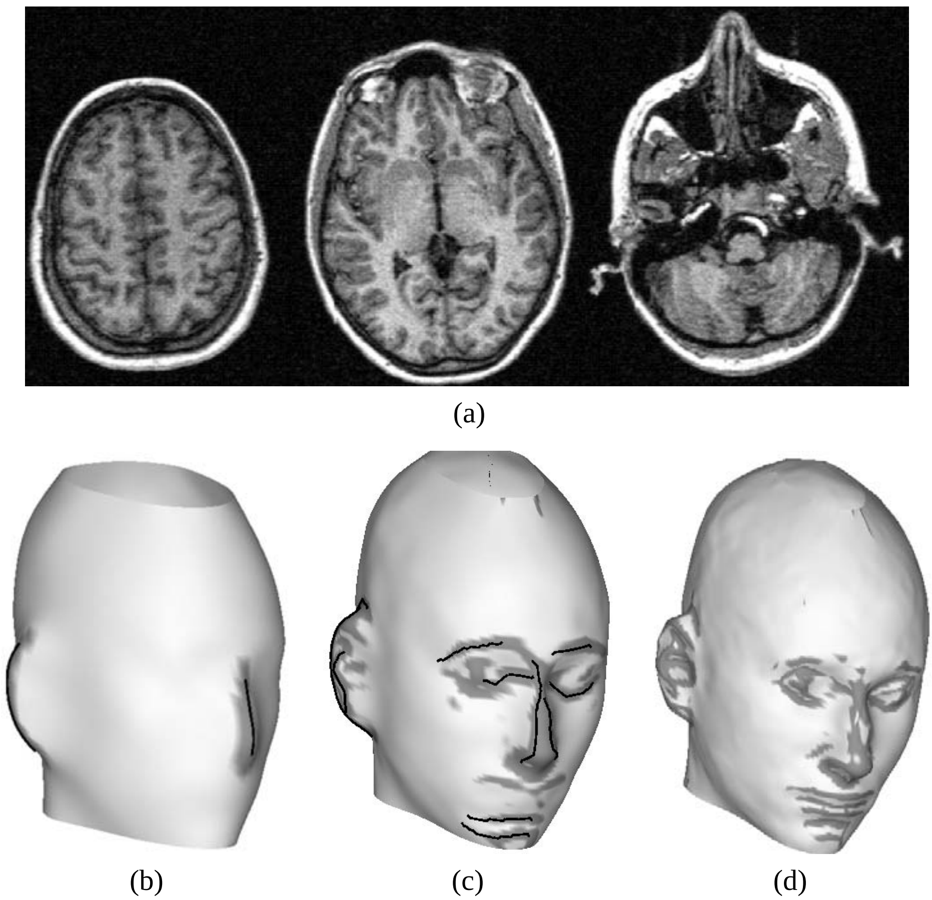



**Figure 6.18**: Segmenting a MRI scan using a B-spline deformable surface model. (a) MRI slices used as input to the B-spline deformable model. (b)–(d): successive iterations of the model with regions of higher curvature highlighted. Dark curves are crest lines, extracted with an algorithm described in [18].

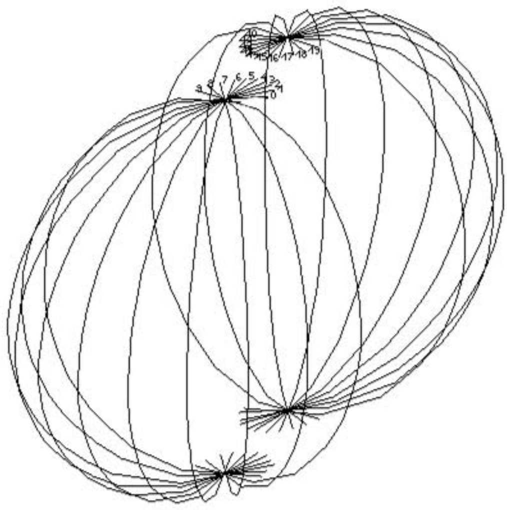


**Figure 6.19**: Network of control vertices for a biquadratic spline surface with spherical topology. Each pair of opposite "meridians"share three vertices at each pole (north, south), including the pole vertex itself.

### *6.6.1 Smoothing*

We describe here a method for smoothing polygonal surfaces that was developed by Taubin [98]. A straightforward technique for smoothing a polygonal mesh consists of replacing each vertex with an average of its neighbor vertices. Taubin noted that this technique resulted in shrinkage for the objects bounded by the polygonal mesh. His solution involves two steps. The set of vertices connected to a vertex $\mathbf{v}$ by an edge is denoted $\mathbf{v}^\star$. For each of the two steps, a vertex $\mathbf{v}$ is updated according to the following equation:

$$\mathbf{v} = \mathbf{v} + \{\lambda, \mu\} \sum_{\mathbf{w_j} \in \mathbf{v}^\star} \omega_j (\mathbf{w_j} - \mathbf{v}) \tag{6.7}$$

where the weights $\omega_j$ sum to one, the scale factor $\lambda$ is positive and applied during the first step, and the scale factor $\mu$ is negative and applied during the second step. Note that for a given step the displacements are first computed for each vertex and then applied all together at the same time. (Otherwise the displacement applied to a particular vertex would influence the displacements applied to its neighbors.)

The first step of this procedure thus corresponds to applying the traditional neighbor averaging step (albeit using a scale factor). While it has been observed that this first step yields shrinkage, the second step moves each vertex in the opposite direction, thereby compensating for the shrinkage effect.

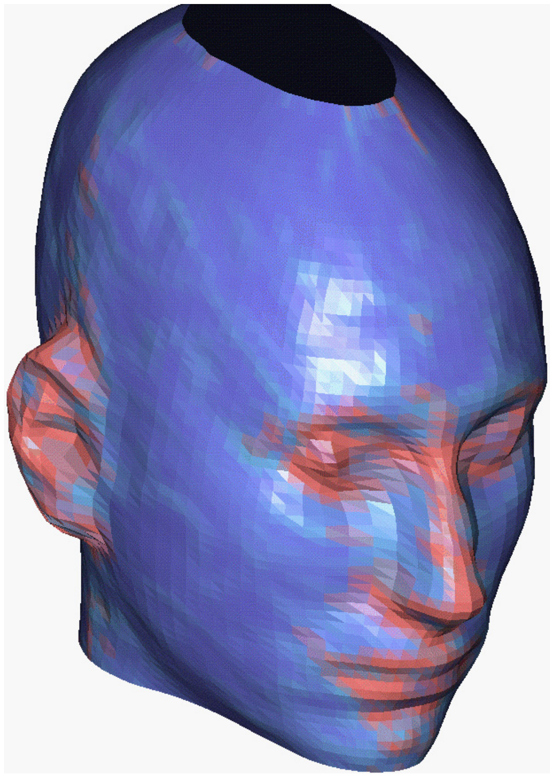
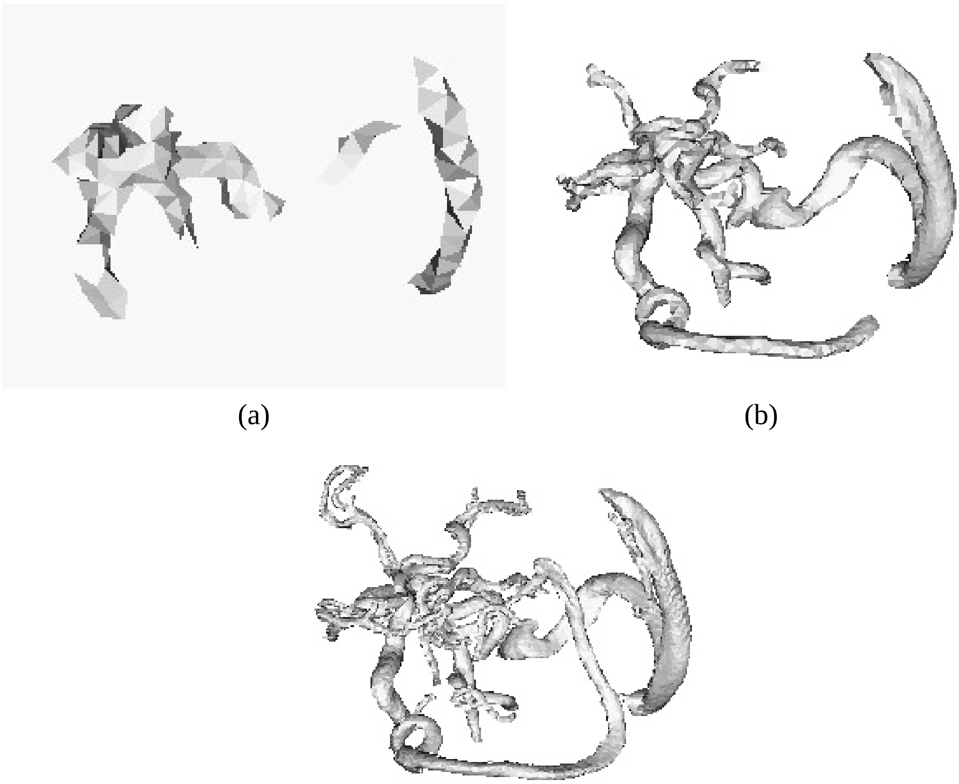


**Figure 6.20**: Curvature of a tensor-product spline surface. Maximum surface principal curvature represented using a colormap that varies from blue to red. (For a color version of this Figure see Plate 3 in the color section of this book.)

Taubin has provided a detailed mathematical justification of this process in [98], based on application of signal processing. The process of mapping each vertex $\mathbf{v}$ to $\sum_{\mathbf{w_j} \in \mathbf{v}^\star} \omega_j (\mathbf{w_j} - \mathbf{v})$ (the equivalent of a Laplacian operator) can be expressed by multiplying a column vector containing the surface vertices with a (sparse) matrix $K$ containing the weights $\omega_j$. Eigenvalues of this matrix may be considered as frequencies and eigenvectors as modes of the triangular mesh. Depending upon the distribution of these frequencies, a given triangular mesh can be considered to be *smooth*, or smoother than another mesh. A smoothing step is performed using the matrix $(I + \mu K)(I - \lambda K)$. If $k$ is an eigenvalue of $K$, $(1 + \mu k)(1 - \lambda k)$ is an eigenvalue of the matrix $(I + \mu K)(I - \lambda K)$. Hence $f(k) = (1 + \mu k)(1 - \lambda k)$ represents a transfer function. $\{\lambda, \mu\}$ can be set to create various types of filters, such as a low-pass filter.

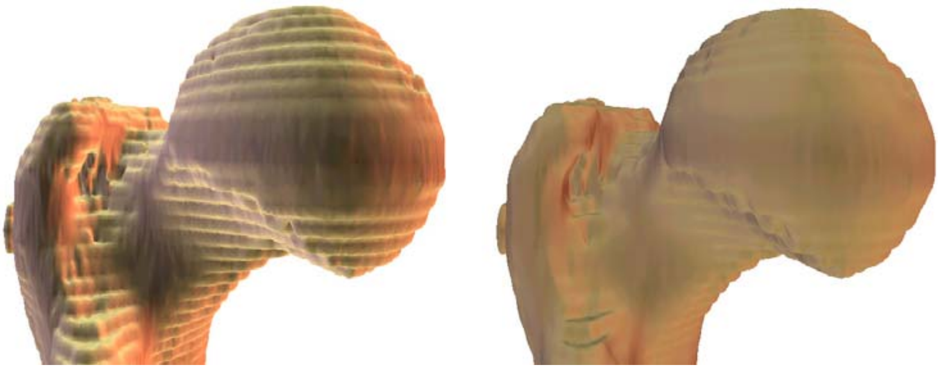
Taubin has proposed various suitable combinations for the $\{\lambda, \mu\}$ scaling factors. For instance $\{0.6307, -0.6732\}$ is a valid combination. Another parameter of the method is the number of iterations of the process, or smoothing steps.



(a) (b)

(c)

**Figure 6.21:** Extracting brain vessels from a MR scan using an adaptive topology deformable model represented as a triangular mesh. The model evolves inside an image pyramid; several iterations are performed for a given level [96]: (a) level 3 of the image pyramid; (b) level 1; (c) level 0. (Scan courtesy of UMDS Group, London. Images courtesy of J.O. Lachaud and A. Montanvert.)



(a) (b)

**Figure 6.22**: Illustration of surface smoothing using the method of [98]: (a) before smoothing; (b) after smoothing: 200 smoothing steps, $\{\lambda = .33, \mu = -.34\}$.

Figure 6.22 illustrates the operation of this method on a femur model comprising 180,854 triangles, extracted from a CT scan. The spacing between slices of the CT data is about 6 times the size of a pixel within a slice. The "staircase" effect resulting from this difference in sampling is illustrated in Fig. 6.22(a). The result of smoothing is visualized in Fig. 6.22(b).

The smoothing affects the vertex coordinates and, thus, the accuracy of the model. For some applications in which the geometric accuracy can't be compromised, the difference between the surfaces before and after smoothing may be tracked using the methods in Section 6.6.2.

Smoothing techniques are not limited to the ones presented here. For instance, a new technique, probably more complex than Taubin's, was recently introduced in [99].

### *6.6.2 Simplification and levels of detail*

Simplification deals with approximating a polygonal (spline) surface using fewer polygons (spline patches). This is important for processing, visualizing, and transmitting datasets that are too large for available computing and networking equipment.

Polygonal surfaces that result from iso-surface extraction (see Section 6.3) are generally over-tessellated and can greatly benefit from polygon reduction techniques without compromising accuracy.

Also, simplification techniques can be used for generating multi resolution hierarchies comprising several levels of detail. These hierarchies are very useful for interactive visualization applications, allowing interactive selection of the most appropriate level of detail of each structure to adapt to the parameters of the visualization. Figure 6.23 illustrates six surface structures of the human torso at three different levels of detail. For this application, the simplification process should preserve the visual appearance of surfaces as much as possible.

A technique for reducing the number of patches in a spline model was proposed by Gopi and Manocha [100]. This method merges adjacent triangular Bézier patches according to several patterns, to form larger triangular Bézier patches.

To guarantee the accuracy of the approximations, and the faithfulness of visualizations, it is very useful to be able to bound the deviation between the original model and the approximation. This operation is difficult because the maximum distance between two polygonal meshes is not reached in general for a pair of vertices. Methods for bounding the maximum surface deviation during simplification are described in [101–105]. Surface simplification methods are reviewed in [84]. Here are some additional references: [106–122].

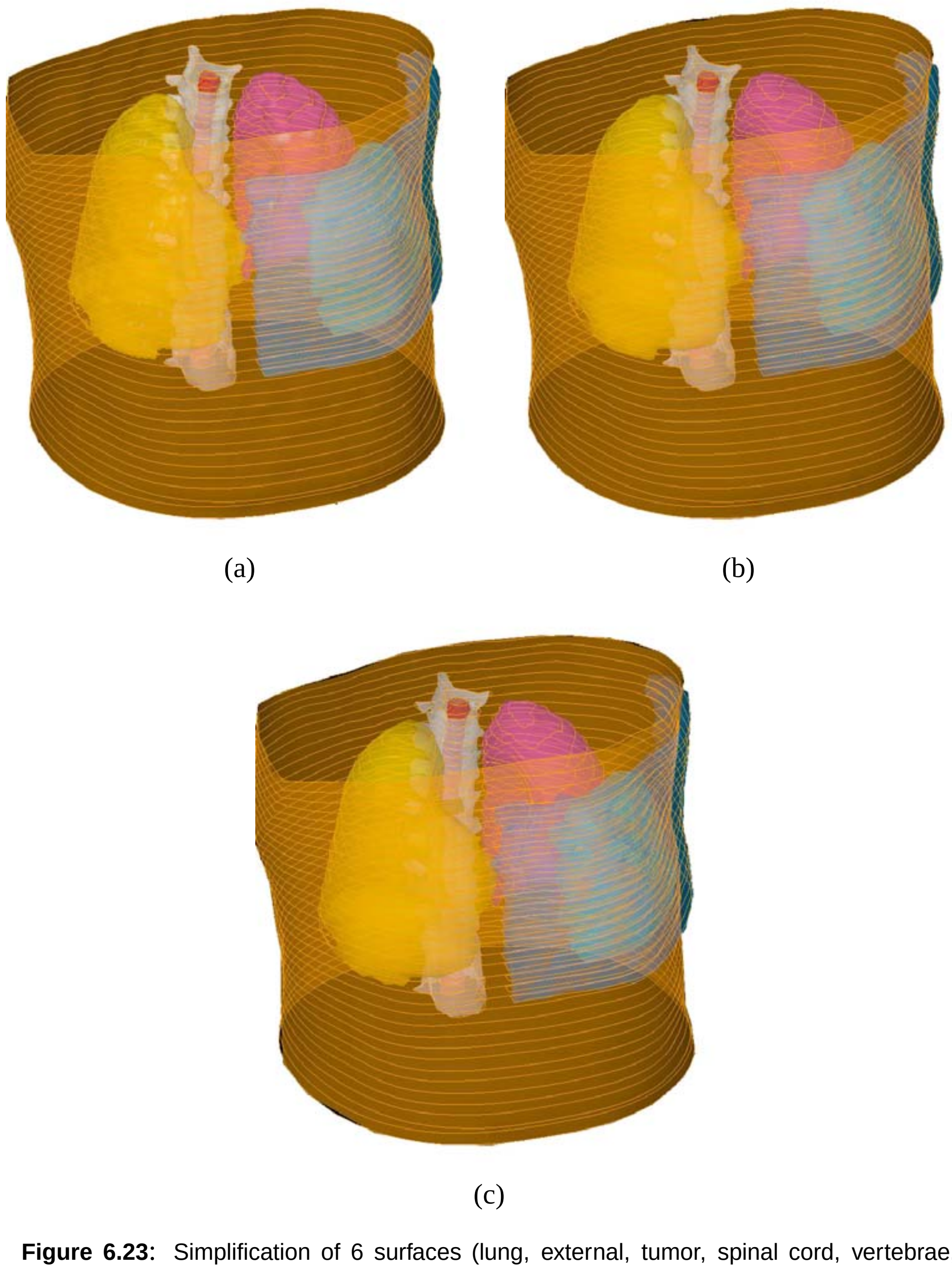
We next focus on the *variable tolerance* method, which is described fully in [84] [11]. This method, along with several of the methods referenced above, relies on the atomic operation represented in Fig. 6.24 to reduce the resolution of the surface: the *edge contraction* brings together the two endpoints of an edge, thereby eliminating one vertex and two triangles.

The *variable tolerance* method further relies on two main processes. The first process, called the *subdivision process*, measures the incremental deviation between two corresponding portions of a polygonal surface before and after a simplification operation (i.e., an edge contraction, but this would work with other operations such as vertex or triangle removals [124]). Given the two graphs representing the two surface portions, the subdivision process builds subdivisions of both surfaces in piecewise planar polygons such that there is a one-to-one correspondence between the elements and a direct comparison may be performed. Such subdivisions are shown in Fig. 6.25, denoted by $G$ and $H$. For each corresponding pair of planar polygons the maximum distance between the two must occur for a pair of corresponding vertices of the polygons. The *error bounding* process described below uses this information to keep track of a bound on the deviation between the simplified surface and the original surface.

In practice, the subdivision process only generates pairs of corresponding vertices. As illustrated in Fig. 6.25, for each vertex of one graph that does not appear in the other graph, a representative must be found in the other graph. This is done by projecting the vertex on the closest triangle. Pairs of edges belonging to different graphs must also be tested for potential *bridges*, which occur when the shortest segment that can be drawn between the two edges has its endpoints inside the edges: this is the case for edges $(\mathbf{v_1}, \mathbf{v_3})$ and $(\mathbf{v_0}, \mathbf{v_4})$ in Fig. 6.25. The full detail of this process is described in [84].

The second process, or *error bounding* process, is used for keeping track of the deviation after an arbitrary number of simplification operations. The error bounding process uses an *error volume* for reporting an error for each surface point (not

[11] An earlier description appeared in [123].



(a) (b) (c)

**Figure 6.23:** Simplification of 6 surfaces (lung, external, tumor, spinal cord, vertebrae, bolus) extracted and visualized using a clinical radio-therapy visualization system: (a) high resolution: 46,469 vertices; (b) low resolution: 4,688 vertices; (c) very low resolution: 1,811 vertices. Here, the difference between the surfaces can hardly be seen, which is the ideal outcome of surface simplification. The simplification method of [84] was used. (Courtesy of Mike Zeleznik, RAHD Oncology Products.) (For a color version of this Figure see Plate 4 in the color section of this book.)

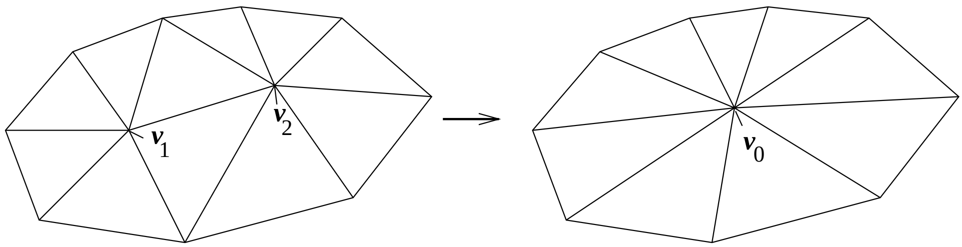



**Figure 6.24:** An edge contraction consists of bringing together the two endpoints $(\mathbf{v_1}, \mathbf{v_2})$ of an edge, until they become a single vertex $\mathbf{v_0}$.

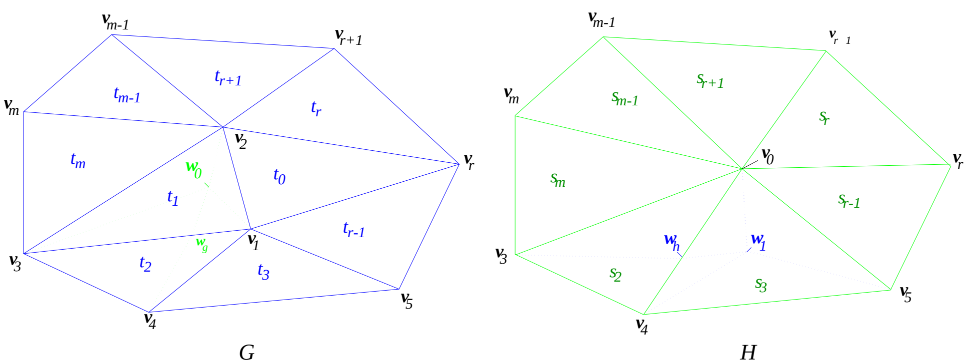



**Figure 6.25:** Graphs $G$ and $H$ built by the subdivision process. For each vertex of one graph that does not appear in the other graph, a representative must be inserted in the other graph (vertices $\mathbf{w_i}$). This is done by projecting the vertex on the closest triangle of the other graph: e.g., $\mathbf{v_0}$ of $H$ is projected to $\mathbf{w_0}$ on the closest triangle of $G$: $(\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3})$. In addition, for each pair of edges, a test determines whether the shortest segment that can be drawn between the two edges has its endpoints inside the edges. When this is the case, as for edges $(\mathbf{v_1}, \mathbf{v_3})$ and $(\mathbf{v_0}, \mathbf{v_4})$, both segment endpoints $\mathbf{w_g}$ and $\mathbf{w_h}$ are inserted in their respective graph. (For a color version of this Figure see Plate 5 in the color section of this book.)

only vertex). The error volume is defined by sweeping a ball across the simplified surface. The radius of the ball varies linearly over a surface triangle, interpolating the ball radii at the triangle vertices. To specify the error volume only one floating point value is necessary for each vertex of the simplified surface, representing the radius of the ball, or *error value,* at that vertex. The error volume is built gradually as the simplification progresses, containing the previous error volumes in a manner similar to Russian dolls. The properties of the error volume are summarized in Fig. 6.26(a), (b).

The objective of the error bounding process is to compute error values at the vertices of the surface, representing a bound to the deviation between original and simplified surface. Referring to Figs. 6.24 and 6.25, the simplest strategy is to determine an error value at the vertex $\mathbf{v_0}$, $\delta_0$ defining a valid error volume, and to leave all other error values unchanged. (One floating point value must be computed.) Other possible objectives are discussed in [84].

Each pair of corresponding vertices in the graphs $G$ and $H$ defined above, such as $\mathbf{w_g}$ and $\mathbf{w_h}$ in Fig. 6.25, allows us to define a constraint on the error values: the error volume should be at least as wide at this location as $\mathrm{dist}(\mathbf{w_g}, \mathbf{w_h})$ plus the width of the previous error volume (before the edge contraction occurred). Each constraint is represented by an error $\epsilon$ and a vector $\mathbf{d}$ whose origin is $\mathbf{w_h}$ (on the surface before the edge contraction) and whose destination is $\mathbf{w_g}$ (on the surface after the edge contraction). The origin of $\mathbf{d}$ is a linear combination of $\mathbf{v_0}$ and a point $\mathbf{v_l}$ of a triangle incident to $\mathbf{v_0}$ (on the edge opposite $\mathbf{v_0}$ ) with $(\alpha, 1-\alpha)$ weights. Then, to each constraint we associate the vector $\mathbf{u} = \mathbf{d}/\alpha$ and the "ball" of radius $r = \dfrac{\epsilon - (1-\alpha)\epsilon_l}{\alpha}$ centered at the tip of $\mathbf{u}$, as shown in Fig. 6.27(c).

We then consider the set of vectors $\mathbf{u}_j$ and balls or radii $r_j$ corresponding to all the constraints, as illustrated in Fig. 6.27(d). The error value $\delta_0$ may be computed by determining the smallest ball $B_0$ centered at the origin enclosing the set of balls. (For a ball of negative radius, $B_0$ is not required to enclose it, but to contain at least one of its points, potentially a single point where both balls are tangent.)

By allowing the center to move, we can reduce $\delta_0$, the radius of the enclosing ball. Given an initial ball enclosing a set of balls, we wish to obtain a smaller enclosing ball. We are not particularly interested in the smallest possible ball because after moving $\mathbf{v_0}$, the constraints become stale. A simple solution is to construct the bounding box of the set of balls, take its center, and determine the ball of smallest radius with that center.

The method can be extended for preservation of data attached to surface vertices, assuming a linear variation across triangles. This is illustrated by the example of Fig. 6.30.

#### *6.6.2.1 Examples*

The example of Fig. 6.23 is a visualization of 6 surfaces of a human torso at three levels of detail: high resolution, low resolution (10-fold reduction factor),

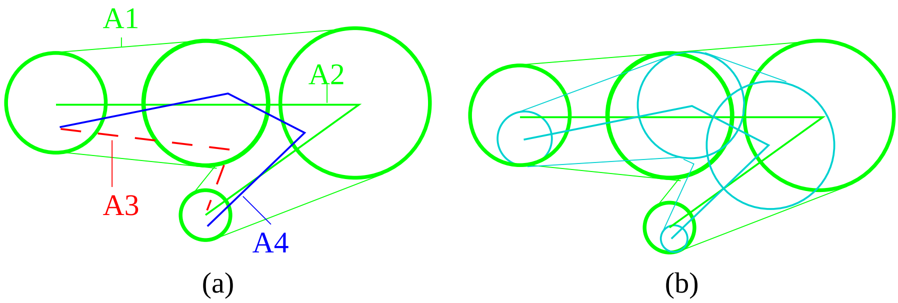



**Figure 6.26**: Properties of the error volume. The error volume is defined by sweeping a ball across the simplified surface. The radius of the ball varies linearly over a surface triangle, interpolating the ball radii at the triangle vertices. (a) Error volume, in green (A1), centered on the simplified surface (A2). The radii of the balls are such that the original surface (A3 or A4) is not only contained in the error volume, in dashed red (A3: incorrect), but also intersects all the spheres, in blue (A4: correct). (b) After several edge contractions, the resulting error volume contains all the intermediate error volumes. (For a color version of this Figure see Plate 6 in the color section of this book.)

and very low resolution (25-fold reduction factor). The difference between the surfaces can hardly be seen in Fig. 6.23, which is a very desirable goal in surface simplification. Statistics about the simplification of these surfaces are provided in Table 6.2. The availability of several levels of detail allows for real-time interaction with the surfaces on a computer display.

**Table 6.2**: Vertex counts and timings (CPU seconds measured on a DEC Alpha) for the level-of-detail computations on the 6 structures of the human torso dataset.

| Structure | Hi Res | Lo Res | Time | Very Lo Res | Time |
|---|---|---|---|---|---|
| 0 | 8,418 | 590 | 4.7 | 185 | .35 |
| 1 | 9,350 | 934 | 5.6 | 353 | .65 |
| 2 | 6,567 | 1,074 | 4.0 | 434 | .73 |
| 3 | 9,402 | 910 | 5.7 | 337 | .63 |
| 4 | 1,322 | 343 | 0.85 | 167 | .23 |
| 5 | 6,424 | 513 | 3.6 | 228 | .38 |
| 6 | 4,986 | 324 | 2.8 | 107 | .22 |
| Total | 46,469 | 4,688 | 27.5 | 1,811 | 3.2 |

The human Femur model, shown in Fig 6.28, is defined as an iso-surface of CAT scan data. It has 180,800 triangles and 90,400 vertices, with a bounding box of $90.1 \times 75.7 \times 225.6\text{mm}^3$ (life-size coordinates). We apply a succession of simplifications, providing statistics in Table 6.3. We use flat shading to visualize models

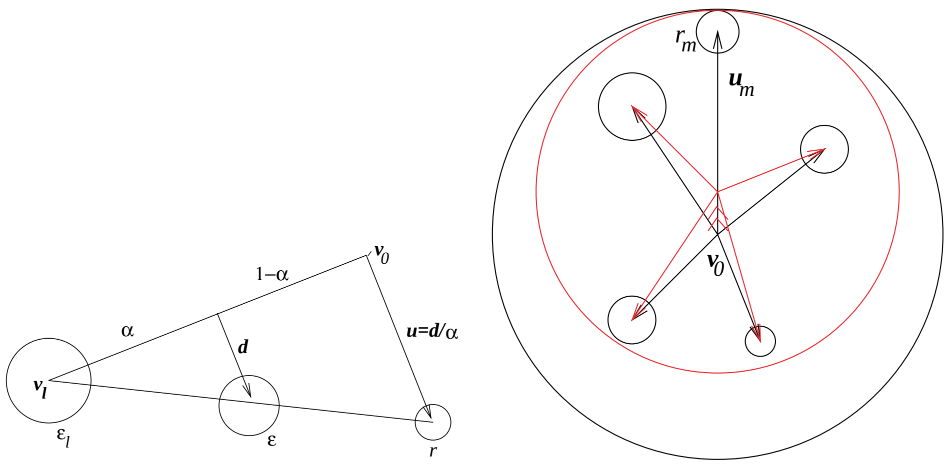



**Figure 6.27:** Error bounding process. (a) Each constraint determined during the subdivision process is represented by an error $\epsilon$ and a vector $\mathbf{d}$ whose origin lies on the surface before the edge contraction and whose destination lies on the surface after the edge contraction. The origin of $\mathbf{d}$ is a linear combination of $\mathbf{v_0}$ and a point $\mathbf{v_l}$ of a triangle incident to $\mathbf{v_0}$ with $(\alpha, 1-\alpha)$ weights. To each constraint we further associate the vector $\mathbf{u} = \mathbf{d}/\alpha$ and the "ball" of radius $r = \dfrac{\epsilon - (1-\alpha)\epsilon_l}{\alpha}$ centered at the tip of $\mathbf{u}$. (b) When considering the set of vectors $\mathbf{u}_j$ and balls of radii $r_j$ corresponding to all the constraints, the error value $\delta_0$ specifying the new error volume may be computed by determining the smallest ball $B_0$ centered at $\mathbf{v_0}$ and enclosing the set of balls.

in Figs. 6.28(a),(c),(e),(g), and (i). Associated error volumes are represented by coloring each vertex according to its error value using the color map of Fig. 6.28(b), and by using Gouraud shading for interpolating colors inside triangles. This example illustrates how a new simplification process may start with the simplified model and error volume produced by a previous simplification process and still preserve error bounds with respect to the original surface model. The error volume provides all necessary information to do so.

The Arteries model of Fig. 6.29 represents a network of carotid arteries extracted from a MRI scan, with 56,500 triangles and 28,500 vertices, and with a bounding box diameter of 127.2 mm. The model was simplified to 2,500 triangles.

The next example, courtesy of RAHD Oncology Products, is extracted from a radiation therapy planning system. A surface model representing a lung is simplified from 12,782 triangles in Fig. 6.30(b) to 917 triangles in Fig. 6.30(a) using the *variable tolerance* method in 5.3 seconds on a DEC alpha. The radiation dose, a scalar field, is mapped from $\mathbf{R}^3$ to surface vertices before simplification in Figs. 6.30(a), (b); in Fig. 6.30(a) the dose data is ignored during simplification. In Fig. 6.30(b) no simplification occurs; regions of low dose are colored blue, while

**Table 6.3**: Simplification statistics for Femur and Arteries models. The CPU time was measured in seconds on an IBM RS6000 580 workstation.

| Femur | | | | |
|---|---|---|---|---|
| Tolerance | | Number of | Percentage | CPU |
| % diam. | mm | Triangles | of Original | time |
| 0 | | 180.8K | 100% | |
| 0.2 % | 0.5 | 26.8K | 15% | 30.1 |
| 0.39% | 1 | 9,592 | 5.3% | 4.03 |
| 0.63% | 1.6 | 4,618 | 2.5% | 1.56 |
| 0.78% | 2 | 3,124 | 1.7% | 0.9 |
| Arteries | | | | |
| Tolerance | | Number of | Percentage | CPU |
| % diam. | mm | Triangles | of Original | time |
| 0 | | 56.5 K | 100% | |
| 0.8% | 1 | 5,633 | 9.9% | 9.2 |
| 1.6% | 2 | 2,571 | 4.5% | 1.2 |

high dose regions are colored red, with a threshold appropriately set between the two. The result of (a) is not satisfactory, because knowledge of the (rectangular) radiation beam geometry indicates that the iso-dose curve shown should be approximately rectangular. Using the variable tolerance method the dose data may be preserved during simplification, resulting in the surface of Fig. 6.30(c) with 2,747 triangles, computed in 10.5 seconds.

## 6.7 Exemplary algorithms operating on polygonal surfaces

A polygonal surface is a powerful representation for various algorithms. In this section, we provide examples of algorithms that operate on a polygonal surface model. These examples relate to projecting for simulations and projective registration.

### *6.7.1 Apparent contours and perspective registration*

Surface *apparent contours*, or *silhouette curves* are such that rays emanating from a center of perspective (possibly very far from the surface for a parallel projection) and tangent to the surface meet the surface on an apparent contour.

More specifically, for each surface triangle, the *viewing direction* is defined as the vector originating from the center of perspective to the triangle centroid. If the triangle normal (defined by the cross product of ordered oriented triangle edges) makes an obtuse angle with the viewing direction, the triangle is said to be *visible* and *invisible* otherwise. Apparent contours are a subset of surface edges, such that the triangle on one side of the edge is visible and the triangle on the other side of

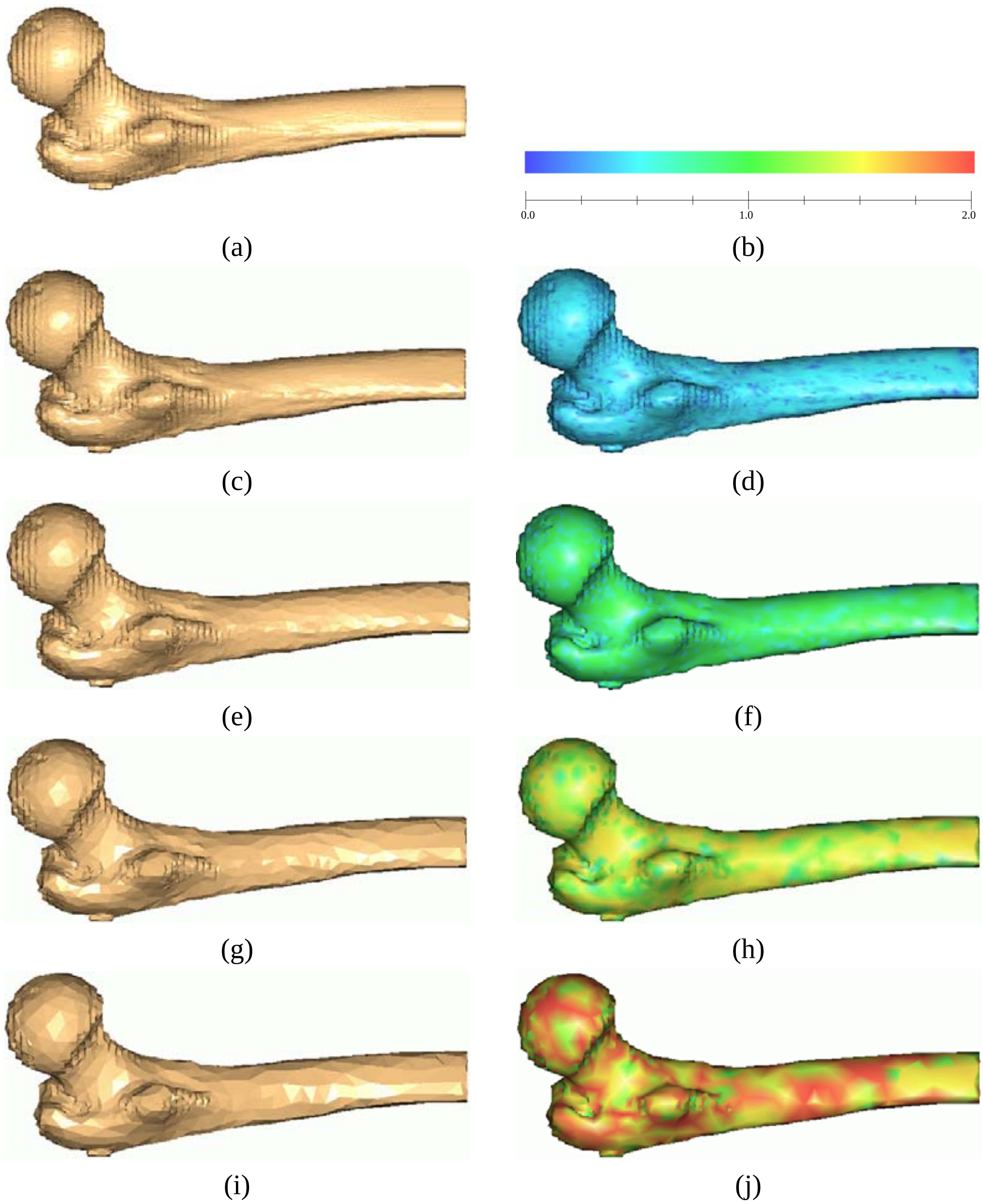



**Figure 6.28:** Successive simplification of a femur model, reusing error volumes computed with the previous simplification: (a) femur model (181,000 triangles); (b) error volume color-map; (c) simplification with 0.5 mm tolerance (26,800 triangles); (d) error volume of (c); (e), (f) simplification with 1.0 mm tolerance (9,592 triangles); (g) (h) 1.6 mm tolerance (4,618 triangles); (i), (j) 2 mm tolerance (3,124 triangles). (For a color version of this Figure see Plate 7 in the color section of this book.)

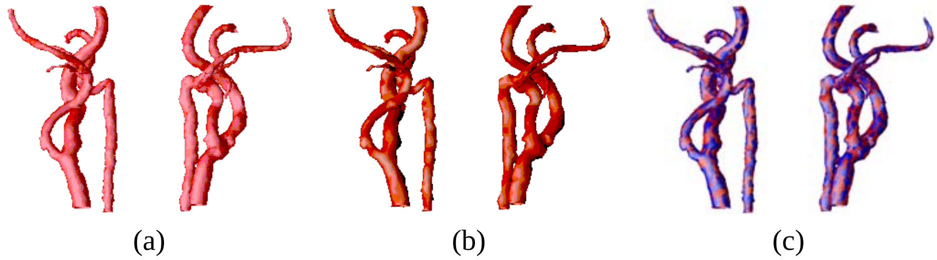


**Figure 6.29**: Simplification of a surface model of carotid arteries: (a) carotid arteries (57,000 triangles); (b) Simplification (5,600 triangles) with a maximum tolerance of 2 mm (0.8% of the bounding box diameter); (c) superimposition of (a) and (b). One surface is colored in blue and the other in red. (For a color version of this Figure see Plate 8 in the color section of this book.)

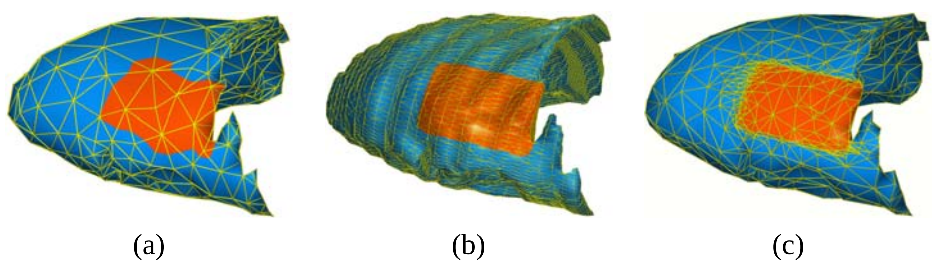
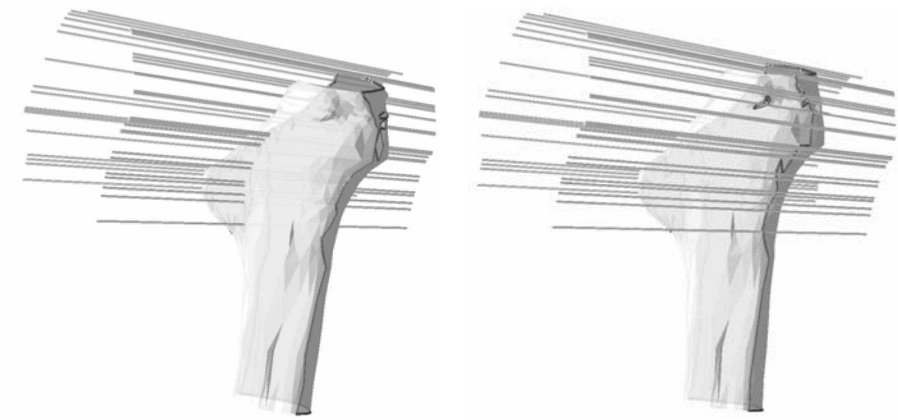


**Figure 6.30**: Preserving during simplification data mapped on the surface: (a) radiation dose data mapped before simplification and not preserved during simplification (917 triangles); (b) no simplification (12,782 triangles); (c) simplification with dose data preservation (2,747 triangles). (Images courtesy of Mike Zeleznik, RAHD Oncology Products.) (For a color version of this Figure see Plate 9 in the color section of this book.)

the edge is invisible. The apparent contours are such that the edges are linked to form (non-planar) closed polygonal curves in three dimensions.

To build the apparent contours, we first identify all edges belonging to any apparent contour using the criterion defined above and add such edges to a list. We orient edges such that the visible triangle is on the left side of the edge, thus defining an edge origin and an edge destination. We then iterate on the following process: (1) we take the first edge in the list; we create a new apparent contour starting with that edge; (2) we complete the apparent contour containing that edge as follows: starting from the destination of the current edge we determine a next edge; we visit the triangles incident to the destination vertex in a counter-clockwise fashion and determine the first edge that belongs to the list of apparent contour edges [12]. We reapply step (2) until the next edge is the same as the first edge that was processed

[12] This step is necessary because there may be several such edges.



(a) (b)

**Figure 6.31**: Use of apparent contours for projective registration. Pre operative proximal femur model before (a) and after (b) registration to three-dimensional lines emanating from the intra operative x-ray images. Straight lines are x-ray paths. The dark three-dimensional curves on the femur models are apparent contours (depending on the position).

in (1); (3) we remove all the edges forming that contour from the list of apparent contour edges. We reapply steps (1) to (3) until the list of apparent contour edges is empty. Apparent contours obtained using this algorithm can be seen in Fig. 6.31.

A method for projective (2D to 3D) registration using apparent contours was described in [72]. The method of [72] was designed within the framework of Total Hip Replacement surgery as part of a joint study with Integrated Surgical System (ISS), of Davis, CA, to extend the ROBODOC® system. Used for Total Hip Replacement (THR) surgery, the surgical robot of the ROBODOC® system accurately mills the cavity for the femoral implant. To register the surgical robot to its planned trajectory, a clinical protocol uses two or three metallic markers surgically inserted in the femur [125,126]. The registration method described in [72] was developed to study the feasibility of replacing external marker-based registration with anatomy-based and x-ray-based registration. The proposed method requires acquiring, and calibrating with respect to the surgical robot, intra operative x-rays of the anatomical structure of interest.

Having extracted 2D bone contours from the x-ray images, bundles of straight lines in three-dimensional space (x-ray paths) emanating from the position of the x-ray source and passing through the 2D bone contours can be determined (using image calibration information). When the pre operative femur model is perfectly registered with respect to physical space (surgical robot and x-ray images), the x-ray paths determined above are grazing the surface of the femur model. Since the initial position is expected to be misregistered, x-ray paths are a priori not grazing (but intersecting, or "missing," the femur model altogether).

The registration process described in detail in [72], and illustrated in Fig. 6.31, finds a rigid transformation optimizing the position and orientation of the surface model such that as many X-ray paths as possible become tangent to the surface.

Additional material on registration, and particularly 2D to 3D registration, can be consulted in Chapter 8.

### *6.7.2 Surface projection for x-ray simulations*

After a registration was performed, as discussed in Section 6.7.1, in this section we address the issue of providing clinicians with visual information to validate the registration accuracy [13].

Motivated by this, we simulate postoperative x-ray images by superimposing (projecting) an implant positioned preoperatively on the x ray using the image calibration and the transformation computed by the 2D/3D registration. We also superimpose a preoperative femur model that was used for anatomy-based registration to verify how it aligns with the bone contours in the x-ray images. With these images, a judgment on the registration accuracy can be made.

For a given x-ray image, using calibration information, we first determine a center of perspective and use it to compute apparent contours as explained in Section 6.7.1. Then, the method works independently of whether we wish to correct for a potential geometric distortion in x-ray images or not: for each pixel of the x-ray image (original image pixels, or rectified image pixels) we determine an x-ray path from the $(u, v)$ (grid) coordinates corresponding to the pixel and the center of perspective. For a pictorial representation of x-ray paths, see Fig. 6.31.

The main idea is to substitute a straightforward projection of the geometry with the following process based on distance computations and on the conversion of distances to pixel gray-scale values. This process allows much more flexibility in the "projection" of surface models, allowing to reproduce the geometric distortion in an x ray and to avoid aliasing effects when drawing the contour of the surface projection.

We compute the distance from each x-ray path to the apparent contours of the three-dimensional model of the implant (or other preoperative three-dimensional shape): we must compute the shortest distance between a straight line and a curve in three-dimensional shape. A hierarchical description of the curves may be used for this purpose. This computation is described in [72].

Finally, we convert the computed distances to gray-scale values, for a useful range of the distances. These gray-scale values form an (masked) image that is inserted in the original x-ray image mask to change the x-ray image.

Various mappings may be used for converting distances to gray-scale values. To produce line drawings, only very short distances should produce alterations in gray-scale values. For instance, to produce the image of Fig. 6.32(c) the following

[13] Although numerous registration approaches have been presented, the topic of registration validation has scarcely been addressed in the literature.

mapping was used: if the distance was less than 0.05 mm, a gray-scale value of 0 was used, otherwise, if the distance was less than 0.1 mm, a gray-scale value of 30 was used, otherwise, if the distance was less than 0.2 mm, a gray-scale value of 60 was used, and otherwise, no change to the existing gray-scale value was done. (This mapping assumed that 8 bits, or 256 gray values, were used per pixel.) The resulting image, shown in Fig. 6.32(c), shows only the projected silhouette of the implant (or other preoperative three-dimensional model), which is sufficient to precisely indicate the position of the implant, but does not obscure any of the anatomy (see Fig. 6.32). For more details, the reader may also consult [127].

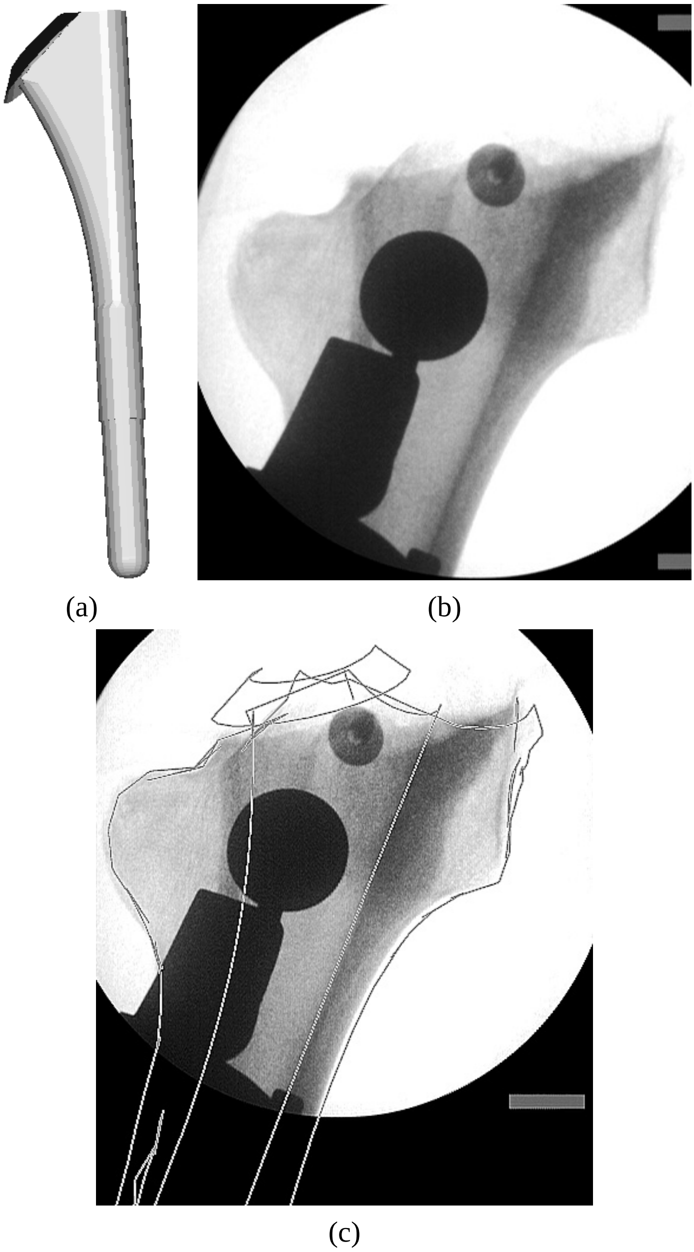


**Figure 6.32:** Simulating post-operative x-ray images: (a) three-dimensional model of implant; (b) intraoperative x-ray image, characterized by a limited field of view, geometric distortion, and occlusion (a bone fixator appears in black); (c) projecting the preoperative implant model and femur models after registration onto the intra-operative x-ray image. The x-ray image was not corrected for distortion; instead, the same distortion parameters are applied to the model when projecting.

## 6.8 Conclusion and perspective

We have provided a sample of methods for extracting, optimizing and exploiting surface representations in medical images. We have put an emphasis on polygonal (particularly, triangular) meshes, which are quite simple and yet represent a powerful representation for visualization and other processes.

Many notions and operations that appear at first restricted to curved surfaces (such as spline surfaces) extend to polygonal meshes as well: smoothing (Section 6.6.1), curvature computation (Section 6.3.4), reduction of complexity (Section 6.6.2), etc.

Segmenting and extracting detailed surface models from large data sets is still a challenging problem today. Concerning the surface extraction and representation part of the problem, polygonal simplification and hierarchical descriptions have permitted significant progress, and many applications of them can still be exploited: for instance, levels of details have only been used for visualization so far. Other interesting applications of levels of details are registration, efficient collision detection, etc.

While direct volume rendering has made significant progress in recent years, with the development of efficient software and hardware [2, 128], surface representations are very effective for various applications, only some of which do relate to visualization. The combination of surface and volume rendering is also promising.

## 6.9 References

[1] U. Tiede, T. Schiemann, and K. Höhne, "High quality rendering of attributed volume data," in *Visualization'98*, (Raleigh, NC.), pp. 255–262, IEEE, Oct. 1998.

[2] H. Pfister, J. Hardenbergh, J. Knittel, H. Lauer, and L. Seiler, "The VolumePro real-time ray-casting system," in *ACM Siggraph*, pp. 251–260, Aug. 1999.

[3] M. Zeleznik, "3D visualization: What does it mean?," in *XII International Conference on the Use of Computers in Radiation Therapy (ICCR)*, (Salt Lake City, UT), May 27-30 1997. www.rahd.com.

[4] W. Grimson, G. Ettinger, S. White, P. Gleason, T. Lozano-Perez, W. Wells III, and R. Kikinis, "Evaluating and validating an automated registration system for enhanced reality visualization in surgery," in *First International conference on computer vision, virtual reality and robotics in medicine, CVRMed'95* (N. Ayache, ed.), (Nice, France), Springer-Verlag, 1995. Lecture Notes in Computer Science.

[5] A. Colchester, J. Zhao, K. Holton-Tainter, C. Henri, N. Maitland, P. Roberts, C.Harris, and R. Evans, "Development and preliminary evaluation of VISLAN, a surgical planning and guidance system using intra-operative video imaging," *Medical Image Analysis*, vol. 1, no. 1, 1996.

[6] L. Joskovicz, L. Tockus, Z. Yaniv, A. Simkin, and C. Milgrom, "Computer-aided image-guided bone fracture surgery–concept and implementation," in *12th Int. Symp. on Computer Assisted Radiology and Surgery* (Lemke *et al.*, eds.), 1998.

[7] C. Pelizzari, G. T. Y. Chen, D. R. Spelbring, R. R. Weichselbaum, and C. T. Chen, "Accurate three-dimensional registration of CT, PET and/or MR images of the brain," *Journal of Computer Assisted Tomography*, vol. 13, no. 1, pp. 20–26, 1989.

[8] D. Simon, *Fast and Accurate Shape-Based Registration*. Ph.D. thesis, Carnegie Mellon University, Dec. 1996. CMU-RI-TR-96-45.

[9] J. Herring, B. Dawant, C. Maurer, D. Muratore, R. Galloway, and J. Fitzpatrick, "Surface-based registration of CT images to physical space for image-guided surgery of the spine: A sensitivity study," *IEEE Transactions on Medical Imaging*, vol. 17, pp. 743–752, Oct. 1998.

[10] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," in *ACM Siggraph*, pp. 71–78, July 1992.

[11] F. Bernardini and C. Bajaj, "Sampling and reconstructing manifolds using $\alpha$-shapes," in *9th Canadian Conference on Computational Geometry*, pp. 193–198, Aug. 1997.

[12] N. Amenta, M. Bern, and M. Kamvysselis, "A new voronoi-based surface reconstruction algorithm," in *ACM Siggraph*, pp. 415–421, July 1998.

[13] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," Tech. Rep. RC21463, IBM T.J. Watson Research Center, Yorktown Hts, NY, 1999.

[14] M. Deering, "Geometry compression," in *Siggraph'95 Conference Proceedings*, (Los Angeles), pp. 13–20, August 1995.

[15] G. Taubin, W. Horn, F. Lazarus, and J. Rossignac, "Geometry coding and VRML," *Proceedings of the IEEE*, vol. 86, pp. 1228–1243, June 1998.

[16] S. Cotin, H. Delingette, and N. Ayache, "Real-time elastic deformations of soft tissues for surgery simulation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, pp. 62–73, January-March 1999.

[17] G. Farin, *Curves and surfaces for computer aided geometric design: a practical guide*. Academic Press, 1988.

[18] A. Guéziec, "Surface representation with deformable splines: Using decoupled variables," *IEEE Computational Science and Engineering Magazine*, vol. 2, pp. 69–80, March 1995.

[19] P. Dierckx, *Curve and surface Fitting with Splines*. Oxford Science Publications, 1993.

[20] S. Sinha and P. Seneviratne, "Single valuedness, parametrization and approximating 3D surfaces using B-splines," in *Geometric Methods in Computer Vision II*, vol. 2031, pp. 193–204, SPIE, July 1993.

[21] D. Terzopoulos and D. Metaxas, "Dynamic 3D models with local and global deformations: Deformable superquadrics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 703–714, July 1991.

[22] E. Bardinet, L. Cohen, and N. Ayache, "Superquadric and free-form deformations: A global model to fit and track 3D medical data," in *First International conference on computer vision, virtual reality and robotics in medicine, CVRMed'95*, no. 905 in Lecture Notes in Computer Science, pp. 319–326, Apr. 1995.

[23] C. Bajaj, F. Bernardini, S. Cutchin, K. Lin, E. Sacks, and D. Schikore, "Comprehensive analysis of joints from patient clinical data," Tech. Rep. 97-019, Purdue University Comp. Sci. Dept., 1997.

[24] P. Ning and J. Bloomenthal, "An evaluation of implicit surface tilers," *IEEE Computer Graphics & Applications*, vol. 13, pp. 33–44, Nov. 1993.

[25] J. Bloomenthal, "Surface tiling," in *Introduction to Implicit Surfaces* (J. Bloomenthal, ed.), Computer Graphics and Geometric Modeling, pp. 126–165, Morgan Kaufmann, 1997.

[26] A. Guéziec, "Comparing two fundamental strategies for iso-surface extraction in regular volume data," in *Computational and Visualization Methods for Surface Analysis* (K. Chana, ed.), Marcel Dekker, 2000. To appear (available as a draft from the author).

[27] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," in *ACM Siggraph*, vol. 21-4, (Anaheim, California), pp. 163–169, July 1987.

[28] A. Kalvin, *Segmentation and Surface-Based Modeling of Objects in Three-Dimensional Biomedical Images*. Ph.D. thesis, New York University, June 1991.

[29] G. Nielson and B. Hamann, "The Asymptotic Decider: Resolving the ambiguity in marching cubes," in *Conference on Visualization*, pp. 83–91, IEEE, 1991.

[30] G. Wyvill, C. McPheeters, and B. Wyvill, "Data structures for soft objects," *Visual Computer*, vol. 2, pp. 227–234, 1986.

[31] G. Herman and H. Liu, "Three-dimensional display of human organs from computed tomograms," *Computer Graphics and Image Processing*, vol. 9, pp. 1–21, 1979.

[32] G. Frieder, G. Herman, G. Meyer, and J. Udupa, "Large software problems for small computers: An example from medical imaging," *IEEE Software*, pp. 39–47, Sept. 1985.

[33] H. Cline, W. Lorensen, D. Ludke, C. Crawford, and B. Teeter, "Two algorithms for the three-dimensional reconstruction of tomograms," *Medical Physics*, vol. 15, no. 3, pp. 320–327, 1988.

[34] H. H. Baker, "Building surfaces of evolution: The weaving wall," *The International Journal of Computer Vision*, vol. 3, pp. 51–71, 1989.

[35] B. Payne and A. Toga, "Surface mapping brain function on 3D models," *IEEE Computer Graphics & Applications*, vol. 10, pp. 33–41, September 1990.

[36] M. Hall and J. Warren, "Adaptive polygonalization of implicitly defined surfaces," *IEEE Computer Graphics & Applications*, vol. 10, pp. 33–42, November 1990.

[37] J. Wilhelms and A. van Gelder, "Topological considerations in isosurface generation," *Computer Graphics*, vol. 24, pp. 79–86, Nov. 1990. Workshop on Volume Visualization.

[38] A. Wallin, "Constructing isosurfaces from CT data," *IEEE Computer Graphics & Applications*, vol. 11, pp. 28–33, November 1991.

[39] B. Natarajan, "On generating topologically consistent isosurfaces from uniform samples," *The Visual Computer*, vol. 11, no. 1, pp. 52–62, 1994.

[40] T. Itoh and K. Koyamada, "Automatic isosurface propagation using an extrema graph and sorted boundary cell lists," *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, pp. 319–327, December 1995.

[41] C. Bajaj, V. Pascucci, and D. Schikore, "The contour spectrum," in *IEEE Visualization*, (Phoenix, AZ), pp. 167–173, Oct. 1997.

[42] Y.-J. Chiang, C. Silva, and W. Schroeder, "Interactive out-of-core isosurface extraction," in *Visualization'98*, (Raleigh, NC.), pp. 167–174, IEEE, Oct. 1998.

[43] A. Guéziec and R. Hummel, "Exploiting triangulated surface extraction using tetrahedral decomposition," *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, pp. 328–342, December 1995.

[44] A. Koide, A. Doi, and K. Kajioka, "Polyhedral approximation approach to molecular orbital graphics," *J. Molecular Graphics*, vol. 4, pp. 149–160, 1986.

[45] A. Doi and A. Koide, "An efficient method of triangulating equi-valued surfaces by using tetrahedral cells," *IEICE Transactions*, vol. E74, pp. 214–224, January 1991.

[46] D. Dean, P. Buckley, F. Bookstein, J. Kamath, D. Kwon, L. Friedman, and C. Lys, "Three dimensional MR-based morphometric comparison of schizophrenic and normal cerebral ventricles," in *Visualization in Biomedical Computing*, vol. 1131 of *Lecture Notes in Computer Sciences*, pp. 363–371, Springer Verlag, 1996.

[47] D. M. Muratore, J. L. Herring, B. M. Dawant, and R. L. J. Galloway, "Effect of ultrasonic transducer frequency on registration of ultrasound to CT vertebral images," in *Medical Imaging—Image Processing*, vol. 3661-85, SPIE, Feb. 1999.

[48] N. Gencer, S. Williamson, A. Guéziec, and R. Hummel, "Optimal reference electrode selection for electric source imaging," *Electro-Encephalography and Clinical NeuroPhysiology*, vol. 99, pp. 163–173, Aug. 1996.

[49] A. Guéziec and D. Dean, "The Wrapper algorithm: A surface optimization algorithm that preserves highly curved areas," in *Second Conference on Visualization in Biomedical Computing*, vol. 2359, (Rochester, Minnesota), pp. 631–42, SPIE, October 4-7 1994.

[50] O. Monga, S. Benayoun, and O. Faugeras, "Using third order derivatives to extract ridge lines in 3D images," in *Conference on Vision and Pattern Recognition*, (Urbana Champain, Illinois), IEEE, June 1992.

[51] J.-P. Thirion and A. Gourdon, "Computing the differential characteristics of isointensity surfaces," *Computer Vision and Image Understanding*, vol. 61, pp. 190–202, March 1995.

[52] B. Hamann, “Modeling contours of trivariate data,” *Mathematical Modelling and Numerical Analysis*, vol. 26, no. 1, pp. 51–57, 1992. Gauthier-Villars, France.

[53] B. Hamann, “Curvature approximation for triangulated surfaces,” in *Geometric Modelling* (G. Farin, H. Hagen, and H. Noltemeier, eds.), Computing Suppl. 8, pp. 139–153, New York: Springer-Verlag, 1993.

[54] G. Taubin, “Estimating the tensor of curvature of a surface from a polyhedral approximation,” in *Fifth International Conference on Computer Vision*, pp. 902–907, IEEE, 1995.

[55] F. Bookstein and C. Cutting, “A proposal for the apprehension of curving craniofacial form in three dimensions,” in *Craniofacial Morphogenesis and Dysmorphogenesis* (K. Vig and A. Burdi, eds.), pp. 127–140, Ann Arbor, MI: Center for Human Growth and Development, 1988.

[56] C. Cutting, “Applications of computer graphics to the evaluation and treatment of major craniofacial malformation,” in *3-D Imaging in Medicine* (Udupa and Herman, eds.), CRC Press, 1989.

[57] O. Monga, S. Benayoun, and O. Faugeras, “From partial derivatives of 3D volumetric images to ridge lines,” in *IEEE Conf. on Computer Vision and Pattern Recognition, CVPR’92*, (Urbana Champaign), 1992.

[58] N. Ayache, A. Guéziec, J. Thirion, and A. Gourdon, “Evaluating 3-D registration of CT-scan images using crest lines,” in *Mathematical Methods in Medical Imaging II*, vol. 2035-06, (San Diego), pp. 60–71, SPIE, July 1993.

[59] N. Ayache, “Medical computer vision, virtual reality and robotics,” *Image and Vision Computing*, vol. 13, pp. 295–313, may 1995.

[60] J.-P. Thirion and A. Gourdon, “The 3D marching lines algorithm,” *Graphical Models and Image Processing*, vol. 58, pp. 503–509, November 1996.

[61] A. Guéziec and N. Ayache, “Smoothing and matching of 3-D space curves,” *Int. Journal of Computer Vision*, vol. 12, no. 1, pp. 79–104, 1994.

[62] J. Koenderink, *Solid Shape*. Cambridge, Massachusetts: M.I.T. Press, 1990.

[63] M. Hosaka, *Modeling of Curves and Surfaces in CAD /CAM*. Berlin: Springer Verlag, 1992.

[64] I. Porteous, *Geometric Differentiation: For the Intelligence of Curves and Surfaces*. Cambridge University Press, 1994.

[65] S. Markatis, *Some Generic Phenomena in Families of Surfaces in R3*. Ph.D. thesis, University of Liverpool, 1980.

[66] G. Malandain, *Filtrage, Topologie et mise en correspondance d’images médicales*. Ph.D. thesis, Ecode Centrale Paris, Sept. 1992.

[67] J. Fuchs, Z. Kedem, and S. Uselton, “Optimal surface reconstruction from planar contours,” *Commun. ACM*, vol. 20, pp. 693–702, Oct. 1977.

[68] H. Christiansen and T. Sederberg, “Conversion of complex contour line definitions into polygonal element mosaics,” in *Compt. Graph.*, vol. 12, No. 2, pp. 187–192, Aug. 1978.

[69] D. Meyers, S. Skinner, and K. Sloan, "Surfaces from contours," *ACM Transactions on Graphics*, vol. 11, pp. 228–258, July 1992.

[70] B. Geiger, "Three-dimensional modeling of human organs and its application to diagnosis and surgical planning," Tech. Rep. 2105, INRIA, Sophia-Antipolis, Valbonne, France, 1993.

[71] C. Maurer, G. Aboutanos, B. Dawant, R. Maciunas, and J. Fitzpatrick, "Registration of 3-D images using weighted geometrical features," *IEEE Transactions on Medical Imaging*, vol. 15, pp. 836–849, Dec. 1996.

[72] A. Guéziec, P. Kazanzides, B. Williamson, and R. Taylor, "Anatomy-based registration of CT-scan and intraoperative X-ray images for guiding a surgical robot," *IEEE Transactions on Medical Imaging*, vol. 17, pp. 715–728, Oct. 1998.

[73] A. Guéziec, "Assessing the registration of CT scan data to intra-operative x-rays by integrating x-rays with pre-operative information," in *Medical Imaging—Image Processing*, vol. 3661-88, SPIE, Feb. 1999.

[74] R. Taylor, L. Joskowicz, B. Williamson, A. Guéziec, A. Kalvin, P. Kazanzides, R. Van Vorhis, R. Kumar, A. Bzostek, A. Sahay, M. Boerner, and A. Lahmer, "Computer-integrated revision total hip replacement surgery: Concept and preliminary results," *Medical Image Analysis*, 1999. To appear.

[75] P. Kazanzides, B. Mittelstadt, B. Musits, W. Bargar, J. Zuhars, B. Williamson, P. Cain, and E. Carbone, "An integrated system for cementless hip replacement," *IEEE Engineering in Medicine and Biology Magazine*, vol. 14, May/June 1995.

[76] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," in *International Conference on Computer Vision*, (London), pp. 259–268, IEEE, June 1987.

[77] G. Grevera and J. Udupa, "Shape-based interpolation of $n$D grey scenes," *IEEE Transactions on Medical Imaging*, vol. 15, pp. 881–892, Dec. 1996.

[78] P. Danielsson, "Euclidean distance mapping," *Computer Graphics and Image Processing*, vol. 14, pp. 227–248, 1980.

[79] J. Boissonnat, "Shape reconstruction from planar cross sections," *Computer Vision, Graphics and Image Processing*, vol. 44, no. 1, pp. 1–29, 1988.

[80] A. Shaw and E. Schwartz, "Automatic construction of polyhedral surfaces from voxel representations," Tech. Rep. 381, New York University, June 1988. Robotics Report No. 158.

[81] G. Barequet and M. Sharir, "Piecewise-linear interpolation between polygonal slices," in *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pp. 93–102, 1994.

[82] C. Bajaj, E. Coyle, and K. Lin, "Arbitrary topology shape reconstruction from planar cross sections," *Graphical Models and Image Processing*, vol. 58, no. 6, pp. 524–543, 1996.

[83] E. Keppel, "Approximating complex surfaces by triangulation of contour lines," *IBM Journal of Research*, vol. 19, pp. 2–11, 1975.

[84] A. Guéziec, “Locally toleranced polygonal surface simplification,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, pp. 178–199, April-June 1999.

[85] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. Mac Graw Hill, 1989.

[86] J. O’Rourke, *Computational Geometry in C.* Cambridge University Press, 1994.

[87] A. Guéziec, P. Kazanzides, R. Taylor, B. Williamson, and D. Lord, “Anatomy based registration of CT-scan and X-ray fluoroscopy data for intra-operative guidance of a surgical robot,” in *Medical Imaging: Image Processing* (K. Hanson, ed.), vol. 3338-07, SPIE, Feb. 1998.

[88] J. Holladay, “Smoothest curve approximation,” *Math. Tables Aids Computation*, vol. 11, pp. 233–243, 1957.

[89] A. Guéziec, “Spline curves and surfaces for data modeling,” in *Advances in Morphometrics* (L. Marcus, M. Corti, A. Loy, G. Naylor, and D. Slice, eds.), NATO ASI Series A: Life Sciences vol. 284., Plenum: New York. x1v+587 pp, 1996.

[90] I. Cohen, L. Cohen, and N. Ayache, “Using deformable surfaces to segment 3-D images and infer differential structures,” *Computer Vision, Graphics, and Image Processing: Image Understanding*, vol. 56, no. 2, 1992.

[91] M. P. do Carmo, *Differential Geometry of Curves and Surfaces*. Englewood Cliffs: Prentice-Hall, 1976.

[92] A. Guéziec, *Reconnaissance Automatique de Surfaces et Courbes Gauches. Application à l’Analyse d’Images Volumiques*. Ph.D. thesis, Université de Paris XI à ORSAY, Feb. 1993. Thèse de Doctorat d’Université.

[93] X. Pennec and J. Thirion, “Validation of 3D registration methods based on points and frames,” in *International Conference of Computer Vision*, pp. 557–562, IEEE, 1995.

[94] A. Guéziec, X. Pennec, and N. Ayache, “Medical image registration using geometric hashing,” *IEEE Computational Science and Engineering Magazine*, vol. 4, pp. 29–41, Oct. 1997.

[95] F. Leitner and P. Cinquin, “Dynamic segmentation: Detecting complex topology 3D-object,” in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 13, pp. 295–296, 1991.

[96] J.-. Lachaud and A. Montanvert, “Deformable meshes with automated topology changes for coarse-to-fine three-dimensional surface extraction,” *Medical Image Analysis*, vol. 3, no. 2, pp. 187–207, 1999. Oxford University Press.

[97] T. McInerney and D. Terzopoulos, “Medical image segmentation using topologically adaptable surfaces,” in *CVRMed-MRCAS: International Conference on medical computer vision, virtual reality, robotics and computer assisted surgery*, vol. 1205 of *Lecture Notes in Computer Science*, pp. 23–32, Springer, 1997.

[98] G. Taubin, “A signal processing approach to fair surface design,” in *Siggraph*, (Los Angeles), pp. 351–358, ACM, August 1995.

[99] M. Desbrun, M. Meyer, P. Schroeder, and A. Barr, “Implicit fairing of irregular meshes using diffusion and curvature flow,” in *Siggraph*, (Los Angeles), pp. 317–324, ACM, August 1999.

[100] M. Gopi and D. Manocha, “A unified approach for simplifying polygonal and spline models,” in *Visualization’98*, (Raleigh, NC.), pp. 271–278, and 543, IEEE, Oct. 1998.

[101] R. Ronfard and J. Rossignac, “Full-range approximation of triangulated polyhedra,” *Computer Graphics Forum*, vol. 15, no. 3, pp. C67–C76, 1996. Proc. Eurographics’96.

[102] A. Kalvin and R. Taylor, “Superfaces: Polygonal mesh simplification with bounded error,” *Computer Graphics and Applications*, vol. 16, pp. 64–77, May 1996.

[103] C. Bajaj and D. Schikore, “Error-bounded reduction of triangle meshes with multivariate data,” in *Visual Data Exploration and Analysis III* (G. G. Grinstein and R. F. Erbacher, eds.), vol. 2656, pp. 34–45, SPIE, Mar. 1996.

[104] J. Cohen, A. Varshney, D. Manocha, G. Turk, H. Weber, P. Agarwal, F. Brooks, and W. Wright, “Simplification envelopes,” in *Siggraph*, pp. 119–128, ACM, Addison Wesley, August 1996.

[105] J. Cohen, M. Olano, and D. Manocha, “Appearance-preserving simplification,” in *Siggraph*, (Orlando), pp. 115–122, ACM, July 1998.

[106] M. DeHaemer and M. Zida, “Simplification of objects rendered by polygonal approximations,” *Computer & Graphics*, vol. 15, no. 2, pp. 175–184, 1991.

[107] G. Turk, “Retiling polygonal surfaces,” *Computer Graphics (Proc. Siggraph)*, vol. 26, pp. 55–64, July 1992.

[108] W. Schroeder, J. Zarge, and W. Lorensen, “Decimation of triangular meshes,” in *ACM Siggraph*, vol. 26-2, pp. 65–70, July 1992.

[109] P. Hinker and C. Hansen, “Geometric optimization,” in *Visualization 93*, pp. 189–195, IEEE, 1993.

[110] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, “Mesh optimization,” in *Siggraph*, (Anaheim), pp. 19–25, ACM, July 1993.

[111] J. Rossignac and P. Borrel, “Multi-resolution 3D approximations for rendering,” in *Modeling in Computer Graphics* (B. Falcidieno and T. Kunii, eds.), pp. 455–465, Springer-Verlag, 1993.

[112] P. Agarwal and S. Suri, “Surface approximation and geometric partitions,” in *Proceedings of the Fifth Symposium on Discrete Algorithms*, (Arlington, VA), pp. 24–33, January 1994.

[113] B. Hamann, “A data reduction scheme for triangulated surfaces,” *Computer Aided Geometric Design*, vol. 11, no. 2, pp. 197–214, 1994.

[114] A. Gourdon, “Simplification of irregular surface meshes in 3D medical images,” in *Computer Vision, Virtual Reality and Robotics in Medicine*, no. 905 in Lecture Notes in Computer Science, (Nice), pp. 413–419, April 1995.

[115] R. Klein, G. Liebich, and W. Strasser, “Mesh reduction with error control,” in *Visualization’96* (Yagel and Nielson, eds.), pp. 311–318, IEEE, October 1996.

[116] T. He, L. Hong, A. Varshney, and S. Wang, “Controlled topology simplification,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 2, pp. 171–184, June 1996.

[117] M.-E. Algorri and F. Schmitt, “Mesh simplification,” *Computer Graphics Forum*, vol. 15, no. 3, pp. C77–C86, 1996. Proc. Eurographics’96.

[118] M. Soucy, G. Godin, and M. Rioux, “A texture-mapping approach for the compression of colored 3D triangulations,” *The Visual Computer*, vol. 12, pp. 503–514, 1996.

[119] M. Garland and P. Heckbert, “Surface simplification using quadric error metrics,” in *Siggraph*, (Los Angeles), pp. 209–216, ACM, August 1997.

[120] J. Popovic and H. Hoppe, “Progressive simplicial complexes,” in *Siggraph*, (Los Angeles), pp. 217–224, ACM, August 1997.

[121] P. Veron and J. Leon, “Static polyhedron simplification using error measurements,” *Computer Aided Design*, vol. 29, pp. 287–298, April 1997.

[122] P. Lindstrom and G. Turk, “Fast and memory efficient polygonal simplification,” in *Visualization 98*, pp. 279–285, IEEE, October 1998.

[123] A. Guéziec, “Surface simplification with variable tolerance,” in *Second Annual International Symposium on Medical Robotics and Computer Assisted Surgery*, (Baltimore, MD), pp. 132–139, November 1995.

[124] T. Gieng, B. Hamann, K. Joy, G. Schussman, and I. Trotts, “Constructing hierarchies for triangle meshes,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 4, pp. 145–161, April-June 1996.

[125] R. Taylor, H. Paul, P. Kazanzides, B. Mittelstadt, W. Hanson, J. Zuhars, B. Williamson, B. Musits, E. Glassman, and W. Bargar, “An image-directed robotic system for precise orthopaedic surgery,” *IEEE Transactions on Robotics and Automation*, vol. 10, no. 3, pp. 261–275, 1994.

[126] B. Mittelstadt, P. Kazanzides, J. Zuhars, B. Williamson, P. Cain, F. Smith, and W. Bargar, “The evolution of a surgical robot from prototype to human clinical use,” in *Computer-Integrated Surgery* (R. H. Taylor *et al.*, eds.), pp. 397–407, MIT Press, 1996.

[127] A. Guéziec, “Registering intra-operative x-rays images with pre-operative data to simulate post-operative x-ray images,” in *2nd Israeli Symposium on Computer-Aided Surgery, Medical Robotics and Medical Imaging*, May 1999.

[128] A. Kaufman, M. Brady, B. Lorensen, F. Kitson, and H. Pfister, “Why is real-time volume rendering no longer a year away,” in *Visualization’98*, (Raleigh, NC.), pp. 497–499, IEEE, Oct. 1998.

# CHAPTER 8
# Image Registration

J. Michael Fitzpatrick
*Vanderbilt University*

Derek L. G. Hill
*King's College London*

Calvin R. Maurer, Jr.
*University of Rochester*

## Contents

## 8.1 Introduction

Registration is the determination of a geometrical transformation that aligns points in one view of an object with corresponding points in another view of that object or another object. We use the term "view" generically to include a three-dimensional image, a two-dimensional image, or the physical arrangement of an object in space. Three-dimensional images are acquired by tomographic modalities, such as computed tomography (CT), magnetic resonance (MR) imaging, single-photon emission computed tomography (SPECT), and positron emission tomography (PET). In each of these modalities, a contiguous set of two-dimensional slices provides a three-dimensional array of image intensity values. Typical two-dimensional images may be x-ray projections captured on film or as a digital radiograph or projections of visible light captured as a photograph or a video frame. In all cases, we are concerned primarily with digital images stored as discrete arrays of intensity values. In medical applications, which are our focus, the object in each view will be some anatomical region of the body. (See Volume I of this handbook for a discussion of medical imaging modalities.) The two views are typically acquired from the same patient, in which case the problem is that of *intrapatient registration*, but *interpatient registration* has application as well (see Chapter 17).

### *8.1.1 Operational goal of registration*

From an operational view, the inputs of registration are the two views to be registered; the output is a geometrical transformation, which is merely a mathematical mapping from points in one view to points in the second. To the extent that corresponding points are mapped together, the registration is successful. The determination of the correspondence is a problem specific to the domain of objects being imaged, which is, in our case, the human anatomy. To make the registration beneficial in medical diagnosis or treatment, the mapping that it produces must be applied in some clinically meaningful way by a system that will typically include registration as a subsystem. The larger system may combine the two registered images by producing a reoriented version of one view that can be "fused" with the other. This fusing of two views into one may be accomplished by simply summing intensity values in two images, by imposing outlines from one view over the gray levels of the other, or by encoding one image in hue and the other in brightness in a color image. Regardless of the method employed, image fusion should be distinguished from image *registration*, which is a necessary first step before fusion can be successful.

The larger system may alternatively use the registration simply to provide a pair of movable cursors on two electronically displayed views linked via the registering transformation so that the cursors are constrained to visit corresponding points. This latter method generalizes easily to the case in which one view is the physical patient and one of the movable "cursors" is a physical pointer held by the surgeon. The registration system may be part of a robotically controlled treatment system whose guidance is based on registration between an image and the physical anatomy. Drills, for example, may be driven robotically through bone by following a path determined in CT and registered to the physical bone. Gamma rays produced by a linear accelerator or by radioactive isotopes may be aimed at tissue that is visible in MR but hidden from view during treatment with the aiming being accomplished via automatic calculations based on a registering transformation. Registration also serves as a first step in multimodal segmentation algorithms that incorporate information from two or more images in determining tissue types (see Chapter 2). Fusion, linked cursors, robotic controls, and multimodal segmentation algorithms exploit knowledge of a geometrical relationship between two registered views in order to assist in diagnosis or treatment. Registration is merely the determination of that relationship. The goal of registration is thus simply to produce as output a geometrical transformation that aligns corresponding points and can serve as input to a system further along in the chain from image acquisition to patient benefit.

### *8.1.2 Classification of registration methods*

There are many image registration methods, and they may be classified in many ways [1–3]. Maintz has suggested a nine-dimensional scheme that provides an excellent categorization [1]. We condense it slightly (see below) into the following eight categories: image dimensionality, registration basis, geometrical transformation, degree of interaction, optimization procedure, modalities, subject, and object [1].

"Image dimensionality" refers to the number of geometrical dimensions of the image spaces involved, which in medical applications are typically three-dimensional but sometimes two-dimensional. The "registration basis" is the aspect of the two views used to effect the registration. For example, the registration might be based on a given set of point pairs that are known to correspond or the basis might be a set of corresponding surface pairs. Other loci might be used as well, including lines or planes (a special case of surfaces). In some cases, these correspondences are derived from objects that have been attached to the anatomy expressly to facilitate registration. Such objects include, for example, the stereotactic frame and point-like markers, each of which have components designed to be clearly visible in specific imaging modalities. Registration methods that are based on such attachments are termed "prospective" or "extrinsic" methods and are in contrast with the so-called "retrospective" or "intrinsic" methods, which rely on anatomic features only. Alternatively, there may be no known correspondences as input. In that case, intensity patterns in the two views will be matched, a basis that we call "Intensity."

The category, "geometrical transformation", is a combination of two of Maintz's categories, the "nature of transformation" and the "domain of transformation". It refers to the mathematical form of the geometrical mapping used to align points in one space with those in the other. We take up these transformations in detail in the next section. “Degree of interaction” refers to the control exerted by a human operator over the registration algorithm. The interaction may consist simply of the initialization of certain parameters, or it may involve adjustments throughout the registration process in response to visual assessment of the alignment or to other measures of intermediate registration success. The ideal situation is the fully automatic algorithm, which requires no interaction. “Optimization procedure” refers to the standard approach in algorithmic registration in which the quality of the registration is estimated continually during the registration procedure in terms of some function of the images and the mapping between them. The optimization procedure is the method, possibly including some degree of interaction, by which that function is maximized or minimized. The ideal situation here is a closed-form solution which is guaranteed to produce the global extremum. The more common situation is that in which a global extremum is sought among many local ones by means of iterative search.

“Modalities” refers to the means by which the images to be registered are acquired. Examples are listed at the beginning of this chapter. Registration methods designed for like modalities are typically distinct from those appropriate for differing modalities. Registration between like modalities, such as MR-MR, is called “intramodal” or “monomodal” registration; registration between differing modalities, such as MR-PET, is called “intermodal” or “multimodal” registration. “Subject” refers to patient involvement and comprises three subcategories: intrapatient, interpatient, and atlas, the latter category comprising registrations between patients and atlases, which are themselves typically derived from patient images (e.g., [4]). “Object” refers to the particular region of anatomy to be registered (e.g., head, liver, vertebra).

To build a registration hierarchy based on these eight categorizations, one categorization must be placed at the top level, which, in the organization of this chapter, is the registration basis. The three categories of registration basis mentioned above are examined in three major sections below: Section 8.3, Point-based methods, Section 8.4, Surface-based methods, and Section 8.5, Intensity-based methods. Before we begin these discussions of the basis for registration, however, we take up in the next section the category of geometrical transformation. We examine in detail the rigid transformations and other transformations that preserve the straightness of lines. These transformations are most appropriate for intrapatient registration of rigid anatomy. The remaining transformations, the “curved” transformations, are used for interpatient registration, for the registration of anatomy to an atlas, and for intrapatient registration of tissue that has been deformed between the acquisition of the two views. These situations are examined in Chapters 3 and 17.

## 8.2 Geometrical transformations

Each view that is involved in a registration will be referred to a coordinate system, which defines a space for that view. Our definition of registration is based on geometrical transformations, which are mappings of points from the space $X$ of one view to the space $Y$ of a second view. The transformation $\mathcal{T}$ applied to a point in $X$ represented by the column vector $\mathbf{x}$ produces a transformed point $\mathbf{x}'$,

$$\mathbf{x}' = \mathcal{T}(\mathbf{x}) \; . \tag{8.1}$$

If the point $\mathbf{y}$ in $Y$ corresponds to $\mathbf{x}$, then a successful registration will make $\mathbf{x}'$ equal, or approximately equal, to $\mathbf{y}$. Any nonzero displacement $\mathcal{T}(\mathbf{x}) - \mathbf{y}$ is a registration error. The set of all possible $\mathcal{T}$ may be partitioned into rigid and nonrigid transformations with the latter transformations further divided into many subsets. This top-level division makes sense in general because of the ubiquity of rigid, or approximately rigid, objects in the world. It makes sense for medical applications in particular because of the rigid behavior of many parts of the body, notably the bones and the contents of the head (not during surgery). It is also a simple class with only six parameters completely specifying a rigid transformation in three dimensions. (We note here that, while one and two-dimensional motion is possible, such limited motion is sufficiently rare that we will ignore it in this chapter.)

### *8.2.1 Rigid transformations*

Rigid transformations, or rigid mappings, are defined as geometrical transformations that preserve all distances. These transformations also preserve the straightness of lines (and the planarity of surfaces) and all nonzero angles between straight lines. Registration problems that are limited to rigid transformations are called *rigid registration* problems. Rigid transformations are simple to specify, and there are several methods of doing so. In each method, there are two components to the specification, a translation and a rotation. The translation is a three-dimensional vector $\mathbf{t}$ that may be specified by giving its three coordinates $t_x, t_y, t_z$ relative to a set of $x, y, z$ Cartesian axes or by giving its length and two angles to specify its direction in polar spherical coordinates. There are many ways of specifying the rotational component, among them Euler angles, Cayley-Klein parameters, quaternions, axis and angle, and orthogonal matrices [5–9]. In our discussions we will utilize orthogonal matrices. With this approach, if $\mathcal{T}$ is rigid, then

$$\mathbf{x}' = R\mathbf{x} + \mathbf{t} \; , \tag{8.2}$$

where $R$ is a $3\times3$ orthogonal matrix, meaning that $R^tR = RR^t = I$ (the identity). Thus $R^{-1} = R^t$. This class of matrices includes both the *proper* rotations, which describe physical transformations of rigid objects, and *improper* rotations, which do not. These latter transformations both rotate and reflect rigid objects, so that, for example, a right-handed glove becomes a left-handed one. Improper rotations can be eliminated by requiring $\det(R) = +1$.

Proper rotations can be parameterized in terms of three angles of rotation, $\theta_x, \theta_y, \theta_z$, about the respective Cartesian axes, the so-called "Euler angles." The rotation angle about a given axis is, with rare exception, considered positive if the rotation about the axis appears clockwise as viewed from the origin while looking in the positive direction along the axis. The rotation of an object (as opposed to the coordinate system to which it is referred) about the $x$, $y$, and $z$ axes, in that order leads to

$$
\begin{aligned}
R &= \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix} \\
&= \begin{bmatrix} \cos\theta_y\cos\theta_z & -\cos\theta_x\sin\theta_z+\sin\theta_x\sin\theta_y\cos\theta_z & \sin\theta_x\sin\theta_z+\cos\theta_x\sin\theta_y\cos\theta_z \\ \cos\theta_y\sin\theta_z & \cos\theta_x\cos\theta_z+\sin\theta_x\sin\theta_y\sin\theta_z & -\sin\theta_x\cos\theta_z+\cos\theta_x\sin\theta_y\sin\theta_z \\ -\sin\theta_y & \sin\theta_x\cos\theta_y & \cos\theta_x\cos\theta_y \end{bmatrix}
\end{aligned}
$$

with the three matrices in the first line representing the rotations $R_z(\theta_z)$, $R_y(\theta_y)$, and $R_x(\theta_x)$ about $z$, $y$, and $x$, respectively (in reverse order because they are applied from right to left). Other angular parameterizations are sometimes used, including all permutations of the order of $R_x$, $R_y$, and $R_z$. General rotations can also be produced by three rotations about only two of the Cartesian axes provided that successive rotations are about distinct axes. The most common of these is $R = R_z(\theta_{z2})R_x(\theta_x)R_z(\theta_{z1})$.

The *axis-and-angle* specification provides a parameterization of $R$ comprising a unit vector representing an axis of rotation $\hat{\omega}$ and a single angle of rotation $\theta$ about that axis. The rotational axis $\hat{\omega}$ is the (lone) real eigenvector of $R$. The rotational angle can be calculated from the relation $\cos\theta = (\text{trace } R - 1)/2$. With this parameterization,

$$
R = \begin{bmatrix} {\omega_x}^2 V + C & \omega_x\omega_y V - \omega_z S & \omega_x\omega_z V + \omega_y S \\ \omega_x\omega_y V + \omega_z S & {\omega_y}^2 V + C & \omega_y\omega_z V - \omega_x S \\ \omega_x\omega_z V - \omega_y S & \omega_y\omega_z V + \omega_x S & {\omega_z}^2 V + C \end{bmatrix}, \tag{8.3}
$$

where the $\omega_x, \omega_y, \omega_z$ are the components of $\hat{\omega}$, $V = 1 - \cos\theta$, $C = \cos\theta$, and $S = \sin\theta$ [9].

The *quaternion* specification provides a parameterization that is closely related to the axis and angle. The quaternion is a vector $\mathbf{q}$ consisting of four elements, $q_0, q_x, q_y, q_z$, and obeying special rules for multiplication [7, 8]. A quaternion of unit length can be used to represent a rotation, where the elements are equal respectively to $\cos(\theta/2)$, $\omega_x \sin(\theta/2)$, $\omega_y \sin(\theta/2)$, and $\omega_z \sin(\theta/2)$ [7, 8]. The parameterization in Eq. (8.3) can be converted to a quaternion parameterization with the identities, $2\sin(\theta/2)\cos(\theta/2) = \sin\theta$ and $2\sin^2(\theta/2) = 1 - \cos\theta$,

$$
R = \begin{bmatrix} q_0^2 + q_x^2 - q_y^2 - q_z^2 & 2q_xq_y - 2q_0q_z & 2q_xq_z + 2q_0q_y \\ 2q_xq_y + 2q_0q_z & q_0^2 - q_x^2 + q_y^2 - q_z^2 & 2q_yq_z - 2q_0q_x \\ 2q_xq_z - 2q_0q_y & 2q_yq_z + 2q_0q_x & q_0^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix}. \tag{8.4}
$$

### 8.2.2 Nonrigid transformations

Nonrigid transformations are important not only for applications to nonrigid anatomy, but also for interpatient registration of rigid anatomy and intrapatient registration of rigid anatomy when there are nonrigid distortions in the image acquisition procedure. In all cases, it is preferable to choose transformations that have physical meaning, but in some cases, the choice is made on the basis of convenient mathematical properties.

#### 8.2.2.1 Scaling transformations

The simplest nonrigid transformations are rigid except for scaling,

$$\mathbf{x}' = RS\mathbf{x} + \mathbf{t} \tag{8.5}$$

and

$$\mathbf{x}' = SR\mathbf{x} + \mathbf{t} \ , \tag{8.6}$$

where $S = \text{diag}(s_x, s_y, s_z)$ is a diagonal matrix whose elements represent scale factors along the three coordinate axes. Because $RS$ is not in general equal to $SR$, these equations represent two different classes of transformations. Such transformations may be needed to compensate for calibration errors in image acquisition systems. They are appropriate, for example, when gradient strengths are in error in MR. The diagonal elements of $S$ then become the respective correction factors for the $x$, $y$, and $z$ gradients (see Section 8.2.3, below). If the scaling is isotropic, the transformation is a *similarity transformation*,

$$\mathbf{x}' = sR\mathbf{x} + \mathbf{t} \ , \tag{8.7}$$

where $s$ is a positive scalar, sometimes known as a “dilation” (for values less than one as well). This transformation preserves the straightness of lines and the angles between them. Both Eq. (8.5) and Eq. (8.6) reduce to Eq. (8.7) when $s_x = s_y = s_z = s$. The coupling of scaling with the rigid transformation is effective when registrations must account for erroneous or unknown scales in the image acquisition process.

#### 8.2.2.2 Affine transformations

The scaling transformations are special cases of the more general *affine* transformation,

$$\mathbf{x}' = A\mathbf{x} + \mathbf{t} \ , \tag{8.8}$$

in which there is no restriction on the elements $a_{ij}$ of the matrix $A$. The affine transformation preserves the straightness of lines, and hence, the planarity of surfaces, and it preserves parallelism, but it allows angles between lines to change. It is an appropriate transformation class when the image may have been skewed during acquisition as, for example, when the CT gantry angle is incorrectly recorded.

The affine transformations and their associated special cases are sometimes represented by means of *homogeneous coordinates*. In this representation, both $A$ and $\mathbf{t}$ are folded into one 4×4 matrix $M$ whose elements are defined as follows: $m_{ij} = a_{ij}, i = 1, 2, 3, j = 1, 2, 3$; $m_{i4} = t_i, i = 1, 2, 3$; $m_{4j} = 0, j = 1, 2, 3$; and $m_{44} = 1$. To accomplish the transformation, augmented vectors $\mathbf{u}$ and $\mathbf{u}'$ are used for which $u_i = x_i$ and $x'_i = u'_i$ for $i = 1, 2, 3$ and $u_4 = u'_4 = 1$.

$$\mathbf{u}' = \begin{pmatrix} u'_1 \\ u'_2 \\ u'_3 \\ 1 \end{pmatrix} = M\mathbf{u} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & t_1 \\ a_{21} & a_{22} & a_{23} & t_2 \\ a_{31} & a_{32} & a_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ 1 \end{pmatrix} . \tag{8.9}$$

While the use of homogeneous coordinates does not produce any extra power or generality for rigid transformations, it does simplify notation, especially when rigid transformations must be combined with projective transformations, which we consider next.

### *8.2.2.3 Projective transformations*

So far, the nonrigid transformations that we have considered, all of which are affine transformations, preserve parallelism. The more general nonrigid transformations include the *projective* transformations, which preserve the straightness of lines and planarity of surfaces, and the *curved* transformations, which do not. The projective transformations, which have the form,

$$\mathbf{x}' = (A\mathbf{x} + \mathbf{t})/(\mathbf{p} \cdot \mathbf{x} + \alpha) , \tag{8.10}$$

can be written simply in homogeneous coordinates,

$$\mathbf{u}' = \begin{pmatrix} u'_1 \\ u'_2 \\ u'_3 \\ u'_4 \end{pmatrix} = M\mathbf{u} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & t_1 \\ a_{21} & a_{22} & a_{23} & t_2 \\ a_{31} & a_{32} & a_{33} & t_3 \\ p_1 & p_2 & p_3 & \alpha \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ 1 \end{pmatrix} , \tag{8.11}$$

where, as for the affine transformation, $u_i = x_i$ for $i = 1, 2, 3$, $u_4 = 1$, and $m_{ij} = a_{ij}, i = 1, 2, 3, j = 1, 2, 3$, but $u'_4$ is no longer necessarily equal to 1, $m_{4j} = p_j, j = 1, 2, 3$, $m_{44} = \alpha$, and $x'_i = u'_i/u'_4, i = 1, 2, 3$. The linearity of Eq. (8.11) can provide considerable simplification for the projective transformations and the *perspective projections*. This latter class of transformations relate three-dimensional anatomy to two-dimensional planar images acquired by means of a camera and are taken up in the next section.

*8.2.2.4 Perspective transformations*

Images obtained by x-ray projection, endoscopy, laparoscopy, microscopy, and direct video acquisition are all two-dimensional views of three-dimensional objects rendered by means of projecting light rays or x-rays from a three-dimensional scene onto a two-dimensional plane. The geometrical transformation, which we call a *perspective projection*, produced by each of these modalities is equivalent to that of photography.

These perspective projections are a subset of the projective transformations of Eqs. (8.10) and (8.11). The projective transformations, unlike the perspective projections, do not, in general, transform $\mathbf{x}$ to a plane. Furthermore, the affine portion of the transformation is typically assumed to be the identity for perspective projections. Specializing now to perspective projections, we let $f = 1/|\mathbf{p}|$ in Eq. (8.10), and let $\hat{\mathbf{p}}$ be a unit vector in the direction of the projection axis, $\mathbf{p}$. These substitutions lead to

$$\mathbf{x}' = f\mathbf{x}/(\mathbf{x} \cdot \hat{\mathbf{p}} + \alpha f) \; . \tag{8.12}$$

If $\alpha$ is nonzero, then Eq. (8.12) does not, in fact, transform $\mathbf{x}$ to a plane and, hence, is not a perspective projection. Perspective projection can be produced, however, by zeroing the component of $\mathbf{x}'$ in the direction of $\mathbf{p}$:

$$\mathbf{x}' \longrightarrow \mathbf{x}' - (\mathbf{x}' \cdot \hat{\mathbf{p}})\hat{\mathbf{p}} \; . \tag{8.13}$$

Equation (8.12) and substitution (8.13) give the general form of the transformation produced when a photograph of a three-dimensional scene is acquired with a “pinhole camera,” which is a camera in which a small hole substitutes for the lens system. Fig. 8.1 shows a schematic camera. A ray from a point $\mathbf{x}$ in the scene is projected through the pinhole onto a film screen, which is perpendicular to the axis of projection $\mathbf{p}$ and located at a distance $f$ from the pinhole. Fortunately, all the systems mentioned above can be approximated by the pinhole camera system by identifying the unique point relative to the lens system through which light rays travel undeflected, or the point from which the x rays emanate. That point, also known as the “center of perspectivity” [10], plays the role of the pinhole. Because the film image is inverted, it is convenient to treat instead an equivalent upright “image” located in front of the camera, which is also perpendicular to $\mathbf{p}$ and located the same distance $f$ from the pinhole, as shown in the figure. The transformed point, $\mathbf{x}'$, of Eq. (8.12), followed by substitution (8.13) lies in that plane.

Figure 8.2 shows the relationship between the parameters in Eq. (8.12) and the imaging system both (a) for lens systems and (b) for x-ray systems. The geometric difference is that $\mathbf{x} \cdot \hat{\mathbf{p}} + \alpha f$ is greater than $f$ for lens systems and less than $f$ for x-ray systems. The parameter $f$ is called the *focal length*, or, alternatively, the “camera constant” or “principal distance.” The name “focal length” is derived from lens systems. It is meant to imply that the lens is adjusted so that all light

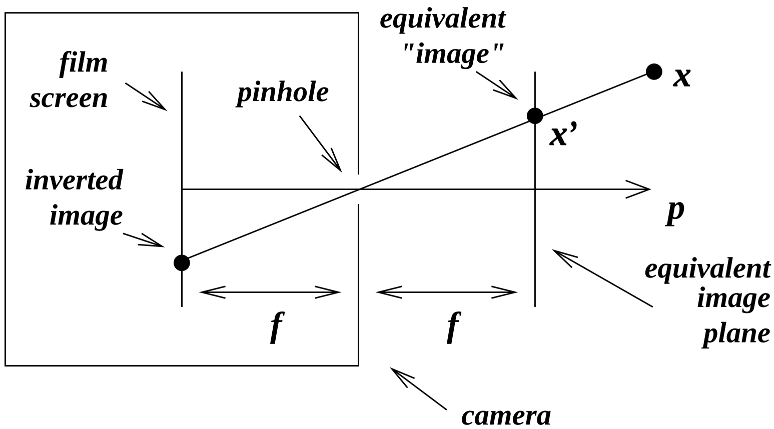



**Figure 8.1:** A schematic of a pinhole camera. A single point $\mathbf{x}$ in the scene is shown being imaged. The projection axis $\mathbf{p}$ points to the right. The image is produced within the camera on a film screen that is perpendicular to $\mathbf{p}$ and located a distance $f$ from the pinhole. Analysis is simplified by treating the transformation as if an equivalent, but upright, image were produced instead on a plane placed outside the camera at the same orientation and same distance from the pinhole. The point $\mathbf{x}$ is transformed to $\mathbf{x}'$ on this plane.

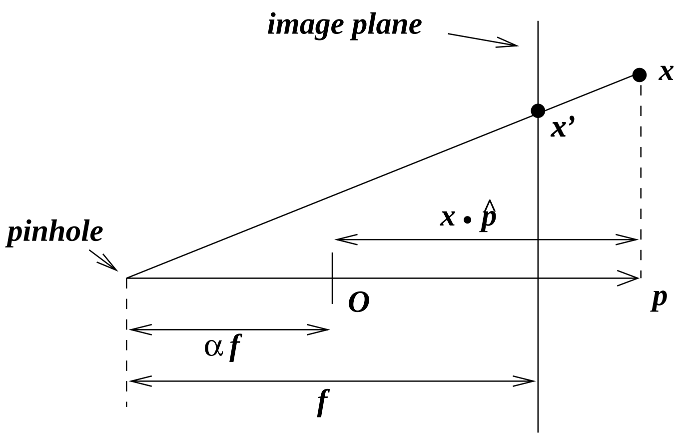
emanating from a given point in front of the camera and passing through its lens will be focused to a single point on a screen located at that distance from the effective pinhole. The focusing is only approximate and varies in quality with the distance of the anatomy from the lens. The value of $\alpha$ in Eq. (8.12) is determined by the placement of the origin. Typically, the origin is placed at the pinhole, for which $\alpha = 0$, or at the intersection of $\mathbf{p}$ and the image plane, for which $\alpha = 1$.

#### *8.2.2.5 Curved transformations*

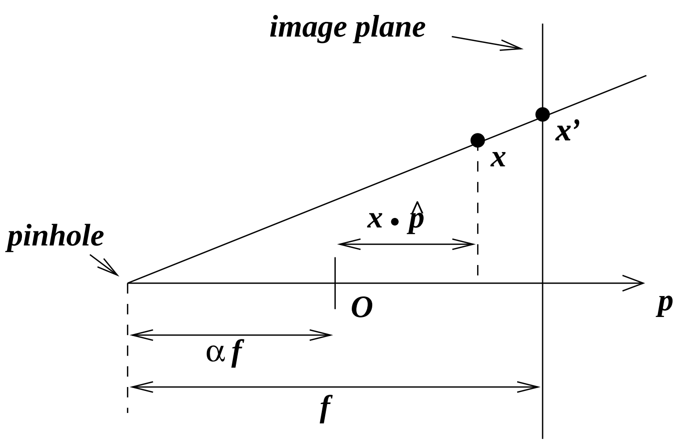
Curved transformations are those that do not preserve the straightness of lines. In curved transformations, the simplest functional form for $\mathcal{T}$ is a polynomial in the components of $\mathbf{x}$ [11, 12],

$$\mathbf{x}' = \sum_{ijk}^{IJK} \mathbf{c}_{ijk} x^i y^j z^k \,, \tag{8.14}$$

where $\mathbf{c}_{ijk}$ is the three-element vector of coefficients for the $i, j, k$ term in the polynomial expression for the three components $x', y', z'$ of $\mathbf{x}'$. Modifications may be employed that include all terms for which $i+j+k \leq M$. These transformations are rarely used with values of $I$, $J$, and $K$ greater than 2 or M greater than 5 because of spurious oscillations associated with high-order polynomials.




(a) Pinhole Camera System. $\mathbf{x}\quad \hat{\mathbf{p}} + \alpha f > f$.




(b) X-ray Projection. $\mathbf{x}\quad \hat{\mathbf{p}} + \alpha f < f$.

**Figure 8.2:** Perspective projection for the pinhole camera system and for x-ray projection. The point $\mathbf{x}$ is projected to $\mathbf{x}'$. The origin of the coordinate system is located at $O$. The projection axis $\mathbf{p}$ points to the right. Typical values for $\alpha$ are 0 or 1.

These oscillations can be reduced or eliminated by employing piecewise polynomials. The resulting transformations are defined by first partitioning the space into a set of three-dimensional rectangles by means of three sets of cut planes, each perpendicular to one of the Cartesian axes. Then, within each rectangle $i, j, k$ of dimensions, $u_i, v_j, w_k$, $\mathcal{T} = P^{(x)}_{ijk}(x)P^{(y)}_{ijk}(y)P^{(z)}_{ijk}(z)$, where each $P()$ is a univariate polynomial of degree $m$ defined only over that rectangle. (Polynomial forms that are more complex than products of univariate polynomials in $x$, $y$, and $z$ are available as well [13].) By judicious choice of the polynomials' coefficients, it is possible to ensure that the polynomials join smoothly across the rectangle interfaces in the sense that $\mathcal{T}$ is $m - 1$ times continuously differentiable. Such transformations are called *splines*, and the corners $\mathbf{x}_{ijk}$ of the rectangles are called "knots." The properties of splines have been carefully studied in approximation theory [14]. The most common choice for $m$ is 3, producing the "cubic spline," for which $\mathcal{T}$ and its first and second derivatives are continuous. Splines of degree $m$ are often expressed in terms of a convenient basis set of polynomials of the same degree, called *B-splines* (see Chapter 6 for their definition). Noting that the coordinates of every $\mathbf{x}$ must be equal to $u_i \times (l + \alpha), v_j \times (m + \beta), w_k \times (n + \gamma)$, for some unique set $l, m, n, \alpha, \beta, \gamma$ where $l, m, n$ are integers and $0 \leq \alpha, \beta, \gamma \leq 1$, the expansion of the cubic spline in terms of the B-splines can be expressed as follows:

$$\mathbf{x}'(l, m, n, \alpha, \beta, \gamma) = \sum_{ijk} B_{i-l}(\alpha)B_{j-m}(\beta)B_{k-n}(\gamma)\mathbf{c}_{ijk}, \tag{8.15}$$

where each $B_q(t)$ is a segment of the cubic B-spline with $B_q(t) = 0$ unless $-1 \leq q \leq 2$. Simple algorithms are available for determining the form of the $B_q(t)$ (see for example [13]), but for the special case in which the knot spacing along a given direction is uniform, $u_i = u, v_j = v, w_k = w$, they have an especially simple form [13, 15, 16],

$$\begin{aligned} B_{-1}(t) &= (\ -t^3 + 3t^2 - 3t + 1)/6, \quad B_0(t) = (3t^3 - 6t^2 + 4)/6, \\ B_1(t) &= (-3t^3 + 3t^2 + 3t + 1)/6, \quad B_2(t) = \ \ t^3/6. \end{aligned} \tag{8.16}$$

A transformation that has been heavily used for two-dimensional problems is the *thin-plate spline*, which was originally called the "surface spline." This form was first proposed by Harder in 1972 [17] for designing aircraft wings. (Their mathematical properties are explored by Duchon [18], who coined the term "thin-plate spline," and Meinguet [19].) The thin-plate splines were first employed to describe deformations within the two-dimensional plane by Goshtasby in 1988 [20].[1] Goshtasby's formulation, which is now widely employed in the image processing literature, is as follows:

$$\mathbf{x}' = A\mathbf{x} + \mathbf{t} + \sum_{i}^{N} \mathbf{c}_i r_i^2 \ln r_i^2\ , \tag{8.17}$$

[1] For suggestions as to their use in comparing two-dimensional shapes see [21].

where $r_i = |\mathbf{x} - \mathbf{x}_i|$ and $\mathbf{x}_i$ is a control point. Unlike the rectangular grid of knots required for the cubic-splines, the control points can be placed arbitrarily, a feature that is of considerable advantage in the registration of medical images. For three-dimensional transformations, the thin-plate spline has a simpler form in which $r_i^2 \ln r_i^2$ in Eq. (8.17) is replaced by $r_i$. For both the two-dimensional and the three-dimensional forms, the affine portion of Eq. (8.17) is a necessary part of the transformation. Without this component there may be no set of $\mathbf{c}_i$ that satisfies the equation at all $N$ points. With the affine part included, it is always possible, by means of the imposition of a set of side conditions on the $\mathbf{c}_i$ (see Section 8.3.4), to ensure that a solution exists for any arrangement of points.

The thin-plate splines are examples of the more general category of *radial basis functions*, which, for two-dimensional and three-dimensional spaces, have the same form as Eq. (8.17), but with $r_i^2 \ln r_i^2$ replaced by a general function $f(r_i)$. Other examples of $f(r_i)$ that have been employed to interpolate among points in two or three dimensions (not necessarily for registration) include the "volume spline" $r_i^3$, the "multiquadric" $(r_i^2 + b)^{1/2}$ ($b > 0$), its inverse, and the Gaussian, $\exp(-r_i^2/a)$ [22]. Because none of these example functions has compact support, a change in any control point has an affect on the transformation at all points. Radial functions with compact support have been employed to limit this effect to points in a neighborhood surrounding a given control point [23, 24].

Other curved transformations have been employed, including solutions to the equations of continuum mechanics describing elastic and fluid properties attributed to the anatomy being registered [25–31]. These equations, which are derived from conservation of mass, momentum, and energy and from experimentally measured material properties, involve the displacement vector, $\mathbf{x}' - \mathbf{x}$, and the first and second spatial derivatives of its components. The nonrigid transformations that result from the numerical solution of these partial differential equations are appropriate for intrapatient registration when the anatomy is nonrigid, especially when surgical resection has changed its shape. In these cases, the major problem in registration is the determination of the material properties and forces that act on the tissue. With that information available, the solution of the differential equations may be carried out numerically by means of finite-difference or finite-element methods. These transformations have been also used for interpatient registration and for the closely related problem of mapping an atlas to a patient, and they are taken up in detail in Chapter 17.

### *8.2.3 Rectification*

Nonrigid transformations may be necessary in order to rectify images that are distorted in the acquisition process. Images are distorted when the transformation from physical object to image fails to have the ideal form assumed for the acquisition process. We define *rectification* as the process of transforming an acquired image into a "rectified" image, for which the resulting overall transformation from physical object to rectified image has the ideal form. It is possible to perform registration in the face of such distortions, but if the distortion can be determined, a more efficient approach is to perform rectification to remove the distortion as a separate preprocessing step before registration. Efficiency is achieved by this division of the problem into two steps because additional information from scanner calibrations, phantom images, or other corrective measures can be used to reduce the size of the space of transformations to be searched during registration.

While for three-dimensional images the ideal transformation is normally rigid, as given by Eq. (8.2), in some cases, as for example when registration is to be effected between serial images acquired on the same system within a short time, Eq. (8.7) may be acceptable. Then, the scaling $s$ can be expected to remain constant between acquisitions and can thus be set equal to one for the purposes of registration. The simplest three-dimensional rectification problem is compensation for an isotropic scaling. In this case, rectification can be accomplished by means of the transformation $\mathbf{x}' = s\mathbf{x}$, which is a special case of Eq. (8.7). A slightly more complex example is provided by gantry tilt in CT imaging. The resultant skew produced in a volume CT image can be removed by transformations of the form $\mathbf{x}' = A\mathbf{x}$, which is a special case of Eq. (8.8) (i.e., no translation).

For projection by a camera system, the ideal transformation is Eq. (8.12), or the simpler form given below by Eq. (8.30), but, in many cases, the projection image captured by a camera system is distorted in the image plane. Such distortions occur not only for light focused by lens systems but for x-ray images produced in conjunction with image intensifiers. That distortion may be rectified by means of a two-dimensional transformation within the image plane before registration is attempted. The required transformation is typically a member of the class of curved transformations. The rectification step in that case may employ any of the curved transformations given by Eqs. (8.14), (8.15), or (8.17). The transformation may be determined by acquiring an image of a phantom consisting of a regularly spaced grid of fiducial points arranged in a plane perpendicular to the projection axis. That image is then compared with the expected image based on a model of the phantom and the ideal transformation for the camera. The difference between the pattern of imaged points and the pattern of model points is used in a point-based registration to determine the transformation that will restore the regular pattern to the image (see Section 8.3.4). An important special case is the rotationally symmetric “pin cushion” distortion exhibited near the image periphery by wide-angle lenses, such as those employed for endoscopy, and by image intensifiers in digital radiographic systems (see Chapter 3, Section 3.2.2 of Volume 1 of this handbook). In this case, the two-dimensional transformation reduces to a one-dimensional transformation in the radial distance from a central point, and the distortion correction reduces to a simple univariate interpolation [32]. Once the transformation is determined, it can be employed to rectify any subsequent anatomical image acquired with the same acquisition setup. Thus, by means of the separate rectification step, the task of distortion correction is separated from the task of registering specific anatomical image pairs.

MR imaging is subject to geometric distortions whose rectification requires curved transformations. These distortions arise from nonuniform gradient fields due to imperfections in the scanner and from spatial variation in the static magnetic field that may be due either to imperfections in the scanner or to induced magnetization of the anatomy. Rectifying transformations can be developed for scanner imperfections by acquiring images of phantoms that include regularly spaced, three-dimensional grids of point-like objects in analogy to the two-dimensional rectification of camera images in projection imaging. For distortions arising from the susceptibility of the anatomy to magnetization, special protocols must be employed involving MR pulse sequences specifically designed for rectification [33,34]. These transformations have a special form in which all displacements are parallel to the so-called "readout" gradient. Because the magnitude of the displacements and their sense (forward or backward) varies from point to point, the resultant transformation is in general curved. Removal of these distortions by a rectification step can replace a complicated nonrigid registration problem with a rigid one.

For three-dimensional images, rectification may appear to be equivalent to registration between the image view and the physical view. In fact, the rectification problem is considerably simpler. The goal of registration from an image to physical space is to map points in the image view to the corresponding points in the physical view, while the goal of rectification is merely to bring all points in an image to within an ideal transformation of their corresponding physical points. The simplification is that the latter transformation may remain unknown. Thus, the goal of rectification is to find a transformation, which, *when composed with some ideal transformation,* will map corresponding points together. The determination of the unknown transformation is necessary to complete the registration process. A simple example is provided by errors in the magnitude of MR gradients. With such errors, the imaging process transforms physical points according to Eq. (8.6). The appropriate class of transformations for registering such distorted images to undistorted images is then Eq. (8.5) (see also page 473). If, however, the correcting scale factors, $s_x, s_y, s_z$, can be determined by means of calibration (see Algorithm 8.3 below), then rectification can be accomplished by applying Eq. (8.5) with $S = \text{diag}(s_x, s_y, s_z)$ and with any arbitrary rotation $R$ and translation $\mathbf{t}$. Typically $R = I$ and $\mathbf{t} = 0$ are chosen for simplicity, in which case, the rectifying transformation is $\mathbf{x}' = S\mathbf{x}$. The transformation between the resultant rectified image and the physical object now has the ideal form for the acquisition process, namely Eq. (8.2), as does the registering transformation between any two such rectified images.

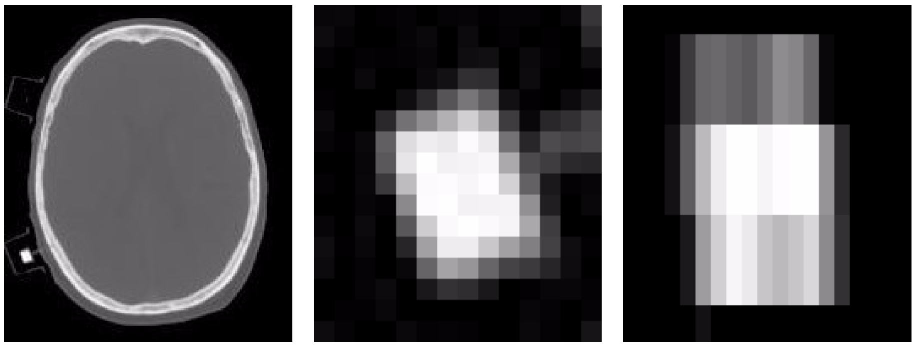
## 8.3 Point-based methods

If some set of corresponding point pairs can be identified *a priori* for a given pair of views, then registration can be effected by selecting a transformation that aligns the points. Because such points are taken as being reliable for the purposes of registration, they are called *fiducial points*, or *fiducials*. To be reliable, they must lie in clearly discernible features, which we will call *fiducial features*. The determination of a precise point within a feature is called *fiducial localization*. The transformation that aligns the corresponding fiducial points will then interpolate the mapping from these points to other points in the views.

The fiducial localization process may be based on interactive visual identification of anatomical landmarks, such as the junction of two linear structures, e.g., the central sulcus with the midline of the brain or the intersection of a linear structure with a surface, e.g., the junction of septa in an air sinus, etc. [35]. Alternatively, the feature may be a marker attached to the anatomy and designed to be accurately localizable by means of automatic algorithms. In either case, the chosen point will inevitably be erroneously displaced somewhat from its correct location. This displacement in the determination of the fiducial point associated with a fiducial feature is commonly called the *fiducial localization error* (**FLE**). Such errors will occur in both image spaces. They cannot ordinarily be observed directly, but they can be observed indirectly through the registration errors that they cause.

Marker-based registration has the considerable advantage over landmark-based registration that the fiducial feature is independent of anatomy. Automatic algorithms for locating fiducial markers can take advantage of knowledge of the marker's size and shape in order to produce a consistent fiducial point within it [36]. Typically, the fiducial point chosen by a localization algorithm will lie near its center. Hence, the point is often referred to as the fiducial "centroid". However, registration accuracy depends only on the degree to which the chosen points correspond in the two views. Random errors in the localized position will be caused by noise in the image and by the varying positioning of the marker relative to the voxel grid, but, for reasonable localization algorithms, the mean of the fiducial points chosen should be the same in the two views relative to a coordinate system fixed in the marker. Because of this consistency, the effective mean displacement, $\langle \mathbf{FLE} \rangle$, in each view is zero. (We use $\langle x \rangle$ to indicate the expected value of $x$.) The variance $\langle \mathrm{FLE}^2 \rangle$ may be appreciable, however. (Note that we use a bold font to indicate the vector displacement **FLE** and a normal font to indicate the magnitude FLE.)

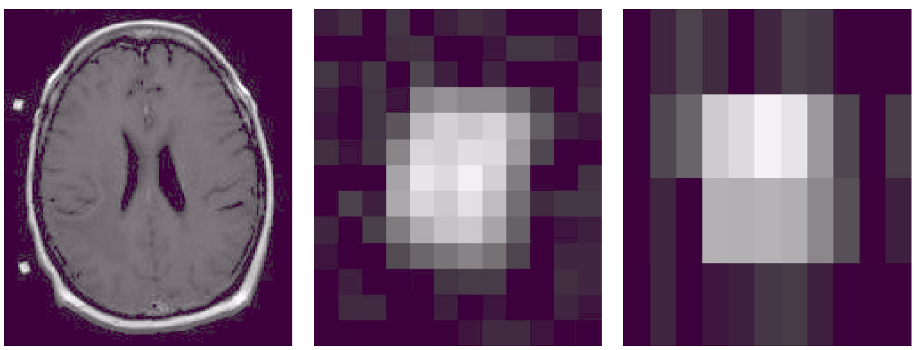
The goal of fiducial design and of the design of the accompanying fiducial localization algorithm is to produce a small variance. In general, as the marker volume becomes larger, and as the signal per volume produced in the scanner by its contents becomes larger, FLE will become smaller. Figure 8.3 shows sample images of fiducial markers. Because of the discrete nature of the digital images, it is important to use a marker that is larger than an image voxel. Figure 8.3 shows a hollow marker that is of cylindrical shape. Its inside dimensions are height, 5 mm



(a) Transaxial CT

(b) Magnified CT

(c) Magnified CT, coronal reformatting



(d) Transaxial MR

(e) Magnified MR

(f) Magnified MR, coronal reformatting

**Figure 8.3:** Sample CT and MR images of a fiducial marker. The marker is a cylinder of height 5 mm and diameter 7 mm. (a) Transverse CT of a human head with an attached marker at the lower left. (Part of a second marker is just visible at the upper left. The structures surrounding the markers are protective housings that are visible in CT but not MR.) Voxel dimensions, 0.65 by 0.65 by 3.0 mm. (b) Magnification of the marker in (a). (c) Magnification of the marker in (a) after reformatting into the coronal orientation. (d) Transverse MR of head with markers at upper and lower left. Voxel dimensions, 1.1 by 1.1 by 4.0 mm. (e) Magnification of the upper marker in (d). (f) Magnification of the upper marker in (d) after reformatting into the coronal orientation.

and diameter, 7 mm. The marker is filled with a liquid that is imageable in CT and MR [37]. Both CT and an MR (Spin-Echo, T1-weighted) transaxial images are shown in Fig. 8.3. The CT voxel dimensions are 0.65 by 0.65 by 3.0 mm; the MR voxel dimensions are 1.1 by 1.1 by 4.0 mm. The marker is larger than a CT voxel, but only slightly so in the slice direction. Thus, the rendition is good within a CT slice, as seen in (b), but poor in the slice direction, as seen in (c). The MR voxels are larger than the CT voxels. Thus, the renditions are relatively poor, both within an MR slice, as seen in (e) and in the slice direction, as seen in (f).

The rendition of the marker in an image is related to the potential accuracy with which it can be localized. Smaller markers, as measured in voxels, will be more poorly localized than larger ones. The shape of a marker that is smaller than a voxel cannot be represented at all. More importantly, with regard to registration accuracy, such a marker can be situated entirely within a voxel with the result that the brightness "pattern" is independent of the marker's position within the voxel. Markers that span two or more voxels will ordinarily (absent reconstruction artifacts) produce a signal for each occupied voxel that is a monotonically increasing function of the volume of intersection of the voxel and the signal-producing contents of the marker. This effect is illustrated in Fig. 8.4. To first order, ignoring noise and reconstruction artifacts, the function will be linear,

$$I = aV + I_0 \,, \tag{8.18}$$

where $I$ is the voxel intensity and $V$ is the volume of intersection. $I_0$ is the intensity for an empty voxel, sometimes called a "background" voxel. While $I_0 = 0$ for most modalities, it is sometimes nonzero (CT, for example) and must be accounted for. If $I_i$ is measured for one voxel $i$ of a set of voxels that includes all those occupied by a given marker, an approximate centroid can be calculated by a weighted sum,

$$\mathbf{x} = \sum_i^n (I_i - I_0)\mathbf{x}_i / \sum_i^n (I_i - I_0) \,, \tag{8.19}$$

where $\mathbf{x}_i$ is the centroid of voxel $i$. (Note that the value of $a$ is not needed for this calculation.)

Equation (8.19) provides a good approximation, but it is flawed by the implicit assumption that the centroid of the marker-filled portion of each voxel is coincident with the centroid $\mathbf{x}_i$ of the voxel itself. The error can be appreciated in Fig. 8.4 by comparing the marker-filled portions of voxels $a$ and $b$, at the right and left, respectively. Voxel $a$ is almost completely filled, while only the right third of voxel $b$ is filled. The centroid of the marker-filled portion of $a$ lies approximately at the voxel centroid, $\mathbf{x}_a$, while the centroid of the marker-filled portion of $b$ lies somewhere between the white dotted border of the marker and the right edge of the voxel, which is considerably to the right of centroid $\mathbf{x}_b$. Thus, in Eq. (8.19), the contribution of voxel $b$ contributes an erroneous left shift to the marker centroid.

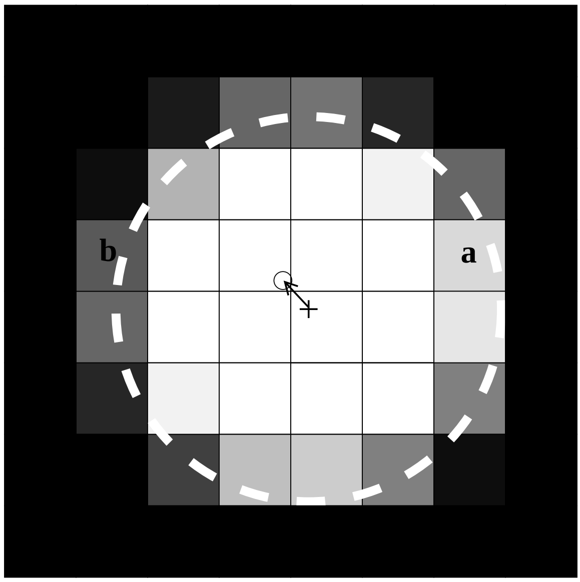



**Figure 8.4:** An illustration in two dimensions of intensity-weighted localization applied to a circular fiducial marker. The white, gray, and black squares represent voxels. The shadings of the voxels indicate their intensities, lighter shading indicating higher intensity. The marker is isolated from the anatomy such that none of the voxels contains both marker and anatomical structure. The black voxels around the periphery are empty and thus have the background intensity $I_0$. The dotted white outline is the edge of the marker. The small "plus" sign is the physical centroid of the marker. The small circle is the estimate of the centroid chosen by intensity-weighted localization. The small arrow is the fiducial localization error **FLE**. Standard intensity-weighted localization implicitly assumes that the centroid of the marker-filled portion of each voxel is coincident with the centroid of the voxel itself. This assumption is almost correct for voxel **a** but not for **b**.

This error can be appreciable and can be corrected for one-dimensional images, but only heuristic algorithms are available for two and three dimensions [36].

As indicated in Fig. 8.4, the localization error can be expected to be considerably smaller than the dimension of a voxel if the visible portion of the marker occupies many voxels. Large marker size (relative to voxel size) provides three benefits. First, the fraction of the voxels within the marker that are only partially filled tends to be smaller for larger markers, so their errors make a relatively smaller contribution to Eq. (8.19). Second, the number of partially filled voxels (as opposed to the fraction) tends to increase with marker size, resulting in a greater likelihood of cancellation among the erroneous shifts. Third, a larger marker results in an averaging of image noise over a larger number of voxels, both partially and completely filled. If the noise is additive and uncorrelated among the voxels, its effect on Eq. (8.19) can be expected to be smaller. Thus, larger markers tend to exhibit smaller FLEs. Brighter markers also have smaller FLEs because of the smaller contribution of image noise relative to marker intensity in Eq. (8.18).

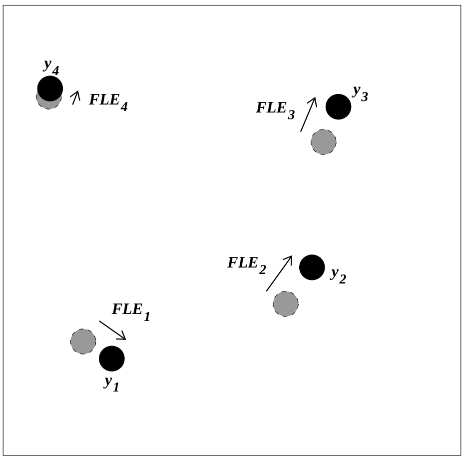
As mentioned at the beginning of Section 8.2, any nonzero displacement $\mathcal{T}(\mathbf{x}) - \mathbf{y}$ between a transformed point $\mathcal{T}(\mathbf{x})$ and its corresponding point $\mathbf{y}$ is a registration error. To the extent that FLE is small and that the form of the transformation correctly describes the motion of the object, the alignment of the fiducial points in the two views will lead to small registration errors for all points. If the transformation is selected from some constrained set (as, for example, the rigid transformations), then, because of FLE, it will ordinarily not be possible to achieve a perfect alignment of fiducials. The resultant misalignment may, in some cases, be used as feedback to assess whether or not the registration is successful (but see the warnings on page 471). A common measure of overall fiducial misalignment, is the root-mean-square (RMS) error. This error, which we will call the *fiducial registration error*, or FRE, is defined as follows. First, we define an individual fiducial registration error,

$$\mathbf{FRE}_i = \mathcal{T}(\mathbf{x}_i) - \mathbf{y}_i \,, \tag{8.20}$$

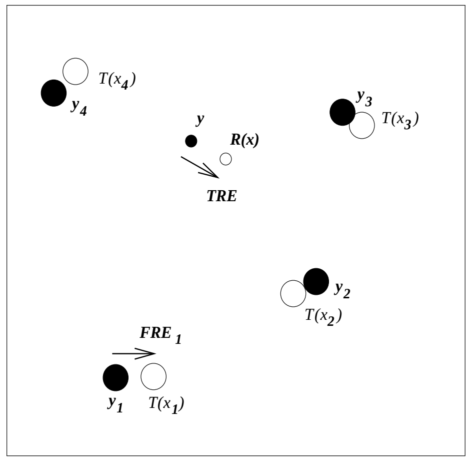
where $\mathbf{x}_i$ and $\mathbf{y}_i$ are the corresponding fiducial points in views $X$ and $Y$, respectively, belonging to feature $i$, as depicted in Fig. 8.5b. We define FRE in terms of the magnitudes of the $\mathbf{FRE}_i$.

$$\mathrm{FRE}^2 = (1/N) \sum_i^N w_i^2 \mathrm{FRE}_i^2 \,, \tag{8.21}$$

where $N$ is the number of fiducial features used in the registration and $w_i^2$ is a non-negative weighting factor, which may be used to decrease the influence of less reliable fiducials. For example, if $\langle \mathrm{FLE}_i^2 \rangle$ is the expected squared fiducial localization error for fiducial $i$, then we may choose to set $w_i^2 = 1/\langle \mathrm{FLE}_i^2 \rangle$, where $\mathrm{FLE}_i$ is the fiducial localization error for fiducial $i$.




(a) Schematic of point-based registration illustrating *fiducial localization error* (FLE). Black circles represent positions $y$ at which points are determined by the localization process in one of two spaces involved in the registration process. The light circles represent the actual positions.




(b) Schematic of point-based registration illustrating two measures of registration error. Black circles represent positions $y$ in one space. The unfilled circles represent positions $x$ in the other space after they have been mapped by the registering transformation $\mathcal{T}$. The larger, numbered circles are the points used to effect the registration. *Fiducial registration error* (FRE) is the alignment error between these. *Target registration error* (TRE) is the registration error at a point (smaller circles) not used in the registration.

**Figure 8.5:** Errors in Point-Based Rigid Registration

Figure 8.5b also depicts *target registration error*, or **TRE**, which is, simply, registration error calculated at some point of interest,

$$\mathbf{TRE}(\mathbf{x}) = \mathcal{T}(\mathbf{x}) - \mathbf{y} \; . \tag{8.22}$$

The term "target" is meant to suggest that the point is the subject of some diagnosis or treatment. (As with FLE we use a bold font for both FRE and TRE to indicate the displacement and a normal font to indicate magnitude.)

### *8.3.1 Points in rigid transformations*

If the transformation to be determined is constrained to be rigid, then Eq. (8.21) can be written as

$$\mathrm{FRE}^2 = (1/N) \sum_i^N w_i^2 |R\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i|^2 \; . \tag{8.23}$$

If the $\mathbf{FLE}_i$ are random errors with zero means and isotropic distributions for all fiducials, then an optimum registration can be achieved by minimizing $\mathrm{FRE}^2$ with $w_i = 1/\langle \mathrm{FLE}_i^2 \rangle$. The minimization of Eq. (8.23) is known as the "Orthogonal Procrustes" problem in the statistics literature.[2] Closed-form solutions for this problem have been available from that discipline since the first one was published by Green in 1952 [39]. The problem is also important in the theory of shape [40–42]. The solution is unique unless the fiducial configuration is linear (in which case $\mathrm{FRE}^2$ is independent of rotations about the fiducial line). Thus, we assume henceforth that the configuration is nonlinear, and, hence, that $N \geq 3$. Algorithm 8.1 provides a simple, reliable method of solution.

**Algorithm 8.1: Point-based, rigid registration**

Find $R$ and $\mathbf{t}$ to minimize $\sum_i^N w_i^2 |R\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i|^2$ :

1. Compute the weighted centroid of the fiducial configuration in each space:

$$\begin{aligned} \bar{\mathbf{x}} &= \sum_i^N w_i^2 \mathbf{x}_i / \sum_i^N w_i^2 \\ \bar{\mathbf{y}} &= \sum_i^N w_i^2 \mathbf{y}_i / \sum_i^N w_i^2 \; . \end{aligned}$$

[2]The term "Procrustes" was originally pejorative. It was first used by Hurley and Cattell [38] in 1962 to express disapproval of a perceived tendency of some to distort one set of observations to support the claim that they fit another set. Hurley and Cattell were drawing an analogy to the habits of the character of the same name from Greek mythology, who stretched, squeezed, or otherwise altered visitors to fit his guest bed. The term is now used in the statistical theory of shape with no negative connotation attached. The "orthogonal" version involves only rigid motion.

2. Compute the displacement from the centroid to each fiducial point in each space:

$$\begin{aligned}\tilde{\mathbf{x}}_i &= \mathbf{x}_i - \bar{\mathbf{x}} \\ \tilde{\mathbf{y}}_i &= \mathbf{y}_i - \bar{\mathbf{y}} \,.\end{aligned}$$

3. Compute the weighted fiducial covariance matrix:

$$H = \sum_i^N w_i^2 \tilde{\mathbf{x}}_i \tilde{\mathbf{y}}_i^t \,,$$

4. Perform singular value decomposition of $H$:

$$H = U \Lambda V^t \,,$$

where $U^t U = V^t V = I$, $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \lambda_3)$, and $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$.

5. $R = V \mathrm{diag}(1, 1, \det(VU)) U^t$.

6. $\mathbf{t} = \bar{\mathbf{y}} - R\bar{\mathbf{x}}$.

---

The diagonal matrix interposed between $V$ and $U^t$ in step 5 is there to insure that $R$ is a proper rotation. It is necessary only when the fiducial points are approximately planar or when the expected size of FLE is on the order of the separation of the fiducials.

(Historical Note: The use of singular value decomposition for this problem was introduced in 1966 by Schönemann along with the first proof of solution for general fiducial configurations [43]. That method was later rediscovered independently by others [44–46]. These methods all permitted improper rotations. The use of $\det(VU)$ to restrict the solution to proper rotations was presented without proof later in 1966 by Farrell [44]. It was rediscovered and proved in 1991 by Umeyama [47]. The original method by Green, which is based on the matrix square root, was later independently derived as well [48, 49]. Green's method was first proved to be a general solution in 1978 by Sibson [50]. A third solution method based on the unit quaternion was discovered in 1986 simultaneously by Faugeras [51] and Froimowitz [52] and then independently by Horn [8]. A recent comparison of these methods plus a method using dual quaternions, whose elements are dual numbers [53], shows no substantial differences among them [54].)

The application of Algorithm 8.1 minimizes FRE$^2$, but, as indicated in Fig. 8.5, finite fiducial localization error FLE can be expected to make both FRE and TRE nonzero. The relationships among the expected values of FLE$^2$, FRE$^2$, and TRE$^2$ are known to an excellent approximation for the most common case, in which the

$\mathbf{FLE}_i$ are random, their probability distributions are isotropic, independent, and identical for all $N$ fiducials, and a uniform weighting is used in the registration ($w_i^2 = 1$). The simplest relationship is that between the expected values of $\mathrm{FLE}^2$ and $\mathrm{FRE}^2$,

$$\langle \mathrm{FRE}^2 \rangle \approx (1 - 2/N)\langle \mathrm{FLE}^2 \rangle \;, \tag{8.24}$$

where the approximation ignores terms involving $\mathrm{FLE}^4$ and higher [55]. This equation and Eqs. (8.25) through (8.29), which follow below, apply both to the situation in which $\langle \mathrm{FLE}^2 \rangle$ arises from the distribution of fiducial localization error in one only space, with the fiducial localization error in the other space being negligible, and to the situation in which there is error in both spaces, in which case $\langle \mathrm{FLE}^2 \rangle$ equals $\langle \mathrm{FLE}_a^2 \rangle + \langle \mathrm{FLE}_b^2 \rangle$, where the subscripts $a$ and $b$ refer to the two spaces [56].

It should be noted that the relationship given by Eq. (8.24) is *completely independent of the fiducial configuration*. Furthermore, the same holds for the probability distribution of $\mathrm{FRE}^2$ [55]. Thus, for a given number of fiducials and given distribution for **FLE**, $\mathrm{FRE}^2$ will be distributed in the same way regardless of whether the fiducials are clustered within a few millimeters of each other or are miles apart, nor do, planar, regular polyhedral, or approximately linear arrangements exhibit any differences with regard to FRE. An advantage of this independence of configuration is that an estimate of $\langle \mathrm{FLE}^2 \rangle$ can be obtained for a given fiducial design and image acquisition protocol by performing a set of registrations involving pairs of images of possibly differing configurations and numbers of fiducials. An estimate for the expected value of $\mathrm{FLE}^2$ can be obtained from the weighted average,

$$\langle \mathrm{FLE}^2 \rangle \approx (1/M) \sum_{m}^{M} N_m/(N_m - 2) \times \mathrm{FRE}^2(m) \;, \tag{8.25}$$

where $M$ is the number of registrations performed and $N_m$ is the number of fiducials involved in registration $m$, and $\mathrm{FRE}^2(m)$ is the FRE for registration $m$.

Unlike $\langle \mathrm{FRE}^2 \rangle$, the value of $\langle \mathrm{TRE}^2 \rangle$ does depend on the fiducial configuration, and it depends on the target position as well. It is most easily stated in terms of quantities measured relative to the principal axes of the fiducial configuration:

$$\langle \mathrm{TRE}^2 \rangle \approx \frac{1}{N}\left(1 + \frac{1}{3}\sum_{k=1}^{3}\frac{d_k^2}{f_k^2}\right)\langle \mathrm{FLE}^2 \rangle \;, \tag{8.26}$$

where $d_k$ is the distance of the target from principal axis $k$, and $f_k$ is the RMS distance of the fiducials from the same axis [57]. This approximation, like that of Eq. (8.24), ignores terms involving $\mathrm{FLE}^4$ and higher. An equivalent approximation was first suggested by Hawkes [58, 59].

FRE is sometimes used as an indication of accuracy in a point-based registration system. Unfortunately, it may in fact be a very poor indicator of accuracy. First of all, as we have seen from Eqs. (8.26) and (8.24), $\langle \mathrm{TRE}^2 \rangle$ is a function of the fiducial configuration, while $\langle \mathrm{FRE}^2 \rangle$ is not. Thus, for example, an approximately linear fiducial configuration will result in a very small value for one of the $f_k$ in the denominator of Eq. (8.26). If the target marker is at an appreciable distance from the line of fiducials, then the corresponding $d_k$ in the numerator will not be small. The result will be a large $\langle \mathrm{TRE}^2 \rangle$, even if $\langle \mathrm{FLE}^2 \rangle$, is small. For this same configuration, $\langle \mathrm{FRE}^2 \rangle$, on the other hand, will be small. By contrast, if the target is located near the centroid of a non-linear fiducial configuration, then all the $d_k$ will be small, none of the $f_k$ will be small, and $\langle \mathrm{TRE}^2 \rangle$ can be smaller than $\langle \mathrm{FRE}^2 \rangle$. Second, if the fiducial configuration and target position are kept constant from registration to registration, variation in FRE is no indicator of variation in TRE for a properly working system. No correlation between these two quantities for a given system has been observed. Thus, for a given registration with a properly working point-based registration system, a value of FRE that is lower or higher than expected for that system does not indicate that the value of TRE for that registration is lower or higher than expected. Third, the registration error $\mathrm{FRE}_i$ for a specific marker $i$ is a perverse indicator of TRE in the vicinity of that marker. The problem can be seen from the following relationship, which, like Eqs. (8.24) and (8.26), is accurate to third order in FLE [57]:

$$\langle \mathrm{TRE}^2(\mathbf{x}_i) \rangle \approx \langle \mathrm{FLE}^2 \rangle - \langle \mathrm{FRE}_i^2 \rangle \, . \tag{8.27}$$

A consequence of this equation is that, for a given fiducial configuration, TRE tends to be worse in the vicinity of those fiducials whose alignments tend to be better! (See [57] for an explanation of this anti-intuitive behavior.)

If interpreted correctly, FRE can, however, provide useful feedback. It can be seen from Eqs. (8.24) and (8.26) that systems with a smaller $\langle \mathrm{FRE}^2 \rangle$ will have a smaller $\langle \mathrm{TRE}^2 \rangle$. These expected values are in fact proportional and do show that a system that tends to exhibit a lower FRE will also tend to produce a lower TRE. A very large FRE on a given registration with a point-based system that usually exhibits a small FRE may indicate that at least one part of the registration system, which includes image acquisition, fiducial localization, and fiducial registration, may not be working properly. A formal statistical decision about the likelihood that a given system with a known $\langle \mathrm{FRE}^2 \rangle$ is performing correctly can be based on the probability that the observed value of FRE will occur at random. For that calculation, it is necessary to use the probability distribution of FRE, which is chi-square with 3N-6 degrees of freedom [55, 57].

### *8.3.2 Points in scaling transformations*

If the transformation to be determined is a similarity transformation, i.e., is of the form of Eq. (8.7), then Eq. (8.21) becomes

$$\mathrm{FRE}^2 = (1/N) \sum_i^N w_i^2 |sR\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i|^2 \,. \tag{8.28}$$

A simple extension of Algorithm 8.1 determines the scaling factor $s$, rotation $R$, and translation $\mathbf{t}$ that minimize FRE$^2$. The extension is given by Algorithm 8.2 [60].

**Algorithm 8.2: Point-based registration: Similarity Transformations**

Find $R$, $\mathbf{t}$, and $s$ to minimize $\sum_i^N w_i^2 |sR\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i|^2$ :

1. Set $s = 1$ and determine $R$ using steps 1 through 5 of Algorithm 8.1.
2. Calculate $s$: $\quad s = \frac{\sum_i^N w_i^2 R\tilde{\mathbf{x}}_i \cdot \tilde{\mathbf{y}}_i}{\sum_i^N w_i^2 \tilde{\mathbf{x}}_i \cdot \tilde{\mathbf{x}}_i}$ .
3. $\mathbf{t} = \bar{\mathbf{y}} - sR\bar{\mathbf{x}}$.

---

A weakness of this solution is that the scaling factor $s$ does not have a reciprocal relationship with the factor in the reverse problem. Thus, if the point sets, $\{\mathbf{x}_i\}$ and $\{\mathbf{y}_i\}$ are exchanged, the new scale factor will not necessarily equal $1/s$. The analog to Eq. (8.24) for similarity transformations is

$$\langle \mathrm{FRE}^2 \rangle \approx (1 - 7/(3N)) \langle \mathrm{FLE}^2 \rangle \,, \tag{8.29}$$

and the probability distribution of FRE is chi-square with $3N - 7$ degrees of freedom [56].

There are no known closed-form solutions for the anisotropic scaling problems, in which the scaling matrix $S$, the rotation matrix $R$, and the translation vector $\mathbf{t}$ are to be determined for transformations of the form of Eqs. (8.5) or (8.6). Instead, an iterative search is required. For scaling of the form of Eq. (8.5) the problem is especially simple. This problem, which is encountered when the $x$, $y$, and $z$ scales of an imaging device are incorrect to differing degrees, as, for example, with inaccurate MR gradients (see page 462 in Section 8.2.3), reduces to a search of the three-dimensional space of diagonal elements of $S$, as described by Algorithm 8.3.

**Algorithm 8.3: Point-based registration: Nonisotropic scaling**

Find $R$ , $\mathbf{t}$, and $S$ to minimize $\sum_i^N w_i^2 |RS\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i|^2$ :

1. Perform steps 1 and 2 of Algorithm 8.1 to determine the centroids, $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$, in each space and the displacement of each fiducial point $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{y}}_i$ from the centroid in its respective space.
2. Set iterative count $n = 0$.
3. Choose an initial scaling matrix $S^{(0)}$.
4. Repeat the following steps:
   (a) Set $\tilde{\mathbf{x}}_i^{(n)} = S^{(n)}\tilde{\mathbf{x}}_i$.
   (b) Perform steps 3 through 5 of Algorithm 8.1 to find $R$.
   (c) Add one to $n$.
   (d) Determine a new value of $S^{(n)}$.
5. Stop when FRE $<$ threshold or $n >$ maximum iteration count.
6. $\mathbf{t} = \bar{\mathbf{y}} - RS\bar{\mathbf{x}}$.

---

While the problem is reduced to a three-dimensional search for the optimal $S$, no method is specified for conducting that search. In particular, no methods are given for steps 3 or 4d of Algorithm 8.3. A review of methods for multidimensional optimization can be found in [61].

For scaling of the form of Eq. (8.6), the problem is complicated by the fact that $S$ is applied to the rotated $\tilde{\mathbf{x}}_i$. As a result, the substitution in $\tilde{\mathbf{x}}_i^{(n)} = S^{(n)}\tilde{\mathbf{x}}_i$ in step 4a is no longer correct. Instead, a six-dimensional parameter space must be searched. The space consists of the diagonal elements of $S$ and three independent parameters to specify $R$, e.g., three Euler angles (see Section 8.2.1). Once these parameters are determined, the translation is $\mathbf{t} = \bar{\mathbf{y}} - SR\bar{\mathbf{x}}$.

### *8.3.3 Points in perspective projections*

When tomographic images are used in conjunction with images acquired by means of x-ray projection, endoscopy, laparoscopy, microscopy, direct video, or photography, or when any of these latter modalities are used to guide surgery or other therapy to a target location, a 2-D-to-3-D registration problem must be solved. When several corresponding point pairs can be identified in two-dimensional and three-dimensional views of a single rigid object, the problem can be solved by means of point-based, rigid registration. This problem is common in robot and computer vision applications and has been thoroughly studied in that domain, where it is known as the "object-pose estimation problem" [6, 10, 62, 63].

The form of the transformation associated with these projections from three dimensions to two is the perspective transformation. That transformation is described in Section 8.2.2.4 in terms of the pinhole camera and is given by Eq. (8.12) and substitution (8.13). Without loss of generality, the problem can be treated by orienting the coordinate system so that $\hat{\mathbf{p}}$ is aligned with the $z$ axis and by placing the origin at the pinhole. With these choices, we have $\mathbf{x} \cdot \hat{\mathbf{p}} = z$ and $\alpha = 0$. Eq. (8.12) then simplifies to

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \frac{f}{z} \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \tag{8.30}$$

which transforms all points to the $z' = f$ plane. Solving Eq. (8.30) for $x$, $y$, and $z$ leads to $\mathbf{x} = \lambda \mathbf{x}'$, where $\lambda$ is an arbitrary scalar. Thus, a given two-dimensional image point, $(x', y')^t$ lying in the $z' = f$ image plane is projected to a line in three-dimensional space that passes through the three-dimensional origin (the pinhole) and through the three-dimensional image point, $(x', y', f)^t$.

Thus, it is not possible to determine a unique three-dimensional position for a single point from its perspective projection alone. Some additional information is required to determine where on the projected line the point lies. At least one other constraint must be provided to resolve the ambiguity. A common constraint in computer vision that is uncommon in medical applications is $ax+by+cz+d = 0$, which means that the point is known to lie on a given plane (e.g., the floor, a wall, etc.). Another constraint restricts the line to intersect a "silhouette curve," which is the apparent outline of a three-dimensional surface when viewed in perspective from a given point. The use of this latter constraint for registration of tomographic images of the anatomy to radiographic images is described in Chapter 6. In the most common point-based object-pose estimation problem, the projections $(x'_i, y'_i, f)^t$ of $N$ three-dimensional points $\mathbf{x}_i, i = 1, \ldots, N$ are used to determine all $N$ $\mathbf{x}_i$ simultaneously. The additional information needed to solve the problem is provided by (a) stipulating that the $N$ $\mathbf{x}_i$ all lie on the same rigid object and (b) giving their three-dimensional positions for some orientation of that object. Thus, (a) it is known that, for some $R$ and $\mathbf{t}$, $R\mathbf{x}_i + \mathbf{t} = \mathbf{y}_i$, and (b) the points $\mathbf{y}_i, i = 1, \ldots, N$ are given. In medical applications, these points are typically localized from a tomogram of rigid anatomy. The pose estimation is accomplished by searching for the $R$ and $\mathbf{t}$ that solve the $2N$ equations represented by the $x$ and $y$ components of Eq. (8.30) for the $N$ points. The equations are typically solved in the least-square sense, and because of the nonlinear constraints on the form of $R$, they must be solved iteratively. A common approach is carried out in terms of an $R$ that is parameterized by rotation angles, such as the Euler angles (see Section 8.2.1): start with an approximate solution $\mathbf{x}_i, i = 1, \ldots, N$, calculate the perspective projections of the $\mathbf{x}_i$, determine the sum of squared errors between the calculated projections and the actual projections, linearize the resulting expression (assumes small error), solve for the first-order corrections to the angles and $\mathbf{t}$, and iterate [10]. If the camera constant $f$ is unknown, then a similar approach is required with $f$ included as an additional unknown. This latter problem is known as the "interior orientation" problem in photogrammetry [10].

### *8.3.4 Points in curved transformations*

When points are used for registration with curved transformations, it is usually possible to align all fiducial points exactly, but the mapping of the non-fiducial points can vary strongly with the form of transformation used. Knowledge of the genesis of the morphological changes from one view to the other should be used to determine the class of curved transformations to be employed whenever possible, as, for example, with rectification (see Section 8.2.3) or with interpatient registration (see Chapters 3 and 17). Otherwise, $\mathcal{T}$ is typically chosen from some set of continuous, differentiable functions, such as the polynomials, the polynomial splines, or the thin-plate splines. For example, an interpolating polynomial (see Section 8.2.2) of the form of Eq. (8.14) can reduce FRE to zero at $(I+1)(J+1)(K+1)$ points. The thin-plate spline of Eq. (8.17) reduces FRE to zero at $N$ points. The cubic spline of Eq. (8.15), which is typically implemented in terms of B-splines, can reduce FRE to zero at a given set of points, but only for very uniform point distributions or for very closely spaced knots. These restrictions have limited the use of B-splines as interpolants in medical applications of point-based registration primarily to rectification problems based on phantoms with uniformly spaced fiducials (but see Section 8.5.3.5 for an application to intensity-based registration). In these situations, the knot spacing is typically forced to be uniform by the machining of the phantom. For this case, the cubic B-spline expansion can be given in terms of the simple segments of splines given by Eqs. (8.16).

With the polynomial, cubic-spline, and thin-plate spline transformations [and with the affine transformation of Eq. (8.8)], the point-based registration problem reduces to the determination of a set of coefficients. There is in fact a separate problem for each of the three components of $\mathbf{x}'$. For each class of transformation, the problem can be solved individually for each component by finding the solution to a set of simultaneous linear algebraic equations for coefficients that can, in each case, be organized into a one-dimensional vector $\mathbf{a}$. The meaning of the elements of $\mathbf{a}$ varies with each class of transformation. The vector for the $x$ coordinates, for example, is uniquely determined by using the $x, y, z$ coordinates of the $N$ fiducial localizations $\mathbf{x}_i$ on the right side of Eqs. (8.14), (8.15), or (8.17) and substituting the corresponding $x$ coordinates of the $N$ fiducial localizations $\mathbf{x}'_i$ into the left side. The form of the resultant equation is in every case,

$$M\mathbf{a} = \mathbf{b}. \tag{8.31}$$

Thus, the point-based registration problem with any of these transformations is reduced to the solution of three sets of linear algebraic equations—one set each for $x$, $y$, and $z$. While there are more coefficients involved, this situation is simpler than that of the rigid-body transformation problem (see Algorithm 8.1) because it lacks the nonlinear side-conditions that insure that $R$ is orthogonal. For the polynomial transformation of Eq. (8.14), with points $\mathbf{x}_n$, $n = 1, \ldots, N$ where $N$ equals $(I + 1)(J + 1)(K + 1)$, the matrix $M$ is $N \times N$, and its elements are the polynomials $x_n^i y_n^j z_n^k$. $M$ is nonsingular but tends to be ill-conditioned [64]. (Techniques for increasing numerical accuracy with polynomial interpolation can be found in [61].) For the cubic-spline it is necessary to impose some additional conditions along with Eq. (8.15) in order to produce as many equations as unknowns. A common condition is to restrict the second derivatives to be zero at the borders, producing the so-called "natural splines." $M$ is again nonsingular. For the thin-plate splines, both two-dimensional and three dimensional, side conditions are necessary for each component in order to insure that Eq. (8.31) has a solution. With the $x$ component of $\mathbf{x}'$ again serving as the example, the conditions are for the two-dimensional case [17]

$$\sum_i^N c_i = \sum_i^N x_i c_i = \sum_i^N y_i c_i = 0 \tag{8.32}$$

with a similar condition for $z$ in the three-dimensional case. When these conditions are incorporated into Eq. (8.31) along with Eq. (8.17), $M$ is $(N + 3) \times (N + 3)$ for two dimensions, or $(N + 4) \times (N + 4)$ for three dimensions, and nonsingular.

Transformations derived from the equations of continuum mechanics can be used with point-based registration as well. In this case the disparity between corresponding points may be used to determine the forces that drive the displacements. The correct form of the force as a function of these disparities is not clear, nor is the means to interpolate that force to other points. Recently the three-dimensional thin-plate spline has been employed for this purpose for the registration of breast images [30].

It should be pointed out that the use of curved transformations to align all fiducial points exactly, i.e., to interpolate them, is sometimes not a sensible approach. The problem is that the localization of the points will inevitably include some localization error. Aligning these erroneous points exactly will result in the propagation of the error throughout the space. To accommodate localization error, the polynomial and cubic-spline interpolants should be employed with more fiducial points than coefficients to allow for a least-squares fit in analogy with the rigid-body case. In that case, Eq. (8.31) has more unknowns than equations. The least-squares approximation may then be found with standard techniques [61, 64].

## 8.4 Surface-based methods

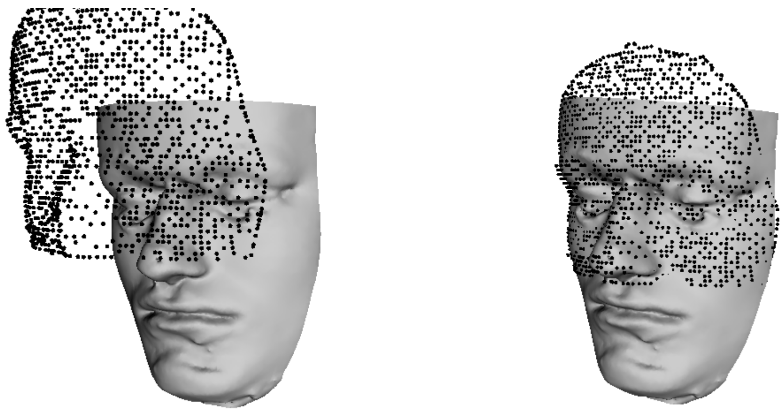
The 3-D boundary surface of an anatomic object or structure is an intuitive and easily characterized geometrical feature that can be used for medical image registration. Surface-based image registration methods involve determining corresponding surfaces in different images (and/or physical space) and computing the transformation that best aligns these surfaces.

The skin boundary surface (air-skin interface) and the outer cranial surface are obvious choices that have frequently been used for both image-to-image (e.g., CT-MR, serial MR) and image-to-physical registration of head images (see Fig. 8.6). The surface representation can be simply a point set (i.e., a collection of points on the surface), a faceted surface (e.g., triangle set), an implicit surface, or a parametric surface (e.g., B-spline surface). Extraction of a surface such as the skin or bone is relatively easy and fairly automatic for head CT and MR images. Extraction of many soft tissue boundary surfaces is generally more difficult and less automatic. Image segmentation algorithms can generate 2-D contours in contiguous image slices that are linked together to form a 3-D surface, or they can generate 3-D surfaces directly from the image volume. Surface extraction from images is covered in detail in Chapter 6. The related problem of two-dimensional contour extraction is treated in Chapter 3. In physical space, skin surface points can be easily determined using laser range finders; stereo video systems; and articulated mechanical, magnetic, active and passive optical, and ultrasonic 3-D localizers. Bone surface points can be found using tracked A-mode [65] and B-mode [66] ultrasound probes. The computer vision sensors, 3-D localizers, and tracked A-mode ultrasound probes produce surface point sets. Tracked B-mode probes produce a set of 2-D images (or a single compounded 3-D image) from which bone surface points need to be segmented.

Surfaces can provide basic features for both rigid-body and nonrigid registration. A central and difficult question that must be addressed by any nonrigid surface-based registration algorithm is how deformation of the contents of an object is related to deformation of the surface of the object. Most of the surface-based registration algorithms that have been reported are concerned with rigid-body transformation, occasionally with isotropic or nonisotropic scaling. Thus, in this section, we are concerned primarily with rigid-body registration. The registration of 2-D projection images and 3-D tomographic images is considered in Chapter 6.

### *8.4.1 Disparity functions*

There is a large body of literature in computer vision concerned with the surface-based registration problem. Some of this work has addressed limited classes of shapes, e.g., polyhedral models [69] and piecewise-superquadric models [70, 71]. Other authors rely on the existence of specific simple features or make assumptions about the global shape (see [72] for an extensive review). For example, the work in [51] assumes the existence of reasonably large planar regions within a free-form



(a) (b)

**Figure 8.6**: Illustration of surface-based registration of the head using the facial skin surface. The dots represent skin surface points acquired with a four-camera photogrammetry system. The surface rendering represents a triangle set model of the skin surface extracted from an MR image volume. The left panel (a) shows the initial position of the data sets. The right panel (b) shows the data sets after registration. The registration was performed using an independent implementation [67] of the iterative closest point algorithm [68]. The surfaces overlap only partially, which is a common situation in surface-based registration. The problem of partial overlap was dealt with by setting the weights of outliers to zero after the first search converged before running a second search.

shape. Some approaches represent the surface with generalized cylinders [73] or extended Gaussian images [74]. The latter method allows matching of convex and restricted sets of nonconvex shapes based on surface normal histograms.

The approach for solving the surface-based registration problem that is frequently used in more recent computer vision literature (where it is often called the "free-form" surface matching problem), and that is normally used in the medical image processing community, is to search for the transformation that minimizes some disparity function or metric between the two surfaces $X$ and $Y$. The disparity function is generally a distance. In mathematics, the distance between two feature sets $A$ and $B$ is normally defined as the minimum distance between a point in $A$ and a point in $B$, namely, $d(A,B) = \min_a \min_b d(\mathbf{a}, \mathbf{b})$, where $\mathbf{a} \in A$, $\mathbf{b} \in B$, and $d(\mathbf{a}, \mathbf{b})$ is the Euclidean distance between $\mathbf{a}$ and $\mathbf{b}$. Thus $d(A,B)$ is small if one pair of points in these two sets are close. Another common measure is the Hausdorff distance. The Hausdorff distance from $A$ to $B$ is $d(A,B) = \max_a \min_b d(\mathbf{a}, \mathbf{b})$, and the Hausdorff distance between two sets $A$ and $B$ is $d_H(A,B) = \max(d(A,B), d(B,A))$. Thus $d_H(A,B)$ is small if and only if every point of $A$ is close to a point in $B$ and vice versa. Neither of these distance measures is very useful as a disparity function. The Hausdorff distance is not meaningful in practice because the corresponding surfaces generally overlap only partially, and because it is sensitive to statistical outliers. The disparity function normally used for surface-based image registration is an average, and optionally weighted, distance between points on one surface and corresponding points on the other surface. Let $\{\mathbf{x}_j\}$ for $j = 1, \ldots, N_x$ be a set of $N_x$ points on the surface $X$. The general approach is to search for the transformation that minimizes the disparity function

$$d(\mathcal{T}(X), Y) = \sqrt{\sum_{j=1}^{N_x} w_j^2 d^2(\mathcal{T}(\mathbf{x}_j), Y)} = \sqrt{\sum_{j=1}^{N_x} w_j^2 ||\mathcal{T}(\mathbf{x}_j) - \mathbf{y}_j||^2}\,, \tag{8.33}$$

where

$$\mathbf{y}_j = \mathcal{C}(\mathcal{T}(\mathbf{x}_j), Y) \tag{8.34}$$

is a point on the surface $Y$ "corresponding" to the point $\mathbf{x}_j$, $\mathcal{C}$ is a "correspondence" function (e.g., closest point operator), and $\{w_j\}$ is a set of weights associated with $\{\mathbf{x}_j\}$. We note that Eq. 8.33 is similar to Eq. 8.21 of point-based registration. The principal difference between point-based registration and surface-based registration is in the availability of point correspondence information. It is the lack of exact point correspondence information that causes surface-based registration algorithms to be based on iterative search. Eq. 8.34 merely provides approximate point correspondence information for a particular $\mathcal{T}$ during an iterative search.

The point set $\{\mathbf{x}_j\}$ and the surface $Y$ have been called, respectively, the "hat" and "head" [75], the "dynamic" and "static" feature sets [76], and the "data" point set and “model” surface shape [68]. Typically one surface contains more information than the other. The surface from the image that covers the larger volume of the patient and/or has the highest resolution is generally picked as the model shape.[3] For example, when using the skin surface to perform CT-to-physical space registration, the triangle set representation of the CT skin surface typically contains $\sim 10^4$–$10^5$ vertices, whereas the number of physical space skin surface points typically is $\sim 10^2$–$10^3$. In this case, the CT triangle set is chosen as the model surface shape and the physical space point set is chosen as the data point set. We note that Eq. 8.33 is generally not symmetric in $X$ and $Y$.

In point-based registration (Eq. 8.21), if the fiducial localization error $\mathbf{FLE}_i$ for each point is independently random and normally distributed around the true position, then the maximum likelihood estimate of the transformation parameters is obtained by weighting the $i$th point by $1/\langle \mathrm{FLE}_i^2 \rangle$ in Eq. 8.21. In surface-based registration (Eq. 8.33), statistical independence of errors is unlikely. For example, if an isointensity surface is extracted from an image, and the isointensity value used is too high or too low, the extracted surface relative to the actual surface will be either slightly dilated or slightly eroded. The skin is a movable and deformable structure, and local deformations tend to be highly correlated. Physical space surface points acquired with a sensor can have biased error due to miscalibration. Nonetheless, weights can be useful to reduce the influence of less reliable surface points. For example, the gradient magnitude of an edge in an image can provide information about surface segmentation error at that point. Often a triangle set representation of a surface is extracted and simplified to reduce storage requirements and execution time. Some surface simplification algorithms provide an estimate of error arising from the simplification process. Many sensors and tracking devices have less accuracy at the edges of the working volume. Weights could potentially be used to account for the sensitivity of the registration to the perturbation of individual surface points (e.g., see the geometrical constraint analysis in [77]). For example, a few points in areas of high curvature might be more important to the registration than many points in relatively planar regions. Weights can be used to account for nonuniform sampling density [67]. Finally, weights can also be used to deal with outliers that can arise from nonoverlapping sections of surfaces, as seen at the top of Fig. 8.6(b), poor segmentation, and erroneous sensor data [67, 78]. For example, outliers can be defined as points whose distance from the closest corresponding surface point is more than two or three standard deviations above the mean distance. The weights of outliers can be reduced or set to zero, either after each iteration, or after one search converges before running another search.

Many variations of Eq. 8.33 are possible. For example, mean distance can be

[3] The terms “data” and “model” arise from an industrial application: registration of digitized data from unfixtured rigid objects obtained using high-accuracy noncontact devices with an idealized geometrical (e.g., computer-aided design) model prior to shape inspection (Besl & McKay wrote [68] while working at General Motors Research Laboratories).

used rather than root-mean-square distance (i.e., L1 norm versus L2 norm) [79]. Outliers can be handled using a thresholded distance, or by using a sigmoidal distance function (which is essentially a gradually tapered thresholded distance). If surface points are collected in physical space using a probe with a spherical ball at the tip, the recorded surface points $\mathbf{x}_j$ are displaced from the actual surface $Y$ by the radius $r$ of the ball-point tip. This is true for any orientation of the localization probe relative to the surface normal at the point of contact.[4] In this case, a more appropriate disparity function is [78]

$$d(\mathcal{T}(X), Y) = \sqrt{\sum_{j=1}^{N_x} w_j^2 (||\mathcal{T}(\mathbf{x}_j) - \mathbf{y}_j|| - r)^2} \,. \tag{8.35}$$

### 8.4.2 Head and hat algorithm

The first investigators to apply surface-based registration to a medical problem were Pelizzari, Chen, and colleagues [75]. They used their "head and hat" algorithm to register CT, MR, and PET images of the head. The "hat" is a skin surface point set $\{\mathbf{x}_j\}$. The "head" is a polygon set model of the skin surface $Y$ created by segmenting contours in contiguous transverse image slices. They define $\mathbf{y}_j$ as the intersection with the "head" $Y$ of a line joining the transformed "hat" point $\mathcal{T}(\mathbf{x}_j)$ and the centroid of the "head" $Y$. The intersection is efficiently calculated by reducing the 3-D line-polyhedron intersection problem to a 2-D line-polygon intersection problem. The transformation $\mathcal{T}$ that minimizes Eq. 8.33 is found using a standard gradient descent technique. The major limitations of this technique are due to the particular distance used, i.e., the distance from the surface point to the surface intersection along a line passing through the surface centroid. This definition of distance requires that the surface be approximately spherical. It also requires that a good initial transformation be supplied as input to the transformation parameter search. Finally, it is probably related to the observation by the authors and others that the search frequently terminates in local minima and thus requires substantial user interaction.

### 8.4.3 Distance definitions

A more general definition of distance between a point and a surface is the distance between the point and the closest point on the surface. That is, the correspondence function $\mathcal{C}$ in Eq. 8.34 is the closest point operator, and $\mathbf{y}_j$ is the point on the surface $Y$ closest to the transformed point $\mathcal{T}(\mathbf{x}_j)$. The closest point and distance

[4] A localization probe is often calibrated by placing the ball-point tip in a hemispherical divot and pivoting the probe about the center of the divot. The position of the probe tip relative to the coordinate system of the probe is determined by finding the most invariant point (in a least squares sense) in these pivot motions. This invariant point is the center of the ball-point tip. That is, the ball-point tip pivots about the center of the ball rather than a point on the surface of the ball.

calculation depends on the surface representation. For example, a common representation is a triangle set. Let $t$ be the triangle defined by the three vertices $\mathbf{r}_1$, $\mathbf{r}_2$, and $\mathbf{r}_3$. The distance between the point $\mathbf{x}_j$ and the triangle $t$ is

$$d(\mathbf{x}_j, t) = \min_{u+v+w=1} ||u\mathbf{r}_1 + v\mathbf{r}_2 + w\mathbf{r}_3 - \mathbf{x}_j|| \,, \tag{8.36}$$

where $u \in [0,1]$, $v \in [0,1]$, and $w \in [0,1]$. The required closed-form computations are straightforward. Let $T = \{t_i\}$ for $i = 1, \ldots, N_t$ be a set of $N_t$ triangles. The distance between the point $\mathbf{x}_j$ and the triangle set $T$ is given by

$$d(\mathbf{x}_j, T) = \min_{i \in \{1, \ldots, N_t\}} d(\mathbf{x}_j, t_i) \,. \tag{8.37}$$

The closest point $\mathbf{y}_j$ on the the triangle set $T$ satisfies the equality $d(\mathbf{x}_j, \mathbf{y}_j) = d(\mathbf{x}_j, T)$. Another relatively common representation is a parametric surface $S = \{\mathbf{r}(u, v)\}$, where, for example, the components of $\mathbf{r}$ are spline functions. The point-to-parametric-entity distance $d(\mathbf{x}_j, S)$ is easily computed with a 2-D minimization in $u$ and $v$ using derivatives of $\mathbf{r}$ (e.g., Newton's method). A good initial value $(u_0, v_0)$ is necessary. Initial values can be obtained by approximating the parametric surface with a triangle set, tagging each triangle vertex with the corresponding $(u, v)$ value of the parametric surface, and then, for each $\mathbf{x}_j$, finding the closest triangle vertex and using as the initial value the value stored with that vertex.

The most computationally expensive step in the registration process is finding the closest points. Given a data shape point set with $N_x$ points and a model shape with $N_y$ geometrical primitives (point, line segments, and triangles), the computational complexity of finding the closest points using an exhaustive search is $O(N_x N_y)$. One way to speed up the search process is to use a special data structure. One such data structure is a multidimensional binary search tree ($k$-D tree, where $k$ is the dimensionality of the search space) [80, 81]. For 3-D image registration, $k = 3$. The $k$-D tree is a binary tree in which each node represents a subset of data records (encoding in our case, geometrical primitives) and a partitioning of that subset. Each nonterminal node has two children that represent the two subsets defined by the partitioning. The terminal nodes represent mutually exclusive small subsets of the records called buckets. A 3-D tree divides space into a collection of rectangular parallelepipeds that correspond to the terminal nodes. This data structure provides an efficient method for examining only those points closest to a given point. A $k$-D tree can be constructed in $O(N_y \log N_y)$ time. Each closest point search can be performed in $O(\log N_y)$ time. Thus the computational complexity of finding $N_x$ closest points at each step of an iterative transformation parameter search using a $k$-D tree is $O(N_x \log N_y)$.

### *8.4.4 Distance transform approach*

The calculation of point-to-surface distance is computationally intensive, even when using special data structures and other optimizations. A computationally efficient alternative is to use a distance transform (DT). A DT of a binary image $I$

is an assignment to each voxel $\mathbf{v}$ of the distance between $\mathbf{v}$ and the closest feature voxel in $I$. A DT of a binary image where the feature voxels are surface voxels is a gray-level image in which each voxel $\mathbf{v}$ has a value that is the distance from the center of $\mathbf{v}$ to the center of the nearest surface voxel. Thus a DT provides a method for precomputing and storing point-to-surface distance. Normally squared distance is stored. Then, at each step of an iterative transformation parameter search, the value of the disparity function in Eq. 8.33 is computed simply by summing the values of the voxels in the squared distance image that contain the transformed points $\{\mathcal{T}(\mathbf{x}_j)\}$. One limitation of this approach is that a DT is spatially quantized, i.e., a DT image contains exact point-to-surface distance only at regularly spaced lattice points (centers of voxels). A slight improvement over using the distance at the nearest lattice point can be achieved by using a trilinear interpolation of the distances at the nearest eight lattice points. Nonetheless, the surface is fundamentally represented by the point set consisting of the centers of all feature (surface) voxels, and thus subvoxel surface position information is lost. Spatial quantization might be the reason that registrations produced by surface-based methods using DTs have been reported to be considerably less accurate than registrations produced by surface-based methods not using DTs (e.g., see [82]). Integer approximations of Euclidean distance (e.g., the chamfer 3-4-5 algorithm [83]) are frequently used to reduce the DT computation time, but several reasonably efficient algorithms for computing an exact Euclidean DT now exist (e.g., [84]). Surface-based registration using a DT was first applied to medical images by [85].

### *8.4.5 Iterative closest point algorithm*

All surface-based registration algorithms must search for the transformation $\mathcal{T}$ that minimizes the disparity function in Eq. 8.33 or a variation thereof. This is a general nonlinear minimization problem that is typically solved using one of the common gradient descent techniques (e.g., see [61]). The search will typically converge to, or very close to, the correct minimum of the disparity function minimum if the initial transformation is within about 20–30 degrees and 20–30 mm of the correct solution. To help minimize the possibility of the search getting stuck in a local minimum, many investigators perform the search in a hierarchical coarse-to-fine manner.

Besl & McKay [68] presented an algorithm which reduces the general nonlinear minimization problem to an iterative point-based registration problem. Their iterative closest point (ICP) algorithm is a general-purpose, representation-independent, shape-based registration algorithm that can be used with a variety of geometrical primitives including point sets, line segment sets, triangle sets (faceted surfaces), and implicit and parametric curves and surfaces. One shape is assigned to be the "data" shape and the other shape to be the "model" shape. For surface-based registration, the shapes are surfaces. The data shape is decomposed into a point set (if it is not already in point set form). Then the data shape is registered to the model shape by iteratively finding model points closest to the data points, registering the two point sets (e.g., using Algorithm 8.1), and applying the resulting transformation to the data points. Details of this approach are provided in Algorithm 8.4.[5] This algorithm converges to a local minimum of the disparity function in Eq. 8.33. The proof of convergence depends on the correspondence function $\mathcal{C}$ being the closest point operator.

### *8.4.6 Weighted geometrical feature algorithm*

Most registration methods that align 3-D image volumes by matching geometrical features such as points or surfaces use a single type of feature. It is possible and potentially advantageous to perform registration using multiple geometrical features simultaneously. One approach that was suggested by Collignon et al. [86] is to define a new disparity function that is a weighted sum of the point-based (Eq. 8.21) and surface-based (Eq. 8.33) disparity functions. Let $\{X_i\}$ for $i = 1, \ldots, N_s$ be a set of $N_s$ data shapes to be registered with another set of model shapes $\{Y_i\}$. For example, $i = 1$ might be a point set and $i = 2$ a surface. Then we can define a new disparity function

$$d(\mathcal{T}) = \sum_{i=1}^{N_s} \sqrt{\sum_{j=1}^{N_{X_i}} w_{ij}^2 ||\mathcal{T}(\mathbf{x}_{ij}) - \mathbf{y}_{ij}||^2} \;, \tag{8.38}$$

where

$$\mathbf{y}_{ij} = \mathcal{C}_i(\mathcal{T}(\mathbf{x}_{ij}), Y_i) \;. \tag{8.39}$$

The term under the square root symbol is analogous to Eqs. 8.21 and 8.33. The weights $\{w_{ij}\}$ can incorporate both *intra*-shape weighting and *inter*-shape weighting. A similar approach was suggested by Maurer *et al.* [67], but with the outer summation inside the square root symbol, that is,

$$d(\mathcal{T}) = \sqrt{\sum_{i=1}^{N_s} \sum_{j=1}^{N_{X_i}} w_{ij}^2 ||\mathcal{T}(\mathbf{x}_{ij}) - \mathbf{y}_{ij}||^2} \;. \tag{8.40}$$

The transformation $\mathcal{T}$ that minimizes either of these hybrid disparity functions can be found using any of the common gradient descent techniques. A consequence of putting the outer summation inside the square root symbol is that Eq. 8.40 can be minimized using an extension of the ICP algorithm called the weighted geometrical feature (WGF) algorithm [67]. Details of this approach are provided in Algorithm 8.4.

[5] Algorithm 8.4 details the more general weighted geometrical feature (WGF) algorithm. The ICP algorithm is a special case of the WGF algorithm where the number of shapes $N_s = 1$.

The WGF algorithm is potentially useful for image-to-physical registration. For example, the position of a bone-implanted marker can be determined much more accurately than that of a skin-affixed marker or an anatomic landmark. A disadvantage of using bone-implanted markers is that an invasive pre-operative procedure is required to implant each marker. By combining surface information, the WGF algorithm allows registration to be performed using only one or two such markers (point-based registration requires a minimum of three noncollinear points). For example, very accurate (TRE $\sim$ 1–2 mm) point-and-surface-based CT-to-physical registration of the head was achieved with one bone-implanted marker and 25 surface points [78].

**Algorithm 8.4: Weighted geometrical feature (WGF) rigid-body registration**

Find the rigid-body transformation $\mathcal{T}$ that minimizes the disparity function in Eq. 8.40:

1. Initialization: $k = 1, \mathbf{x}_{ij}^{(0)} = \mathbf{x}_{ij}, \mathbf{x}_{ij}^{(1)} = \mathcal{T}^{(0)}(\mathbf{x}_{ij}^{(0)})$,

   where $\mathcal{T}^{(0)}$ is some initial transformation. The variable $k$ and the superscript in parentheses are iteration indices. The algorithm can be repeated using multiple initial transformations to solve the local minimum problem.

2. Iteratively apply the following steps, incrementing $k$ after each loop, until convergence within a tolerance $\epsilon$ is achieved:

   (a) For each shape $X_i$, compute the closest points $\mathbf{y}_{ij}^{(k)} = \mathcal{C}_i(\mathbf{x}_{ij}^{(k)}, Y_i)$ for $j = 1, \ldots, N_{X_i}$.

   (b) Compute the transformation $\mathcal{T}^{(k)}$ between the initial point set, $\{\mathbf{x}_{ij}^{(0)}\}$, and the current set, $\{\mathbf{y}_{ij}^{(k)}\}$, using the weights $\{w_{ij}\}$. This step is effected by means of Algorithm 8.1 with the points for all shapes collected in each of the two point sets to produce two corresponding point sets.

   (c) Apply the transformation to produce registered points $\mathbf{x}_{ij}^{(k+1)} = \mathcal{T}^{(k)}(\mathbf{x}_{ij}^{(0)})$.

   (d) Terminate the iterative loop when $d(\mathcal{T}^{(k)}) - d(\mathcal{T}^{(k+1)}) < \epsilon$, where $d(\mathcal{T})$ is given by Eq. 8.40.

## 8.5 Intensity-based methods

According to the classifications introduced earlier in this chapter, image intensity is an alternative “registration basis” to points or surface features. It has recently become the most widely used registration basis for several important applications. In this context, the term intensity is invariably used to refer to the scalar values in image pixels or voxels. The physical meaning of the pixel or voxel value depends on the modalities being registered and is very often not a direct measure of optical power (the strict definition of intensity).

Intensity-based registration involves calculating a transformation between two images using the pixel or voxel values alone. In its purest form, the registration transformation is determined by iteratively optimizing some “similarity measure” calculated from all pixel or voxel values. Because of the predominance of three-dimensional images in medical imaging, we refer to these measures as *voxel similarity* measures. In practice, many intensity-based registration algorithms use only a subset of voxels and require some sort of preprocessing. For example, the algorithm may run faster if only a subset of voxels are used. This subset can be chosen on a regular grid, or be randomly chosen. It is normal in these circumstances to blur the images before sampling to avoid aliasing in the subsampled images, and the amount of blurring used may be application dependent. Alternatively, an algorithm may work reliably only if the similarity measure is calculated from the voxels in a defined region of interest in the image, rather than all voxels. In this case, some sort of pre-segmentation of the images is required, and this is likely to depend both on the modalities being registered and the part of the body being studied. In some other intensity-based algorithms, the similarity measures work on derived image parameters such as image gradients, rather than the original voxel values.

For retrospective registration, a major attraction of intensity-based algorithms is that the amount of preprocessing or user-interaction required is much less than for point-based or surface-based methods. As a consequence, these methods are relatively easy to automate. The need for preprocessing does, however, mean that many intensity-based algorithms are restricted to a quite limited range of images. One of the aims of recent research in this area has been to devise general algorithms that will work on a wide variety of image types, without application-specific preprocessing.

Intensity-based registration algorithms can be used for a wide variety of applications: registering images with the same dimensionality, or different dimensionality; both rigid transformations and registration incorporating deformation; and both intermodality and intramodality images. Most algorithms are applicable to only a subset of these applications, but some are quite generally applicable.

In this section, we review the main similarity measures used for intensity-based image registration and describe the applications in which they are used. All these algorithms are iterative, so each similarity measure needs to be optimized. Issues specifically related to optimization are discussed in Section 8.5.2.

In this section we introduce some new notation. The images to be registered are $A$ and $B$. The sets of voxels in these images are $\{A(i)\}$ and $\{B(i)\}$, respectively. We will treat image $A$ as a reference image, and $B$ as an image that is iteratively transformed to $B' = \mathcal{T}(B)$ by successive estimates of the registration transformation $\mathcal{T}$. The transformation estimates will change the overlap between the images being registered. Voxel-similarity measures are invariably calculated for the set of voxels in the overlapping region of $A$ and $B'$, i.e., within $A \cap B'$, which is a function of $\mathcal{T}$ and so changes as the algorithm iterates. For some voxel-similarity measures, information from the intensity histogram is used, so we need to refer directly to intensity values in the image, rather than index voxels. Medical images may have 10 bits (1024 values), 12 bits (4096 values), or even 16 bits (65536 values) worth of intensity information per voxel. Many algorithms that use intensity information group voxel values into a smaller number of partitions, for example 64, 128, or 256 partitions. We refer to the sets of intensity partitions in images $A$ and $B'$ as $\{a\}$ and $\{b\}$, respectively, and the number of intensity partitions used as $N_a$ and $N_b$. Because the range of voxel intensities in an image is dependent on $\mathcal{T}$, $\{b\}$ may also be a function of $\mathcal{T}$.

### *8.5.1 Similarity measures*

#### *8.5.1.1 Image subtraction*

If the assumption is made that the images $A$ and $B$ being registered are identical, except for the misalignment, then an intuitively obvious similarity measure to use is the sum of squares of intensity differences (SSD). In this case, SSD will be zero when the images are correctly aligned and will increase with misregistration-registration error. In the slightly more realistic scenario in which the $A$ and $B$ differ only by Gaussian noise, then it can be shown that SSD is the optimum measure [87].

Certain image registration problems are reasonably close to this ideal case. For example, in serial registration of MR images, it is expected that the images being aligned will be identical except for small changes, which might result from disease progression or response to treatment. Similarly, in functional MR experiments, only a small number of the voxels are expected to change during the study, so all the images that need to be registered to correct for patient motion during the study are very similar to each other. If only a small fraction of the voxels being aligned are likely to have changed between image acquisitions, SSD is likely to work well. This approach has been used by Hajnal et al. [88] and is used in the SPM software by Friston et al. [89]. This approach can fail if the data diverges too much from the ideal case. For example, if a small number of voxels change intensity by a large amount, they can have a large effect on the change in squared intensity difference. For this reason, it is sometimes desirable to pre-segment parts of the image prior to registration. This preprocessing is commonly done for the scalp when carrying out serial MR brain registration, where the scalp can deform.

It is worth remembering that the assumption of Gaussian noise is frequently broken in medical images. In MR magnitude images, the noise is Rician [90], which is approximately Gaussian for high intensity parts of the image but is a long way from Gaussian in low intensity regions. Also, artifacts caused by reconstruction or subject motion introduce noise that is non-Gaussian.

**Algorithm 8.5: Sum of Squares of Differences**

For images $A$ and $B$ with voxels $i$, find the transformation $\mathcal{T}$ to minimize:

$$\mathrm{SSD} = \frac{1}{N} \sum_{i}^{N} |A(i) - B'(i)|^2 \qquad \forall i \in A \cap B' \;.$$

---

*8.5.1.2 Correlation Coefficient*

If the intensities in images $A$ and $B$ are linearly related, then the correlation coefficient CC can be shown to be the ideal similarity measure [87]. Once again, few registration applications will precisely conform to this requirement, but many intramodality applications come sufficiently close for this to be a useful measure.

**Algorithm 8.6: Correlation Coefficient**

For images $A$ and $B$ with voxels $i$, find the transformation $\mathcal{T}$ to maximize:

$$\mathrm{CC} = \frac{\sum_i (A(i) - \bar{A})(B'(i) - \bar{B}')}{\{\sum_i (A(i) - \bar{A})^2 \sum_i (B'(i) - \bar{B}')^2\}^{1/2}} \qquad \forall i \in A \cap B' \;,$$

where $\bar{A}$ and $\bar{B}'$ are the mean values of voxels in image $A$ and the transformed image $B$, respectively.

---

#### *8.5.1.3 Ratio-Image Uniformity*

An alternative intramodality registration measure was proposed by Woods in 1992 [91]. This algorithm was initially devised for registration of multiple PET images of the same subject and has subsequently been widely used for serial MR registration of the brain [92]. The name Ratio-Image Uniformity (RIU) has become used quite recently for this algorithm, and it is also known as Variation of Intensity Ratios (VIR). The RIU name does, however, explain well what the algorithm does.

For each estimate of the registration transformation, a ratio image $R$ is calculated by dividing each voxel in $A$ by each voxel in $B'$. The uniformity of $R$ is then determined by calculating the normalized standard deviation of $R$. The algorithm iteratively determines the transformation $\mathcal{T}$ that minimizes the normalized standard deviation, i.e., maximizes uniformity.

In some cases, in order to get good results, it may be necessary to preprocess the images to remove some anatomy. In the original PET-PET application, for example, it was necessary to segment the brain from the images, removing all extradural material so that both $A$ and $B$ include only brain voxels. This step is not necessary, however, for serial MR registration (see Section 8.5.3.2). Furthermore, an alternative registration will be obtained in almost all cases by reversing the roles of images $A$ and $B$ (i.e., the result will not be the inverse of the forward transformation). Woods proposed that possible bias of the registration solution produced by this asymmetry can be reduced by averaging the forward transformation with the inverse of the reverse transformation [91].

**Algorithm 8.7: Ratio-Image Uniformity**

For two images $A$ and $B'$, find $\mathcal{T}$ to minimize RIU, where RIU is calculated over the region $A \cap B'$ as follows:

1. Smooth and interpolate the images to have cubic voxels (optional), and determine the number of voxels $N$ in $B'$ that lie within $A \cap B'$.
2. Calculate the ratio image $R(i) = B'(i)/A(i)$.
3. Calculate the mean of $R$: $\mu_R = \frac{1}{N}\sum_i R(i)$.
4. Calculate the standard deviation of $R$: $\sigma_R = \frac{1}{N}\sum_i (R(i) - \mu_R)^2$.
5. RIU = the normalized standard deviation: $\sigma_R/\mu_R$.

#### *8.5.1.4 Partitioned Intensity Uniformity*

Shortly after proposing the RIU method for intramodality registration, Woods proposed a modified version of the algorithm for MR-PET registration [93]. This second algorithm, which we call Partitioned Intensity Uniformity (PIU), makes an idealized assumption that "all pixels with a particular MR pixel value represent the same tissue type so that values of corresponding PET pixels should also be similar to each other." The algorithm therefore minimizes the normalized standard deviation of PET voxel values for each MR intensity value (or partitioned intensity value). The PIU algorithm can usefully be thought of in terms of intensity histograms. For each intensity partition $a$ in the MR image, there will be $n_A(a)$ voxels distributed throughout the MR image volume that have an intensity within this partition. For a given estimate $\mathcal{T}$ of the registration transformation, there will be $n_A(a)$ PET voxels that are co-located with these MR voxels. The intensities of these PET voxels could be plotted as a histogram. The PIU algorithm aims to find the transformation $\mathcal{T}$ that minimizes the spread of that histogram for all partitions. The algorithm can fail unless the histogram for each partition is unimodal.

For MR-PET registration of head images it is normal to remove extra-dural tissue before running the algorithm. This preprocessing step helps to avoid bimodal or trimodal histograms. Successful registrations are possible even with a relatively crude segmentation process [82]. As with the RIU method, the registration obtained when $A$ and $B$ are interchanged will rarely be consistent with the original registration. The choice of PIU measure will depend on the application.

**Algorithm 8.8: Partitioned Intensity Uniformity**

For two images $A$ and $B'$, find $\mathcal{T}$ to minimize:

$$\mathrm{PIU} = \sum_{a \in \{a\}} \frac{n_A(a)}{N} \frac{\sigma_{B'}(a)}{\mu_{B'}(a)} ,$$

where $B'$ has been interpolated to have voxels of the same size and shape as $A$, the sum is carried out over the $N$ voxels in $A \cap B'$, $n_A(a)$ is the number of voxels in image $A$ with intensity $a$, and $\mu_{B'}(a)$ and $\sigma_{B'}(a)$ are the mean and standard deviation of the voxels in image $B'$ that co-occur with voxels whose intensities lie in partition $a$ in image $A$.

For MR-PET registration, 256 intensity partitions are typical.

*8.5.1.5 Joint histograms and joint probability distributions*

Statistical classifiers have been widely used in MR image analysis for segmentation of multi-spectral data for many years. In these approaches, a joint histogram is constructed from two images that are correctly aligned (e.g.: the first and second echo images from a spin-echo acquisition). A joint histogram is $n$-dimensional where $n$ is the number of images used to generate it. The axes of the histogram are the intensities (or intensity partitions) in each image, and the value at each point in the histogram is the number of voxels with a particular combination of intensities in the different spectral components.

If the joint histogram is normalized, it becomes an estimate of the joint probability distribution function (PDF) of intensities in the $n$ images.

**Algorithm 8.9: Calculating a joint PDF**

For two images $A$ and $B'$ related by a transformation $\mathcal{T}$, calculate the two-dimensional PDF for intensity partitions $\{a\}$ and $\{b\}$:

1. Allocate an $N_a$ by $N_b$ array HIST$[j,k]$.
2. Initialize the histogram: HIST$[j,k] = 0$ for all $j,k$.
3. For each voxel $i \in A \cap B'$, calculate intensity values $A(i)$ and $B'(i)$, calculate the intensity partition numbers $a$ and $b$ corresponding to $A(i)$ and $B'(i)$, and increment HIST$[a,b]$.
4. Calculate $\sum_{j,k}$ HIST$[j,k]$.
5. Normalize the histogram to calculate the PDF:

$$\mathrm{PDF}[j,k] = \frac{\mathrm{HIST}[j,k]}{\sum_{j,k} \mathrm{HIST}[j,k]} .$$

Note that elements in the array PDF are floating point, not integer.

For many image modality combinations, the PDF changes with $\mathcal{T}$. Example histograms as a function of $\mathcal{T}$ are shown in Figure 8.7. This observation lead several research groups to investigate similarity measures calculated from the PDF, which could be optimized to register the images. The attraction of this approach is that the observed change in PDF with $\mathcal{T}$ is qualitatively similar for many modality combinations; the signal in the PDF is “clustered” at registration but diffuses with misregistration.

The most successful similarity measures calculated from the PDF are based on information theory.

*8.5.1.6 Joint entropy*

The Shannon entropy $H$ is widely used as a measure of information in many branches of engineering. It was originally developed as part of information theory in the 1940s and describes the average information supplied by a set of symbols $\{s\}$ whose probabilities are given by $\{p(s)\}$.

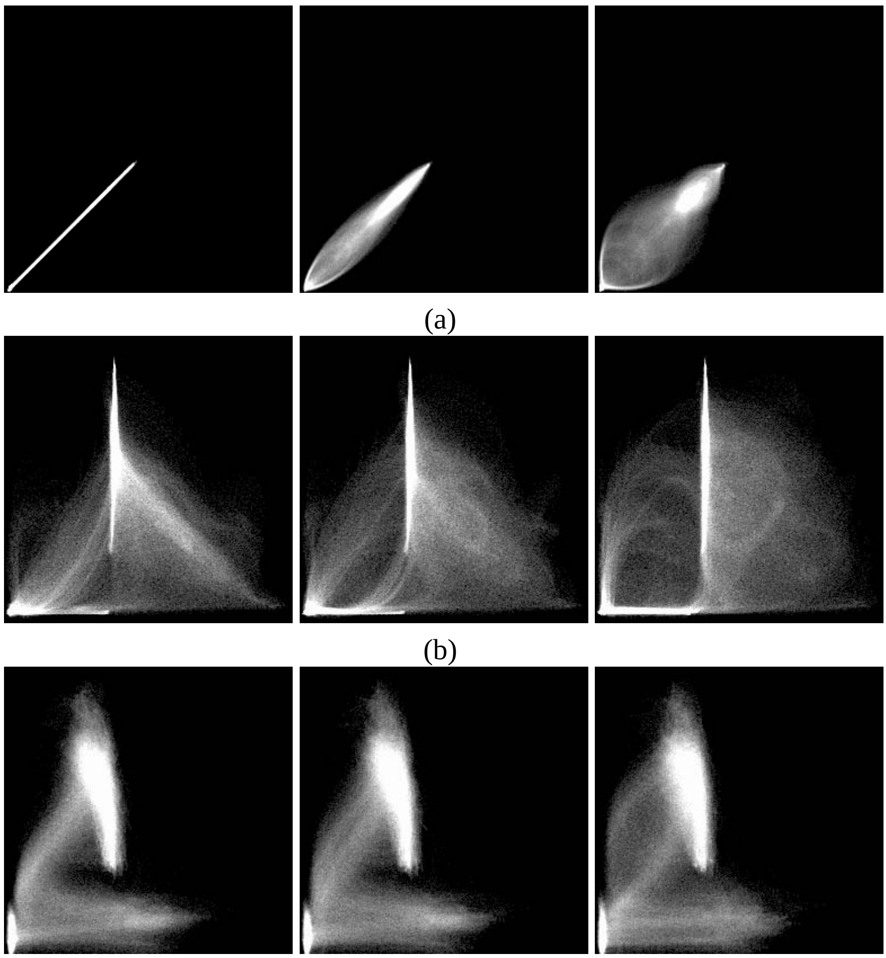
$$H = -\sum_s p(s) \log p(s) \ .$$

If all symbols $s$ have equal probability, then entropy will be at maximum. If one symbol has a probability of 1 and all others have a probability of zero, then entropy will have a minimum value.

At first sight, image registration has little to do with measuring the amount of information being transmitted down a communication channel. The use of entropy and other information-theoretic measures for image registration came about, however, after inspection of joint histograms and PDFs. When the images are correctly aligned, the joint histograms have tight clusters, surrounded by large dark regions. These clusters disperse as the images become less well registered. The tight clusters in the histograms at registration represent a small number of symbols $s$ having high probabilities $p(s)$. Surrounding dark regions in the joint histogram represent large numbers of symbols with probability zero. As the clusters disperse, the high intensity regions of the joint histogram become less intense (their probabilities decrease), and previously dark regions in the histograms become brighter (there are fewer histogram entries with zero or very low probabilities). Misregistration, therefore, results in an increase in histogram entropy.

As a consequence of this observation, it was proposed that the entropy of the PDF calculated from images $A$ and $B'$ should be iteratively minimized to register these images.

Minimizing joint histogram entropy to register images can be thought of as an extension of PIU minimization, Algorithm 8.8. PIU minimization tends to minimize the spread of the histogram of voxels in image $B'$ for each intensity partition in image $A$. Provided this histogram is unimodal, then minimizing the spread will also minimize the entropy. The joint entropy measure has two advantages over PIU. Firstly, it minimizes the spread of clusters in two dimensions rather than just one. Secondly, minimizing entropy does not require that the histograms are unimodal in the way that minimizing variance does. For this reason, joint entropy would seem more generally applicable to multimodality registration than PIU, and also obviates the need to segment some parts of the image in order to avoid bi-modal histograms.



(a)

(b)

(c)

**Figure 8.7:** Example joint histograms, or joint probability distribution functions, from Hill et al. [94]. These are produced from images $A$ and $B'$ using Algorithm 8.9. Intensities from image $A$ are plotted along the horizontal axis, and those from image $B'$ on the vertical axis. In row (a), $A$ and $B$ are identical MR images of the head. In row (b), $A$ is a CT image of the head and $B$ an MR scan of the head, and in row (c), $A$ is an MR image of the head, and $B$ a PET image of the head. For all modality combinations, the histograms are calculated for three different transformations $\mathcal{T}$: the identity (left), a lateral translation of 2mm (center) and a lateral translation of 5 mm (right).

**Algorithm 8.10: Minimization of Joint Entropy**

For the PDF$[j,k]$ calculated from images $A$ and $B$ using algorithm 8.9, find the transformation $\mathcal{T}$ to minimize:

$$H = -\sum_{j,k} \text{PDF}[j,k] \log \text{PDF}[j,k] \, .$$

Note: For most registration applications, a slight modification to this measure is needed, as described in Sections 8.5.1.7 and 8.5.1.8.

---

*8.5.1.7 Mutual information*

Entropy minimization as described in Section 8.5.1.6 is not a robust voxel-similarity measure for all types of image registration. The problem is that the PDF from which the joint entropy is calculated is defined only for the region of overlap between the two images, i.e., within $A \cap B'$. The range and distribution of intensity values in the portion of either image that overlaps with the other is a function of $\mathcal{T}$. The change in overlap with $\mathcal{T}$ can lead to histogram changes that mask the clustering effects described above.

The solution to this difficulty, proposed independently by researchers at Leuven, Belgium [95] and MIT in the US [96], is to use the information-theoretic measure mutual information (MI) instead of entropy $H$. MI normalizes the joint entropy with respect to the partial entropies of the contributing signals. In terms of image registration, this measure takes account of the change in the intensity histogram of images $A$ and $B'$ with $\mathcal{T}$.

**Algorithm 8.11: Maximizing Mutual Information**

Find the transformation $\mathcal{T}$ to maximize MI$(A, B')$ as follows:

1. Calculate the PDF$[j,k]$ from images $A$ and $B'$ using Algorithm 8.9.
2. Calculate the joint entropy $H(A, B')$ using the formula in Algorithm 8.10.
3. Calculate the marginal entropies $H(A)$ and $H(B')$:

$$H(A) = \sum_j \left( \sum_k \text{PDF}[j,k] \log \sum_l \text{PDF}[j,l] \right) \, ,$$

$$H(B') = \sum_k \left( \sum_i \mathrm{PDF}[i,k] \log \sum_j \mathrm{PDF}[j,k] \right) .$$

4. Calculate the mutual information $\mathrm{MI}(A, B')$:

$$\mathrm{MI}(A,B') = H(A) + H(B') - H(A,B') .$$

---

#### 8.5.1.8 *Normalization of mutual information*

Mutual information overcomes many of the shortcomings of joint entropy but can still fail for some types of clinical image, particularly those which contain large amounts of air (noise) around the outside of the subject. Improved performance of mutual information can be obtained by various normalization schemes. These algorithms are not taken from the information theory literature but have been arrived at through experiments on medical images. Despite its heuristic origins, the variant given below (from [97]) works extremely well in practice. Current validation studies have shown that it works at least as well as mutual information and in some cases performs better.

**Algorithm 8.12: Maximizing Normalized Mutual Information**

Perform Algorithm 8.11 but replace $\mathrm{MI}(A, B')$ with:

$$\mathrm{NMI}(A,B') = \frac{H(A) + H(B')}{H(A,B')} .$$

---

### 8.5.2 *Capture ranges and optimization*

#### 8.5.2.1 *Optimization*

In order to register two images using Algorithms 8.5 to 8.12, it is necessary to find the optimal value of the similarity measure over a parameter space with dimensionality defined by the number of degrees of freedom of $\mathcal{T}$. Each registration algorithm that makes use of a voxel-similarity measure tends to use a different optimization algorithm. There are two classes of optimization algorithms that can be used: those that use derivative information and those that do not. Where derivatives can be calculated efficiently, they should be used to speed up optimization. A detailed discussion of optimization techniques is outside the scope of this chapter. A review of optimization techniques is given in Numerical Recipes in C [61]. Some techniques are also discussed in Chapter 1.

### *8.5.2.2 Capture ranges*

In many optimization problems it is desirable to determine the globally optimal solution. For image registration, however, the desired optimum is frequently a local rather than global extremum of the voxel-similarity measure. This has several important consequences for medical image registration. The reason that the global optimum is not desirable can be understood by careful consideration of the behavior of the similarity measures for large transformations. For the sum of squares of difference (SSD) measure, we have previously assumed that for two images that are very similar the smallest value of SSD will arise when the images are correctly aligned and that SSD will increase with misregistration. However, if we consider two head images surrounded by low intensity noise, then we may be able to obtain a lower value of SSD if we misalign the images so completely that only the noise in the two images overlaps. This transformation produces a registration that is clearly incorrect, since no features of interest in the images overlap, but it is nevertheless a more nearly optimal registration according to the similarity measure. Similarly, with regard to the information-theoretic measures, we have been assuming that the clustering of features in the joint histogram decreases with increasing misregistration. Once again, if we transform the images sufficiently so that only background noise overlaps, we observe a very tight cluster in that area of the histogram corresponding to background intensities in both images. This transformation will produce a lower joint entropy, and a higher mutual information, than the correct alignment.

The fact that the desired optimum is local rather than global does not invalidate the use of voxel-similarity measures for medical image registration. It does, though, have implications for robust implementations.

The correct local optimum value of the similarity measure will be the optimum value within a portion of parameter space termed the *capture range*. Provided the starting estimate of $\mathcal{T}$ is within the capture range, the registration algorithm can be expected to converge to the correct solution. The capture range will depend on the field of view and intensity information in the images, so cannot be known a priori. Intuitively, it might seem that the starting estimate should be closer to the correct solution, than to the solution in which only background voxels overlap, and it has been shown that for MR-CT and MR-PET registration, similarity measures will converge to the correct solution within about 20–30 mm and 20–30 degrees of this solution [98, 99]. These ranges should be treated as a "rule of thumb." It is essential when registering images using these algorithms to inspect the solution visually. A solution that lies outside the capture range is immediately recognizable as incorrect, and a better solution can be found by re-starting the algorithm with a user-provided starting estimate that is nearer to the correct solution.

A further implication of the existence of a capture range is that the direct application of stochastic optimization algorithms such as simulated annealing and genetic algorithms are inappropriate for medical image registration using voxel-similarity measures.

#### *8.5.2.3 Optimizing similarity measures for sub-voxel registration*

The calculation of similarity measures for a given value of $\mathcal{T}$ in general requires interpolation of the images. For an iterative registration algorithm, fast interpolation (most commonly linear) is required. Because the interpolation is not perfect, it alters the intensity histogram. Linear interpolation, for example, causes a spatially varying, low-pass filtering of the images [100], whose interaction with edges will tend to smooth the histogram, and in spatially smooth areas will reduce noise and hence contribute to a sharpening of the histogram. The consequence of this effect can be an interpolation-dependent change in the value of the similarity measure, which can cause fluctuations of the similarity measure with the period of the voxel separation. When using an entropy-based measure, blurring of the image results in increased entropy. The blurring is zero at positions coinciding with voxel locations and is maximum at displacements of half a voxel from these positions. In multi-modality registration, the required registration accuracy is often no better than the dimensions of a single voxel, so these interpolation errors are unimportant. In some multimodality applications, however, and also in intramodality applications such as serial MR registration, an accuracy considerably better than the dimension of a voxel is required. In these cases, the artifacts caused by interpolation can introduce local extrema in parameter space, leading to reduced registration accuracy. The interpolation artifacts are greatest for high spatial frequency features in the images. One solution to the problem is, therefore, to blur the images prior to carrying our registration, thereby reducing the artifacts [101]. The effect is also reduced if the starting estimate of $\mathcal{T}$ includes a rotational component, or if the voxel dimensions in the images being registered are different.

### ***8.5.3 Applications of intensity-based methods***

#### *8.5.3.1 Types of transformation*

In the discussion of Algorithms 8.5 to 8.12, we have not stated the form of geometrical transformation $\mathcal{T}$ that we wish to find. Many of these measures have been applied to a wide variety of registration problems, including:

- rigid registration of 3-D images of the same subject taken at different times using the same modality
- rigid registration of 3-D images of the same subject taken at different times using different modalities

- rigid registration of 2-D projection images and 3-D images of the same subject taken at different times
- nonrigid registration of images of the same subject taken at different times
- nonrigid inter-subject registration of the 3-D images of the head.

Mutual information and normalized mutual information have been demonstrated to provide good quality image registration for all these types of applications. The validation of the registration quality is a particularly difficult problem when calculating nonrigid transformations and is a rather poorly defined problem when aligning images from different subjects. See Chapter 10 for a discussion of validation methods. In the sections below, we describe the application of voxel-similarity measures to a selection of medical image registration tasks.

#### *8.5.3.2 Serial MR*

It is increasingly common for a patient to have multiple magnetic resonance images acquired at different times, in order to monitor disease progression or response to treatment. Traditional radiological viewing of these images involves printing the images onto film and viewing them side-by-side on light boxes. This approach can make it difficult to identify small changes in the images between scans. A more sensitive technique is to subtract the images, generating difference images, and to view the difference images to identify regions of the images that have changed. The generation of difference images requires that the images acquired at the different times have identical contrast characteristics and that any drift in the scanner scaling parameters is corrected using suitable quality-assurance procedures, such as phantom scanning [102, 103]. Serial MR registration also requires accurate alignment, because registration errors of less than a voxel can lead to artifacts that can mask the changes of interest in the patient. The required registration accuracy will be a function of the spatial frequency content of the images and of the contrast-to-noise ratio. The required accuracy is often stated as of the order of tens or hundreds of microns.

Serial MR registration is almost invariably applied to the brain and makes use of a rigid transformation. Indeed, for visual inspection of difference images, it is undesirable to use nonrigid registration transformations because they may produce a transformation that removes the changes of interest.

With serial MR registration the images are likely to be very similar, and so the measures SSD and CC are widely used. RIU, MI, and NMI are also frequently used. The advantage of the information-theoretic measures in this case is that they are less sensitive to small numbers of voxels that change by large amounts, than are the SSD and CC measures. When using SSD or CC, data preparation by segmentation of extra-dural tissue, or small lesions that have changed substantially in intensity, is often essential. The information-theoretic approaches should be less sensitive to these small numbers of outlying voxels, but where the changes are large, data pre-processing may still be required.

For many applications, the choice of an optimization algorithm that will iteratively find the optimum transformation to subvoxel accuracy is as important as the choice of similarity measure.

In serial MR registration, the choice of interpolation algorithm is also important. The use of tri-linear interpolation is equivalent to convolving the data with a triangular kernel of variable width, resulting in a spatially varying blurring of the data. When subtracting image $A$ from the blurred image $B'$, edge artifacts result. Higher order interpolation methods, such as those that incorporate truncated sinc kernels [88] or skew transformations [104], are more effective.

#### *8.5.3.3 MR and CT*

Registration of MR and CT images is most often applied to images of the head, and is used for surgery and radiotherapy planning. The registration transformation determined is usually a rigid one. On some occasions, however, it may be beneficial to search for an affine transformation because it can generate more accurate registration in cases when there are scaling or skew errors in the data. When the images being registered are subsequently going to be used for guiding treatment (image-guided surgery or radiation therapy), then it is not advisable to use an affine transformation unless one modality is known to have satisfactory geometrical properties and is treated as the reference image $A$. Otherwise, the images may be well registered but measurements made in the registered images may be inaccurate. It is preferable in these cases to calibrate the imaging devices as part of a routine quality-assurance procedure and to rectify the images prior to registration to remove any residual geometric distortion (see Section 8.2.3). The most problematic aspects of imaging geometry are geometric distortion in MR and errors in CT bed speed and gantry tilt angle. For 3-D gradient echo MR images, object dependent distortion is predominantly in the readout direction and can be minimized by selecting a high bandwidth-per-pixel (readout-gradient strength) in that direction (at a slight cost in signal to noise ratio). The residual scaling errors in MR due to gradient strength errors, and the CT errors, can be corrected using phantom experiments. For some CT systems, it is advisable to carry out phantom measurements with the CT couch loaded with weight similar to that of a patient.

Since MR and CT images have very different intensity distributions, image subtraction is never used for viewing the aligned images. As a result, the very high accuracy requirements of serial MR registration do not apply here. The accuracy requirement for MR-CT registration is usually governed by the accuracy of the treatment system that will subsequently be used. Since neurosurgery and radiosurgery systems are accurate to a small number of millimeters, registration accuracy of about 1 mm is normally sufficient.

A study by West et al. established that, if the images are free from motion artifacts and have been rectified to remove scaling errors and distortion, registration accuracy of better than 1 mm is possible when information-theoretic measures are used [82]. It is important to note that the images used in that validation study had similar fields of view in both modalities, covering most of the head, and had very similar slice orientations. It is possible to register MR and CT images that have very different slice orientations and fields of views. For example, it is possible to register coronal CT images with axial MR images or to register CT images with a field of view of about 20 mm around the temporal bone to MR images of the entire head. In both these cases, the rigid transformation solution may include translations of several tens of millimeters and rotations of many tens of degrees. In such cases, it is advisable to provide the algorithm with an initial estimate of $\mathcal{T}$ to increase the likelihood that it is starting within its capture range (see Section 8.5.2). Also, since the overlap between images at the correct registration transformation includes only a small portion of one or both modalities, the registration accuracy in these cases may be lower than in the validation study by West et al. [82].

#### *8.5.3.4 MR or CT and PET*

Registration of MR or CT images with PET images of the head was one of the first applications of medical image registration [105]. Because of the relatively low resolution of PET images, it is frequently desirable to make use of anatomical detail from MR or CT images to assist in their interpretation. The resolution of PET images is of the order of 6–10 mm, and a registration accuracy of 2–4 mm is considered desirable. Although the low resolution of PET images reduces the required registration accuracy compared to MR-CT registration, there are several difficulties in this registration application. Firstly, some PET tracers are very specific to certain parts of the brain. Although several algorithms, including Partitioned Intensity Uniformity (Algorithm 8.8) and mutual information (Algorithm 8.11), have been shown to work well when registering MR images to PET fluoro-deoxyglucose images [82], the accuracy of registration of the more specific tracers will be tracer-dependent and may be much lower. Secondly, for patients with large lesions, the PET images can have very large intensity abnormalities. A lesion that makes a small difference to image intensities in an MR scan may make a very large difference to intensities in the PET image. Many of the images used in the validation study by West et al. [82] had large lesions of this sort, and the median registration accuracy of both PIU and MI was better than 4 mm, suggesting that these measures are reasonably robust to these intensity abnormalities.

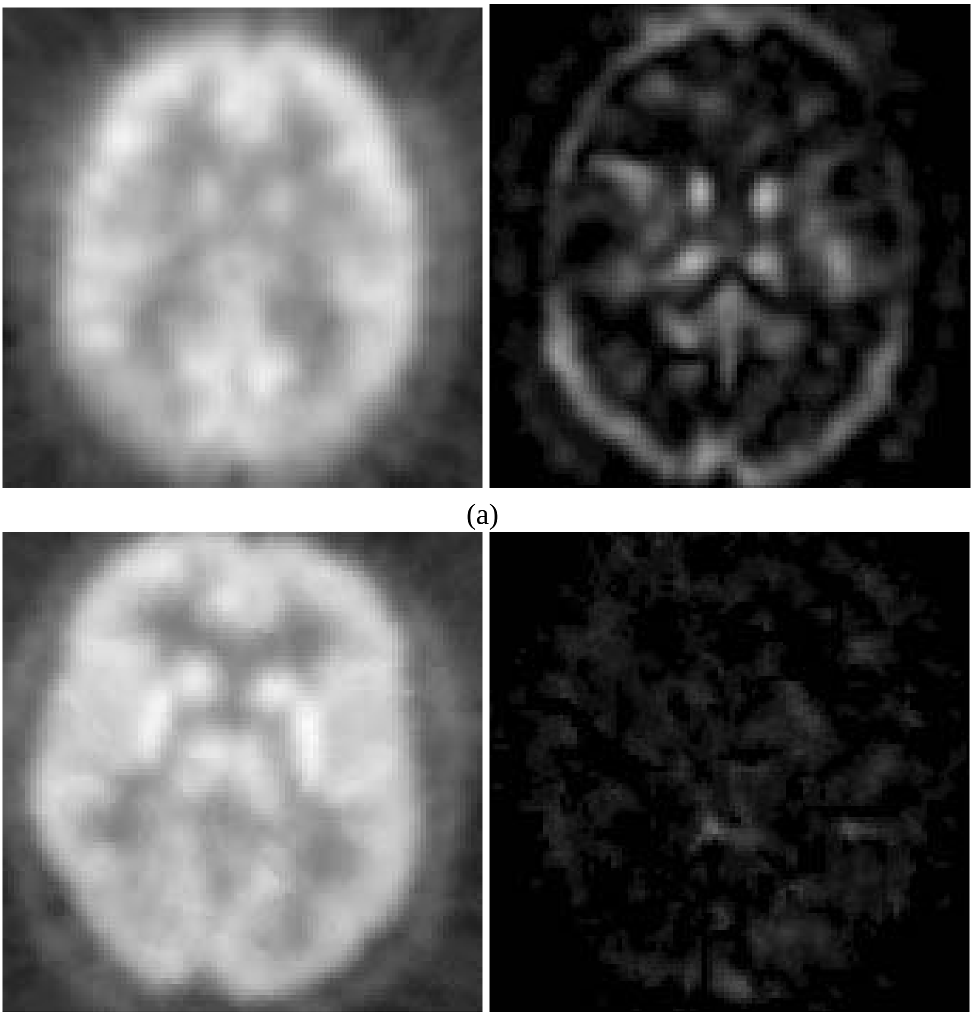
#### *8.5.3.5 Nonrigid 3-D registration*

This chapter focuses on rigid registration algorithms because these are the most mature image registration methodologies. There is, however, considerable research going on in extending the use of intensity-based registration algorithms to nonrigid transformations. While we do not attempt in this section to a give a comprehensive review of this field, we do explain how voxel-similarity measures can be extended beyond rigid or affine transformations.

It is possible to apply voxel-similarity measures to nonrigid registration by placing a regular grid of $N$ nodes across images $A$ and $B$ being registered. The position of these nodes in $B$ is then iteratively altered while optimizing a voxel-similarity measure. One approach is to calculate a rigid or affine transformation for a sub-image around each node, and then to smooth the resulting set of discrete transformations [106]. Alternatively, each node can be treated as a spline knot, and the nodes iteratively adjusted, with the the entire image being transformed using the spline transformation at each iteration. The approximating B-spline, which is described by Eqs. (8.15) and (8.16), has the advantage that change in position at a node has only a local effect on the deformation (extending about two node separations), and this approach is used in an implementation by Rueckert et al. [107].

The first approach involves multiple registrations of subimages. The accuracy of registration at each sub-image will be dependent on the number of voxels in the sub-images and the intensity information contained. Each subimage registration involves 6 or 12 degrees of freedom (for rigid and affine transformations respectively), and this is carried out $N$ times. Because each subimage contains only a small number of voxels, the information-theoretic approaches may suffer from a poor estimate of the PDF, and hence noisy values of the voxel-similarity measures. The SSD and CC measures will be less sensitive to this problem. The second approach involves three translational degrees of freedom per node, producing $3N$ degrees of freedom in total, with the similarity measure being calculated for the entire images, rather than just for subimages. This approach has the advantage that large numbers of voxels are used in each iteration, making the information-theoretic measures practicable.

These algorithms have been applied to both intrapatient registration in which there is tissue deformation and to interpatient registration. Example mean and variance images from a cohort of PET FDG patients are shown in Fig. 8.8. In this figure, an affine transformation and a spline interpolation are compared for use in interpatient registration. The similarity measure is, in both cases, normalized mutual information. The mean image is sharper, and the variance image has lower values when the transformation with larger numbers of degrees of freedom is used. Appropriate techniques for validation of these algorithms remain a matter of debate. By changing the spacing of the nodes, the algorithms are able to find a transformation with more degrees of freedom. If the number of nodes is too low, then the transformation may be too smooth to align the images. If the number of nodes is too high, then the transformations may inappropriately "correct" for features of interest, such as lesions.



(a)

(b)

**Figure 8.8**: Mean and variance images generated from a cohort of five PET images of different subjects. Row (a) shows an example axial slice from the mean (left) and variance (right) images after registration using an affine transformation with 12 degrees of freedom. Row (b) shows the corresponding slices after registration using a transformation with 2160 degrees of freedom. The transformation in (b) is determined by iteratively adjusting the positions of B-spline control points on a 20 mm grid. Normalized mutual information was the voxel similarity measure used by both algorithms. The variance images are displayed with the same intensity scale in both cases.

#### *8.5.3.6 2-D-to-3-D registration*

Many medical imaging modalities generate images by perspective projection. (See Sections 8.2.2 and 8.2.2.4 above.) The most common of these is conventional radiography or fluoroscopy. However, with video endoscopy and microscopy becoming more widely used, video images are increasingly common. Aligning projection images with 3-D tomographic modalities is a type of 2-D-to-3-D registration and requires an estimate of both the 3-D orientation of the objects being registered and the projection transformation. For rigid structures, the 3-D orientation information is a rigid transformation, so $\mathcal{T}$ is the composition of a rigid transformation and the projection transformation.

In many medical imaging applications, the projection transformation can be measured as part of a calibration procedure, so the registration problem has just six degrees of freedom.

The approach used for registration of either video images or radiographs to tomographic images is to optimize a similarity measure while iteratively transforming the tomographic image. For each iteration, a simulated projection image is generated from the tomographic image, and the similarity measure is calculated from these. In this case, image $A$ is the projection image, and $B$ the tomographic image. For each estimation of the transformation $\mathcal{T}$, image $B'$ is a simulated projection generated from the tomographic image $B$. For example, when registering a radiograph $A$ to a CT scan $B$, a digitally reconstructed radiograph $B'$ is produced using $\mathcal{T}$ and a model of the x-ray image formation process to simulate an x-ray projection by integrating the CT voxel intensities. This approach has been used by several authors [108, 109]. When registering a video image $A$ to a tomographic image $B$, $B'$ can be a perspective surface rendering of the image structures in $B$ that are visible in $A$ [110, 111].

## 8.6 Conclusion

This chapter represents the current state of the field of medical image registration. Registration is defined as the determination of a geometrical transformation that aligns points in one view of an object with corresponding points in another view of that object or another object. Using this definition, the chapter treats registration as it is applied to images of anatomy that have been acquired by medical imaging modalities, especially the tomographic modalities, such as CT, MR, and PET. Because the body moves in three dimensions, the emphasis is on three-dimensional registration problems. Commonly used classes of geometrical transformations are presented with an emphasis on rigid and scaled transformations. Methods of registration are divided into point-based, surface-based, and intensity-based; well-established algorithms are described for all three categories, and both theoretical and practical discussions of their applications are provided.

The emphasis in this chapter is on rigid registration because, as of this writing, most of the work and most of the progress in registration has been made in this area.

Rigid registration is important because of its simplicity and because of its many applications to the relatively rigid head and to orthopedics, especially in image-guided surgery. Thanks to the researchers listed in the bibliography of this chapter, much is now known about this area, and many effective algorithms have been developed, some of the best of which are described above. These rigid registration problems include both prospective registration, in which fiducial markers are employed, and retrospective registration, in which they are not. Recent research has demonstrated that the most effective registration algorithms for many types of retrospective application are those based on intensities. Point-based and surface-based methods can also be used for these applications, but they require a greater degree of user interaction and have typically exhibited lower accuracy than the intensity-based methods. Techniques based on points and surfaces do, however, play an important role in the registration of images to physical space because the internal information necessary for intensity-based registration is typically unavailable in physical space. Research in point-based and surface-based approaches to this latter problem are central to the advancement of image-guided surgery and radiosurgery.

While the head and bones are rigid, most of the rest of the body is nonrigid, and, therefore, nonrigid transformations must be employed to register images of most of the anatomy. Furthermore, interpatient registration requires nonrigid transformation for all parts of the anatomy. These transformations are far more complex and varied than rigid ones. The appropriate form of nonrigid transformation is for intrapatient registration strongly affected by the mechanics of the tissue involved and for interpatient registration is determined by the natural variation in anatomic development. While there are still many important problems to be solved in the area of rigid registration, it seems likely that much of the future work in the field of medical image registration will focus on nonrigid registration.

## 8.7 Acknowledgments

Much of the work on the theory of errors in point-based registration reported in Section 8.3 is the result of a collaboration with Jay West of the Department of Electrical Engineering and Computer Science, Vanderbilt University. JMF is grateful to Professors David Hawkes and Derek Hill of the Radiological Sciences Department, King's College London for their collaboration with him on point-based registration and for their support of his visit to their laboratory during June–July, 1999. He also benefited from many discussions with Dr. Philippe Batchelor of that laboratory during his visit. The images in Fig. 8.3 were provided as part of the "Evaluation of Retrospective Registration" project, sponsored by the National Institutes of Health, grant number R01 NS33926. JMF gratefully acknowledges support for work on point-based registration provided by the National Science Foundation through grant number BES-9802982, by the UK Engineering and Physical Sciences Research Council Grant GR/M85753, entitled "Improving and Predicting the Accuracy of Image Guided Surgery," and by the Vanderbilt University Research Council. CRM gratefully acknowledges support for work on surface-based registration provided by the Ronald L. Bittner Brain Tumor Research Fund. DLGH is grateful to his colleagues and students in Radiological Sciences at King's College London for their suggestions about Section 8.5. In particular, he is grateful to David Atkinson and Daniel Rueckert for useful discussions (both funded by the UK Engineering and Physical Sciences Research Council). The authors are grateful to Matthew Clarkson of the Radiological Sciences Department, King's College London for helping produce Fig. 8.6.

### 8.8 References

[1] J. B. A. Maintz and M. A. Viergever, "A survey of medical image registration," *Med. Image Anal.*, vol. 2, pp. 1–36, 1998.

[2] C. R. Maurer, Jr. and J. M. Fitzpatrick, "A review of medical image registration," in *Interactive Image-Guided Neurosurgery* (R. J. Maciunas, ed.), pp. 17–44, Park Ridge, IL: American Association of Neurological Surgeons, 1993.

[3] P. A. van den Elsen, *Multimodality matching of brain images*. Ph.D. thesis, Utrecht University, Utrecht, The Netherlands, 1993.

[4] D. L. Collins, A. P. Zijdenbos, V. Killokian, J. G. Sled, N. J. Kabani, C. J. Holmes, and A. C. Evans, "Design and construction of a realistic digital brain phantom," *IEEE Trans. Med. Imaging*, vol. 17, pp. 463–468, 1998.

[5] H. Goldstein, *Classical Mechanics*. Reading, MA: Addison-Wesley, 1950.

[6] G. H. Rosenfield, "The problem of exterior orientation in photogrammetry," *Photogrammetric Engineering*, vol. 25, pp. 536–553, 1959.

[7] B. K. P. Horn, *Robot Vision*. Cambridge, MA: MIT Press, 1986.

[8] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Opt. Soc. Amer. A*, vol. 4, pp. 629–642, 1987.

[9] K. C. Fu, R. C. Gonzalez, and C. S. G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*. New York: McGraw-Hill, 1987.

[10] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, vol. II. Reading, MA: Addison-Wesley, 1993.

[11] A. Goshtasby, "Piecewise linear mapping functions for image registration," *Pattern Recognit.*, vol. 19, pp. 459–466, 1986.

[12] G. Q. Maguire, Jr., M. E. Noz, H. Rusinek, J. Jaeger, E. L. Kramer, J. J. Sanger, and G. Smith, "Graphics applied to medical image registration," *IEEE Comput. Graph. Appl.*, vol. 11, pp. 20–29, March 1991.

[13] E. V. Shikin and A. I. Plis, *Handbook on Splines for the User*. New York: CRC Press, 1995.

[14] L. L. Schumaker, *Spline Functions: Basic Theory*. New York: John Wiley & Sons, 1981.

[15] T. Pavlidis, *Algorithms for Graphics and Image Processing*. Rockville, Maryland: Computer Science Press, 1982.

[16] D. H. Ballard and C. M. Brown, *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1982.

[17] R. L. Harder and R. N. Desmarais, "Interpolation using surface splines," *J. Aircraft*, vol. 9, pp. 189–191, 1972.

[18] J. Duchon, "Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces," *R.A.I.R.O. Analyse Numérique*, vol. 10, pp. 5–12, 1977.

[19] J. Meinguet, "An intrinsic approach to multivariate spline interpolation at arbitrary points," in *Polynomial and Spline Approximation*, pp. 163–190, Boston: D. Reidel Publishing Company, 1978.

[20] A. Goshtasby, "Registration of images with geometric distortions," *IEEE Trans. Geosci. Remote Sensing*, vol. 26, pp. 60–64, 1988.

[21] F. L. Bookstein, "Principal warps: Thin-plate splines and the decomposition of deformations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, pp. 567–585, 1989.

[22] R. Franke, "Scattered data interpolation: tests of some methods," *Mathematics of Computation*, vol. 38, pp. 181–200, 1982.

[23] G. M. Nielson, "Scattered data modeling," *IEEE Comput. Graph. Appl.*, vol. 13, pp. 60–70, 1993.

[24] M. Fornefett, K. Rohr, and H. S. Stiehl, "Radial basis functions with compact support for elastic registration of medical images," in *Biomedical Image Registration* (F. F. Pernuš, S. Kovačič, H. s. Stiehl, and M. A. Viergever, eds.), Ljubljana, Slovenia: Slovenian Pattern Recognition Society, 1999.

[25] C. Broit, *Optimal registration of deformed images*. Ph.D. thesis, University of Pennsylvania, Pennsylvania, USA, 1981.

[26] R. Bajcsy and S. Kovacic, "Multiresolution elastic matching," *Comput. Vision Graph. Image Processing*, vol. 46, pp. 1–21, 1989.

[27] J. C. Gee, M. Reivich, and R. Bajcsy, "Elastically deforming a three-dimensional atlas to match anatomical brain images," *J. Comput. Assist. Tomogr.*, vol. 17, pp. 225–236, 1993.

[28] P. Thompson and A. W. Toga, "A surface-based technique for warping three-dimensional images of the brain," *IEEE Trans. Med. Imaging*, vol. 15, pp. 402–417, 1996.

[29] G. E. Chirstensen, R. D. Rabbitt, and M. I. Miller, "Deformable templates using large deformation kinematics," *IEEE Trans. Med. Imaging*, vol. 5, pp. 1435–1447, 1996.

[30] M. H. Davis, A. Khotanzad, D. P. Flamig, and S. E. Harms, "A physics-based coordinate transformation for 3-D image matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, pp. 317–328, 1997.

[31] M. I. Miga, K. D. Paulsen, J. M. Lemery, S. D. Eisner, A. Hartov, F. E. Kennedy, and D. W. Roberts, "Model-updated image guidance: initial clinical experiences with gravity-induced brain deformation," *IEEE Trans. Med. Imaging*, vol. 18, pp. 866–874, 1999.

[32] W. E. Smith, N. Vakil, and S. A. Maislin, “Correction of distortion in endoscope images,” *IEEE Trans. Med. Imaging*, vol. 11, pp. 117–122, 1992.

[33] H. Chang and J. M. Fitzpatrick, “A technique for accurate magnetic resonance imaging in the presence of field inhomogeneities,” *IEEE Trans. Med. Imaging*, vol. 11, pp. 319–329, 1992.

[34] T. S. Sumanaweera, G. H. Glover, T. O. Binford, and J. R. Adler, “MR susceptibility misregistration correction,” *IEEE Trans. Med. Imaging*, vol. 12, pp. 251–259, 1993.

[35] D. L. G. Hill, D. J. Hawkes, M. J. Gleeson, T. C. S. Cox, A. J. Strong, W.-L. Wong, C. F. Ruff, N. D. Kitchen, D. G. T. Thomas, J. E. Crossman, C. Studholme, A. J. Gandhe, S. E. M. Green, and G. P. Robinson, “Accurate frameless registration of MR and CT images of the head: Applications in surgery and radiotherapy planning,” *Radiology*, vol. 191, pp. 447–454, 1994.

[36] M. Y. Wang, J. M. Fitzpatrick, and C. R. Maurer, Jr., “Design of fiducials for accurate registration of CT and MR volume images,” *Medical Imaging 1995: Image Processing*, vol. Proc. SPIE 2434, pp. 96–108, 1995.

[37] C. R. Maurer, Jr., J. M. Fitzpatrick, M. Y. Wang, R. L. Galloway, Jr., R. J. Maciunas, and G. S. Allen, “Registration of head volume images using implantable fiducial markers,” *IEEE Trans. Med. Imaging*, vol. 16, pp. 447–462, 1997.

[38] J. R. Hurley and R. B. Cattell, “The Procrustes program: Producing direct rotation to test a hypothesized factor structure,” *Behav. Sci.*, vol. 7, pp. 258–262, 1962.

[39] B. F. Green, “The orthogonal approximation of an oblique structure in factor analysis,” *Psychometrika*, vol. 17, pp. 429–440, 1952.

[40] D. G. Kendall, “A survey of the statistical theory of shape,” *Statist. Sci.*, vol. 4, pp. 87–120, 1989.

[41] C. Goodall, “Procrustes methods in the statistical analysis of shape,” *J. R. Statist. Soc. B*, vol. 53, pp. 285–339, 1991.

[42] F. L. Bookstein, *Morphometric Tools for Landmark Data: Geometry and Biology*. Cambridge: Cambridge University Press, 1991.

[43] P. H. Schönemann, “A generalized solution of the orthogonal Procrustes problem,” *Psychometrika*, vol. 31, pp. 1–10, 1966.

[44] J. L. Farrell and J. C. Stuelpnagel, “Problem 65–1: A least squares estimate of satellite attitude,” *SIAM Rev.*, vol. 8, pp. 384–386, 1966.

[45] G. H. Golub and C. F. van Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins University Press, 1983.

[46] K. S. Arun, T. S. Huang, and S. D. Blostein, “Least-squares fitting of two 3-D point sets,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 9, pp. 698–700, 1987.

[47] S. Umeyama, “Least-squares estimation of transformation parameters between two point patterns,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, pp. 376–380, 1991.

[48] R. H. Wessner, “Problem 65–1: A least squares estimate of satellite attitude,” *SIAM Rev.*, vol. 8, p. 386, 1966.

[49] B. K. P. Horn, H. M. Hilden, and S. Negahdaripour, "Closed-form solution of absolute orientation using orthonormal matrices," *J. Opt. Soc. Amer. A*, vol. 5, pp. 1127–1135, 1988.

[50] R. Sibson, "Studies in the robustness of multidimensional scaling: Procrustes statistics," *J. R. Statist. Soc. B*, vol. 40, pp. 234–238, 1978.

[51] O. D. Faugeras and M. Hebert, "The representation, recognition, and locating of 3-D objects," *Int. J. Robotics Res.*, vol. 5, pp. 27–52, 1986.

[52] M. Froimowitz and S. Matthysse, "Geometrical correspondence between phenazocine and the enkephalins," *J. Med. Chem.*, vol. 29, pp. 573–578, 1986.

[53] M. W. Walker and L. Shao, "Estimating 3-D location parameters using dualnumber quaternions," *CVGIP: Image Understanding*, vol. 54, pp. 358–367, 1991.

[54] D. W. Eggert, A. Lorusso, and R. B. Fisher, "Estimating 3-D rigid body transformations: a comparison for four major algorithms," *Mach. Vision Appl.*, vol. 9, pp. 272–290, 1997.

[55] R. Sibson, "Studies in the robustness of multidimensional scaling: Perturbational analysis of classical scaling," *J. R. Statist. Soc. B*, vol. 41, pp. 217–229, 1979.

[56] S. P. Langron and A. J. Collins, "Perturbation theory for generalized Procrustes analysis," *J. R. Statist. Soc. B*, vol. 47, pp. 277–284, 1985.

[57] J. M. Fitzpatrick, J. West, and C. R. Maurer, Jr., "Predicting error in rigid-body, point-based registration," *IEEE Trans. Med. Imaging*, vol. 17, pp. 694–702, 1998.

[58] D. J. Hawkes, "The accuracy of 3D image registration using point landmarks," Tech. Rep. RS94/1, UMDS, London, 1995.

[59] P. J. Edwards, D. J. Hawkes, D. G. Hill, D. Jewell, R. Spink, A. Strong, and M. Gleeson, "Augmented reality in the stereo operating microscope for otolaryngology and neurosurgical guidance," *Computer Assisted Surgery*, vol. 1, pp. 172–178, 1995.

[60] P. H. Schönemann and R. M. Carroll, "Fitting one matrix to another under choice of a central dilation and a rigid motion," *Psychometrika*, vol. 35, pp. 245–255, 1970.

[61] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge, England: Cambridge University Press, 2nd ed., 1992.

[62] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: John Wiley & Sons, 1973.

[63] W. M. Newman and R. F. Sproull, *Principles of Interactive Computer Graphics, 2nd ed.* New York: McGraw-Hill, 1979.

[64] D. Kincaid and W. Cheney, *Numerical Analysis*. Pacific Grove, CA: Brooks/Cole, 1991.

[65] C. R. Maurer, Jr., R. P. Gaston, D. L. G. Hill, M. J. Gleeson, M. G. Taylor, M. R. Felon, P. J. Edwards, and D. J. Hawkes, "AcouStick: A tracked A-mode ultrasonography system for registration in image-guided surgery," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI) 1999* (C. J. Taylor and A. C. F. Colchester, eds.), pp. 953–962, Berlin: Springer-Verlag, 1999.

[66] S. Lavallee, J. Troccaz, P. Sautot, B. Mazier, P. Cinquin, P. Merloz, and J.-P. Chirossel, "Computer-assisted spinal surgery using anatomy-based registration," in *Computer-Integrated Surgery: Technology and Clinical Applications* (R. H. Taylor, S. Lavallee, G. Burdea, and R. Mösges, eds.), Cambridge, MA: MIT Press, 1996.

[67] C. R. Maurer, Jr., G. B. Aboutanos, B. M. Dawant, R. J. Maciunas, and J. M. Fitzpatrick, "Registration of 3-D images using weighted geometrical features," *IEEE Trans. Med. Imaging*, vol. 15, pp. 836–849, 1996.

[68] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, pp. 239–256, 1992.

[69] W. E. L. Grimson and T. Lozano-Perez, "Model-based recognition and localization from sparse range or tactile data," *Int. J. Robotics Res.*, vol. 3, pp. 3–35, 1984.

[70] R. Bajcsy and F. Solina, "Three-dimensional object representation revisited," *Proc. 1st Int. Conf. Comput. Vision*, pp. 231–240, 1987.

[71] E. Hall, J. Tio, C. McPherson, and F. Sadjadi, "Measuring curved surfaces for robot vision," *Comput.*, vol. 15, pp. 42–54, 1982.

[72] P. J. Besl and R. C. Jain, "Three dimensional object recognition," *ACM Comput. Surveys*, vol. 17, pp. 75–145, 1985.

[73] R. Nevatia and T. O. Binford, "Description and recognition of curved objects," *Artificial Intell.*, vol. 8, pp. 77–98, 1977.

[74] P. Brou, "Using the gaussian image to find the orientation of objects," *Int. J. Robotics Res.*, vol. 3, pp. 89–125, 1984.

[75] C. A. Pelizzari, G. T. Y. Chen, D. R. Spelbring, R. R. Weichselbaum, and C.-T. Chen, "Accurate three-dimensional registration of CT, PET, and/or MR images of the brain," *J. Comput. Assist. Tomogr.*, vol. 13, pp. 20–26, 1989.

[76] T. Zuk, S. Atkins, and K. Booth, "Approaches to registration using 3D surfaces," *Medical Imaging 1994: Image Processing*, vol. Proc. SPIE 2167, pp. 176–187, 1994.

[77] D. A. Simon and T. Kanade, "Geometric constraint analysis and synthesis: Methods for improving shape-based registration accuracy," in *CVRMed-MRCAS '97* (J. Troccaz, E. Grimson, and R. Mösges, eds.), pp. 181–190, Berlin: Springer-Verlag, 1997.

[78] C. R. Maurer, Jr., R. J. Maciunas, and J. M. Fitzpatrick, "Registration of head CT images to physical space using a weighted combination of points and surfaces," *IEEE Trans. Med. Imaging*, vol. 17, pp. 753–761, 1998.

[79] C. R. Maurer, Jr., G. B. Aboutanos, B. M. Dawant, S. Gadamsetty, R. A. Margolin, R. J. Maciunas, and J. M. Fitzpatrick, "Effect of geometrical distortion correction in MR on image registration accuracy," *J. Comput. Assist. Tomogr.*, vol. 20, pp. 666–679, 1996.

[80] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Comm. ACM*, vol. 18, pp. 509–517, 1975.

[81] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*. Berlin: Springer-Verlag, 1985.

[82] J. B. West, J. M. Fitzpatrick, M. Y. Wang, B. M. Dawant, C. R. Maurer, Jr., R. M. Kessler, R. J. Maciunas, C. Barillot, D. Lemoine, A. Collignon, F. Maes, P. Suetens, D. Vandermeulen, P. A. van den Elsen, S. Napel, T. S. Sumanaweera, B. Harkness, P. F. Hemler, D. L. G. Hill, D. J. Hawkes, C. Studholme, J. B. A. Maintz, M. A. Viergever, G. Malandain, X. Pennec, M. E. Noz, G. Q. Maguire, Jr., M. Pollack, C. A. Pelizzari, R. A. Robb, D. Hanson, and R. P. Woods, "Comparison and evaluation of retrospective intermodality brain image registration techniques," *J. Comput. Assist. Tomogr.*, vol. 21, pp. 554–566, 1997.

[83] G. Borgefors, "Distance transformations in arbitrary dimensions," *Comput. Vision Graph. Image Processing*, vol. 27, pp. 321–345, 1984.

[84] C. T. Huang and O. R. Mitchell, "A Euclidean distance transform using grayscale morphology decomposition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, pp. 443–448, 1994.

[85] H. Jiang, R. A. Robb, and K. S. Holton, "A new approach to 3-D registration of multimodality medical images by surface matching," *Visualization in Biomedical Computing 1992*, vol. Proc. SPIE 1808, pp. 196–213, 1992.

[86] A. Collignon, D. Vandermeulen, P. Suetens, and G. Marchal, "Registration of 3D multi-modality medical images using surfaces and point landmarks," *Pattern Recognit. Lett.*, vol. 15, pp. 461–467, 1994.

[87] P. A. Viola, *Alignment by maximization of mutual information*. Ph.D. thesis, Massachusetts Institute of Technology, 1995.

[88] J. V. Hajnal, N. Saeed, E. J. Soar, A. Oatridge, I. R. Young, and G. M. Bydder, "A registration and interpolation procedure for subvoxel matching of serially acquired MR images," *J. Comput. Assist. Tomogr.*, vol. 19, pp. 289–296, 1995.

[89] K. J. Friston, S. Williams, R. Howard, R. S. Frackowiak, and R. Turner, "Movement-related effects in fMRI time-series," *Magn. Reson. Med.*, vol. 35, pp. 346–355, 1996.

[90] H. Gudbjartsson and S. Patz, "The Rician distribution of noisy MRI data," *Magn. Reson. Med.*, vol. 36, pp. 910–914, 1995.

[91] R. P. Woods, S. R. Cherry, and J. C. Mazziotta, "Rapid automated algorithm for aligning and reslicing PET images," *J. Comput. Assist. Tomogr.*, vol. 16, pp. 620–633, 1992.

[92] P. A. Freeborough and N. C. Fox, "The boundary shift integral: an accurate and robust measure of cerebral volume changes from registered repeat MRI," *IEEE Trans. Med. Imaging*, vol. 16, pp. 623–629, 1997.

[93] R. P. Woods, J. C. Mazziotta, and S. R. Cherry, "MRI-PET registration with automated algorithm," *J. Comput. Assist. Tomogr.*, vol. 17, pp. 536–546, 1993.

[94] D. L. G. Hill and D. J. Hawkes, "Voxel similarity measures for automated image registration," *Visualization in Biomedical Computing 1994*, vol. Proc. SPIE 2359, pp. 205–216, 1994.

[95] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens, "Multimodality image registration by maximization of mutual information," *IEEE Trans. Med. Imaging*, vol. 16, pp. 187–198, 1997.

[96] W. M. Wells, III, P. Viola, H. Atsumi, S. Nakajima, and R. Kikinis, "Multi-modal volume registration by maximization of mutual information," *Med. Image Anal.*, vol. 1, pp. 35–51, 1996.

[97] C. Studholme, D. L. G. Hill, and D. J. Hawkes, "An overlap invariant entropy measure of 3D medical image alignment," *Pattern Recognition*, vol. 32, pp. 71–86, 1999.

[98] C. Studholme, D. L. G. Hill, and D. J. Hawkes, "Automated 3D registration of MR and CT images of the head," *Med. Image Anal.*, vol. 1, pp. 163–175, 1996.

[99] C. Studholme, D. L. G. Hill, and D. J. Hawkes, "Automated 3D registration of MR and PET brain images by multi-resolution optimisation of voxel similarity measures," *Med. Phys.*, vol. 24, pp. 25–35, 1997.

[100] J. Parker, R. V. Kenyon, and D. Troxel, "Comparision of interpolating methods for image resampling," *IEEE Trans. Med. Imaging*, vol. 2, pp. 31–39, 1983.

[101] J. Pluim, J. B. A. Maintz, and M. A. Viergever, "Mutual information matching and interpolation artifacts," *Medical Imaging 1999: Image Processing*, vol. Proc. SPIE 3661-01, pp. 56–65, 1999.

[102] D. L. G. Hill, C. R. Maurer, Jr., C. Studholme, J. M. Fitzpatrick, and D. J. Hawkes, "Correcting scaling errors in tomographic images using a nine degree of freedom registration algorithm," *J. Comput. Assist. Tomogr.*, vol. 22, pp. 317–323, 1998.

[103] L. Lemieux and G. J. Barker, "Measurement of small inter-scan fluctuations in voxel dimensions in magnetic resonance images using registration," *Med. Phys.*, vol. 25, pp. 1049–1054, 1998.

[104] W. F. Eddy, M. Fitzgerald, and D. C. Noll, "Improved image registration by using Fourier interpolation," *Magn. Reson. Med.*, vol. 36, pp. 923–931, 1996.

[105] G. Q. Maguire, Jr., M. E. Noz, E. M. Lee, and J. H. Schimpf, "Correlation methods for tomographic images using two and three dimensional techniques," in *Information Processing in Medical Imaging 1985* (S. L. Bacharach, ed.), pp. 266–279, Dordrecht: Martinus Nijhoff Publishers, 1986.

[106] D. L. Collins, C. J. Holmes, T. M. Peters, and A. C. Evans, "Automatic 3D model-based neuroanatomical segmentation," *Human Brain Mapping*, vol. 3, pp. 190–208, 1995.

[107] D. Rueckert, C. Hayes, C. Studholme, P. Summers, M. Leach, and D. J. Hawkes, "Non-rigid registration of breast MR images using mutual information," in *Med. Image Comput. and Computer-Assisted Intervention, 1998* (A. C. F. Colchester, S. L. Delp, and W. M. Wells, III, eds.), pp. 1144–1152, Berlin: Springer-Verlag, 1998.

[108] L. Lemieux, N. D. Kitchen, S. W. Hughes, and D. G. T. Thomas, "Voxel-based localization in frame-based and frameless stereotaxy and its accuracy," *Med. Phys.*, vol. 21, pp. 1301–1310, 1994.

[109] G. Penny, J. Weese, L. J. A., P. Desmedt, D. L. G. Hill, and D. J. Hawkes, "A comparison of similarity measures for use in 2D-3D medical image registration," *IEEE Trans. Med. Imaging*, vol. 17, pp. 586–595, 1998.

[110] P. Viola and W. Wells, "Alignment by maximization of mutual information," in *Proc. 5th Int. Conf. Computer Vision*, pp. 16–23, 1995.

[111] M. J. Clarkson, D. Rueckert, D. L. G. Hill, and D. J. Hawkes, "Registration of multiple video images to pre-operative CT for image guided surgery," *Medical Imaging 1999: Image Processing*, vol. Proc. SPIE 3661-01, pp. 14–23, 1999.

# Index

**N**



**O**

**Q**



**R**

HANDBOOK OF

# Medical Imaging

## Volume 2. Medical Image Processing and Analysis

**Milan Sonka**
**J. Michael Fitzpatrick**
*Editors*

During the last few decades of the twentieth century, partly in concert with the increasing availability of relatively inexpensive computational resources, medical imaging technology, which had for nearly eighty years been almost exclusively concerned with conventional film/screen x-ray imaging, experienced the development and commercialization of a plethora of new imaging technologies. Computed tomography, MRI, digital subtraction angiography, Doppler ultrasound imaging, and various imaging techniques based on nuclear emission (PET, SPECT, etc.) have all been valuable additions to the radiologist's arsenal of imaging tools toward ever more reliable detection and diagnosis of disease.

In addition to the concurrent development of rapid and relatively inexpensive computational resources, this era of rapid change owes much of its success to an improved understanding of the information theoretic principles on which the development and maturation of these new technologies is based. A further important corollary of these developments in medical imaging technology has been the relatively rapid development and deployment of methods for archiving and transmitting digital images. Much of this engineering development continues to make use of the ongoing revolution in rapid communications technology offered by increasing bandwidth.

*A little more than 100 years after the discovery of x rays, this three-volume Handbook of Medical Imaging is intended to provide a comprehensive overview of the theory and current practice of Medical Imaging as we enter the twenty-first century. As evidenced by the variety of the research described in these volumes, medical imaging is still undergoing very rapid change. Herein is the most current information available for students, researchers, and practitioners working in this exciting field.*

**Volume 1. Physics and Psychophysics**
**Volume 2. Medical Image Processing and Analysis**
**Volume 3. Display and PACS**

ISBN 978-0-8194-3622-1

9 780819 436221 90000

SPIE

P.O. Box 10
Bellingham, WA 98227-0010

ISBN-13: 9780819436221
SPIE Vol. No.: PM80/SC