ANALYTICAL CALCULATION OF PHOTON DISTRIBUTIONS IN SPECT PROJECTIONS

By

R. Glenn Wells

B.Sc.(Hon), The University of Calgary, 1991

M.Sc., The University of British Columbia, 1994

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE STUDIES

DEPARTMENT OF PHYSICS

We accept this thesis as conforming

to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

August 1997

Department of Physics
The University of British Columbia
6224 Agricultural Road
Vancouver, B.C., Canada
V6T 1Z1

Date: Aug 28 /97

Abstract

In this work is presented a method for analytically calculating the distribution of photons detected in SPECT projections. The technique is applicable to sources in homogeneous and non-homogeneous media. The photon distribution (primary, first and second order Compton scatter, and first order Rayleigh scatter) is computed using precalculated camera-dependent lookup tables in conjunction with an attenuation map of the scattering object and a map of the activity distribution.

The results of the technique are in excellent agreement with those of Monte Carlo simulation and experimental phantom studies. It has been validated with respect to sources in both homogeneous and non-homogeneous media.

Compared with a similar analytical technique, it offers a factor of 40–60 decrease in the calculation time for higher order Compton scatter distributions. For small sources, it improves on computation time required by Monte Carlo simulators by a factor of 20–150.

Finally, this method has been applied to the problem of correcting for cross-talk in ^{123}I–^{99m}Tc dual-isotope SPECT studies. It has demonstrated the ability to accurately reproduce the shape of the cross-talk distribution and to reproduce the absolute activity of the sources to within 7%, allowing accurate removal of cross-talk contamination.

Table of Contents

List of Figures

List of Tables

ACKNOWLEDGMENTS

I would like to thank my supervisor, Anna Celler for all of her help and guidance throughout this project. She has been consoling and sympathetic, encouraging and enthusiastic. She has pushed me when I needed pushing and argued with me when I was wrong. She has given me all of her support. She's been great!

My thanks also to all of the members of MIRG for making the workplace a pleasant place to be; I've enjoyed working with you all. My special thanks to R. Harrop for his encouragement, his insights, and his interest in the project (in spite of his "retirement"). I would like to thank C. Dykstra and S. Barney for their many helpful discussions and suggestions and Christi for reformatting the chapter pages.

My special thanks to R.R. Johnson whose interest and assistance made this work possible, to A. MacKay for allowing me the use of his computer, and to all of the members of my committee for their comments and time throughout this project.

I must also express my gratitude to all those who lent me their expertise: S. Vannoy for his assistance in the use of SIMSET, the Physics secretaries for all their assistance, especially in my struggles with the UBC administration, Ann and the other technologists at VHHSC for managing to fulfill my requests for weird source configurations and for helping me figure out the cameras, C. Nylander for making administrative dealings a pleasant experience, and all of the doctors at VHHSC for answering all my questions.

Lastly, I would like to thank my family and friends for their continual encouragement and support: N. Lamb for being there to talk to, C. Roehrig for his computing expertise, all of the members of *house.org* for stress relief, and C.J. Meyer for everything. My final thanks are for my parents who have never given me anything but hope, support, and encouragement; for them, my deepest love.

Chapter 1

Introduction

1.1 Aim of this work – Hypothesis

Single Photon Emission Computed Tomography (SPECT) is a diagnostic medical imaging technique based on the detection of gamma radiation emitted by tracers injected into the patient. Photon scattering within the patient is one of the effects which reduces the quality of the images produced by SPECT. Accurate scatter correction requires a precise identification of the scattered photons in the acquired data. Accurate scatter and attenuation correction will allow physicians to obtain quantitatively precise information about the activity distribution and consequently about the functionality of the patients' organs.

It is hypothesized that the technique herein presented can provide an accurate analytical method for calculating scatter (and cross-talk) in SPECT and that this technique can provide the means for accurate scatter and/or cross-talk correction. It is believed that quantitative information about body function will greatly increase the diagnostic capabilities of SPECT imaging; reducing hospital costs and improving patient care.

1.2 Thesis Highlights

- We have developed a new analytical method for calculating photon distributions in SPECT projections.
- The results produced by our technique demonstrate excellent agreement with those from both Monte Carlo simulation and phantom experiments.
- Advantages of the technique
  - The technique is based on a knowledge of the physics of photon interactions with matter; it is not phenomenologically based.
  - The technique uses precalculations of the photon transport kernel to greatly improve the computation time.
  - The technique accurately computes the distribution of the unscattered photons, the first and second order Compton scattered photons, and the first order Rayleigh scattered photons. The technique can be extended to compute higher order scatter distributions.
  - The technique is patient specific. It is based upon a transmission scan of the patient tissue density distribution and is applicable to non-homogeneous attenuation/scattering media such as the chest region.
  - The technique is camera specific. It is based upon the specific characteristics of the SPECT camera being used, the chosen collimator, and the selected diagnostic protocol.
- Important Applications
  - This technique can be used as an alternative to Monte Carlo simulation for studying SPECT data acquisition and correction techniques.

- The technique can be used to estimate scatter distributions and thereby allow for accurate scatter correction.
- The technique can be used to calculate and correct for cross-talk contamination in both simultaneous dual-isotope SPECT studies and also in transmission/emission SPECT scanning.

1.3 Structure of the Thesis

This thesis is composed of seven chapters. This first chapter contains some basic background information. It describes what SPECT is, how it works, and briefly indicates its usefulness in medical diagnosis. It also includes short discussions about the camera, about image reconstruction, and about the types of photon interactions with matter which occur at the energies relevant to SPECT imaging. The chapter concludes with a more detailed explanation of the motivation behind this work.

The primary reason for identifying scattered photons in the acquired data is to allow for appropriate scatter correction. A review of the current state of knowledge in the field of SPECT scatter correction is included in Chapter 2.

Chapter 3 contains a description of our method. The first three sections of this chapter detail our approach to the calculation, for point sources, of primary (unscattered), Compton scattered, and Rayleigh scattered photon distributions. The mathematical derivations and approximations that are used in order to improve the calculation time are provided. These sections are followed by a discussion of the extension of the technique to larger source distributions.

As suggested in [1], any scatter estimation and/or correction technique should be validated through comparison with the results of both Monte Carlo simulations and phantom experiments. In Chapter 4 the results of our calculations are compared with those of the Monte Carlo simulators EGS4 [2] and SIMSET [3]. The accuracy as well as the speed of our calculations are investigated for the cases of point and extended sources in both homogeneous and non-homogeneous scattering media. This is followed in Chapter 5 by verification of the accuracy of our technique when compared to phantom experiments. These experiments were performed with small ^{99m}Tc sources in both homogeneous and non-homogeneous media.

In Chapter 6 our technique is applied to a current problem in nuclear medicine, that of accurately separating the photon distributions acquired from each source distribution in a simultaneous dual-isotope SPECT scan. Dual-isotope scans involve the injection into the patient of two different tracers followed by a single SPECT scan. Simultaneous scanning is important as it provides a cost-effective means of acquiring automatically co-registered images of a single physiological parameter in two different states (*eg* stress/rest) or of two different physiological parameters (*eg* lung ventilation/perfusion). The dual-isotope protocol we chose was that for ^{99m}Tc–^{123}I brain studies. The energies of the photons emitted by these two isotopes are very close together. Consequently, the images acquired with this protocol suffer significant contamination from the presence of the second isotope. Our technique is used to accurately predict the relative activity levels for the two isotopes from experimentally acquired phantom data.

Finally, the work and results are summarized in Chapter 7. The main capabilities and limitations of the technique are reviewed and some of the future directions for this work are provided.

For reference, the computer codes, written in the 'C' language, which were used to implement our method are provided in Appendix A.

1.4 Nuclear Medicine

Modern medical imaging began more than one hundred years ago in 1895 when Wilhelm Roentgen discovered that "X-rays" could produce an image of the bones of his hand. Medical diagnosis now also uses a wide range of other techniques such as X-ray computed tomography (CT), ultrasound, magneto- and electro-encephalography (MEG and EEG), magnetic resonance imaging (MRI), planar imaging, positron emission tomography (PET), and single photon emission computed tomography (SPECT). All of these techniques are non-invasive, that is, without the need of a scalpel, they provide information about the anatomy and physiology of a patient, allowing for distinction between normal and diseased states.

Nuclear medicine covers the last three of the medical imaging technologies[1] mentioned above: planar imaging, PET, and SPECT. All of these techniques use radiotracers, drugs labelled with a radioactive isotope, which are injected into (or inhaled by or ingested by) the patient. The tracers are taken up by or otherwise become located in different systems or organs in the body so that their distribution provides information as to how well the targeted organ is functioning. They are used to study and detect some of the most common and important diseases in our society such as heart disease, cancer, and cerebrovascular disorders. For instance, stress-rest scans of the heart use a ^{99m}Tc labelled drug, sestamibi. Sestamibi is a lipophilic, positively charged complex that is taken up by the cells of the heart during their normal function (necrotic or dead cells do not take up any of this drug). The quantity of drug taken in by the cells is proportional to the amount of blood flow through the heart muscle. As the isotope attached to the drug decays, the emitted radiation is detected and it is possible to see where and in what

[1] There are also some therapeutic procedures in nuclear medicine such as, for example, the treatment of the thyroid gland with radioactive iodine.

quantity the drug is present in the heart. This type of study can indicate areas of the heart which are no longer functioning or no longer adequately supplied with blood. This is important because if undersupplied tissues are reperfused rapidly enough (such as by reopening clogged arteries), then they can be saved and heart attacks can be prevented.

1.5 SPECT

In this work, we have concentrated on modelling SPECT data acquisition. The data from a SPECT scan consists of a collection of projections, each of which is equivalent to a single planar image (Figure 1.1). A planar image is obtained when the detector is stationary at a single position. This image is like a chest X-ray, except that the source of the radiation is inside of the patient, in the form of a radioactive tracer. One of the problems with a single planar image (or any one projection) is its inability to distinguish the depth of the source that is creating the image. Because there is no depth distinction, the images from sources at different depths all pile up on top of one another creating an amalgamation like a multiply-exposed photograph and obscuring information about the source distribution and location.

SPECT overcomes this problem by acquiring many different views (projections) of the object which allow the determination of source depth. For each projection, the camera is positioned at a different angle (Figure 1.1). Just as one can mentally reconstruct a three-dimensional image of an object from a set of pictures showing all of its sides, so can the 3D distribution of the radiation be obtained from the set of different SPECT projections. With simple objects one can look at the set of projections and visualize the whole object, but to obtain a precise image, especially in more complicated situations, requires computerized reconstruction. The final product of SPECT is a 3D map of the patient source distribution which is normally examined by physicians as a set of 2D

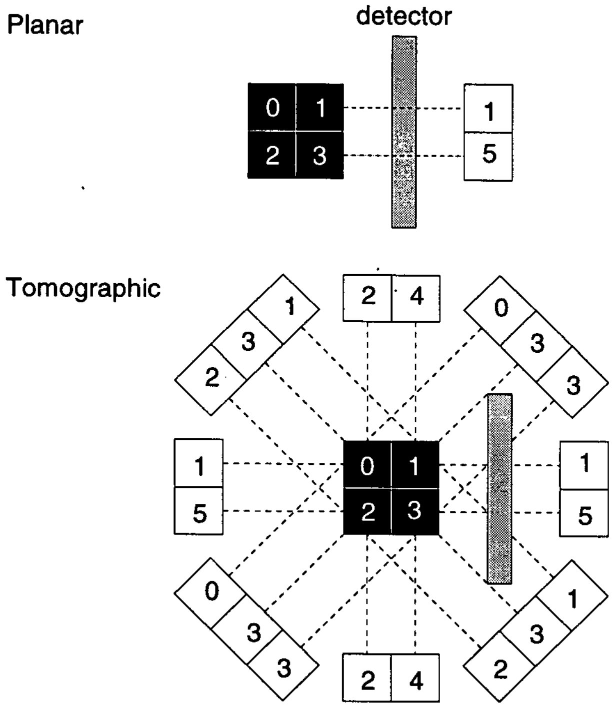

Figure 1.1: Planar versus SPECT imaging. Figures *Planar* and *Tomographic* respectively show a very simplified view of the acquisition of planar and SPECT data. The numbers in the black pixels represent the activity in the object pixels. The numbers in the white pixels correspond to the activity seen by the detector pixel. Because it acquires data from all angles around the patient, SPECT is capable of reconstructing the 3D distribution of the radiation sources whereas planar imaging is not. This figure is based on a similar figure in [4].

cross-sectional slices.

1.6 SPECT Applications

SPECT is a widely used clinical technique for diagnosis in all parts of the body (*ie* [5, 6]). SPECT provides enhanced image contrast over planar imaging and hence allows for a more accurate diagnosis. It is routinely used, for example, to find areas of ischemia and infarction in the heart muscle, to look for areas of epilepsy and stroke in the brain, and to detect tumor locations throughout the body.

SPECT is inherently a very sensitive technique and is a valuable diagnostic tool, but it only provides qualitative images. Because of the degrading influence of, amongst other things, attenuation and scatter, it is not currently possible to provide quantitatively accurate images in a clinical setting. Quantitative SPECT would provide more accurate information, regarding issues such as degree of tumor malignancy and myocardial viability, and would thus improve the specificity of the technique, avoiding erroneous diagnosis and hence unnecessary therapy or treatment. This, in turn, would provide better patient care at a lower cost. Quantitative SPECT would open the door to the development of new and more precise diagnostic procedures that are not possible with the current quality of SPECT images. As stated in 1982 by J.W. Keyes, one of the pioneering scientists in SPECT research [7]:

> The ability to visualize radiopharmaceutical distributions in the body in three dimensions, the ability to quantify these dimensional relationships, and finally the ability to extract true quantitative values noninvasively from structures deep within the body should provide significant improvements in the way we practice nuclear medicine.

A hurdle which must be overcome before quantitative SPECT can become a reality is the development of a clinically applicable method of accurate scatter correction. To understand some of the difficulties associated with scatter correction in SPECT, it is useful to be familiar with the SPECT camera and with the relevant physics of the interactions of photons with matter.

1.7 The SPECT Camera

The detector normally used to acquire modern SPECT images is the Anger gamma camera. The idea for this camera was first proposed by H. Anger in 1958 [8]. It consisted of a scintillation crystal attached to several photomultiplier tubes which were in turn connected, via electronic circuitry, to an oscilloscope screen. The main advantage of this camera is that it allows simultaneous imaging of its entire field of view. Though originally described with a pin-hole collimator (a lead sheet with a single small hole through it), the modern version of this camera normally uses a multichannel collimator as suggested in [9]. The camera has also been improved through increasing the number of photomultiplier tubes (PMT) over the original seven (a modern camera head contains 90 or more PMTs), increasing the size of the detector crystal (camera crystals are now more than 40x50cm^2), and replacing the oscilloscope with a computer. A schematic of the Anger camera is shown in Figure 1.2.

The gamma ray or photon is emitted from the radioactive source somewhere inside of the patient. To be detected, this photon must first pass out of the patient, then through the collimator, and finally interact with the detector crystal to produce a shower of lower energy light photons. These light photons ionize electrons from the photocathode at one end of the PMT. The electrons are accelerated through a series of potential differences along the length of the PMT with more electrons being released at each step. This "snowball" effect produces a much amplified signal at the other end of the PMT. The PMT signal is read in by electronic circuits which convert it into a position and an energy for the original detected photon. The position/energy information is then stored in the computer.

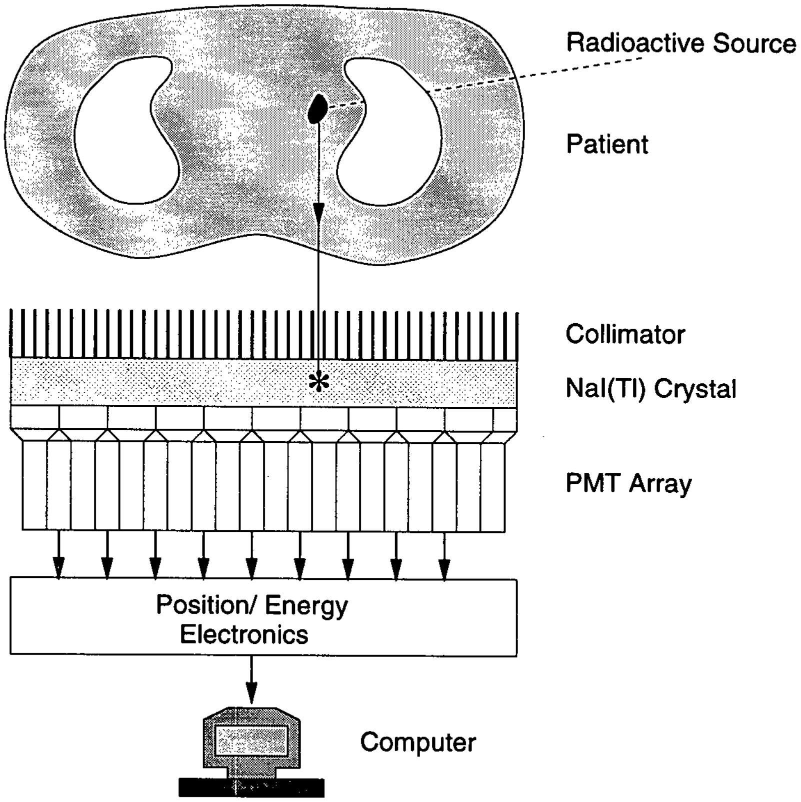

Figure 1.2: The Anger Camera. This is a schematic diagram showing the basic components of a typical SPECT camera system. These components are the collimator, the (scintillation) detector crystal, positioning electronics, and a computer for data recording and storage. The camera head rotates around the patient to acquire the different views needed for SPECT image reconstruction.

1.7.1 The Collimator

The collimator is very important in SPECT because it restricts the direction of the photons incident on the detector crystal. Detection by the scintillation crystal gives information about the position of the photon at the time of detection, but no indication of the direction from which it came. The collimator provides that information by acting like a sieve and restricting the photons which pass through it to those travelling in certain directions. The rest of the photons are absorbed by the collimator and never allowed to reach the detector crystal. The collimator is essentially a large lead sheet with many holes through it.

There are many different types of collimators, one of the more common of which is the parallel hole collimator. It is this collimator which we have chosen to model throughout the work in this thesis. With this collimator, the holes are aligned parallel to one another and also parallel to the normal of the collimator (and consequently to the normal of the detector crystal). Although an ideal collimator only accepts photons which are travelling straight through the holes, because the holes have finite width and length, any photon which is travelling within a small angle (the acceptance angle) of the straight path will also pass through the collimator. Additionally, photons which are travelling at angles somewhat larger than the acceptance angle also have a small chance of getting through to the detector. They do this by actually penetrating through the lead walls of the holes (the septa). This effect is called septal penetration.

An unfortunate feature of collimators is that they block most of the photons, typically letting through only on the order of one in every ten thousand photons[2]. This greatly reduces the efficiency of the camera and consequently increases the noise in the data. Because it is a medical technology, the strength of source activity is limited by the level of dose the patient can receive and the scan time is limited by the time one can reasonably expect a patient to remain virtually motionless. As a result, SPECT is always working in a realm of statistically poor information.

[2] This low efficiency is caused simply by the geometry of the collimator. Consider the simplified case of a small source located in air (no attenuation) 10cm away from the back of a parallel hole collimator. Assume that the collimator has circular holes and an acceptance angle of 2.44°. The radiation emitted from the source is spread uniformly over a sphere of area $400\pi\text{cm}^2$. However, photons must pass through the collimator hole to be detected and this subtends an area of approximately $0.045\pi\text{cm}^2$. Additionally, the holes only cover about 80% of the collimator surface. This means that only 1 in every 11000 photons emitted from the source will reach the NaI detector.

1.7.2 The Scintillation Crystal

Once a photon has passed through the collimator, it is detected using a scintillation crystal. Scintillation crystals absorb high energy radiation and re-emit this radiation in the form of a cascade of lower energy (light) photons. The scintillation crystal used in SPECT is thallium-activated sodium iodide, NaI(Tl) or more simply just NaI. Pure sodium iodide crystal is a poor scintillator at room temperatures but becomes a very good one if small amounts of impurities such as thallium (0.1–0.4 mole percent) are added. One reason NaI crystal is chosen is that it has a very high light output. This is important because, in the statistically poor domain of SPECT imaging, every detected photon is important. Another reason is that the amount of light emitted by the crystal is proportional to the energy of incident gamma ray, making the energy easier to determine. The NaI crystal is also relatively inexpensive to make, keeping camera costs down, and can be grown to very large sizes. SPECT detectors must be made from a single crystal and the size to which NaI crystals can be grown has allowed modern detectors to be more than 50x40cm^2 in area. Finally, the NaI crystal is transparent at the wavelengths of the scintillation light it produces, causing very little light to be lost through self-absorption, even in thick crystals. The thickness of the crystals used in SPECT is usually 3/8" or 1/2". The thickness is a trade off between spatial resolution and stopping power. A thicker crystal has less chance that a photon will pass through undetected but also has poorer spatial resolution. The crystal thickness is usually optimized for 140keV photons because this is the energy of the primary radiation emitted by ^{99m}Tc , one of the more commonly used isotopes in SPECT. A 3/8" thick crystal results in an efficiency of 70–75% for ^{99m}Tc photons [10] and an intrinsic spatial resolution for the camera of about 3.8mm.

NaI crystals also have disadvantages. They are quite fragile and, once cracked, the crystal is useless. Another draw-back of NaI is that it has poor energy resolution, making it difficult to reduce scatter by means of energy discrimination (section 2.2.1). NaI is also hydroscopic; it will react with the water vapor in the air reducing the transparency of the crystal and consequently its efficiency. However, the advantages of NaI outweigh its disadvantages and make it the current detector of choice in SPECT imaging.

1.7.3 Electronic Circuitry

Behind the scintillation crystal are attached many photomultiplier tubes (PMTs), usually between 50 and 100 in modern SPECT cameras. These tubes translate the light emitted by the scintillating crystal into an electronic signal that can be reliably detected and recorded by the camera electronics. The tubes are arranged in a hexagonal array which allows for a tight packing, improving the uniformity of the crystal coverage. The position of the site of interaction between the incoming photon and the detector crystal is determined by techniques such as Anger logic (Figure 1.3). The (X,Y) position of the detected event is determined as given below.

$$\begin{aligned} \mathrm{X} &= \kappa(\mathrm{X}^{+} - \mathrm{X}^{-})/\mathrm{Z} \qquad (1.1) \\ \mathrm{Y} &= \kappa(\mathrm{Y}^{+} - \mathrm{Y}^{-})/\mathrm{Z} \end{aligned}$$

where Z is the total light signal of the event and κ is a scaling factor to convert the output signal into a physical position on an oscilloscope screen or into a coordinate length. The values for $\mathrm{X}^{\pm}$ and $\mathrm{Y}^{\pm}$ are as shown in Figure 1.3.

The energy of the photon is related to Z and is determined using a multichannel pulse height analyser.

A problem associated with all forms of radiation detection is deadtime. Deadtime is related to the time required by the detector and associated electronics to process and record the detection of a photon. If a second photon enters the detector before it has

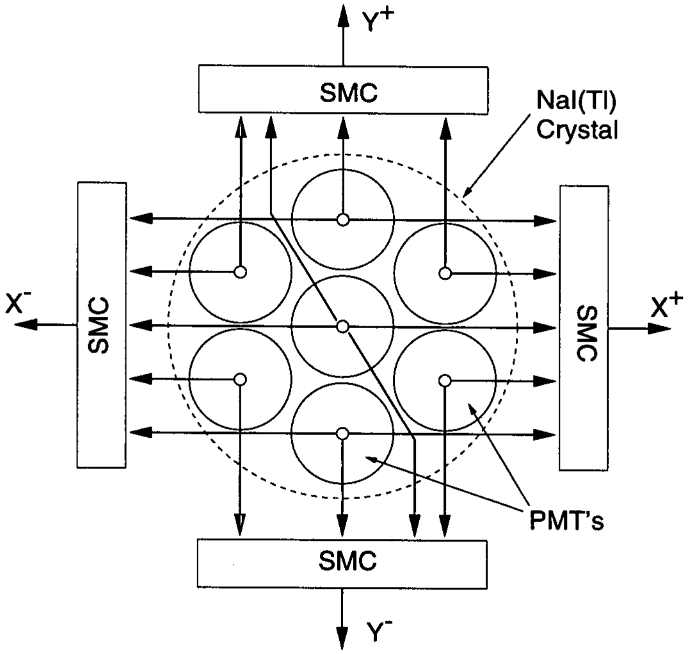

Figure 1.3: Anger Position Logic. This is one of the standard techniques used to determine the position of a detected photon interaction with the crystal. This figure shows a simple schematic of an Anger camera. A modern camera has between 50 and 100 photomultiplier tubes (PMT) and not just seven as is depicted here. The arrowed lines indicate which PMTs are connected to which summing matrix circuit (SMC). When a photon strikes the crystal and produces a shower of light photons, these are picked up and amplified by the PMTs. Each PMT receives a portion of the light shower the size of which is related to the proximity of the PMT to the position of the event. The SMCs add together the outputs from the PMTs such that the output signals ($X^{\pm}$, $Y^{\pm}$) are proportional to the distance of the incident photon from the centre of the detector crystal. By comparing the output from the four SMCs, one can determine the position where the gamma ray photon interacted with the crystal (equation (1.1)). This figure is based on a similar figure in [10].

finished processing the first, the signals from the two photons can interfere with each other resulting in one or both of the photons not being recorded. For example, when a photon is stopped by the scintillation crystal, it produces a shower of light photons and this shower has a certain duration. If a second photon strikes the crystal before the shower from the first photon is finished, then the two showers may overlap (pulse pileup) and the detector can see this as a single event. Because the energy of the radiation is related to the number of light photons in the shower it creates, if two showers overlap, the detector may disregard both events because their sum falls outside the acceptable energy range. Another example is the multichannel analyser which may not accept a second input until it has finished processing the first. In this case the second detected event would just be ignored. Although each component of the system has its own particular deadtime, normally a single deadtime is given for the entire detector system. For NaI detector systems, the deadtime is typically on the order of microseconds. It should also be remarked that as the activity of sources becomes greater, the average time between detection events decreases and the effect of deadtime on the number of detected photons becomes more significant.

1.7.4 Computer

Once the position and energy of the incident gamma ray has been determined, the data is digitized, recorded, tabulated, and stored by the computer. With planar images or single SPECT projections, the image is stored in an array which can be from 64x64 bins to 1024x1024 bins. Each bin or pixel corresponds to a small square area on the surface of the camera head. For example, on the Siemens MS3 triple head camera, each of the pixels in the 1024x1024 array is 0.45x0.45mm^2. For SPECT projections, the array size is normally 64x64 or 128x128 bins and the number of projections acquired is typically 64 or 128. The size of the array used in SPECT and the number of projections acquired is limited by the storage capabilities of the computer, the limited number of detected photons, and the time required to reconstruct the image.

The computer is also used to control the acquisition, automatically moving the camera heads to the appropriate positions at the appropriate times as per the commands given to it by the technologist performing the study. The computer also performs image reconstructions, data analysis and manipulation (according to the proprietary software that comes with the camera), and redisplays both the projections and the reconstructed images for the physician.

1.8 SPECT Reconstruction

Once the set of projections has been acquired, it is necessary to reconstruct the data into a useful format for viewing by the physicians. The standard approach to doing this is to reconstruct the 3D object as a stacked set of 2D slices or tomograms [3]. There are three conventional orientations for the slices: transaxial, coronal, and sagittal. The transaxial slice is in a plane parallel to the waist of the patient, the coronal slice is parallel to face of the patient, and the sagittal plane is parallel to the side of the patient [4] Normally, transaxial slices are reconstructed and stacked to form a 3D image (array) from which the other slices (coronal and sagittal) are extracted.

The technique which is used clinically for reconstruction is filtered back-projection

[3] Tomography comes from the Greek words *tomos* meaning "cut or slice" and *graphein* meaning "to draw".

[4] Because the heart is not aligned along any of the body's major axes, the slices used to view the heart are oriented with respect to the organ itself, not the patient's body. The typical heart slices are the short axis slice (cutting through the heart perpendicular to the long axis which extends from the base of the heart to the apex), the vertical long axis slice (cutting parallel to the long axis from the anterior to the posterior side of the heart), and the horizontal long axis slice (cutting the heart, parallel to the long axis, from the septa separating the two ventricles to the lateral wall).

(FBP), an idea adapted for nuclear medicine in 1971 [11]. The basic idea behind back-projection is that the information (number of photons) recorded in each pixel of the projection is spread back along the direction from which the photons came. Because the distance of the actual source from the detector pixel is unknown, the number of photons recorded in the pixel is added to every possible source location. Applying this procedure to all of the projection pixels generates lines of potential source positions and where these lines intersect, they reinforce each other indicating the true source locations (Figure 1.4). However, the density of the intersecting lines falls off inversely with distance from the point of intersection. Consequently, each reconstructed source point has a spike- or star-like appearance (Figure 1.4C). To correct for this a ramp filter is applied to the projection before reconstruction[5] which has the effect of decreasing counts near the centre of the reconstructed source.

Filtered back-projection is also frequently discussed in terms of the spatial frequency domain. In the frequency domain, the convolution operation (required to apply the ramp filter to the data) becomes a much simpler multiplication operation. Additionally, many other filters, such as those described below, are also much easier to apply in this domain. A spatial frequency domain image (projection) is obtained by taking the Fourier transform of the spatial domain image (projection). The low spatial frequencies of a Fourier transformed image correspond to the large smooth components of an object while high spatial frequencies describe small, spiky, or sharp features. To operate in the Fourier domain, one makes use of the projection slice theorem which states that

> the Fourier transform of the projection of an object at angle, θ, $(P(k,\theta))$ is equal to a slice of the (2D) Fourier transform of the object along angle, θ [12].

[5] Filtering can also be done after reconstruction using, for example, a 2D ramp filter on the reconstructed slice.

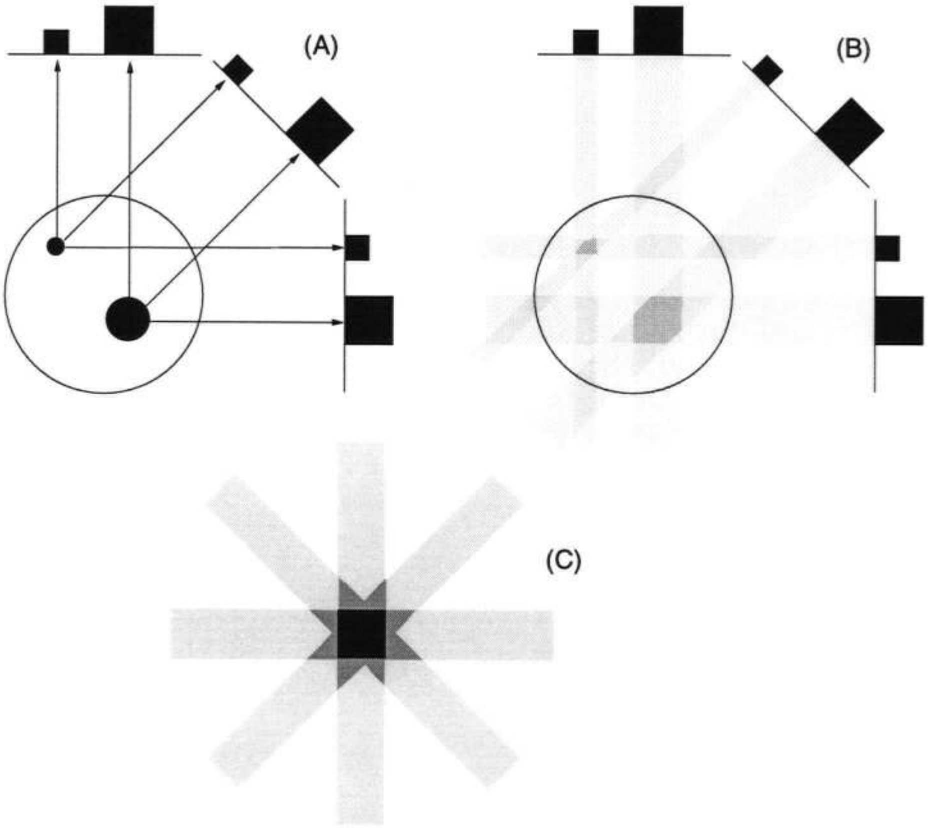

Figure 1.4: Back-Projection. Back-projection is the technique which is clinically used to reconstruct 3D images from the 2D projection data. Figure (A) shows the acquisition of data, figure (B) shows the basic back projection of this data (without application of the ramp filter). Figure (C) shows the classic star shaped pattern that one gets without the ramp filter. Application of the ramp filter "removes" this effect.

If the spatial frequency domain has been sufficiently sampled, the actual image, $f(x, y)$, can be recovered simply by taking the inverse Fourier transform (equation 1.2).

$$f(x,y) = \int_0^\pi \int_{-\infty}^\infty |k| P(k,\theta) \exp(2\pi i k(x\cos(\theta) + y\sin(\theta))) \mathrm{d}k\mathrm{d}\theta \tag{1.2}$$

Projection data samples the spatial frequency domain with a density proportional to the inverse of the spatial frequency. This is analogous to the density of the intersecting back-projected lines in the spatial domain. The changing sampling density is corrected for by the application of the ramp filter which appears in this form of filtered back-projection as the factor $|k|$ and which linearly scales the integrand by the absolute value of the spatial frequency.

The advantage of FBP is its speed, an issue of critical importance for clinical implementation, but there are some disadvantages to using FBP.

The first disadvantage is that FBP tends to amplify noise in the image. Statistical noise is always a factor in clinical SPECT data because of the limitations on patient dose and acquisition times, and the low efficiency of the camera system. Although white noise appears uniformly at all spatial frequencies, at high frequencies the signal to noise ratio in the image is much lower and the noise tends to dominate. Because the ramp filter scales data linearly with frequency amplitude, this low signal to noise region of the Fourier transformed image gets amplified, increasing the noise in the reconstructed image.

The most common approach used to reduce the high frequency noise is to modify the ramp filter. One choice is to apply a rectangular window to the ramp filter, effectively setting it to zero above a chosen cut-off frequency. The solution is not ideal, however, as the sharp cut-off caused by the rectangular filter causes overshoot or "ringing" [13, 14]. The ringing can be reduced by rolling off the frequency cut-off such as is done by the Hanning, Hamming, and Butterworth filters [12, 14]. A more fundamental problem associated with removing high frequency components is that doing so blurs the edges of image features and can "smooth away" small objects (like small or early tumors).

A second problem with FBP is aliasing [13, 15]. Because the data acquired is a discrete sampling of a continuous function, there are frequency components of the Fourier transformed image (all those above half of the Nyquist frequency) which will not be adequately sampled. Undersampling of these frequencies results in the appearance of false data components at lower frequencies (aliasing). Aliasing can be reduced by physically filtering out high frequency components of the image (blurring) before sampling or by using a finer sampling of the data [15]. Note, however, that increasing the sampling also increases image noise as fewer photons will be recorded in each sample (pixel).

A last problem with FBP is the uniqueness of the image. It is known that mathematically, the true image, $f(x, y)$, is uniquely determined only if the data is perfect (*ie* no noise) and there are an infinite number of projections [13, 16]. This is never the case in SPECT as there is always noise and only a finite number of projections are acquired. Consequently, FBP is prone to artifacts, features in the image which are created solely by the reconstruction process and do not correspond to true features in the scanned object.

Alternatives to FBP are the iterative reconstruction methods. One appealing feature of these techniques is that they permit inclusion of photon interaction (*eg* scatter) models and features such as positivity of the image (one cannot have negative activity) directly into the reconstruction procedure itself. The common iterative methods which have been considered in nuclear medicine are ART (algebraic reconstruction technique) and variants upon it [17, 18], iterative least squares [19], conjugant gradient [20], and expectation maximization using maximum likelihood (MLEM) [21] or ordered subsets (OSEM) [22]. Although these techniques can produce a better image [23] they are not used in clinical applications because they are typically too slow; often hours are required for the technique to converge on the best solution. Additionally, there is often not a good way of determining the end-point of the reconstruction; the iteration after which the technique is fitting more to noise than to information.

1.9 Photon Interactions with Matter

The isotopes used in SPECT imaging decay through the emission of gamma ray photons. The gamma ray energies are specific to the isotope. For example ^{99m}Tc emits photons with an energy of 140keV. The range of photon energies for the isotopes typically used in SPECT is 80–400keV[6]. As these relatively low energy photons travel away from their source inside of the patient, they can interact with patient tissues causing image degradations such as a reduction in contrast and image blurring. In SPECT imaging, photon interactions occur primarily by one of three different modes: the photoelectric effect, Rayleigh (coherent) scattering, and Compton (incoherent) scattering. A brief description of these modes as they pertain to medical physics is provided in the following sections and shown schematically in Figure 1.5. A more complete description can be found in, for example, [24].

1.9.1 Photoelectric Absorption

Photoelectric absorption occurs when a collision between a photon and an atom results in the complete absorption of the photon and in the ionization of an atomic electron. The probability that this type of interaction will occur (the cross-section) is proportional to Z^3 where Z is the atomic number of the atom. The interaction probability also decreases with increasing photon energy. Most biological materials have relatively low effective Z values (Table 1.1) and consequently, at the energies used in SPECT, the contribution of

[6] Energies outside this range are occasionally used in SPECT imaging, such as the 511keV positron annihilation photons from PET tracers like FDG, but this is much less common and often restricted to research applications.

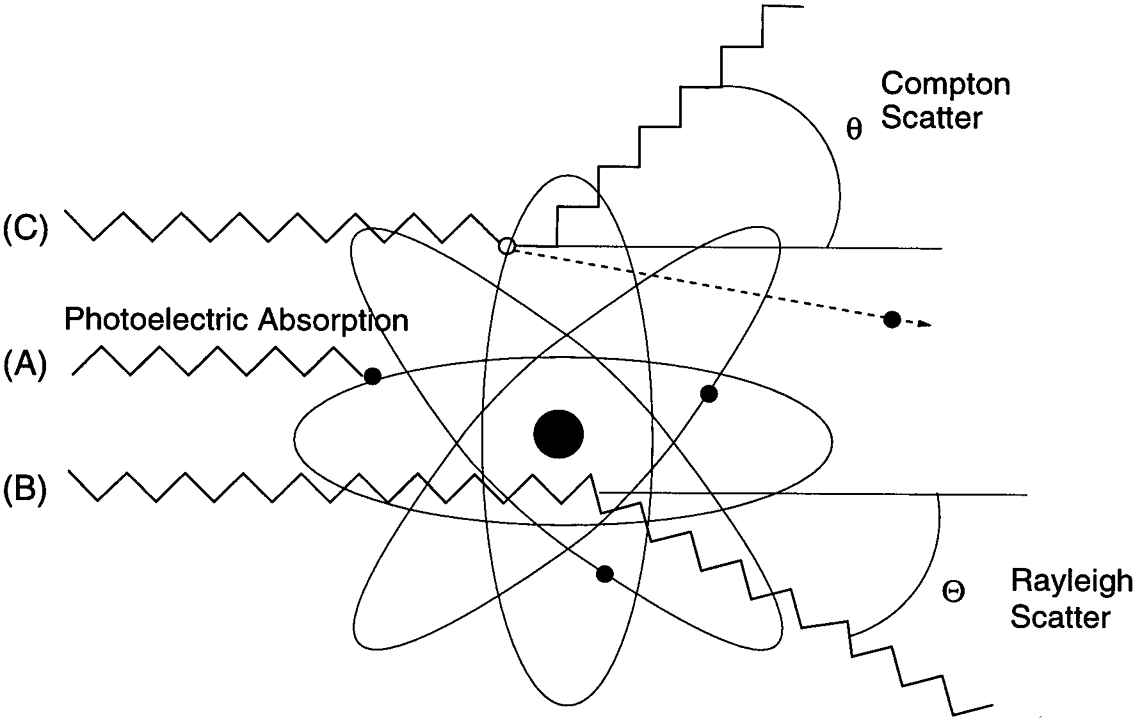

Figure 1.5: Schematic representation of the three most common types of photon interaction with patient tissues in SPECT imaging. Paths (A), (B), and (C) represent, respectively, photoelectric absorption, Rayleigh scattering, and Compton scattering.

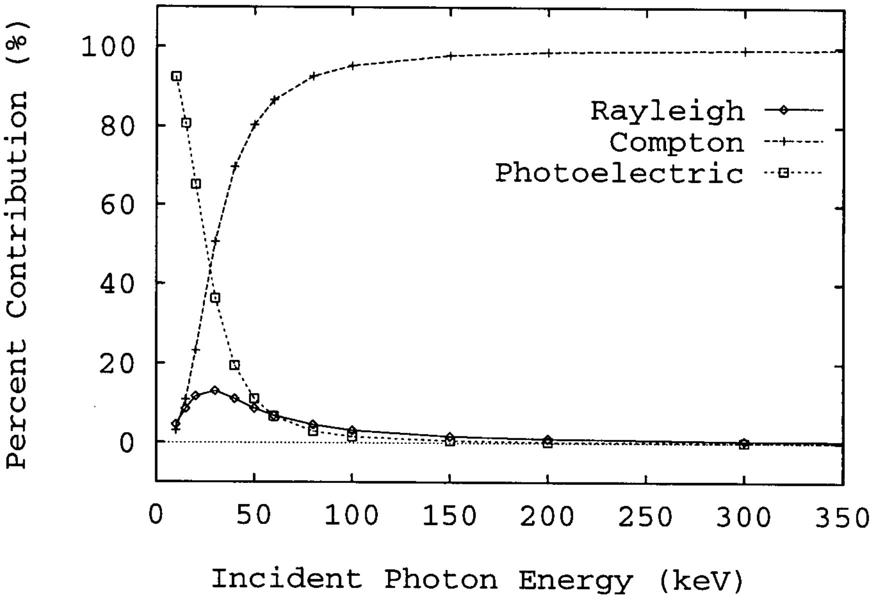

Figure 1.6: Relative importance of the three principal modes of photon interaction over the energy range relative to SPECT imaging.

photoelectric absorption to the total cross-section is small (1.9% in water at 140keV). Its relative importance as a function of energy is shown in Figure 1.6.

Table 1.1: Example effective atomic numbers (Z_{eff}) for some biologically relevant materials. The coefficient for lead, a common absorber in medical physics, is also included.

Material	Z_{eff}	Material	Z_{eff}
Muscle	7.64	Lung	7.55
Bone	12.31	Fat	6.46
Water	7.51	Lead	82.00

1.9.2 Rayleigh Scattering

Rayleigh scattering, also known as coherent scattering, occurs when the electromagnetic field of the incident gamma ray interacts with the electrons surrounding the nucleus. The gamma ray starts the electrons oscillating and they in turn retransmit the gamma ray. The direction of the retransmitted ray is changed but its energy is not. The Rayleigh scatter contribution to the total cross-section is small (Figure 1.6) in SPECT imaging, and hence often ignored. For example, in water at 140keV, the Rayleigh scattering contributes 0.7% to the total cross-section. Nevertheless, because its differential cross-section is strongly peaked at low scatter angles and because it does not lose any energy through the interaction, the contribution of Rayleigh scatter to the data acquired in the photopeak energy window is not negligible and it should be included for accurate calculations of the photon distribution. The Rayleigh scatter contribution is of the same magnitude as the second order Compton scatter contribution which is about 12% of the detected scattered photons (section 3.3).

The formula describing the differential cross-section, $\mathrm{d}\sigma_{coh}(\theta_R)/\mathrm{d}\Omega$, for Rayleigh scattering is given in equation (1.3).

$$\frac{\mathrm{d}\sigma_{coh}}{\mathrm{d}\Omega}(\theta_R) = \frac{r_o^2}{2}(1+\cos^2\theta_\mathrm{R})\,\mathcal{F}(x,\overline{Z}) \tag{1.3}$$

where

Ω is the solid angle of the scattering cone (there is rotational symmetry in the scattering cross-section about the direction of the incident photon),

r_o is the classical electron radius,

θ_R is the Rayleigh scattering angle,

$\mathcal{F}(x,\overline{Z})$ is the coherent scattering atomic form factor,

x is a parameter, sometimes called the momentum transfer of the scattering photon (*eg* [25]), which is given in equation (1.4), and

$\overline{Z}$ is the (effective) atomic number of the scattering material.

The momentum transfer of an incident photon is given by

$$x = \frac{\sin(\frac{\theta_R}{2})}{\lambda} \tag{1.4}$$

where λ is the wavelength of the scattering photon.

Although the energy of a coherently scattered photon does not change, its momentum does. Because momentum is conserved, the atom off which the photon scatters acquires a recoil momentum, x. The coherent scattering form factor, $\mathcal{F}(x, \overline{Z})$, gives the probability that the Z electrons of the atom acquire momentum x without any energy absorption [26]. The Rayleigh scattering form factors are tabulated, for example, in [25, 26, 27]. Coherent-incoherent interference [28] can significantly alter the angular dependence of the Rayleigh form factors in materials and at energies of interest to medical imaging. Interference should, therefore, be considered when computing material form factors from the tabulated atomic form factors.

1.9.3 Compton Scattering

Compton or incoherent scattering is the most important type of interaction that a medical imaging photon can undergo. This type of scatter accounts for 97.4% of the total cross-section for 140keV photons in water. In this interaction, the photon scatters off one of the electrons in the material and transfers some of its energy to this electron. Both the energy and the direction of the photon are changed.

The cross-section for this type of scatter was derived mathematically by Klein and Nishina and experimentally verified by A.Compton in 1923 [29]. The differential Klein-Nishina cross-section for Compton scattering, $\mathrm{d}\sigma(\theta)/\mathrm{d}\Omega$, is given by

$$\frac{\mathrm{d}\sigma(\theta)}{\mathrm{d}\Omega} = \frac{r_o^2}{2}(1+\cos^2\theta)\left(\frac{1}{1+\alpha(1-\cos\theta)}\right)^2\left(1+\frac{\alpha^2(1-\cos\theta)^2}{(1+\alpha[1-\cos\theta])(1+\cos^2\theta)}\right) \tag{1.5}$$

where

Ω is the solid angle of the scattering cone (there is rotational symmetry in the scattering cross-section about the direction of the incident photon),

θ is the Compton scattering angle,

r_o is the classical electron radius, and

α is the energy of the photon divided by the rest mass energy of the electron.

The Klein-Nishina formula assumes that the electrons are stationary and free, that is, not bound electromagnetically to the nucleus of the atom. This is not the case for most electrons in materials and, therefore, to account for the motion and bounded electron state, this formula is corrected by the incoherent scattering function: $\mathcal{S}(x, Z)$ where θ is the Compton scattering angle and x is the momentum transfer (equation (1.4)). The incoherent scattering function can be calculated using quantum mechanics and indicates the probability that an electron, which acquired energy from a scattering photon, will ionize or escape from the atom. This function is applied as a multiplicative correction to the differential cross-section. The functions, $\mathcal{S}(x, Z)$, are tabulated in [25, 26, 27]. They are not significantly influenced by coherent-incoherent interference effects [28].

The energy lost by a photon which Compton scatters can be derived by assuming conservation of momentum [29]. The equation for the energy of a scattered photon, E', is

$$E' = E\left(\frac{1}{1+\alpha(1-\cos(\theta))}\right) \tag{1.6}$$

where

E is the initial energy of the photon,

θ is the Compton scattering angle, and

α is the energy of the photon divided by the rest mass energy of the electron.

1.10 Motivation for Analytical Scatter Calculations in SPECT

Accurate scatter correction is one of the major problems facing quantitative SPECT. The typical energy resolution of a NaI(Tl) SPECT camera is only about 10% at 140keV and thus many scattered photons contaminate the acquired data. These photons misrepresent the position and strength of the radioactive source and hence degrade the quality of the reconstructed image by reducing contrast. Although, as will be discussed in Chapter 2, a number of different methods have been proposed to correct for scatter, none of these techniques are entirely satisfactory.

To properly correct for scattered photons, it is necessary to be able to accurately estimate their contribution to the acquired data. To have a clinically useful correction method requires that the estimate be obtained very quickly. The goal of this work is to develop an analytical technique for calculating the photon distribution in SPECT projections. This technique is applicable to arbitrary source distributions in both homogeneous and non-homogeneous scattering media and provides quantitatively accurate results. Additionally, it offers a significant improvement in calculation time over other equally accurate techniques.

One can perform scatter correction with this technique in two ways: either by subtracting the estimate of the scattered photon distribution from the acquired projections, or by incorporating the technique into the projector-reprojector pair of an iterative reconstruction scheme. We envision the inclusion of this technique in a method for generating quantitatively accurate SPECT images as is shown in Figure 1.7.

In this scheme, a transmission scan is used to generate a map of the patient tissue density distribution (an attenuation map). This attenuation map is then used to create an attenuation corrected estimate of the source distribution from the emission scan (SPECT)

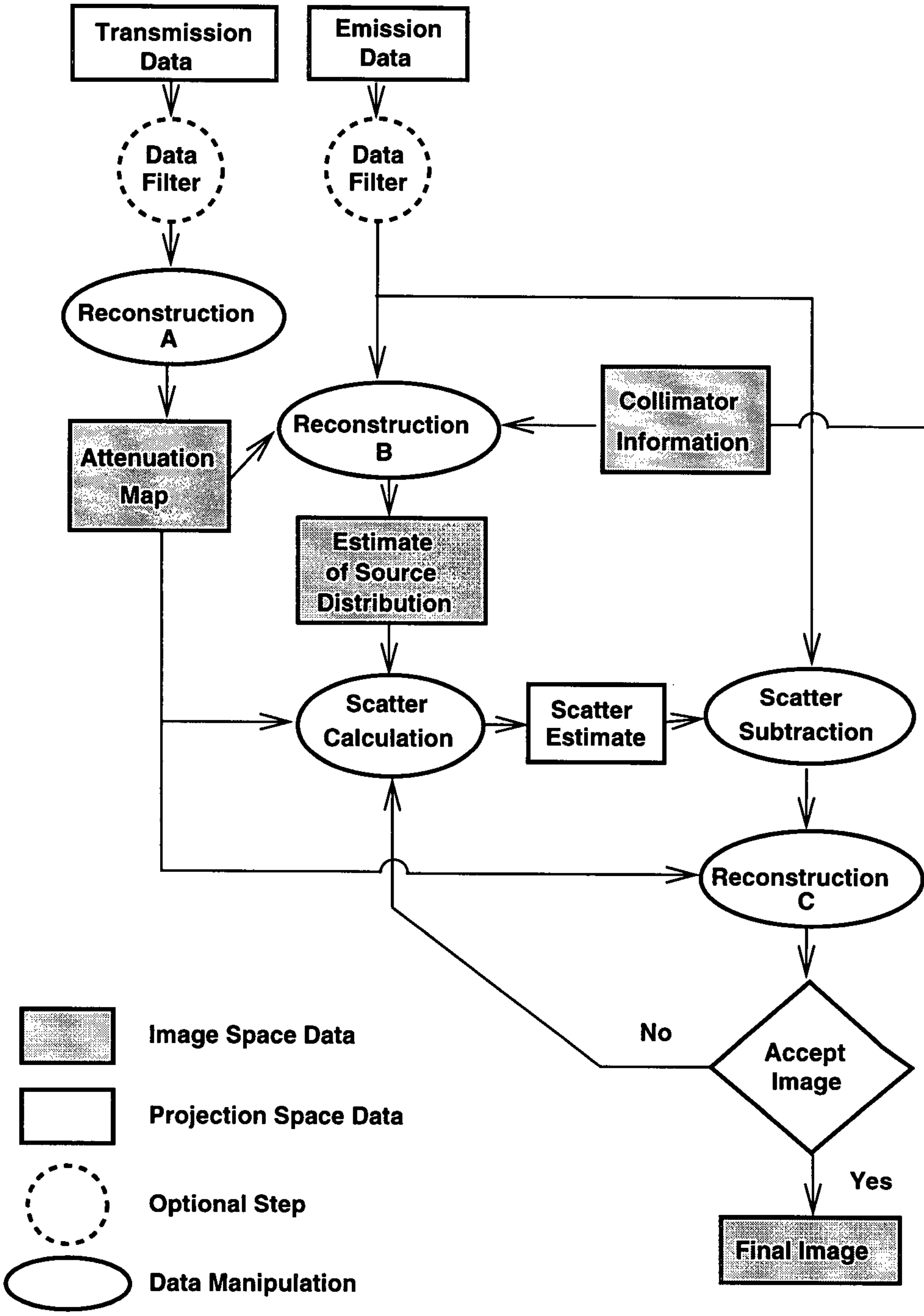

Figure 1.7: A quantitative SPECT image reconstruction scheme. This flowchart indicates the place where our approach to scatter estimation/correction could be included in a quantitative SPECT reconstruction method. The reconstructions at A, B, and C may be either FBP or an iterative technique such as MLEM.

data. The source distribution estimate and the attenuation map are used with our technique to calculate the scatter within the emission data. The scatter distribution is subtracted out of the emission data and a scatter corrected image is reconstructed. The reconstructed image can be used, if necessary, as an improved estimate of the source distribution and the scatter correction can proceed iteratively.

Chapter 2

Literature Review

Scatter is an important problem in SPECT. The presence of scattered photons in the acquired data blurs the image (reducing contrast) and can cause cold regions in a warm background to have an apparent activity of as high as 30% of the background activity [30]. Many different methods have been proposed to correct for scatter and a good review of these methods can be found in Buvat *et al* [1]. The problem of scatter correction is still, nevertheless, a current one because none of the methods developed to date have been both fast enough for routine clinical use and accurate enough (in complex scattering media) for quantitative correction.

For all scatter correction schemes, one of the central issues is the estimation of the distribution of scattered photons either in the projections or in the reconstructed image. Indeed, much of the work in this field has concentrated solely on the problem of determining the scatter distribution. For this reason, the body of this chapter has been divided into four sections based on the different types of information used to estimate scatter. The sections are: holistic approaches to scatter correction, estimation based on the energy spectrum of the acquired data, estimation based on phenomenological observations, and finally theoretical calculation of the scattered photon distributions.

2.1 Holistic Approaches to Scatter Estimation/Correction

The holistic approaches to scatter correction do not correct for scatter *per se* but rather try to attempt to compensate for the effects of scatter. Three examples of this are the use of a modified attenuation coefficient, multiplicative scaling of image, and the use of modified reconstruction filters.

2.1.1 Modified Attenuation Coefficient

This technique compensates for scatter by adjusting the attenuation coefficient used in attenuation correction schemes such as that suggested by Chang [31]. An attenuation coefficient is used to describe the amount by which the intensity of a photon beam passing through materials is reduced and incorporates all of the possible types of photon interaction. In the narrow beam definition of attenuation, it is assumed that any photon which interacts will not be detected. However, in reality, many of the photons which scatter are in fact still detected. As suggested in [32], detected scatter events can be compensated for by reducing the attenuation coefficient used in the correction method. For example, the value of the attenuation coefficient, μ, for water at ^{99m}Tc energy levels (140keV) is changed from $\mu = 0.15$ to $\mu = 0.12$ [33].

One difficulty with this technique is that the amount by which the attenuation coefficient needs to be adjusted is case dependent [34, 35]. Another problem is that the spatial distribution of the scattered photons is treated as being the same as that of the unscattered photons whereas, in truth, it is quite different. Consequently, while this approach can qualitatively improve the appearance (uniformity) of the image [33], it does not produce quantitatively reliable images [30, 36].

Another way of representing this approach is not to modify the attenuation coefficient but instead, to incorporate a build-up factor into the attenuation factor. The build-up factor is a scaling factor which accounts for those photons which have scattered in the medium (and hence are attenuated) but which have done so in such a way as to still be detected. The build-up factor depends on both the composition of the medium and on the location of the source within it. Attempts have then been made to use a spatially dependent build-up factor which is measured either experimentally [35, 37, 38] or with Monte Carlo (MC) techniques [39]. This can lead to improved quantitation of the reconstructed image [35, 38 – 40] but measuring patient dependent build-up factors is neither practical nor feasible in a clinical setting.

2.1.2 Multiplicative Scaling

A second compensation technique is to scale the image by a multiplicative constant based on the scatter fraction for the data [33, 41]. The scatter fraction is the ratio of scattered to unscattered photons detected by the camera. This fraction can be determined either experimentally [42] or through Monte Carlo simulation. The use of this global scaling factor, though correct on average, will over-correct some parts of the image while under-correcting others. Additionally, it does not correct for the mis-positioning of scattered photons. For example, radiation detected as coming from cold spots in the image is entirely due to scattered radiation and will not be removed through rescaling by the global scatter fraction.

2.1.3 Reconstruction Filters

During the FBP reconstruction process, it is possible to employ filters that do a single correction not only for high frequency noise but also for other image degrading effects such as the blurring caused by the imperfect intrinsic spatial resolution of the detector. These reconstruction filters are based on, for example, the modulation transfer function of the system. These filters also contain parameters which can be adjusted to compensate for the effects of scatter. Two filters which have been used in this way are the Weiner [43] and Metz [41, 44] filters. Although such filters can improve the qualitative appearance of an image through such techniques as contrast enhancement and noise suppression, the quantitative effect of these filters on the data is complex and difficult to interpret.

2.2 Scatter correction based on Energy Information

The second general category of scatter correction techniques attempts to use information about the energy or energy spectrum of the detected photons to distinguish scattered events from unscattered ones. The majority of detected scattered photons have Compton scattered and consequently have lower energy than unscattered photons. The NaI detectors are capable of determining the photon energy and so this information can be used. A general problem with all methods based on energy information is that they cannot correct for photons which have Rayleigh scattered and lost no energy.

2.2.1 The Photopeak Energy Window

One of the simplest and most direct ways of removing scatter from the data is to use energy discrimination. Unfortunately, the energy resolution of a NaI scintillation camera is not perfect; typically it is about 10% FWHM at 140keV. This means that the energy, as detected by the camera, for unscattered photons from ^{99m}Tc is distributed about 140keV with a normal distribution having a FWHM of 14keV. A similar distribution of energies occurs for scattered photons as well and consequently, complete energy discrimination is not possible.

Nevertheless, application of an energy window does reduce the number of accepted scattered events [45, 46] and so improves the image quality and its use takes very little time. Consequently, this technique is often used in clinical practice and almost always, when scatter correction is discussed, it is the correction of scatter in the photopeak window which is being referred to[1].

The standard choice of energy window is a 20% symmetric region centred on the energy of the unscattered photon. For ^{99m}Tc , this corresponds to the window from 126keV to 154keV (Figure 2.1) and encompasses 98% of the energy range for unscattered photons for a camera with 10% energy resolution.

Energy discrimination is more effective with higher resolution detectors, such as

[1] An example where scatter correction outside of the photopeak window might be desirable is a dual-isotope study (Chapter 6) where one is interested in removing the scattered photons of one isotope from a second energy window (the photopeak window of the second isotope).

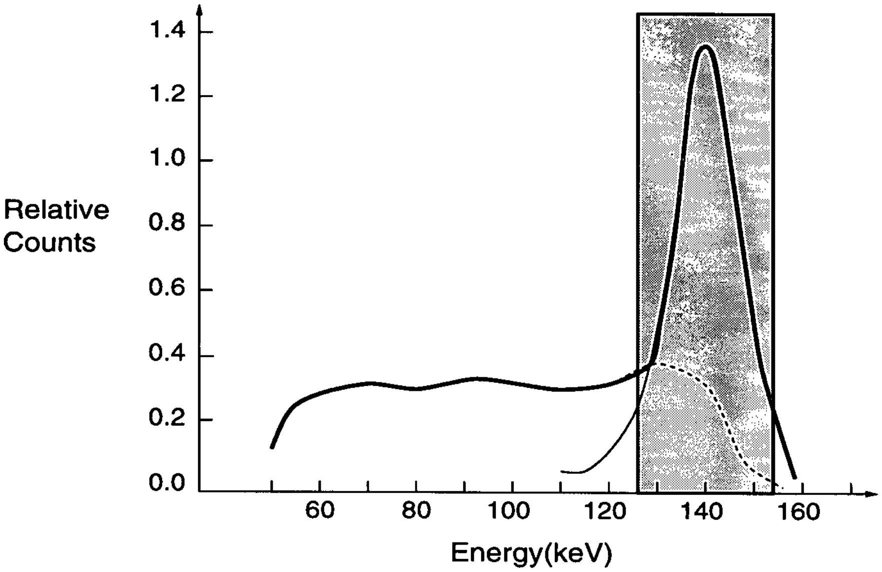

Figure 2.1: Standard ^{99m}Tc Energy Window. The energy window is shown as the shaded region. The thick solid line is the distribution of detected photons from a ^{99m}Tc source in a scattering medium. The thin solid line corresponds to unscattered photons and the dashed line to the total scatter distribution of photons. The figure is based on data given in [45].

lithium doped germanium, Ge(Li), or high-purity germanium, HPGe, detectors which have energy resolutions of 0.4%–0.7% [10, 47, 48]. Unfortunately, these detectors are very expensive to build in the sizes required for SPECT imaging and may require complicated maintenance such as liquid nitrogen cooling. For these reasons, they are not used commercially.

A modification to the standard photopeak window is to use an offset energy window. This was first considered in planar imaging [49 – 52], and later in SPECT [53, 54]. Because scattered photons have a lower energy than unscattered ones, shifting the photopeak window to higher energies biases detection in favor of unscattered photons and reduces the percentage of scattered photons accepted by the detector. However, offsetting the photopeak window also reduces the number of unscattered photons accepted by the camera which is detrimental to image quality.

2.2.2 Dual Energy Window (DEW)

One of the most popular scatter correction methods is the dual energy window technique first proposed for planar imaging [55] and later for SPECT by Jaszczak [30]. In addition to the standard photopeak window, this technique uses data acquired in a second, lower energy window (for example a window extending from 91–125keV [56]). A distribution, consisting essentially of only scattered photons, is collected in this window[2], scaled by a constant factor k, and subtracted from the distribution in the photopeak energy window. The subtraction is done either pre-reconstruction (on the projections themselves such as in [57]–[59]) or post-reconstruction (on the images obtained independently from the two energy windows such as in [30]).

Although popular and easy to implement, the DEW method has a few fundamental

[2] There are non-scatter events in this window but they comprise only a small fraction of the total events, on the order of 1.3% [56]

problems. Firstly, it assumes that the spatial distribution of scattered photons in the lower energy window is the same as that of the scattered photons in the photopeak window. This is incorrect because the lower energy window photons have typically scattered through angles much larger than those in the photopeak window and also have a higher probability of having undergone multiple scatters. The discrepancy in the scatter spatial distributions from the two energy windows has been shown in [56, 60 – 63].

Secondly, the scaling factor, k, is often assumed to be constant throughout the image and to have a value of 0.5, the value used in [30] (for example [62, 64]). It has been shown, though, that the best value for k depends on the energy of the emitted photons, the object geometry, the width and location of the scatter energy window, and the type of attenuating material [48, 56, 60, 62, 64 – 66]. For example, Koral [66] finds that the best values for his study are between 1.2 and 1.3, rather than 0.5. Different approaches for determining accurate values of k have been tried such as analytical approximation [55], Monte Carlo simulation [30, 62, 64, 65], or actual experimental measurements [30, 36, 48, 66 – 70]. However, obtaining an accurate value of k is difficult and consequently, quantitative correction with this technique is not practical.

Some effort has also been made to correct for the difference in the spatial distributions of the scatter and photopeak window data. The approach taken by Meikle *et al* [63] was to correct for differences in the spatial distribution of scatter by using a convolution function to convert the spatial distribution of scatter in the lower energy window into that of the higher energy window. The convolution function is determined empirically using water phantoms and is assumed to be spatially invariant.

2.2.3 Other Techniques Using Two Energy Windows

Some other techniques which also use two energy windows are the asymmetric photopeak energy windows method, the dual photopeak window method, and the channel ratio technique.

The asymmetric photopeak windows technique [71] splits the standard photopeak window into two asymmetric windows. The energy range of the two windows is chosen such that the number of scattered counts in each window is the same. The window positions were determined using homogeneous water phantoms and found to be fairly independent of source depth within the phantom. As unscattered photons are distributed more symmetrically in the photopeak window than are scattered photons, subtracting one of the asymmetric windows from the other results in scatter free data. The problems with this approach are that the subtraction also reduces the number of unscattered photons (by 20% in [71]) and that the technique is quite sensitive to the position of the line dividing the two windows and hence quite susceptible to electronic drift in the camera. The method also assumes invariance of the window positions with the source distribution and scattering medium.

The dual photopeak method [60, 72 – 74] divides the photopeak into two equal windows and assumes that there exists a relationship between the scatter-to-primary ratio, SF(i,j), and the ratio of the total number of counts in the two windows, R_S(i,j), for each pixel i,j. The scatter estimate, ES(i,j) is then given by

$$\mathrm{ES(i,j)} = \frac{\mathrm{SF(R_S(i,j))}}{\mathrm{SF(R_S(i,j))+1}}\mathrm{TC(i,j)} \tag{2.1}$$

where TC(i,j) corresponds to the total number of counts in the pixel i,j.

The ratio R_S(i,j) is determined from measurements in a homogeneous phantom. It was found that the ratios varied significantly with depth, camera position, and between different camera heads when the combined energy range of the two windows spanned just the photopeak [72]. When, instead of splitting the photopeak, this idea was applied to the Compton and photopeak windows, more stable ratios were obtained and additionally, the amount of noise in the signal was reduced. The use of the Compton window, though, did not yield as much improvement in image contrast. The ratios also varied with position due to the changing efficiencies of the photomultiplier array across the camera head (although this can be corrected for by a calibration measurement) and it is likely that they will be dependent on the type of attenuating medium, making it difficult to use this method for correcting non-homogeneous scatter. Noise due to low statistics in the images can be a problem and so it is suggested [72] that the estimated scatter image be low-pass filtered before subtraction from the original image. As with the asymmetric window technique, this method is sensitive to the position of the energy windows and consequently prone to errors due to electronic drift in the camera.

A last method is the channel ratio method [75] which, like the asymmetric and dual photopeak methods, divides the photopeak into upper and lower energy windows. This approach assumes that the ratio of the unscattered photons in the two windows is a constant (k_1) and that there is a similar ratio for the scattered photons in the two windows (k_2). Knowing the total number of photons (scattered plus unscattered) detected in each window then gives four equations with four unknowns and one can solve for the number of unscattered photons.

However, a constant k_2 assumes, incorrectly, that the shape of the scatter spectrum does not vary from pixel to pixel. This method is also particularly sensitive to poor counting statistics in the upper energy window and can result in negative values for the number of unscattered photons in a pixel [75].

2.2.4 Triple Energy Window

A variation on the DEW method is the triple energy window (TEW) method [76] which uses two small (≈2keV wide) energy windows bracketing a photopeak window. The scatter in the primary window is estimated by linear interpolation between the two side windows. As the number of scattered photons in the photopeak does not vary linearly with energy, the accuracy of the interpolation will depend on the location and width of the two side windows. Changes in the scatter spectrum due to inhomogeneities in the patient and non-uniformities in the detector response may also alter the optimal side window configuration from pixel to pixel. Additionally, the thin width of the two side windows reduces the number of counts they receive, making the TEW approach prone to statistical fluctuations.

This technique is modified through the addition of a fourth (Compton) window at a lower energy [77]. The difference between the two side windows is compared to this fourth window and, through comparison with a Monte Carlo generated database, the photopeak scatter is determined. The relationship between the Monte Carlo generated data and the experimentally acquired ratio is, however, object dependent and so, accurate scatter estimation could require a larger database than is practical to store. The use of the fourth window also does not remove the problem of statistical noise in the two side windows.

2.2.5 Neural Networks

It has also been suggested that a neural network could be used to estimate scatter [78]. With this technique, the photopeak window is divided into five sub-windows. The number of photons recorded in each of these windows is fed into a neural network. The network is trained on Monte Carlo simulations where the truth is known. Although the results of this are promising, it is still necessary to determine if the network is capable of dealing accurately with the wide variety of possible imaging situations seen in a clinical setting.

2.2.6 Energy Weighted Acquisition

The scatter estimation employed by techniques such as the DEW method can be generalized as suggested by Beck [79]. He proposed that the weight given to each photon should not be just 1 or 0 (the standard energy window) or possibly -0.5 (DEW) depending on its energy, but rather generalized to any number, positive or negative, depending on parameters such as the energy, the detected position, and the relative abundance of photons at that energy. This idea was adopted and incorporated into an energy weighted acquisition approach [80]–[86] which weights photons based on their energy and also repositions them. This technique uses a symmetrically arrayed energy-dependent weight matrix which is applied to each detected photon. The matrix depends on system characteristics such as the isotope and collimator used [84] and can be determined either by Monte Carlo simulation or by experiments [81].

As the same matrix is applied to all points in the projection, the method does not take into account the geometry of or inhomogeneities in the scattering medium. Also, the scatter tails of the point spread function (PSF) of a detected photon often extend beyond the radial range of the suggested weight matrix (consisting of 21 pixel sized elements) and so this technique cannot completely correct for the detected scatter. To apply energy-weighted acquisition also requires additional hardware such as the WAM (weighted acquisition module) produced by Siemens Electric Limited. This module preprocesses the data using a weighting function which is digitized in 1keV increments. Finally, the weighting functions which are used to generate the weight matrices depend on the isotope and collimator used and may vary with different imaging geometries [80].

2.2.7 Local Energy Spectrum Analysis

An alternate approach to the use of the energy information is to decompose the energy spectra observed in each pixel into its scattered and unscattered components. The first technique proposed for doing this was peak erosion [87] but it was abandoned due to the dependence of the results on the number of iterations used in the erosion. The second technique is to fit the observed data to a model of the energy spectra [88]–[92].

The difficulties with these types of approaches are that some of them require a knowledge of the unscattered photon energy spectrum in each pixel which is not easy to acquire [89]. Additionally, there is the possibility that the models chosen to represent the scatter spectrum will be overly simplified [1]. The necessity of collecting information in multiple energy windows may also present a problem for many cameras in clinical use. Finally, the optimal choices of sampled energy range and scatter spectrum model can vary from one pixel to the next [93].

2.2.8 Global Energy Spectrum Analysis

A final approach for using the energy information from the detected photons is similar to the methods of the previous section (section 2.2.7). However, instead of treating the energy spectrum of each pixel independently, it uses the correlations between neighboring pixels; it examines the energy spectrum of the image in its entirety using multi-dimensional analysis techniques. There are two different methods which use this approach: holospectral imaging and constrained factor analysis.

Holospectral imaging [94 – 98] represents each pixel of each projection as an energy space vector. Principal component analysis is used to find the two primary axes of the space, corresponding to the unscattered photons and the Compton scattered photons. Those photons which lie further away from the unscattered photon axis than can be accounted for by statistical variations are designated as Compton scattered photons and are eliminated.

A drawback of this technique is that it requires data acquisition in 10–16 energy windows and all of these windows must be unbiased (corrected for different efficiencies and for uniformity at different energies). A method of correcting camera bias is described in [95], but this technique also slightly reduces spatial resolution.

Constrained factor analysis [99] is based on factor analysis [100, 101] of the energy spectra of the data. The energy spectra are represented in a factor space whose bases are found from the decomposition of the covariance matrix for the data. Constrained factor analysis uses a modified version of FAMIS (Factor Analysis of Medical Image Sequences) and restricts the factor space to two dimensions. Similar to holospectral imaging, eigenvectors of this space can be found which correspond to scattered and unscattered photons.

The restriction of the space to two dimensions (one for primary photons and one for scatter) results in a stationary scatter spectrum and thus these techniques have associated with them the problems of a spatially invariant scatter estimate [1, 99]. An alternate method called "target apex seeking" [102] has also been proposed which uses a prior estimate of the scatter free spectrum and searches the factor space for the spectrum most similar to the expected one. This results in a spatially variant estimate of the scatter spectrum as it is no longer restricted to one dimension [1]. The advantage of this approach is that the unscattered and scattered spectra do not need to be known *a priori* but are determined from the experimental data and thus will adapt for different camera features and object geometries. However, this technique still requires that the energy response of the camera is spatially uniform which is not necessarily the case [72].

With all of the global energy spectrum analysis techniques, there is the problem of high statistical noise in the data acquired in each energy window. To mitigate the problems of noisy data, most of these methods suggest first grouping together the data into bins comprised of several pixels. This reduces noise but introduces different errors due to the summing together of energy spectra which have different shapes.

2.3 Phenomenologically Based Reprojectors

One of the difficulties associated with using energy information to estimate the scatter is that the techniques often require data to be acquired simultaneously in several energy windows which may not be possible with some, especially older, SPECT cameras. Also energy based methods cannot correct for Rayleigh scatter. An alternate class of methods uses extrapolation of empirical observations of scatter from test cases to estimate scatter distributions. These techniques typically convolve an estimate of the source distribution with either a model of the point spread function (PSF) which can be decomposed into scattered and unscattered components or a direct model of the scatter PSF. This approach is usually called convolution subtraction (CS) as the scatter estimate is normally just subtracted from the actual data. The PSF model can also be used in projector-reprojector iterative methods as in [103, 104]. One can also attempt to use the information in the scattered photons by correctly repositioning them by deconvolving the PSF with the original data such as in [105]–[107]. These empirical methods differ in how they estimate the PSF.

2.3.1 Spatially invariant PSF

Experimental data shows that the wings of a PSF, which correspond to scattered photons, have a linear slope in a semi-log plot. For this reason, a popular function used to describe scatter from a point source is the mono-exponential [62, 65, 105, 108 – 112]. It can be determined by fitting a mono-exponential to the wings of experimental data from line (or point) sources [109]. Other, somewhat more complicated functions have also been suggested [63, 113 – 115]. The underlying assumption of the existence of a stationary convolution function is, however, incorrect [116, 117]. The shape and size of the scatter convolution function is dependent on the source position and the object geometry.

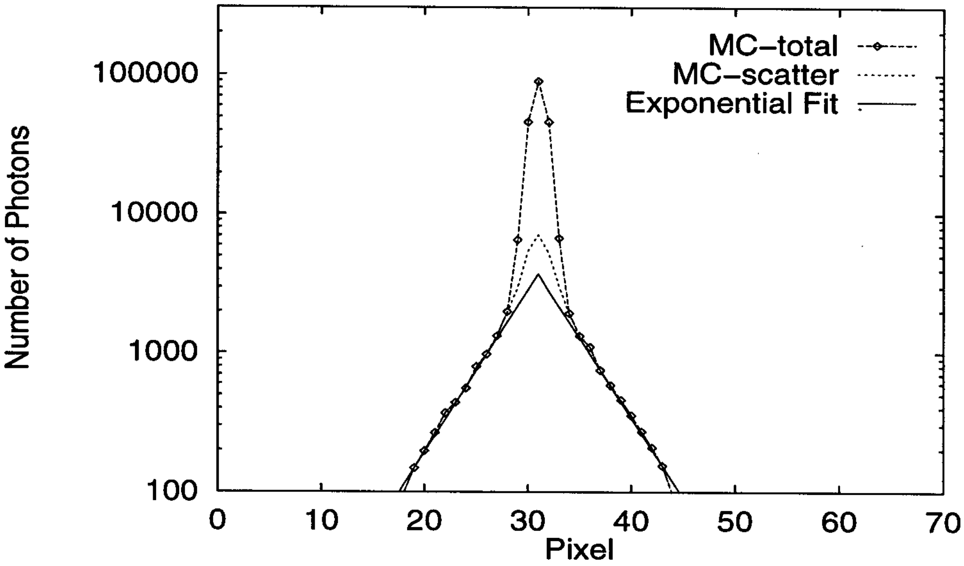

Figure 2.2: Convolution Subtraction Method. This method estimates the scatter in an experimentally acquired PSF by fitting exponential functions to the scatter wings of the PSF. This approach does not accurately model the shape of the scatter distribution and tends to underestimate the number of counts in the peak. Data is based on the Monte Carlo simulation of a point source on-axis in a 10cm radius water cylinder.

An additional problem with the choice of an exponential function is that it does not accurately model the scatter in the central high count region of the distribution. This is shown in Figure 2.2 which is based on data from a Monte Carlo simulation of a point source on-axis in a 10cm radius water cylinder.

Modifications to the PSF model have incorporated some corrections for the position of the source within the patient. For example, the iterative convolution subtraction (ICS) technique [63, 113], uses a convolution kernel whose scaling is based on scatter fractions determined using depth dependent build-up functions [35]. The method is, however, still based upon a mono-exponential spatially invariant kernel.

A different modification uses the weighted sum of convolutions in several different energy windows to determine the scatter [64]. This approach combines the use of energy information into convolution subtraction. While it takes into account some of the spatial variation of the convolution function with changing energy, it still does not account for variations due to source position or geometry.

A final approach of this type uses a convolution function which is dependent on the source-collimator distance [97] but this method incorrectly assumes that the convolution function is radially symmetric and invariant to translation in the plane parallel to the collimator surface.

2.3.2 Slab Derived Scatter Estimation

Perhaps the most promising of the PSF based scatter estimators is slab derived scatter estimation (SDSE) developed in [103, 118 – 122] and concurrently in [104, 123, 124]. This approach is based upon a database of experimentally acquired, camera specific, line spread functions (LSFs) which is used to generate object dependent LSFs (and PSFs) as necessary. The database is generated using measurements of a line source in a homogeneous water phantom at different depths in the phantom and at different distances from the collimator surface. These measurements can be performed quickly using a triangular phantom [125]. The estimated depth of the source within the object, the geometrical shape of the object, and the distance of the object from the collimator are then used to generate an object specific LSF (or PSF). Although initially developed for homogeneous media which are invariant in the axial (z-) direction, the technique has since been extended generate 3D PSFs [104].

A drawback of this approach is that, because the technique is based upon the empirical measurements in homogeneous material, it is hard to generalize to non-homogeneous materials. Effort has been made to use water equivalent depths to extend this technique to non-homogeneous media [121] but the accuracy of this approach is limited. There remains a 10% error in the estimated scatter.

2.4 Theoretical Calculations

An alternative to the experimentally based scatter estimation methods is to use a theoretically based calculation of the scattered photon distribution. The physics behind the interactions of photons with matter (both in the patient and in the NaI camera) are well known and formulae exist which describe these phenomena. It is possible, therefore, to take an estimate of the source distribution and accurately calculate the scatter distribution which corresponds to it. The difficulty with this type of approach is that realistic modelling of photon propagation from the point of emission to the point of detection results in a complex equation that does not have a simple analytical solution. One approach, at this stage, is to resort to approximations which do permit an analytical solution. Another approach is to reduce the complexity of the equation through application of physically reasonable approximations and then solve it in one of two ways: either by Monte Carlo or numerical evaluation techniques.

2.4.1 Monte Carlo

Monte Carlo simulators stochastically propagate photons through a system and track what happens to them. The probabilities of a photon travelling a certain distance, of it interacting with the attenuating/scattering medium, and of it being detected can all be determined. At each step along the way, "dice are rolled" (random numbers are generated by the computer) and the fate of that individual photon is determined. Does it get absorbed? Does it scatter? If so, in which direction does it now travel? Does it reach the detector, does it pass through the collimator, is it detected with an energy within the energy window and so on? This technique derives its name from the fact that all of these questions are answered using random numbers and probabilities.

By propagating enough photons through this type of simulator, one acquires a statistical sampling of all possible photon paths, the "true image". The main drawback of Monte Carlo simulation is that, because there are a large number of possible paths, it requires a large number of photons to be propagated before the statistical error in the result is low. In consequence, it takes a very long computational time to acquire "good" (low statistical noise) images with this technique.

The advantages of it, however, are many. Because one has complete control over what happens to the photon, it is possible to turn on and off different effects such as scattering, energy resolution, and the intrinsic spatial resolution of the camera. This allows for the testing of techniques in simplified and controlled environments. Also, because one knows exactly what is happening to the photon, it is possible, for example, to have perfect knowledge of which photons have scattered and how often. This makes Monte Carlo an attractive technique for testing correction techniques.

Consequently, Monte Carlo is used extensively to evaluate the performance of scatter correction techniques [45, 39, 40, 60, 62, 76, 89, 92, 93, 99, 105, 122, 126]–[132] and more and more aspects of gamma camera imaging systems are being taken into account by the simulators [96, 133 – 135]. Monte Carlo is becoming capable of producing simulated data that are closer and closer to the actual reality, but, the major drawback of Monte Carlo simulators still remains, namely that they are slow and the more one builds into them, the slower they get.

A lot of work has gone into trying to improve the performance of Monte Carlo simulators. One major field of advancement has been in the use of variance reduction, for example [3, 129, 133, 135]–[137]. With variance reduction, the photon is forced to travel along a path that results in it being detected. The photon is then weighted based on the probability that it would have travelled along that path. The weight given to the photon determines how much it contributes to the final image.

Variance reduction can be divided into three different categories [3]: forced detection, forced non-absorption, and stratification. Forced detection means that the photon is forced to travel or scatter in directions towards the detector and forced to remain within the acceptance angle of the camera collimator. With forced non-absorption, the photon is not permitted to undergo an interaction with the medium which results in it being completely absorbed or in it dropping below the energy threshold for detection. Also, when a photon has an opportunity to undergo several different types of interactions, the photon history is split and all possible paths are followed. Stratification affects the starting location of the photon. The initial starting point is not chosen completely randomly, but instead those locations which are more likely to result in useful data are chosen preferentially.

Another avenue which has been explored to shorten the time required for executing Monte Carlo code is the use of multiprocessing [133, 138, 139]. Multiprocessing makes use of the ability of multiple CPUs to operate in parallel on independent pieces of data and consequently generate final results much faster. Monte Carlo simulation can benefit from this type of approach because most of the calculations performed in order to determine a photon history are independent, as is each individual photon history. For example, a simulation can be run on several computers simultaneously and the results simply combined afterwards. Vectorization is a variant of multiprocessing in which a set of CPUs in a single machine execute the same commands in unison on an array of data. Pipelining is another multiprocessing technique which speeds up calculations by setting up a series of processors as an assembly line along which the data is passed. The use of multiple processors is not specific to Monte Carlo codes and can be applied to any computer code involving many independent calculations.

Unfortunately, even with the use of both variance reduction and multiprocessing, the simulation of low statistical noise results still typically requires many hours of computation for small sources and even days for more complex situations.

The direct application of Monte Carlo as a reprojector in an iterative correction method was investigated by Floyd *et al* in 1985 [131, 140]. This technique produces an accurate estimate of the scatter distribution but is extremely slow and computer intensive, making it impractical for use in a clinical environment.

It is possible to precompute the photon transport matrix required for this technique but, because of the non-sparse nature of matrix once scatter is included, approximations are needed to reduce the calculation time and storage requirements down to a reasonable level [130].

2.4.2 Complete Analytical Calculations

With the recent improvements in computer technology, direct evaluation of the photon propagation equations has become possible. Because it models the same processes as does MC simulation, an analytical evaluation of the photon transport equation can retain the MC code's accuracy, but may be able to improve on the calculation time as it is no longer necessary to use stochastic techniques to produce results. The use of analytical calculations to estimate scatter has been explored in positron emission tomography (PET) [141, 142] and also to some extent in SPECT itself [143 – 151]. Many of the "analytical" techniques proposed for SPECT have begun with an analytical approach but then, part way through the development of the method, switched over to very simplified models or Monte Carlo calculations. There are some methods in SPECT which do perform a complete analytical calculation but, to date, these have either been restricted to simplified cases (first order Compton scatter in homogeneous media) or still require long computation times to include even second order Compton scatter.

For example, Walrand *et al* begin with an analytical expression for the first order

Compton scatter but then account for higher orders of scatter by means of a build-up function [143]. Nuyts *et al*, through the use of several approximations, develop a parameterized model which calculates the complete scatter PSF in homogeneous materials [144]. The parameters of the model are determined by a combination of Monte Carlo simulation and experimental measurements with water phantoms. This model, though based on physical arguments, does not attempt to evaluate the equations which actually describe the probabilities of photon interactions. The method in [145] uses Monte Carlo evaluation of sources in homogeneous models to fit a simple Gaussian function to the scatter distribution. Although this permits a fast evaluation of the scatter, this model does not accurately describe the shape of the distribution and has problems accounting for inhomogeneities in the scattering medium. The approach suggested in [146] uses a complete analytical evaluation of the scatter from the last scatter site. The fraction of the number of photons emitted from the source which reach the last scatter site, is determined by referencing a table of Monte Carlo results. This allows for the effective inclusion of higher order scatters which may have occurred before the last scatter site. However, the MC results were obtained in homogeneous media and extension of the table to include all possible inhomogeneous cases is impractical. Extrapolation on the table using water-equivalent depths would likely produce inaccuracies.

Techniques such as [147, 148] do evaluate an analytical expression of the photon interaction probabilities, but greatly simplify the problem. Their solutions are restricted to first order Compton scatter and the method in [147] is evaluated only for line sources in homogeneous media (*ie* the 2D LSF is calculated, not the full 3D PSF).

The only technique, to date, which does attempt a complete evaluation of the exact photon propagation probability is that proposed in [149 – 151]. This method is applicable to inhomogeneous media. It calculates the full 3D PSF "on the fly" using direct numerical integration of the photon transport equations. The technique computes the primary photon distribution and is capable of calculating all orders of Compton scatter. This approach, however, still requires very long calculation times for PSFs which include higher than first order Compton scatter. This can be a problem as it has been shown that second order Compton contributes a significant percentage of the detected scatter even in the photopeak window [45, 149, 152, 153]. To generate results in a reasonable time, one is also forced to restrict the accuracy of the numerical integration. Additionally, Rayleigh scatter is not included in this technique, although in principlé the technique could be extended to do so.

2.5 Conclusion

Although most of the methods discussed in this chapter do improve the image quality to one extent or another, none is both fast enough for use clinically and also sufficiently accurate for quantitative correction of scatter in non-homogeneous media. The ideal is, therefore, to create a method of scatter correction which is both quick to execute and yet provides accurate estimates of the scatter distribution even in the case of complex inhomogeneous media.

Our work has focused on the aspect of accurately determining the PSF. It falls into the category of an analytically based reprojector for SPECT which models both primary and scattered photon distributions. The technique is based upon a theoretical calculation of the scatter similar to those methods discussed in section 2.4. The aim of the work has been to develop a fast method of accurately calculating photon (scatter) distributions.

Chapter 3

The Method

Our method of determining the scatter distribution in SPECT projections is to numerically evaluate the analytical transport equations which describe the photon propagation from the source position, through the patient tissues, and to the detector. Direct numerical evaluation of these multi-dimensional integrals is accurate but very slow. However, many components of these equations are independent of the parameters in the system which change from one clinical study to the next: the patient tissue distribution and the activity distribution within the patient. By precalculating the patient independent parameters, it is possible to improve the calculation times while maintaining the accuracy of the original expressions.

Our technique uses an estimate of the source distribution, a patient specific attenuation map, and a precalculated lookup table [154]. It computes the primary photon distribution as well as that due to first and second order Compton scatter and first order Rayleigh scatter. The method can be extended to calculate higher order scatter but this is not necessary for ^{99m}Tc photopeak imaging. The technique does not have any free fitting parameters and so the photon distributions produced are completely determined by the characteristics of the SPECT camera system, the energy of the emitted photon, and the selected energy window.

The first three sections in this chapter deal with the approach we have taken to calculating the primary photon, Compton scatter, and Rayleigh scatter distributions respectively. The equations describing these distributions as well as the approximations we have used to decrease computation times are presented and discussed.

An extended source can be treated as a sum (integral) of several point sources and, therefore, in sections 3.1–3.3 we have concentrated on determining only the projections for a point source. The application of this technique to larger, more realistic sources, is discussed in section 3.4.

3.1 Primary Photon Distribution

The primary (unscattered) photon distribution for a point source is calculated by evaluating a two dimensional numerical integration over the detector surface. Unlike the scatter calculations, determining this distribution does not require an integration over the scattering medium and thus proceeds comparatively quickly. As a result, this distribution can be produced without resorting to extensive precalculations.

The probability that a photon is emitted from the point $\vec{s}$, passes through the patient without scattering, and is detected at a point $\vec{n}$ (Figure 3.1, thin solid line) can initially be expressed as continuous function of $\vec{s}$ and $\vec{n}$. However, the SPECT projection data, the reconstructed estimate of the source distribution, and also the attenuation map[1] are discrete sets of information. It is more meaningful, therefore, to bin this continuous probability and acquire the total probability that a photon is emitted from the point $\vec{s}$ and is detected within the pixel centred at $\vec{n}_c$.

[1] In simulated studies, like Monte Carlo simulations (Chapter 4) and phantom experiments (Chapter 5), the attenuation map is known exactly and so a continuous distribution of the tissue densities could be determined. However, in the case of real data, such as that from a patient study, the attenuation map would be reconstructed from a transmission scan and hence it would also be discretized.

The discrete primary photon distribution function, $\mathrm{PDF}(\vec{s},\vec{n}_c)$, for photons emitted from point $\vec{s}$ and detected within a pixel centred at the point $\vec{n}_c$ is obtained by integrating the continuous probability function over the pixel (equation (3.1)). To speed up the calculation, the attenuation path of the photons is approximated as the path going directly from $\vec{s}$ to $\vec{n}_c$.

$$\mathrm{PDF}(\vec{s},\vec{n}_c) \;=\; \mathrm{Q}\,\mathrm{P}_E(0)\exp\left(-\int_{\vec{s}}^{\vec{n}_c}\mu(y)\mathrm{d}y\right)\int_{\mathrm{A}_c}\mathrm{d}^2n\;\mathrm{F}(\xi)\frac{\cos(\xi)}{4\pi\mathrm{r}_{ns}^2}, \tag{3.1}$$

where

Q is the source strength (the number of photons emitted from the source during the time of acquisition of the projection),

A_c is the area of a pixel centred at $\vec{n}_c$ on the detector surface,

$\mathrm{F}(\xi)$ is the probability that a photon reaching the collimator surface at an angle ξ with respect to the detector normal will pass through the collimator and be detected,

$\mathrm{P}_E(0)$ is the probability that an unscattered photon will be recorded within a 20% energy window centred at the peak energy of the emitted photons,

r_{ns} is the distance between two points $\vec{n}$ and $\vec{s}$,

$\frac{\cos(\xi)}{\mathrm{r}_{ns}^2}\mathrm{d}^2n$ is the solid angle of a sphere centred at $\vec{s}$ and subtended by the detector pixel element d^2n, and

$\mu(y)$ is the attenuation coefficient at position y along the path travelled by the photon.

The collimator acceptance function $\mathrm{F}(\xi)$ is approximated as the probability of photon detection averaged over the entire surface of the detector. This function can be calculated theoretically (as is done in Chapter 4) or determined experimentally (as is done in Chapter 5). As an example of a simple theoretical formulation, one can model a parallel

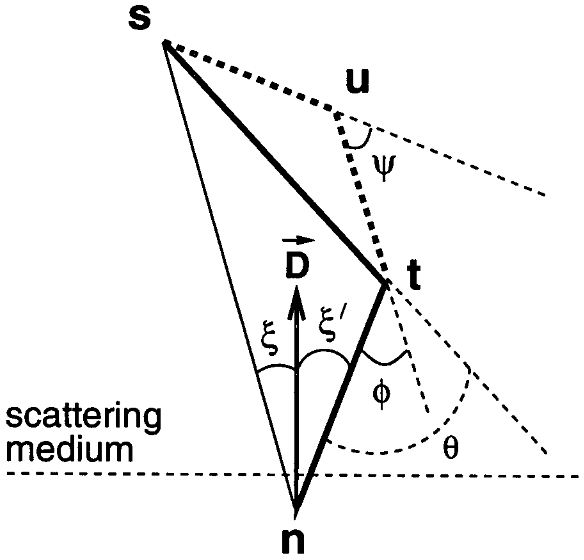

Figure 3.1: Photon paths. The unscattered photon path goes directly from the source at **s** to the detector at **n** (thin solid line). The thick solid line depicts a photon path which scatters once at **t**. A photon which scatters twice originates at the source s, scatters at **u** and **t**, and is detected at **n** (thick dotted line). The detector normal is denoted by $\vec{\mathbf{D}}$.

hole collimator as an array of cylindrical holes with no septal penetration and express the acceptance function as:

$$\mathrm{F}(\xi) = \mathrm{A}_{holes}\mathrm{E}(\xi)(1 - \exp(-\mu_{\mathrm{NaI}}L)). \tag{3.2}$$

Here, the fractional area, A_{holes}, is the average density of holes on the collimator surface. Similar to [149, 155], E(ξ) is the area of apparent intersection between the front end and the back end of the collimator hole when the front end is viewed from an angle of ξ. The function E(ξ) is normalized to one when ξ is zero. Finally, the last term in equation (3.2) gives the probability that the incident photon will interact in a NaI crystal with thickness L. We note that this formula for F(ξ) is independent of the photon energy (as was assumed by the MC codes used in Chapter 4). However, crystal efficiency is energy dependent [153] and a more accurate modelling of the detector would require a more complete formulation of F(ξ).

We also note that our acceptance function implicitly assumes a rotational symmetry about the detector normal and averages out the variation of F(ξ) over the detector surface. Real collimators more typically have hexagonal holes and this rotational symmetry effectively approximates these hexagons by circles[2]. The average acceptance function was chosen over a more exact, spatially varying representation in order to increase the symmetry in the geometry of the imaging situation and thus reduce the size of the lookup tables required for the scatter calculations in sections 3.2 and 3.3. The errors introduced by this choice are mitigated by the image blurring caused by the intrinsic spatial resolution of the SPECT camera. Additionally, the extended nature of realistic sources, as opposed to mathematical point sources, serves to effectively average the source over a larger area, further reducing the effects of this approximation.

For example, Figure 3.2 shows the difference between the exact collimator acceptance function and the average *theoretical* acceptance function calculated using equation (3.2). Shown in this figure is the probability that a photon, emitted from a point source 10cm above one of the collimator holes (0.0255cm from the hole centre), will reach the detector crystal. The collimator simulated was a hexagonally arrayed circular hole collimator with a hole diameter of 0.14cm, a collimator thickness of 3.28cm (the acceptance angle is 2.44^o), and a minimum septal thickness of 0.0152cm. No septal penetration was allowed.

Although the average acceptance function is quite different from the exact acceptance function, once convolution with the intrinsic spatial resolution is taken into account this difference is diminished. The discrepancy shown at angles less than 1^o in Figure 3.2 is caused by the point source being positioned over a collimator hole.

The value $P_E(0)$ from equation (3.1) is a special case of $P_E(\theta)$, the probability that a photon which has Compton scattered through an angle θ will be detected in the chosen energy window of the detector. The general form of $P_E(\theta)$ is discussed in association

[2] The collimators in the MC simulations used in Chapter 4 have circular holes and so, in this case, the approximation is exact.

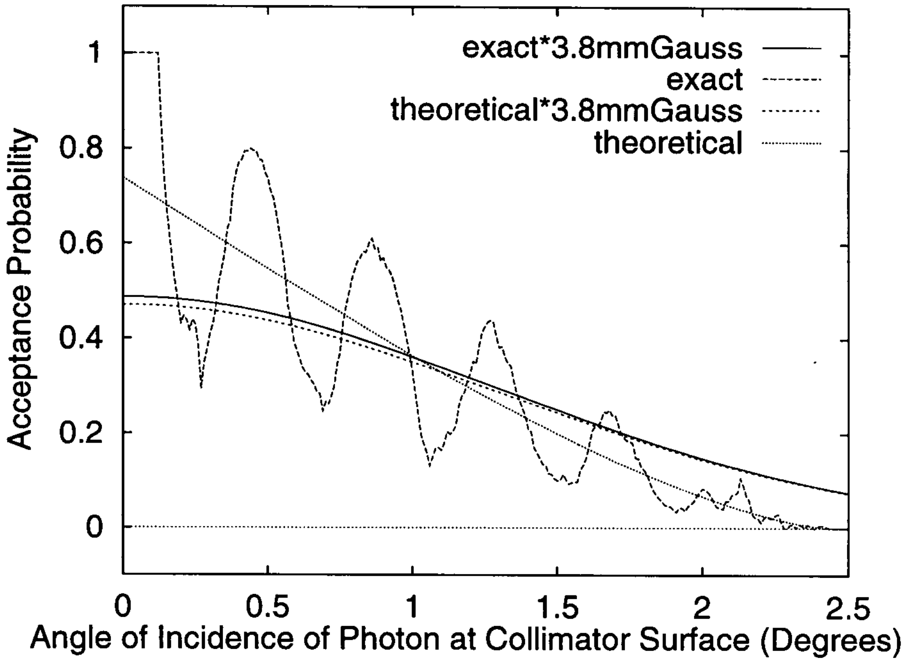

Figure 3.2: The average acceptance function F(ξ). Shown here are acceptance functions for a point source located 10cm above a collimator surface. The collimator is an hexagonally arrayed circular hole collimator with an acceptance angle of 2.44°. There is assumed to be no septal penetration. The curve *exact* corresponds to the exact collimator acceptance function, and *theoretical* to the average collimator acceptance function as determined theoretically as per equation (3.2). These two functions are also shown after convolution with the intrinsic spatial resolution (a Gaussian function of 3.8mm FWHM).

with equation 3.3. For unscattered ^{99m}Tc photons, $P_E(0)$=0.9815.

The PDF as written in equation (3.1) is valid for a camera with perfect intrinsic spatial resolution[3]. To account for the intrinsic spatial resolution of a realistic camera, one convolves the PDF with a function, G_d, which describes the true resolution [119, 123, 144, 155]. The convolution can be done after the calculation of the PDF with no significant increase in the overall computation time. Note that the distance dependence of the camera spatial resolution (collimator blurring) is incorporated into our calculation of the PDF through our modelling of the collimator, $F(\xi)$.

3.2 Compton Scatter Distribution

The Compton scatter distribution function (SDF) is calculated by evaluating the probability that a photon starting at a given point in the medium will scatter one or more times within the object and then be detected at a given point on the detector surface. The probability is based on the Klein-Nishina cross-section for Compton scattering. The SDF is obtained by integrating the probability over all points within the scattering medium and then scaling the result by the number of photons emitted during the acquisition of the projection.

A large portion of the SDF calculation depends only on the characteristics of the camera, the initial energy of the photons emitted from the radioactive source, the chosen energy window, and on the positions of the source, the scattering point(s), and the detector and it is, therefore, possible to precalculate this portion and store its values in a reference table. The use of reference tables leads to a substantial decrease in the time required to generate photon distributions.

[3] Perfect intrinsic spatial resolution means that the position recorded by the camera for a photon is exactly the position at which it entered the crystal.

The techniques we have used to calculate the first and second order Compton scatter distributions are described in detail in sections 3.2.1 and 3.2.2 respectively. Our method can, however, be extended to calculate third and higher order Compton scatter as well (section 3.2.3) if this is desired.

3.2.1 First Order Compton Scatter

In order to determine the first order scatter distribution function (SDF_1), we consider photons that travel along the path shown in Figure 3.1 (thick solid line). The photon is emitted from the source located at the point $\vec{s}$ and propagates through the medium to a scattering point $\vec{t}$. After scattering, it travels to the boundary of the medium, leaves, and arrives at the detector at the point $\vec{n}$. The angle at which the photon enters the detector with respect to the detector normal $\vec{\mathrm{D}}$ is denoted by ξ'.

The probability of detecting a photon is initially expressed as a continuous function of the positions $\vec{s}$, $\vec{t}$, and $\vec{n}$. However, as was done with the primary distribution (PDF), the continuous function is discretized and the total probability is determined for a photon which is emitted from the point $\vec{s}$, scatters anywhere in the voxel centred at $\vec{t}_i$ and is detected within the pixel centred at $\vec{n}_c$. The discrete first order Compton scatter distribution function $\mathrm{SDF}_1(\vec{s},\vec{t}_i,\vec{n}_c)$, is obtained by integrating the continuous function $\mathrm{SDF}_1(\vec{s},\vec{t},\vec{n})$ over the area, A_c, of a pixel on the detector surface centred at $\vec{n}_c$ and also integrating over the volume, V_i, of the voxel centred at $\vec{t}_i$.

$$\begin{aligned}\mathrm{SDF}_1(\vec{s},\vec{t}_i,\vec{n}_c) \;=\; & \mathrm{Q}\int_{\mathrm{A}_c} \mathrm{d}^2 n \int_{\mathrm{V}_i} \mathrm{d}^3 t\, \rho_e(\vec{t}) \frac{\mathrm{d}\sigma(\theta,\alpha)}{\mathrm{d}\Omega} \mathrm{F}(\xi')\mathrm{P}_E(\theta) \frac{\cos(\xi')}{4\pi \mathrm{r}_{nt}^2 \mathrm{r}_{ts}^2} \\ & \exp\left(-\int_{\vec{s}}^{\vec{t}} \mu(y)\mathrm{d}y - \int_{\vec{t}}^{\vec{n}} \mu(y,\theta)\mathrm{d}y\right) \qquad (3.3)\end{aligned}$$

where

$\frac{d\sigma}{d\Omega}(\theta, \alpha)$ is the Klein-Nishina scattering cross-section (equation 1.5),

$\rho_e(\vec{t})$ is the electron density at position $\vec{t}$,

θ is the Compton scattering angle,

$\mu(y, \theta)$ is the attenuation coefficient at position y for a photon which has scattered through an angle θ ($\mu(y, 0) \equiv \mu(y)$)[4],

α is the energy of the photon divided by the rest mass of the electron.

The remaining symbols in equation (3.3) are as defined in equation (3.1).

$\mathrm{P}_E(\theta)$ describes the probability that a photon which has Compton scattered through an angle θ will be detected in the chosen energy window of the detector. For a 20% symmetric energy window centred at 140keV (^{99m}Tc), this probability is expressed as:

$$\mathrm{P}_E(\theta) = \frac{1}{2}\left[\mathrm{Erf}\left(\frac{154 - \mathrm{En}(\theta)}{\sqrt{2}\Lambda(\theta)}\right) - \mathrm{Erf}\left(\frac{126 - \mathrm{En}(\theta)}{\sqrt{2}\Lambda(\theta)}\right)\right] \qquad (3.4)$$

where

$$\begin{aligned} \mathrm{En}(\theta) &= \frac{140}{1 + \frac{140}{511}(1 - \cos(\theta))} \\ \Lambda(\theta) &= \mathrm{En}(\theta)\frac{0.1}{\sqrt{8\log(2)}}. \end{aligned}$$

This form of $\Lambda(\theta)$ is the same as in [10, 92] and was chosen to match the energy resolution of the Monte Carlo simulators used in Chapter 4. Other forms of $\Lambda(\theta)$, such as those in [118, 149], could equally well have been used. The Erf(Z) function is the error function as defined in, for example, [156] which evaluates to the area under a Gaussian curve between −Z and +Z.

The energy dependent attenuation coefficients, $\mu(y, \theta)$, are stored as a one dimensional

[4] More precisely, μ depends on the path length (y) and the energy of the photon. However, the energy depends on the scattering angle (equation (1.6)) and so for notational convenience we have denoted this dependence as $\mu(y, \theta)$.

array for each different type of attenuating material. The attenuation coefficients used in the calculation are determined by linear interpolation on these arrays.

As with the PDF, equation (3.1), SDF_1 is also convolved with a function G'_d to account for a realistic camera intrinsic spatial resolution. The function, G'_d, is independent of the attenuation map and the source distribution and could, therefore, be included in the precalculation. This would avoid the need to perform the convolution after the real-time calculation and also allow one to account for the energy dependence of G'_d. This was not done because it would increase the size of the reference table without decreasing the speed of the real-time calculation and because the MC codes we used (Chapter 4) did not model G'_d energy dependence.

The function G'_d is not exactly the same as intrinsic spatial resolution function, G_d, used for the unscattered distribution. The difference is an effect of the extra binning used to discretize the continuous scatter function and create our lookup table. The binning operation does not commute with convolution and thus the extra binning must be undone before G_d can be applied. The "unbinning operator" is incorporated into G_d to create G'_d.

Equation (3.3) calculates the number of photons recorded in a detector pixel which have scattered within a single voxel. However, the scattered photons recorded in a SPECT projection are the photons which come from scattering events occurring throughout the entire medium, not just those within one voxel. Therefore, to obtain the actual discrete scatter distribution in a SPECT projection, $\mathrm{SDF}_1(\vec{s},\vec{n}_c)$, we sum over all of the scattering voxels.

$$\mathrm{SDF}_1(\vec{s},\vec{n}_c) = \sum_{i=1}^{I} \mathrm{SDF}_1(\vec{s},\vec{t}_i,\vec{n}_c) \qquad (3.5)$$

where I is the total number of voxels which contain scattering material.

As the attenuation map of the scattering object is discrete, so too is the map of the electron density distribution. Therefore, the electron density is constant over each voxel and we can rewrite $\rho_e(\vec{t})$ as $\rho_e(\vec{t_i})$ for all $\vec{t}$ within the voxel. We now approximate the path of the photon as that travelling through the centre of the scattering voxel to the centre of the detector pixel.

$$\exp\left(-\int_{\vec{s}}^{\vec{t}} \mu(y)\mathrm{d}y - \int_{\vec{t}}^{\vec{n}} \mu(y,\theta)\mathrm{d}y\right) \approx \exp\left(-\int_{\vec{s}}^{\vec{t_i}} \mu(y)\mathrm{d}y - \int_{\vec{t_i}}^{\vec{n}_c} \mu(y,\theta_{ic})\mathrm{d}y\right) \tag{3.6}$$

where θ_{ic} is the angle between the vectors $(\vec{t_i} - \vec{s})$ and $(\vec{n}_c - \vec{t_i})$.

Equations (3.3) and (3.5) are combined and rewritten as

$$\mathrm{SDF}_1(\vec{s},\vec{n}_c) \;=\; \mathrm{Q}\sum_{i=1}^{I} \rho_e(\vec{t_i})\,\mathrm{K}_{st_in_c} \exp\left(-\int_{\vec{s}}^{\vec{t_i}} \mu(y)\mathrm{d}y - \int_{\vec{t_i}}^{\vec{n}_c} \mu(y,\theta_{ic})\mathrm{d}y\right), \tag{3.7}$$

where

$$\mathrm{K}_{st_in_c} = \int_{\mathrm{A}_c} \mathrm{d}^2 n \int_{\mathrm{V}_i} \mathrm{d}^3 t \,\frac{\mathrm{d}\sigma(\theta,\alpha)}{\mathrm{d}\Omega}\,\mathrm{F}(\xi')\,\mathrm{P}_E(\theta)\,\frac{\cos\xi'}{4\pi \mathrm{r}_{ts}^2 \mathrm{r}_{nt}^2}. \tag{3.8}$$

The factor $\mathrm{K}_{st_in_c}$ is independent of the patient and is precalculated using Romberg's method of numerical integration [157]. The numerical integrations are evaluated to within 2% of their true values. This error could be decreased by increasing the accuracy of the integration, but this also increases the time required to generate the look-up table.

The attenuation coefficient, $\mu(y,\theta_{ic})$, is determined by adjusting the value from the attenuation map, $\mu(y)$, based on interpolation in a one dimensional array which specifies the change in the attenuation coefficient as a function of photon energy. A different array is stored for each different type of attenuating material. The final SDF, equation (3.7), is evaluated in real-time and incorporates the specifics of the patient, namely the attenuation map and the source distribution.

The memory required to store our look-up table, for a scattering domain of 64x64x64 voxels, is only 25 Mbytes. To store a separate value for each combination of 64^3 possible source positions, 64^3 possible scattering sites, and 64^2 possible detector pixels would require many terabytes of memory. However, by removing the patient dependencies,

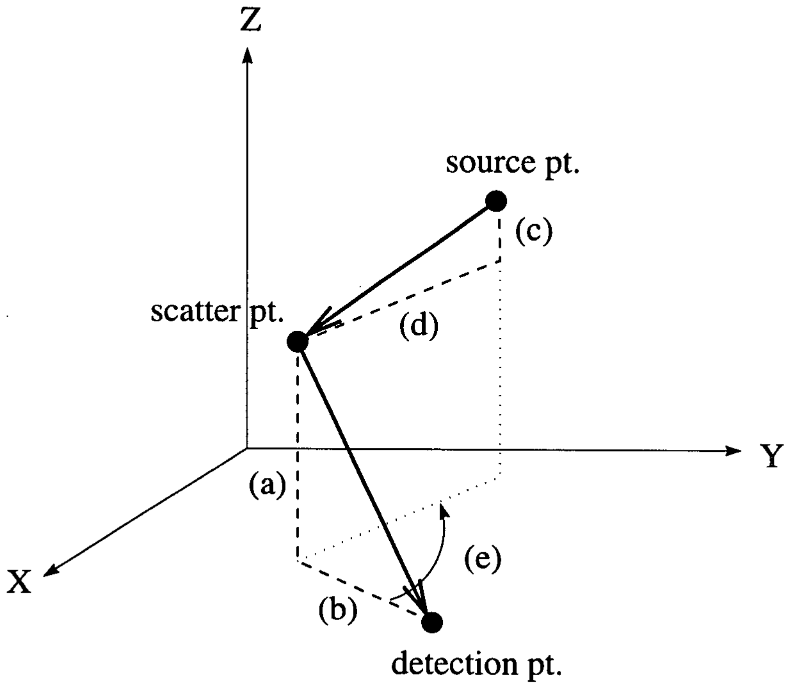

Figure 3.3: Parameterization of the five dimensional look-up table. The parameters are four distances (a-d) and one angle (e). The detector normal is in the direction of the z-axis while the detector plane is parallel with the x-y plane.

one finds that the majority of these values are redundant and that it is possible to parameterize all possible combinations with four distances and one angle.

As shown in Figure 3.3, the distances are those from the centre of the detector pixel to the centre of the last scattering voxel both parallel (a) and perpendicular (b) to the detector normal (z-axis) and the distance from the source to the centre of the scattering voxel both parallel (c) and perpendicular (d) to the detector normal. The angle (e) is that between the vector from the scattering point to the detector point and the vector from the scattering point to the source point projected into the plane perpendicular to the detector normal (x-y plane).

These parameters span the range of values (attainable in the limits of the scattering space) which result in non-zero table entries. Interpolation between entries in the table is used during the real-time calculation if the required entry is not explicitly available. The voxelized grid of the scattering space remains constant with respect to the detector surface and so the attenuation map is recalculated for each different projection angle. The detector-scatter distances thus form a fixed discrete set of values which reduces the amount of interpolation required during the real-time evaluation of the SDF. Our parameterization of the look-up table implies rotational symmetry about the detector normal which is violated by a rectangular voxel(pixel)ization. This is corrected for, however, by using the correlation between the scatter voxel–detector pixel distances and the voxel–pixel orientation.

The time required to generate this look-up table is still quite long, 10 weeks on a single Sun Sparc10 with 32 Mbytes of internal memory, because each of the more than six million entries requires the numerical evaluation of a five-dimensional integral. However, the calculation of this table need only be done once for each camera system and initial photon energy and it is readily adaptable to multiple processors. For example, our tables were actually generated in 1.75 weeks by running the code on six different machines: two Sparc10s, a Sparc2, a Sparc Classic, a Pentium Pro (200MHz) and a Pentium (120MHz), and concatenating the results.

It should be noted that equation (3.7) does not include the incoherent scattering functions, $\mathcal{S}(x, \overline{Z})$. The Klein-Nishina formula for the Compton cross-section assumes that the electron off of which the photon scatters is a free electron. This is not, of course, true for the majority of electrons in biological materials. To account for the electrons being bound to atoms and not free, one needs to include the incoherent scattering function.

The incoherent scattering functions, $\mathcal{S}(x, \overline{Z})$, multiply the integrand in equation 3.3. They depend on $\overline{Z}$, the effective atomic number of the scattering material, and the parameter x which is proportional to the momentum transferred to the electron from the photon when it scatters. The parameter x is related to the scattering angle θ by:

$$x = \frac{1}{\lambda} \sin\left(\frac{\theta}{2}\right), \tag{3.9}$$

where λ is the wavelength of the incident scatter photon.

Although the functions, $\mathcal{S}(x, \overline{Z})$, depend on the scattering material, it is possible to include an approximation for them in the precalculation (equation (3.8)). This is an accurate approximation because the incoherent scattering functions are very similar for different biological materials (Figure 3.4). For example, the difference between bone and water, two of the more dissimilar materials, is greatest at 7^{o}[5] and corresponds to a 5% percent deviation from the water value for 140keV photons (^{99m}Tc). One also notes that the energy dependence of $\mathcal{S}(x, \overline{Z})$ is small (Figure 3.5). For photons in water with an energy of 126keV, the product of $\mathcal{S}(x, \overline{Z})$ with the Klein Nishina differential cross-section differs most from 140keV photons at 11^{o} where it deviates by 1.3%. Therefore, to include the incoherent scattering functions into the precalculation, $\mathcal{S}(x, \overline{Z})$ is approximated as that for water at the unscattered photon energy (140keV for ^{99m}Tc). The overall effect of the incoherent scattering functions on the acquired photon distribution is small ($<$ 3%) and consequently, the error introduced by this approximation is less than 0.2%.

The incoherent scattering functions were not included in the comparisons of our technique with Monte Carlo simulations (Chapter 4) because the SIMSET simulator did not use them. For the comparison with the experimental results (Chapter 5), the incoherent scattering functions were included approximately and incorporated into the precalculation of the lookup tables. The functions, $\mathcal{S}(x, \overline{Z})$, used to estimate the error introduced by our approximation (and also incorporated into the lookup tables used for the experimental work in Chapter 5) are those tabulated in [25]. Although these scattering

[5] This difference takes into account the angular distribution of the Klein-Nishina differential cross section $d\sigma/d\theta$.

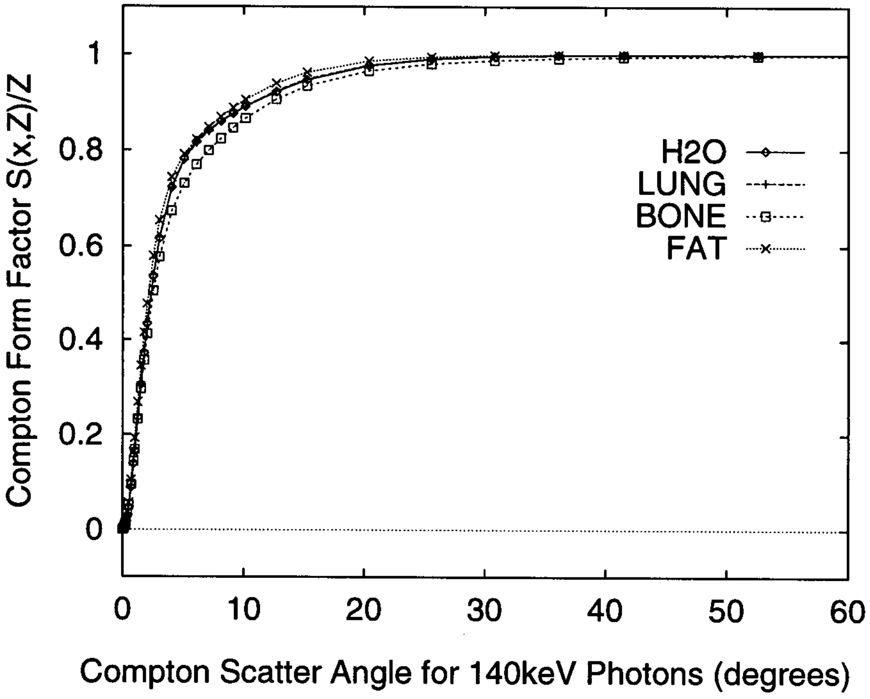

Figure 3.4: Incoherent scattering functions for different biological materials. The scattering functions are calculated from the data in [25].

functions do not include the effects of coherent-incoherent interference, this effect does not significantly change them, it primarily effects only the Rayleigh form factors [28].

3.2.2 Second Order Compton Scatter

Second order Compton scatter should be included in the photon distribution calculation as it can amount to 12% of the total scattered photons [45, 152] detected in the photopeak window. The calculation of the second order scatter, if done exactly, takes much longer than that of the first order scatter as it requires an additional 3D integration over the scattering space. Therefore, some approximations are made which allow for many of the additional calculations required for second order scatter to be included in the precalculated lookup table. The resultant table is the same size as the first order lookup table and including second order scatter only increases the calculation time by approximately

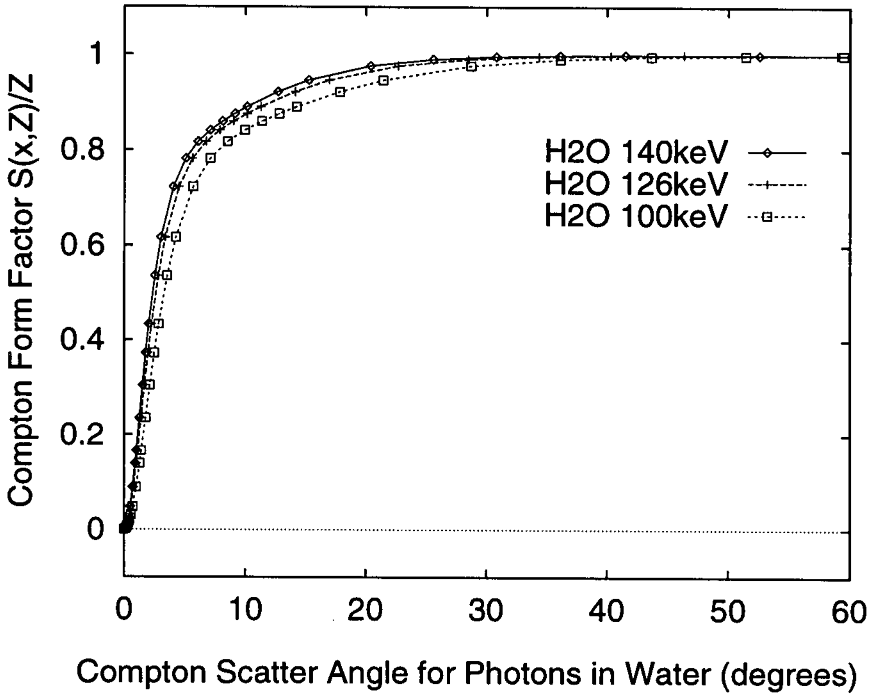

Figure 3.5: Incoherent scattering functions for water with different energies of incident photon. The scattering functions are calculated from the data in [25].

50% over the first order scatter calculation.

The second order scatter distribution function ($\mathrm{SDF}_2(\vec{s},\vec{u},\vec{t},\vec{n})$) is calculated for those photons that travel along the paths such as that shown in Figure 3.1 (thick dotted line). The photon on this path is emitted from $\vec{s}$, travels through the medium and scatters at $\vec{u}$ through an angle ψ and then at $\vec{t}$ through an angle ϕ. Finally, it exits the scattering medium and reaches the detector at the point $\vec{n}$.

Analogous to equation (3.3), the equation which gives the number of photons which scatter twice, in voxels centred at $\vec{u}_j$ and $\vec{t}_i$, before being detected in a pixel centred at $\vec{n}_c$, is given by

$$\mathrm{SDF}_2(\vec{s},\vec{u}_j,\vec{t}_i,\vec{n}_c) = \mathrm{Q}\int_{\mathrm{A}_c}\mathrm{d}^2n\int_{\mathrm{V}_i}\mathrm{d}^3t\int_{\mathrm{V}_j}\mathrm{d}^3u\left(\frac{\mathrm{d}\sigma(\psi,\alpha_0)}{\mathrm{d}\Omega}\right)\left(\frac{\mathrm{d}\sigma(\phi,\alpha_\psi)}{\mathrm{d}\Omega}\right)\frac{\cos(\xi')}{4\pi\,\mathrm{r}_{us}^2\,\mathrm{r}_{tu}^2\,\mathrm{r}_{nt}^2} \quad (3.10)$$
$$\mathrm{P}_E(\psi,\phi)\,\mathrm{F}(\xi')\rho_e(\vec{t})\rho_e(\vec{u})\exp\left(-\int\mu(y,\psi,\phi)\mathrm{d}y\right)$$

where $\mathrm{P}_E(\psi,\phi)$ is defined similar to equation (3.4) but with $\mathrm{En}(\theta)$ replaced by

$$\mathrm{En}(\psi,\phi) \;=\; \frac{140}{1+\frac{140}{511}(2-\cos(\psi)-\cos(\phi))}$$

and the remaining symbols are as defined in equation (3.3). The second Compton scattering cross-section term depends on α_ψ which is the energy of the photon, in units of the rest mass of an electron, after it has scattered through an angle ψ.

The second order projection is then given by the summation

$$\mathrm{SDF}_2(\vec{s},\vec{n}_c) = \sum_{i=1}^{I}\sum_{j=1}^{I}\mathrm{SDF}_2(\vec{s},\vec{u}_j,\vec{t}_i,\vec{n}_c) \quad (3.11)$$

where i and j are scatter voxel indices and I is defined as in equation (3.5).

As was done during the first order calculation, we approximate the electron density in a voxel as a constant and the attenuation path to be through the centre of the scattering voxels and detector pixel. We also compute the Klein-Nishina cross-section at the second scattering point assuming that the photon energy was unchanged during the first scatter. Because its energy must fall within the primary photopeak energy window in order for a photon to be detected, the majority of detected twice-scattered photons have first scatter angles less than 30° and this approximation to the Klein-Nishina cross-section introduces an error of less than 0.5% in 95% of detected twice-scattered events (Figure 3.6). These three approximations are summarized as follows

$$\rho_e(\vec{t})\left(\frac{\mathrm{d}\sigma(\phi,\alpha_\psi)}{\mathrm{d}\Omega}\right)\frac{1}{\mathrm{r}_{tu}^2}\mathrm{P}_E(\phi) \;\approx\; \rho_e(\vec{t}_i)\left(\frac{\mathrm{d}\sigma(\phi_\mathrm{j},\alpha_0)}{\mathrm{d}\Omega}\right)\frac{1}{\mathrm{r}_{tu_j}^2}\mathrm{P}_E(\phi_j),$$
$$\exp\left(-\int_{\vec{u}}^{\vec{t}}\mu(y,\psi)\mathrm{d}y-\int_{\vec{t}}^{\vec{n}}\mu(y,\psi,\phi)\mathrm{d}y\right)\approx\exp\left(-\int_{\vec{u}}^{\vec{t}_i}\mu(y,\psi_i)\mathrm{d}y-\int_{\vec{t}_i}^{\vec{n}_c}\mu(y,\psi_j,\phi_j)\mathrm{d}y\right),$$

$$\left(\frac{\mathrm{d}\sigma(\psi,\alpha_0)}{\mathrm{d}\Omega}\right)\frac{\mathrm{P}_E(\psi,\phi)}{\mathrm{P}_E(\phi)} \approx \left(\frac{\mathrm{d}\sigma(\psi_i,\alpha_0)}{\mathrm{d}\Omega}\right)\frac{\mathrm{P}_E(\psi_i,\phi_{ic})}{\mathrm{P}_E(\phi_{ic})},$$

where

$\phi_j = \phi$ evaluated at $\vec{u} = \vec{u}_j$ (ϕ_j still depends on $\vec{t}$ and $\vec{n}$),

$\psi_i = \psi$ evaluated at $\vec{t} = \vec{t}_i$ (ψ_i depends on $\vec{u}$),

$\psi_j = \psi$ evaluated at $\vec{u} = \vec{u}_j$ (ψ_j depends on $\vec{t}$),

$\phi_{ic} = \phi$ evaluated $\vec{t} = \vec{t}_i$ and $\vec{n} = \vec{n}_c$ (ϕ_{ic} depends on $\vec{u}$), and

α_0 is the unscattered photon energy in units of the rest mass of the electron.

With some additional algebraic rearrangements of the attenuation term, equations (3.10) and (3.11) are rewritten as

$$\begin{aligned}\mathrm{SDF}_2(\vec{s},\vec{n}_c) = & \mathrm{Q}\sum_{i=1}^{I}\sum_{j=1}^{I}\rho_e(\vec{t}_i)\,\mathrm{K}_{u_j t_i n_c}\exp\left(-\int_{\vec{s}}^{\vec{t}_i}\mu(y)\mathrm{d}y - \int_{\vec{t}_i}^{\vec{n}_c}\mu(y,\psi_j,\phi_j)\mathrm{d}y\right) \qquad (3.12)\\ & \left[\int_{\mathrm{V}_j}\mathrm{d}^3u\left(\frac{\mathrm{d}\sigma(\psi_i,\alpha_0)}{\mathrm{d}\Omega}\right)\frac{1}{\mathrm{r}_{us}^2}\,\frac{\mathrm{P}_E(\psi_i,\phi_{ic})}{\mathrm{P}_E(\phi_{ic})}\rho_e(\vec{u})\right.\\ & \left.\exp\left(\int_{\vec{s}}^{\vec{t}_i}\mu(y)\mathrm{d}y - \int_{\vec{s}}^{\vec{u}}\mu(y)\mathrm{d}y - \int_{\vec{u}}^{\vec{t}_i}\mu(y,\psi_i)\mathrm{d}y\right)\right]\end{aligned}$$

where $\mathrm{K}_{u_j t_i n_c}$ is the same factor as $\mathrm{K}_{s t_i n_c}$ (equation (3.8)) but with the source position moved from $\vec{s}$ to $\vec{u_j}$.

In order to precalculate the j summation, we approximate $\rho_e(\vec{u})$ as the average electron density along the path directly connecting the source point, $\vec{s}$, and the centre of the second scattering voxel, $\vec{t_i}$. We then note that the ratio of the electron density of water to that of other biological materials is approximately the same as the ratio of their attenuation coefficients[6]. We use this fact to express the average electron density as the ratio of the average attenuation coefficient to the coefficient for water, μ_w, times the

[6] For example, at 140 keV, the percentage differences between these two ratios for bone, lung, and muscle tissues are 3.55%, 0.06%, and 0.06% respectively [158].

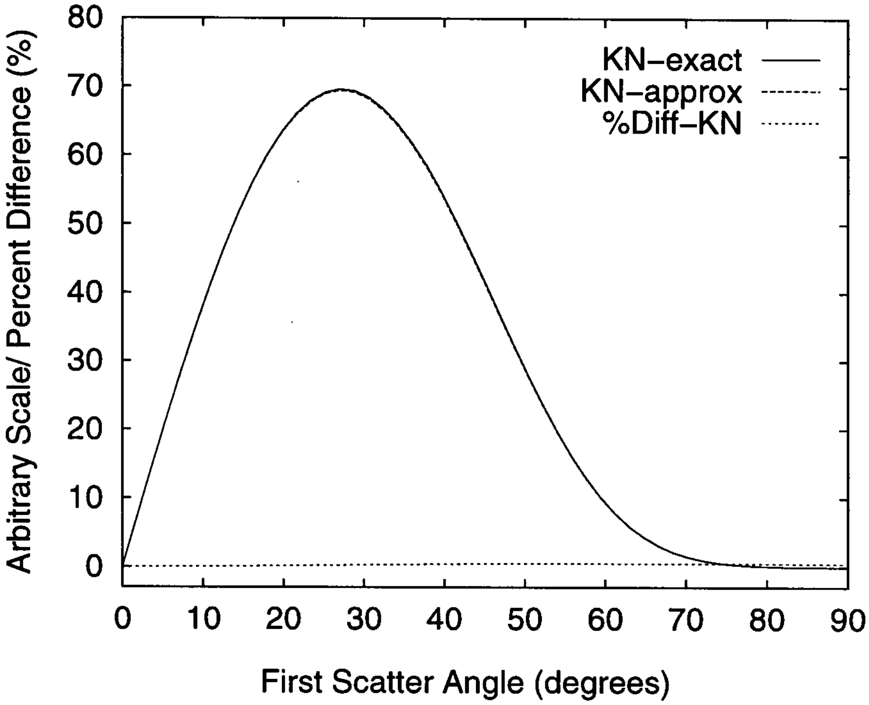

Figure 3.6: Angular distribution of detected twice scattered photons. This figure shows the effect of the photon energy approximation on the Klein-Nishina cross-section. *KN-exact* refers to the product $\frac{d\sigma}{d\Omega}(\theta,\mathrm{E}(0))$ * $\frac{d\sigma}{d\Omega}(\theta',\mathrm{E}(\theta))$ * $\mathrm{P_E}(\mathrm{E}(\theta,\theta'))$ where $\frac{d\sigma}{d\Omega}$ is the Klein-Nishina differential cross section. The function $\mathrm{P_E}$ is the energy detection probability for the standard 20% ^{99m}Tc photopeak energy window (equation 3.1). The function $\mathrm{E}(\theta)$ gives the energy of a photon which has scattered through the angle θ, $\mathrm{E}(\theta,\theta')$ is the energy if the photon has scattered through θ and θ'. *KN-approx* is the approximate relation used in our calculations: $\frac{d\sigma}{d\Omega}(\theta,\mathrm{E}(0))$ * $\frac{d\sigma}{d\Omega}(\theta',\mathrm{E}(0)$ * $\mathrm{P_E}(\mathrm{E}(\theta,\theta'))$. *%Diff-KN* gives the percent difference between *KN-exact* and *KN-approx*.

electron density of water, ρ_w.

$$\rho_e(\vec{u}) \approx \frac{\int_{\vec{s}}^{\vec{t}_i} \mu(y)\mathrm{d}y}{\mu_w\, \mathrm{r}_{st_i}} \rho_w \tag{3.13}$$

The attenuation of the twice-scattered photon is taken to be the attenuation along the path directly connecting the source, $\vec{s}$, to the second scatter site, $\vec{t}_i$, plus a correction factor; the second exponential term in equation (3.12). This correction factor is approximated as

$$\exp\left(\int_{\vec{s}}^{\vec{t}_i} \mu(y)\mathrm{d}y - \int_{\vec{s}}^{\vec{u}} \mu(y)\mathrm{d}y - \int_{\vec{u}}^{\vec{t}_i} \mu(y,\psi_i)\mathrm{d}y\right) \approx \exp\left(\mu_w \left(\mathrm{r}_{st_i} - \mathrm{r}_{us} - \mathrm{r}_{ut_i}\right)\right).$$

The effect of this attenuation approximation for a source in water is shown in Figure 3.7. It introduces an error of less than 2.4% in 95% of detectable twice scattered photons.

These attenuation and electron density approximations may seem to be poor approximations in regions such as the chest where lung density is significantly different than that of water, but this is not so. For example consider the extreme case where the source, the second scattering site, and the path connecting them are all in water, but are located on the border of a lung. For a connecting path length of 10cm, our approximations result in errors of less than 23% for 95% of the scattered photons (Figure 3.8). The average error for all of the detectable twice-scattered photons is 17%. However, second order scatter is only 12% of the total scatter and so this is only an error of 2% in the overall detected scatter. Additionally, each projection pixel contains the sum of contributions from all possible scatter sites, not just this one, and the majority of the second scatter sites are not on the water/lung boundary and so would contain even less error, further reducing the influence of our approximations on the final projection. An example of this 'worst case' scenario is presented in our simulation of a point source next to a lung wall in a simplified small chest model (section 4.2, case 2).

Additionally, we approximate $\mu(y, \psi_j, \phi_j)$ as $\mu(y, \theta_{ic})$ (θ_{ic} defined as in equation (3.6)) over the segment of the path between $\vec{t}_i$ and $\vec{n}_c$. That is, we calculate the attenuation

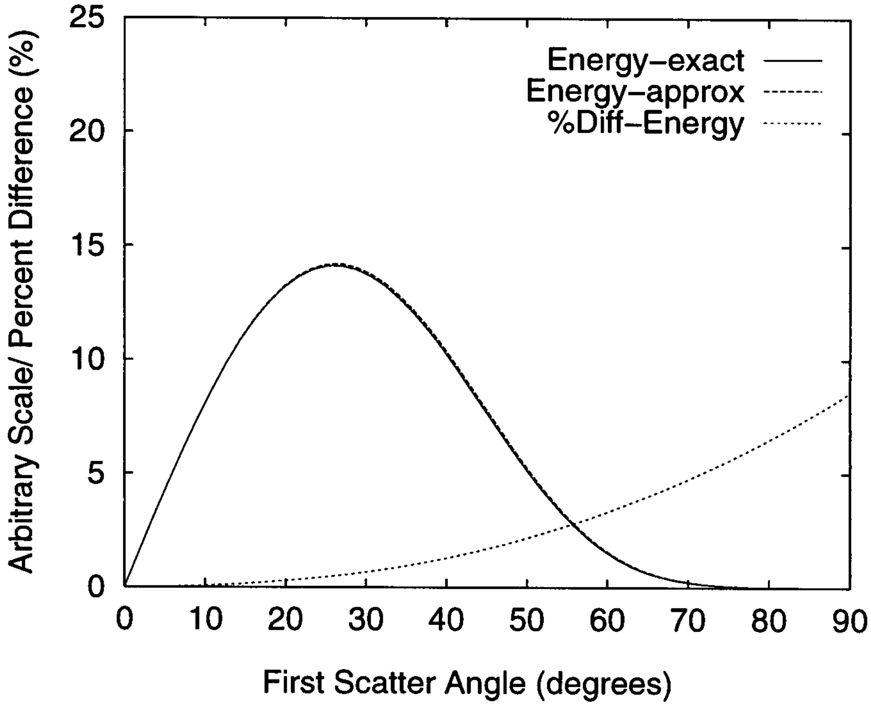

Figure 3.7: The energy-attenuation approximation. This figure shows the effect of assuming no energy is lost by the photon when computing the attenuation factor from source point to the second scatter site. *Energy-exact* is the *KN-exact* line from Figure 3.6 attenuated using attenuation coefficients which are exactly adjusted for the change in energy of the scattered photon. *Energy-approx* is the *KN-approx* line from Figure 3.6 attenuated using the attenuation approximation discussed in the text. *%-Diff-Energy* is the percent difference between *Energy-exact* and *Energy-approx*.

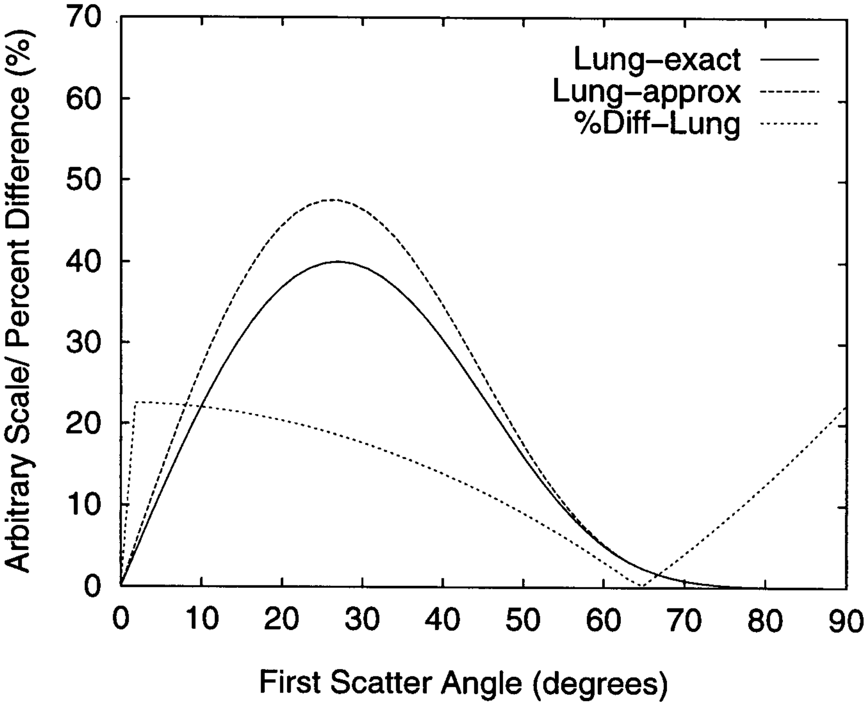

Figure 3.8: Angular distribution of twice scattered photons. This figure shows the relative number of detected twice scattered photons as a function of the first scattering angle for a "worst case" scenario. These curves are for a source point and second scattering site both located in water but positioned on the boundary of a lung. The two points are 10cm apart. The first scattering site is assumed to be always in the lung. The *Lung-exact* curve is the *KN-exact* line from Figure 3.6 scaled by the electron density for lung tissue and exactly corrected for the attenuation along the path through the first scattering site to the second. The *Lung-approx* curve is the *KN-approx* line from Figure 3.6 scaled by the electron density for water (as would be determined by our electron density approximation) and attenuated using the attenuation approximation discussed in the text. *%-Diff-Lung* is the percent difference between *Lung-exact* and *Lung-approx.*

coefficient at the energy of a photon which has scattered through the angle θ_{ic} instead of at the energy of a photon which has scattered through angles ψ_j and ϕ_j. For detectable twice-scattered photons, the average error that this approximation introduces in the attenuation coefficients for lung tissues and water is less than 1%. The error this produces in the attenuation factor varies approximately linearly with the depth of the last scatter point in the tissues and is 1.5% for 10cm of water. Again, this is the error in the number of twice-scattered photons; the error in the total number of scattered photons would be 0.18%.

Finally, we assume that all potential second order scattering sites do contribute, that is we run the summation over j from 1 to J, the total number of voxels, instead of to I, the total number of voxels which actually contain scattering material. This is done because, *a priori*, we do not know which of the voxels will contain scattering material. However, some of the possible positions for $\vec{u}$ could be outside of the scattering body and would not, therefore, contribute to the scatter. This type of error is most significant when both $\vec{s}$ and $\vec{t}$ are near the edge of the body. For example, if the body edge was flat, this approximation would falsely double the number of second-order scatter events. However, human bodies are generally convex, and most detectable second-order scatter events have a first scatter angle of less than 30^o (95% have a first scatter angle less than 60^o), and thus most of the significant $\vec{u}$ positions will be inside of the body and will, therefore, contribute to the scatter. This argument is similar to that used in [119, 123]. A source near the edge of the scattering medium is one of the cases examined in Chapter 4 (case 1(ii)).

Equation (3.12) may now be rewritten as

$$\begin{aligned}\mathrm{SDF}_2(\vec{s},\vec{n}_c) &= \\ &\mathrm{Q}\sum_{i=1}^{I} \rho_e(\vec{t}_i)\,\mathrm{K}'_{st_in_c}\left(\int_{\vec{s}}^{\vec{t}_i}\mu(y)\mathrm{d}y\right)\exp\left(-\int_{\vec{s}}^{\vec{t}_i}\mu(y)\mathrm{d}y-\int_{\vec{t}_i}^{\vec{n}_c}\mu(y,\theta_{ic})\mathrm{d}y\right), \qquad (3.14)\end{aligned}$$

where

$$\mathrm{K}'_{st_i n_c} = \frac{\rho_w}{\mu_w \mathrm{r}_{st_i}} \sum_{j=1}^{J} \mathrm{K}_{u_j t_i n_c} \int_{\mathrm{V}_j} \mathrm{d}^3 u \left(\frac{\mathrm{d}\sigma(\psi_i, \alpha_0)}{\mathrm{d}\Omega} \right) \frac{1}{\mathrm{r}_{us}^2} \frac{\mathrm{P}_E(\psi_i, \phi_{ic})}{\mathrm{P}_E(\phi_{ic})} \exp\left(\mu_w(\mathrm{r}_{st_i} - \mathrm{r}_{us} - \mathrm{r}_{ut_i})\right)$$

The second order lookup table is the same size as that of the first order table; 25 Mbytes. The additional time required to calculate the second order table, once the first order table has been generated, is 5.75 weeks on a single Sun Sparc10 workstation.

3.2.3 Higher Order Compton Scatter

The technique used for the calculation of the second order Compton scatter distribution can be extended to calculate the third and higher orders as well. This extension is incorporated by writing the complete Compton scatter distribution function as

$$\mathrm{SDF}(\vec{s}, \vec{n}_c) = \mathrm{Q} \sum_{i=1}^{I} \rho_e(\vec{t}_i) \tilde{\mathrm{K}}_{st_i n_c} \exp\left(- \int_{\vec{s}}^{\vec{t}_i} \mu(y) \mathrm{d}y - \int_{\vec{t}_i}^{\vec{n}_c} \mu(y, \theta_{ic}) \mathrm{d}y \right), \quad (3.15)$$

where

$$\tilde{\mathrm{K}} = \sum_{m=0}^{\infty} \mathrm{K}_{st_i n_c}^{(m)} \left[\int_{\vec{s}}^{\vec{t}_i} \mu(y) \mathrm{d}y \right]^m .$$

In this equation, $\mathrm{K}_{st_i n_c}^{(0)} = \mathrm{K}_{st_i n_c}$ (equation (3.8)) and, for m> 0, $\mathrm{K}_{st_i n_c}^{(m)}$ is defined recursively as

$$\mathrm{K}_{st_i n_c}^{(m)} = \frac{\rho_w}{\mu_w \mathrm{r}_{st_i}} \sum_{j=1}^{J} \mathrm{K}_{u_j^m t_i n_c}^{(m-1)} \int_{\mathrm{V}_j} \mathrm{d}^3 u^m \left(\frac{\mathrm{d}\sigma(\psi^m, \alpha_0)}{\mathrm{d}\Omega} \right) \frac{1}{\mathrm{r}_{u^m s}^2} \frac{\mathrm{P}_E(\phi^0, \phi^1, ..., \phi^m)}{\mathrm{P}_E(\phi^0, \phi^1, ..., \phi^{m-1})}$$
$$\exp\left(\mu_w \left(\mathrm{r}_{st_i} - \mathrm{r}_{u^m s} - \mathrm{r}_{u^m t_i} \right) \right)$$

where u^p is the scatter site which is p sites before the last scatter site and ϕ^p is the Compton scatter angle of the photon at site u^p. The angle ϕ^p is measured using the centres of the voxels at u^{p-1}, u^p, and u^{p+1} where $u^{m+1} = \vec{s}$, $u^0 = \vec{t_i}$, and $u^{-1} = \vec{n_c}$.

For isotopes with energies similar to ^{99m}Tc, such as ^{123}I (159keV), higher order Compton scatter does not contribute significantly to the photopeak window data. For example, in the case of a small ^{99m}Tc line source deep in a homogeneous water medium, the

Compton-scattered photons detected in the photopeak window are primarily those which scattered once (87%) or twice (12%) [45, 152] with third and higher orders only contributing 1%. If one wanted to use this method for the calculation of scatter in a lower energy window, such as for down scatter from ^{99m}Tc into the primary ^{201}Tl window (70keV), the higher order scatter events would play a greater role. In this case it would be necessary to reassess the effect of our approximations.

The determination of the distribution for each different order of scatter requires a separate calculation and consequently, significantly increases the total computation time. Because the third and higher order scatter events constitute a very small fraction of the scatter in the photopeak window of ^{99m}Tc, in the comparisons of our calculated results with Monte Carlo and experimental data (Chapters 4 and 5), they are included as a simple rescaling.

Although the fraction of the detected scatter events which are attributable to third and higher order scattering will depend on the source depth and the configuration of the scattering medium, we originally chose to approximate this fraction roughly as a constant (1%) and correct for higher order scatter by scaling the first and second order scatter by 100/99. The complete Compton scatter distribution function is, therefore, written in a simplified form by modifying $\tilde{\mathrm{K}}$ in equation (3.15) as

$$\tilde{\mathrm{K}}_{st_in_c} = \left(\mathrm{K}_{st_in_c} + \mathrm{K}'_{st_in_c}\int_{\vec{s}}^{\vec{t_i}} \mu(y)\mathrm{d}y\right)\frac{100}{99}. \qquad (3.16)$$

This approximation was used for the work done in Chapters 4 and 5.

This approximation was not, however, used in Chapter 6. The shape of the higher order scatter contribution in the photopeak energy window more closely resembles that of the second order scatter distribution alone, rather than the sum of the first and second order distributions [45]. In light of this, for the work done in Chapter 6, the approximation for the higher order scatter contribution was altered. Instead of scaling the sum of the first and second order distributions by 100/99, the first order distribution was left unscaled and the second order scatter distribution was scaled by a factor of 13/12. The work in Chapters 4 and 5 was not re-analyzed using this improved approximation as this would not significantly alter the results and conclusions of those chapters.

3.3 Rayleigh Scatter

The total cross-section for Rayleigh scattering is only a small fraction of that for Compton scattering (1.6). However, because Rayleigh scattering occurs preferentially for low scatter angles and because the collimator has a small but finite acceptance angle, there is a question as to whether the effect of Rayleigh scattering might in fact still be significant.

We briefly investigated this question by considering the case of a point source positioned on the axis of a homogeneous water cylinder with a radius of 10cm. We considered the radial distribution of photons in a projection with and without Rayleigh scattering where the attenuation coefficients are adjusted accordingly. The experimentally verified Monte Carlo code, EGS4 [2, 159], was used to obtain the data.

Our version of EGS4 does not incorporate the effects of Compton-Rayleigh interference [28] on the form factors for coherent scattering. However, for 140 keV photons scattering in water at angles greater than 5^o, the effects of interference do not alter the form factors [160]. In this range, the results of our comparison, Figure 3.9, show that the magnitude of Rayleigh scatter is the same as that for second order Compton scatter. Therefore, for accurate calculations, Rayleigh scattering should be included.

The calculation of the first order Rayleigh scatter is very similar to that of the first order Compton scatter. The probability of detecting a photon is initially expressed as a continuous function of the positions $\vec{s}$, $\vec{t}$, and $\vec{n}$ (Figure 3.1). The discrete Rayleigh scatter distribution function $\mathrm{RDF}_1(\vec{s}, \vec{t_i}, \vec{n}_c)$, is obtained by integrating the continuous

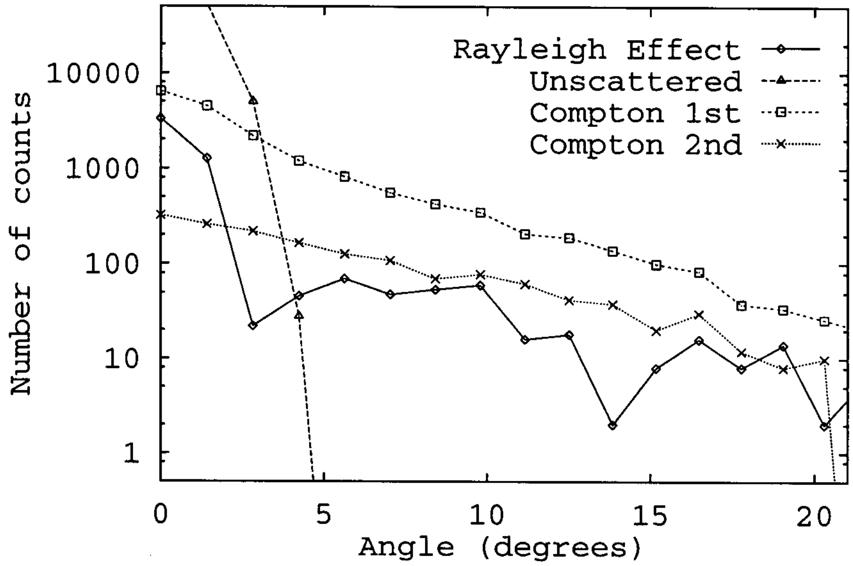

Figure 3.9: The Rayleigh scatter contribution. This figure shows the angular distribution of photons from an on-axis point source in a 10cm radius water cylinder. The *Rayleigh Effect* line is the absolute value of the difference between MC simulations which do not include Rayleigh scattering and those which do. The *Unscattered* and *Compton* lines are for the case where Rayleigh scattering is not included. *Compton 1st* refers to the first order Compton scattering events whereas *Compton 2nd* refers to the second order events. The angle measured is between the detector normal and a line from the source to the detector point being considered.

function $\mathrm{RDF}_1(\vec{s},\vec{t},\vec{n})$ both over the area, A_c, of a pixel on the detector surface centred at $\vec{n}_c$ and also over the volume, V_i, of the voxel centred at $\vec{t}_i$.

$$\begin{aligned}\mathrm{RDF}_1(\vec{s},\vec{t}_i,\vec{n}_c) \;=\; & \mathrm{Q}\int_{\mathrm{A}_c}\mathrm{d}^2n\int_{\mathrm{V}_i}\mathrm{d}^3t\;\rho_{atom}(\vec{t})\;\frac{\mathrm{d}\sigma_{coh}}{\mathrm{d}\Omega}(\theta_R)\;\mathcal{F}(x,\overline{Z})^2 \\ & \mathrm{F}(\xi')\mathrm{P}_E(0)\frac{\cos(\xi')}{4\pi\mathrm{r}_{nt}^2\mathrm{r}_{ts}^2}\exp\left(-\int_{\vec{s}}^{\vec{t}}\mu(y)\mathrm{d}y-\int_{\vec{t}}^{\vec{n}}\mu(y)\mathrm{d}y\right)\end{aligned} \tag{3.17}$$

where

$\rho_{atom}(\vec{t})$ is the atomic density of the material at point $(\vec{t})$,

$\dfrac{\mathrm{d}\sigma_{coh}}{\mathrm{d}\Omega}(\theta_R)$ is the coherent or Rayleigh scattering cross-section (Chapter 1),

θ_R is the Rayleigh scattering angle,

$\mathcal{F}(x,\overline{Z})$ is the form factor for coherent scattering,

x is described by equation 3.9 with θ replaced by θ_R,

$\overline{Z}$ is the effective atomic number of the scattering material,

and the remaining symbols are as defined in equation (3.3).

Equation (3.17) and equation (3.3) are very similar in form to each other and by following the same procedure as described in section 3.2.1, it is possible to rewrite equation (3.17) in the form of equation (3.7) with the factor $\mathrm{K}_{st_in_c}$ for Compton scattering replaced by $\mathrm{K}^R_{st_in_c}$.

$$\mathrm{RDF}_1(\vec{s},\vec{n}_c)=\mathrm{Q}\sum_{i=1}^{I}\rho_{atom}(\vec{t}_i)\,\mathcal{F}(x_{ic},\overline{Z})^2\,\mathrm{K}^R_{st_in_c}\exp\left(-\int_{\vec{s}}^{\vec{t}_i}\mu(y)\mathrm{d}y-\int_{\vec{t}_i}^{\vec{n}_c}\mu(y)\mathrm{d}y\right), \tag{3.18}$$

where x_{ic} is evaluated at $\theta_R=\theta_{ic}$ with θ_{ic} as defined in equation (3.6) and

$$\mathrm{K}^R_{st_in_c}=\int_{\mathrm{A}_c}\mathrm{d}^2n\int_{\mathrm{V}_i}\mathrm{d}^3t\;\frac{\mathrm{d}\sigma_{\mathrm{coh}}}{\mathrm{d}\Omega}(\theta_R)\,\mathrm{F}(\xi')\,\mathrm{P}_E(0)\,\frac{\cos\xi'}{4\pi\mathrm{r}_{ts}^2\mathrm{r}_{nt}^2}. \tag{3.19}$$

Although the Rayleigh scattering contribution can be computed separately using equation (3.18), this increases the total calculation time by approximately 35%. Alternatively, one can avoid this increase in calculation time by including Rayleigh scattering

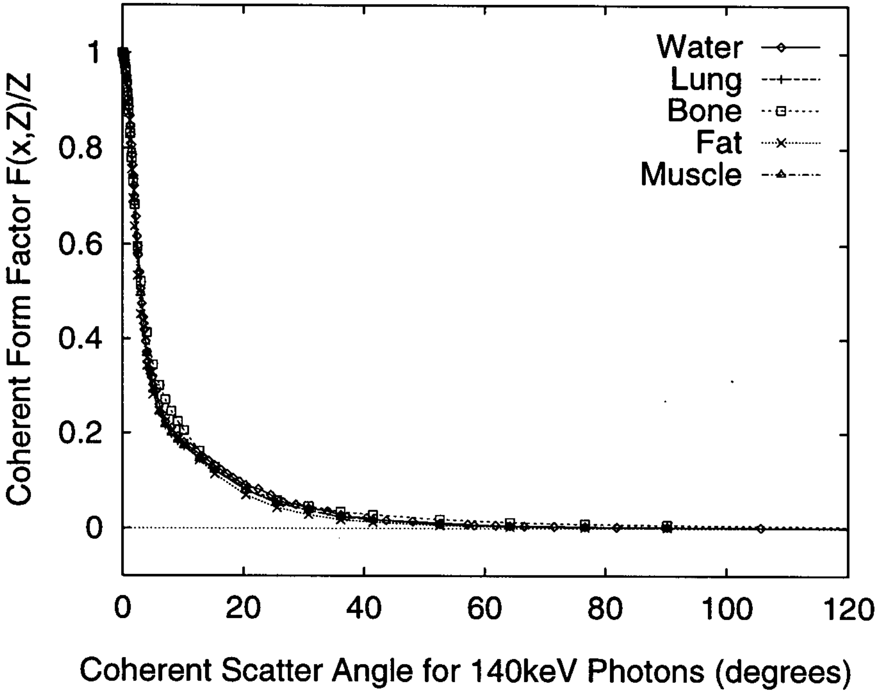

Figure 3.10: Coherent (Rayleigh) form factors divided by the effective atomic number for biological materials at 140keV (^{99m}Tc). Data is from [27].

through a modification of the precalculated lookup table for first order Compton scatter. This is accomplished in the following manner. First, it is noted that

$$\rho_{atom}(\vec{t}_i)\,\mathcal{F}(x_{ic},\overline{Z})^2 = \frac{\rho_e(\vec{t}_i)}{\overline{Z}}\left(\frac{\mathcal{F}(x_{ic},\overline{Z})}{\overline{Z}}\right)^2\overline{Z}^2 \tag{3.20}$$

Second, we observe that $\mathcal{F}(x_{ic},\overline{Z})/\overline{Z}$ has very similar functional behavior for most biological materials near those energies used in SPECT imaging (Figure 3.10). Our observations are based on the form factors tabulated in [27].

The material which deviates most from the water line in Figure 3.10 is bone. The greatest difference between bone and water occurs at θ_R equal to 7^o where it deviates by 18%. The average percent deviation, weighted by the detection probability, would be considerably less. Finally, we note that the ratio of the effective atomic numbers for water and other biological materials is also close to 1 for many different materials (Table 3.1). It does, however, deviate from 1 by 28% for bone and by 16% for fat.

Table 3.1: Effective Atomic Number Ratios. The effective atomic numbers are the weighted average of the atomic numbers of the constituent atoms of the material. The material compositions used are from [24]. The effective atomic number for different biological materials is compared to that of water and the percent difference is given.

Material	$\overline{Z}$	$\frac{\overline{Z}}{\overline{Z}_{\mathrm{H2O}}}$	Difference (%)
Water	7.22	1.00	0.0
Lung	7.12	0.99	1.4
Muscle	7.10	0.98	1.7
Bone	9.25	1.28	28.1
Fat	6.06	0.84	16.0

With these observations in mind, we first approximate

$$\left(\frac{\mathcal{F}(x_{ic},\overline{Z}_{\mathrm{Mat}})}{\overline{Z}_{\mathrm{Mat}}}\right)^2 \overline{Z}_{\mathrm{Mat}} \approx \left(\frac{\mathcal{F}(x_{ic},\overline{Z}_{\mathrm{H2O}})}{\overline{Z}_{\mathrm{H2O}}}\right)^2 \overline{Z}_{\mathrm{H2O}}, \tag{3.21}$$

where $\overline{Z}_{\mathrm{Mat}}$ and $\overline{Z}_{\mathrm{H2O}}$ are the effective atomic numbers of the scattering material and of water, respectively, and where the form factors for water include the effects of coherent-incoherent interference.

Secondly, we approximate the attenuation coefficient in equation (3.17) as the attenuation coefficient for the same material but evaluated at an energy equal to the energy of a photon which has Compton scattered through θ_R degrees. In other words, the second component of the attenuation factor in equation (3.17) is approximated as

$$\exp\left(-\int_{\vec{t}_i}^{\vec{n}_c} \mu(y)\mathrm{d}y\right) \approx \exp\left(-\int_{\vec{t}_i}^{\vec{n}_c} \mu(y,\theta_R)\mathrm{d}y\right). \tag{3.22}$$

This introduces an error as the energy of a Rayleigh scattered photon does not decrease when it scatters. This error could be quite large, for example, in water, $\mu(140\mathrm{keV}) = 0.154$ whereas $\mu(126\mathrm{keV}) = 0.160$, a difference of 3.9%. However, for ^{99m}Tc photons, 85% of Rayleigh scattering events occur at angles less than 15^o. A 140keV photon which has

Compton scattered through 15^o has an energy of 138.7keV which corresponds to an error in the attenuation coefficient of only 0.3%.

With these approximations, RDF_1 can be added to SDF_1 creating a single lookup table for both first order Compton and first order Rayleigh scattering. While this does increase the calculation time for the lookup tables themselves, it causes no decrease in the speed of the real-time evaluation of the photon distributions.

For our comparisons with the Monte Carlo simulators in Chapter 4, we do not include Rayleigh scattering because the SIMSET code does not yet model it. For these comparisons we adjusted our total attenuation coefficients to also not include coherent scattering. For the comparison of the results of our analytical photon (AP) calculation with those acquired from phantom experiments (Chapter 5), Rayleigh scattering has been included, using the approximations discussed, as a modification of the first order Compton scatter lookup table.

3.4 Extended Sources

In sections 3.1 – 3.3 we have concentrated on a description of the AP calculation for point sources. Although useful for evaluating against Monte Carlo simulations, point sources are not realistic and for clinical situations, one needs to be able to generate projections for larger source distributions.

As we indicated at the start of this chapter, a finite source can be described as a sum (integral) of point sources. The simplest way to extend our calculations to larger sources is, therefore, to express the larger source as a discrete sum of small sources (*eg* voxelize the source). The total photon distribution function is then the sum of point distributions where each voxel in the source distribution is treated as a separate point-like source.

This approach, while easy to implement, is quite slow to execute because of the possibly huge number of source voxels in a distributed source. For example, a 10cm radius cylinder 20cm long which is filled with activity (Case 4, Chapter 4) contains 55000 active voxels. Independent computation of all 55000 photon distributions would involve many redundant calculations such as the calculation of the attenuation along photon paths. Code which avoided these redundancies would greatly improve the calculation time for larger sources. The development of this extended source code is beyond the scope of this thesis, but is a topic for future investigation. In a similar manner, symmetries in the attenuating medium can also be used to greatly reduce total calculation times.

3.5 Conclusions

The development of this method of computing photon distributions in SPECT projections has required a large number of approximations. Though each of these approximations is physically reasonable and individually constitutes only a small inaccuracy in the calculation, the cumulative sum of these approximations could be devastating. One of the goals of this project was to create an accurate method of estimating scatter and, therefore, it is necessary to check the accuracy of our calculated projections against other, more established techniques. For this reason, the next two chapters are devoted to a comparison of the accuracy of our technique with respect to Monte Carlo simulation and phantom experiments.

The second goal of this work was to improve on the calculation times of currently available similar approaches. A time comparison has, therefore, been done between our codes and Monte Carlo simulations and also between our codes and the analytical approach of Riauka and Gortel [149]. The results of both of these comparisons are provided in Chapter 4, section 4.4.

CHAPTER 4

MONTE CARLO VALIDATION

Monte Carlo (MC) simulation is a well established technique in medical imaging research. It is possible with this technique to completely control the modelling of the interactions of the photons with the attenuating/scattering medium. Monte Carlo also permits the classification of photons as to the number of times they have scattered (and what type(s) of scatter they have undergone) before being detected and in particular allows a separation of scattered and unscattered events. Unlike with phantom experiments, it is possible with MC simulations to know the correct result (within the context of the chosen mathematical model). Consequently, MC is a very useful and stable platform on which to test different methods of scatter estimation and correction.

We decided to use Monte Carlo to test the accuracy of our method of determining photon distributions [154]. It was used to assist in the debugging of the algorithms and also to assess the cumulative effect of our various approximations (discussed in Chapter 3) on the resultant projections. We also compared the time needed to compute projections using our technique to the time required to generate projections of "equivalent" accuracy using MC simulation.

This chapter consists of five sections. First, the two Monte Carlo simulators which we used for our tests are briefly described. This is followed by a description of the simulations we performed. The third section is devoted to a study of the accuracy of our technique when compared to the two MC simulators and the fourth compares the required calculation times for the different methods. The fifth and final section summarizes the results of our comparisons.

4.1 The Monte Carlo Simulators

In our Monte Carlo validation studies we used two different simulators, the first of which was the EGS4 (Electron Gamma Shower 4) MC code [2]. This simulator was chosen because it has been in use for many years, has been verified experimentally [159], and because it uses a simple algorithm for tracking photon histories (no variance reduction or vectorization). The absence of variance reduction makes it easier to determine the statistical error in the projections and consequently easier to determine if our results lie within this error. The EGS4 code faithfully reproduces the magnitude of the acquired projection for a given number of photons emitted from the source and thus allows us to check the absolute quantitative accuracy of our calculation.

The second simulator we chose was the variance reduced code called the SIMSET PHG (Simulator from Seattle Photon History Generator) [3]. It was chosen in order to compare our approach to a state-of-the-art MC simulator. The variance reduction options in SIMSET, forced detection, forced non-absorption and stratification (with an initial stratification run of 50000 photons), were all used. The SIMSET projections were each scaled uniformly to contain the same total number of counts as the corresponding result from EGS4. It was hoped that the SIMSET code would provide an accurate estimate of the current speed capabilities of Monte Carlo simulators.

It is possible to further improve upon the calculation times of variance reduced codes by using vectorization. A vector processing machine performs the same operation on different pieces of data simultaneously using multiple processors. Vectorization can improve MC computation times by a factor of 2–4 [138, 139]. Vectorization is a general technique, however, that can be applied to any computer algorithm. Our code, like that of the Monte Carlo simulators, involves a large number of independent and/or repetitive operations and would thus also benefit greatly from vectorization. As we do not have ready access to vector processing machines, we have not compared vectorized forms of our code and MC simulators.

In both the analytical photon (AP) calculation and the MC codes, the energy response and intrinsic spatial resolution of the camera's detector as well as the geometry of its collimator were chosen to be the same. The energy response of the camera was modelled as a Gaussian with a FWHM equal to 10% of the incident photon energy. The intrinsic spatial resolution was taken to be a Gaussian function with a FWHM of 0.5cm. The collimator was modelled as a parallel hole collimator with no septal penetration, a minimum septal thickness of 0.015cm, circular holes of diameter 0.14cm, and a collimator thickness of 3.28cm, yielding an acceptance angle of 2.44 degrees. The efficiency of the detector crystal was assumed to be 1.0.

The EGS4 code uses an analytical description of the scattering body. In an effort to obtain results which were equivalent to those produced by EGS4's analytical description of the phantom, we used a relatively fine voxelization (129x129x129 0.25x0.25x0.25cm^3) in the SIMSET code. The efficiency of SIMSET could be increased by a factor of two by using a much coarser voxelization but this noticeably reduces the accuracy of the resultant projections. Our code uses a 64x64x64 (0.5x0.5x0.5cm^3) voxelization. All three techniques generated 64x64 projections with 0.5x0.5cm pixels.

Rayleigh scattering was not simulated by either the EGS4 or AP codes in order to compare them with the SIMSET code, which does not model Rayleigh scattering.

Our code used the electron density and linear attenuation coefficients from Picard *et.*

al. [158].

4.2 The Simulations

It is impossible with any finite number of phantom configurations to completely cover all possible situations which might be encountered in a clinical setting. However, with our selection of four different configurations, we have tried to consider a representative sample of the different types of situations one might encounter.

In all of the cases modelled, the scattering/attenuation media were 20cm in length and centred within the camera's field of view. The cross-section of each medium, perpendicular to the axis of rotation of the camera (z-axis), was invariant along the length of the medium. The source, in cases 1 and 2, was located halfway along the z-axis of the medium, that is 10cm from either end. The four cases considered were:

1. A homogeneous water-filled cylinder 10cm in radius centred on the camera's axis of rotation. The source was a point source located in two different positions: (i) on the axis of rotation, and (ii) 9cm off of the axis of rotation along the y-axis of the medium.

2. A non-homogeneous small chest model (as shown in Figure 4.1) whose cross-section consisted of two lungs and a spinal column in a water-filled ellipse. Each lung had semi-major and -minor axes of 5cm and 4.5cm respectively. The spinal column was circular with a radius of 1.25cm. The enclosing water ellipse had a semi-major axis of 15cm and a semi-minor axis of 11cm. The mass density of the lungs, spinal column, and water body were 0.26, 1.65, and 1.00 g/cm^3 respectively. A point source was positioned 2.5cm away from the axis of rotation along the positive y-axis of the model. The origin of the chest model and the axis of rotation of the camera coincided.

3. The chest model with an extended source of activity (Figure 4.1). The source consisted of an hollow cylinder 8.0cm long with an inner radius of 1.5cm and an outer radius of 2.0cm (roughly modelling a heart wall). Its axis of rotation was parallel to that of the camera and displaced by 1.0cm along the positive y-axis. The source was positioned centrally and thus began 6.0cm above the lower end of scattering medium. This source corresponded to 800 active voxels.

4. A 10cm radius water cylinder filled uniformly with activity. This source corresponded to 55000 active voxels.

The first phantom configuration, case 1(i), was chosen as a simple first example. The symmetry of the source position and the attenuating medium allowed highly accurate

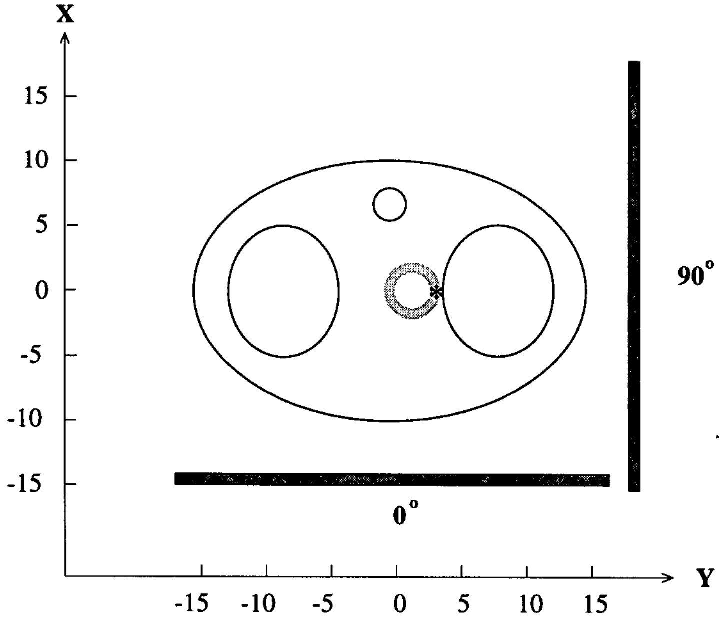

Figure 4.1: A cross-section of the chest model used for cases 2 and 3. The model consists of a bone cylinder (a spinal column) and two elliptical lung cylinders within an elliptical water cylinder. The asterisk indicates the point source location for case 2 and the shaded ring indicates the source position for case 3. The solid bars show the detector positions for 0^o and 90^o projections. The axes are marked in centimeters

Monte Carlo simulations to be obtained fairly rapidly. This was possible because the projections from the MC simulations were all equivalent and could thus be summed together, improving count statistics. The high number of counts attained for this case allowed an accurate comparison of the separate components of the photon distribution and especially of the second order scatter which consists of much lower count levels than the unscattered or first order Compton scatter distributions.

Phantom configuration case 1(ii) was chosen to study the effect of positioning the source in an extremely asymmetric position and also to consider the effect of having a source very close to the edge of the attenuating body.

The second configuration, case 2, was chosen to ascertain the effects of inhomogeneities on the accuracy of the projection. The chest represents a commonly imaged region in the body which contains a large degree of variation in the tissue density distribution. The point source was positioned off-centre to reduce the symmetry of the situation and was placed near the lung wall to study the influence of boundaries on our accuracy.

The third case was chosen to represent a more realistic source distribution in the non-homogeneous medium. This configuration loosely models cardiac studies where the myocardial walls are the main source of radiation in the image.

The last configuration, case 4, was chosen in order to consider a very extended source distribution. Often in medical imaging, the background is active or one is interested in the cold spots in an extended hot region. It is necessary, therefore, to be able to simulate warm background regions. A simple scattering medium was chosen to reduce the necessary simulation times.

In cases 1(ii), 2, and 3, the projections are labelled according to the angle which the collimator normal makes with the positive x-axis of the scattering medium. The positive y-axis of the phantom points towards the 90^o projection. Examples of the positions which

Table 4.1: Effective number of simulated photons and estimated statistical error in the peak pixel of the MC projections. The quoted accuracies were attained by summing together all 64 acquired projections from cases 1 and 4 and the central seven transaxial slices in cases 3 and 4 (as discussed in section 4.3).

Case	Photons Simulated		Error in Peak	
	(counts x10^6)		(%)	
	EGS4	SIMSET	EGS4	SIMSET
1(i)	12800	1280	0.42	0.50
1(ii) 0^o	1000	120	0.97	1.71
90^o			0.58	1.18
2 0^o	1270	120	1.42	1.60
45^o			1.06	1.56
90^o			0.91	1.47
180^o			1.42	1.61
3 0^o	17605	1680	1.82	1.79
45^o			1.42	1.81
90^o			1.45	1.82
180^o			1.85	1.81
4	324800	26880	1.05	1.46

the detector would be at for the acquisition of the 0^o and 90^o projections are shown in Figure 4.1.

The MC simulations were run until the estimated statistical error in the peak value (the single pixel containing the highest count) of each projection was less than 2%. The accuracies in the projections were comparable for the two MC simulations as is shown in Table 4.1.

4.3 Results: The Monte Carlo Accuracy Comparison

An example of the type of data that was obtained in this study is shown in Figure 4.2. The images shown in this figure are those for 180^o projection of the point source in the chest medium (case 2). The *EGS4* and *AP* images are the projections obtained using the

EGS4 MC and AP codes respectively. The *Difference* image shows the absolute value of the difference of the *EGS4* and *AP* projections and in *Normalized Difference*, this difference is divided by the standard deviation of the counts in the corresponding pixel of the EGS4 image. If the differences between the two images is due to random statistical errors in the MC, the *Normalized Difference* image would appear simply as white noise. The lack of structure in the *Normalized Difference* image indicates that there is a good agreement throughout between our results and those of the Monte Carlo simulation.

The two dimensional projections such as are shown in Figure 4.2 are difficult to read and so, for the remainder to this work, we will present horizontal profiles through the peak of the projections. For example, the line A in Figure 4.2 indicates the position of one such profile through the *EGS4* and *AP* projections. The profiles for case 2 are shown in Figure 4.6.

Examples of profiles through the projections which were acquired during the simulations of the four cases outlined in section 4.2 are presented in Figures 4.3–4.8. In these figures, the EGS4 results are depicted by dashed lines marked by diamonds, the SIMSET results are shown using dotted lines, and the AP results are presented using solid lines. Tables 4.2–4.3 contain the results of our quantitative comparison of the projections. In all of the figures and tables, EGS4 refers to the EGS4 MC, SIM to the SIMSET MC, and AP to our analytical photon distribution calculations.

The profiles shown in Figures 4.3–4.6 are transaxial (perpendicular to the z-axis) and correspond to the sum of seven adjacent rows in the projection centred on the peak of the photon distribution, that is over a span of 3.5cm in the z-direction. The summation was done in order to reduce the statistical noise in the off-peak regions of the Monte Carlo results.

Figure 4.3 shows semi-logarithmic transaxial profiles through the projection from case 1. In this figure we have separated the unscattered and the first and second order

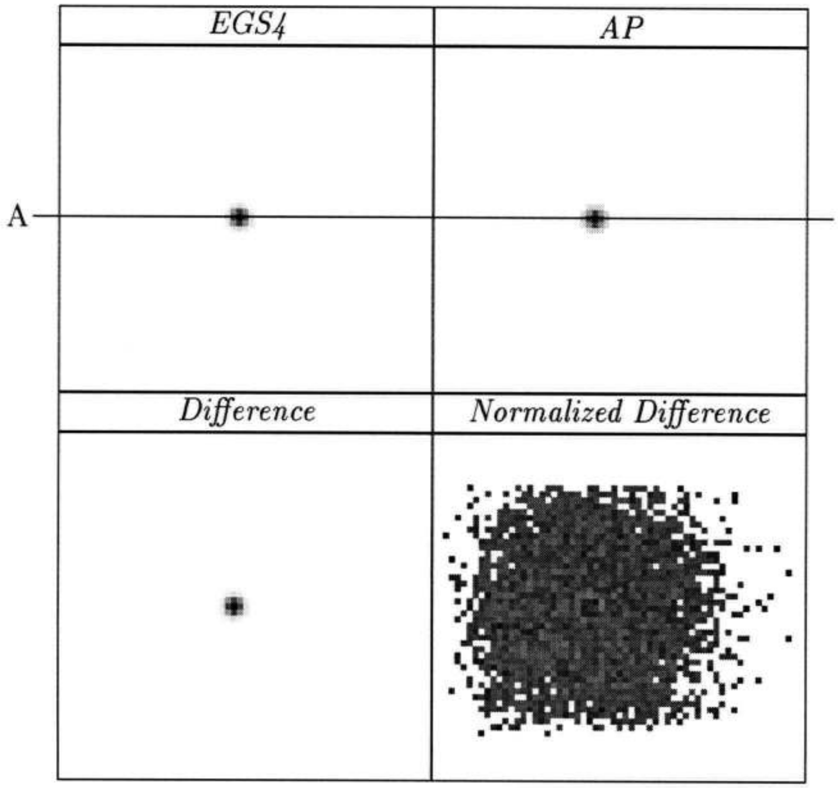

Figure 4.2: The MC (*EGS4*) and AP (*AP*) calculated projections for the chest model (180° projection) as well as the corresponding absolute difference (*Difference*) and normalized difference (*Normalized Difference*) images. The lack of structure in the normalized difference image indicates that there are no systematic deviations between our results and the MC ones. It also indicates that the differences between the EGS4 and AP images are consistent with the differences one would expect from the statistical fluctuations in the MC data. The line A indicates the position of a horizontal profile through the peak of the MC and AP projections.

Compton scattered photons distributions from one another. These three distributions are calculated with three distinct computations (equations (3.1), (3.7), and (3.14) from Chapter 3). The distributions are compared separately in order to determine if each is being modelled accurately and that there are not fortuitous error cancellations when the three distributions are summed together.

Figure 4.3 is shown with a logarithmic scale on the vertical axis. This is done to show the scatter distribution more clearly. The same figure can be shown on a linear scale (Figure 4.4) but the structure of the scattered photon distribution is not as readily apparent as with the semi-log plots. For this reason, the remainder of the figures in this thesis are shown with a logarithmic scale on the vertical axis.

Additionally, in all of the remaining figures, only the total photon distribution is shown. The AP distribution corresponds to the sum of the results of the unscattered,

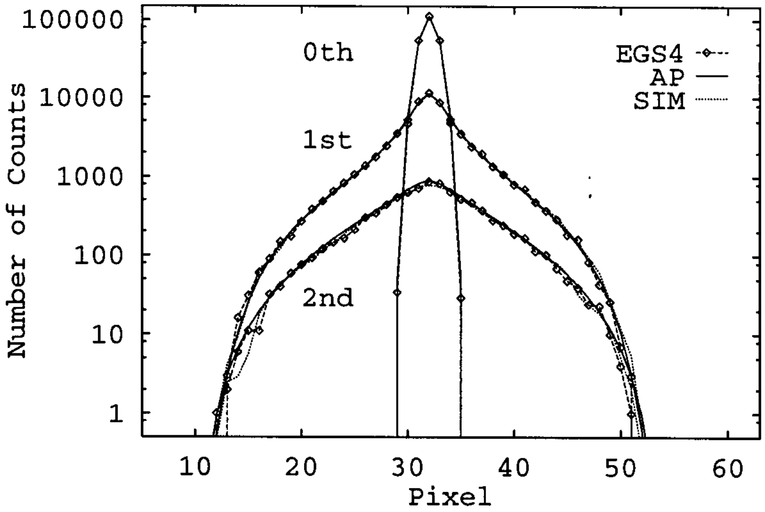

Figure 4.3: Profiles for an on-axis point source in a water cylinder, case 1(i). The unscattered distribution as well as the first and second order scatter distributions are separated and labelled 0th, 1st, and 2nd respectively. *EGS4* refers to the EGS4 MC simulation results, *SIM* to the SIMSET MC simulation results, and *AP* to our results.

first order Compton scattered, and second order Compton scattered AP calculations, with higher order scatter being included as a linear scaling of the first and second order scatter distributions. The EGS4 and SIMSET distributions include unscattered photons and the first five orders of Compton scatter.

Figures 4.5–4.8 show transaxial profiles through phantoms (1(ii))–(4). As shown by these figures, our method matches the Monte Carlo results very well in all cases. The profiles shown for the extended sources (Figures 4.7 and 4.8) were generated using an interpolation modification to our method which will be discussed in section 4.4.2.

Table 4.2 shows the percent differences between the total number of photons in the projections as determined by EGS4 and by our method. Because there are no arbitrary scaling factors in the AP technique, these numbers give an indication of its global quantitative accuracy. Also shown in this table are the average percent differences between

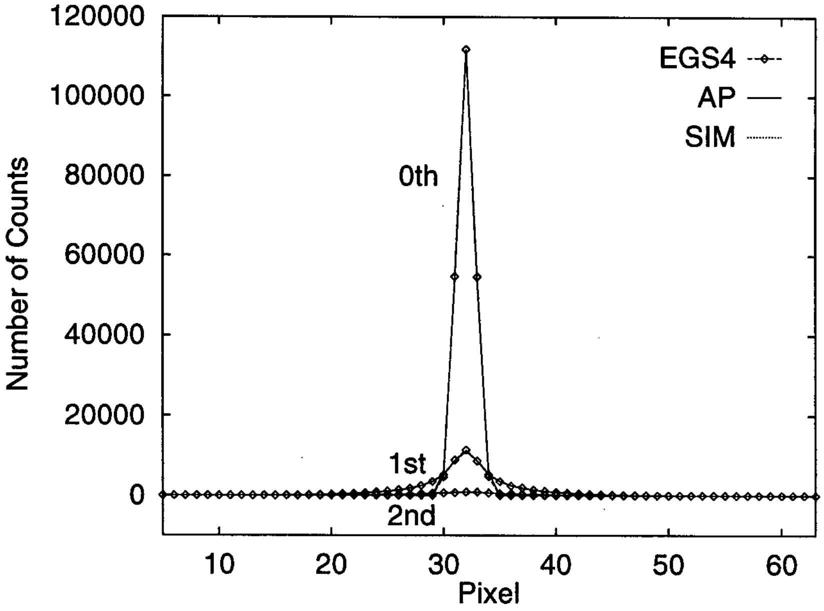

Figure 4.4: On-axis point source in a water cylinder with a linear vertical scale. This figure is identical to Figure 4.3 except that the vertical scale is linear instead of logarithmic.

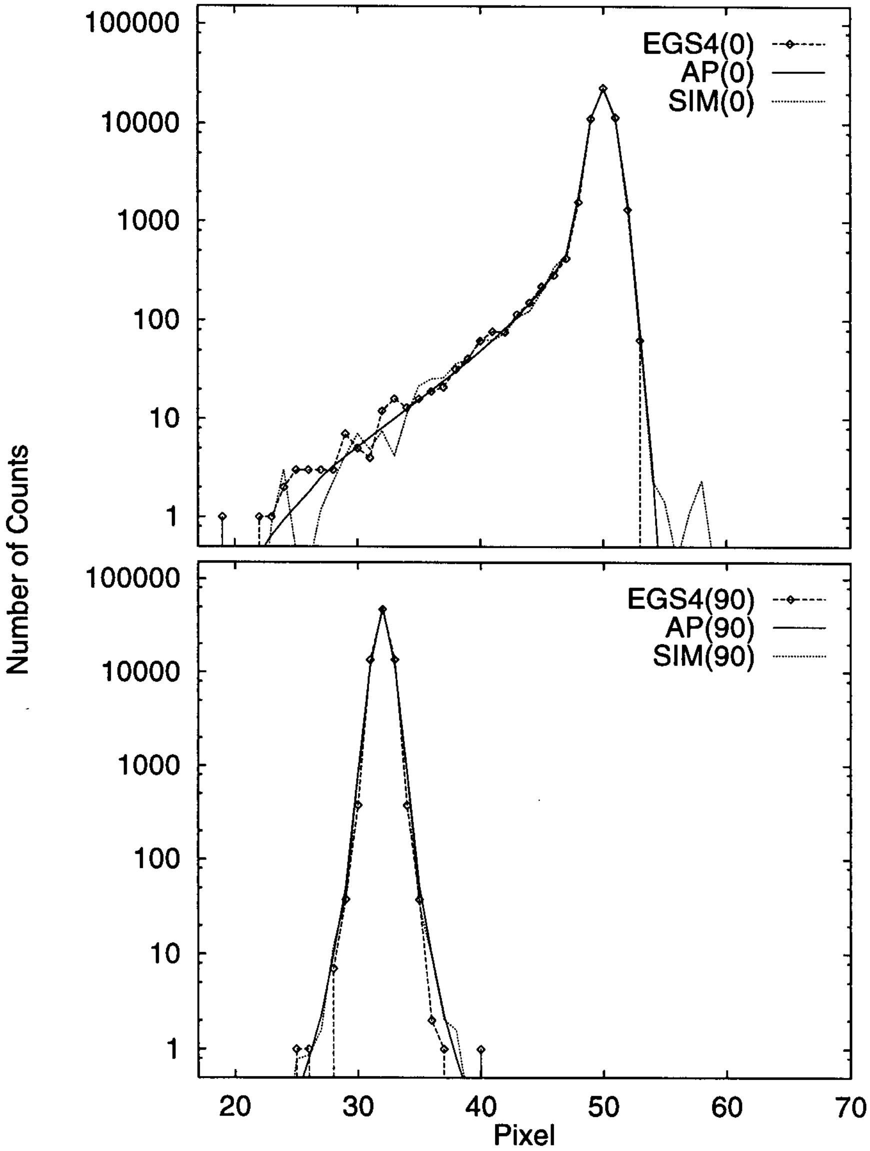

Figure 4.5: The 0^o and 90^o profiles for a point source 9cm along the positive y-axis in a 10cm radius water cylinder, case 1(ii). The number in brackets after the legend label indicates the projection angle. For example, *EGS4(0)* corresponds to the EGS4 MC simulation results for the 0^o projection.

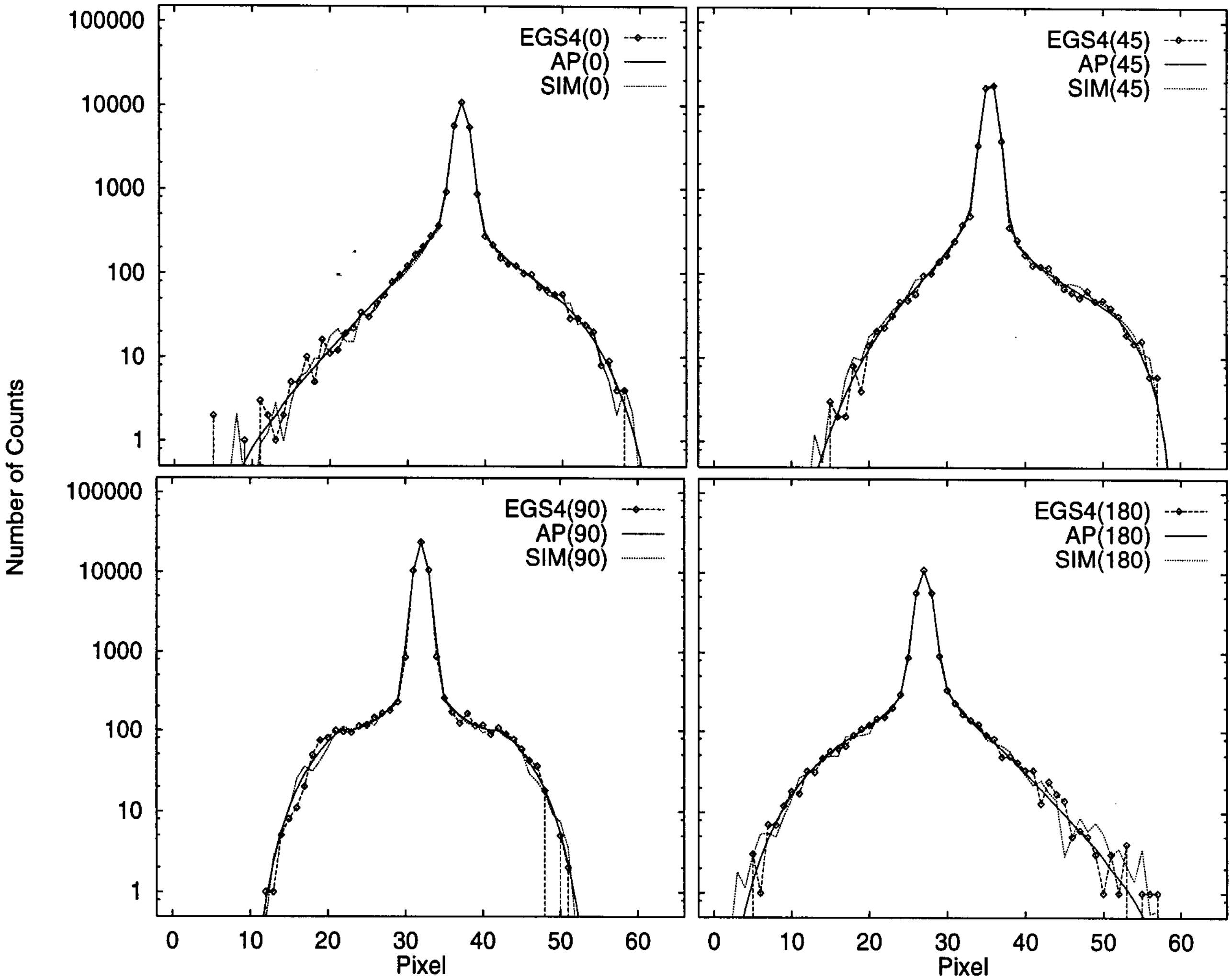

Figure 4.6: Transaxial profiles for a point source 2.5cm off axis in the thoracic phantom, case 2. Profiles are shown for projections at 0^o, 45^o, 90^o, and 180^o.

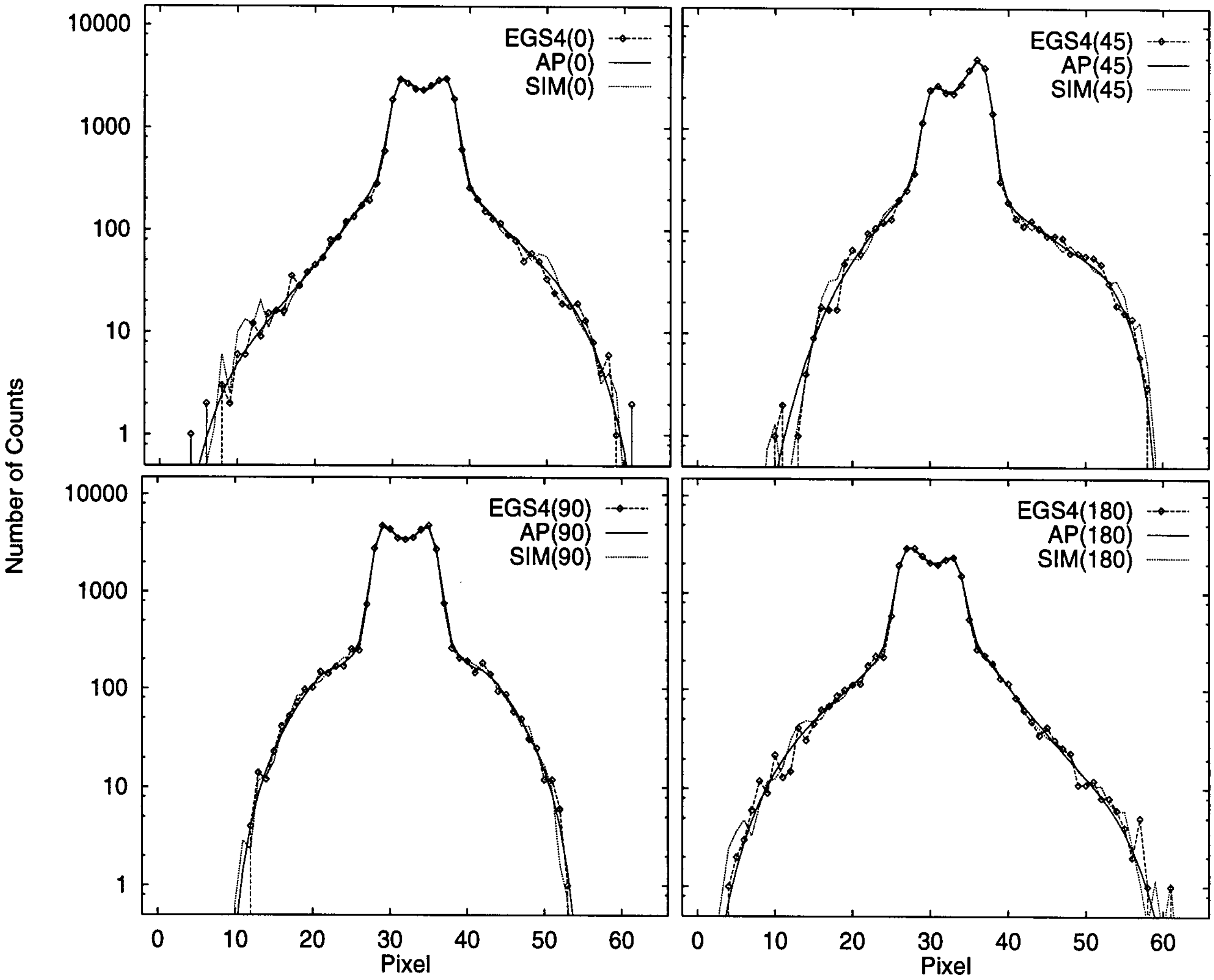

Figure 4.7: Transaxial profiles for a hollow cylindrical source centred 1.0cm off axis (y-direction) in the thoracic phantom, case 3. Profiles are shown for projections at 0^o, 45^o, 90^o, and 180^o.

the results from the three different techniques. This average is taken over all of those pixels for which the MC codes had an estimated statistical error of less than 2%. The AP results agree as well with the two MC results as the MC results agree with each other at the two percent error level. Although the average percentage difference between our technique and the Monte Carlo simulators did become quite large in some instances (for example, the 12.6% error between EGS4 and AP for case 1(ii) (90°)), the error between the two Monte Carlo codes is equally large. These differences are likely due to different approaches to modelling the photon transport and describing the attenuating medium (pixelization versus analytical descriptions). Which Monte Carlo code better approximates reality can only be determined through experiments and this is one of the reasons we have also performed experimental validation of the accuracy of our method (Chapter 5).

The profiles in Figures 4.3–4.8 demonstrate the match between our method and the

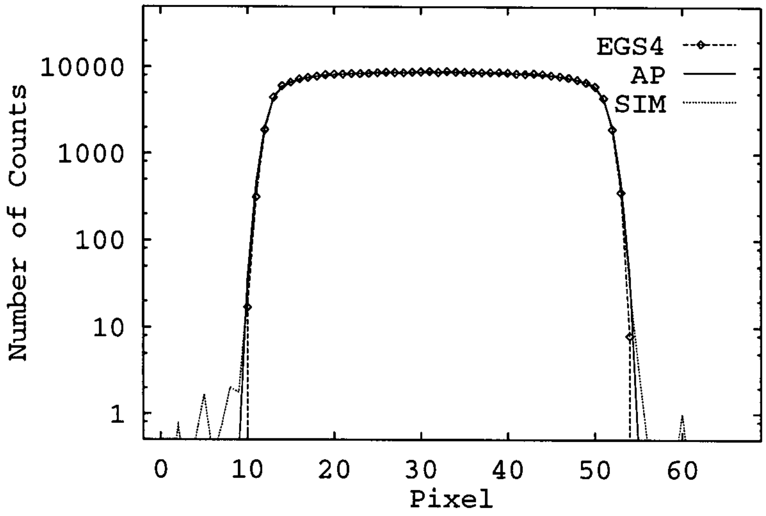

Figure 4.8: A transaxial profile through the centre of a 10cm radius water cylinder filled uniformly with activity, case 4.

Table 4.2: Percentage Differences between Simulations. *Total Diff.* refers to the percentage difference, between EGS4 and our method, in the total number of photon counts in the projection. *Average percentage difference* is the absolute percentage difference on a pixel by pixel basis averaged over all pixels in which the EGS4 values have an estimated statistical error less than two percent.

Case	Total	Average Percentage Difference		
	Diff.	EGS4-AP	SIM-AP	EGS4-SIM
	(%)	(%)	(%)	(%)
1(i)	-0.03	6.12	0.68	5.64
1(ii) 0°	0.16	2.26	7.78	8.59
90°	0.43	12.60	0.74	11.95
2 0°	-0.16	1.75	0.59	1.90
45°	2.18	1.69	3.25	3.16
90°	-0.01	3.24	0.61	3.14
180°	-0.54	2.84	1.11	2.83
3 0°	1.13	1.95	2.45	3.21
45°	0.65	2.93	1.88	2.86
90°	-0.66	2.49	2.39	2.25
180°	2.14	1.45	2.55	3.46
4	-0.75	1.27	1.31	1.44

Monte Carlo results on a narrow section of the projection. They do not, however, give a quantitative value of the goodness of fit over the entire projection. A better indicator of this is the normalized mean squared difference (NMSD) between the projections given by

$$\mathrm{NMSD} \;=\; \frac{1}{\mathrm{N}}\sum_{i=1}^{\mathrm{N}}\left(\frac{\mathrm{A}_i - \mathrm{B}_i}{\sigma_i}\right)^2 \tag{4.1}$$

where A_i is the number of counts recorded in the i^{th} pixel of an image, B_i is the number of counts in the corresponding pixel in the second image, and N is the total number of pixels used. The error σ_i is given by the statistical error in the Monte Carlo when either EGS4 or SIMSET is being compared to the AP calculation, or

$$\sigma_i \;=\; \sqrt{\sigma^2_{EGS4,i} + \sigma^2_{SIM,i}}$$

Table 4.3: Normalized mean squared differences between the simulations as defined in equation 4.1. The last column, *EGS4-SIM*†, is the NMSD value for the difference between the two MC simulators calculated with $\sigma_i = \sigma_{EGS4}$.

Case	Normalized Mean Squared Difference			
	EGS4-AP	SIM-AP	EGS4-SIM	EGS4-SIM†
1(i)	1.60	0.46	0.81	2.21
1(ii) 0°	1.26	0.46	0.50	2.87
90°	44.98	0.66	8.98	39.17
2 0°	1.07	0.45	0.53	1.94
45°	1.54	0.42	0.64	2.48
90°	1.84	0.42	0.76	2.47
180°	1.33	0.49	0.58	2.38
3 0°	1.51	1.05	0.78	2.47
45°	1.56	0.81	0.76	2.26
90°	1.56	1.25	0.73	2.02
180°	1.42	1.01	0.70	1.92
4	5.62	1.25	0.92	3.21

when the two MC projections are being compared to each other. Table 4.3 shows the NMSD values for the four different phantom configurations. The NMSD values computed for the AP results assume no error in the AP values. To aid in comparing the values, the column, *EGS4-SIM*†, has been added to Table 4.3 which gives the NMSD value for the difference between the two Monte Carlo simulators with σ_i in equation (4.1) equal to just σ_{EGS4}. This is equivalent to assuming no error in the SIMSET results and allows direct comparison of the values in the last column to those in the second (*EGS4-AP*). For reference, a random (Gaussian) distribution of differences would yield an NMSD value of 1.

In the cases simulated, there are often large portions of the projections which contain no detected counts either in the AP calculated or the MC simulated results. In these regions there is no data to compare and so these pixels are not included in the NMSD estimation. In areas of the projection which have low numbers of counts, there are often pixels for which the MC simulation produced no counts but the AP calculation gave a non-zero result. For these situations, the error, $\sigma_{EGS4,i}$ is assumed to be one. Finally, in the case of the SIMSET simulation, the variance in a given pixel is sometimes inconsistent with the variance in neighboring pixels. When this occurred, the pixel in question was not included in the NMSD calculation.

In almost all cases, the NMSD values indicate good agreement over the entire projection (within the statistical error of the Monte Carlo simulations) between the AP and MC codes. It may be noted that there is a trend towards better agreement with the SIMSET results than with the EGS4 results. We believe this is due to the nature of the modelling of the attenuation/ scattering media. Both the SIMSET and AP calculations use a voxelized description of the attenuating media whereas the EGS4 MC uses an analytical representation. The effects of voxelization will be most noticeable at the boundaries of objects and this, we believe, accounts for the NMSD value of 5.62 between EGS4 and the AP results in case 4. This value is caused by errors at the edges of the cylinder in the projection. If the edge pixels are not included in the calculation of the NMSD value, it drops to 1.76, a value much more in line with those seen in the other projections. The difference between the EGS4 and SIMSET projections for case 4 is similarly more pronounced at the edges of the cylinder, though the error is not as great because we used a finer voxelization with SIMSET (each SIMSET voxel was one eighth the volume of an AP voxel). The NMSD value between the two MC results drops from 0.92 to 0.73 if the edge region is omitted from the NMSD calculation. It is possible to use a finer voxelization with the AP calculation; however, this causes a significant increase in computation time.

We believe that this voxelization effect is also the cause of the large difference between the SIMSET and AP results when compared to the EGS4 results in case 1(ii)-90°.

4.4 Results: Time Comparisons

Our investigation of the relative speed of the AP calculation consists of two components. The first is a comparison with a similar analytical technique. This technique was developed by Riauka and Gortel [149] and, as mentioned previously (Chapter 2, section 2.4), computes the photon distribution by direct numerical integration of the multi-dimensional integrals describing the photon detection probabilities. The second component is a comparison of the times required for our calculations as compared with those of the two Monte Carlo simulators used in the accuracy comparison of section 4.3.

4.4.1 The Analytical Technique of Riauka and Gortel

The times required by Riauka and Gortel for calculating photon distributions increase dramatically when second order scatter distributions are included. Our technique, through the use of look-up tables, generates photon distributions containing second order Compton scatter events 40–60 times faster.

A comparison of the two methods is given in Table 4.4. The times quoted in this table correspond to the calculation of a single 64x64 projection on a Sun Sparc2 workstation with 48Mbytes internal memory. The times to calculate the first order Compton scattered photon distributions as well as the first plus second order distributions are both provided. The times given for Riauka and Gortel's method (R.G.) are from the doctoral thesis of Riauka [150]. The results using our approach are listed in the AP columns of the table. Projections in this table are calculated for a point source in a 11.2cm radius water cylinder. The front collimator face is located 13.2cm from the rotational axis of the cylinder. Here, configuration A is for the point source positioned on the axis of rotation of the cylinder, configurations B and C refer to the point source 5.6cm away from the axis of rotation in the positive x- and positive y-directions respectively (where the positive

Table 4.4: Time comparison of Riauka and Gortel's method to our method. Times are for the calculation, on a Sun Sparc2, of a single 64x64 projection including first or first and second order scatter.

Configuration	R.G. 1st order (sec)	R.G. order 1+2 (sec)	AP 1st order (sec)	AP order 1+2 (sec)
A	49.85	5452.24	49.2	89.0
B	24.36	1774.18	21.2	43.0
C	50.50	5273.92	48.8	89.9

x-axis is again directed towards the detector). The pixel size used in this simulation was 0.6x0.6cm.

4.4.2 Monte Carlo Simulators

In comparing the two different MC simulators with our method, we considered the time needed to generate a single 64x64 projection by each of the different techniques using a Sun Sparc10 workstation with 32Mbytes internal memory. A summary of the calculation times is given in Table 4.5. The times quoted for the Monte Carlo simulations are the times required to generate the projections to an accuracy of 1.5% in the peak value of the projection. This corresponds approximately to the accuracy attained in the majority of the MC simulations. Because the accuracy in each projection was not exactly 1.5%, the times for each projection have been adjusted accordingly. Also, although SIMSET cannot generate a single projection, a variance reduced code could, in principle, force all of the photons to be detected in just one projection. For this reason we determined the time required to generate 64 projections and then quote this number divided by 64 to give the time for only one projection. This procedure was applied to the times of both EGS4 and SIMSET.

Table 4.5: Summary of the times required, on a Sun Sparc10 workstation, to generate a single 64x64 projection. Monte Carlo times are corrected to correspond to an accuracy of 1.5% in the peak of the projection.

Case	AP (min)	EGS4 (min)	SIM (min)	$\left(\frac{\mathrm{EGS4}}{\mathrm{AP}}\right)$	$\left(\frac{\mathrm{SIM}}{\mathrm{AP}}\right)$
1(i)	1.25	116	183	92.8	146
1(ii) 0°	1.25	27.3	108	21.8	86.4
90°	0.10	9.74	50.9	97.4	509
2 0°	1.98	142	163	71.7	82.3
45°	2.58	79.4	155	30.8	60.1
90°	1.38	58.5	137	42.4	99.3
180°	2.05	142	165	69.5	80.4
3 0°	1654	3523	2978	2.1	1.8
45°	1385	2145	3045	1.5	2.2
90°	1775	2236	3078	1.3	1.7
180°	1636	3640	3044	2.2	1.9
4	82888	13212	21990	0.16	0.27

As an example, consider the point source on-axis in the homogeneous water cylinder (case 1(i)). The time to generate the EGS4 results was 1479min. Because all 64 projections were summed together, the time needed to generate an equivalent number of counts, without summation, is 64×1479min. This time yielded a peak accuracy of 0.42%. To correct to 1.5%, we multiply by a factor of $(0.42/1.5)^2$. This number is then divided by 64 to get the time for a single projection. The final time for comparison with our technique is

$$1479\mathrm{min} \times 64 \times \left(\frac{0.42}{1.5}\right)^2 \times \frac{1}{64} \;=\; 116\mathrm{min}$$

The AP calculation times for cases 3 and 4 in Table 4.5 are an estimate of the time that would be required for a direct application of the method to generating extended source projections. This approach calculates a separate point source projection for each voxel in the extended source distribution and then sums together these contributions in order to obtain the projection for the extended source. The AP projections shown in this thesis were not, however, generated in this manner. Instead, the symmetries in the z-direction of both the attenuating medium and the source distributions allowed us to reduce the actual time required by a factor of 6 for case 3 and 30 for case 4. The time was reduced by calculating the projections for three cross-sectional planes (of constant z) through the source distribution and then interpolated linearly between them to obtain the full projection. Each plane was one pixel wide and the three planes were located at the centre of the source distribution, at one end of the distribution, and at a z-position 75% of the distance away from the centre towards the end. Because of reflection symmetry across the x-y plane, we were able to acquire very accurate projections for the extended source without explicitly calculating the projection from every voxel. The times given in Table 4.5 are an extrapolation of the time required to generate these three planes.

We should note that symmetries can also be used to reduce the MC calculation times. For example the z-translation symmetry in the extended sources allowed us to sum the central seven slices of the MC projections and thus acquire the accuracy indicated in one seventh of the time. Also MC can further reduce the calculation time by using a coarser voxelization of the source distribution and attenuating medium. This can, however, affect the accuracy of results. For example, using a voxelization of 5^3 instead of 129^3 to describe the attenuating medium in the case of a point source on-axis in a homogeneous water cylinder, decreases the time by approximately a factor of two for an equal number of simulated photons but has a noticeable effect on the projection profile as is shown in Figure 4.9.

Because different symmetries can be used to decrease the computation times of both AP and MC codes, to remain as unbiased as possible in comparing times we have quoted all times as those required without the use of symmetries.

For the point sources, we have a speed improvement of 20–150 times over the MC simulations. One should note, however, that the error in our calculations depends on the effect of the approximations that we have made and the accuracy to which we have evaluated our numerical integrations. As such, our errors will be consistent throughout the projection. In contrast, the MC error in each pixel is dependent on the number of photons acquired in that pixel. In the scatter wings of the projections, the count level is often orders of magnitude less than the count level in the peak. In the projections shown in this thesis, there is a factor of 100 difference in the peak count compared to the scatter wings. For the MC codes to acquire a 2% error level in the scatter wings would thus require roughly 100 times as many simulated photons, or 100 times as much simulation time.

The AP calculation time for the hollow cylindrical source (case 3) is comparable to that of the MC, and the AP time for the cylinder uniformly filled with activity (case 4) is

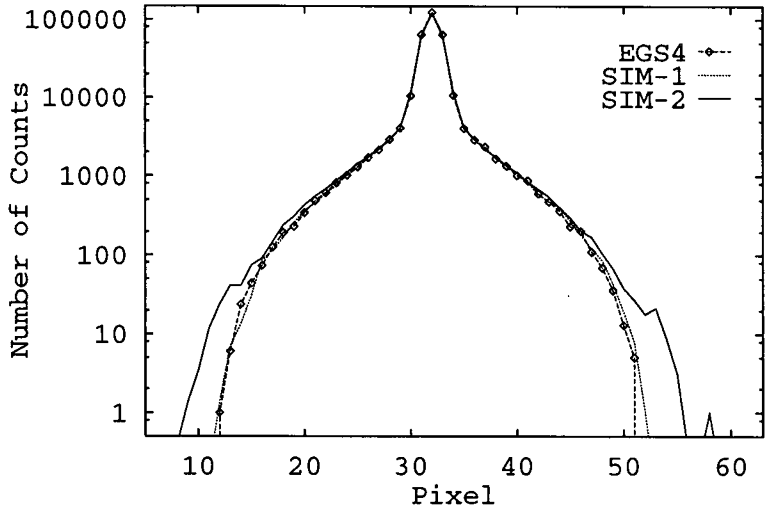

Figure 4.9: The effect of coarse pixelization on the projection profile. A transaxial profiles for an on-axis point source in a homogeneous, 10cm radius water cylinder is shown. *SIM-1* is the profile obtained using a fine pixelization (0.25cm/pixel) while *SIM-2* is the profile obtained using a much coarser pixelization (6.4cm/pixel). For comparison, the EGS4 data is also shown (*EGS4*).

approximately five times longer. The nature of the errors in these projections is the same as for cases 1 and 2 and, if low errors in the scatter region of the profile are desired, then the MC will require much longer run times. Additionally, no attempt has been made to optimize the AP method for extended sources. There are, however, many redundant calculations made when the projections from each individual source voxel are calculated independently. The removal of these redundancies would improve the calculation time for the larger sources.

4.5 Conclusions

In this chapter we have demonstrated that the AP method does accurately reproduce the results of Monte Carlo simulations. At a two percent accuracy level, it is as consistent with the MC simulators as they are with each other. The only area where the AP technique appears to have problems is near the edge of the scattering objects where pixelization becomes an issue. This problem is not specific to the AP calculations as it was also seen to a lesser extent with the SIMSET simulations. The significance of this effect can be reduced through the use of finer pixelization.

We have also shown that the AP technique offers a significant speed increase over a similar analytical calculation technique in the calculation of second order scatter distributions. When compared to MC simulators, it offers a time reduction for small sources although this reduction drops off as the source size increases. For sources of about 800 voxels the calculation times are comparable but for larger sources, the AP calculations take longer. There is, however, a large amount of room for optimization of extended source AP calculations.

There are two limitations to our technique when compared with Monte Carlo simulations. The first is that our method has a fixed calculation time. Monte Carlo is able to obtain noisy data faster by running fewer photon histories. A rough indication of the photon distributions can, therefore, be obtained faster than with our technique. Additionally, because Monte Carlo continues to give data with lower and lower statistical noise the longer that it is run, it is possible to eventually obtain more accurate data with Monte Carlo than is possible with our method. The accuracy of our calculations can improve through the use of more accurate lookup tables but we will eventually be limited by the accuracy of our approximations.

The second limitation of our technique is the time required to generate the look-up tables. The calculation of the look-up tables used in these simulations took sixteen weeks on a Sun Sparc10 workstation. This limitation is mitigated, however, by the fact that the tables need to be generated only once. Also, the algorithm for generating these tables contains many independent calculations and would greatly benefit from the use of multiple processors or vector machines for their one time generation.

Our AP method is in good agreement with Monte Carlo, and MC simulations provide a good approximation of the data acquired in SPECT. However, it is also necessary to directly check the agreement of our calculated results with experimentally acquired projections.

Chapter 5

Experimental Validation

Monte Carlo simulation is a useful technique for studying photon propagation in a fully controlled setting. However, with Monte Carlo you only get out what you put in. That is, only those effects which were programmed into the code affect the results and they do so in exactly the manner in which they were programmed to. Often our understanding of the processes which govern photon propagation from the point of emission to the point of acquisition is incomplete. Therefore, the equations and parameters used to describe the process may not be perfectly accurate. Additionally, the complexity of the real world is such that it is impractical if not impossible to include a perfect model of photon propagation in a Monte Carlo simulation. In consequence, an approximate model is frequently used (for example SIMSET doesn't model Rayleigh scatter and the version of EGS4 which we used did not take into account the effects of interference between Rayleigh and Compton scatter). Although it is assumed that the approximations do not significantly affect the outcome of the simulation, this cannot be known for certain unless actual experiments are performed. For these reasons, in addition to the Monte Carlo validations which were described in Chapter 4, we also performed accuracy validation studies using experimentally acquired data.

We first determined the characteristics of the Siemens MS3 SPECT camera system which was used for these studies. These characteristics are required for the generation of our lookup tables and were obtained by analyzing data acquired using four point sources suspended in air above one of the camera's detectors. We then acquired projections from a ^{99m}Tc point source (2mm in diameter) within a homogeneous water bath, a small spherical ^{99m}Tc source (1cm in diameter) within a homogeneous water cylinder, the small spherical source within a non-homogeneous medium consisting of air and water, and the small spherical source together with a larger spherical source (2.8cm in diameter) in a non-homogeneous air and water medium.

The remainder of this chapter discusses these experiments and is divided into three sections. The first section details our measurements of the camera characteristics for the MS3 system. The next section describes the four phantom validation experiments and the corresponding results. A summary of our experimental work is given in the third and last section. This work has been presented in part in the journal *IEEE Transactions on Nuclear Science* [161].

5.1 The Camera Characteristics

As mentioned previously, our method relies upon pregenerated look-up tables of the patient-independent parameters of the system as well as on an estimate of the patient-specific activity distribution and map of the attenuation coefficients for the scattering medium. The patient-independent parameters include the energy of the gamma rays emitted by the source, the collimator geometry, the detector efficiency, and the intrinsic spatial resolution of the camera. These parameters can be estimated theoretically or determined experimentally and need only be determined once for each camera-collimator-isotope combination. A theoretical estimation of these parameters suffers from the same difficulties as does MC simulation and in order to generate camera specific reference tables and also incorporate effects such as septal penetration which are difficult to include theoretically, we elected to determine the parameters experimentally.

The specific functions which are required to describe the camera characteristics were presented in detail in Chapter 3. They are the normalized collimator acceptance function E(ξ) (describing the probability that a photon, incident on the collimator surface at an angle ξ with respect to the collimator normal, is recorded by the camera), a function describing the intrinsic spatial resolution, G_d, and a scaling factor which accounts for the intrinsic efficiency of the camera.

We determined these functions from data acquired through experiments with no attenuating medium. Four point sources were positioned in air above a camera head at nine different distances using two different methods: Styrofoam supports (collimator-source distances of 5cm, 10cm, 15cm, 20cm, 31cm, and 42cm) and suspension from the camera gantry (distances of 43cm, 50cm, and 60cm). Data was acquired with only this camera head and it remained in a fixed position throughout the experiments. The point-like sources used in these experiments were constructed by injecting activity into a capillary tube with an internal diameter of 1.1mm and a wall thickness of 0.2mm. The length of activity within the tube was restricted to 1–2mm. The activities of the point sources were 17.8, 18.6, 21.8, and 24.3 MBq, determined at the beginning of the experiment. To generate projections with low statistical noise, data was acquired until a total of $\sim 10^6$ photons were recorded in each (four source) image. This required 20–30min for each projection.

A Low Energy Ultra High Resolution (LEUHR) parallel hole collimator was used. This collimator is 35.6mm thick and contains an array of hexagonal holes with a diameter of 1.16mm and a septal thickness of 0.13mm. The collimator was chosen to provide the highest possible spatial resolution for comparison with our calculations. The projections were acquired using a 1024x1024 array which corresponds to a pixel size of 0.045cm. In all cases the isotope used was ^{99m}Tc and the energy window chosen was the standard 20% symmetric window centred at 140keV.

5.1.1 The Collimator Acceptance Function, E(ξ)

The point spread functions (PSF) measured at 60cm were used to obtain a radially symmetric collimator acceptance function, E(ξ). The 60cm data was chosen as it provides the greatest angular resolution (in ξ). The PSFs were deconvolved with the source size (a 1.12mm diameter radial step function corresponding to the external diameter of the enclosing capillary tube) and an initial estimate of the intrinsic spatial resolution of the camera (a Gaussian function with a FWHM of 3.8mm as given by the manufacturer's specifications for the Siemens MS3 camera and as confirmed by quality control tests performed on the camera). A seventh order polynomial was then fit to the data to obtain E(ξ) (Figure 5.1). The seventh order polynomial was chosen as it was the lowest order fit which provided both a good fit to the data and also agreement with the theoretical calculation at low angles. The theoretical curve was found assuming circular holes instead of hexagonal ones and also assuming no septal penetration. Deviation of the experimental curve from the theoretical one at angles between 1^o and 2^o was suspected to be due principally to differences in the shape of the holes and to septal penetration of the collimator. The experimental fit to the data was the one used in our calculations as it is a better representation of the actual acceptance function than is the theoretical curve.

5.1.2 The Intrinsic Spatial Resolution

Approximating the intrinsic spatial resolution of the camera as a simple Gaussian function proved to be insufficient for the reconstruction of the PSFs from our experiments. A more accurate intrinsic spatial resolution function was, therefore, acquired by determining the

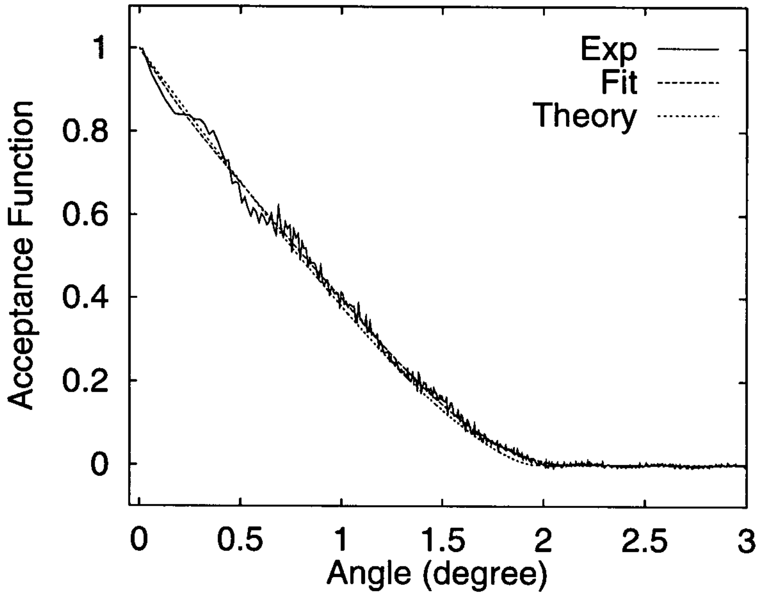

Figure 5.1: The collimator acceptance function E(ξ) is obtained by a seventh order polynomial fit (*Fit*) to the experimental data (*Exp*) from a point source in air 60cm from the collimator surface. A theoretical acceptance function (with no septal penetration) is also shown (*Theory*).

radially symmetric convolution kernel which was best able to convert a calculation of the unscattered image into the experimental images obtained with the sources 10cm from the collimator surface. The convolution kernel was calculated using the iterative least squares procedure described in [19]. The data at 10cm was chosen in order to minimize the influence of geometric blurring caused by distance from the collimator. This increased the dominance of the intrinsic spatial resolution of the camera on the shape of the acquired PSF. The data at 5cm was not considered adequate because at this distance the influence of distinct collimator holes may be important and because the number of data points available in the 1024x1024 array was not sufficient. A cross-section through the peak of our convolution function is shown in Figure 5.2. For comparison, we have also shown a Gaussian function with a 3.8mm FWHM.

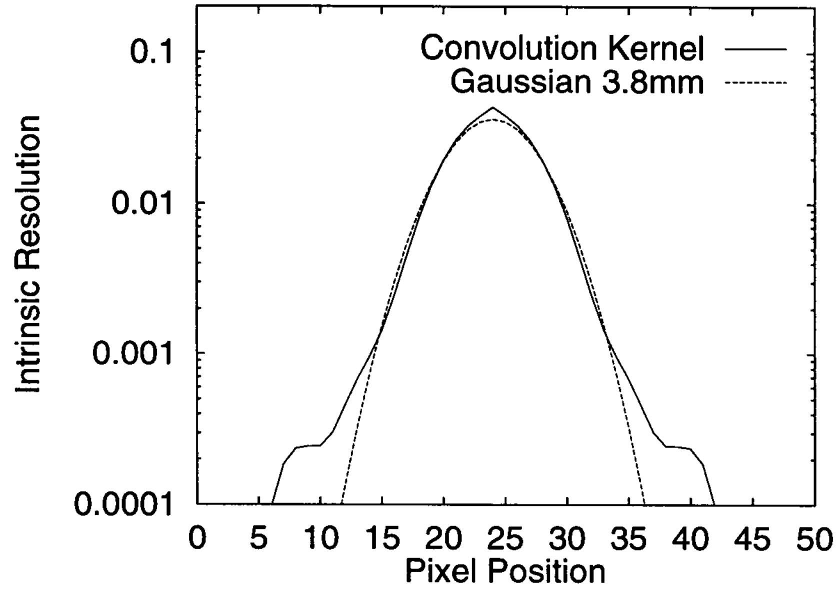

Figure 5.2: The intrinsic spatial resolution function. An AP calculation of the unscattered PSF for a point source 10cm from the collimator surface was performed assuming perfect intrinsic spatial resolution. The radially symmetric convolution kernel that best converts the AP calculated PSF into the corresponding experimentally acquired data was then computed. The kernel (*Convolution Kernel*) was found by an iterative least squares technique. Also shown is a Gaussian function with FWHM=3.8mm (*Gaussian 3.8mm*) which is the manufacturer's specification for the intrinsic spatial resolution. Each pixel is 0.045x0.045cm^2. The bump in the kernel seen for example at pixel 40 is caused by a ringing artifact in the computed fit. These features would be smoothed out if the kernel was collapsed into the 0.72x0.72cm^2 pixel size used in AP calculations.

The convolution kernel shown in Figure 5.2 is for a pixel size of 0.045x0.045cm corresponding to the 1024x1024 array. Before it can be applied to the 64x64 projections normally produced by our AP technique, this kernel must be compressed. However, because the binning function does not commute with the convolution operation, one cannot simply rebin the 1024x1024 kernel (add together 16x16 pixels from the 1024x1024 kernel to form a single pixel of the 64x64 kernel). Instead, the kernel is applied to a 1024x1024 image (in this case, one of the experimental images acquired at a collimator-source distance of 60cm was used). Both this convolved image and the original image are then rebinned into 64x64 arrays. The convolution kernel for the 64x64 image size is determined as before using least squares fitting.

5.1.3 The Intrinsic Efficiency

The intrinsic efficiency of the camera (the efficiency of the crystal coupled with the electronics) was estimated based on the total number of counts recorded by the camera (with the collimator present) as compared to the total number of counts computed with the AP calculation based on our calibration of the source activities. The effects of collimator size and acceptance angle on the camera efficiency are not included in this parameter as they have already been taken into account in our calculations.

The scaling factor required to match the total number of counts recorded in the experimental data with the total number computed by our method was first obtained for the point sources in air at each distance in an effort to determine the effective intrinsic efficiency of the camera. The average efficiencies obtained from the point sources in air experiments (as shown in Table 5.1) ranged from 58% for the 24.3MBq source to 65% for the 17.8MBq source. As can be seen from the table, there seems to be a systematic error

Table 5.1: Intrinsic camera efficiencies for point sources in air.

Distance	Source Strength (MBq)			
(cm)	17.8	18.6	21.8	24.3
5	0.672	0.657	0.638	0.594
10	0.650	0.658	0.642	0.595
15	0.659	0.654	0.641	0.588
20	0.656	0.646	0.644	0.583
31	0.651	0.645	0.632	0.584
42	0.621	0.621	0.610	0.563
43	0.657	0.637	0.617	0.579
50	0.656	0.634	0.616	0.577
60	0.653	0.632	0.615	0.577
average	0.653	0.642	0.628	0.582

correlated with the strength of the source which we believe was caused by deadtime[1] in the camera. Although the total count rate of the source did not exceed the recommended system limits, we believe that, because of the use of tightly defined point sources, the local deadtime did become significant. The higher activity sources will lose more counts due to deadtime which will result in a correspondingly lower detector efficiency. Because of the variation in the observed efficiencies, we were unable to use this calibration for the experiments discussed in section 5.2. Instead, each AP projection was scaled such that the number of counts in the peak of the image (the nine pixels centred around the image pixel containing the largest number of counts) approximately matched those in the experimental image. Additional experiments were later performed to calibrate the efficiency of the camera and these are discussed in detail in section 5.3.

[1] A brief discussion of deadtime is given in section 1.7.3.

5.1.4 Confirmation of Camera Characteristics

The point sources in air at distances other than 10cm and 60cm were used to check the consistency of our determination of the functions describing the camera characteristics. As an example, Figure 5.3 shows the profiles in both the horizontal and vertical directions through the peak of a projection obtained from a point source positioned 20cm above the collimator surface.

Here, and also in Figures 5.5–5.9, the profile is the sum of the central three rows of the projection. This summation was done to reduce noise in the wings of the experimental data. The figures demonstrate good agreement between the experiments and our calculations indicating that we have correctly determined the collimator acceptance function E(ξ) and the intrinsic spatial resolution function for the camera.

5.2 Phantom Measurements of Photon Distributions

Once the camera characteristics had been determined and our lookup tables generated, we executed a series of phantom experiments designed to test the accuracy of our AP calculations in a more realistic situation. Data was acquired as a 1024x1024 array but subsequently collapsed into a 64x64 array in order to compare it with the projections produced by our calculations. Each pixel in the 64x64 projection is 0.72cm on a side. As with the camera characteristics experiments, a LEUHR collimator was used. In order to obtain good statistics, the data acquisition was run until 20–40 million photons were recorded in each image. In all cases, planar images (single SPECT projections) were acquired.

The AP calculations included unscattered photons, first and second order Compton scatter (including the incoherent scattering functions) and an estimate of the Rayleigh scatter (the form factors for which include the effects of coherent-incoherent interference).

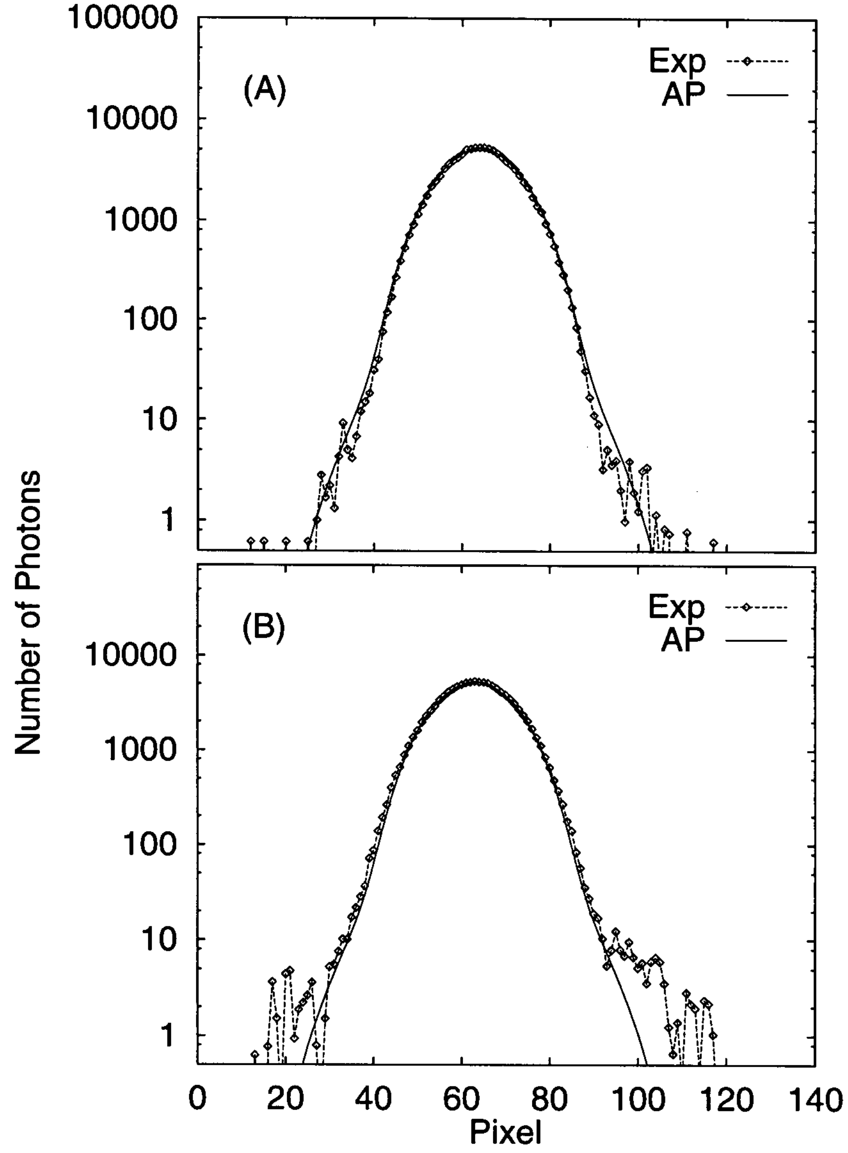

Figure 5.3: Semi-log plots of the profiles through the peak of a projection in which the source was 20cm away from the collimator surface with no attenuating medium. Horizontal (A) and vertical (B) profiles are shown for the experimental data (*Exp*) and for our calculations (*AP*). The pixel size is 0.045cm. Deviations of the experimental data from the AP calculated results occur in the noisy region of the experimental projection (for example near pixel 20 of image (B)).

The Rayleigh scatter was included as an adjustment to the first order Compton scatter lookup table as discussed in Chapter 3.

The AP technique requires an estimate of the source distribution and the tissue density distribution (attenuation map). In principle, the source distribution estimate can be obtained by filtered back-projection of (uncorrected) SPECT data while the attenuation map can be obtained from a transmission scan. In our phantom experiments, however, the back-projection and the transmission scan were unnecessary as both the source and material density distributions were known *a priori*. The source activity was calibrated before each experiment and the source was positioned at a known location within the phantom. The position and distribution of the attenuating/scattering material was carefully measured and values for the attenuation coefficients and electron densities were taken from the literature [25, 27, 24, 158]. As mentioned in section 5.1.3, each AP calculation was scaled to match the counts in the peak of the experimental projection.

5.2.1 The Phantom Configurations

The sources we considered were a point source, similar to that used to determine the camera characteristics, and two spherical sources which were 0.46cm and 1.4cm in radius and filled uniformly with activity.

The following cases were considered.

1. The point source positioned 9.35cm below the water surface in a large water filled cylinder. The cylinder was 25cm in diameter, 25cm high, and filled to a depth of 15cm. The camera head was positioned directly above the open end of the cylinder so that the surface of the collimator was 10cm above the water surface. The axis of rotation of the water cylinder was oriented parallel to the collimator surface normal. The source was positioned 2cm off the axis of rotation of the cylinder.

2. The small sphere was positioned at location 1, shown in Figure 5.4A, inside a homogeneous water cylinder 22.2cm in diameter. The axis of rotation of the cylinder was perpendicular to the collimator surface normal and parallel to the axis of rotation of the camera.

3. The small sphere was positioned at location 1 (as in case 2) and an air cylinder (6cm in diameter) was inserted at location 2. The axis of rotation of the air insert was parallel to that of the enclosing water cylinder.

4. Two spherical sources were placed in the 22.2cm diameter water cylinder and the relative position of the air insert was changed (Figure 5.4B). The small sphere (0.46cm radius) was fixed at location 3 while the larger sphere (1.4cm radius) was placed at location 4. The ratio of the total activity in the small sphere to that in the large sphere was 1.85:1.[2] The air cylinder (6cm in diameter) was moved to location 5 but its axis of rotation remained parallel to that of the water cylinder.

The point source in a water bath (case 1) was chosen because it represents the simplest configuration we could devise. The camera head was positioned directly above the open water container in order to avoid complications due to scattering and attenuation within the container walls.

The remaining three experiments gradually increase the complexity of the imaging situation. In case 2, the source size is increased and it is moved into a closed container, the Jaszczak phantom. In case 3, an inhomogeneity is added to the phantom in an asymmetric position. Finally, in case 4, the source is moved to an asymmetrical position in the phantom and an additional source with a different activity is added.

[2] The ratio of the activity concentrations of the two spheres was 50:1.

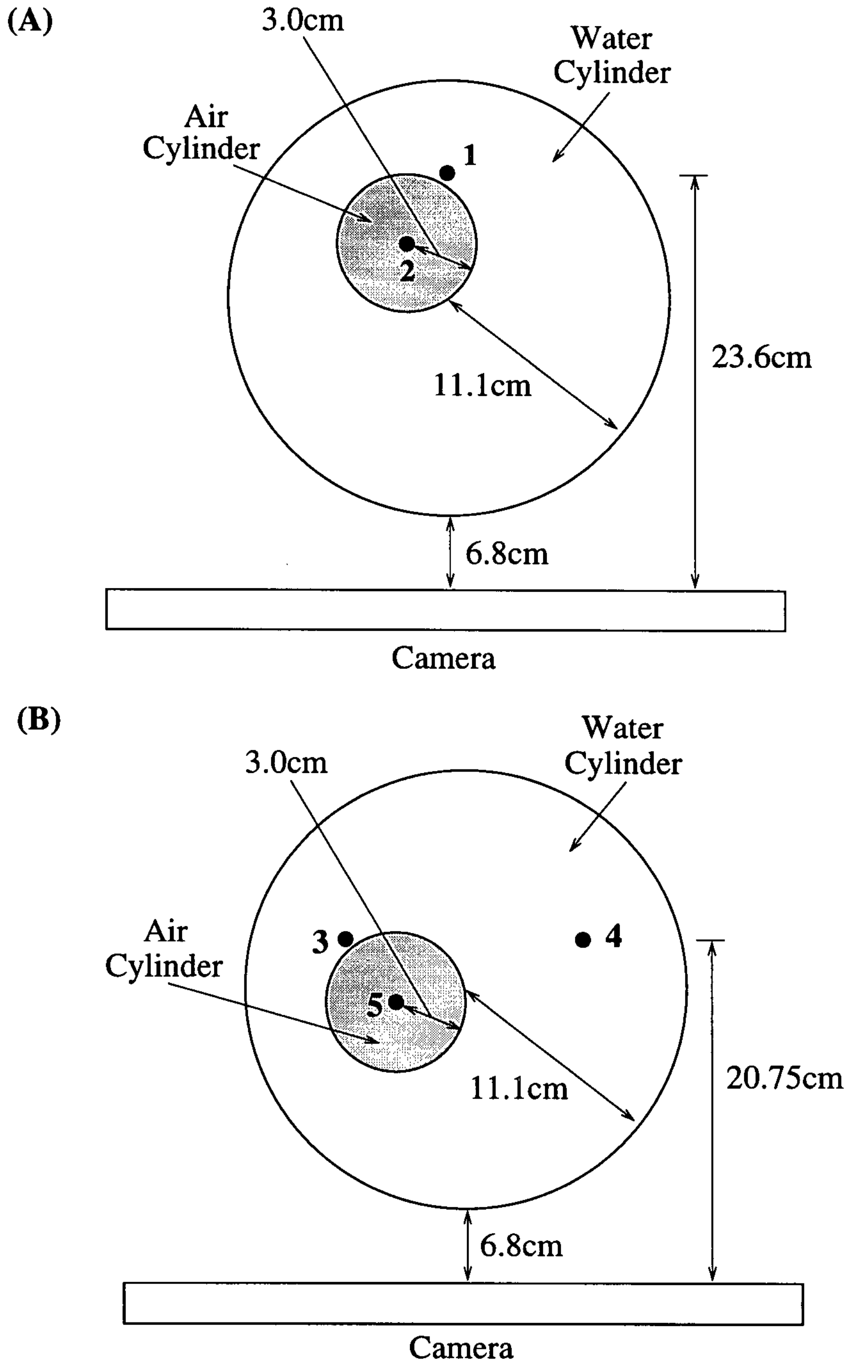

Figure 5.4: The phantom configurations for cases 2 and 3 (A) as well as for case 4 (B). Point 1 is at (0,23.6) with respect to the collimator surface, point 2 is at (-1.5,20.5), point 3 is at (-4.94,20.75), point 4 is at (4.94,20.75), and point 5 is at (-3.0,17.9). All distances are in centimeters.

5.2.2 Results: Point Source in a Water Bath

Figure 5.5 shows semi-log profiles of the point source in the water bath. One can see from this figure that a good match is obtained across the entire projection although the wings of the logarithmic profile are slightly underestimated. This discrepancy is believed to be caused by the absence of accurately calculated distributions for the 3rd and higher order scatter events and also by the effect of deadtime. The influence of deadtime will be most significant in the peak of the image where the count rate was the highest and will result in a lower effective count rate. Therefore, because the AP projection is fit to the peak of the experimental image, the AP calculation will underestimate the counts in the wings of the image where deadtime effects are less prominent.

In most of the experimental profiles presented in this chapter, there can be seen an abrupt truncation of the experimental data which occurs at the edge of the camera's field of view. In the profile shown in Figure 5.5B, this is seen at pixels 8 and 50. The results of the AP calculations have not been correspondingly truncated.

It will also be noted that the AP calculated profile levels out at a non-zero value towards the edges of the image. This flat region represents the background activity present during the experimental acquisition. The background can be seen, for example, in the experimental data of Figure 5.5A between pixels 6–14 and 52–61. This is not, however, seen in most of the experimental profiles because of truncation at the edge of the field of view of the camera or because it has been masked by back-scatter (section 5.2.3). The background activity was measured for 15 minutes at the beginning of each experiment. The average background level was adjusted for the experimental acquisition time and added uniformly to all pixels in the AP projection.

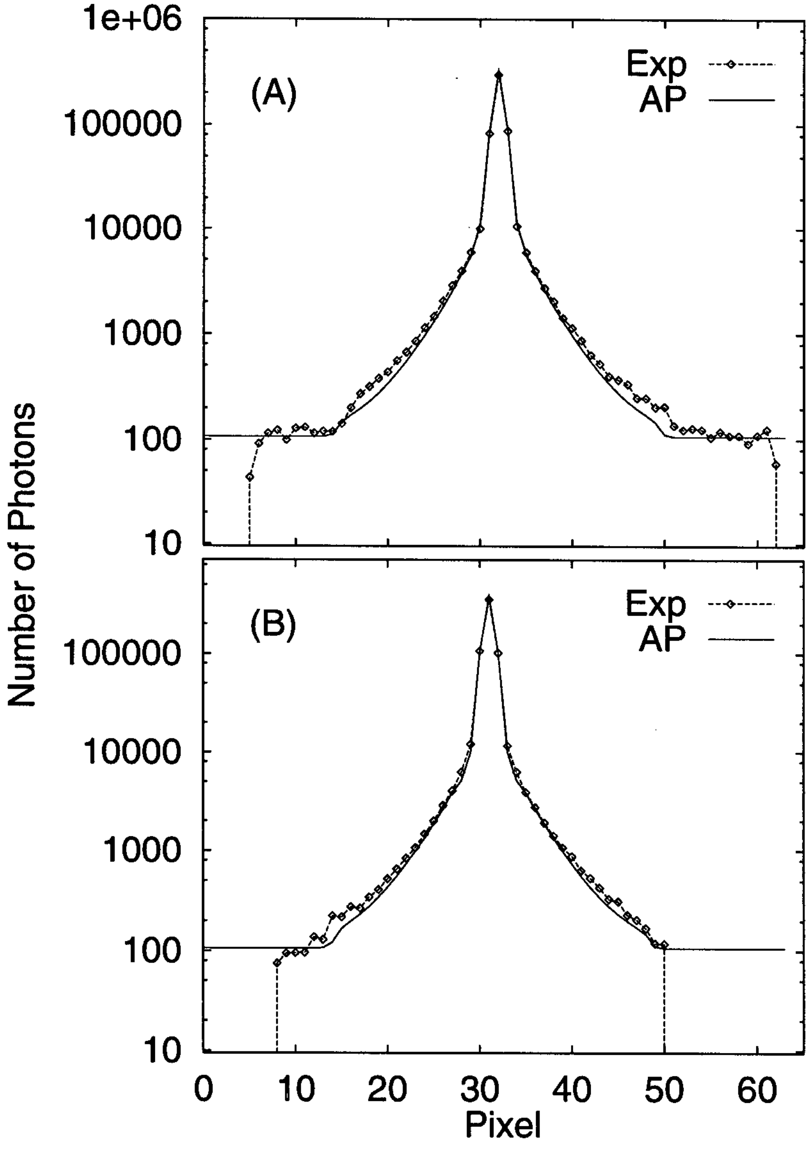

Figure 5.5: Semi-log plots of the horizontal (A) and vertical (B) profiles through the peak of a projection in which the source was 9.35cm beneath the surface of a water bath whose surface was 10cm from the collimator. Pixel size is 0.72cm. Shown are the experimental data (*Exp*) and our calculations (*AP*).

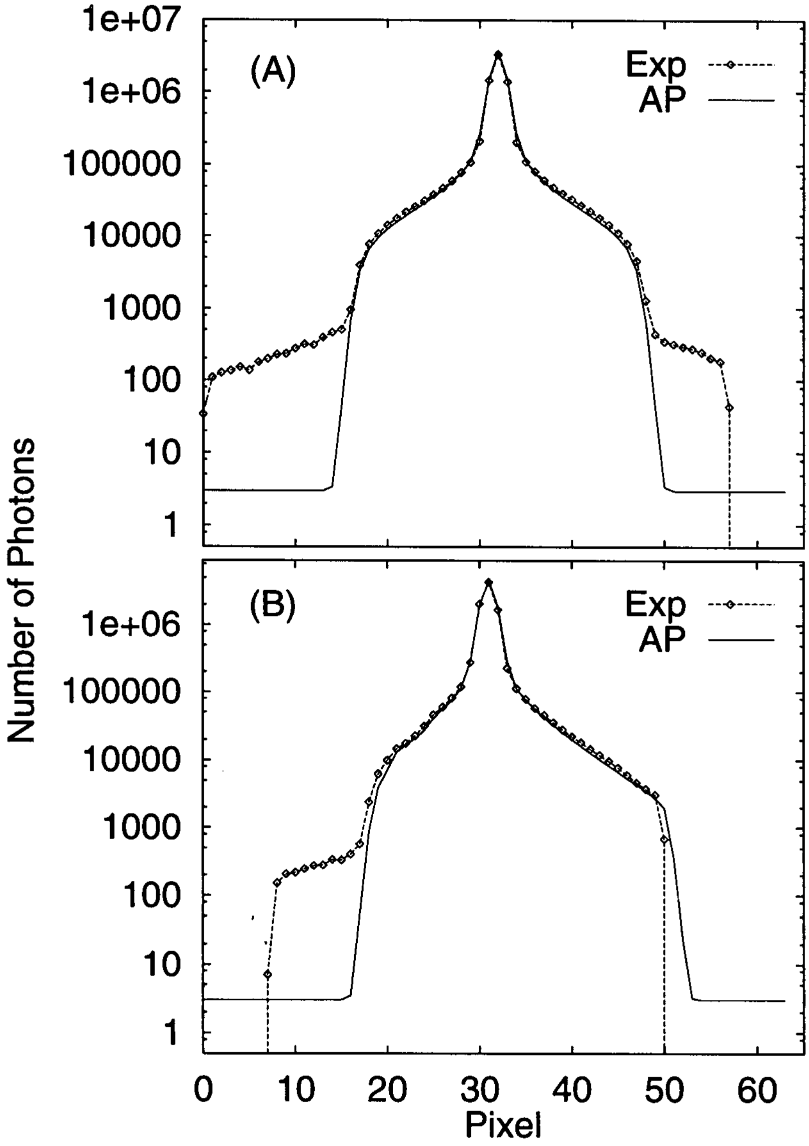

Figure 5.6: Semi-log plots of the profiles through the peak of a projection in which the source was located at position 1 in the homogeneous water cylinder. Pixel size is 0.72cm. Shown are horizontal (A) and vertical (B) profiles of the experimental data (*Exp*) and our calculations (*AP*).

5.2.3 Results: Small Sphere in a Water Cylinder

Figure 5.6 shows semi-log profiles of the small spherical source in the water cylinder. The 0.46cm radius sphere was modelled in 3D as seven point sources: one source with 80% of the activity centrally located and symmetrically surrounded by six additional sources in which the remaining 20% of activity was evenly distributed. To obtain the final projection for this source, a projection was independently generated for each of the seven sources and these seven projections were later summed together. The final projection was then blurred by a 2D Gaussian function of 0.4cm FWHM to account for the even distribution of the activity throughout the voxels. (The width of the Gaussian function was chosen such that 95% of the counts are retained in the width of one voxel, 0.72cm.) The projection was further convolved with the intrinsic spatial resolution function and then compared to the experimental data.

The shape of the calculated distribution agrees well with the experimental data. There is a good match across the majority of the profile, although the wings are again slightly underestimated. This is believed to be caused by the absence of calculated distributions for the 3rd and higher order scatter events, by the effect of deadtime on the image peak, and also possibly by slight differences in the attenuation coefficients and electron densities of water compared to the plastic in the phantom. The phantom was modelled as pure water with no differentiation between the water-equivalent plastic of the walls and the true water within the container.

The calculated distribution does, however, deviate significantly from the experimental data in the extreme wings of the profile. This is seen in Figure 5.6A in pixels 1–15 and pixels 49–56. We hypothesized that these events were caused by back-scatter outside of the cylinder, possibly from other parts of the camera such as the gantry or the other two camera heads. Scatter events from outside of the water cylinder were not modelled in our simulations.

To test this possibility we repeated this experiment with the addition of a lead apron. The apron was placed beneath the cylindrical phantom and extended away from it on the left side at roughly the same angle as the retracted collimator head. The apron was only inserted on the left side of the phantom such that the right side of the phantom configuration was unaltered and remained the same as in case 2. The projection acquired with this configuration was then examined for left-right asymmetries. A semi-log profile with the left side folded over onto the right is shown in Figure 5.7.

The presence of the apron results in a strong increase in the amount of scatter detected in the extreme wings of the projection (outside of the region corresponding to scatter from the water cylinder). This supports our hypothesis that the photons detected in

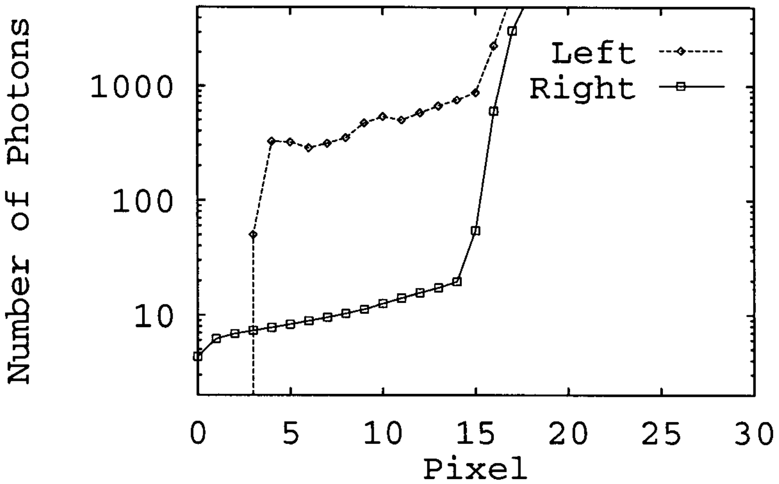

Figure 5.7: The influence of back-scatter from the camera heads. Shown is a horizontal profile for a small sphere placed symmetrically in a homogeneous water cylinder. This profile shows the effect of a lead apron placed close to the cylinder on the left hand side (*Left*). There is no apron on the right hand side (*Right*).

the extreme wings are caused by scatter off objects outside the water cylinder such as the lead collimators on the other two camera heads. As these were not modelled in our simulations, these counts are not seen in the AP calculations. Back-scatter from the camera heads is also not seen in clinical imaging situations. The distribution in the extreme wings is two orders of magnitude lower than that due to scatter in the cylinder and three or more orders lower than the unscattered distribution. Thus, with clinically relevant activity levels and scan times, the back-scatter, while present, is dominated by the statistical noise in the data.

5.2.4 Results: Small Sphere in a Non-homogeneous Cylinder

The profiles for the small spherical source in a water cylinder containing an air insert (case 3) are shown in Figure 5.8. The abrupt truncation of the experimental data in Figure 5.8B at pixel 44 is another example of the truncation mentioned in section 5.2.2. It is caused by the scatter extending beyond the active field of view of the camera head. The AP profile has not been correspondingly truncated.

Once again, there is a good agreement to the general shape of the profile with a slight underestimation of the wings. The alterations in the profile caused by the air insert are faithfully reproduced by our calculations. The scatter seen in the extreme wings of Figure 5.6 is also seen in this figure and we believe has the same source.

5.2.5 Results: Two Spheres in a Non-homogeneous Cylinder

Finally, we considered our most complex case of a small spherical source together with a larger spherical source and an air insert in the water cylinder (case 4). The semi-log profiles for this situation are shown in Figure 5.9. Figure 5.9A is the horizontal profile through both peaks while Figures 5.9B and 5.9C are vertical profiles through each of the two spheres.

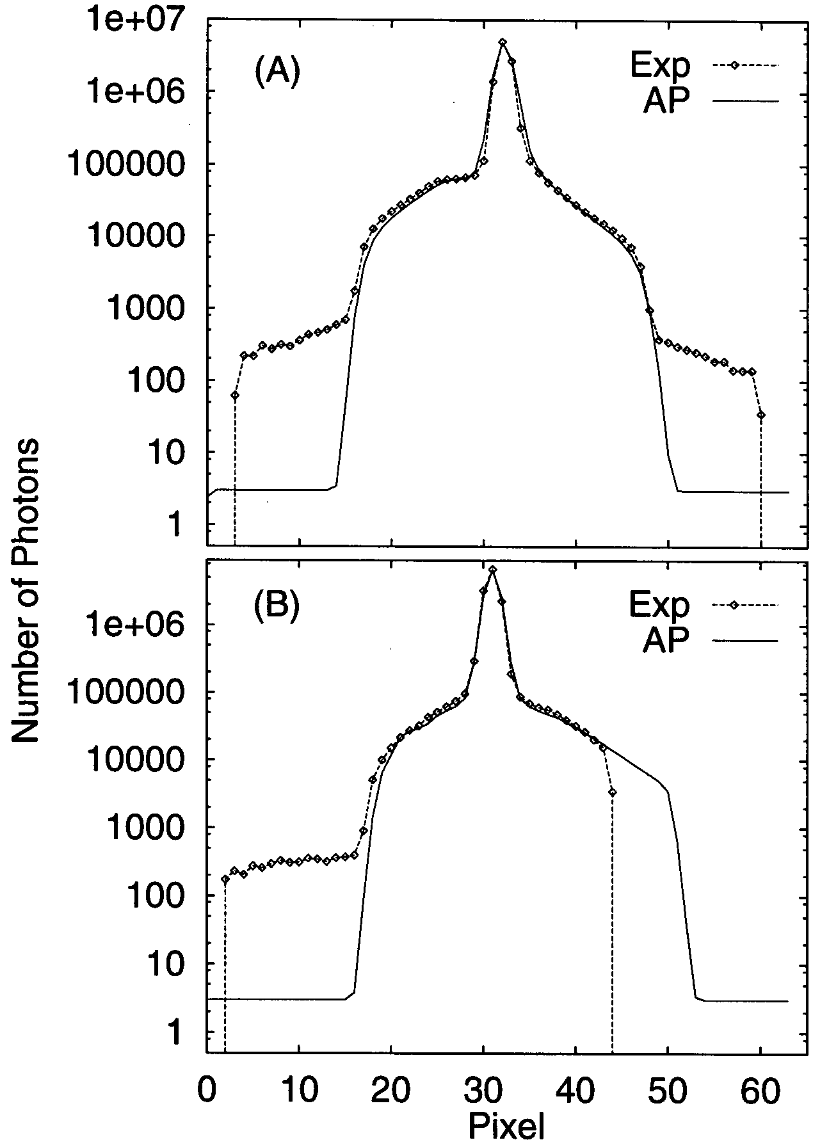

Figure 5.8: Semi-log plots of the profiles through the peak of a projection from a spherical source placed at position 1 (Figure 5.4A) in a water cylinder with a cylindrical air insert, 3cm in diameter, at position 2. Shown are horizontal (A) and vertical (B) profiles of the experimental data (*Exp*) and our calculations (*AP*). The pixel size is 0.72cm.

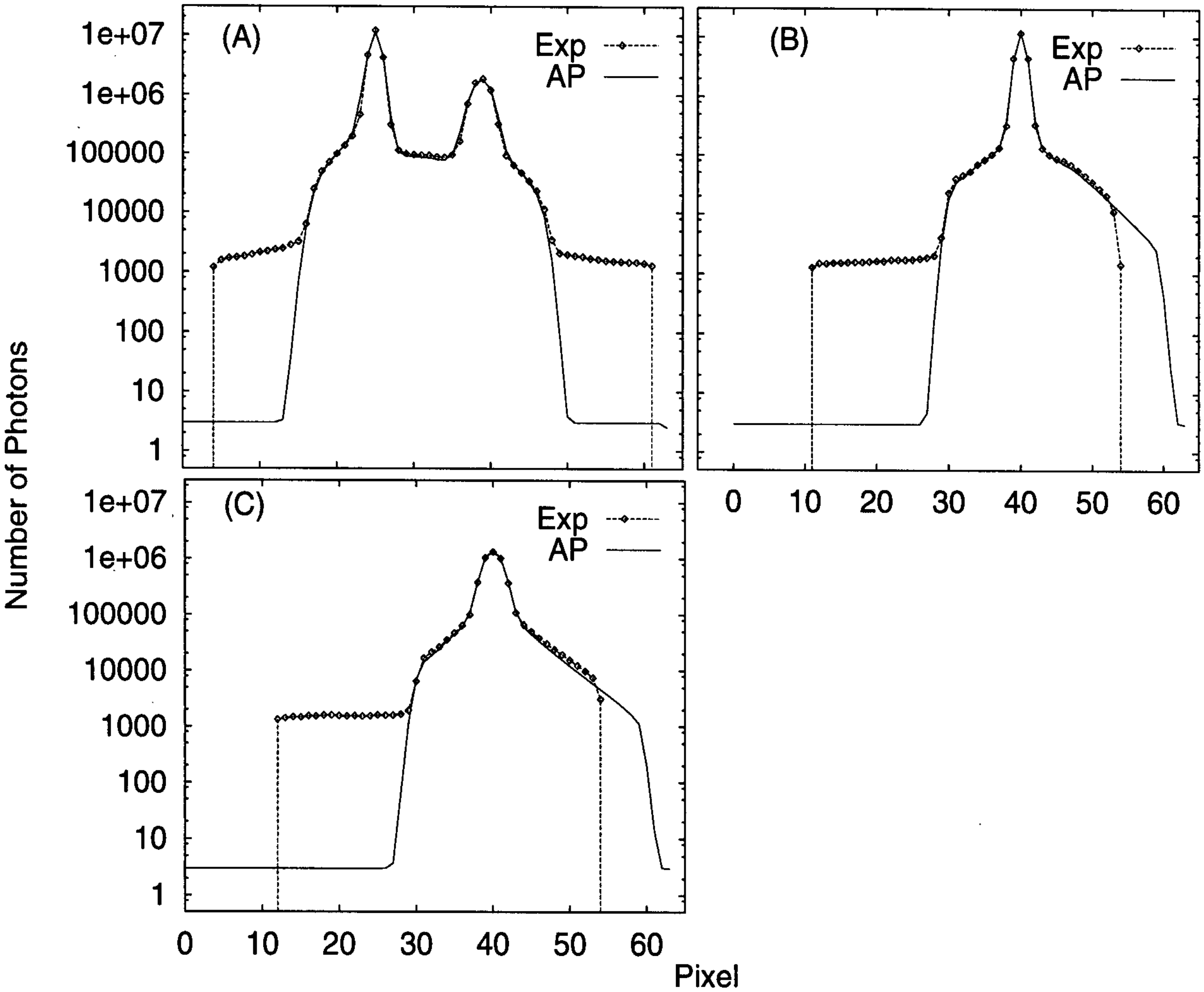

Figure 5.9: Semi-log plots of profiles through the peaks in a projection obtained with two spherical sources placed in a water cylinder containing a 3cm diameter cylindrical air insert, (Figure 5.4B). The small sphere is at position 3, the larger sphere at position 4 and the air insert is at position 5. The ratio of the total activity in the small source to that in the large is 1.85:1. Shown is a horizontal profile through the two spheres (A) and vertical profiles through the small (B) and large (C) spheres. The experimental data is denoted *Exp* while our calculations are labelled *AP*. The pixel size is 0.72cm.

There is an excellent match in the general shape of the profile between the experiment and our results. The alterations in the profile caused by the air insert are again faithfully reproduced. A single scaling factor was used for the entire image (corresponding to an efficiency of 64%). This one scale factor resulted in an accurate reproduction of the relative sizes of the two peaks (corresponding to a ratio in total activity of 1.85:1). The scatter from objects outside of the region modelled is also seen in the extreme wings of this figure.

5.3 Efficiency Experiments

To accurately calibrate the intrinsic efficiency of the camera system, we repeated the experiments using the phantom configurations described in case 1 and case 2. The source strengths were greatly reduced in order to avoid the effects of deadtime. (Clinically, deadtime should not be an issue as the activity concentrations normally used for patient studies are much lower than those originally used in these experiments.) To acquire good statistics in the projections, the acquisition times were also increased.

For case 1, a 0.87 MBq source was imaged for 13 hours and for case 2 a 108 MBq (0.5ml) source was imaged for 21 hours. Semi-logarithmic profiles for these experiments are shown in Figures 5.10 and 5.11.

One can see from Figure 5.10 that a good match is obtained across the entire profile. By making a uniform correction for the background activity and then scaling our data so that the total number of counts in the experimental and calculated projections was the same, we obtained a value of 68.3% for the intrinsic camera efficiency.

An efficiency of 68% is used to generate Figure 5.11 which shows profiles through the projection acquired with the small sphere in the homogeneous medium (case 2). Using this value for the intrinsic efficiency of the camera results in less than 2% error in the

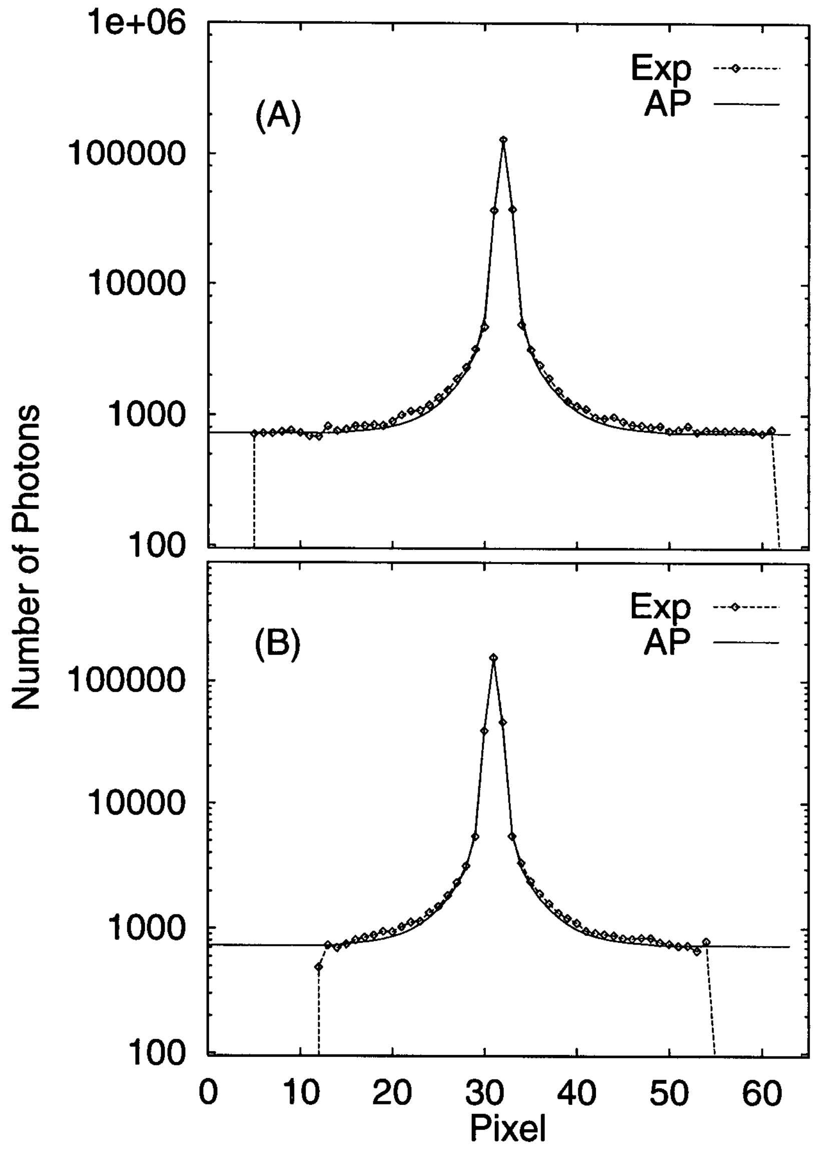

Figure 5.10: Efficiency experiment for a point source in a water bath. The experiment was set-up in the same manner as in Figure 5.5 but with a weaker 0.87 MBq source. The efficiency of the camera determined from this experiment was 68.3%.

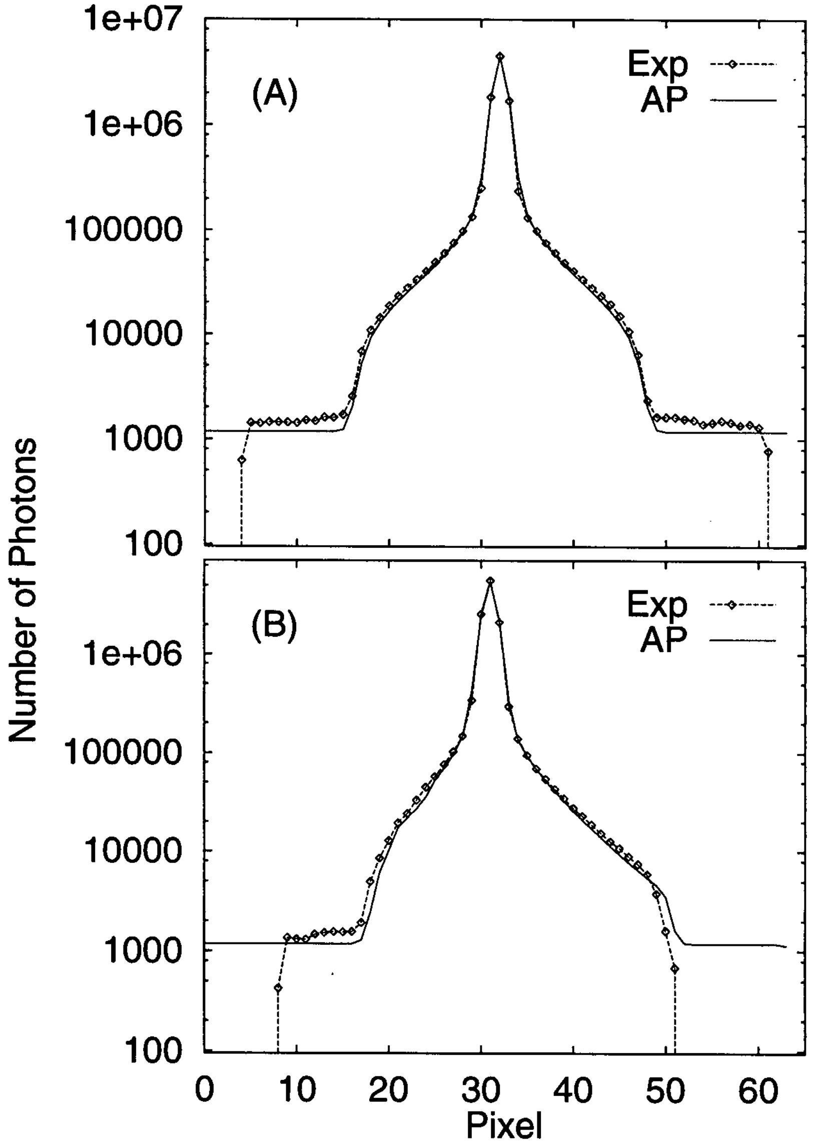

Figure 5.11: Efficiency experiment for a small sphere in a water cylinder. The figure contains semi-log plots of the profiles through the peak of a projection in which the source was located at position one in the homogeneous water cylinder. Pixel size is 0.72cm. Shown are horizontal (A) and vertical (B) profiles of the experimental data (*Exp*) and our calculations (*AP*).

total number of counts recorded for this projection.

The back-scatter effect seen in Figure 5.6 is still present in Figure 5.11 but its appearance is masked by the relative increase in the background levels as caused by the increased acquisition time. In order to determine the intrinsic efficiency from this projection, an average back-scatter level was also added uniformly to the image as an approximate correction. The back-scatter contribution was determined by averaging those values in the experimental image for which the AP calculation predicts no recorded counts. The scale factor required to match the total counts in this image is 69.1%.

Averaging the results of these two experiments yields an intrinsic efficiency of 68.7±0.4% for the camera. This value agrees with the efficiency of the NaI crystal (70 – 75%) at 140keV as given in [10].

Concern regarding the consistency of the detector efficiency over time prompted us to perform one additional experiment. A 50μCi Co-57 source was placed at a fixed position approximately equidistant from all three camera heads. Images with all heads were acquired for 20min. This experiment was repeated four additional times over the following 10 days. The total number of counts recorded by each head was then background corrected and plotted as a function of the day it was acquired. For comparison, this experiment was also performed on a dual head camera made by a different manufacturer (Sopha) using a high resolution collimator. The results of this experiment are shown in Figure 5.12.

The camera heads were mispositioned by 180° for the data acquired on day 6 with the MS3 camera. As camera head one was normally positioned directly above the source, the 180° mispositioning most significantly affected this data point. With the exception of the data for head 1 on day 6, the efficiencies of the three MS3 camera heads are consistent over a period of eleven days to within 3%. If higher accuracy than this is desired, it would be necessary to calibrate the camera on a more frequent (daily) basis.

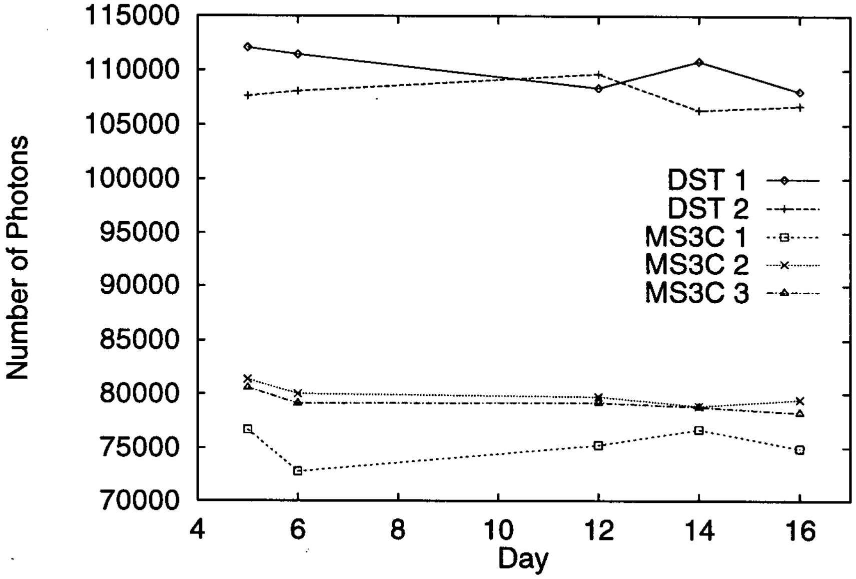

Figure 5.12: Efficiency consistency check of the Siemens MS3 Cardiac camera *MS3C* and the Sopha DST dual head SPECT camera *DST*. The camera heads are referred to by number. Note that a mispositioning of the camera heads on Day 6 may have distorted the data for the MS3 camera.

The variation in the data acquired with the Sopha DST camera (3.5%) is slightly higher than in the MS3 data but also quite consistent. These results indicate that, although one needs to be careful of deadtime issues when calibrating the camera, consistent intrinsic efficiencies can be obtained for modern gamma cameras.

5.4 Summary

It was possible to determine the camera specific characteristics used in the evaluation of our lookup tables using a set of point-source-in-air measurements. It was found that the experimentally acquired acceptance function deviates from the theoretical calculation and this deviation is thought to be caused primarily by septal penetration. The intrinsic spatial resolution of the camera was found to be inadequately modelled by a

Gaussian function with a FWHM of 3.8mm. A more accurate resolution function was determined by a least squares fitting as the convolution kernel which converted the AP calculated projection into the corresponding experimentally acquired projection. The intrinsic efficiency of the camera was found to be $68 \pm 1\%$.

The AP calculations accurately reproduced the shape of the experimental projections for small sources, including the deviations caused by inhomogeneities in the attenuating medium. The quantitative accuracy of our calculations was assessed in the case of a point source in a water bath and a small spherical source in a water cylinder. We found that an intrinsic efficiency of 68% produced agreement to within 2% between the total number of counts recorded in AP calculated and experimentally acquired projections. The technique also resulted in good relative quantitation for the case of the two ^{99m}Tc sources in the inhomogeneous medium, reproducing the total activity ratio of 1.85:1.

The results of these experiments coupled with those from the Monte Carlo comparisons show that the AP calculation method that we have developed is capable of accurately reproducing the true distribution of photons which would be acquired in SPECT projections. Discretization of an extended source distribution into a collection of point sources allows for the calculation of the larger source projections as the sum of point source projections. The AP technique provides an accurate way to separate primary and scattered photons in the projection. This capability has many potential applications one of which is the correction for cross-talk in dual-isotope studies.

CHAPTER 6

DUAL ISOTOPE CROSS-TALK CORRECTION

Dual-isotope SPECT studies use two radiopharmaceuticals, each of which has been labelled with a different isotope. The purpose of using two different tracers is to examine two different aspects of a single physiological parameter or to view two different physiological parameters at the same time. One of the major uses of dual-isotope imaging in SPECT is for determining myocardial ischemia using ^{201}Tl and ^{99m}Tc-sestamibi (for example [162, 163]). Many other uses, however, have also been suggested. For example, ventilation-perfusion studies of the lungs [164] can be performed using ^{81m}Kr gas to show open air passages and ^{99m}Tc-labelled microspheres to show blood perfusion. Additionally, regional cerebral blood flow has been examined using a combination of ^{99m}Tc-HMPAO[1] and ^{123}I-IMP[2] (for example [165]) and a ^{123}I-^{201}Tl combination has been used to determine parathyroid pathology [166].

One major advantage of dual isotope studies is that they can reduce the time required to perform a diagnostic scan. With stress-rest heart studies, for instance, it is possible

[1] Hexamethylpropylene amine oxime

[2] Iodoamphetamine

to perform the scan using only ^{99m}Tc-sestamibi. A lengthy time is required, however, for the isotope used in the first injection to wash out and/or decay before the second acquisition is performed. Ideally, the time suggested for this study is two days, but, as this is often impractical, alternative single isotope protocols have been developed which have shortened the time between scans by varying the dose given in each injection [167]. Another possibility, though, is to use two isotopes. Because the gamma rays emitted by the two isotopes have different energies, it is possible to distinguish between them using appropriately chosen energy windows. Dual isotope procedures can greatly reduce the time required to do the diagnostic test because it is not necessary to wait until one isotope has disappeared from the patient before doing the second scan.

Some dual isotope protocols suggest sequential acquisition of the two images [167]. The contribution of photons emitted by ^{201}Tl into the ^{99m}Tc window is small ($< 2.9\%$). Therefore, one can acquire a rest image with ^{201}Tl and following this inject the patient with ^{99m}Tc-sestamibi during stress. The stress image is acquired with a second separate SPECT scan. The entire procedure requires approximately two hours.

Alternatively, the images can be acquired simultaneously. This is a major advantage of dual isotope imaging because it provides automatic image co-registration. However, while appealing, simultaneous imaging is not in widespread use because it suffers from the problem of cross-talk. Cross-talk refers to contamination of the image produced by one isotope caused by detection of photons emitted by the second isotope in the energy window of the first. A major source of cross-talk is the photons emitted from the higher energy isotope which down-scatter into the energy range of the second isotope. The closer the energies of the photons emitted by the two isotopes are, the worse is the cross-talk contamination.

We have chosen to investigate the possibility of correcting for cross-talk in simultaneous dual-isotope ^{123}I–^{99m}Tc brain imaging. These isotopes can be used for the measurement of regional cerebral blood flow (rCBF). Measurement of rCBF in SPECT has been used on patients with cerebrovascular disease (for example [168, 169]) and can provide information about at-risk patients suffering from transient ischemic attacks [170]–[172]. It is also useful in determining vasodilatory reserve which can indicate a need for surgical interventions such as extracranial-intracranial bypass [173] or evaluate a patient's tolerance for permanent vessel sacrifice [174]. One method of measuring vasodilatory reserve is to use ^{133}Xe -gas with a dilating agent such as acetazolamide. However, as ^{133}Xe is most commonly delivered as a gas, its availability is limited. An alternative is dual-isotope brain "stress-rest" test such as is proposed in [165, 172, 174, 175]. Two isotopes suggested for this are ^{99m}Tc-HMPAO and ^{123}I-IMP (or ^{123}I-HIPDM) [172].

We have chosen this type of study because it represents a situation in which there is a large degree of cross-talk in both directions between the two isotopes which can greatly degrade image quality [176, 177]. Because ^{123}I and ^{99m}Tc emit photons with very similar energies (159keV and 140keV respectively), the photopeaks of the two isotopes overlap (assuming a typical camera energy resolution of about 10%) even before the issue of scatter is considered. This makes it extremely difficult to remove cross-talk contamination.

Different methods have been suggested to correct for cross-talk. One solution [177, 178] is to determine cross-talk fractions. Similar to scatter fractions, these values indicate the ratio between the number of photons detected in the two energy windows. Cross-talk is corrected for by scaling the photopeak data by the appropriate fraction and subtracting this from the data for the second energy window. It has been shown, however, that the spatial distributions of photons from a single isotope in different energy windows are different [62] and so this approach is not accurate. A second approach is to use offset windows [172, 174, 179]. For example, [172] uses two 10% asymmetric windows. The ^{99m}Tc window extends from 126keV–140keV and the ^{123}I window from 159keV–175keV. One problem with this approach is that there is a range between the two chosen windows in which no data is acquired (in the above example, this excluded region extends from 140keV–159keV) which greatly reduces the sensitivity of the acquisition; fully half of the photopeak of each isotope is not used.

Other cross-talk correction techniques have been suggested for use with ^{201}Tl -^{99m}Tc heart imaging: multiple energy window based methods such as TEW (2.2.4) [162, 180, 181] and subtraction of a filtered version of the ^{99m}Tc-image [182]. These approaches have not, however, been tested with regards to ^{99m}Tc–^{123}I imaging and may be difficult to apply due to the similarity of the primary energies of these two isotopes.

Our approach to this problem is to accurately compute the cross-talk photon distribution for each of the two isotopes. With an accurate estimate of the contamination, the cross-talk can be subtracted from the images. This would allow the use of abutting energy windows and reduce the loss in sensitivity caused by offset windows.

With this method, an initial estimate of the source distributions of each isotope is made using the uncorrected data in each energy window. Using the spatial information about the source distribution but ignoring apparent intensities, the projections for the photopeak and cross-talk windows are generated for each of these two estimated source distributions. Our method of analytically determining photon distributions is used to generate these projections. Employing a technique such as orthogonal distance regression, the projections are then scaled to fit the experimental data. Once fit, the cross talk component can be removed, allowing a corrected image to be reconstructed. Quantitatively accurate estimates of the activity of the two sources can also be determined.

The remainder of this chapter explores the feasibility of this approach and is divided into three sections. The first section describes the experiments which we performed. This is followed by a presentation of the results which are analyzed in order to determine the absolute activity of the sources. Our conclusions regarding this study are given in the third and last section.

6.1 The Experiments

The experiments performed in this chapter were all done on two identical Siemens triple head SPECT (MS3) cameras. In all cases a low energy ultra high resolution collimator (LEUHR) was used. The camera used in the non-homogeneous experiment is the same one as was used in Chapter 5 for which the characteristics had already been determined (section 5.1). The acceptance function and intrinsic spatial resolution of the second MS3 camera were assumed to be the same.

We chose to acquire data in two abutting energy windows. The first was a 14% symmetric energy centred on ^{99m}Tc. This window extended from 130 keV to 150 keV. The width of this window was chosen so that the upper energy limit was approximately half way between the energies of the two isotopes (^{123}I emits photons with 159keV). The second energy window extended from 150 keV to 175 keV. This window was chosen to be approximately the same size as the lower energy window with a slight extension on the high energy end in order to improve the sensitivity for ^{123}I. The increase in sensitivity for ^{123}I was desired because typically the activity of ^{123}I in a brain scan is lower than the ^{99m}Tc activity. The ratio of ^{99m}Tc to ^{123}I activity is normally between 3:1 and 7:1 [177]. The choice of windows selected for this study is very similar to those in [174] but with each window widened to increase sensitivity.

The three phantom configurations used in these experiments are shown in Figure 6.1. The cases we considered were:

1. Two small spherical sources were placed in a homogeneous water cylinder with

a radius of 11.1cm and a height of 21cm. The first sphere (0.46cm radius) was filled with 19.4 MBq of ^{123}I and placed at position 1, 5.8cm off-axis in the water cylinder as shown in Figure 6.1A. The second sphere (0.64cm radius) was filled with 70.2 MBq of ^{99m}Tc and placed at position 2, 5.8cm off-axis.

2. Three spherical sources were placed in the same homogeneous water cylinder. Spheres 1 (0.46cm radius) and 3 (0.72cm radius) were filled with ^{123}I to a total activity of 19.4 MBq and 26.45 MBq respectively. They were placed at positions 1 and 3 in Figure 6.1B. Sphere 2 (0.64cm radius) contained 70.2 MBq of ^{99m}Tc and was located at position 2 (Figure 6.1B). The remainder of the phantom was filled uniformly with ^{99m}Tc at an activity concentration of 0.106 MBq/ml for a total activity of 679 MBq.

3. The configuration was the same as in case 2 but with an air bottle (3.5cm radius) inserted on one side of the phantom (Position 4 in Figure 6.1B). Spheres 1 and 3 were filled with 10.1 MBq and 9.8 MBq of ^{123}I respectively while sphere 2 with filled with 35.2 MBq of ^{99m}Tc. The background activity concentration was 0.051 MBq/ml of ^{99m}Tc(total activity of 330 MBq).

The first two experiments (cases 1 and 2) were run sequentially and the sources were calibrated at the start of the first experiment. The third experiment (case 3) was performed separately.

In order to obtain good statistics, data was acquired for two hours in cases 1 and 2 and for a total of four hours in case 3. In all cases, projections were acquired at $\pm 30^o$ and $\pm 90^o$ (the locations of the detector for the -30^o and -90^o projections are shown on Figure 6.1.

In all cases the data in two projections was acquired simultaneously using two of the three camera heads. The third head was retracted as far as possible in order to reduce

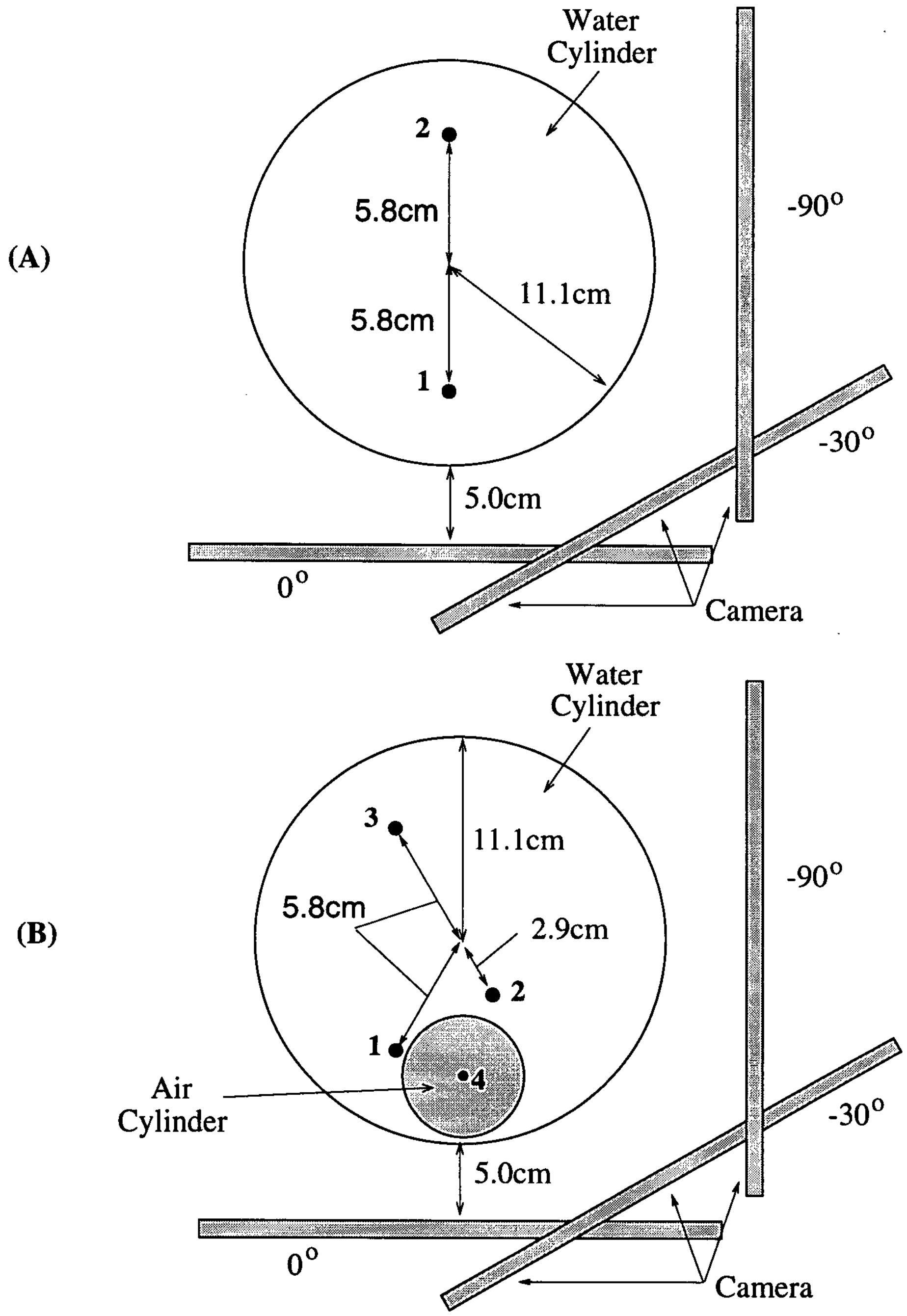

Figure 6.1: The phantom configurations for case 1 (A) and for cases 2 and 3 (B). In figure (B), the air cylinder insert is present for case 3 but not for case 2. The air cylinder is centred 7.05cm off-axis from the water cylinder placing it against the wall of the enclosing cylinder. The air cylinder has a radius of 3.5cm, a length of 10.6cm, touches the ^{123}I source (sphere 1), and is centred 35° clockwise from sphere 2. Spheres 1 and 3 are positioned 60° and 180° clockwise respectively from sphere 2.

back-scatter. Data was not acquired with the third head because of the interference of the patient bed (the bed was not modelled in the AP calculations) which was used to support the phantom during the scan. Each projection was initially acquired in a 512x512 array but later collapsed to a 64x64 array for comparison with the AP calculations. Each pixel in the 64x64 array is 0.72x0.72cm^2.

6.2 Analysis

For this study, the spatial distribution of the sources and the distribution of the attenuation/scattering materials are assumed to be known. In a clinical situation, the spatial distribution of the sources would be estimated from a reconstruction of the complete SPECT scan with the assumption that cross-talk does not significantly alter the images. The map of the attenuating material would be obtained from a preferably simultaneous transmission scan or, in the case of brain imaging, it may be sufficient to use a generic model of the medium.

The projections are generated for each source distribution. For each source, the photon distribution is determined in both energy windows. When calibrating the efficiencies of the camera heads, the calculated projections are then scaled by the number of photons emitted by the source during the data acquisition. When determining the activity of the sources, the projections are scaled by the camera head efficiencies.

As discussed in Chapter 5, back-scatter from the other camera heads is also detected and should be corrected for in order to obtain quantitatively accurate values. We correct for the back-scatter in these experiments by fitting a log-linear slope to the extreme wings of the profiles (the region of the profile which is beyond the boundaries of the water cylinder). The back-scatter within the cylinder is approximated by linearly interpolating between the back-scatter levels at the edges of the cylinder and then applying a small correction factor which is fit by the computer. In all of the remaining figures in this chapter, the AP profiles are shown both with and without back-scatter correction.

To fit our calculations to the experimental data, we used the code ODRPACK (Version 2.01) which is a Fortran based software package for weighted orthogonal distance regression. It minimizes the sum of the squared weighted orthogonal distance between a set of observations and the surface determined by the fitting parameters. The package is described in [183]. The algorithm it uses is presented in detail in [184, 185] and reviewed in [186].

It was necessary to calibrate the intrinsic efficiencies of the camera heads for the two cameras used in these experiments. Therefore, experiment one (case 1) was used to calibrate the camera for experiment two (case 2). For experiment three (case 3), the data was acquired in two independent sets at each camera head position. The first set was used to calibrate the camera head efficiencies and the second to assess quantitatively the estimated source activities.

6.2.1 Two Sphere Calibration

The first experiment was performed using a homogeneous water phantom with one ^{123}I source and one ^{99m}Tc source in a cold background (case 1). This experiment was used to calibrate the efficiencies of the camera heads. The efficiencies were determined using each projection individually and also by simultaneously fitting all projections corresponding to a single camera head.

The results of the efficiency fits are shown in Table 6.1. For this experiment the data at $+30^o$ and -90^o were acquired for two hours. The camera was then repositioned and the -30^o and $+90^o$ projection data were also acquired for two hours. The camera was then returned to the first position for an additional half hour acquisition. The efficiency factors determined from this last half hour acquisition are consistent with the earlier two

Table 6.1: ^{123}I–^{99m}Tc cross-talk efficiency factors for two spheres in a cold homogeneous background. The parameters k_{II} and k_{ITc} are the efficiency factors for ^{123}I photons being detected in the higher (150–175keV) ^{123}I and lower (130–150keV) ^{99m}Tc energy windows respectively while k_{TcI} and k_{TcTc} are the corresponding factors for ^{99m}Tc photons. The second set of $+30^o/-90^o$ projections, acquired during the last half hour scan, are superscripted with a '†'.

Projection	Head	k_{II}	k_{TcI}	k_{ITc}	k_{TcTc}
-30^o	1	0.952±0.019	0.60±0.15	1.38±0.03	0.654±0.01
$+30^o$	1	0.977±0.015	0.86±0.097	1.34±0.037	0.642±0.01
-90^o	3	0.97±0.02	1.4±0.07	0.849±0.079	0.599±0.012
$+90^o$	2	0.954±0.015	0.631±0.035	1.30±0.064	0.627±0.013
$+30^{o\dagger}$	1	0.957±0.015	0.552±0.099	1.495±0.037	0.647±0.012
$-90^{o\dagger}$	3	0.965±0.024	1.57±0.089	0.787±0.08	0.593±0.013
Simultaneous	1	0.966±0.016	0.734±0.099	1.41±0.033	0.651±0.011
Simultaneous	3	0.968±0.024	1.418±0.075	0.860±0.078	0.600±0.013

hour acquisition indicating stability of the camera over the duration of the experiment.

The values in Table 6.1 corresponding to acquisition of the photopeak window data, k_{II} and k_{TcTc}, are more consistent than the efficiencies determined for the non-photopeak windows where the statistics from the cross-talk distributions are much lower. Also, many of the values in Table 6.1 differ significantly from the expected value of 70–75%. There are a couple of possible sources for these discrepancies.

One source is the energy calibration of the camera. With the cameras we were using, it was not possible to calibrate the two energy windows simultaneously and consequently one or both of the windows may have been positioned incorrectly. Erroneously positioned windows would cause the ranges of photon energies recorded in the two windows during the experimental acquisitions to be shifted slightly from the ranges used for the AP calculations. This may have caused the discrepencies seen in Table 6.1 between our measured values and the expected efficiencies.

A second possibility is related to the method of calculation used to determine the photon distributions. Our calculations assume that the detected photon energy is distributed normally about the true energy with a FWHM of 10%. The true distribution is not exactly Gaussian. A deviation from the normal distribution will cause changes in the apparent efficiency which will be more noticeable with the windows positioned off the photopeak.

It will also be noted that the efficiencies for the camera heads differ significantly from one another (note for instance the cross-talk efficiencies for heads 1 and 3). This is due in part to differences in camera characteristics but also possibly related to the fact that the Siemens cameras calibrate the positions of the energy peaks for each detector separately.

Examples of the fits acquired using these factors are shown in Figure 6.2. This figure shows the horizontal, semi-log profile through the +30° projection. To reduce the noise in the wings of the experimental data, the three rows centred on the peaks were summed together.

6.2.2 Homogeneous Phantom

Using the mean efficiency factors determined in section 6.2.1, the absolute source activities for the homogeneous phantom with three sources in a warm background (case 2) were evaluated. This experiment was performed immediately following the efficiency calibration experiment. Table 6.2 contains the true activities of the sources used, as determined with a well counter and corrected for decay and acquisition times. These activities are given as billions (10^9) of photons emitted by the source during the acquisition and have been decay corrected.

The AP calculations were fit to the experimental data using the efficiency factors from the simultaneous fits of camera heads 1 and 3 and the fit for camera head 2 (Table 6.1). Each of the four projections was first fit individually. Secondly, the -30° and $+90^\circ$

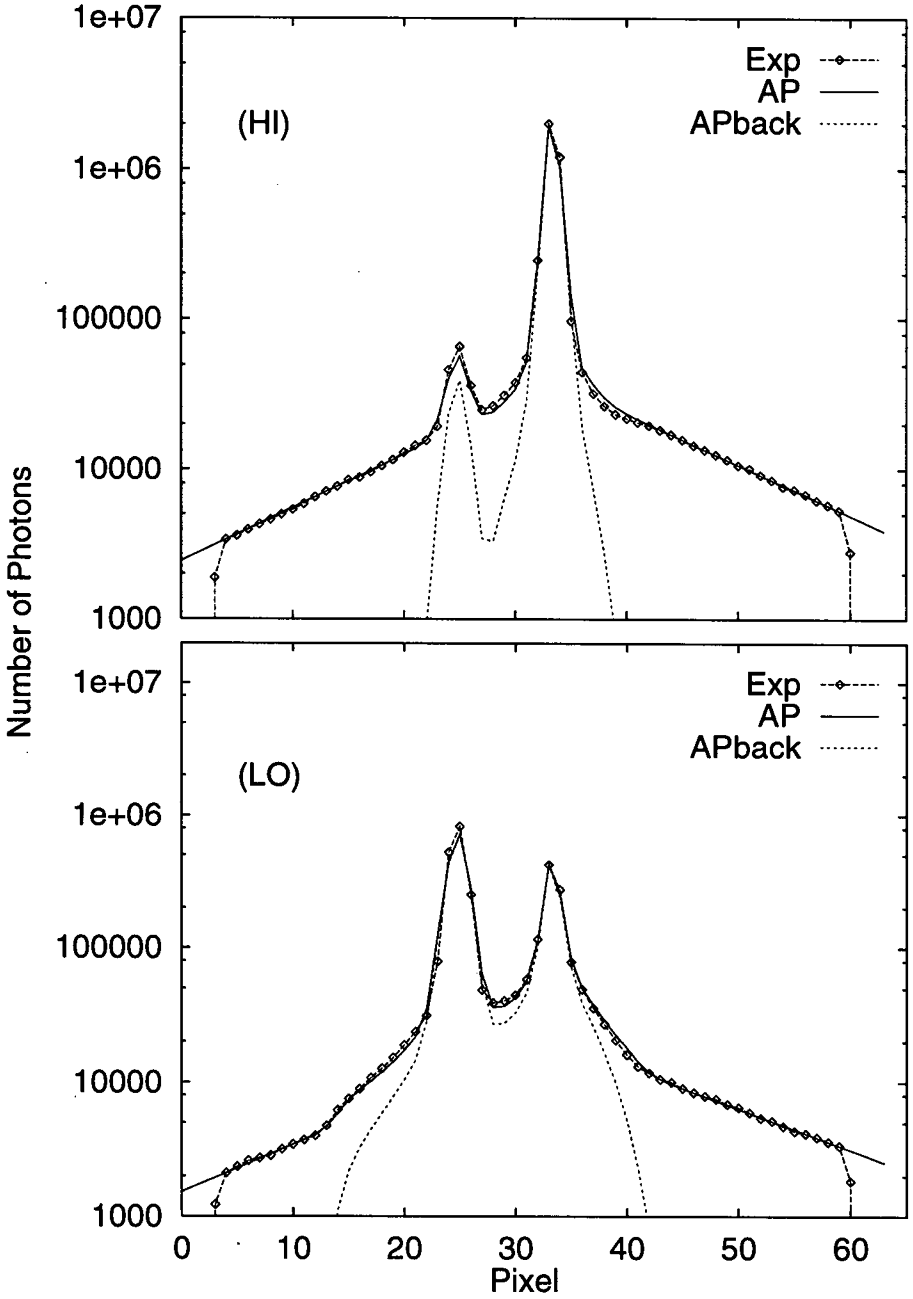

Figure 6.2: Profiles of the +30° projection for the two sphere dual-isotope calibration experiment. Shown are the profiles in both the high (HI) and low (LO) energy windows. The experimental results are labelled with *Exp* and the AP calculations with *AP*. The AP calculation corrected for the back-scatter of photons off the other camera heads is shown by the *APback* line.

Table 6.2: True number of photons ($x10^9$) emitted by the ^{123}I and ^{99m}Tc sources during the experiment using the phantom configuration in case 2. The numbers have been decay corrected and adjusted for the acquisition time. Two scans were performed and data acquired first at −30° and +90° and then at +30° and −90°.

Projection Angle	^{123}I Sphere 1	^{99m}Tc Sphere 2	^{123}I Sphere 3	^{99m}Tc Background
−30° / +90°	96.30	226.12	131.53	2206.54
+30° / −90°	85.79	176.27	117.18	1720.08

Table 6.3: Estimated number of photons (×10^9) emitted by the ^{123}I and ^{99m}Tc sources during the experiment using the phantom configuration in case 2 as determined by the ODRPACK fitting routine (see text).

Projection Angle	^{123}I Sphere 1	^{99m}Tc Sphere 2	^{123}I Sphere 3	^{99m}Tc Background
−30°	98.7±2.5	248.8±6.4	120.9±9.8	2244.6±54.7
+90°	97.2±3.5	257.2±13.4	127.7±4.2	2225.6±130.6
+30°	87.9±1.9	182.5±5.4	117.0±4.5	1800.5±59.1
−90°	90.9±3.3	211.8±4.9	128.5±3.9	1828.3±100.1
Scan 1	96.4±2.9	234.6±7.5	127.7±4.0	2194.4±89.1
Scan 2	86.7±2.4	186.5±5.2	120.2±4.2	1881.4±74.5
All	97.1±2.8	236.5±7.5	129.5±4.3	2255.6±75.4

projections were fit simultaneously as were the +30° and −90° projections. Finally, all four projections were simultaneously fit. In this last fit, the activities for the second scan are adjusted to bring them into accordance with the activities of the first scan. For example, the fit parameter for Sphere 1 is scaled by 96.30/85.79 so that a result of 96.30x10^9 photons corresponds to 85.79x10^9 emitted photons. The results of these fits are given in Table 6.3.

As can be seen from this table the accuracy of the determined activities improves as more of the projections are taken into account. Although the fits to the individual

Table 6.4: Percentage differences and the difference in standard deviations (s.d.) between the calculated and the true number of emitted photons for the fit using all four projections.

	^{123}I Sphere 1	^{99m}Tc Sphere 2	^{123}I Sphere 3	^{99m}Tc Background
Percent Difference	0.83	4.59	-1.54	2.22
Difference (s.d.)	0.286	1.384	-0.472	0.651

projections are roughly correct, they contain differences from the true values of up to 14%. However, the fit which was done using all four projections simultaneously is accurate within 4.6% of the true values. For this fit, the true activities all lie within 1.4 standard deviations of the fitted values. Table 6.4 summarizes the differences between these final fitted values and the true number of emitted photons.

Using the fitted activities from the simultaneous fit of all four projections and the corresponding efficiencies from Table 6.1, a final projection was created for each of the four experimentally acquired projections. Horizontal profiles through each of these four are shown in Figures 6.3–6.6. In each of the figures, the three rows centred on the peak have been summed to reduce noise in the wings of the experimental data. Each image shows the experimental data, the calculated projection, the estimate of the back-scatter, and the calculated projection corrected for the back-scatter. All of these images demonstrate excellent agreement between the results of our calculations and the experimental data.

Once it is possible to accurately predict the number and distribution of cross-talk photons in a projection, they can be corrected for. A simple method of doing this is to subtract them, either from the original data or the calculated data. One difficulty with subtraction is that it increases the noise in the image. An alternative to subtraction is, therefore, to simply use the fitted projections, without the cross-talk component in performing the image reconstruction. In either case, it is possible to produce a cross-talk

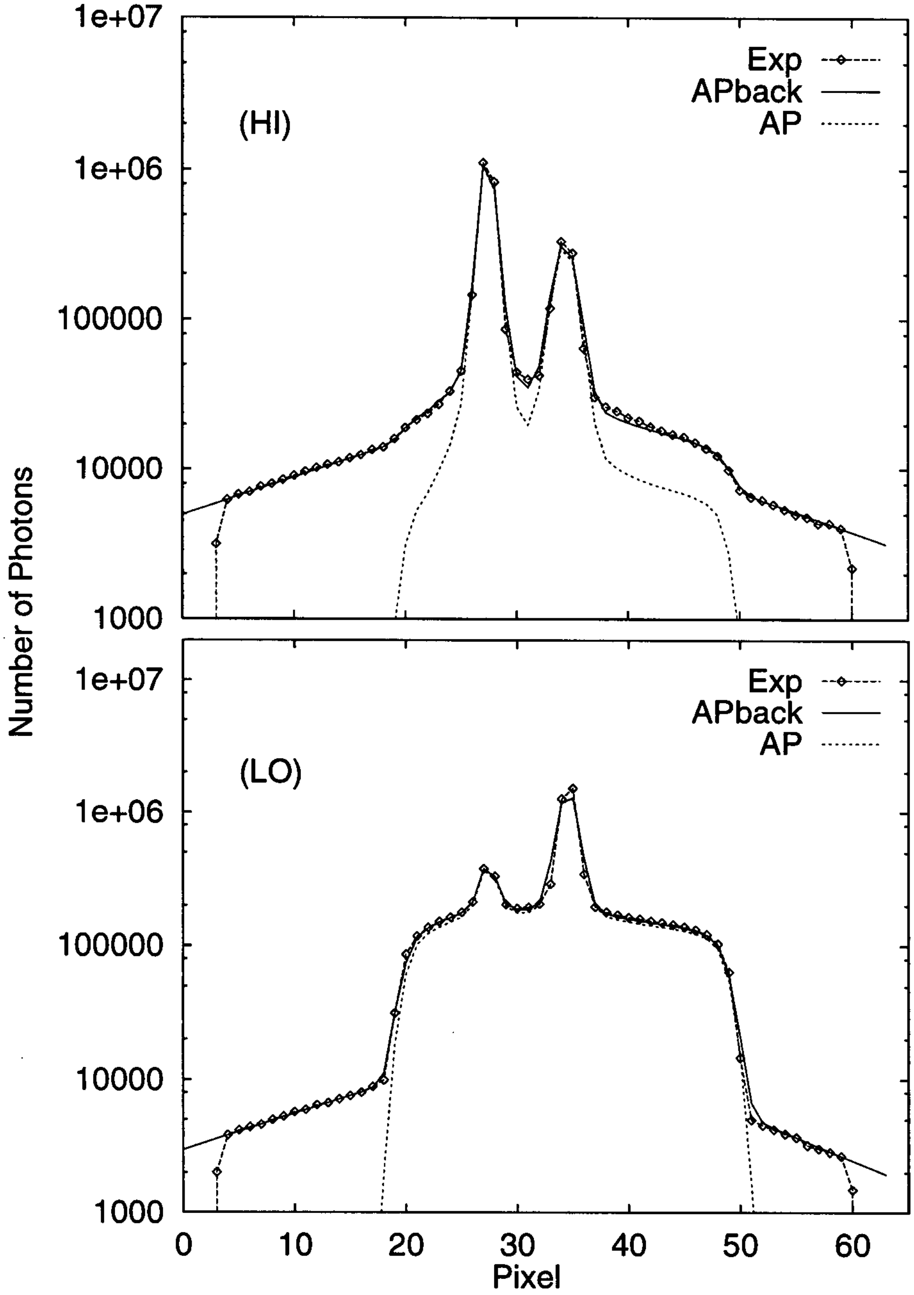

Figure 6.3: Profiles of the -30^o projection for the three sources within an active homogeneous background (case 2). Shown are the profiles in both the high (HI) and low (LO) energy windows. The experimental results are labelled with *Exp* and the AP calculations with *AP*. The AP calculation corrected for the back-scatter of photons off the other camera heads is shown by the *APback* line.

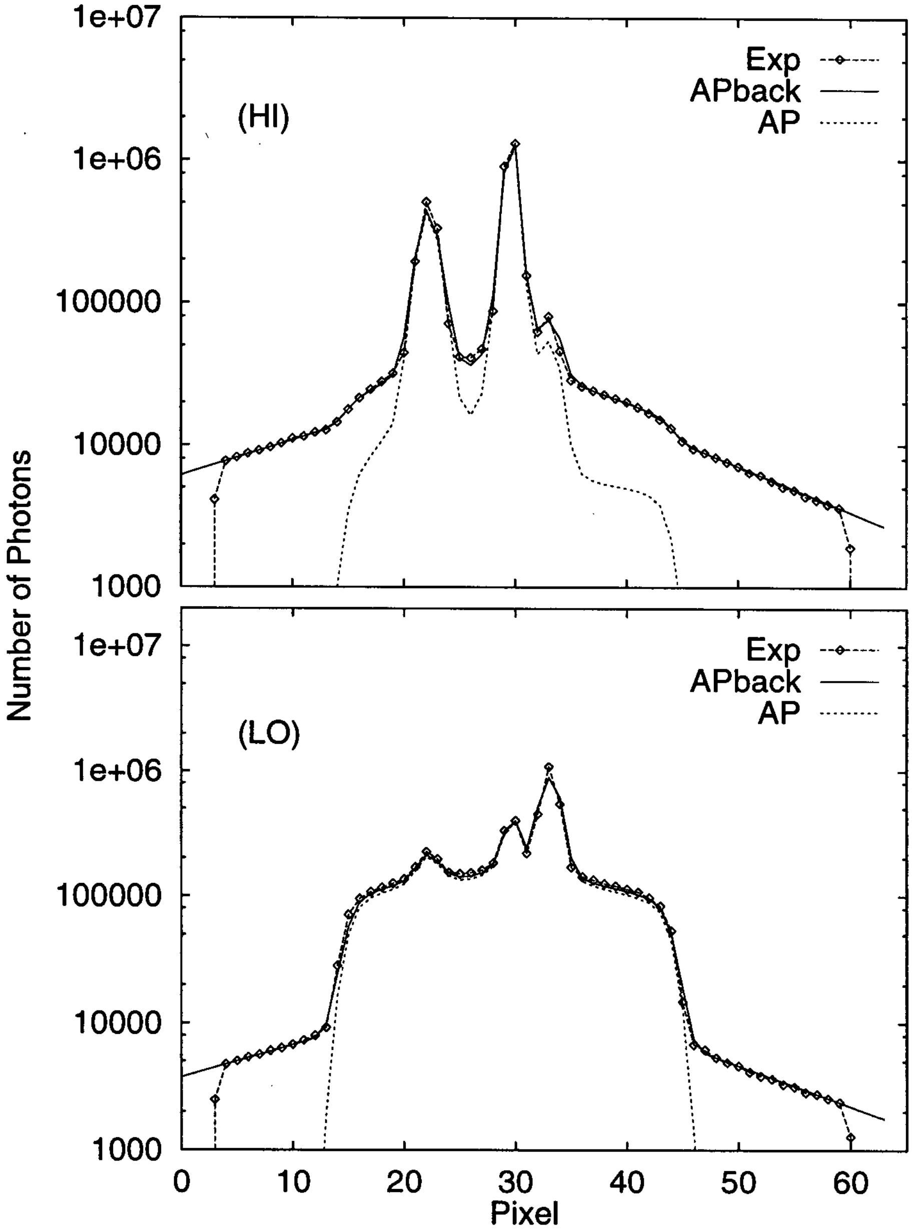

Figure 6.4: The +30° projections from the three spheres in a warm homogeneous background experiment (case 2). Shown are the profiles in both the high (HI) and low (LO) energy windows. The experimental results are labelled with *Exp* and the AP calculations with *AP*. The AP calculation corrected for the back-scatter of photons off the other camera heads is shown by the *APback* line.

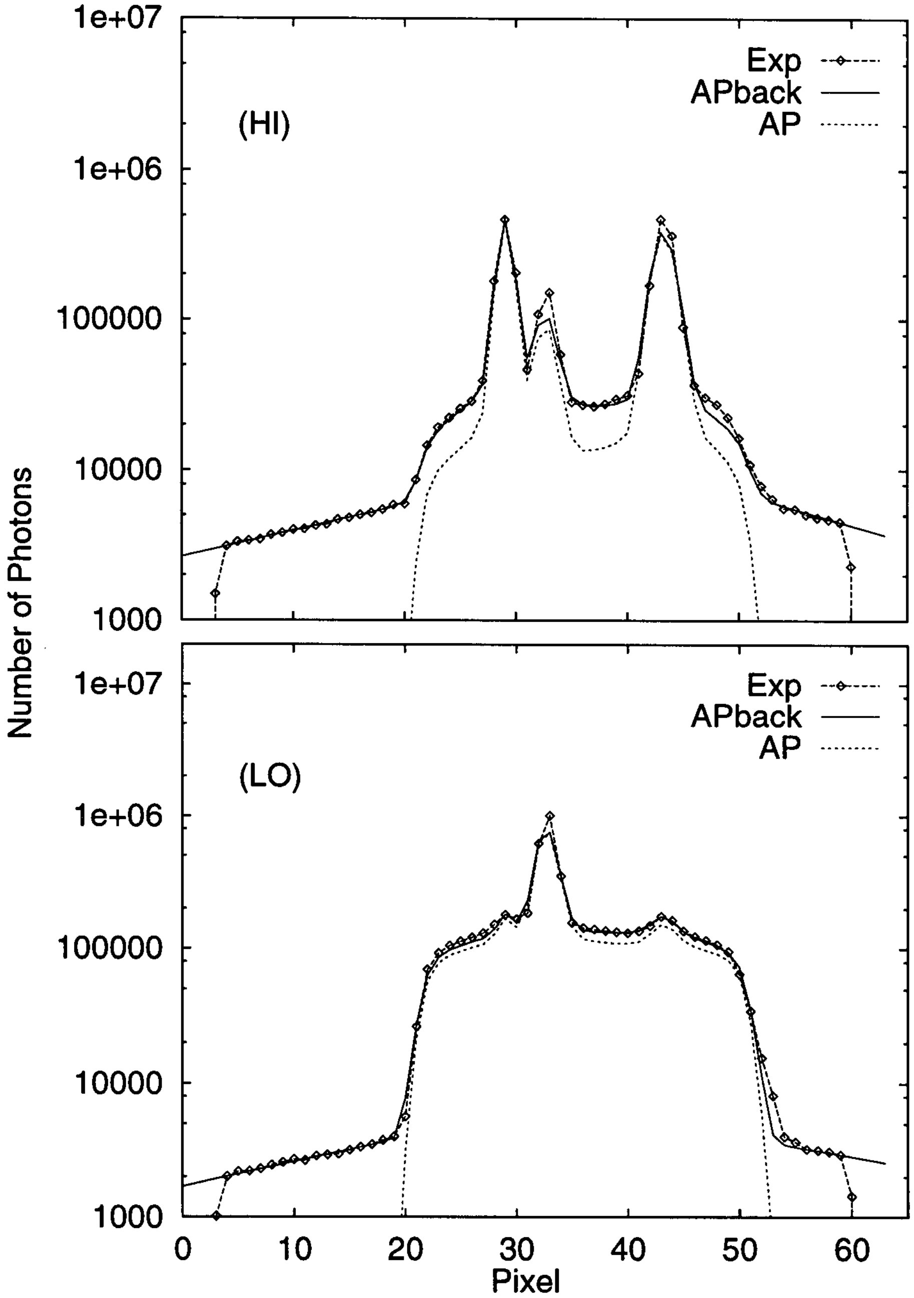

Figure 6.5: The -90^o projections from the three spheres in a warm homogeneous background experiment (case 2). Shown are the profiles in both the high (HI) and low (LO) energy windows. The experimental results are labelled with *Exp* and the AP calculations with *AP*. The AP calculation corrected for the back-scatter of photons off the other camera heads is shown by the *APback* line.

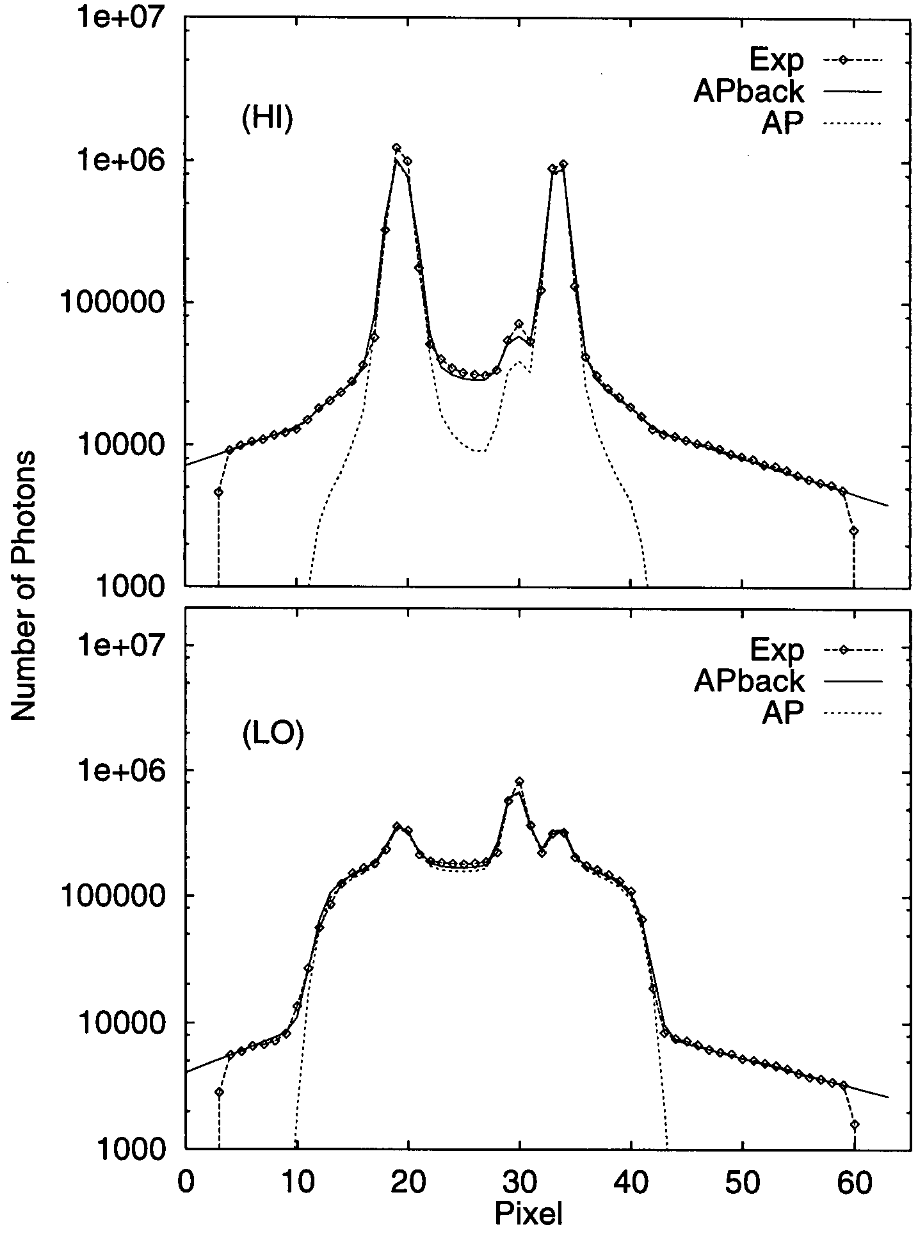

Figure 6.6: The +90° projections from the three spheres in a warm homogeneous background experiment (case 2). Shown are the profiles in both the high (HI) and low (LO) energy windows. The experimental results are labelled with *Exp* and the AP calculations with *AP*. The AP calculation corrected for the back-scatter of photons off the other camera heads is shown by the *APback* line.

free image.

As an example, Figure 6.7 shows the profile through the projection at +30°. The original projection is shown, as is the original projection with the cross-talk subtracted, and the calculated projection without the cross-talk component.

6.2.3 Non-homogeneous Phantom

The experiment with the non-homogeneous phantom (case 3) was performed on a different camera than the experiments with the homogeneous phantom. It was necessary, therefore, to recalibrate the efficiencies for the different camera heads. This was done by performing the data acquisition in two steps at each camera position. Four projections were again acquired at ±30° and ±90° by using two of the detectors at each of two different camera positions.

The data at −30° and +90° was acquired first for one hour and then for three hours and the data at +30° and −90° was acquired in two two hour blocks. The first data sets acquired at each position were used to calibrate the efficiency of the camera heads while the second data sets were used to determine the source activity. Table 6.5 gives the camera efficiencies determined using each projection separately and also by simultaneous fitting of all data associated with each head.

This camera is the same one as was used for the experiments in Chapter 5. We note that the efficiencies for the ^{99m}Tc photons being detected in the ^{99m}Tc photopeak window, k_{TcTc}, agree very well with the efficiency previously determined for the camera (68.7%) despite the differences in the width of the energy window.

It can also be noted that the efficiencies determined for heads 1 and 2, while similar to those found for heads 1 and 2 on the camera used in the homogeneous experiment, are sufficiently different that it would not be possible to use the previous values to analyse this data. This is as expected, given that the individual heads on a single camera also

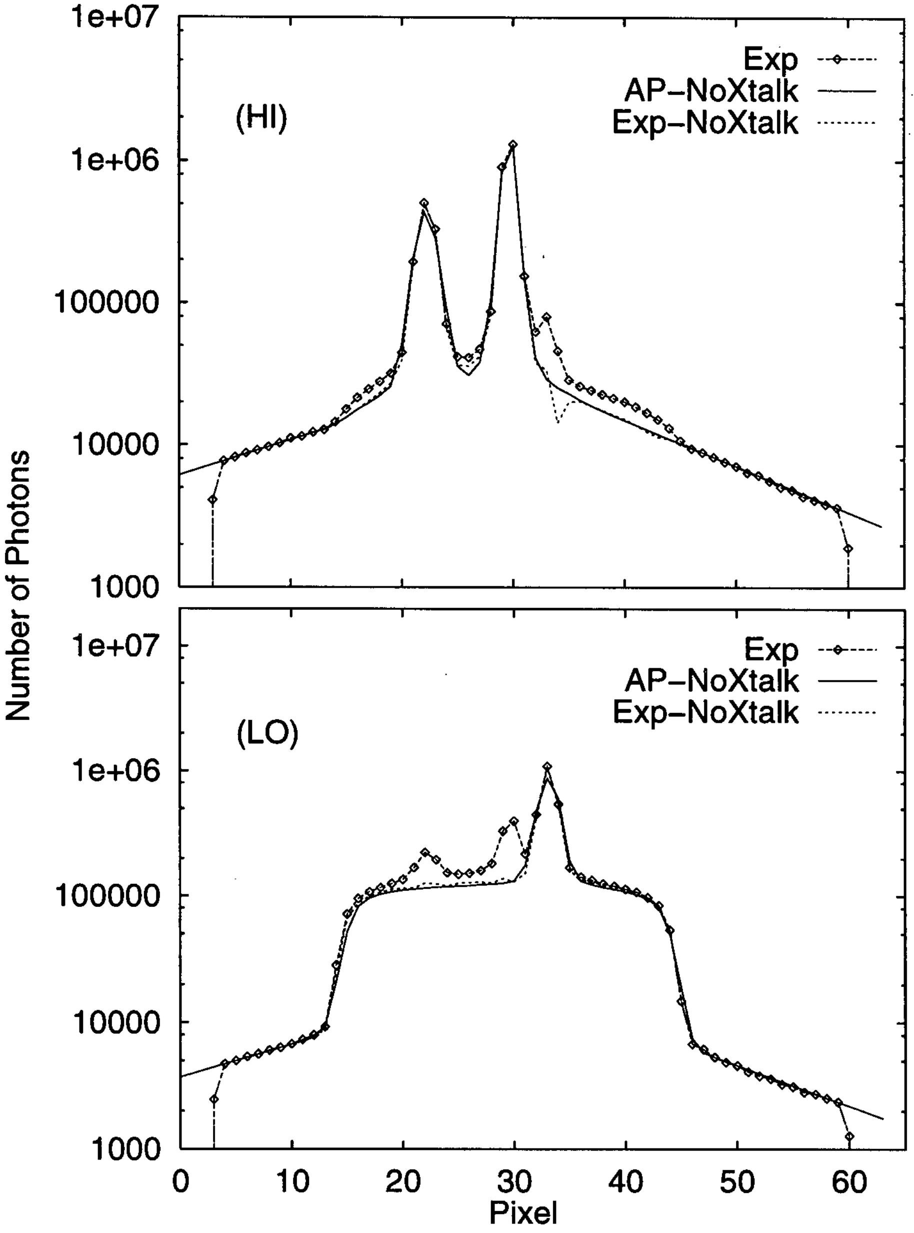

Figure 6.7: Cross-talk correction. Semi-log profiles through the +30^o projection for case 2 are shown. The *Exp* line shows the original experimental data while the AP calculation of the projection without the cross-talk components is shown with the *AP-NoXtalk* line. The experimental data with the AP calculated cross-talk subtracted off is shown with the *Exp-NoXtalk* line. Profiles are given for both the high energy (HI) and low energy (LO) windows.

Table 6.5: ^{123}I–^{99m}Tc cross-talk efficiency factors for the non-homogeneous phantom experiment (case 3).

Projection	Head	k_{II}	k_{TcI}	k_{ITc}	k_{TcTc}
−30°	1	0.785±0.021	0.588±0.127	1.354±0.093	0.627±0.010
+30°	2	0.913±0.015	0.662±0.034	1.313±0.185	0.661±0.015
−90°	1	0.820±0.012	0.543±0.042	1.309±0.097	0.663±0.009
+90°	2	0.862±0.019	0.846±0.124	1.162±0.084	0.659±0.012
Simultaneous	1	0.802±0.017	0.514±0.085	1.374±0.083	0.653±0.010
Simultaneous	2	0.893±0.017	0.696±0.056	1.286±0.135	0.668±0.014

Table 6.6: True number of photons (×10^9) emitted by the ^{123}I and ^{99m}Tc sources during the experiment using the phantom configuration in case 3. The numbers have been decay corrected and adjusted for the length of the acquisition. Two scans were performed and data acquired first at −30° and +90° and second at +30° and −90°.

Projection	^{123}I	^{99m}Tc	^{123}I	^{99m}Tc
Angle	Sphere 1	Sphere 2	Sphere 3	Background
−30° / +90°	86.17	242.01	83.48	2277.54
+30° / −90°	39.37	82.26	38.14	774.17

differ, and further indicates the need to properly calibrate each camera.

The activities of the sources for the second sets of data are given in Table 6.6, expressed in terms of billions of photons emitted. Similar to the homogeneous experiment, the four projections were then fit to the AP calculated projections to determine source activity. The fits were done individually to each projection, simultaneously for the two projections acquired at the same time, and also simultaneously with all four projections. For the last simultaneous fit, the fit parameters were adjusted so that the number of photons emitted corresponds to the number emitted during the three hour acquisition (Scan 1: projections −30° and +90°). The fitted activities are given in Table 6.7.

As with the homogeneous experiment, the fits obtained using multiple projections

Table 6.7: Estimated number of photons (×10^9) emitted by the ^{123}I and ^{99m}Tc sources during the experiment using the non-homogeneous phantom configuration (case 3) as determined by the ODRPACK fitting routine (see text).

Projection Angle	Head	^{123}I Sphere 1	^{99m}Tc Sphere 2	^{123}I Sphere 3	^{99m}Tc Background
−30°	1	94.53± 2.35	245.34± 5.87	110.25±10.00	2275.22±98.16
+90°	1	95.05± 3.72	281.68±11.62	96.45± 4.02	2334.18±148.93
+30°	2	41.49± 1.56	84.22± 2.33	41.36± 3.18	817.90±51.87
−90°	2	36.80± 0.96	86.83± 2.02	36.31± 1.64	731.21±27.67
Scan 1	1	89.81± 2.95	249.36± 8.68	94.95± 5.04	2402.70±135.7
Scan 2	2	39.15± 1.31	84.81± 2.49	38.19± 2.43	812.17±34.11
All	-	88.47± 3.04	249.37± 8.38	91.23± 5.20	2402.31±144.23

Table 6.8: Percentage differences and the difference in standard deviations (s.d.) between the calculated and the true number of emitted photons for the fit using all four projections.

	^{123}I Sphere 1	^{99m}Tc Sphere 2	^{123}I Sphere 3	^{99m}Tc Background
Percent Difference	2.67	3.04	9.28	5.48
Difference (s.d.)	0.757	0.878	1.490	0.865

tended to be more accurate than those found with a single projection. The number of emitted photons determined using fits to a single projection differed from the true values by up to 16%. Using all of the projections, however, resulted in differences no larger than 9.3% (1.5 standard deviations). Table 6.8 summarizes these values for the fit which was made using all four projections.

We also note that all of the resultant activities are slightly high, indicating that perhaps the efficiencies determined were slightly lower than the actual values. Increasing all of the efficiencies by 2%, approximately one standard deviation in the k_{II} and k_{TcTc} efficiency values, improves the resultant fit to the true number of photons by approximately

Table 6.9: Simultaneous fit to all four non-homogeneous projections using efficiency values scaled by +2%.

	^{123}I Sphere 1	^{99m}Tc Sphere 2	^{123}I Sphere 3	^{99m}Tc Background
Emitted photons ($\times 10^9$)	86.68±3.05	244.49±8.22	89.47±4.87	2355.26±141.51
Percent Difference	0.59	1.02	7.18	3.41
Difference (s.d.)	0.167	0.302	1.230	0.549

2%. These adjusted results are shown in Table 6.9.

Profiles through the four projections are shown in Figures 6.8–6.11. Each profile shows the experimental result, the AP calculated result, the estimate of the back-scatter, and the back-scatter corrected AP result. All figures demonstrate an excellent agreement between the experimental profiles and the AP calculated ones.

Once again, it is possible to correct for the cross-talk contamination either by subtraction from the experimental data or by use of the fitted AP-calculated projections with the cross-talk component exactly removed. An example of this correction for the non-homogeneous phantom experiment is shown in Figure 6.12 using the +90° projection.

6.3 Summary

Simultaneous dual-isotope scanning is a valuable technique for nuclear medicine diagnosis. It reduces the time required for diagnostic tests over sequential or single-isotope scanning and consequently reduces its cost. Additionally, it produces perfectly co-registered images which can aid in diagnostic evaluation. In this chapter, we performed a feasibility study on a technique for acquiring dual-isotope images. This approach uses data acquired in abutted energy windows and corrected for cross-talk contamination through use of an

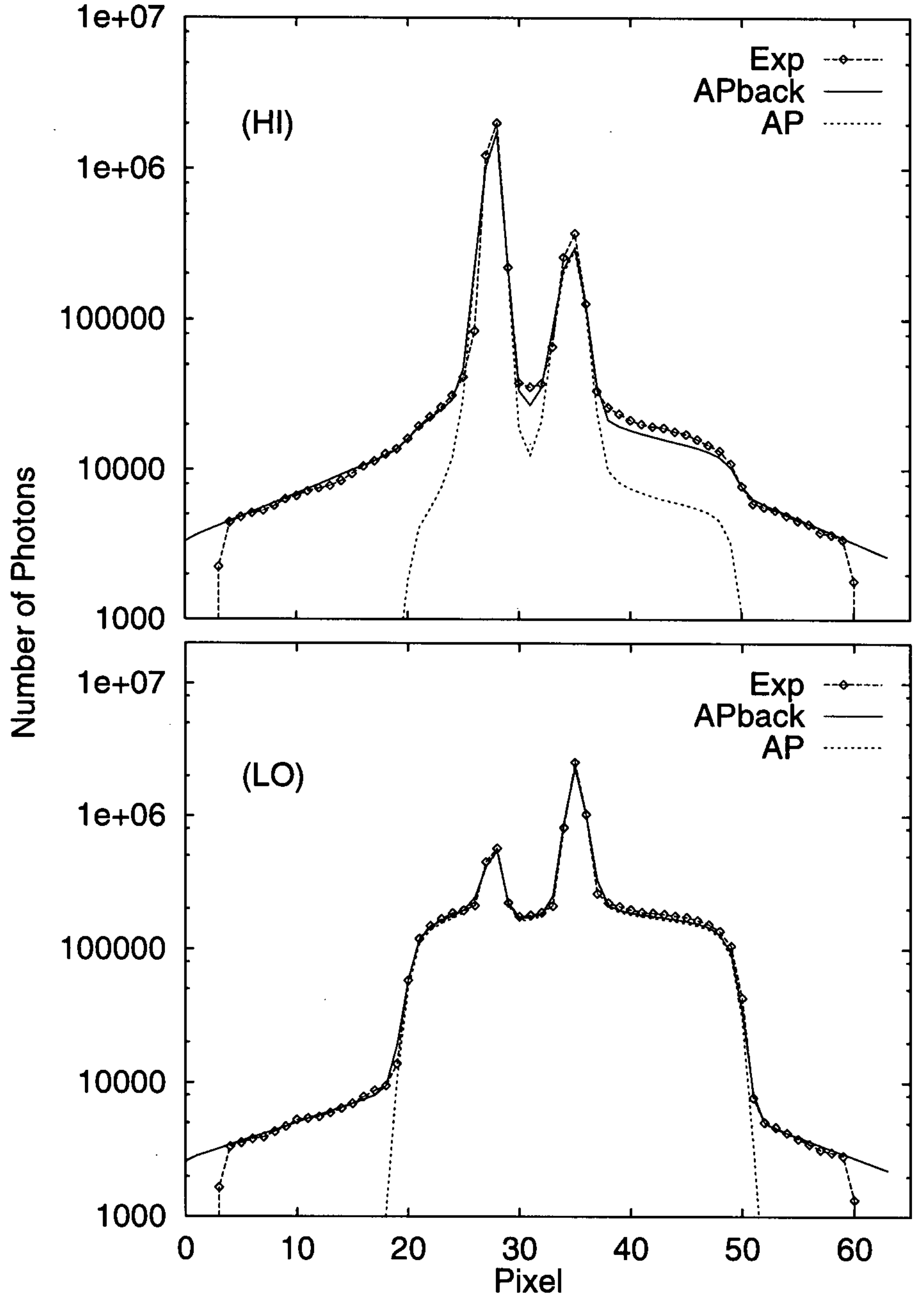

Figure 6.8: Profiles of the -30^o projection for the three sources within an active inhomogeneous background (case 3). Shown are the profiles in both the high (HI) and low (LO) energy windows. The experimental results are labelled with *Exp* and the AP calculations with *AP*. The AP calculation corrected for the back-scatter of photons off the other camera heads is shown by the *APback* line.

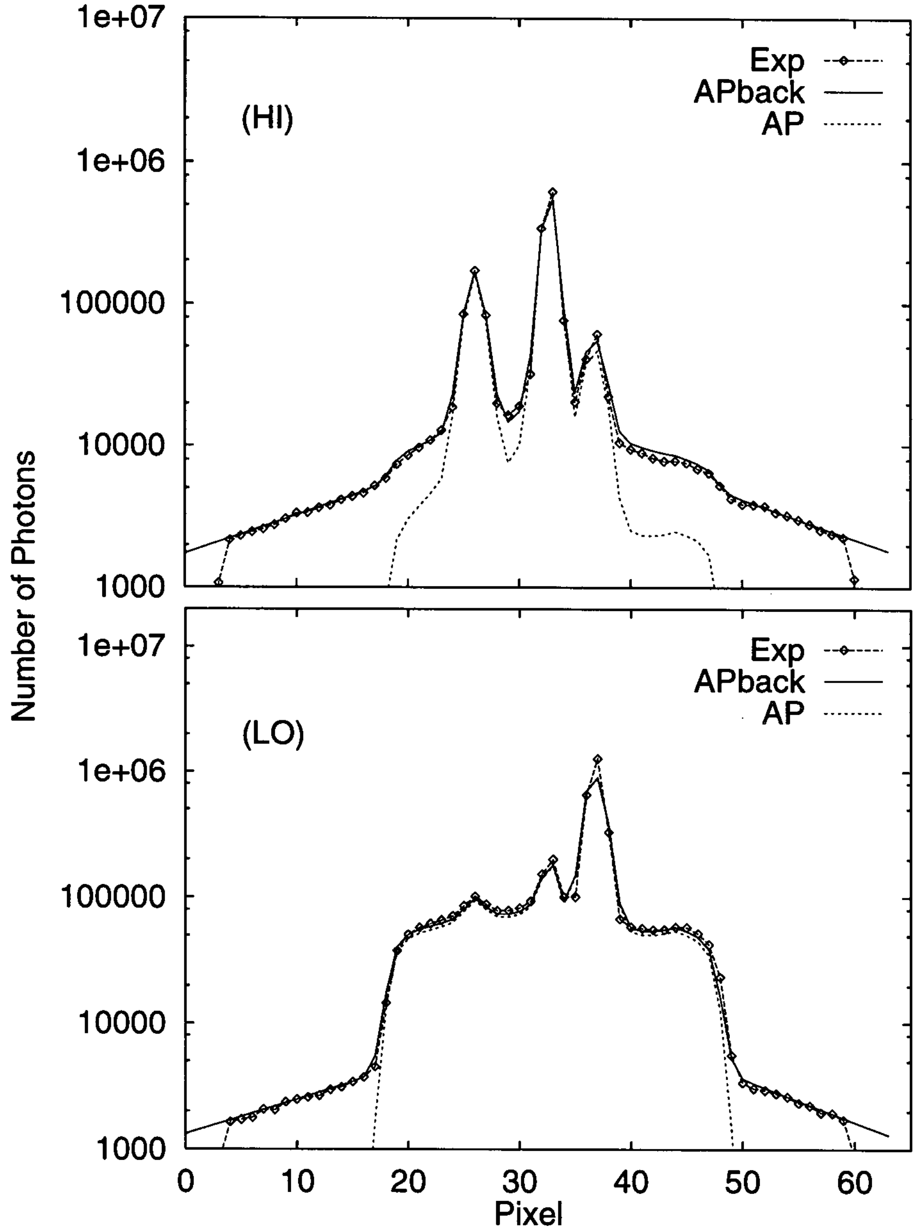

Figure 6.9: The +30° projections from the three spheres in a warm inhomogeneous background experiment (case 3). Shown are the profiles in both the high (HI) and low (LO) energy windows. The experimental results are labelled with *Exp* and the AP calculations with *AP*. The AP calculation corrected for the back-scatter of photons off the other camera heads is shown by the *APback* line.

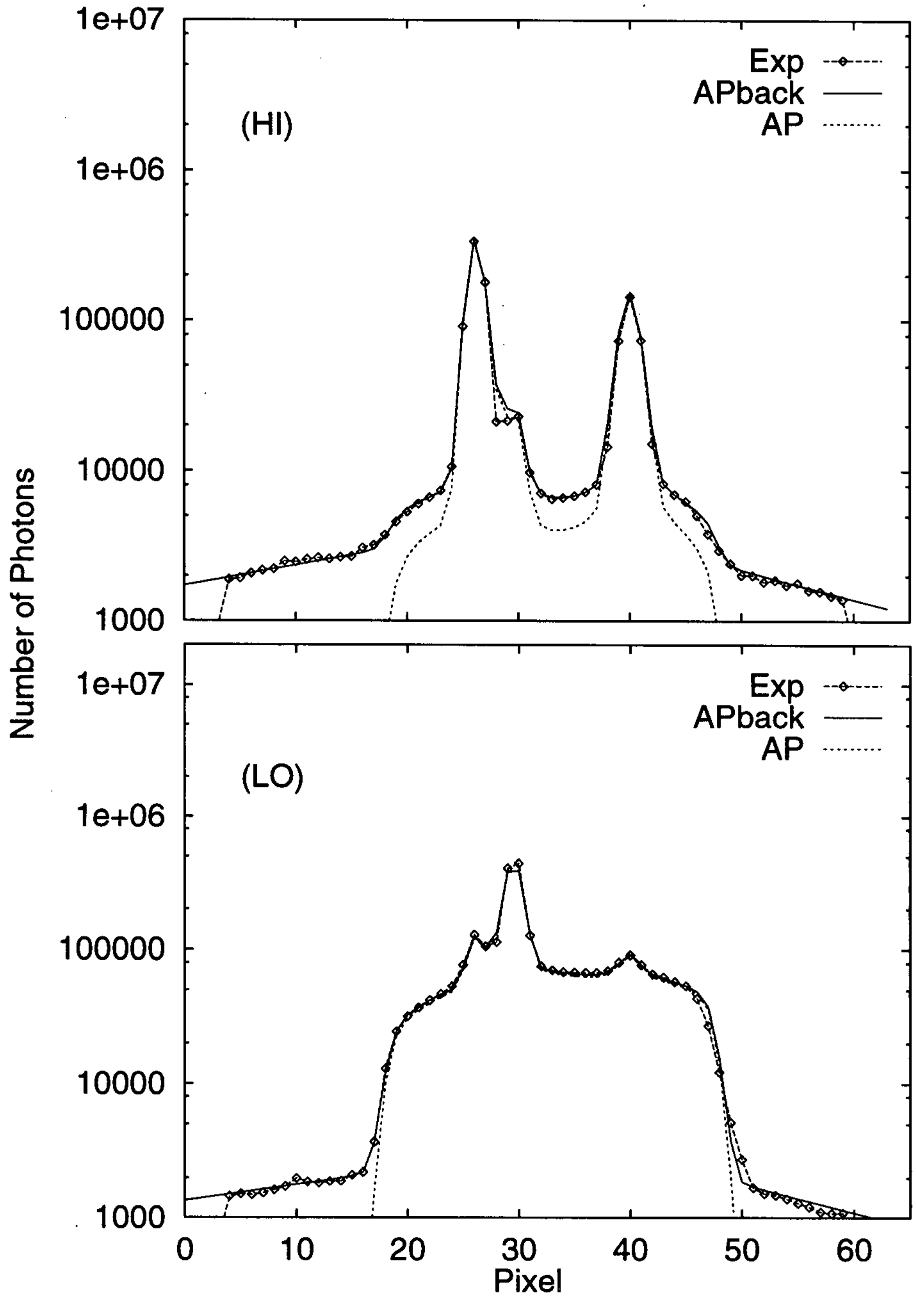

Figure 6.10: The -90^o projections from the three spheres in a warm inhomogeneous background experiment (case 3). Shown are the profiles in both the high (HI) and low (LO) energy windows. The experimental results are labelled with *Exp* and the AP calculations with *AP*. The AP calculation corrected for the back-scatter of photons off the other camera heads is shown by the *APback* line.

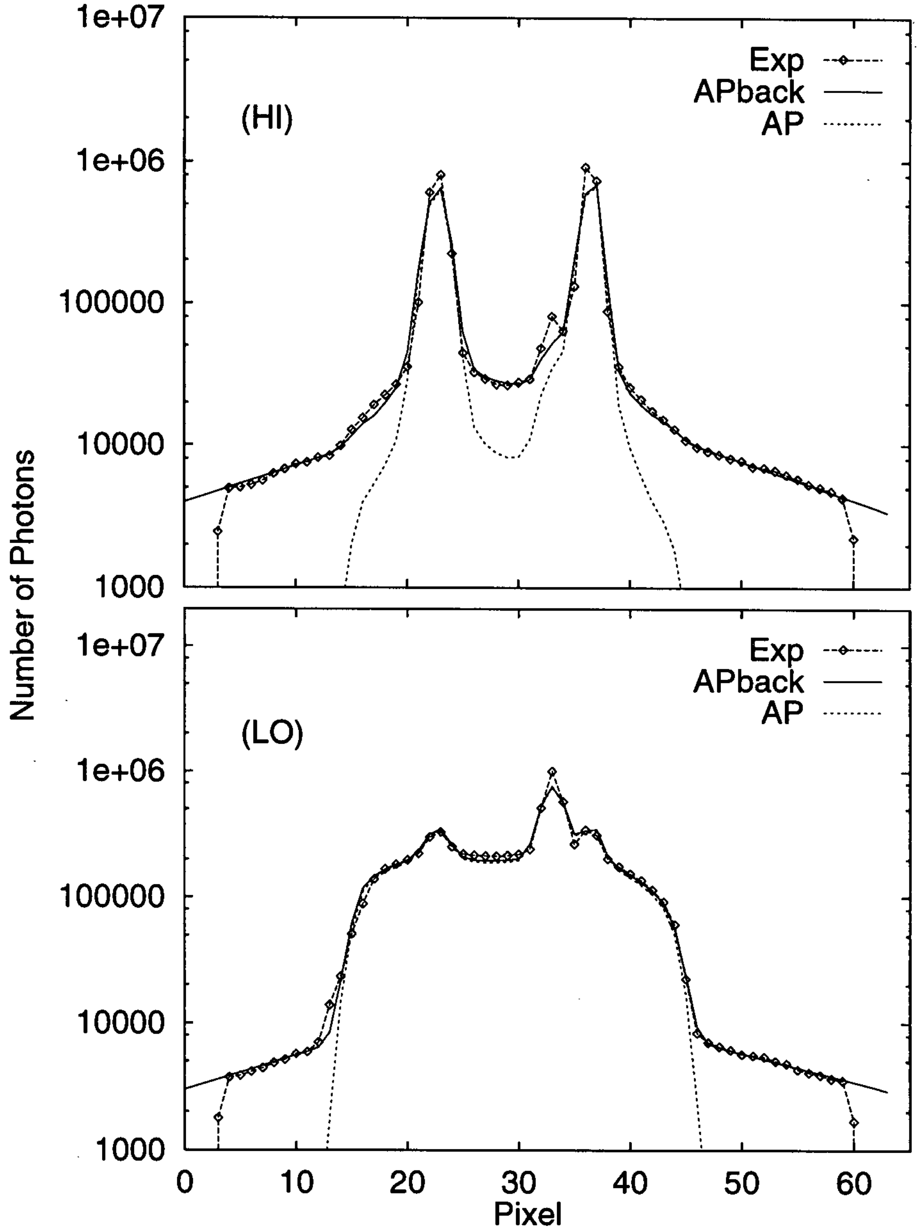

Figure 6.11: The +90^o projections from the three spheres in a warm inhomogeneous background experiment (case 3). Shown are the profiles in both the high (HI) and low (LO) energy windows. The experimental results are labelled with *Exp* and the AP calculations with *AP*. The AP calculation corrected for the back-scatter of photons off the other camera heads is shown by the *APback* line.

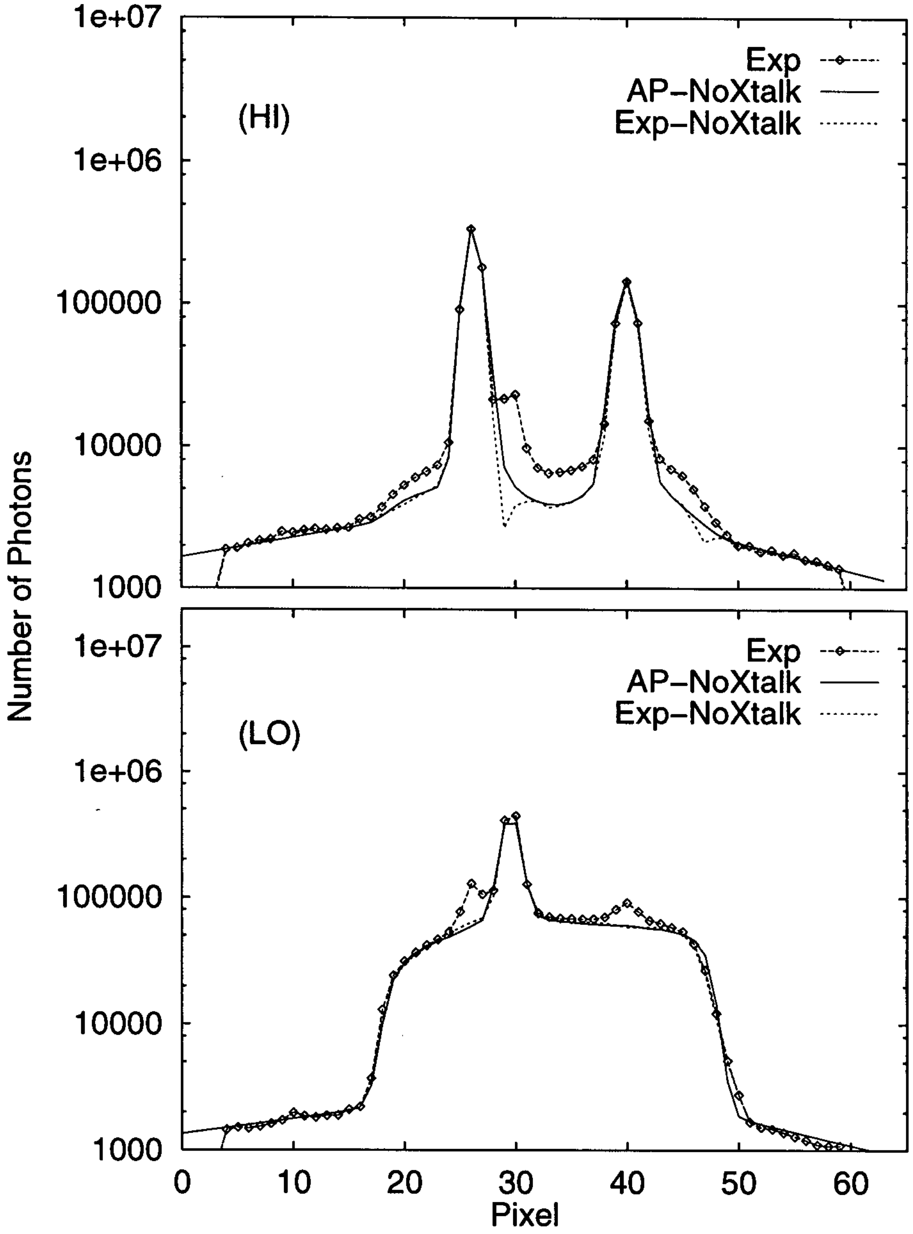

Figure 6.12: Cross-talk correction. Semi-log profiles through the -90^o projection for case 3 are shown. The *Exp* line shows the original experimental data while the AP calculation of the projection without the cross-talk components is shown with the *AP-NoXtalk* line. The experimental data with the AP calculated cross-talk subtracted off is shown with the *Exp-NoXtalk* line. Profiles are given for both the high energy (HI) and low energy (LO) windows.

accurate analytical calculation of the cross-talk photon distributions. An important advantage of this approach is that it does not reduce the camera sensitivity through the use of offset energy windows.

We performed phantom experiments using multiple small sources within both a homogeneous and a non-homogeneous medium. We were able to accurately calculate the photon distributions and consequently correct for cross-talk either through subtraction of the cross-talk estimate from the experimental data or through use of the calculated projections with the cross-talk components exactly removed. The absolute activity of the sources within phantoms were determined to within 5% of the true values for both the homogeneous and non-homogeneous cases.

Chapter 7

Conclusions

In this work, we have developed a method of analytically computing the photon distribution in SPECT projections. This last chapter begins with a brief description of the method. The summary will be followed by a discussion of the current applicability of the work indicating its capabilities and limitations. A third section will outline some of the directions that can be explored as possible future extensions of the technique. The final conclusions are presented in the fourth section.

7.1 Summary of Work

A technique has been described (Chapter 3) which permits the analytical calculation of photon distributions in SPECT projections. This technique is based upon Klein-Nishina formula for Compton (incoherent) scatter and the Rayleigh equation with associated form factors for coherent scattering. It employs precalculated camera-specific reference tables to reduce the real-time computation time of the distributions. Through use of a pixelized attenuation map, it takes into account patient dependent inhomogeneities in the attenuating/ scattering medium. The method uses physically reasonable approximations to increase the speed of the higher order scatter calculations. It also makes an approximate correction for the effects of Compton-Rayleigh interference on the coherent form factors. Though only first and second order scatter were directly computed in this work, the technique can be extended to calculate third and higher order scatter distributions if desired.

The technique has been validated with respect to two Monte Carlo simulators (EGS4 and SIMSET). It agreed with the Monte Carlo simulators to the same degree as the two Monte Carlo techniques agreed with one another (at approximately the two percent error level). Some small discrepancies which were observed near object boundaries are associated with the use of voxelized descriptions of the attenuating/scattering medium. The speed of the method was determined to be between 20 and 150 times faster than Monte Carlo simulations for small source distributions.

Our method was also validated for accuracy with respect to phantom experiments performed using a Siemens MS3 triple head SPECT camera system. The camera characteristics were evaluated experimentally and camera specific lookup tables were generated. The experimentally determined intrinsic efficiency factor of 68.7% yielded an absolute quantitative accuracy of 2% in the total number of counts recorded in the projection. The AP technique also reproduced the relative activity for a two source acquisition.

Finally, we applied our approach of calculating photon distributions to the problem of cross-talk in dual-isotope SPECT scans. Using several small sources in both homogeneous and non-homogeneous media with warm and cold backgrounds, it was shown that it is possible to determine camera efficiencies which allow accurate reproduction of photon cross-talk distributions and also accurate quantitative determination of the source strengths. An accuracy of approximately 5% in the determination of the source activities was achieved using four projections. Our ability to accurately reproduce dual-isotope projections allows for the generation of cross-talk free images through techniques such as subtraction of the cross-talk distributions from the experimental projections or through use of the calculated projections with exact removal of the cross-talk component.

7.2 Current Applicability

At present, our technique is capable of accurately reproducing photon distributions in projections acquired with SPECT cameras. It has immediate application as a possible alternative to Monte Carlo simulation as it produces a distribution with more uniform noise characteristics and does so faster than Monte Carlo for small source distributions.

The technique also has potential application as a research tool in SPECT scatter correction and in the correction of cross-talk contamination in dual-isotope (or multiple-isotope) SPECT scans. Our method produces separate distributions for the scattered and unscattered photons and as such can provide a low-error estimate of the scatter (or cross-talk) in a projection. As an example of this, consider Figures 7.1–7.2. Figure 7.1 reproduces the projection at $+30^o$ for the dual-isotope experiment using the inhomogeneous medium (case 3). Our calculation of this projection is broken down in Figure 7.2, showing the contribution of each separate component to each of the two projections. Figure 7.3 shows the experimental data corrected first for cross-talk and then for scatter and back-scatter. Also shown on this figure is our calculation of the corresponding components.

Our technique does have its limitations. Monte Carlo simulation is able to produce a high noise image quite quickly whereas our approach is a fixed time calculation. Monte Carlo does not have the *overhead cost* which we do, that is, the time needed for precalculating the look-up tables required by our method. Thus, for investigations where system parameters are being frequently altered or where only a few simulations are being performed, Monte Carlo simulation may still be the most efficient approach to use. It

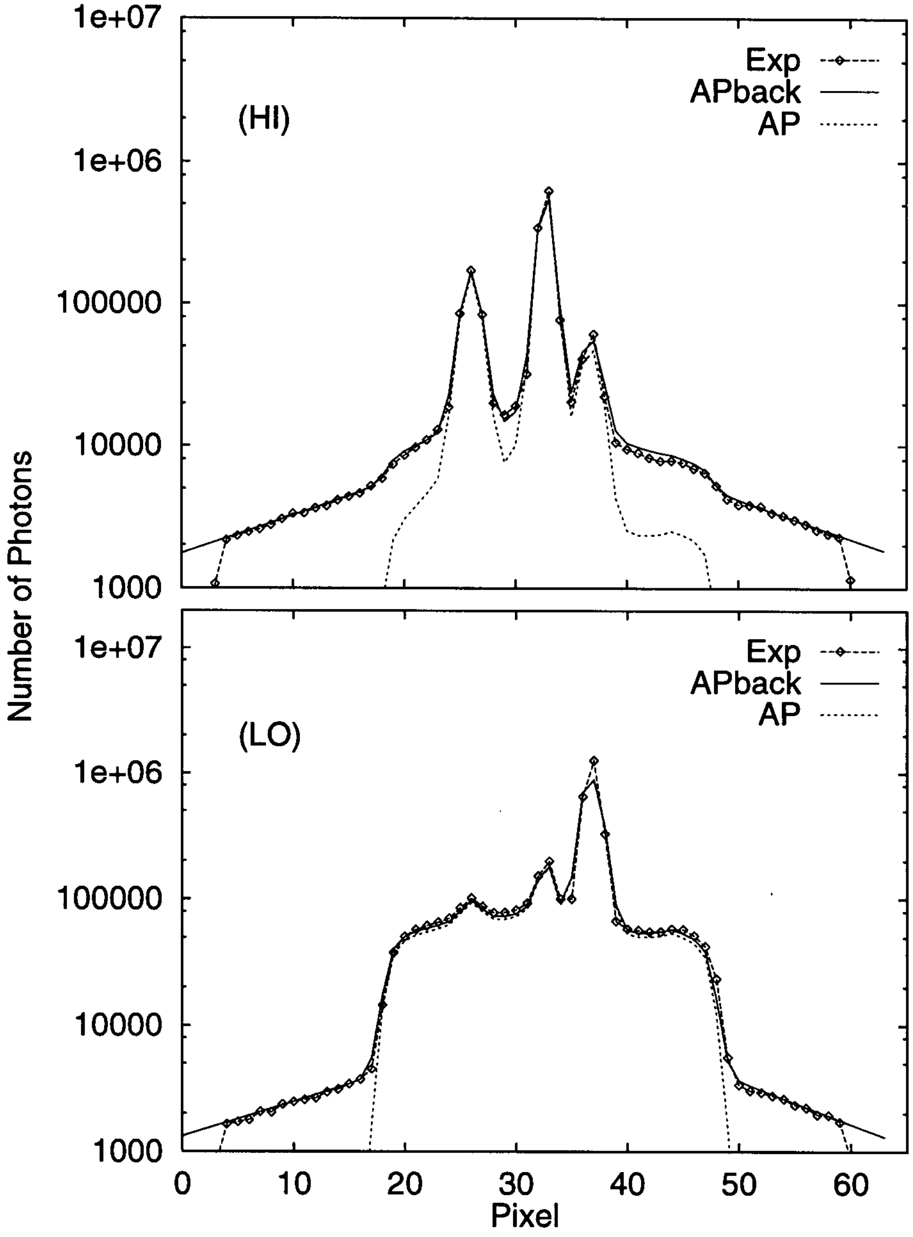

Figure 7.1: For the reader's convenience, Figure 6.9 has been reproduced here. The figure shows the agreement between the experimental results and the AP calculated results for the +30° projection from the inhomogeneous dual-isotope experiment. This projection is decomposed and corrected in Figures 7.2 and 7.3.

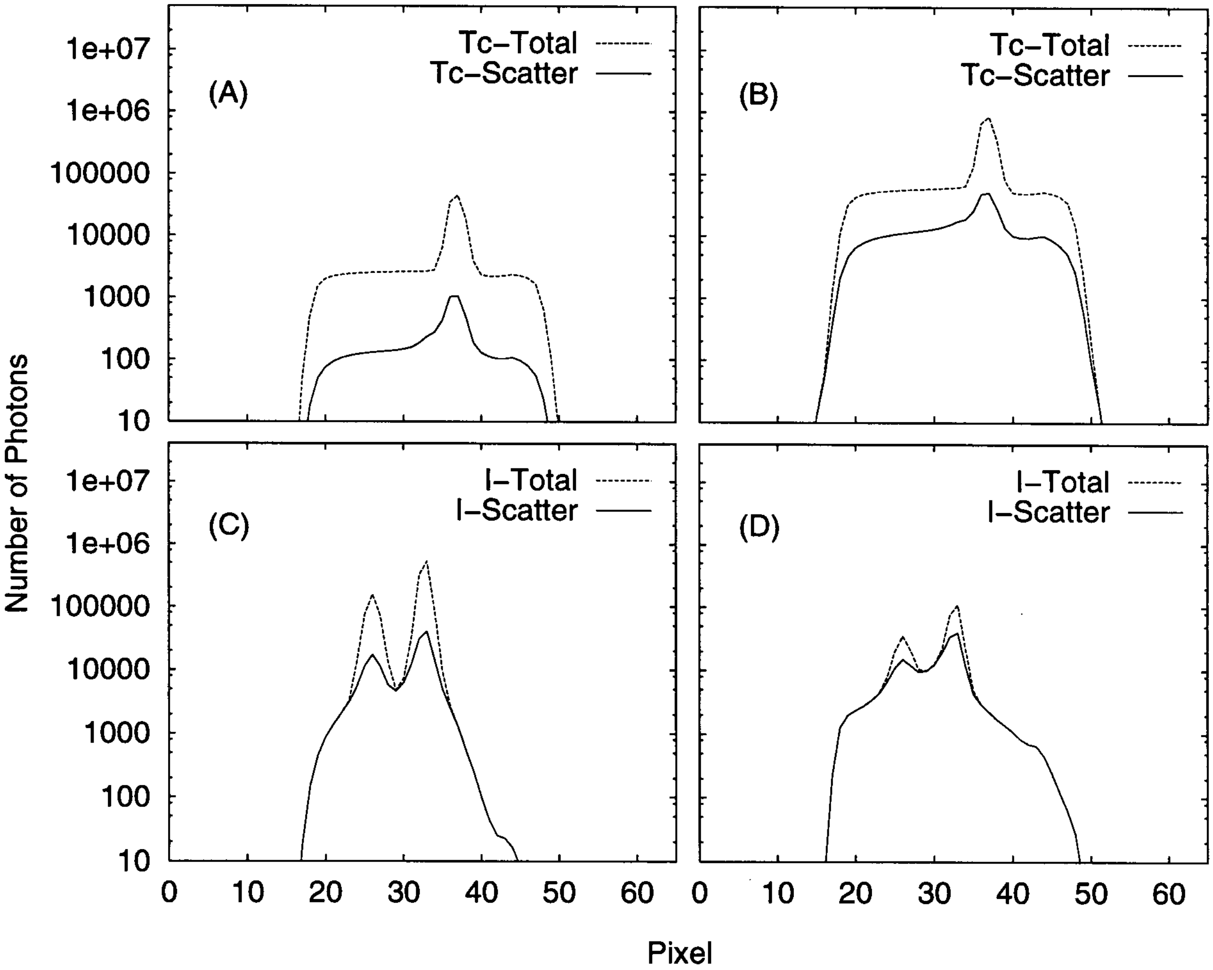

Figure 7.2: Decomposition of the AP calculated profile shown in Figure 7.1. Image (A) shows the contribution of ^{99m}Tc photons to the data collected in the ^{123}I (HI) window. The total distribution *Tc-Total* is shown as well as the scattered photon distribution *Tc-Scatter*. Image (B) shows the contribution of ^{99m}Tc into its own photopeak (LO) window. Similarly, images (C) and (D) show the ^{123}I contribution to the HI and LO energy windows respectively.

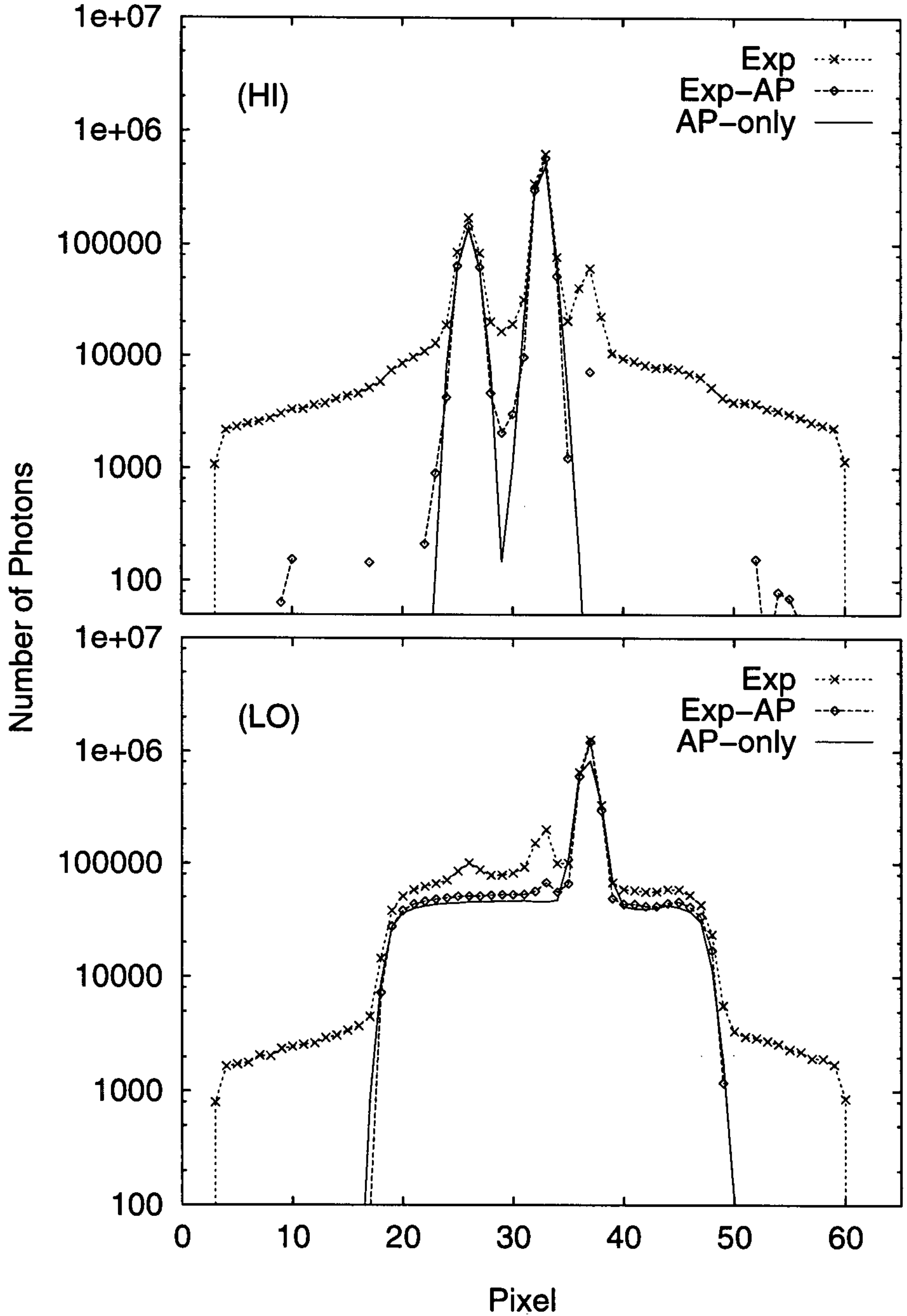

Figure 7.3: The profiles from Figure 7.1 corrected for cross-talk and scatter. The HI and LO images refer to the data obtained respectively in the ^{99m}Tc and ^{123}I photopeak energy windows. The *Exp* curve shows the original, uncorrected experimental data. The *Exp-AP* line shows the same profile with the AP calculated estimate of the cross-talk and scatter subtracted off. The *AP-only* curve shows the unscattered photon distribution with no cross-talk contamination. Although subtraction increases the noise in the image, both corrections markedly improve the accuracy of the projection data.

is when large numbers of simulations are being run on a fixed camera system, that our technique is most advantageous.

Our approach also does not presently compete well with Monte Carlo for the simulation of extended source distributions. We feel that this difficulty is not, however, inherent to the technique. The codes used in this thesis were developed for the calculation of point-like sources. Distributed sources can be broken down into a collection of point sources and, therefore, photon distributions can be calculated for them by simply repeating the point-source calculation numerous times. This simple technique is the method used in this thesis to generate projections for large sources. This approach is, however, quite slow and has in no way been optimized. By independently computing the distributions for numerous point sources within the same attenuating medium, one performs many redundant calculations such as the repeated determination of the attenuation of photons through a given section of the scattering/attenuating medium. A more efficient implementation of this technique for large sources could eliminate many of these redundant calculation and greatly improve calculation times.

Finally, although the technique is currently too slow for clinical implementation in either scatter or cross-talk correction, it has a great deal of potential in these directions. In clinical protocols, the camera system is fixed and what changes from scan to scan is the patient tissue density distribution and the source distribution. This means that only one set of look-up tables would be required and possibly periodic recalibration of the camera efficiency (implementable as part of the regular quality control regimen). The actual time of calculation issue is not as serious. The implementation of the code, as discussed above, for large sources and multiple projections has not yet been optimized and there remains much room there for speed improvements. Additionally, computer technology is rapidly advancing and the computation speed of reasonably priced readily available workstations is improving steadily. The times quoted in this thesis are for calculation on a Sun Sparc10 workstation. An Intel Pentium Pro 200MHz machine runs approximately three times faster than this workstation and is available for a few thousand dollars.

7.3 Future directions

The work presented in this thesis is complete. Nevertheless, there are still many avenues of investigation which could be followed to extend and improve upon it, reducing some of the limitations described in the previous section (section 7.2). Improvement in the calculation time required for the reference tables would greatly increase the general applicability of this method. The numerical integration techniques used to generate the values in our tables were fairly basic ones and as the functions that are being evaluated are quite smooth and regular, we feel intuitively that improvements should be possible. In the same light, optimization of the codes for extended source distributions and multiple projections would also increase the speed of the technique and hence its usefulness and even potentially render the technique clinically applicable on currently available computers.

There is also more work that could be done validating the clinical usefulness of this technique with regards to scatter and cross-talk correction. The dual-isotope investigation performed in Chapter 6 was only a feasibility study and while producing very encouraging results, it should also be extended. It is necessary to test the technique in the context of a full SPECT simulation with more clinically appropriate scan times. The times used in our feasibility study were chosen to acquire high statistics data in order to assess the accuracy of the technique. They are, however, much too long for clinical practice. Also, the sources used in the study were small and distinct. Overlapping extended sources should also be tested. Because of the difficulty in resolving this type of distribution, the ability to compute large source distributions efficiently for a full set of

SPECT projections may be required.

7.4 Final Word

The aim of this work was to produce an accurate and fast method of generating estimates of the scattered photon distribution in SPECT projections. We have succeeded in producing a method which is accurate to within a few percent of both Monte Carlo simulated results and experimentally acquired data. The technique is faster than another similar analytical technique and faster than Monte Carlo simulation for small source distributions. Code optimization for extended sources and multiple projections will soon bring the speed of this approach to a level where practical clinical application is possible.

The method is applicable in many different areas of SPECT imaging. It can act as an alternative to Monte Carlo simulation for the investigation of scatter and scatter correction. Additionally, it calculates scatter (and photon) distributions which can be used in both scatter correction and dual-isotope cross-talk removal.

Bibliography

[1] I. Buvat, H. Benali, A. Todd-Pokropek, and R. Di Paola. Scatter correction in scintigraphy: the state of the art. *Eur. J. Nuc. Med.*, 21:675–694, 1994.

[2] W.R. Nelson, H. Hirayama, and D.W.O. Rogers. The EGS-4 code system. SLAC 256, Stanford Linear Accelerator Center, Stanford, California, 1985.

[3] D.R. Haynor, R.L. Harrison, T.K. Lewellen, A.N. Bice, C.P. Anson, S.B. Gillispie, R.S. Miyaoka, K.R. Pollard, and J.B. Zhu. Improving the efficiency of emission tomography simulations using variance reduction techniques. *IEEE Trans. Nuc. Sci.*, 37(2):749–753, 1990.

[4] P.J. Early and D.B. Sodee. *Principles and Practice of Nuclear Medicine.* Mosby - Year Book, Inc., Toronto, second edition, 1995.

[5] R.E. Coleman, R.A. Binder, and R.J. Jaszczak. Single photon emission computed tomography (SPECT) part II: clinical applications. *Investigative Radiology*, 21(1):1–11, 1986.

[6] M.N. Maisey, K.E. Britton, and D.L. Gilday, editors. *Clinical Nuclear Medicine.* J.B. Lippincott Co., Philadelphia, PA, USA, 2nd edition, 1991.

[7] J.W. Keyes. Perspectives on tomography. *J. Nuc. Med.*, 23:633–640, 1982.

[8] H.O. Anger. Scintillation camera. *Rev. Sci. Instrum.*, 29:27–33, 1958.

[9] H.O. Anger. Scintillation camera with multichannel collimators. *J. Nuc. Med.*, 5:515–531, 1964.

[10] J.A. Sorenson and M.E. Phelps. *Physics in Nuclear Medicine.* W.B. Saunders Company, Toronto, second edition, 1987.

[11] G.N. Ramachandran and A.V. Lakshminarayanan. Three-dimensional reconstruction from radiographs and electron micrographs: application of convolution instead of Fourier transforms. *Proc. Nat. Acad. Sci.*, 68(9):2236–2240, 1971.

[12] J.A. Parker. *Image Reconstruction in Radiology.* CRC Press, Inc., Boston, 1990.

[13] R.A. Brooks and G. Di Chiro. Principles of computer assisted tomography (CAT) in radiographic and radioisotopic imaging. *Phys. Med. Biol.*, 21(5):689–732, 1976.

[14] D.A. Chesler and S.J. Riederer. Ripple suppression during reconstruction in transverse tomography. *Phys. Med. Biol.*, 20(4):632–636, 1975.

[15] R.A. Brooks, G.H. Glover, A.J. Talbert, R.L. Eisner, and F.A. DiBianca. Aliasing: a source of streaks in computed tomograms. *J. Computer Assisted Tomography*, 3(4):511–518, 1979.

[16] S.R. Deans. *The Radon transform and some of its applications.* John Wiley & Sons, Toronto, 1983.

[17] R. Gordon, R. Bender, and G.T. Herman. Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography. *J. Theor. Biol.*, 29:471–481, 1970.

[18] R. Gordon. A tutorial on ART. *IEEE Trans. Nuc. Sci.*, NS-21:78–93, 1974.

[19] M. Goitein. Three-dimensional density reconstruction from a series of two-dimensional projections. *Nucl. Instr. Meth.*, 101:509–518, 1972.

[20] A. Formiconi, A. Pupi, and A. Passeri. Compensation of spatial system response in SPECT with conjugate gradient reconstruction technique. *Phys. Med. Biol.*, 334:69–84, 1989.

[21] L.A. Shepp and Y. Vardi. Maximum liklihood reconstruction for emission tomography. *IEEE Trans. Med. Imag.*, MI-1:113–122, 1982.

[22] H.M. Hudson and R.S. Larkin. Accelerated image reconstruction using ordered subsets of projection data. *IEEE Trans. Med. Imag.*, 13:601–609, 1994.

[23] E.S. Chornoboy, C.J. Chen, M.I. Miller, T.R. Miller, and D.L. Snyder. An evaluation of maximum likelihood reconstruction for SPECT. *IEEE Trans. Med. Imag.*, 9(1):99–110, 1990.

[24] H. Johns and J. Cunningham. *The Physics of Radiology.* Charles C. Thomas, Springfield, Illinois, fourth edition, 1983.

[25] J.H. Hubbell, Wm.J. Veigele, E.A. Briggs, R.T. Brown, D.T. Cromer, and R.J. Howerton. Atomic form factors, incoherent scattering functions, and photon scattering cross sections. *J. Phys. Chem. Ref. Data*, 4(3):471–538, 1975.

[26] E. Storm and H.I. Israel. Photon cross sections from 1keV to 100MeV for elements Z = 1 to Z = 100. *Nuclear Data Tables*, A7:565–681, 1970.

[27] J.H. Hubbell and I. Øverbø. Relativistic atomic form factors and photon coherent scattering cross sections. *J. Phys. Chem. Ref. Data*, 8(1):69–105, 1979.

[28] C.J. Leliveld, J.G. Maas, V.R. Bom, and C.W.E. van Eijk. Monte Carlo modeling of coherent scattering: influence of interference. In *1995 IEEE Nuclear Science Symposium and Medical Imaging Conference Record*, pages 1543–1547, October 1995.

[29] K. Krane. *Modern Physics.* John Wiley & Sons, Inc., Toronto, 1983.

[30] R.J. Jaszczak, K.L. Greer, C.E. Floyd, C.C. Harris, and R.E. Coleman. Improved SPECT quantification using compensation for scattered photons. *J. Nuc. Med.*, 25(8):893–900, 1984.

[31] L.T. Chang. A method for attenuation correction in radionuclide computed tomography. *IEEE Trans. Nuc. Sci.*, NS-25:638–643, 1978.

[32] S.A. Larsson. Gamma camera emission tomography: development and properties of a multi-sectional emission computer tomography system. *Acta Radiologica Supplementum*, 363:32, 1980.

[33] R.J. Jaszczak, R.E. Coleman, and F.R. Whitehead. Physical factors affecting quantitative measurements using camera-based single photon emission computed tomography (SPECT). *IEEE Trans. Nuc. Sci.*, NS-28:69–79, 1981.

[34] C.C. Harris, K.L. Greer, R.J. Jaszczak, C.E. Floyd, E.C. Fearnow, and R.E. Coleman. Tc-99m attenuation coefficients in water-filled phantoms determined with gamma cameras. *Med. Phys.*, 11:681–685, 1984.

[35] R.K. Wu and J.A. Siegel. Absolute quantification of radioactivity using the buildup factor. *Med. Phys.*, 11:189–192, 1984.

[36] D.R. Gilland, R.J. Jaszczak, K.L. Greer, and R.E. Coleman. Quantitative SPECT reconstruction of iodine-123 data. *J. Nuc. Med.*, 32:527–533, 1991.

[37] J.A. Siegel, A.H. Maurer, R.K. Wu, B.S. Denenberg, A.K. Gash, B.A. Carabello, J.F. Spann, and L.S. Malmud. Absolute left ventricular volume by an iterative build-up factor analysis of gated radionuclide study. *Radiology*, 151:477–481, 1984.

[38] J.A. Siegel, R.K. Wu, and A.H. Maurer. The buildup factor: effect of scatter on absolute volume determination. *J. Nuc. Med.*, 26:390–394, 1985.

[39] M. Ljungberg and S-E. Strand. Attenuation correction in SPECT based on transmission studies and Monte Carlo simulations of buildup functions. *J. Nuc. Med.*, 31:493–500, 1990.

[40] M. Ljungberg and S-E. Strand. Attenuation and scatter correction in SPECT for sources in a non-homogeneous object: A Monte Carlo study. *J. Nuc. Med.*, 32:1278–1284, 1991.

[41] M.A. King, M. Coleman, B.C. Penney, and S.J. Glick. Activity quantitation in SPECT: a study of prereconstruction Metz filtering and use of the scatter degradation factor. *Med. Phys.*, 18:184–189, 1991.

[42] S.H. Manglos, R.J. Jaszczak, C.E. Floyd, L.J. Hahn, K.L. Greer, and R.E. Coleman. Nonisotropic attenuation in SPECT: phantom tests of quantitative effects and compensation techniques. *J. Nuc. Med.*, 28(10):1584–1591, 1987.

[43] B. Penney, M. King, and S. Glick. Restoration of combined conjugate images in SPECT: comparison of a new Weiner filter and the image-dependent Metz filter. *IEEE Trans. Nuc. Sci.*, 37(2):707–712, 1990.

[44] M.A. King, B.C. Penney, and S.J. Glick. An image-dependent Metz filter for nuclear medicine images. *J. Nuc. Med.*, 29:1980–1989, 1988.

[45] C. Floyd, R. Jaszczak, C. Harris, and R. Coleman. Energy and spatial distribution of multiple order Compton scatter in SPECT: a Monte Carlo investigation. *Phys. Med. Biol.*, 29:1217–1230, 1984.

[46] C. Floyd, R. Jaszczak, and E. Coleman. Scatter detection in SPECT imaging: dependence on source depth, energy, and energy window. *Phys. Med. Biol.*, 33:1075, 1988.

[47] K.R. Zasadny, K.F. Koral, C.E. Floyd Jr., and R.J. Jaszczak. Measurement of Compton scattering in phantoms by germanium detector. *IEEE Trans. Nuc. Sci.*, 37:642–646, 1990.

[48] M. Singh and C. Horne. Use of germanium detector to optimize scatter correction in SPECT. *J. Nuc. Med.*, 28:1853–1860, 1987.

[49] T.P. Sanders, T.D. Sanders, and D.E. Kuhl. Optimizing the window of an Anger camera for ^{99m}Tc. *J. Nuc. Med.*, 12:703–706, 1972.

[50] B.D. Collier, D.W. Palmer, J. Knobel, A.T. Isitman, R.S. Hellman, and J.S. Zielonka. Gamma camera energy for Tc^{99m} bone scintigraphy: effect of asymmetry on contrast resolution. *Radiology*, 151:495–497, 1984.

[51] L.S. Graham, R.L. La Fontaine, and M.A. Stein. Effects of asymmetric photopeak windows on flood field uniformity and spatial resolution of scintillation cameras. *J. Nuc. Med.*, 27:706–713, 1986.

[52] R. La Fontaine, M.A. Stein, and L.S. Graham. Cold lesions: enhanced contrast using asymmetric photopeak windows. *Radiology*, 160:255–260, 1986.

[53] W.L. Rogers, N.H. Clinthorne, J. Stamos, K.F. Koral, R. Mayans, G.F. Knoll, J. Juni, J.W. Keyes, and B.A. Harkness. Performance evaluation of SPRINT, a single photon ring tomograph for brain imaging. *J. Nuc. Med.*, 25:1013–1018, 1984.

[54] K. Koral, N. Clinthorne, and L. Rogers. Improving emission-computed-tomography quantification by Compton-scatter rejection through offset windows. *Nuc. Instr. Meth. Phys. Res.*, A242:610–614, 1986.

[55] P. Bloch and T. Sanders. Reduction of the effects of scattered radiation on a sodium iodide imaging system. *J. Nuc. Med.*, 14:67–72, 1973.

[56] C.E. Floyd, R.J. Jaszczak, C.C. Harris, K.L. Greer, and R.E. Coleman. Monte Carlo evaluation of Compton scatter subtraction in single photon emission computed tomography. *Med. Phys.*, 12:776–778, 1985.

[57] R. Jaszczak. Scatter compensation techniques for SPECT. *IEEE Trans. Nuc. Sci.*, NS-32:786–793, 1985.

[58] Z. Liang, T.G. Turkinson, D.R. Gilland, R.J. Jaszczak, and R.E. Coleman. Simultaneous compensation from attenuation, scatter and detector response for SPECT reconstruction in three dimensions. *Phys. Med. Biol.*, 37(3):587–603, 1992.

[59] J. Mas, R. Ben Younes, and R. Bidet. Improvement of quantification in SPECT studies by scatter and attenuation compensation. *Eur. J. Nuc. Med.*, 15:351–356, 1989.

[60] G. Hademenos, M. Ljungberg, M. King, and S. Glick. A Monte Carlo investigation of the dual photopeak window scatter correction method. *IEEE Trans. Nuc. Sci.*, 40:179–185, 1993.

[61] F.B. Atkins and R.N. Beck. Effect of scatter subtraction on image contrast. *J. Nuc. Med.*, 16:102–104, 1975.

[62] M. Ljungberg, P. Msaki, and S-E. Strand. Comparison of dual-window and convolution scatter correction techniques using the Monte Carlo method. *Phys. Med. Biol.*, 35:1099–1110, 1990.

[63] S.R. Meikle, B.F. Hutton, D.L. Bailey, R.R. Fulton, and K. Schindhelm. SPECT scatter correction in non-homogeneous media. In A.C.F. Colchester and D.J. Hawkes, editors, *Information Processing in Medical Imaging, XII IPMI International Conference*, pages 34–44, Kent, 1991. Springler-Verlag.

[64] M. Gilardi, V. Bettinardi, A. Todd-Pokropek, L. Milanesi, and F. Fazio. Assessment and comparison of three scatter correction techniques in single photon emission computed tomography. *J. Nuc. Med.*, 29:1971, 1988.

[65] J. Yanch, M. Flower, and S. Webb. Improved quantification of radionuclide uptake using deconvolution and windowed subtraction techniques for scatter compensation in single photon emission computed tomography. *Med. Phys.*, 17:1011–1022, 1990.

[66] K.F. Koral, F.M. Swailem, S. Buchbinder, N.H. Clinthorne, W.L. Rogers, and B.M.W. Tsui. SPECT dual-energy-window compton correction: scatter multiplier required for quantitation. *J. Nuc. Med.*, 31(1):90–98, 1990.

[67] C. Lowry and M. Cooper. The problem of Compton scattering in emission tomography: a measurement of its spatial distribution. *Phys. Med. Biol.*, 32:1187, 1987.

[68] A.J. Green, S.E. Dewhurst, R.H.J. Begent, K.D. Bagshawe, and S.J. Riggs. Accurate quantification of ^{131}I distribution by gamma camera imaging. *Eur. J. Nuc. Med.*, 16:361–365, 1990.

[69] D.R. Gilland, R.J. Jaszczak, T.G. Turkington, K.L. Greer, and R.E. Coleman. Quantitative SPECT imaging with indium 111. *IEEE Trans. Nuc. Sci.*, 38:761–766, 1991.

[70] K.F. Koral, S. Buchbinder, N.H. Clinthorne, W.L. Rogers, F.M. Swailem, and B.M.W. Tsui. Influence of region of interest selection on the scatter multiplier required for quantification in dual-window Compton correction. *J. Nuc. Med.*, 32:186–187, 1991.

[71] K.W. Logan and W.D. McFarland. Single photon scatter compensation by photopeak energy distribution analysis. *IEEE Trans. Nuc. Sci.*, 37:1178–1182, 1990.

[72] M. King, G. Hademeos, and S.J. Glick. A dual photopeak window method for scatter correction. *J. Nuc. Med.*, 33:606–612, 1992.

[73] B. Penney, N. Rajeevan, H. Bushe, G. Hademenos, M. King, D. de Vries, and S. Moore. A scatter reduction method for In-111 scintigrams using five energy windows. In *1991 IEEE Nuclear Science Symposium and Medical Imaging Conference Record*, pages 1866–1873, 1991.

[74] G. Hademenos, M. King, and M. Ljungberg. Influence of phantom size, shape, and density, and collimator selection on the dual photopeak window scatter correction method. *IEEE Trans. Nuc. Sci.*, 41(1):364–368, 1994.

[75] P.H. Pretorius, A.J. van Rensburg, A. van Aswegen, M.G. Lötter, D.E. Serfontein, and C.P. Herbst. The channel ratio method of scatter correction for radionuclide image quantitation. *J. Nuc. Med.*, 34:330–335, 1993.

[76] K. Ogawa, Y. Harata, I. Ichihara, A. Kubo, and S. Hashimoto. A practical method for position-dependent Compton-scattered correction in single photon emission CT. *IEEE Trans. Med. Imag.*, 10:408–412, 1991.

[77] A. Chugo and K. Ogawa. A proposal of an accurate scatter correction method considering energy spectra of scattered photons in single photon emission CT. (preprint).

[78] K. Ogawa and N. Nishizaki. Accurate scatter compensation using neural networks in radionuclide imaging. *IEEE Trans. Nuc. Sci.*, 40:1020–1025, 1993.

[79] R.N. Beck, L.T. Zimmer, D.B. Charleston, and P.B. Hoffer. Aspects of imaging and counting in nuclear medicine using scintillation and semi conductor detectors. *IEEE Trans. Nuc. Sci.*, 19:173–178, 1972.

[80] J.R. Halama, R.E. Henkin RE, and L.E. Friend. Gamma camera radionuclide images: Improved contrast with energy-weighted acquisition. *Radiology*, 169:533–538, 1988.

[81] J. Hamill and R. DeVito. Scatter reduction with energy-weighted acquisition. *IEEE Trans. Nuc. Sci.*, 36:1334–1339, 1989.

[82] J. Hamill and R. DeVito. Gallium-67 imaging with low energy collimators and energy weighted acquisition. *IEEE Trans. Nuc. Sci.*, 37:1189–1193, 1990.

[83] R. DeVito, J. Hamill, J. Treffert, and E. Stoub. Energy-weighted acquisition of scintigraphic images using finite spatial filters. *J. Nuc. Med.*, 30:2029–2035, 1989.

[84] R. DeVito and J. Hamill. Determination of weighting functions for energy-weighted acquisition. *J. Nuc. Med.*, 32:343–349, 1991.

[85] R. Jaszczak, D. Hoffman, and R. DeVito. Variance propagation for SPECT with energy-weighted acquisition. *IEEE Trans. Nuc. Sci.*, 38:739, 1991.

[86] R. Staff, H. Gemmell, and P. Sharp. Assessment of energy-weighted acquisition in SPECT using ROC analysis. *J. Nuc. Med.*, 36(12):2352–2355, 1995.

[87] L.V. East, R.L. Phillips, and A.R. Strong. A fresh approach to NaI scintillation detector spectrum analysis. *Nucl. Instrum. Methods*, 193:147–155, 1982.

[88] K. Koral, X. Wang, L. Rogers, N. Clinthorne, and X. Wang. SPECT Compton-scattering correction by analysis of energy spectra. *J. Nuc. Med.*, 29:195–202, 1988.

[89] K. Koral, X. Wang, K. Zasadny, N. Clinthorne, W. Rogers, C. Floyd Jr., and R. Jaszczak. Testing of local gamma-ray scatter fractions determined by spectral fitting. *Phys. Med. Biol.*, 36:177–190, 1991.

[90] D. Maor, G. Berlad, Y. Chrem, A. Voil, and A. Todd-Pokropek. Klein-Nishina based energy factors for Compton free imaging (CFI). *J. Nuc. Med.*, 32:1000, 1991. (abstract).

[91] D. Haynor, R. Harrison, and T. Lewellen. Scatter correction for gamma cameras using constrained deconvolution. In *1992 IEEE Nuclear Science Symposium and Medical Imaging Conference Record*, pages 1132–1134, 1992.

[92] X. Wang and K. Koral. A regularized deconvolution-fitting method for Compton-scatter correction in SPECT. *IEEE Trans. Med. Imag.*, 11(3):351–360, 1992.

[93] X. Wang, K.F. Koral, N.H. Clinthorne, W.L. Rogers, C.E. Floyd Jr., and R.J. Jaszczak. Effect of noise, order and range in fitting the photopeak region of local, Anger-camera energy spectra. *Nucl. Instrum. Methods. Phys. Res.*, A299:548–553, 1990.

[94] D. Gagnon, A. Todd-Pokropek, A. Arsenault, and G. Dupras. Introduction to holospectral imaging in nuclear medicine for scatter subtraction. *IEEE Trans. Med. Imag.*, MI-8(3):245–250, 1989.

[95] D. Gagnon, N. Pouliot, L. Lapierrière, A. Arsenault, J. Grégoire, and G. Dupras. Post acquisition linearity correction for holospectral imaging in nuclear medicine. *IEEE Trans. Nuc. Sci.*, 37:621–626, 1990.

[96] D. Gagnon, N. Pouliot, and L. Laperrière. Statistical and physical content of low-energy photons in holospectral imaging. *IEEE Trans. Med. Imag.*, 10:284–289, 1991.

[97] A. Todd-Pokropek. Non-stationary deconvolution using a multi-resolution stack. In *Information processing in medical imaging*, pages 277–290, New York, USA, 1988. Plenum.

[98] A. Todd-Pokropek and D. Gagnon. Optimization of scatter correction techniques using energy information: how many (photon) beans make five. In *Information processing in medical imaging*, 1989. (preprint).

[99] J. Mas, P. Hannequin, R. Ben Younes, B. Bellaton, and R. Bidet. Scatter correction in planar imaging and SPECT by constrained factor analysis of dynamic structures (FADS). *Phys. Med. Biol.*, 35:1451–1465, 1990.

[100] D.C. Barber. The use of principal components in the quantitative analysis of gamma camera dynamic studies. *Phys. Med. Biol.*, 25:283–292, 1980.

[101] R. DiPaola, J.P. Bazin, F. Aubry, A. Aurengo, F. Cavailloles, J.Y. Herry, and E. Kahn. Handling of dynamic sequences in nuclear medicine. *IEEE Trans. Nuc. Sci.*, 29:1310–1321, 1982.

[102] I. Buvat, H. Benali, F. Frouin, J.P. Bazin, and R. DiPaola. Target apex-seeking in factor analysis of medical image sequences. *Phys. Med. Biol.*, 38:123–138, 1993.

[103] E. Frey, Z-W Ju, and B. Tsui. A fast projector-backprojector pair modeling the asymmetric, spatially varying scatter response function for scatter compensation in SPECT imaging. *IEEE Trans. Nuc. Sci.*, 40:1192–1197, 1993.

[104] F. Beekman, C. Kamphuis, P. van Rijk, and M. Viergever. Quantitative evaluation of fully 3D iterative non-stationary scatter compensation in SPECT. *Information Processing in Medical Imaging*, 14:53–64, 1995.

[105] C.E. Floyd, R.J. Jaszczak, K.L. Greer, and R.E. Coleman. Deconvolution of Compton scatter in SPECT. *J. Nuc. Med.*, 26(4):403–408, 1985.

[106] S. Webb, A.P. Long, R.J. Ott, M.O. Leach, and M.A. Flower. Constrained deconvolution of SPECT liver tomograms by direct digital image restoration. *Med. Phys.*, 12:53–58, 1985.

[107] B.T.A. McKee. Deconvolution of noisy data using *a priori* constraints. *Can. J. Phys.*, 67:821–826, 1989.

[108] B. Axelsson, P. Msaki, and A. Israelsson. Subtraction of Compton-scattered photons in single-photon emission computerized tomography. *J. Nuc. Med.*, 25:490–494, 1984.

[109] P. Msaki, B. Axelsson, C.M. Dahl, and S.A. Larsson. Generalized scatter correction method in SPECT using point scatter distribution functions. *J. Nuc. Med.*, 28(12):1861–1869, 1987.

[110] P. Msaki, B. Axelsson, and S. Larsson. Some physical factors influencing the accuracy of convolution scatter correction in SPECT. *Phys. Med. Biol.*, 34:283–298, 1989.

[111] J. Yanch, M. Flower, and S. Webb. A comparison of deconvolution and windowed subtraction techniques for scatter compensation in SPECT. *IEEE Trans. Med. Imag.*, 7:13–20, 1988.

[112] J. Yanch, A. Irvine, S. Webb, and M. Flower. Deconvolution of emission tomographic data: a clinical evaluation. *Brit. J. Rad.*, 61:221–225, 1988.

[113] D.L. Bailey, B.F. Hutton, S.R. Meikle, R.R. Fulton, and C.B. Jackson. An attenuation dependent scatter correction technique for SPECT. In *Information Processing in Medical Imaging*, pages 34–44, 1991.

[114] T. Mukai, J.M. Links, K.H. Douglass, and H.N. Wagner. Scatter correction in SPECT using non-uniform attenuation data. *Phys. Med. Biol.*, 33:1129–1140, 1988.

[115] Y.-L. O. An ECAT reconstruction method which corrects for attenuation and detector response. *IEEE Trans. Nuc. Sci.*, 30:632–635, 1983.

[116] P. Msaki. Position-dependent scatter response functions: will they make a difference in SPECT conducted with homogeneous cylindrical phantoms? *Phys. Med. Biol.*, 39:2319–2329, 1994.

[117] P. Msaki, K. Erlandsson, L. Svensson, and L. Nolstedt. The convolution scatter subtraction hypothesis and its validity domain in radioisotope imaging. *Phys. Med. Biol.*, 38:1359–1370, 1993.

[118] E. Frey and B. Tsui. Parameterization of the scatter response function in SPECT imaging using Monte Carlo simulation. *IEEE Trans. Nuc. Sci.*, 37:1308–1315, 1990.

[119] E. Frey and B. Tsui. A practical method for incorporating scatter in a projector-backprojector for accurate scatter compensation in SPECT. *IEEE Trans. Nuc. Sci.*, 40:1107–1116, 1993.

[120] E. Frey, B. Tsui, and M. Ljungberg. A comparison of scatter compensation methods in SPECT: subtraction-based techniques versus iterative reconstruction with accurate modeling of the scatter response. In *IEEE Conference record of the 1992 Nuclear Science Symposium and Medical Imaging Conference*, pages 1035–1037, November 1993.

[121] C.E. Frey, Z.W. Ju, and B.M.W. Tsui. Modeling the scatter response function in inhomogeneous scattering media for SPECT. *IEEE Trans. Nuc. Sci.*, NS-41:1585–1595, 1994.

[122] M. Ljungberg, M. King, and S-E. Strand. Quantitative single photon emission tomography: verification for sources in an elliptical water phantom. *Eur. J. Nuc. Med.*, 19:838–844, 1992.

[123] F. Beekman, E. Eijkman, M. Viergever, G. Borm, and E. Slijpen. Object shape dependent PSF model for SPECT imaging. *IEEE Trans. Nuc. Sci.*, 40:31–39, 1993.

[124] F. Beekman and M. Viergever. Fast SPECT simulation including object shape dependent scatter. *IEEE Trans. Med. Imag.*, 14:271–282, 1995.

[125] F.J. Beekman, E.C. Frey, C. Kamphuis, B.M.W. Tsui, and M.A. Viergever. A new phantom for fast determination of the scatter response of a gamma camera. In *IEEE Conference record of the 1993 Nuclear Science Symposium and Medical Imaging Conference*, pages 1847–1851, November 1993.

[126] M.T. Munley, C.E. Floyd, G.D. Tourassi, J.E. Bowsher, and R.E. Coleman. Out-of-plane photons in SPECT. *IEEE Trans. Nuc. Sci.*, 38(2):776–779, 1991.

[127] M. Ljungberg. *Development and evaluation of attenuation and scatter correction techniques for SPECT using the Monte Carlo method.* PhD thesis, University of Lund, Sweden, 1990.

[128] M. Ljungberg and S-E. Strand. Scatter and attenuation correction in SPECT using density maps and Monte Carlo simulated scatter functions. *J. Nuc. Med.*, 31:1579–1567, 1990.

[129] M. Ljungberg and S-E. Strand. A Monte Carlo program for the simulation of scintillation camera characteristics. *Comp. Meth. Prog. Biomed.*, 29:257–272, 1989.

[130] J. Bowsher and C. Floyd. Treatment of Compton scattering in maximum-likelihood, expectation maximization reconstructions of SPECT images. *J. Nuc. Med.*, 32:1285–1291, 1991.

[131] C.E. Floyd, R.J. Jaszczak, and R.E. Coleman. Inverse Monte Carlo: a unified reconstruction algorithm for SPECT. *IEEE Trans. Nuc. Sci.*, NS-32:779–785, 1985.

[132] Y. Narita, H. Iida, S. Eberl, and T. Nakamura. Monte Carlo evaluation of accuracy and noise properties of two scatter correction methods. In *IEEE Conference record of the 1996 Nuclear Science Symposium and Medical Imaging Conference*, pages 1434–1438, November 1996.

[133] D.J. deVries, S.C. Moore, R.E. Zimmerman, S.P. Mueller, B. Friedland, and R.C. Lanza. Development and validation of a Monte Carlo simulation of photon transport in an Anger camera. *IEEE Trans. Med. Imag.*, 9(4):430–438, 1990.

[134] M.S. Rosenthal and L.J. Henry. Scattering in uniform media. *Phys. Med. Biol.*, 35:265–274, 1990.

[135] J.C. Yanch, A.B. Dobrzeniecki, C. Ramanathan, and R. Behrman. Physically realistic Monte Carlo simulation of source, collimator and tomographic data acquisition for emission computed tomography. *Phys. Med. Biol.*, 37(4):853–870, 1992.

[136] J.W. Beck, R.J. Jaszczak, R.E. Coleman, C.F. Starmer, and L.W. Nolte. Analysis of SPECT including scatter and attenuation using sophisticated Monte Carlo modeling methods. *IEEE Trans. Nuc. Sci.*, 29(1):506–511, 1982.

[137] H. Wang, R.J. Jaszczak, and R.E. Coleman. Solid geometry-based object model for Monte Carlo simulated emission and transmission tomographic imaging systems. *IEEE Trans. Med. Imag.*, 11(3):361–372, 1992.

[138] M. Smith. A vectorized Monte Carlo code for modelling photon transport in SPECT. *Med. Phys.*, 20(4):1121–1127, 1993.

[139] M. Smith. Modelling photon transport in non-uniform media for SPECT with a vectorized Monte Carlo code. *Phys. Med. Biol.*, 38:1459–1474, 1993.

[140] C.E. Floyd Jr., R.J. Jaszczak, K.L. Greer, and R.E. Coleman. Inverse Monte Carlo as a unified reconstruction algorithm for ECT. *J. Nuc. Med.*, 27:1577–1585, 1986.

[141] S. Barney, R. Harrop, and J. Rogers. Scatter correction for positron volume imaging using analytic simulation. In *1991 IEEE Nuclear Science Symposium and Medical Imaging Conference Record*, pages 2101–2106, November 1991.

[142] J.M. Ollinger, G.C. Johns, and T.M. Burney. Model based scatter correction in three dimensions. In *1992 IEEE Nuclear Science Symposium and Medical Imaging Conference Record*, pages 1249–1251, October 1992.

[143] S.H.M. Walrand, L.R. van Elmbt, and S. Pauwels. Quantitation in SPECT using an effective model of the scattering. *Phys. Med. Biol.*, 39:719–734, 1994.

[144] J. Nuyts, H. Bosmans, and P. Suetens. An analytical model for Compton scatter in a homogeneously attenuating medium. *IEEE Trans. Med. Imag.*, 12:421–429, 1993.

[145] A. Welch, G.T. Gullberg, P.E. Christian, and F.L. Datz. A transmission-based scatter correction technique for SPECT in inhomogeneous media. *Med. Phys.*, 22(10):1627–1635, 1995.

[146] E.C. Frey and B.M.W. Tsui. A new method for modeling the spatially-variant, object-dependent scatter response function in SPECT. In *IEEE Conference record of the 1996 Nuclear Science Symposium and Medical Imaging Conference*, pages 1082–1086, November 1996.

[147] Z-J. Cao, E. Frey, and B.M.W. Tsui. A scatter model for parallel and converging beam SPECT based on the Klein-Nishina formula. *IEEE Trans. Nuc. Sci.*, 41:1594–1600, 1994.

[148] Z. Liang, J. Cheng, and J. Ye. A new model for tracing first-order Compton scatter in quantitative SPECT imaging. In *IEEE Conference record of the 1996 Nuclear Science Symposium and Medical Imaging Conference*, pages 1425–1429, November 1996.

[149] T. Riauka and Z. Gortel. Photon propagation and detection in single-photon emission computed tomography – an analytical approach. *Med. Phys.*, 21:1311–1321, 1994.

[150] T. Riauka. *Photon propagation and detection in SPECT: Theory, Experimental validation, and applications.* PhD thesis, University of Alberta, Edmonton, AB, Canada, 1995.

[151] T. Riauka, R. Hooper, and Z. Gortel. Analytical photon detection kernel for SPECT: Experimental validation for nonuniform attenuating media and considerations of 3D reconstruction kernel size. *Med. Phys.*, 99(99):1311–1321, 1996.

[152] J. Sled, A. Celler, S. Barney, and M. Ivanovic. Monte Carlo simulation in SPECT: a comparison of two approaches. In Rodney Shaw, editor, *Medical Imaging 1994: Physics of Medical Imaging.* Proc. SPIE 2163, 1994.

[153] C.E. Floyd, R.J. Jaszczak, C.C. Harris, and R.E. Coleman. Revised scatter fraction results for SPECT. *Phys. Med. Biol.*, 32(12):1663–1666, 1987.

[154] R.G. Wells, A. Celler, and R. Harrop. Analytical calculation of photon distributions in SPECT projections. *IEEE Trans. Nuc. Sci.*, 1997. To appear.

[155] C.E. Metz, F.B. Atkins, and R.N. Beck. The geometric transfer function component for scintillation camera collimators with straight parallel holes. *Phys. Med. Biol.*, 25(6):1059–1070, 1980.

[156] P.R. Bevington. *Data Reduction and Error Analysis for the Physical Sciences.* McGraw-Hill Book Company, Toronto, 1969.

[157] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes: The Art of Scientific Computing.* Cambridge University Press, New York, 1989.

[158] Y. Picard, C.J. Thompson, and S. Marrett. Improving the precision and accuracy of Monte Carlo simulation in positron emission tomography. *IEEE Trans. Nuc. Sci.*, 39:1111–1116, 1992.

[159] M. Ivanovic and D.A. Weber. Monte Carlo simulation code for SPECT imaging of uniform and non-uniform media and source distributions. In *Proceedings of the European Nuclear Medicine Congress 1991*, pages 60–63, 1991.

[160] L.R.M. Morin and A. Berroir. Calculation of x-ray single scattering in diagnostic radiology. *Phys. Med. Biol.*, 28(7):789–797, 1983.

[161] R.G. Wells, A. Celler, and R. Harrop. Experimental validation of an analytical method of calculation of photon distributions. In *1996 IEEE Nuclear Science Symposium Conference Record*, pages 1402–1406, November 1996.

[162] G.J. Hademenos, M. Dahlbom, and E.J. Hoffman. Simultaneous dual-isotope technetium-99m/thallium-201 cardiac SPET imaging using a projection-dependent spilldown correction factor. *Eur. J. Nuc. Med.*, 22:465–472, 1995.

[163] D. Delbeke, S. Videlefsky, J.A. Patton, M.G. Campbell, W.H. Martin, I. Ohana, and M.P. Sandler. Rest myocardial perfusion/metabolism imaging using simultaneous dual-isotope acquisition SPECT with technetium-99m-MIBI/ fluorine-18-FDG. *J. Nuc. Med.*, 36(11):2110–2119, 1995.

[164] A. Klumper and A. Zwijnenburg. Dual isotope (^{81m}Kr and ^{99m}Tc) SPECT in lung function diagnosis. *Phys. Med. Biol.*, 31:751–761, 1986.

[165] F.J. Bonte, M.D. Devous Sr., J.S. Reisch, A.K. Ajmani, M.F. Weiner, J. Hom, and R. Tinter. The effect of acetazolamide on regional cerebral blood flow in patients with Alzheimers's disease or stroke as measured by single-photon emission computed tomography. *Investigative Radiology*, 24(2):99–103, 1989.

[166] D.R. Neumann. Simultaneous dual-isotope SPECT imaging for the detection and characterization of parathyroid pathology. *J. Nuc. Med.*, 33:131–134, 1992.

[167] D.S. Berman, H.S. Kiat, K.F. Van Train, G. Germano, J. Maddahi, and J.D. Frieman. Myocardial perfusion imaging with technetium-99m-sestamibi: comparative analysis of available imaging protocols. *J. Nuc. Med.*, 35:681–688, 1994.

[168] R.S. Hellman and R.S. Tikofsky. An overview of the contribution of regional cerebral blood flow studies in cerebrovascular disease: is there a role for single photon emission computed tomography? *Seminars Nuc. Med.*, 20(4):303–324, 1990.

[169] E. Højer-Pedersen. Effect of acetazolamide on cerebral blood flow in subacute and chronic cerebrovascular disease. *Stroke*, 18(5):887–891, 1987.

[170] S. Vorstrup, R. Hemmingsen, L. Henriksen, H. Lindewald, H.C. Engell, and N.A. Lassen. Regional cerebral blood flow in patients with transient ischemic attacks

studied by xenon-133 inhalation and emission tomography. *Stroke*, 14(6):903–910, 1983.

[171] F. Chollet, P. Celsis, M. Clanet, B. Guiraud-Chaumeil, A. Rascol, and J-P. Marc-Vergnes. SPECT study of cerebral blood flow reactivity after acetazolamide in patients with transient ischemic attacks. *Stroke*, 20(4):458–464, 1989.

[172] M.D. Devous Sr., J.K. Payne, and J.L. Lowe. Dual-isotope brain SPECT imaging with technetium-99m and iodine-123: clinical validation using xenon-133 SPECT. *J. Nuc. Med.*, 33:1919–1924, 1992.

[173] S. Vorstrup, B. Brun, and N.A. Lassen. Evaluation of the cerebral vasodilatory capacity by the acetazolamide test before EC-IC bypass surgery in patients with occlusion of the internal carotid artery. *Stroke*, 17(6):1291–1298, 1986.

[174] D. Mathews, B.S. Walker, B.C. Allen, H. Batjer, and P.D. Purdy. Diagnostic applications of simultaneously acquired dual-isotope single-photon emission CT scans. *American J. Neuroradiology*, 15:63–71, 1994.

[175] F.J. Bonte, M.D. Devous Sr., and J.S. Reisch. The effect of acetazolamide on regional cerebral blood flow in normal human subjects as measured by single-photon emission computed tomography. *Investigative Radiology*, 23(8):564–568, 1988.

[176] D.A. Weber, M. Ivanovic, S. Loncaric, D. Sacker, and C. Wong. Feasibility of dual radionuclide SPECT imaging with I-123 and Tc-99m. *J. Nuc. Med.*, 33:851, 1992. (abstract).

[177] M. Ivanovic and D.A. Weber. Feasibility of dual radionuclide brain imaging with I-123 and Tc-99m. *Med. Phys.*, 21:667–674, 1994.

[178] J.E. Juni, R.C. Bernstein, R.A. Ponto, and P.M. Nuechterlein. Simultaneous dual-tracer brain SPECT with Tc99m-HMPAO and I-123-iodoamphetamine – Method and validation. *J. Nuc. Med.*, 32(5):956, 1991. (abstract).

[179] M.D. Devous Sr., J.L. Lowe, and J.K. Payne. Dual-isotope brain SPECT imaging with technetium and iodine-123: validation by phantom studies. *J. Nuc. Med.*, 33:2030–2035, 1992.

[180] T. Ichihara, K. Ogawa, N. Motomura, A. Kubo, and S. Hashimoto. Compton scatter compensation using the triple-energy window method for single- and dual-isotope SPECT. *J. Nuc. Med.*, 34(12):2216–2221, 1993.

[181] S.C. Moore, C. Syravanh, and D.E. Tow. Simultaneous SPECT imaging of Tl-201 and Tc-99m using four energy windows. *J. Nuc. Med.*, 34(5):188P, 1993. (abstract).

[182] K. Knesaurek. Spill correction technique for simultaneous rest Tl-201/stress Tc-99m myocardial perfusion SPECT: a phantom study. *J. Nuc. Med.*, 34(5):189P, 1993. (abstract).

[183] P.T. Boggs, R.H. Byrd, J.E. Rogers, and R.B. Schnabel. *User's Reference Guide for ODRPACK Version 2.01 Software for Weighted Orthogonal Distance Regression.* Center for Computing and Applied Mathematics, U.S. Department of Commerce, National Institute of Standards and Technology, Gaithersburg, MD 20899, June 1992. Publication NISTIR 92-4834.

[184] P.T. Boggs, R.H. Byrd, and R.B. Schnabel. A stable and efficient algorithm for nonlinear orthogonal distance regression. *SIAM J. Sci. Stat. Comput.*, 8(6):1052–1078, 1987.

[185] P.T. Boggs, R.H. Byrd, J.R. Donaldson, and R.B. Schnabel. Algorithm 676 – ODRPACK: software for weighted orthogonal distance regression. *ACM Trans. Math. Software*, 15(4):348–364, 1989.

[186] P.T. Boggs and J.E. Rogers. Orthogonal distance regression. *Contemporary Mathematics*, 112:183–194, 1990.

APPENDIX A

THE AP CALCULATION CODES

This appendix contains some of the computer codes which were developed to implement the technique discussed in this work. The code is written in the C language.

The first code is that which generates the projections in real-time using the lookup tables described in Chapter 3. The second and third code are used to generate additional reference tables for the first code. The fourth code listed here was that used to generate the attenuation and electron density maps for the phantom configurations used in Chapters 4–6. The last code included here is that which generated the main lookup tables for first order Compton scattering. The codes for generating the second order scatter and first order Rayleigh scatter are not included here as they are similar in structure to the code for generating the first order lookup table.

A.1 DUALHXP.C

This C code, entitled *dualhxp.c* generates SPECT projections in real-time. It assumes a point source distribution whose location specified by the user on the command line. The program references pregenerated lookup tables which specify the probability that a scattered photon is detected in the collimator. A number of parameters for the code can be altered in the parameter file which is read in at the time of execution. The name of the parameter file is also specified on the command line.

```
/* Program to compute the scatter from a point source */
/*      Last modified April 3rd, 1997 */
/* program dualhxp.c */
/* This code encorporates the unscattered calculation with 1st and 2nd */
/*         scatter calculation.  */
/*      Uses parameter input file 'dualhxp.param' */
/*      Does NOT do a separate Rayleigh calculation */
/*    Is for the new format lookup - tables  */

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define DIM 64
#define pi 3.1415927
#define ZERO 1.e-8
#define ZEROSQ 1.e-12
#define ZEROSTQ 1.e-15
#define SMALL 1.e-4
#define CODIST 4.5125 /* Effective Collimator thickness in cm */
                        /* Coll Thick + Crystal Thick */
#define PATHSIZE 9*DCOMAX*PATHMAX
#define MULUNGDEF 0.03929
#define MUBONEDEF 0.24767
#define RHOLUNGDEF 0.0008635
#define RHOBONEDEF 0.0052668
#define tanAA 0.043660943  /* tan(2.5 deg) */
#define SILVL 1.e-4
#define SIN 10
#define SINCNTMAX 200001
#define Kr 5
#define Krm (Kr-1)
#define LINEMAX 12
#define StrSz 80
#define StrSz2 30

#define Lookup(l1,l2,l3,l4,l5) lookup[(l1)*6*10*46+(l3)*10*46+(l4)*46+(l5)]
#define Lookup2(l1,l2,l3,l4,l5) lookup2[(l1)*6*10*46+(l3)*10*46+(l4)*46+(l5)]
#define Attarr(l1,l2,l3) attarr[(l3)*DIM*DIM+(l1)*DIM+(l2)]
#define Elecdens(l1,l2,l3) elecdens[(l3)*DIM*DIM+(l1)*DIM+(l2)]
#define LookSz 41*6*10*46
#define MaxCSO 41
#define MaxCSP 46

/* Initialization of constants and matricies and global variables*/

float *lookup, *lookup2, *pathlen, *enrgval;
short *pathdir;
float *dcsOval, *dcspval;
float *attarr, *elecdens;
float Fofxi[10001]={0.};
double psiz[3], psiz1[3];
double alpha,InitEner;
float *muvals,*rhovals,mudiff,Cradrot,COLLDIST,EVALSZ, MAXENRG;
int centx=DIM/2,centy=DIM/2,centz=DIM/2;
int DCOMAX, PATHMAX, EFSIZE;
int id1,id2,tag;
int ic2,ic1,ic0;

/* Prototype all functions */
int getattarr (float **x, char *);
int getelecdens (float **x, char *);
void getnxtsum (int l2, char *adlk1);
void getnxtsumII (int l2, char *adlk2);
void getpathlen (char *,char *,float **);
void getpathdir (char *,char *);
void getenrgval (char *,char *,float **);
void getFofxi(char *);
void initdcs0val (void);
int dcs0_2ind(float val);
void initdcspval (void);
int dcsp_2ind(float val);
void wrtatslice (int,int *);
double leffdist (int I[3],double in[3],double fnl[3],double Ener);
double meffdist (int I[3],double in[3],double fnl[3],double Ener);
double neffdist (int I[3],double in[3],double fnl[3],double *,double Ener);
double interp(double a,double fa,double b,double fb,double c
              ,double fc,double x);
double quadinterp(double a,double fa,double b,double fb,double c
              ,double fc,double x);
double findmu(float attval,double Enrg);
double etalvl(int l1,int l2, int l3, double eta, int *e,
      double dcsppix, int *d);
double etalvl2(int l1,int l2, int l3, double eta, int *e,
      double dcsppix, int *d);
double dcsplvl(int l1,int l2, int l3, int l4, double dcsppix, int *d);
double dcsplvl2(int l1,int l2, int l3, int l4, double dcsppix, int *d);
double findgfact(double *,double *, double *, double *, double *,int A);
double functn(double cs,double dc, double nil);
double trapz (double (*func)(double aa,double bb, double cc),
      double a, double b, double s,int it,double spos,
      double dpos, int dex);
void polint(double *xa,double *ya, int n, double x, double *y,double *dy);
double qcom2integ(double s, double c, double d);
double qcom3integ(double s, double c, double d);

/* Subroutines Definitions */

int getattarr (float **x, char *fname)
{ FILE *inf;
  float *xtmp,*ytmp;
  int nvals,i,res=1;
  if((inf = fopen(fname,"r"))){
      printf("Reading attenuation map: ’%s’\n",fname);
      xtmp=(float *)calloc(1,sizeof(float));
      fread(xtmp,sizeof(float),1,inf);
      nvals=irint(*xtmp);
      free(xtmp);
      xtmp=(float *)calloc(nvals,sizeof(float));
      fread(xtmp,sizeof(float),nvals,inf);
      if(nvals<3) { ytmp=(float *)calloc(3,sizeof(float));
    *ytmp= *xtmp;
    if(nvals>1) *(ytmp+1)=*(xtmp+1); else *(ytmp+1)=MULUNGDEF;
    if(nvals>2) *(ytmp+2)=*(xtmp+2); else *(ytmp+2)=MUBONEDEF;
    free(xtmp);
    xtmp=ytmp;}
      mudiff= *(xtmp+2) - *(xtmp);
      printf("    %d Values ->  H2O:%g  LUNG:%g  BONE:%g\n", nvals,
     *(xtmp), *(xtmp+1), *(xtmp+2));
      *x=xtmp;
      xtmp=(float *)calloc(5,sizeof(float));
      fread(xtmp,sizeof(float),5,inf);
      for(i=0;i<3;i++) psiz[i]=*(xtmp+i);
      printf("    Psize = (%g,%g,%g)\n",psiz[0],psiz[1],psiz[2]);
      Cradrot = *(xtmp+3);  i=irint(*(xtmp+4));
      printf("    Collimator radius of rotation:  %g (cm)\n",Cradrot);
      if(i!=DIM) printf("ERROR:  map file dimension (%d) != DIM\n",i);
      free(xtmp);
      fread(attarr,sizeof(float),DIM*DIM*DIM,inf);
      fclose(inf);}
   else{
      printf("ERROR: Trouble reading ’%s’ file\n",fname);
      res=0;}
  return(res);
}

int getelecdens (float **x, char *fname)
{ FILE *inf;
  float *xtmp,*ytmp;
  int nvals,i,res=1;

  if((inf = fopen(fname,"r"))){
      printf("Reading electron density map: ’%s’\n",fname);
      xtmp=(float *)calloc(1,sizeof(float));
      fread(xtmp,sizeof(float),1,inf);
      nvals=irint(*xtmp);
      free(xtmp);
      xtmp=(float *)calloc(nvals,sizeof(float));
      fread(xtmp,sizeof(float),nvals,inf);
      if(nvals<3) { ytmp=(float *)calloc(3,sizeof(float));
    *ytmp= *xtmp;
    if(nvals>1) *(ytmp+1)=*(xtmp+1); else *(ytmp+1)=RHOLUNGDEF;
    if(nvals>2) *(ytmp+2)=*(xtmp+2); else *(ytmp+2)=RHOBONEDEF;
    free(xtmp);
    xtmp=ytmp;}
      printf("    Electron Density H2O:%g  LUNG:%g  BONE:%g\n",
     *(xtmp), *(xtmp+1), *(xtmp+2));
      *x=xtmp;
      xtmp=(float *)calloc(5,sizeof(float));
      fread(xtmp,sizeof(float),5,inf);
      i=irint(*(xtmp+4));
      if(i!=DIM) printf("ERROR:  map file dimension (%d) != DIM\n",i);
      free(xtmp);
      fread(elecdens,sizeof(float),DIM*DIM*DIM,inf);
      fclose(inf);}
   else{
      printf("ERROR: Trouble reading '%s' file\n",fname);
      res=0;}
  return(res);
}


void getnxtsum (int l2, char *adlk1)
{ FILE *redf;
  char name[80];
  sprintf(name,"%s%d.mat%2.2d",adlk1,l2,tag);
  if((redf = fopen(name,"r"))){
/*      printf("Reading look-up table from file: '%s'\n",name);*/
      fread(lookup,LookSz,sizeof(float),redf);
     fclose(redf);}
   else{
      printf("ERROR: Unable to read look-up table: '%s' file.\n",name);
    }
}

void getnxtsumII (int l2, char *adlk2)
{ FILE *redf;
  char name[80];
  sprintf(name,"%s%d.mat%2.2d",adlk2,l2,tag);
  if((redf = fopen(name,"r"))){
/*      printf("Reading look-up table from file: '%s'\n",name);*/
      fread(lookup2,LookSz,sizeof(float),redf);
     fclose(redf);}
   else{
      printf("ERROR: Unable to read look-up table: '%s' file.\n",name);
    }
}

void getpathlen (char *root, char *fname, float **plen)
{ FILE *redf;
  char name[80];
  float *ftmp,*x;

  ftmp=(float *)calloc(3,sizeof(float));
  if(ftmp==NULL) printf("Can't allocate enough space!!!\n");
  sprintf(name,"%s/%s",root,fname);
  if((redf = fopen(name,"r"))){
/*      printf("Reading attenuation path lengths from: '%s'\n",name);*/
      fread(ftmp,3,sizeof(float),redf);
      DCOMAX= irint(*(ftmp)); PATHMAX= irint(*(ftmp+1)); COLLDIST= *(ftmp+2);
      x=(float *)calloc(PATHSIZE,sizeof(float));
      fread(x,PATHSIZE,sizeof(float),redf);
      *plen=x;
      fclose(redf);}
   else{
      printf("ERROR: Unable to read path lengths file: '%s'\n",name);
    }
  free(ftmp);
}

void getpathdir (char *root, char *fname)
{ FILE *redf;
  char name[80];
  float *ftmp;

  ftmp=(float *)calloc(3,sizeof(float));
  sprintf(name,"%s/%s",root,fname);
  if((redf = fopen(name,"r"))){
/*      printf("Reading path directions from: '%s'\n",name);*/
      fread(ftmp,3,sizeof(float),redf);
      if((irint(*ftmp)!=DCOMAX)||(irint(*(ftmp+1))!=PATHMAX)||
 (*(ftmp+2)!=COLLDIST)){
  printf("**** ERROR:  mismatch in path description headers  ****\n");
  printf(" DCOMAX: %d %d\n",DCOMAX,irint(*ftmp));
  printf(" PATHMAX:  %d %d\n",
 PATHMAX,irint(*(ftmp+1)));
  printf(" COLLDIST:  %g %g\n",
 COLLDIST,*(ftmp+2));}
      fread(pathdir,PATHSIZE,sizeof(short),redf);
     fclose(redf);}
   else{
      printf("ERROR: Unable to read path directions file: '%s'\n",name);
    }
  free(ftmp);
}

void getenrgval (char *root, char *fname, float **enrptr)
{ FILE *redf;
  char name[80];
  float *ftmp,*x;

  ftmp=(float *)calloc(3,sizeof(float));
  if(ftmp==NULL) printf("Can't allocate enough space!!!\n");
  sprintf(name,"%s/%s",root,fname);
  if((redf = fopen(name,"r"))){
/*      printf("Reading energy fractions from: '%s'\n",name);*/
      fread(ftmp,3,sizeof(float),redf);
      EFSIZE= irint(*(ftmp)); MAXENRG= *(ftmp+1);
      EVALSZ= (*ftmp - 1.)/(MAXENRG- *(ftmp+2));
      x=(float *)calloc(3*EFSIZE,sizeof(float));
      fread(x,3*EFSIZE,sizeof(float),redf);
      *enrptr=x;
      fclose(redf);}
   else{
      printf("ERROR: Unable to energy fractions file: '%s'\n",name);
    }
  free(ftmp);
}

/* Fofxi2p5_ACOS.dat is F'(cos(xi))= F(xi)*cos(xi) */
void getFofxi(char *ptfnm)
{ FILE *ptf;
  if((ptf=fopen(ptfnm,"r"))) {
/*  printf("reading '%s'\n",ptfnm); */
      fread(Fofxi,10001,sizeof(float),ptf);
      fclose(ptf);}
  else printf("PROBLEM WITH READING '%s'\n",ptfnm);
}


/* STrat old */

void initdcsOval(void)
{ int i;
  for(i=0;i<MaxCSO;i++){
      if(i>27) dcsOval[i]=(i-27)*2.+16.;
      else {
  if(i>13) dcsOval[i]=(i-11)*1.;
  if(i<11) dcsOval[i]=(i-5)*0.2;
  if(i==11) dcsOval[i]=1.5;
  if(i==12) dcsOval[i]=2.0;
  if(i==13) dcsOval[i]=2.5;}
  }
}

int dcsO_2ind(float val)
{ int res;
  if(val>16.) res=irint((val-16.)*0.5)+27;
  else{
      if (val>2.75) res=irint(val+11.);
      else { if(val>0.9) res=irint(val*2.)+8;
             else res=irint(val*5.)+5;}}
  return(res);
}

void initdcspval(void)
{ int i;
  for(i=0;i<MaxCSP;i++){
      if(i>40) dcspval[i]=(i-40)*2.+32.;
      else { if (i>10) dcspval[i]=(i-8)*1.;
     else { if(i<6) dcspval[i]=i*0.2;
    else { switch (i) {
    case 6: dcspval[i]=sqrt(2.0); break;
    case 7: dcspval[i]=sqrt(3.0); break;
    case 8: dcspval[i]=2.; break;
    case 9: dcspval[i]=sqrt(5.0); break;
    case 10: dcspval[i]=sqrt(8.0); break;}}
}}
```

```
  }
}

int dcsp_2ind(float val)
{ int res;
  if (val>32.) res=40+irint(val*0.5-16.);
  else { if(val>2.914214) res=irint(val)+8;
 else {if (val<0.9) res=irint(val*5.);
       else { res=4+irint(val*val);
      if(res>9) res=10;}
   }}
  return(res);
}


void wrtatslice (int index, int *Sp)
{ FILE *inf;
  char *fname;
  char name[30];
  float slice[DIM][DIM];
  int i,j;

  for(i=0;i<DIM;i++){
    for(j=0;j<DIM;j++){
         slice[i][j]=Attarr(i,j,index);}}
  slice[Sp[0]][Sp[1]]*=-1;
  if(slice[Sp[0]][Sp[1]]>-0.01) slice[Sp[0]][Sp[1]]=-0.01;
  sprintf(name,"atslice%d.dat",index);
  fname = name;
  printf("writing attenuation map slice #%d to '%s'\n",index,fname);
  inf = fopen(fname,"w");
  fwrite(slice,DIM*DIM,sizeof(float),inf);
  fclose(inf);
}

/* Utility programs */


int normalize(double vec[3])
{ double len=0.;
  int a,ans;
    len = sqrt(vec[0]*vec[0]+ vec[1]*vec[1]+ vec[2]*vec[2]);
    if (len<ZERO) ans=0;
      else {for(a=0;a<3;a++) vec[a]= vec[a]/len;
            ans=1;}
  return(ans);
}

double interp(double a,double fa,double b,double fb,double c
              ,double fc,double x)
{ double A,B,C,tmp,l1,fl1,tmp2;

  B=((fa-fb)*(b*b-c*c) - (fb-fc)*(a*a-b*b))/
      ((a-b)*(b*b-c*c) - (b-c)*(a*a-b*b));
  A= (fa-fb-B*(a-b))/(a*a-b*b);
  C= fa-B*a-A*a*a;
  tmp2 = A*x*x+B*x+C;
  if (tmp2<0.) tmp2=0.;
/*  if (tmp < 0.) {*/
  if(x-a>0.) { if(((b-a)>0.)&&((c-a)>0.)) {
                 if((c-a)<(b-a)) {l1=c; fl1=fc;}
    else {l1=b; fl1=fb;}}
       else { if((b-a)>0.) {l1=b; fl1=fb;}
      else {l1=c; fl1=fc;}}}
  else { if(((a-b)>0.)&&((a-c)>0.)) {
                 if((a-c)<(a-b)) {l1=c; fl1=fc;}
    else {l1=b; fl1=fb;}}
       else { if((a-b)>0.) {l1=b; fl1=fb;}
      else {l1=c; fl1=fc;}}}
  tmp = fa+(fl1-fa)/(l1-a)*(x-a);
/*  }*/
  if (tmp2>tmp) tmp=tmp2;
  return(tmp);
}

double quadinterp(double a,double fa,double b,double fb,double c
              ,double fc,double x)
{ double A,B,C,tmp;
/* double l1,fl1,l2,fl2,tmp2; */

  B=((fa-fb)*(b*b-c*c) - (fb-fc)*(a*a-b*b))/
      ((a-b)*(b*b-c*c) - (b-c)*(a*a-b*b));
  A= (fa-fb-B*(a-b))/(a*a-b*b);
  C= fa-B*a-A*a*a;
  tmp = A*x*x+B*x+C;
  if (tmp<0.) tmp=0.;
/*  if (tmp < 0.) {*/
/*  if(x-a>0.) { if(((b-a)>0.)&&((c-a)>0.)) {
                 if((c-a)<(b-a)) {l1=c; fl1=fc;}
    else {l1=b; fl1=fb;}}
       else { if((b-a)>0.) {l1=b; fl1=fb;}
      else {l1=c; fl1=fc;}}}
  else { if(((a-b)>0.)&&((a-c)>0.)) {
                 if((a-c)<(a-b)) {l1=c; fl1=fc;}
    else {l1=b; fl1=fb;}}
       else { if((a-b)>0.) {l1=b; fl1=fb;}
      else {l1=c; fl1=fc;}}}
  tmp2 = fa+(fl1-fa)/(l1-a)*(x-a);*/
/*  }*/
/*  if (tmp2>tmp) tmp=tmp2;*/
  return(tmp);
}


/* Worker subroutines  */

double neffdist (int I[3],double in[3],double fnl[3],double *watmud,
 double Enrg)
```

```
{  double dist, sgn[3]={1.,1.,1.};
   double v[3],phi,mud=0.;
   double l=0.,m,dt=0.,nxtp[3];
   double inv[3];
   int px,nI[3],cI[3];
   double sinth,sinphi,lengthadj=1.,costh,zerolen;

   zerolen=(fnl[0]-in[0])*(fnl[0]-in[0]);
   if(zerolen>ZERO){
       if((fabs(in[2]-fnl[2])<SMALL)&&(fabs(in[1]-fnl[1])<SMALL)){
   lengthadj=sqrt(1.+(0.16*psiz[1]*psiz[2]/zerolen));
       }
       else if((fabs(in[2]-fnl[2])<SMALL)||(fabs(in[1]-fnl[1])<SMALL)){
   lengthadj=sqrt(1.+(0.08*psiz[1]*psiz[2]/zerolen));
       }}
   dist = sqrt( zerolen + (fnl[1]-in[1])*(fnl[1]-in[1])
               +(fnl[2]-in[2])*(fnl[2]-in[2]) );
   if (dist>ZERO){
       *watmud = dist* findmu(*(muvals),Enrg)*lengthadj;
      nI[0]=cI[0]=I[0]; nI[1]=cI[1]=I[1]; nI[2]=cI[2]=I[2];
      phi = atan((fnl[1]-in[1])/(fnl[0]-in[0]));
      if(phi<0.)phi+=pi;
      if((fnl[1]-in[1])<0.)phi+=pi;
      costh = (fnl[2]-in[2])/dist;
      sinth=sqrt(1.-costh*costh);
      sinphi=sin(phi);
      v[0]=sinth*sqrt(1.-sinphi*sinphi);
      v[1]=sinth*sinphi;
      v[2]=costh;

      if (v[0]<0.) sgn[0]=-1.;
      if (v[1]<0.) sgn[1]=-1.;
      if (v[2]<0.) sgn[2]=-1.;
      if (fabs(v[0])>ZERO) inv[0]=sgn[0]*psiz[0]/v[0];
         else inv[0]=200.*psiz[0];
      if (fabs(v[1])>ZERO) inv[1]=sgn[1]*psiz[1]/v[1];
         else inv[1]=200.*psiz[1];
      if (fabs(v[2])>ZERO) inv[2]=sgn[2]*psiz[2]/v[2];
         else inv[2]=200.*psiz[2];
      nxtp[0]=fabs((I[0]*2.+sgn[0]+1.)*0.5-in[0]*psiz1[0])*inv[0];
      nxtp[1]=fabs((I[1]*2.+sgn[1]+1.)*0.5-in[1]*psiz1[1])*inv[1];
      nxtp[2]=fabs((I[2]*2.+sgn[2]+1.)*0.5-in[2]*psiz1[2])*inv[2];
      while(dt+l<dist){
          l=nxtp[0]-dt; px=0;
          if((m=nxtp[1]-dt)<l) {l=m; px=1;}
          if((m=nxtp[2]-dt)<l) {l=m; px=2;}
          nI[px]+=sgn[px]; nxtp[px]+=inv[px];
          if((nI[px]>DIM-1)||(nI[px]<0)) {nI[px]-=sgn[px];}
          if(fabs(Attarr(nI[0],nI[1],nI[2])-Attarr(cI[0],cI[1],cI[2]))
             >SMALL) {
                 if(dt+l<dist){
                 mud += l*findmu(Attarr(cI[0],cI[1],cI[2]),Enrg);
                  dt += l; l=0.;
                  cI[0]=nI[0]; cI[1]=nI[1]; cI[2]=nI[2];}}
```

```
        }
       mud += (dist-dt)*findmu(Attarr(cI[0],cI[1],cI[2]),Enrg);
      mud=mud*lengthadj;
  }
    else {mud=psiz[0]*0.07*findmu(Attarr(I[0],I[1],I[2]),Enrg);
  *watmud = psiz[0]*0.07* findmu(*(muvals),Enrg);}
    return(mud);
}

double findmu2(float attval,double Enrg)
{ double retmu,xfrac;

  if(fabs(attval- *(muvals+1))<SMALL)
    retmu=(3.4138e-7*Enrg*Enrg-1.89195e-4*Enrg+0.0595829
   +1.5*exp(-Enrg/10.))/0.03979*attval;
  else {if(attval< *(muvals)+SMALL)
  retmu=(1.32e-6*Enrg*Enrg-7.32e-4*Enrg+0.2306
 +10.*exp(-Enrg/10.))/0.154*attval;
  else {if(attval> *(muvals+2)-SMALL)
  retmu=(5.0326e-6*Enrg*Enrg-2.28519e-3*Enrg+0.475784
 +68.*exp(-Enrg/10.))/0.25455*attval;
  else { xfrac=(attval- *(muvals))/mudiff;
 retmu=(1.32e-6*Enrg*Enrg-7.32e-4*Enrg+0.2306
+10.*exp(-Enrg/10.))/0.154*attval*(1.-xfrac);
 retmu+=(5.0326e-6*Enrg*Enrg-2.28519e-3*Enrg+0.475784
 +68.*exp(-Enrg/10.))/0.25455*attval*xfrac;}
    }}
return(retmu);
}

double findmu(float attval,double Enrg)
{ double retmu,xfrac;
  int Eval;

  Eval=irint((MAXENRG-Enrg)*EVALSZ);
  if(fabs(attval- *(muvals+1))<SMALL) retmu= *(enrgval+EFSIZE+Eval)*attval;
  else {if(attval< *(muvals)+SMALL) retmu= *(enrgval+Eval)*attval;
  else {if(attval> *(muvals+2)-SMALL) retmu= *(enrgval+2*EFSIZE+Eval)*attval;
  else { xfrac=(attval- *(muvals))/mudiff;
 retmu= *(enrgval+Eval)*attval*(1.-xfrac);
 retmu+= *(enrgval+2*EFSIZE+Eval)*attval*xfrac;}
    }}
  return(retmu);
}

/* using this meffdist instead of the original one yields errors of
   -0.1 to 0.1 % on a pixel by pixel basis and a total count diff
   of 0.01 % (global diff) */
/* path0 = dc0 - COLLDIST -1 ;  path1 = dc1 */
double meffdist (int I[3],double in[3],double fnl[3],double Ener)
{  double mud=0.,l=0.,lengthadj;
   int nI[3],cI[3],pI[3];
   double usemu,zerolen;
   short delx;
   short *curdir;
   float *curlen;
   int path0,path1,quarter=0,matpos;
   float AttarrnI, AttarrcI;

   lengthadj=psiz[0];
   zerolen=(fnl[0]-in[0])*(fnl[0]-in[0]);

   if(zerolen>ZERO){
   if((fabs(in[2]-fnl[2])<SMALL)&&(fabs(in[1]-fnl[1])<SMALL)){
       lengthadj*=sqrt(1.+(0.16*psiz[1]*psiz[2]/zerolen));
     }
   else if((fabs(in[2]-fnl[2])<SMALL)||(fabs(in[1]-fnl[1])<SMALL)){
       lengthadj*=sqrt(1.+(0.08*psiz[1]*psiz[2]/zerolen));
     }}

   pI[0]=irint((fnl[0]-in[0]-COLLDIST)*psiz1[0]);
   path0=pI[0]-1;
   if(path0<0) mud=0.;
   else {
   pI[1]=irint((fnl[1]-in[1])*psiz1[1]);
   pI[2]=irint((fnl[2]-in[2])*psiz1[2]);
   if(pI[2]>0) quarter++;
   if(pI[1]>0) quarter+=2;
   pI[1]=abs(pI[1]);
   pI[2]=abs(pI[2]);
   if(pI[2]<=pI[1]){
       path1=pI[1]*pI[1]+pI[2]*pI[2];
       if(path1==5) path1=3;
       else { if(path1==8) path1=5;}}
   else { if(pI[1]==0) path1=pI[2]+5;
  else path1=8;}

   nI[0]=cI[0]=I[0]; nI[1]=cI[1]=I[1]; nI[2]=cI[2]=I[2];
   AttarrcI=Attarr(cI[0],cI[1],cI[2]);
   AttarrnI=AttarrcI;
   matpos=(path1*DCOMAX+path0)*PATHMAX;
   curdir = pathdir+matpos;
   curlen = pathlen+matpos;
   delx=1;

   switch (quarter){
   case 0:
   while((delx!=0)&&(AttarrnI>=ZEROSQ)){
         l+= *curlen; curlen++;
 delx= *curdir; curdir++;
 switch (abs(delx)) {
        case 1: if(delx==1) nI[0]++; else nI[0]--; break;
case 2: if(delx==2) nI[1]++; else nI[1]--; break;
case 3: if(delx==3) nI[2]++; else nI[2]--; break;}
 AttarrnI=Attarr(nI[0],nI[1],nI[2]);
 if((fabs(AttarrnI-AttarrcI)>SMALL)||(delx==0)||(AttarrnI<ZEROSQ)){
     usemu=findmu(AttarrcI,Ener);
     mud += l*usemu;
     l=0.;
     cI[0]=nI[0]; cI[1]=nI[1]; cI[2]=nI[2];
     AttarrcI=AttarrnI;}}
   break;
   case 1:
   while((delx!=0)&&(AttarrnI>=ZEROSQ)){
         l+= *curlen; curlen++;
 delx=*curdir; curdir++;
 switch (abs(delx)) {
        case 1: if(delx==1) nI[0]++; else nI[0]--; break;
case 2: if(delx==2) nI[1]++; else nI[1]--; break;
case 3: if(delx==3) nI[2]--; else nI[2]++; break;}
 AttarrnI=Attarr(nI[0],nI[1],nI[2]);
 if((fabs(AttarrnI-AttarrcI)>SMALL)||(delx==0)||(AttarrnI<ZEROSQ)){
     usemu=findmu(AttarrcI,Ener);
     mud += l*usemu;
     l=0.;
     cI[0]=nI[0]; cI[1]=nI[1]; cI[2]=nI[2];
     AttarrcI=AttarrnI;}}
   break;
   case 2:
   while((delx!=0)&&(AttarrnI>=ZEROSQ)){
         l+= *curlen; curlen++;
 delx=*curdir; curdir++;
 switch (abs(delx)) {
        case 1: if(delx==1) nI[0]++; else nI[0]--; break;
case 2: if(delx==2) nI[1]--; else nI[1]++; break;
case 3: if(delx==3) nI[2]++; else nI[2]--; break;}
 AttarrnI=Attarr(nI[0],nI[1],nI[2]);
 if((fabs(AttarrnI-AttarrcI)>SMALL)||(delx==0)||(AttarrnI<ZEROSQ)){
     usemu=findmu(AttarrcI,Ener);
     mud += l*usemu;
     l=0.;
     cI[0]=nI[0]; cI[1]=nI[1]; cI[2]=nI[2];
     AttarrcI=AttarrnI;}}
   break;
   case 3:
   while((delx!=0)&&(AttarrnI>=ZEROSQ)){
         l+= *curlen; curlen++;
 delx=*curdir; curdir++;
 switch (abs(delx)) {
        case 1: if(delx==1) nI[0]++; else nI[0]--; break;
case 2: if(delx==2) nI[1]--; else nI[1]++; break;
case 3: if(delx==3) nI[2]--; else nI[2]++; break;}
 AttarrnI=Attarr(nI[0],nI[1],nI[2]);
 if((fabs(AttarrnI-AttarrcI)>SMALL)||(delx==0)||(AttarrnI<ZEROSQ)){
     usemu=findmu(AttarrcI,Ener);
     mud += l*usemu;
     l=0.;
     cI[0]=nI[0]; cI[1]=nI[1]; cI[2]=nI[2];
     AttarrcI=AttarrnI;}}
   break;}
   mud*=lengthadj;
   }
   return(mud);
}

double leffdist (int I[3],double in[3],double fnl[3], double Enrg)
/* same as neffdist, only no *watmud calculation */
{  double dist, sgn[3]={1.,1.,1.};
   double v[3],phi,mud=0.;
   double l=0.,m,dt=0.,nxtp[3];
   double inv[3];
   int px,nI[3],cI[3];
   double sinth,sinphi,lengthadj=1.,costh,zerolen;
   double fi0,fi1,fi2,ffi1,ffi2;

   fi0=fnl[0]-in[0];
   fi1=fnl[1]-in[1];
   fi2=fnl[2]-in[2];
   ffi1=fabs(fi1);
   ffi2=fabs(fi2);
   zerolen=fi0*fi0;
   if(zerolen>ZERO){
   if((ffi2<SMALL)&&(ffi1<SMALL)){
       lengthadj=sqrt(1.+(0.16*psiz[1]*psiz[2]/zerolen));
     }
   else if((ffi2<SMALL)||(ffi1<SMALL)){
       lengthadj=sqrt(1.+(0.08*psiz[1]*psiz[2]/zerolen));
     }}

   dist = sqrt( zerolen + fi1*fi1 + fi2*fi2 );
   if (dist>ZERO){
      nI[0]=cI[0]=I[0]; nI[1]=cI[1]=I[1]; nI[2]=cI[2]=I[2];
      phi = atan(fi1/fi0);
      if(phi<0.)phi+=pi;
      if(fi1<0.)phi+=pi;
      costh = fi2/dist;
      sinth=sqrt(1.-costh*costh);
      sinphi=sin(phi);
      v[0]=sinth*sqrt(1.-sinphi*sinphi);
      v[1]=sinth*sinphi;
      v[2]=costh;

      if (v[0]<0.) sgn[0]=-1.;
      if (v[1]<0.) sgn[1]=-1.;
      if (v[2]<0.) sgn[2]=-1.;
      if (fabs(v[0])>ZERO) inv[0]=sgn[0]*psiz[0]/v[0];
         else inv[0]=200.*psiz[0];
      if (fabs(v[1])>ZERO) inv[1]=sgn[1]*psiz[1]/v[1];
         else inv[1]=200.*psiz[1];
      if (fabs(v[2])>ZERO) inv[2]=sgn[2]*psiz[2]/v[2];
         else inv[2]=200.*psiz[2];
      nxtp[0]=fabs((I[0]*2.+sgn[0]+1.)*0.5-in[0]*psiz1[0])*inv[0];
      nxtp[1]=fabs((I[1]*2.+sgn[1]+1.)*0.5-in[1]*psiz1[1])*inv[1];
      nxtp[2]=fabs((I[2]*2.+sgn[2]+1.)*0.5-in[2]*psiz1[2])*inv[2];
      while(dt+l<dist){
          l=nxtp[0]-dt; px=0;
          if((m=nxtp[1]-dt)<l) {l=m; px=1;}
          if((m=nxtp[2]-dt)<l) {l=m; px=2;}
          nI[px]+=sgn[px]; nxtp[px]+=inv[px];
          if((nI[px]>DIM-1)||(nI[px]<0)) {nI[px]-=sgn[px];}
          if(fabs(Attarr(nI[0],nI[1],nI[2])-Attarr(cI[0],cI[1],cI[2]))
             >SMALL) {
                 if(dt+l<dist){
                 mud += l*findmu(Attarr(cI[0],cI[1],cI[2]),Enrg);
                  dt += l; l=0.;
                  cI[0]=nI[0]; cI[1]=nI[1]; cI[2]=nI[2];}}
        }
       mud += (dist-dt)*findmu(Attarr(cI[0],cI[1],cI[2]),Enrg);
      mud=mud*lengthadj;
    }
    else mud=psiz[0]*0.07*findmu(Attarr(I[0],I[1],I[2]),Enrg);
    return(mud);
}

double etalvl(int l1,int l2, int l3, double eta, int *e,
      double dcsppix, int *d)
{ double res;
  double dcp1,dcp2,dcp3;
  dcp1=-1; dcp2=-1; dcp3=-1;
  if(fabs(eta- *e*1.)<SMALL){ res=dcsplvl(l1,l2,l3,*e,dcsppix,d);
      dcp1=res;}
  else { res= interp(*e*1.,dcsplvl(l1,l2,l3,*e,dcsppix,d),
     *(e+1)*1.,dcsplvl(l1,l2,l3,*(e+1),dcsppix,d),
     *(e+2)*1.,dcsplvl(l1,l2,l3,*(e+2),dcsppix,d),eta);
 dcp1=dcsplvl(l1,l2,l3,*e,dcsppix,d);
 dcp2=dcsplvl(l1,l2,l3,e[1],dcsppix,d);
 dcp3=dcsplvl(l1,l2,l3,e[2],dcsppix,d);
     }
  return(res);
}

double dcsplvl(int l1,int l2, int l3, int l4, double dcsppix, int *d)
{ double res;
  if(fabs(dcsppix-dcspval[*d])<SMALL)
    res=Lookup(l1,l2,l3,l4,*d);
  else {
      res= interp(dcspval[*d],Lookup(l1,l2,l3,l4,*d),
  dcspval[*(d+1)],Lookup(l1,l2,l3,l4,*(d+1)),
  dcspval[*(d+2)],Lookup(l1,l2,l3,l4,*(d+2)),dcsppix);
     }
  return(res);
}

double etalvl2(int l1,int l2, int l3, double eta, int *e,
      double dcsppix, int *d)
{ double res;
  if(fabs(eta-*e*1.)<SMALL) res=dcsplvl2(l1,l2,l3,*e,dcsppix,d);
  else { res= interp(*e*1.,dcsplvl2(l1,l2,l3,*e,dcsppix,d),
     *(e+1)*1.,dcsplvl2(l1,l2,l3,*(e+1),dcsppix,d),
     *(e+2)*1.,dcsplvl2(l1,l2,l3,*(e+2),dcsppix,d),eta);}
  return(res);
}

double dcsplvl2(int l1,int l2, int l3, int l4, double dcsppix, int *d)
{ double res;
  if(fabs(dcsppix-dcspval[*d])<SMALL)
    res=Lookup2(l1,l2,l3,l4,*d);
  else { res= interp(dcspval[*d],Lookup2(l1,l2,l3,l4,*d),
     dcspval[*(d+1)],Lookup2(l1,l2,l3,l4,*(d+1)),
     dcspval[*(d+2)],Lookup2(l1,l2,l3,l4,*(d+2)),dcsppix);
     }
  return(res);
}


double findgfact(double spos[3],double cpos[3],double dpos[3],double *theta,
 double *resII, int dc0)
{  double cs[3],dc[3];
   double distcs, distdc, eta, ddcp, dcsp;
   int dc1,t[3],d[3],e[3];
   double res, tmp;

   cs[0]=cpos[0]-spos[0];
   cs[1]=cpos[1]-spos[1];
   cs[2]=cpos[2]-spos[2];

   dc[0]=dpos[0]-cpos[0];
   dc[1]=dpos[1]-cpos[1];
   dc[2]=dpos[2]-cpos[2];

   distcs = sqrt(cs[0]*cs[0]+cs[1]*cs[1]+cs[2]*cs[2]);
   distdc = sqrt(dc[0]*dc[0]+dc[1]*dc[1]+dc[2]*dc[2]);

/*   if (fabs(distcs*distdc)<ZERO) *theta =0.;*/
   if (distcs<SMALL) *theta =0.;
     else {
        tmp = (cs[0]*dc[0]+cs[1]*dc[1]+cs[2]*dc[2])/(distcs*distdc);
        if (fabs(tmp-1.)<ZERO) *theta = 0.;
           else{ if (fabs(tmp+1.)<ZERO) *theta = pi;
           else *theta = acos(tmp);}}

/* perpendicular dc distance index */
/*   dc0 = irint(dc[0]*psiz1[0]);
   if (dc0<1) printf("Warning: dc0 is out of range = %3d\n",dc0);*/
/* parallel dc distance index */
   ddcp = sqrt(dc[1]*dc[1]*psiz1[1]*psiz1[1] +
                    dc[2]*dc[2]*psiz1[2]*psiz1[2]);
   if (fabs(ddcp-sqrt(8.))<ZERO) dc1 = 5;
   else {if (fabs(ddcp-sqrt(5.))<ZERO) dc1 = 3;
         else dc1 = irint(ddcp*ddcp);}

   t[0]=dcs0_2ind(cs[0]*psiz1[0]);
   if (t[0]==0) t[1]=2; else t[1]=t[0]-1;
   if (t[0]==40) t[2]=38; else t[2]=t[0]+1;
   dcsp=sqrt( cs[1]*cs[1]*psiz1[1]*psiz1[1] +
      cs[2]*cs[2]*psiz1[2]*psiz1[2] );
   d[0]=dcsp_2ind(dcsp);
   if (d[0]==0) d[1]=2; else d[1]=d[0]-1;
   if (d[0]==45) d[2]=43; else d[2]=d[0]+1;

   if(dcsp>ZERO) {  /*eta is relevant dcsp>0 */
       if(dc1==0){
   tmp=(cs[2]*psiz1[2]/dcsp);
   if(fabs(tmp)>=1.) {eta=0.; e[0]=0;}
   else {eta=acos(tmp);
 while(eta>1.57079633) eta=eta-1.57079633;
 if(eta>0.785398163) eta=1.57079633-eta;
 eta=eta*3.81971864;
 e[0]=irint(eta);}
   if(e[0]==0) e[1]=2; else e[1]=e[0]-1;
   if(e[0]==3) e[2]=1; else e[2]=e[0]+1;  }
       else{
   tmp=(dc[1]*cs[1]*psiz1[1]*psiz1[1]+
dc[2]*cs[2]*psiz1[2]*psiz1[2])/(dcsp*ddcp);
   if(tmp>=1.) {eta=0.; e[0]=0;}
   else{ if(tmp<=-1.) {eta=9.; e[0]=9;}
         else {eta=acos(tmp);
       eta=eta*2.86478898;
       e[0]=irint(eta);}}
   if(e[0]==0) e[1]=2; else e[1]=e[0]-1;
   if(e[0]==9) e[2]=7; else e[2]=e[0]+1;  }

   if ((d[0]>45)||(t[0]>40)||(dc0>55)||(dc0<1)||(t[0]<0)) {res=0.; *resII=0.;}
   else {
       if(fabs(cs[0]*psiz1[0]-dcsOval[t[0]])<SMALL) {
 res=etalvl(t[0],dc0,dc1,eta,e,dcsp,d);
 *resII=etalvl2(t[0],dc0,dc1,eta,e,dcsp,d);}
       else{
 res= interp(dcsOval[t[0]],etalvl(t[0],dc0,dc1,eta,e,dcsp,d),
     dcsOval[t[1]],etalvl(t[1],dc0,dc1,eta,e,dcsp,d),
     dcsOval[t[2]],etalvl(t[2],dc0,dc1,eta,e,dcsp,d),
     cs[0]*psiz1[0]);
 *resII= interp(dcsOval[t[0]],etalvl2(t[0],dc0,dc1,eta,e,dcsp,d),
     dcsOval[t[1]],etalvl2(t[1],dc0,dc1,eta,e,dcsp,d),
     dcsOval[t[2]],etalvl2(t[2],dc0,dc1,eta,e,dcsp,d),
     cs[0]*psiz1[0]);
     }
   }
   }
  /* otherwise eta is not relavent and can just use eta=0 */
  else{ e[0] = 0;
     if ((d[0]>45)||(t[0]>40)||(dc0>55)||(dc0<1)||(t[0]<0))
       {res=0.; *resII=0.;}
     else {
       if(fabs(cs[0]*psiz1[0]-dcsOval[t[0]])<SMALL) {
   res=dcsplvl(t[0],dc0,dc1,e[0],dcsp,d);
   *resII=dcsplvl2(t[0],dc0,dc1,e[0],dcsp,d);}
```

```
       else{
   res= interp(dcsOval[t[0]],dcsplvl(t[0],dc0,dc1,e[0],dcsp,d),
       dcsOval[t[1]],dcsplvl(t[1],dc0,dc1,e[0],dcsp,d),
       dcsOval[t[2]],dcsplvl(t[2],dc0,dc1,e[0],dcsp,d),
       cs[0]*psiz1[0]);
   *resII= interp(dcsOval[t[0]],dcsplvl2(t[0],dc0,dc1,e[0],dcsp,d),
  dcsOval[t[1]],dcsplvl2(t[1],dc0,dc1,e[0],dcsp,d),
  dcsOval[t[2]],dcsplvl2(t[2],dc0,dc1,e[0],dcsp,d),
  cs[0]*psiz1[0]);
       }
     }
    }

   if(*theta<SMALL){
        if(distcs>SMALL)
              *theta=atan(0.45*psiz[1]/distcs)+atan(0.45*psiz[1]/distdc);
         else *theta=0.47;}
   return(res);
 }


double functn(double sdx2, double sdy,double sdz)
{  double peff,rsdm2,cosxi;
   int xival;

   rsdm2=1./(sdx2+sdy*sdy+sdz*sdz);
   cosxi=sqrt(sdx2*rsdm2);
/* if xi>2.44 deg, F(xi)=0.0  cos(2.5 deg)=0.99904823 */
   if(cosxi<0.99905) peff=0.;
      else {if (cosxi>1.) xival=0;
    else xival=irint((1.-cosxi)*10526315.7895);
            peff=Fofxi[xival]*rsdm2;}
   return(peff);
}

double trapz (double (*func)(double aa,double bb, double cc),
      double a, double b, double s,int it,double spos,
      double dpos, int dex)
{  double del,x,tsum=0.;
   int i;
   del = (b-a)/it;
   x= a;
   x+=0.5*del;
   for (i=1;i<=it;i++){
          if(dex==1) tsum+=(*func)(spos,x,dpos);
  else tsum+=(*func)(spos,dpos,x);
          x+=del;}
   s=0.5*(s+del*tsum);
   return(s);
}

void polint(double *xa,double *ya, int n, double x, double *y,double *dy)
{  double c[Kr+1]={0.},d[Kr+1]={0.};
   int ns,i,j;
   double dif,dift,ho,hp,den;

   ns=0;
   dif=fabs(x-xa[0]);
   c[0]=d[0]=ya[0];
   for(i=1;i<n;i++){
     dift=fabs(x-xa[i]);
     if(dift<dif) {ns=i; dif=dift;}
     c[i]=d[i]=ya[i];}
   *y=ya[ns];
   ns--;
   for(j=1;j<n;j++){
     for(i=0;i<n-j;i++){
       ho=xa[i]-x;
       hp=xa[i+j]-x;
       if(fabs(ho-hp)<ZERO*SILVL) printf("Problem in polint\n");
       else {
 den=(c[i+1]-d[i])/(ho-hp);
 d[i]=hp*den;
 c[i]=ho*den;}
     }
     if(2*ns<n-j) *dy=c[ns+1];
     else {*dy=d[ns]; ns=ns-1;}
     *y+=*dy;
 }
}

double qcom2integ(double sdx2, double sdy, double sdz)
{  double start, stop, ss, dss;
   int i,n,j;
   double sr[SIN+1], hr[SIN+1];

   hr[0]=1.;
   start=sdy-0.5*psiz[1];
   stop =sdy+0.5*psiz[1];
   sr[0] = 0.5*psiz[1]*(functn(sdx2,start,sdz)
+functn(sdx2,stop,sdz));
   i=1; n=1; dss=1.e5; ss=1.;
   while ((fabs(dss)>SILVL*fabs(ss)+ZEROSTQ)&&(i<=SIN)){
        hr[i] = hr[i-1]*0.25;
        sr[i] = trapz ((double(*)(double a,double b, double c))functn,
start,stop,sr[i-1],n,sdx2,sdz,1);
n *= 2;
j=i; if(j>Krm) j=Krm;
polint(hr+(i-j),sr+(i-j),j+1,0.,&ss,&dss);
i++;
      }
   return(ss);
}

double qcom3integ(double sdx2, double sdy, double sdz)
{  double start, stop, ss, dss;
   int i,n,j;
   double sr[SIN+1], hr[SIN+1];
   hr[0]=1.;
   start=sdz-0.5*psiz[2];
   stop =sdz+0.5*psiz[2];
   sr[0] = 0.5*psiz[2]*(qcom2integ(sdx2,sdy,start)
+qcom2integ(sdx2,sdy,stop));
   i=1; n=1; dss=1.e5; ss=1.;
   while ((fabs(dss)>SILVL*fabs(ss)+ZEROSTQ)&&(i<=SIN)){

        hr[i] = hr[i-1]*0.25;
        sr[i] = trapz ((double(*)(double a,double b, double c))qcom2integ,
start,stop,sr[i-1],n,sdx2,sdy,2);
n *= 2;
j=i; if(j>Krm) j=Krm;
polint(hr+(i-j),sr+(i-j),j+1,0.,&ss,&dss);
i++;
      }
   return(ss);
}


double pthwin(double th, double lowin, double hiwin)
{ double En,costh,sigma,za,zb,C,res;
  costh=cos(th);
  C=1.+alpha*(1.-costh);
  En=InitEner/C;
  sigma = En*0.060056121;
  za= (lowin-En)/sigma;
  zb= (hiwin-En)/sigma;
  res= fabs((erf(za)-erf(zb))/2.);
  return(res);
}


void main (int argc, char * argv[])
{
   FILE *outf;
   char *addlook1,*addlook2,*addpaths,*nmpathlen,*nmpathdir,*nmenrgyval;
   char *nmfofxi,*nmoutrt,*nmattmap,*nmdensmap;
   char name[90],line[120],*ptr;
/* Sxy=source pt, Dxy= detector pt,  Cxy= Compton scatter pt */
   int Sxy[3], Cxy[3]={0};
   int i,j,g1,g2;
   int ic1b,ic1e,ic2b,ic2e,dc0;
   int id2flip,lineno=1,id2b,id1b,id2e,id1e;
   int OFFDET, offswitch, ic0end;
   float UnscProj[DIM][DIM];
   float Project[DIM][DIM];
   float Project2[DIM][DIM];
   double srcpos[3],detpos[3],attpos[3];
   double mudss,mudsd;
   double sdx,sdx2,dperp;
   double gfact=0.,gfact2=0.,Ener,thval;
   double rotangle,xtmp,ytmp,ztmp,tmp;
double edenssav,distSA,PthWin0,LOWIN,HIWIN;


if(argc<2) printf("dualhxp <param_file> y z x angle\n");
else /* Have parameter file name */{

    for(i=0;i<DIM;i++){
for(j=0;j<DIM;j++){
    UnscProj[i][j]=0.;
    Project[i][j]=0.;
    Project2[i][j]=0.;}}

addlook1= (char *)calloc(StrSz,sizeof(char));
addlook2= (char *)calloc(StrSz,sizeof(char));
addpaths= (char *)calloc(StrSz,sizeof(char));
nmpathlen= (char *)calloc(StrSz2,sizeof(char));
nmpathdir= (char *)calloc(StrSz2,sizeof(char));
nmenrgyval= (char *)calloc(StrSz2,sizeof(char));
nmfofxi= (char *)calloc(StrSz,sizeof(char));
nmoutrt= (char *)calloc(StrSz,sizeof(char));
nmattmap= (char *)calloc(StrSz,sizeof(char));
nmdensmap= (char *)calloc(StrSz,sizeof(char));

sprintf(name,"%s",argv[1]);
printf("Reading from parameter file '%s'\n",name);
if((outf=fopen(name,"r"))){
    while(((ptr=fgets(line,120,outf))!=NULL)&&(lineno<LINEMAX)){
    if(line[0]!='#'){
switch(lineno){
case 1:  sscanf(line,"%s\n",addlook1); break;
case 2:  sscanf(line,"%s\n",addlook2); break;
case 3:  sscanf(line,"%s\n",addpaths); break;
case 4:  sscanf(line,"%s\n",nmpathlen); break;
case 5:  sscanf(line,"%s\n",nmpathdir); break;
case 6:  sscanf(line,"%s\n",nmenrgyval); break;
case 7:  sscanf(line,"%s\n",nmfofxi); break;
case 8:  sscanf(line,"%s\n",nmattmap); break;
case 9:  sscanf(line,"%s\n",nmdensmap); break;
case 10:  sscanf(line,"%s\n",nmoutrt); break;
case 11:  sscanf(line,"%lf%lf%lf\n",
 &InitEner,&LOWIN,&HIWIN); break;
}
lineno++;
    }
}
    fclose(outf);
}
else {
    printf("Cannot read '%s', using defaults!\n",name);
    addlook1="/home/wells/cprog/MCmatrix/dy3xpl2_";
    addlook2="/home/wells/cprog/MCmatrix/strIIxpl2_";
    addpaths="/home/wells/cprog/MCmatrix";
    nmpathlen="Pathlenxp.dat";
    nmpathdir="Pathdirxp.dat";
```

```
       nmenrgyval = "EnrgFracP.dat";
       nmfofxi = "/home/wells/cprog/MCmatrix/Fofxi2p5_ACOS.dat";
       nmattmap = "attmap";
       nmdensmap = "densmap";
       nmoutrt = "dualhxp";
       InitEner = 140.; LOWIN = 126.; HIWIN = 154.;
   }

   if (!((InitEner>0)&&(InitEner<1000.))) InitEner=140.;
       alpha=InitEner/511.;
   if (argc > 5) rotangle=atof(argv[5]); else rotangle=0.;
      while(rotangle<0.) rotangle+=360.;
      while(rotangle>=360.) rotangle-=360.;
   if (rotangle>ZERO) {sprintf(name,"%s%d.dat",nmattmap,irint(rotangle));
       sprintf(nmattmap,"%s",name);
       sprintf(name,"%s%d.dat",nmdensmap,irint(rotangle));
       sprintf(nmdensmap,"%s",name);}
     else {sprintf(name,"%s.dat",nmattmap);
   sprintf(nmattmap,"%s",name);
   sprintf(name,"%s.dat",nmdensmap);
   sprintf(nmdensmap,"%s",name);}

   printf("Lookup Table Paths:\n    %s\n    %s\n",addlook1,addlook2);

   printf("Data File Paths:\n");
   printf("    %s/%s\n    %s/%s\n    %s/%s\n",
  addpaths,nmpathlen,addpaths,nmpathdir,addpaths,nmenrgyval);
   printf("    %s\nLocation of patient maps:\n    %s\n    %s\n",
  nmfofxi,nmattmap,nmdensmap);
   printf("Writing output to: %s_sc0.dat, %s_sc1.dat, %s_sc2.dat\n",
  nmoutrt,nmoutrt,nmoutrt);
   printf("Using Initial Photon Energy of %g(keV)\n\n",InitEner);

   printf("Allocating space for arrays \n");
   lookup = (float *)calloc(LookSz,sizeof(float));
   lookup2 = (float *)calloc(LookSz,sizeof(float));
   attarr = (float *)calloc(DIM*DIM*DIM,sizeof(float));
   elecdens = (float *)calloc(DIM*DIM*DIM,sizeof(float));
   dcs0val=(float *)calloc(MaxCS0,sizeof(float));
   dcspval=(float *)calloc(MaxCSP,sizeof(float));

/* Initialize atten. array, src pos, det nml */

   printf("Acquiring arrays\n");
   g1=getattarr(&muvals,nmattmap);
   g2=getelecdens(&rhovals,nmdensmap);
   if((g1==1)&&(g2==1)){
       for(i=0;i<3;i++){ psiz1[i]=1./psiz[i]; }
       getpathlen(addpaths,nmpathlen,&pathlen);
       printf("Using DCOMAX= %d  PATHMAX= %d\nCollimator Thickness= %g (cm)\n",
      DCOMAX,PATHMAX,COLLDIST);
       pathdir = (short *)calloc(PATHSIZE,sizeof(short));
      getpathdir(addpaths,nmpathdir);
     getenrgval(addpaths,nmenrgyval,&enrgval);
```

```
    getFofxi(nmfofxi);
   tag=irint(100.*(COLLDIST*psiz1[0]-floor(COLLDIST*psiz1[0])));
   initdcs0val();
   initdcspval();

   if (argc > 2) ytmp=atof(argv[2]); else ytmp=0.;
   if (argc > 3) ztmp=atof(argv[3]); else ztmp=0.;
   if (argc > 4) xtmp=atof(argv[4]); else xtmp=0.;

   if(rotangle>ZERO){ tmp=ytmp;
             ytmp=ytmp*cos(rotangle*pi/180.)-xtmp*sin(rotangle*pi/180.);
             xtmp=xtmp*cos(rotangle*pi/180.)+ tmp*sin(rotangle*pi/180.);}

   Sxy[1]=centy+irint(ytmp*psiz1[1]);
   srcpos[1]=(0.5+centy)*psiz[1] + ytmp;
   Sxy[2]=centz+irint(ztmp*psiz1[2]);
   srcpos[2]=(0.5+centz)*psiz[2] + ztmp;
   Sxy[0]=centx+irint(xtmp*psiz1[0]);
   srcpos[0]=(0.5+centx)*psiz[0] + xtmp;

   wrtatslice(32,Sxy);

   printf("\n     sx=%7d     sy=%7d     sz=%7d\n",Sxy[0],Sxy[1],Sxy[2]);
   printf("  sposx=%7.4lf  sposy=%7.4lf  sposz=%7.4lf\n"
             ,srcpos[0],srcpos[1],srcpos[2]);

   ic0=Sxy[0]-1;
   if(ic0<0) ic0=0;

   detpos[0]=(0.5+centx)*psiz[0]+Cradrot+COLLDIST;
   printf("  dposx=%7.4lf\n\n",detpos[0]);

/*   Nelec= 3.343^26 el/kg 1000. kg/m^3 0.01 m/cm
          / 0.15 (atten of full water cube) * 1e-30 (from ro2 in Klein)*/
/*  Assume that srcstr is 1. -- can scale projection later */
/*   srcstr= 1.;         strength of source at Sxy  */

   sdx=detpos[0]-srcpos[0];
   sdx2=sdx*sdx;
   dperp=sdx*tanAA;
   id2b=floor((srcpos[2]-dperp)*psiz1[2]); if(id2b<0) id2b=0;
   id1b=floor((srcpos[1]-dperp)*psiz1[1]); if(id1b<0) id1b=0;
   id2e=ceil((srcpos[2]+dperp)*psiz1[2]); if(id2e>DIM) id2e=DIM;
   id1e=ceil((srcpos[1]+dperp)*psiz1[1]); if(id1e>DIM) id1e=DIM;

/* Begin detector/ matrix loop */

       PthWin0=pthwin(0.,LOWIN,HIWIN)/(4.*pi);

   for(id2=id2b;id2<id2e;id2++){ detpos[2]=(0.5+id2)*psiz[2];
 id2flip=DIM-1-id2;
   for(id1=id1b;id1<id1e;id1++){
       detpos[1]=(0.5+id1)*psiz[1];
```

```
/* attenuation src pt to detector pt*/
       mudsd = leffdist(Sxy,srcpos,detpos,InitEner);
       UnscProj[id2flip][id1]=
 qcom3integ(sdx2,(detpos[1]-srcpos[1]),(detpos[2]-srcpos[2]))
/*    *exp(-mudsd)*pthwin(0.)/(4.*pi);*/
   *exp(-mudsd)*PthWin0;
/*    *exp(-mudsd)*0.078102742;*/
   }}

   sprintf(name,"%s_sc0.dat",nmoutrt);
   printf("Writing projection in wit-raw format to '%s'\n",name);
   outf = fopen(name,"w");
   fwrite(UnscProj,DIM*DIM,sizeof(float),outf);
   fclose(outf);

/* Begin detector/ matrix loop */
   printf("\nic0= ");

   offswitch=irint(detpos[0]*psiz1[0]-23.5);
   OFFDET=2; ic0end=offswitch; if(ic0end>DIM) ic0end=DIM;

while(ic0<DIM){
    if(ic0>=(offswitch+21)) {OFFDET=0; ic0end=DIM;}
    else {if(ic0>=offswitch) {OFFDET=1; ic0end=offswitch+21;
      if(ic0end>DIM) ic0end=DIM;}}

   for(;ic0<ic0end;ic0++){
     printf(" %d",ic0);
     fflush(stdout);
     Cxy[0]=ic0; attpos[0]=(0.5+Cxy[0])*psiz[0];
      dc0 = floor((detpos[0]-attpos[0])*psiz1[0]);
/*      if (dc0<1) printf("Warning: dc0 is out of range = %3d\n",dc0);*/
/*  dc0 = 12  -- distance det to cpos in x-dir */
      getnxtsum(dc0,addlook1);
      getnxtsumII(dc0,addlook2);
   for(ic1=0;ic1<DIM;ic1++){ Cxy[1]=ic1; attpos[1]=(0.5+Cxy[1])*psiz[1];
      if(ic1<OFFDET) ic1b=0; else ic1b=ic1-OFFDET;
      if(ic1>=(DIM-OFFDET-1)) ic1e=DIM; else ic1e=ic1+OFFDET+1;
   for(ic2=0;ic2<DIM;ic2++){ Cxy[2]=ic2; attpos[2]=(0.5+Cxy[2])*psiz[2];

      edenssav=Elecdens(ic0,ic1,ic2);
   if (edenssav != 0.) {
      if(ic2<OFFDET) ic2b=0; else ic2b=ic2-OFFDET;
      if(ic2>=(DIM-OFFDET-1)) ic2e=DIM; else ic2e=ic2+OFFDET+1;

/* attenuation src to scat pt */
        mudss = neffdist(Sxy,srcpos,attpos,&distSA,InitEner);

   for(id2=ic2b;id2<ic2e;id2++){ detpos[2]=(0.5+id2)*psiz[2];
     id2flip=DIM-1-id2;
   for(id1=ic1b;id1<ic1e;id1++){ detpos[1]=(0.5+id1)*psiz[1];
   {

/* find the geometrical factor of the scatter */
```

```
        gfact = findgfact(srcpos,attpos,detpos,&thval,&gfact2,dc0);

/*   if (gfact>ZERO) { */
   if ((gfact>0.)||(gfact2>0.)) {

/* determine energy of scattered photon */
        Ener = InitEner/(1.+alpha*(1.-cos(thval)));
/* attenuation scat pt to detector pt*/
        mudsd = meffdist(Cxy,attpos,detpos,Ener);
/*        mudsd = neffdist(Cxy,attpos,detpos);*/

    Project[id2flip][id1]+=gfact*edenssav*exp(-(mudsd+mudss));
    if(distSA<SMALL*psiz[0])
      Project2[id2flip][id1]+=gfact2*edenssav*exp(-(mudsd+mudss));
    else
      Project2[id2flip][id1]+=gfact2*edenssav*exp(-(mudsd+mudss))*mudss/distSA;

/* Adjust by distsa*muH2O/mudss */
      }  /* end gfact >0. */

    }  }}  }   }
   }
 }
}
  printf("\n");

   sprintf(name,"%s_sc1.dat",nmoutrt);
    printf("Writing projection in wit-raw format to '%s'\n",name);
    outf = fopen(name,"w");
    fwrite(Project,DIM*DIM,sizeof(float),outf);
    fclose(outf);

   sprintf(name,"%s_sc2.dat",nmoutrt);
    printf("Writing projection in wit-raw format to '%s'\n",name);
    outf = fopen(name,"w");
    fwrite(Project2,DIM*DIM,sizeof(float),outf);
    fclose(outf);

   }
}
}
```

A.2 MKATPATH.C

The code *mkatpath.c* is used to generate a lookup table which allows a more rapid determination of some of the attenuation factors for the photon paths. This code uses a parameter file called *mkatpath.param* for variable specification.

```
/* program mkatpath.c */
/*  Generate attenuation lengths through pixelized paths */
/*  MC configuration , ie psiz=0.5cm COLLDIST=3.28cm */

#include <stdio.h>
#include <math.h>

#define pi 3.1415927
#define ZERO 1.e-8
#define Pathlen(l1,l2,l3) pathlen[(l1)*DCOMAX*PATHLENMAX+(l2)*PATHLENMAX+(l3)]
#define Pathdir(l1,l2,l3) pathdir[(l1)*DCOMAX*PATHLENMAX+(l2)*PATHLENMAX+(l3)]
#define StrSz1 180
#define LINEMAX 8

int DCOMAX, PATHLENMAX;
short *pathdir;
float *pathlen;
double COLLDIST;

void buildpath(double in[3],int dc0,int dc1)
{  double dist, sgn[3]={1.,1.,1.}, fulpath;
   double v[3],phi;
   double l=0.,m,dt=0.,nxtp[3];
   double inv[3];
   int px,count=0;
   double sinth,sinphi,costh;
   int i;

   phi = atan(in[1]/in[0]);
   if(phi<0.)phi+=pi;
   if(-in[1]<0.)phi+=pi;
   sinphi=sin(phi);

   fulpath = sqrt(in[0]*in[0] + in[1]*in[1] + in[2]*in[2]);
   costh = (-in[2])/fulpath;
   sinth=sqrt(1.-costh*costh);

   v[0]=sinth*sqrt(1.-sinphi*sinphi);
   v[1]=sinth*sinphi;
   v[2]=costh;

   dist=fabs((fabs(in[0])-COLLDIST)/v[0]);

   if (v[0]<0.) sgn[0]=-1.;
   if (v[1]<0.) sgn[1]=-1.;
   if (v[2]<0.) sgn[2]=-1.;
   if (fabs(v[0])>ZERO) inv[0]=sgn[0]/v[0]; else inv[0]=200.;
   if (fabs(v[1])>ZERO) inv[1]=sgn[1]/v[1]; else inv[1]=200.;
   if (fabs(v[2])>ZERO) inv[2]=sgn[2]/v[2]; else inv[2]=200.;

   nxtp[0]=fabs(sgn[0]*0.5)*inv[0];
   nxtp[1]=fabs(sgn[1]*0.5)*inv[1];
   nxtp[2]=fabs(sgn[2]*0.5)*inv[2];

/*   for(i=0;i<3;i++){printf("v[%d]=%g  sgn[%d]=%g  inv[%d]=%g  nxtp[%d]=%g\n",
   i,v[i],i,sgn[i],i,inv[i],i,nxtp[i]);}
*/
      while(dt+l<dist){
          l=nxtp[0]-dt; px=0;
          if((m=nxtp[1]-dt)<l) {l=m; px=1;}
          if((m=nxtp[2]-dt)<l) {l=m; px=2;}
  nxtp[px]+=inv[px];
  if(dt+l<dist){
     Pathdir(dc1,dc0,count)=irint(sgn[px]*(px+1));
     Pathlen(dc1,dc0,count)=l;
/*    printf("count=%d  dir=%d  len=%g\n",count,Pathdir(dc1,dc0,count),l);*/
     dt += l; l=0.;
  }
  count++;
      }
      count--;
      Pathlen(dc1,dc0,count)=(dist-dt);
      Pathdir(dc1,dc0,count)=0;
/*      printf("count=%d  dir=%d  len=%g\n",count,0,dist-dt);*/
}


void main (int argc, char * argv[])
{  FILE *outf;
   char *outdirnm,*outlennm;
   char *ptr,line[120];
   int i,j,lineno=1,count,addheader;
   int dc1,dc0;
   short direct;
   float total,dist,collcm,Psiz,*header;
   double attpos[3];

/* Read in parameter file */
```

```
   if(!(outf=fopen("mkatpath.param","r"))){
   printf("Cannot read 'mkatpath.param'\n");}
       else{
   outdirnm=(char *)calloc(StrSz1,sizeof(char));
   outlennm=(char *)calloc(StrSz1,sizeof(char));
   while(((ptr=fgets(line,120,outf))!=NULL)&&(lineno<LINEMAX)){
       if(line[0]!='#'){
   switch(lineno){
   case 1:  sscanf(line,"%d\n",&DCOMAX); break;
   case 2:  sscanf(line,"%d\n",&PATHLENMAX); break;
   case 3:  sscanf(line,"%f\n",&Psiz); break;
   case 4:  sscanf(line,"%f\n",&collcm); break;
   case 5:  sscanf(line,"%s\n",outdirnm); break;
   case 6:  sscanf(line,"%s\n",outlennm); break;
   case 7:  sscanf(line,"%d\n",&addheader); break;
   }
   lineno++;
       }
   }
   fclose(outf);

   COLLDIST = collcm/Psiz;
   printf("Max X-dir= %d (pix)     Max Path length= %d (pix)\n",
  DCOMAX,PATHLENMAX);
   printf("Effective Collimator thickness= %g (cm), or %g (pix)\n",
  collcm,COLLDIST);
   if(addheader==0)
     printf("Not including header information in files.\n");
   else printf("Including header information in files.\n\n");

   header = (float *)calloc(3,sizeof(float));
   pathlen = (float *)calloc(9*DCOMAX*PATHLENMAX,sizeof(float));
   pathdir = (short *)calloc(9*DCOMAX*PATHLENMAX,sizeof(short));

   if((pathlen==NULL)||(pathdir==NULL))
     printf("Cannot allocate enough memory!\n");
   else{

/* Begin generation of tables */

       for(dc1=0;dc1<9;dc1++){
   switch (dc1) {
   case 0:  attpos[1]=0.; attpos[2]=0.; break;
   case 1:  attpos[1]=1.; attpos[2]=0.; break;
   case 2:  attpos[1]=1.; attpos[2]=1.; break;
   case 4:  attpos[1]=2.; attpos[2]=0.; break;
   case 3:  attpos[1]=2.; attpos[2]=1.; break;
   case 5:  attpos[1]=2.; attpos[2]=2.; break;
   case 6:  attpos[1]=0.; attpos[2]=1.; break;
   case 7:  attpos[1]=0.; attpos[2]=2.; break;
   case 8:  attpos[1]=1.; attpos[2]=2.; break;}
   for(dc0=0;dc0<DCOMAX;dc0++){
       attpos[0]=-1.*(dc0+1+COLLDIST);
       buildpath(attpos,dc0,dc1);
   }}

/* Assign header */
       *(header)=DCOMAX*1.;
       *(header+1)=PATHLENMAX*1.;
       *(header+2)=collcm;

/* Write out the files */

       outf=fopen(outdirnm,"w");
       printf("Writing to '%s'\n",outdirnm);
       if(addheader==1) fwrite(header,3,sizeof(float),outf);
       fwrite(pathdir,9*DCOMAX*PATHLENMAX,sizeof(short),outf);
       fclose(outf);

       outf=fopen(outlennm,"w");
       printf("Writing to '%s'\n",outlennm);
       if(addheader==1) fwrite(header,3,sizeof(float),outf);
       fwrite(pathlen,9*DCOMAX*PATHLENMAX,sizeof(float),outf);
       fclose(outf);

/* Print out a simple path */

       dc0=10; dc1=0;
       printf("\nChecking dc0=%d  dc1=%d\n",dc0,dc1);
       count=0;
       total=0.;
       direct=Pathdir(dc1,dc0,count);
       dist=Pathlen(dc1,dc0,count);
       while((count<PATHLENMAX)&&(direct!=0)){
   total+=dist;
   printf("Pdir=%d  Plen=%g  Cumlen=%g\n",direct,dist,total);
   count++;
   direct=Pathdir(dc1,dc0,count);
   dist=Pathlen(dc1,dc0,count);}
       total+=dist;
       printf("Pdir=%d  Plen=%g  Cumlen=%g\n",direct,dist,total);
       printf("Total count=%d\n",count);
   } /* End enough memory */
       } /* End found parameter file */
}
```

A.3 MKENRG.C

The code *mkenrg.c* is used to generate a lookup table which is used to determine the variation of the attenuation coefficient with energy. The data for each attenuating material is stored as a different array. This program accesses data files which contain the atomic attenuation coefficients as a function of x, the momentum transfer, and also contain a description of the different materials being modelled.

```
/* program mkenrg.c */
/*    Create a table of lookup values for the diff materials */
/*    Reading from Simset values */

#include <stdio.h>
#include <math.h>

#define pi 3.1415927
#define ZERO 1.e-8
#define Enrgval(l1,l2) enrgval[(l1)*10001+(l2)]

float *enrgval;

int getatttbl(float **table)
{ FILE *outf;
  char name[50],*ptr,line[120];
  float *xtmp;
  int i,j,errflag=0;
  float in1,in2,in3,in4,in5,in6;

  xtmp=(float *)calloc(951*4,sizeof(float));
  for(j=0;j<3;j++){
      switch(j) {
        case 0: sprintf(name,"water.att"); break;
        case 1: sprintf(name,"lung.att"); break;
/*        case 1: sprintf(name,"poumon.att"); break;*/
        case 2: sprintf(name,"bone.att"); break; }
      if(!(outf=fopen(name,"r"))){ j=4;
   printf("Cannot access '%s'\n",name);
          errflag=1;}
      else {
  i=0;
  while((ptr=fgets(line,120,outf))!=NULL){
      sscanf(line,"%f%f%f%f%f %f\n",&in1,&in2,&in3,&in4,&in5,&in6);
      if(j==0){*(xtmp+i*4)=in1;}
      *(xtmp+i*4+j+1)=in6;
      i++;
  }
      }
  }
      if(errflag==1) free(xtmp);
      else *table=xtmp;
  return(errflag);
}

float intattval(int *initx,float xEnrg,int mat,float *table)
{ int i,j,k;
  float curE,res,Ei,Eim1;

  if(*initx<=2) i=1;
  else i=*initx-1;
  curE=*(table+i*4);
  while((curE<xEnrg)&&(i<950)){
      i++;
      curE=*(table+i*4);}
  *initx=i;
  j=i*4+mat+1;
  k=(i-1)*4+mat+1;
  Ei=*(table+i*4);
  Eim1=*(table+i*4-4);
  res=(*(table+j)- *(table+k))/(Ei-Eim1)*(xEnrg-Eim1)+ *(table+k);
  return(res);
}


void main (int argc, char * argv[])
{  FILE *outf;
   char *filnm;
   int i,j,initi;
   float frac,*atttbl;
   double Enrg;

   i=getatttbl(&atttbl);
   if(i!=1){

/*   filnm="mkenrg.tmp";
       outf=fopen(filnm,"w");
       for(i=0;i<951;i++){
   fprintf(outf,"%g %g %g %g\n",*(atttbl+i*4),*(atttbl+i*4+1),
   *(atttbl+i*4+2),*(atttbl+i*4+3));
       }
```

```
*/

   enrgval = (float *)calloc(3*10001,sizeof(float));

/* i: H2O (0), Lung (1), Bone (2) */

   for(j=0,initi=0;j<10001;j++){
       for(i=0;i<3;i++){
   Enrg=140.-40.*j/10000.;
   Enrgval(i,j)=intattval(&initi,Enrg,i,atttbl);}}
/*
   switch(i){
   case 0: frac=(1.32e-6*Enrg*Enrg-7.32e-4*Enrg+0.2306
 +10.*exp(-Enrg/10.))/0.154; break;
   case 1: frac=(3.4138e-7*Enrg*Enrg-1.89195e-4*Enrg+0.0595829
 +1.5*exp(-Enrg/10.))/0.03979; break;
   case 2: frac=(5.0326e-6*Enrg*Enrg-2.28519e-3*Enrg+0.475784
 +68.*exp(-Enrg/10.))/0.25455; break;}
   Enrgval(i,j)=frac;}}
*/

   for(j=0;j<10001;j+=1000){
       Enrg=140.-40.*j/10000.;
       printf("Enrg=%g  H2O=%g  Lung=%g  Bone=%g\n",
      Enrg,Enrgval(0,j),Enrgval(1,j),Enrgval(2,j));}

/*   filnm="EnrgFracP.dat";*/
   filnm="EnrgFracSim.dat";
   outf=fopen(filnm,"w");
   printf("Writing energy data to '%s'\n",filnm);
   fwrite(enrgval,3*10001,sizeof(float),outf);
   fclose(outf);

/*   filnm="mkenrg.tmp2";
   outf=fopen(filnm,"w");
   printf("Writing energy data to '%s'\n",filnm);
   for(j=0;j<10001;j+=100){
       Enrg=140.-40.*j/10000.;
       fprintf(outf,"%g %g %g %g\n",Enrg,*(enrgval+j),
       *(enrgval+10001+j),*(enrgval+2*10001+j));
   }*/

}

}
```

A.4 MAPGEN3D.C

The *mapgen3d.c* code generates 3D attenuation and electron density maps as specified by the input file. The input filename is specified at the time of execution on the command line. The maps are generated in the form appropriate for reading by the code *dualhxp.c.*

```
/* Building the attenuation map */
/* program mapgen3d.c */
/*    uses input file given by argument 1 */

#include <stdio.h>
#include <math.h>
#include <string.h>

#define pi 3.1415927
#define ZERO 1.e-11
#define MAXCASE 10
#define ObjEntries 6

/* h2o, lung, bone */
/*float muvals[3]={0.15108,0.03903,0.24767};*/
/*float rhovals[3]={0.003343,0.0008635,0.0052668};*/
/* Now put in SIMSET values for 140 keV */
/*float muvals[3]={0.15217,0.0392,0.24826};*/
/*float rhovals[3]={0.003323,0.0008555,0.0052357};*/

int *BINS, *NMAT,*NOBJ,*OUTTYPE;
float *matdata,*objdata,*PSIZE,*rotangle,*zlims,*Cradrot;
char attfile[50],densfile[50];
float centx, centy;

void chstatt(float *datline, int DIM, float filval, float *attarr);
void att2dens(float *attarr,int DIM, float *matdat, int Nmat);
void dorotation(float angle, int DIM, float *attarr);
void binattarr(int NBIN, int BIN, float *store, float *attarr);
void initmat(float *,int);
void wrtattarr(char *name,int NBIN,float *head,int Nmat, float *klims,
       float *matr);
void prterror(int isOK,int *nline,char *line,int errl);
int getint(int numdat,char *look,int **res);
int getflt(int numdat,char *look,float **res);
int indinlst(int ind,int Nmat,float *matdata);
void get1str(char *line,char *attfile);
int getdatafile(char *filnm);
void fixind(float *mat,int Nmat,float *obj,int Nobj);
void chstatt(float *datline, int DIM, float filval, float *attarr)
{ int i,j,type;
  float si,sj;
  float cxp,cyp,cxrad,cyrad;
  float SBIN,Px,Py;

  SBIN= *(BINS+1)*1.;
  Px= *(PSIZE) *1.;
  Py= *(PSIZE+1) *1.;
  cxp=(*(datline+1)/Px)*SBIN + centx;
  cyp=(*(datline+2)/Py)*SBIN + centy;
  cxrad=(*(datline+3)/Px)*SBIN;
  cyrad=(*(datline+4)/Py)* SBIN;

  type=irint(*(datline+5));
  /* If don't recognize type, set to circle */
  if((type<0)||(type>1)) type=0;

  switch(type){
  /* Circle */
  case 0:
      cxrad=cxrad*cxrad;
      cyrad=cyrad*cyrad;
      for (i=0;i<DIM;i++){ si=i*1.;
for(j=0;j<DIM;j++){ sj=j*1.;
           if ((si-cxp)*(si-cxp)/cxrad+(sj-cyp)*(sj-cyp)/cyrad < 1. )
     {attarr[i*DIM+j]= filval;}
    }}
      break;
  case 1:
  /* Square */
      cxrad*=0.5;
      cyrad*=0.5;
      for (i=0;i<DIM;i++){ si=i*1.;
for(j=0;j<DIM;j++){ sj=j*1.;
   if ((fabs(si-cxp)<=cxrad)&&(fabs(sj-cyp)<=cyrad))
     {attarr[i*DIM+j]= filval;}
    }}
      break;
  }
}

void att2dens(float *attarr,int DIM, float *matdat, int Nmat)
{ int i,j;
  for(i=0;i<DIM*DIM;i++){
      for(j=0;j<Nmat;j++){
  if(fabs(*(attarr+i)- *(matdat+j*3+1))<ZERO)
    *(attarr+i)= *(matdat+j*3+2);}}
```

```
}

void dorotation(float angle, int DIM, float *attarr)
{ int i,j,in,jn;
  float xc,yc,th,rad,thp,xn,yn, tvalue;
  float si,sj;
  float *rotarr;

  rotarr=(float *)calloc(DIM*DIM,sizeof(float));

  printf(" Performing a rotation of %g degrees\n",angle);
  for (i=0;i<DIM;i++){
      si=i*1.;
      for(j=0;j<DIM;j++){
  sj=j*1.;
  xc=si-centx;
  yc=sj-centy;
  rad=sqrt(xc*xc+yc*yc);
  if (fabs(xc)<ZERO) {if(yc>0) th=pi/2.;
      else th=-pi/2.;}
  else {th=atan(yc/xc);
if (xc<0.) th+=pi;}
  thp=th+angle*pi/180.;
/*   while(thp>2.*pi) thp-=(2.*pi);
  while(thp<0.) thp+=(2.*pi);*/
  xn=rad*cos(thp);
  yn=rad*sin(thp);
  in=irint(centx+xn);
  jn=irint(centy+yn);
  if((in<0)||(in>DIM-1)||(jn<0)||(jn>DIM-1)) tvalue = 0.;
  else tvalue = attarr[in*DIM+jn];
  rotarr[i*DIM+j]=tvalue;
      }}
  for (i=0;i<DIM*DIM;i++){*(attarr+i)=*(rotarr+i); }
  free(rotarr);
}

void binattarr(int NBIN, int BIN, float *store, float *attarr)
{ int i,j,k;
  int l,m,n;
  float sum;

  printf(" Reducing array to %d\n",NBIN);

  for(l=0;l<NBIN;l++){
  for(m=0;m<NBIN;m++){
    sum = 0.;

  for(i=0;i<BIN;i++){
    for(j=0;j<BIN;j++){
            sum += attarr[(l*BIN+i)*NBIN*BIN+m*BIN+j];}}

    store[l*NBIN+m]=sum/(BIN*BIN*1.);
  }}
}

void initmat(float *matr,int DIM)
{ int i;
  for(i=0;i<DIM*DIM;i++) *(matr+i)=0.;}

void wrtattarr(char *name,int NBIN,float *head,int Nmat, float *klims,
       float *matr)
{ FILE *outf;
  float *themap;
  int i,j,k;
  int kbot,ktop;
  float kbflt,ktflt;
  float scale;

  kbflt=(NBIN/2.)+0.5+ *(klims)/ *(PSIZE+2);
  ktflt=(NBIN/2.)+0.5+ *(klims+1)/ *(PSIZE+2);
  kbot=floor(kbflt);
  ktop=floor(ktflt);

  printf(" Writing out to file '%s' in the new (0) format\n",name);
  outf = fopen(name,"w");
  themap=(float *)calloc(NBIN*NBIN*NBIN,sizeof(float));

  for(i=0;i<NBIN;i++){
  for(j=0;j<NBIN;j++){
  for(k=0;k<NBIN;k++){
      if((k<kbot)||(k>ktop)) themap[k*NBIN*NBIN+i*NBIN+j]=0.;
       else {
 scale=1.;
 if(k==kbot){ scale=fabs((k+1)*1.-kbflt); }
 if(k==ktop){ scale=fabs((k+1)*1.-ktflt); }
 themap[k*NBIN*NBIN+i*NBIN+j]=scale*matr[i*NBIN+j];}
     }}}
  kbflt=Nmat*1.;
  fwrite(&kbflt,1,sizeof(float),outf);
  fwrite(head,Nmat,sizeof(float),outf);
  fwrite(PSIZE,3,sizeof(float),outf);
  fwrite(Cradrot,1,sizeof(float),outf);
  kbflt=NBIN*1.;
  fwrite(&kbflt,1,sizeof(float),outf);
  fwrite(themap,NBIN*NBIN*NBIN,sizeof(float),outf);
  fclose(outf);
}
```

```
void wrtattold(char *name,int NBIN,float *head,int Nmat, float *klims,
       float *matr)
{ FILE *outf;
  float *themap;
  int i,j,k;
  int kbot,ktop;
  float kbflt,ktflt;
  float scale;

  kbflt=(NBIN/2.)+0.5+ *(klims)/ *(PSIZE+2);
  ktflt=(NBIN/2.)+0.5+ *(klims+1)/ *(PSIZE+2);
  kbot=floor(kbflt);
  ktop=floor(ktflt);

  printf(" Writing out to file '%s' in the old (1) format\n",name);
  outf = fopen(name,"w");
  themap=(float *)calloc(NBIN*NBIN*NBIN,sizeof(float));

  for(i=0;i<NBIN;i++){
  for(j=0;j<NBIN;j++){
  for(k=0;k<NBIN;k++){
      if((k<kbot)||(k>ktop)) themap[k*NBIN*NBIN+i*NBIN+j]=0.;
       else {
 scale=1.;
 if(k==kbot){ scale=fabs((k+1)*1.-kbflt); }
 if(k==ktop){ scale=fabs((k+1)*1.-ktflt); }
 themap[k*NBIN*NBIN+i*NBIN+j]=scale*matr[i*NBIN+j];}
     }}}
  fwrite(themap,NBIN*NBIN*NBIN,sizeof(float),outf);
  kbflt=Nmat*1.;
  fwrite(&kbflt,1,sizeof(float),outf);
  fwrite(head,Nmat,sizeof(float),outf);
/*
   fwrite(PSIZE,3,sizeof(float),outf);
   fwrite(Cradrot,1,sizeof(float),outf);
   kbflt=NBIN*1.;
   fwrite(&kbflt,1,sizeof(float),outf);
*/
  fclose(outf);
}

void prterror(int isOK,int *nline,char *line,int errl)
{ if (isOK==0) printf("\n**** Error on line %d ****\n%s\n",errl,line);
}

void wrtslice(char *name,int NBIN,float *matr)
{ FILE *outf;
  outf=fopen(name,"w");
  printf(" Writing only the slice to '%s'\n",name);
  fwrite(matr,NBIN*NBIN,sizeof(float),outf);
  fclose(outf);
}

int getint(int numdat,char *look,int **res)
{ int isOK=1,j;
  char *lpos;
  char number[]="-0123456789.";
  char spacer[]=" \n ,";
  int *rtmp,inp1;

  rtmp=(int *)calloc(numdat,sizeof(int));
  for(j=0;j<numdat;j++){
      lpos=strpbrk(look,number);
      if(lpos==NULL){ isOK=0; j=numdat;}
      else{
  sprintf(look,"%s",lpos);
  sscanf(look,"%d\n",&inp1);
  lpos=strpbrk(look,spacer);
  sprintf(look,"%s",lpos);
  *(rtmp+j)=inp1;}}
  *res=rtmp;
  return(isOK);
}

int getflt(int numdat,char *look,float **res)
{ int isOK=1,j;
  char *lpos;
  char number[]="-0123456789.";
  char spacer[]=" \n ,";
  float *rtmp,inp1;

  rtmp=(float *)calloc(numdat,sizeof(float));
  for(j=0;j<numdat;j++){
      lpos=strpbrk(look,number);
      if(lpos==NULL){ isOK=0; j=numdat; }
      else{
  sprintf(look,"%s",lpos);
  sscanf(look,"%f\n",&inp1);
  lpos=strpbrk(look,spacer);
  sprintf(look,"%s",lpos);
  *(rtmp+j)=inp1;}}
  *res=rtmp;
  return(isOK);
}

int indinlst(int ind,int Nmat,float *matdata)
{ int i,flag=0;
```

```
  for(i=0;i<Nmat;i++){
      if(ind==irint(*(matdata+i*3))) flag=1;}
  return(flag);
}

void get1str(char *line,char *attfile)
{  char *lpos;
   char spacer[]=" \n ,";
   lpos=strpbrk(line,spacer);
   while(lpos==line){
       line++;
       lpos=strpbrk(line,spacer);}
   *(lpos)='\0';
   sprintf(attfile,"%s",line);
}

int getdatafile(char *filnm)
{  FILE *outf;
   char *lpos,line[120],*ptr,savline[120];
   char number[]="-0123456789.";
   char spacer[]=" \n ,";
   char outtypestr[20];
   int nline=0,i,j,errline=0;
   int *tmpint,FLAG=1;
   float *tmpfloat,*omatdata;

   if(!(outf=fopen(filnm,"r"))){ printf("Problem reading '%s'\n",filnm);
         FLAG=0;}
   else {
      printf("Reading %s\n",filnm);
      while(((ptr=fgets(line,120,outf))!=NULL)&&(nline<MAXCASE)
    &&(FLAG==1)){ /*while*/
  errline++;
  if(line[0]!='#'){
      switch(nline){
      case 0:  sprintf(savline,"%s",line);
  FLAG=getint(2,line,&BINS);
  printf("DIM(subdivisions):  %d (%d)\n",*(BINS),*(BINS+1));
  prterror(FLAG,&nline,savline,errline);
  nline++;
  break;
      case 1:  sprintf(savline,"%s",line);
  FLAG=getflt(3,line,&PSIZE);
  printf("Pixel Size: (%6.4f,%6.4f,%6.4f)\n",
 *(PSIZE),*(PSIZE+1),*(PSIZE+2));
  prterror(FLAG,&nline,savline,errline);
  nline++;
  break;
      case 2:  sprintf(savline,"%s",line);
  FLAG=getflt(1,line,&Cradrot);
  printf("Collimator Radius of Rotation: %6.4f (cm)\n",
 *Cradrot);
  prterror(FLAG,&nline,savline,errline);
  nline++;
  break;
      case 3: sprintf(savline,"%s",line);
  FLAG=getint(1,line,&NMAT);
  printf("Number of Materials: %d\n",*NMAT);
  if(FLAG==1){
      matdata=(float *)calloc(*NMAT*3,sizeof(float));
      printf("%3s%12s%12s\n","Ind","Attenuation","e-_Density");
      for(i=0;i<*NMAT;i++){
  ptr=fgets(line,120,outf);  errline++;
  if(line[0]=='#'){i--;}
  else{
  sprintf(savline,"%s",line);
  FLAG=getflt(3,line,&omatdata);
  if(FLAG==1){
    for(j=0;j<3;j++) *(matdata+i*3+j)=*(omatdata+j);
    free(omatdata);
    for(j=0;j<3;j++){
if(j==0)printf("%3.0f",*(matdata+i*3));
else printf("%12.7f",*(matdata+i*3+j));}
    printf("\n");}
  else i=*NMAT+1;
      }}
  }
  prterror(FLAG,&nline,savline,errline);
  nline++;
  break;
      case 4: sprintf(savline,"%s",line);
  FLAG=getint(1,line,&NOBJ);
  printf("Number of Attenuation Objects: %d\n",*NOBJ);
  if(FLAG==1){
      objdata=(float *)calloc(*NOBJ*ObjEntries,sizeof(float));
      printf("%3s%10s%10s%10s%10s%10s\n","Ind","Xpos"
      ,"Ypos","Xradius","Yradius","Obj_Type");
      for(i=0;i<*NOBJ;i++){
  ptr=fgets(line,120,outf); errline++;
  sprintf(savline,"%s",line);
  if(line[0]=='#'){ i--; }
  else{
  FLAG=getflt(ObjEntries,line,&omatdata);
  if(FLAG==1) FLAG=indinlst(*(omatdata),*NMAT,matdata);
  if(FLAG==1){
    for(j=0;j<ObjEntries;j++){
     *(objdata+i*ObjEntries+j)=*(omatdata+j);
     if(j==0)printf("%3.0f",*(objdata+i*ObjEntries));
    else printf("%10.4f",*(objdata+i*ObjEntries+j));}
  printf("\n");}
else{ i=*NOBJ+1; }
free(omatdata);}}
}
prterror(FLAG,&nline,savline,errline);
nline++;
break;
    case 5: sprintf(savline,"%s",line);
FLAG=getflt(2,line,&zlims);
if(*(zlims+1)<= *(zlims)) FLAG=0;
if(FLAG==1)
  printf("Objects extend (along z-axis) from %g to %g\n",
 *(zlims),*(zlims+1));
prterror(FLAG,&nline,savline,errline);
nline++;
break;
    case 6: sprintf(savline,"%s",line);
FLAG=getflt(1,line,&rotangle);
prterror(FLAG,&nline,savline,errline);
nline++;
break;
    case 7: get1str(line,attfile);
nline++;
break;
    case 8: get1str(line,densfile);
nline++;
break;
    case 9:  sprintf(savline,"%s",line);
FLAG=getint(1,line,&OUTTYPE);
if((*OUTTYPE<0)||(*OUTTYPE>2)) *OUTTYPE=0;
switch(*OUTTYPE){
case 0: sprintf(outtypestr,"new (%d)",*OUTTYPE); break;
case 1: sprintf(outtypestr,"old (%d)",*OUTTYPE); break;
case 2: sprintf(outtypestr,"single slice (%d)",*OUTTYPE);
    break;
}
printf("Writing output in %s format\n",outtypestr);
prterror(FLAG,&nline,savline,errline);
nline++;
break;
    }
}
    }
    if ((nline<MAXCASE)&&(FLAG==1)) {
printf("\n**** Input file is incomplete ****\n");
FLAG=0;}
 }
 return(FLAG);
```

```
}

void fixind(float *mat,int Nmat,float *obj,int Nobj)
{ int i,j;
  float index;
  for(i=0;i<Nobj;i++){
      index=*(obj+i*ObjEntries);
      for(j=0;j<Nmat;j++){
  if(fabs(index- *(mat+j*3))<ZERO){ *(obj+i*ObjEntries)=j*1.;
    j=Nmat+1;}
      }
  }
}

void main (int argc, char * argv[])
{  int i,j;
   float *attarr,*store,*dataline;
   int DIM,ind;
   float *head;

   if(argc<2) printf("map <input_file> (rot_angle(deg)) (output_format)\n");
   else {
       if(getdatafile(argv[1])==1){

       if(argc>2) {*rotangle=atof(argv[2]);
   printf("\nCommand line forcing rot_angle= %g (deg)\n",
  *rotangle);}
       if(argc>3) {*OUTTYPE=atoi(argv[3]);
   printf("Command line forcing output_format= %d\n",
  *OUTTYPE);}
       i=irint(*rotangle);
       if(i==0) sprintf(attfile,"%s.dat",attfile);
       else sprintf(attfile,"%s%d.dat",attfile,i);
       if(i==0) sprintf(densfile,"%s.dat",densfile);
       else sprintf(densfile,"%s%d.dat",densfile,i);

       centx=(*(BINS)+1)* *(BINS+1)/2.-0.5;
       centy=(*(BINS)+1)* *(BINS+1)/2.-0.5;
       DIM= *(BINS)* *(BINS+1);

       attarr=(float *)calloc(DIM*DIM, sizeof(float));
       store=(float *)calloc(*(BINS)* *(BINS),sizeof(float));
       head=(float *)calloc(NMAT,sizeof(float));

       fixind(matdata,*NMAT,objdata,*NOBJ);

       printf("\nWorking on attenuation map, object(%d): ",*(NOBJ));
       for(i=0;i< *(NOBJ);i++){
   printf("%3d",i+1); fflush(stdout);
   dataline=objdata+ObjEntries*i;
   ind=irint(*(dataline));
   chstatt(dataline,DIM,*(matdata+ind*3+1),attarr);
       }
       printf("\n");
       if(fabs(*rotangle)>0.) dorotation(*rotangle,DIM,attarr);
       binattarr(*(BINS),*(BINS+1),store,attarr);
       for(i=0;i<*NMAT;i++) *(head+i)=*(matdata+i*3+1);
       switch(*OUTTYPE){
         case 0: wrtattarr(attfile,*(BINS),head,*NMAT,zlims,store); break;
         case 1: wrtattold(attfile,*(BINS),head,*NMAT,zlims,store); break;
         case 2: wrtslice(attfile,*(BINS),store); break;
         default: printf("\n\n**** Error:  Unknown output type: %d  ****\n",
          *OUTTYPE); break;
       }

       att2dens(attarr,*(BINS)* *(BINS+1),matdata,*(NMAT));

       printf("Converting attenuation map to electron density map\n");
       att2dens(attarr,*(BINS)* *(BINS+1),matdata,*(NMAT));
       binattarr(*(BINS),*(BINS+1),store,attarr);
       for(i=0;i<*NMAT;i++) *(head+i)=*(matdata+i*3+2);
       switch(*OUTTYPE){
         case 0: wrtattarr(densfile,*(BINS),head,*NMAT,zlims,store); break;
         case 1: wrtattold(densfile,*(BINS),head,*NMAT,zlims,store); break;
         case 2: wrtslice(densfile,*(BINS),store); break;
         default: printf("\n\n**** Error:  Unknown output type: %d  ****\n",
       *OUTTYPE); break;
       }
   }
   }
}
```

A.5 NEW1GENSHRT.C

The code *new1genshrt.c* is used to generate the first order Compton scatter lookup tables used by *dualhxp.c* which specify the probability of a photon scattering in a given location and then being detected. This version of the code generates sparse version of the lookup tables which are then filled in by interpolation. This program accesses data files which describe the camera characteristics (see Chapter 5).

The tables for Rayleigh scatter are generated by the same code using a different input data file (the file *pknfnm* which fills the array *pknpthwindat*).

The code used to generate the second order Compton scatter lookup tables is very similar in structure to this code and so has not been included in this appendix.

```
/* program new1genshrt.c */
/*        Generating lookup tables from scratch
 *  ckerrp.c from March 4, 1996 converted back to full generation
 *     New Format
 *     Parameter File:  new1genshrt.param (nominal)
 * Command line:    new1genshrt <parameter file> (out_rt) (log_nm)
 */

#include <stdio.h>
#include <math.h>

#define DIM 64
#define pi 3.1415927
#define ZERO 1.e-11
#define L2look(l1,l3,l4,l5) l2look[(l1)*6*10*46+(l3)*10*46+(l4)*46+(l5)]
#define Kr 5
#define Krm (Kr-1)
#define LookSz 41*6*10*46
#define MaxCSO 41
#define MaxCSP 46
#define LINEMAX 11

/* Initialization of constants and matricies and global variables*/

float *l2look, *dcsOval, *dcspval;
float pknpthwindat[24001]={0.};
float Fofxi[10001]={0.};
double psiz[3], psiz1[3], halfpsiz[3], SILVL, COLLDIST;
double lowin=126.;
double hiwin=154.;
int centx=DIM/2,centy=DIM/2,centz=DIM/2;
int sincount,tag,SIN,SINCNTMAX;
double ZEROSTP;
FILE *bugf;

/* Prototype all functions */
double trapz (double (*func)(double *,double *, double *),
      double a[3], double b[3], double s,int it,double spos[3],
      double dpos[3], int dex);
double functn(double cs[3],double dc[3], double nil[3]);
```

```
void initdcs0val (void);
int dcs0_2ind(float val);
void initdcspval (void);
int dcsp_2ind(float val);
double qdetinteg(double spos[3], double cpos[3], double dpos[3]);
double qdet2integ(double spos[3], double cpos[3], double dpos[3]);
double qcom1integ(double spos[3], double cpos[3], double dpos[3]);
double qcom2integ(double spos[3], double cpos[3], double dpos[3]);
double qcom3integ(double spos[3], double cpos[3], double dpos[3]);
void polint(double *xa,double *ya, int n, double x, double *y,double *dy);
void calctpos(double sinth,double costh, double fi,double rad,
      double cpos[3],double npos[3]);
double pdet2integ(double spos[3], double cpos[3], double dpos[3]);
double pcom1integ(double spos[3], double cpos[3], double dpos[3]);
double pcom2integ(double *, double *, double *, double th,double rad);
double pcom3integ(double spos[3], double cpos[3], double dpos[3],double rad);
double srem1integ(double spos[3], double cpos[3], double dpos[3],double RadZ2);
double srem2integ(double spos[3], double cpos[3], double dpos[3],double Rad);

/* Utility programs */

void initdcs0val(void)
{ int i;
  for(i=0;i<MaxCS0;i++){
      if(i>27) dcs0val[i]=(i-27)*2.+16.;
      else {
  if(i>13) dcs0val[i]=(i-11)*1.;
  if(i<11) dcs0val[i]=(i-5)*0.2;
  if(i==11) dcs0val[i]=1.5;
  if(i==12) dcs0val[i]=2.0;
  if(i==13) dcs0val[i]=2.5;}
  }
}

int dcs0_2ind(float val)
{ int res;
  if(val>16.) res=irint((val-16.)*0.5)+27;
  else{
      if (val>2.75) res=irint(val+11.);
      else { if(val>0.9) res=irint(val*2.)+8;
             else res=irint(val*5.)+5;}}
  return(res);
}

void initdcspval(void)
{ int i;
  for(i=0;i<MaxCSP;i++){
      if(i>40) dcspval[i]=(i-40)*2.+32.;
      else { if (i>10) dcspval[i]=(i-8)*1.;
```

```
     else { if(i<6) dcspval[i]=i*0.2;
    else { switch (i) {
    case 6: dcspval[i]=sqrt(2.0); break;
    case 7: dcspval[i]=sqrt(3.0); break;
    case 8: dcspval[i]=2.; break;
    case 9: dcspval[i]=sqrt(5.0); break;
    case 10: dcspval[i]=sqrt(8.0); break;}}
}}
  }
}

int dcsp_2ind(float val)
{ int res;
  if (val>32.) res=40+irint(val*0.5-16.);
  else { if(val>2.914214) res=irint(val)+8;
 else {if (val<0.9) res=irint(val*5.);
       else { res=4+irint(val*val);
      if(res>9) res=10;}
   }}
  return(res);
}

void calctpos(double sinth,double costh, double fi,double rad,
      double cpos[3],double npos[3])
{ double cosfi,sinfi;
  cosfi=cos(fi);
  if(fi<pi) sinfi=sqrt(1.-cosfi*cosfi);
  else sinfi= -sqrt(1.-cosfi*cosfi);
  npos[0]=costh*sinfi*rad+cpos[0];
  npos[1]=sinth*sinfi*rad+cpos[1];
  npos[2]=cosfi*rad+cpos[2];
}

void writemat(char *ptfnm,float *matr,int Sz)
{ FILE *ptf;
  ptf=fopen(ptfnm,"w");
  fwrite(matr,Sz,sizeof(float),ptf);
  fclose(ptf);
}

int getmatr(char *ptfnm,int Sz,float *matr)
{ FILE *ptf;
  if( ptf=fopen(ptfnm,"r") ) {  printf("reading '%s'\n",ptfnm);
                fread(matr,Sz,sizeof(float),ptf);
                fclose(ptf);
return(1);}
  else {printf("PROBLEM WITH READING '%s'\n",ptfnm);
return(0);}
}
```

```
double functn(double cs[3], double dc[3],double nil[3])
{  double peff,ldcI,ldc2I,tmp;
   int thwin,xival;

   ldc2I=1./(dc[0]*dc[0]+dc[1]*dc[1]+dc[2]*dc[2]);
   ldcI=sqrt(ldc2I);
   tmp=dc[0]*ldcI;
/* if xi>2.44 deg, F(xi)=0.0  cos(2.5 deg)=0.99904823 */
   if(tmp<0.99905) peff=0.;
      else {if (tmp>1.) xival=0;
    else xival=irint((1.-tmp)*10526315.7895);
            peff=Fofxi[xival];
   tmp = (cs[0]*dc[0]+cs[1]*dc[1]+cs[2]*dc[2])*ldcI;
/* max th val considered is 120 deg => cos(120)=-0.5 */
   if(tmp<-0.5) peff = 0.0;
      else { if (tmp>1.) thwin=0;
     else thwin=irint((1.-tmp)*16000.);
     peff *=pknpthwindat[thwin]*ldc2I;
   } /* ends theta < 120 deg */
  } /* ends xi < 2.44 deg */
     return(peff);
 }


void polint(double *xa,double *ya, int n, double x, double *y,double *dy)
{  double c[Kr+1]={0.},d[Kr+1]={0.};
   int ns,i,j;
   double dif,dift,ho,hp,den;

   ns=0;
   dif=fabs(x-xa[0]);
   c[0]=d[0]=ya[0];
   for(i=1;i<n;i++){
     dift=fabs(x-xa[i]);
     if(dift<dif) {ns=i; dif=dift;}
     c[i]=d[i]=ya[i];}
   *y=ya[ns];
   ns--;
   for(j=1;j<n;j++){
     for(i=0;i<n-j;i++){
       ho=xa[i]-x;
       hp=xa[i+j]-x;
       if(fabs(ho-hp)<ZERO) printf("Problem in polint\n");
       else {
 den=(c[i+1]-d[i])/(ho-hp);
 d[i]=hp*den;
 c[i]=ho*den;}
     }
    if(2*ns<n-j) *dy=c[ns+1];
    else {*dy=d[ns]; ns=ns-1;}
    *y+=*dy;
 }
}



/* Worker subroutines  */

double trapz (double (*func)(double *,double *, double *),
      double a[3], double b[3], double s,int it,double spos[3],
      double dpos[3], int dex)
{  double del,x[3],tsum=0.;
   int i;
   del = (b[dex]-a[dex])/it;
   x[0]= a[0];
   x[1]= a[1];
   x[2]= a[2];
   x[dex]+=0.5*del;
   for (i=1;i<=it;i++){
          tsum+=(*func)(spos,x,dpos);
          x[dex]+=del;}
   s=0.5*(s+del*tsum);
   return(s);
}

double rtrapz (double (*func)(double *,double *, double *, double),
      double a[3], double b[3], double s,int it,double spos[3],
      double dpos[3], int dex, double RadZ2)
{  double del,x[3],tsum=0.;
   int i;
   del = (b[dex]-a[dex])/it;
   x[0]= a[0];
   x[1]= a[1];
   x[2]= a[2];
   x[dex]+=0.5*del;
   for (i=1;i<=it;i++){
          tsum+=(*func)(spos,x,dpos,RadZ2);
          x[dex]+=del;}
   s=0.5*(s+del*tsum);
   return(s);
}

/* Polar det2integ */
double pdet2integ(double spos[3], double cpos[3], double dpos[3])
{  double start[3], stop[3], ss, dss, lcs, lcs2, cs[3];
   int i,n,j;
   double sr[SIN+1], hr[SIN+1];
   hr[0]=1.;
   cs[0]=cpos[0]-spos[0];
   cs[1]=cpos[1]-spos[1];
   cs[2]=cpos[2]-spos[2];
   lcs2=cs[0]*cs[0]+cs[1]*cs[1]+cs[2]*cs[2];
   if(lcs2<ZERO) ss=0.;
   else{
       lcs=1./sqrt(lcs2);
       cs[0]*=lcs;
       cs[1]*=lcs;
       cs[2]*=lcs;
       start[0]=stop[0]=dpos[0];
       start[2]=stop[2]=dpos[2];
       start[1]=dpos[1]-halfpsiz[1];
       stop[1]=dpos[1]+halfpsiz[1];

       sr[0] = halfpsiz[1]*(qdetinteg(cs,start,cpos)+qdetinteg(cs,stop,cpos));
       i=1; n=1; dss=1.e5; ss=sr[0];
       while ((fabs(dss)>SILVL*fabs(ss)+ZEROSTP)&&(i<=SIN)){
   hr[i] = hr[i-1]*0.25;
   sr[i] = trapz ((double(*)(double *,double *, double *))qdetinteg,
  start,stop,sr[i-1],n,cs,cpos,1);
   n *= 2;
   j=i; if(j>Krm) j=Krm;
   polint(hr+(i-j),sr+(i-j),j+1,0.,&ss,&dss);
   i++;
       }
       if(i>SIN) sincount++;
   }
   return(ss);
}

/* The R integration */
double pcom1integ(double spos[3], double cpos[3], double dpos[3])
{  double start, stop, rstart[3], rstop[3];
   double rdist[3];
   double del,x,tsum;
   double dss,ss,sr[SIN+1],hr[SIN+1];
   double pss,pdss,rss,rdss;
   double psr[SIN+1],phr[SIN+1],rsr[SIN+1],rhr[SIN+1];
   int i=0,n=1,j,k;

   ZEROSTP=ZERO*SILVL*0.01;
   hr[0]=1.; phr[0]=1.; rhr[0]=1.;
   for(j=0;j<3;j++){
       rdist[j]=halfpsiz[0]-fabs(cpos[j]-spos[j]);
       if(rdist[j]<0.) rdist[j]=2000.;
   }
   for(j=2;j>0;j--){
       if(rdist[j]<rdist[j-1]) rdist[j-1]=rdist[j];}

   start=0.;
   stop=rdist[0];
   psr[0] = 0.5*stop*(pcom3integ(spos,cpos,dpos,start) +
pcom3integ(spos,cpos,dpos,stop));

   rstart[0]=rstop[0]=cpos[0];
   rstart[1]=rstop[1]=cpos[1];
   rstart[2]=cpos[2]-halfpsiz[2];
   rstop[2]=cpos[2]+halfpsiz[2];
   rsr[0] = halfpsiz[2]*(srem2integ(spos,rstart,dpos,stop)
+srem2integ(spos,rstop,dpos,stop));

   sr[0]=psr[0]+rsr[0];

   i++; dss=1.e5; ss=sr[0];
   pdss=dss; rdss=dss; pss=ss; rss=ss;
   ZEROSTP=sr[0]*SILVL*0.01;
   while ((fabs(dss)>SILVL*fabs(ss)+ZEROSTP)&&(i<=SIN)&&(sincount<SINCNTMAX)){
        hr[i] = hr[i-1]*0.25;
phr[i] = phr[i-1]*0.25;
rhr[i] = rhr[i-1]*0.25;
k=i; if(k>Krm) k=Krm;

if(fabs(pdss)>SILVL*fabs(pss)+ZEROSTP){
del = (stop-start)/n;
x= start+0.5*del;
for (j=1,tsum=0.;j<=n;j++){
    tsum+=pcom3integ(spos,cpos,dpos,x);
    x+=del;}
psr[i]=0.5*(psr[i-1]+del*tsum);
polint(phr+(i-k),psr+(i-k),k+1,0.,&pss,&pdss);}
else psr[i]=psr[i-1];

if(fabs(rdss)>SILVL*fabs(rss)+ZEROSTP){
        rsr[i]= rtrapz((double(*)(double *,double *, double *,
  double))srem2integ,
rstart,rstop,rsr[i-1],n,spos,dpos,2,stop);
polint(rhr+(i-k),rsr+(i-k),k+1,0.,&rss,&rdss);}
else rsr[i]=rsr[i-1];

n *= 2;
sr[i]=psr[i]+rsr[i];
polint(hr+(i-k),sr+(i-k),k+1,0.,&ss,&dss);

ZEROSTP=sr[i]*SILVL*0.01;
i++;
```

```
      }

   if(i>SIN) sincount++;
   return(ss);
}

/* The fi integration */
double pcom2integ(double spos[3], double cpos[3], double dpos[3],
  double th,double rad)
{  double start, stop, ss, dss, sinth,costh;
   int i,n,j;
   double sr[SIN+1], hr[SIN+1];
   double tpos[3];
   double del,x,tsum;

   hr[0]=1.;
   start=0.;
   stop=2.*pi;
   costh=cos(th);
   sinth=sqrt(1.-costh*costh);
   calctpos(sinth,costh,0.,rad,spos,tpos);
   sr[0] = pi*pdet2integ(spos,tpos,dpos);
/*     otots = pdet2integ(spos,tpos,dpos);
     calctpos(sinth,costh,stop,rad,spos,tpos);
     otots += pdet2integ(spos,tpos,dpos);
     total = 0.5*stop*otots; */

   i=1; n=1; dss=1.e5; ss=sr[0];
   while ((fabs(dss)>SILVL*fabs(ss)+ZEROSTP)&&(i<=SIN)){
        hr[i] = hr[i-1]*0.25;
del = (stop-start)/n;
x= start+0.5*del;
for (j=1,tsum=0.;j<=n;j++){
    calctpos(sinth,costh,x,rad,spos,tpos);
    tsum+=pdet2integ(spos,tpos,dpos);
    x+=del;}
sr[i]=0.5*(sr[i-1]+del*tsum);

n *= 2;
j=i; if(j>Krm) j=Krm;
polint(hr+(i-j),sr+(i-j),j+1,0.,&ss,&dss);
i++;
      }
   if(i>SIN) sincount++;
   return(ss*sinth);
}

/*  The theta integration  from 0 to pi */
double pcom3integ(double spos[3], double cpos[3], double dpos[3],
  double rad)
{  double start, stop, ss, dss;
   int i,n,j;
   double sr[SIN+1], hr[SIN+1];
   double del,x,tsum;

   hr[0]=1.;
   start=0.;
   stop=pi;

   del=stop;
   x= 0.5*stop;
   tsum=pcom2integ(spos,cpos,dpos,x,rad);
   sr[0]=0.5*del*tsum;

   i=1; n=2; dss=1.e5; ss=sr[0];
   while ((fabs(dss)>SILVL*fabs(ss)+ZEROSTP)&&(i<=SIN)){
        hr[i] = hr[i-1]*0.25;
del = (stop-start)/n;
x= start+0.5*del;
for (j=1,tsum=0.;j<=n;j++){
    tsum+=pcom2integ(spos,cpos,dpos,x,rad);
    x+=del;}
sr[i]=0.5*(sr[i-1]+del*tsum);

n *= 2;
j=i; if(j>Krm) j=Krm;
polint(hr+(i-j),sr+(i-j),j+1,0.,&ss,&dss);
i++;
      }
   if(i>SIN) sincount++;
   return(ss);
}

double srem1integ(double spos[3], double cpos[3], double dpos[3], double RadZ2)
{  double start[3], stop[3], ss, dss;
   int i,n,j;
   double sr[SIN+1], hr[SIN+1], RadY2, hold1, yskip[3];

   hr[0]=1.;
   start[0]=cpos[0]-halfpsiz[0];
   stop[0]=cpos[0]+halfpsiz[0];
   start[1]=stop[1]=cpos[1];
   start[2]=stop[2]=cpos[2];
   RadY2=RadZ2-(cpos[1]-spos[1])*(cpos[1]-spos[1]);
   if((RadZ2<=0.)||(RadY2<=0.)){
       sr[0] = halfpsiz[0]*(qdet2integ(spos,start,dpos)
    +qdet2integ(spos,stop,dpos));
       i=1; n=1; dss=1.e5; ss=sr[0];
      while ((fabs(dss)>SILVL*fabs(ss)+ZEROSTP)&&(i<=SIN)
     ){
  hr[i] = hr[i-1]*0.25;
  sr[i] = trapz ((double(*)(double *,double *, double *))qdet2integ,
 start,stop,sr[i-1],n,spos,dpos,0);
  n *= 2;
  j=i; if(j>Krm) j=Krm;
  polint(hr+(i-j),sr+(i-j),j+1,0.,&ss,&dss);
  i++;
      }
      if(i>SIN) sincount++;
  }
  else{
      RadY2=sqrt(RadY2);
      yskip[1]=start[1];
      yskip[2]=start[2];
      yskip[0]=spos[0]-RadY2;
      if(yskip[0]<start[0]) {
if(start[0]-yskip[0]>ZERO){
    printf("  Error start>yskip:  %g  %g  diff=%g\n"
   ,start[0],yskip[0],start[0]-yskip[0]);
  printf("      RadY=%g  spos[0]=%g\n",RadY2,spos[0]);
  printf("      cpos=(%g,%g,%g)\n",cpos[0],cpos[1],cpos[2]);
  fflush(stdout);}
ss=0.;}
      else{
      sr[0] = 0.5*(yskip[0]-start[0])*(qdet2integ(spos,start,dpos)
   +qdet2integ(spos,yskip,dpos));
      i=1; n=1; dss=1.e5; ss=sr[0];
      while ((fabs(dss)>SILVL*fabs(ss)+ZEROSTP)&&(i<=SIN)
     ){
  hr[i] = hr[i-1]*0.25;
  sr[i] = trapz ((double(*)(double *,double *, double *))qdet2integ,
 start,yskip,sr[i-1],n,spos,dpos,0);
  n *= 2;
  j=i; if(j>Krm) j=Krm;
  polint(hr+(i-j),sr+(i-j),j+1,0.,&ss,&dss);
  i++;
      }
     }
      if(i>SIN) sincount++;
      hold1=ss;

      yskip[0]=spos[0]+RadY2;
      if(yskip[0]>stop[0]) {
if(yskip[0]>ZERO+stop[0]){
  printf("  Error stop<yskip  %g  %g  diff=%g\n"
 ,stop[0],yskip[0],stop[0]-yskip[0]);
  fflush(stdout);}
 ss=0.;}
       else {
   sr[0] = 0.5*(stop[0]-yskip[0])*(qdet2integ(spos,yskip,dpos)
    +qdet2integ(spos,stop,dpos));
       i=1; n=1; dss=1.e5; ss=sr[0];
       while ((fabs(dss)>SILVL*fabs(ss)+ZEROSTP)&&(i<=SIN)
      ){
   hr[i] = hr[i-1]*0.25;
   sr[i] = trapz ((double(*)(double *,double *, double *))qdet2integ,
  yskip,stop,sr[i-1],n,spos,dpos,0);
   n *= 2;
   j=i; if(j>Krm) j=Krm;
   polint(hr+(i-j),sr+(i-j),j+1,0.,&ss,&dss);
   i++;
       }
       }
       if(i>SIN) sincount++;
       ss+=hold1;}
   return(ss);
}

double srem2integ(double spos[3], double cpos[3], double dpos[3],double Rad)
{  double start[3], stop[3], ss, dss, RadZ2;
   int i,n,j;
   double sr[SIN+1], hr[SIN+1];

   hr[0]=1.;
   RadZ2=Rad*Rad-(cpos[2]-spos[2])*(cpos[2]-spos[2]);
   start[0]=stop[0]=cpos[0];
   start[2]=stop[2]=cpos[2];
   start[1]=cpos[1]-halfpsiz[1];
   stop[1]=cpos[1]+halfpsiz[1];
   sr[0] = halfpsiz[1]*(srem1integ(spos,start,dpos,RadZ2)
+srem1integ(spos,stop,dpos,RadZ2));
   i=1; n=1; dss=1.e5; ss=sr[0];
   while ((fabs(dss)>SILVL*fabs(ss)+ZEROSTP)&&(i<=SIN)){
        hr[i] = hr[i-1]*0.25;
        sr[i] = rtrapz ((double(*)(double *,double *, double *,
   double))srem1integ,
start,stop,sr[i-1],n,spos,dpos,1,RadZ2);
n *= 2;
j=i; if(j>Krm) j=Krm;
polint(hr+(i-j),sr+(i-j),j+1,0.,&ss,&dss);
i++;
      }
   if(i>SIN) sincount++;
   return(ss);
}
```

```
/* for now will assume that dnml= (-1,0,0); */
double qdetinteg(double cs[3], double dpos[3], double cpos[3])
/* cs is normalized */
{  double start[3], stop[3], ss, dss;
   int i,n,j;
   double sr[SIN+1], hr[SIN+1];

   hr[0]=1.;
   start[0]=stop[0]=dpos[0]-cpos[0];
   start[1]=stop[1]=dpos[1]-cpos[1];
   start[2]=dpos[2]-cpos[2]-halfpsiz[2];
   stop[2]=dpos[2]-cpos[2]+halfpsiz[2];
   sr[0] = halfpsiz[0]*(functn(cs,start,cpos)+functn(cs,stop,cpos));
   i=1; n=1; dss=1.e5; ss=sr[0];
   while ((fabs(dss)>SILVL*fabs(ss)+ZEROSTP)&&(i<=SIN)){
        hr[i] = hr[i-1]*0.25;
        sr[i] = trapz ((double(*)(double *,double *, double *))functn,
start,stop,sr[i-1],n,cs,cpos,2);
n *= 2;
j=i; if(j>Krm) j=Krm;
polint(hr+(i-j),sr+(i-j),j+1,0.,&ss,&dss);
/* if(i>=Krm) polint(hr+(i-Krm),sr+(i-Krm),Kr,0.,&ss,&dss);
else { ss=sr[i]; dss=ss-sr[i-1]; }*/
i++;
      }
   if(i>SIN) sincount++;
   return(ss);
}

double qdet2integ(double spos[3], double cpos[3], double dpos[3])
{  double start[3], stop[3], ss, dss, lcs, lcs2, cs[3];
   int i,n,j;
   double sr[SIN+1], hr[SIN+1];

   hr[0]=1.;
   cs[0]=cpos[0]-spos[0];
   cs[1]=cpos[1]-spos[1];
   cs[2]=cpos[2]-spos[2];
   lcs2=cs[0]*cs[0]+cs[1]*cs[1]+cs[2]*cs[2];
   if(lcs2<ZERO) ss=0.;
   else{
       lcs=1./sqrt(lcs2);
       cs[0]*=lcs;
       cs[1]*=lcs;
       cs[2]*=lcs;
       start[0]=stop[0]=dpos[0];
       start[2]=stop[2]=dpos[2];
       start[1]=dpos[1]-halfpsiz[1];
       stop[1]=dpos[1]+halfpsiz[1];

       sr[0] = halfpsiz[1]*(qdetinteg(cs,start,cpos)+qdetinteg(cs,stop,cpos));
       i=1; n=1; dss=1.e5; ss=sr[0];
       while ((fabs(dss)>SILVL*fabs(ss)+ZEROSTP)&&(i<=SIN)){
   hr[i] = hr[i-1]*0.25;
   sr[i] = trapz ((double(*)(double *,double *, double *))qdetinteg,
  start,stop,sr[i-1],n,cs,cpos,1);
   n *= 2;
   j=i; if(j>Krm) j=Krm;
   polint(hr+(i-j),sr+(i-j),j+1,0.,&ss,&dss);
   i++;
       }
       if(i>SIN) sincount++;
       ss=ss/lcs2;
   }
   return(ss);
 }

double qcom1integ(double spos[3], double cpos[3], double dpos[3])
{  double start[3], stop[3], ss, dss;
   int i,n,j;
   double sr[SIN+1], hr[SIN+1];

   hr[0]=1.;
   start[1]=stop[1]=cpos[1];
   start[2]=stop[2]=cpos[2];
   start[0]=cpos[0]-halfpsiz[0];
   stop[0]=cpos[0]+halfpsiz[0];
   sr[0] = halfpsiz[0]*(qdet2integ(spos,start,dpos)
+qdet2integ(spos,stop,dpos));
   i=1; n=1; dss=1.e5; ss=sr[0];
   while ((fabs(dss)>SILVL*fabs(ss)+ZEROSTP)&&(i<=SIN)){
        hr[i] = hr[i-1]*0.25;
        sr[i] = trapz ((double(*)(double *,double *, double *))qdet2integ,
start,stop,sr[i-1],n,spos,dpos,0);
n *= 2;
j=i; if(j>Krm) j=Krm;
polint(hr+(i-j),sr+(i-j),j+1,0.,&ss,&dss);
i++;
      }
   if(i>SIN) sincount++;
   return(ss);
}

double qcom2integ(double spos[3], double cpos[3], double dpos[3])
{  double start[3], stop[3], ss, dss;
   int i,n,j;
   double sr[SIN+1], hr[SIN+1];
   hr[0]=1.;
   start[0]=stop[0]=cpos[0];
   start[2]=stop[2]=cpos[2];
   start[1]=cpos[1]-halfpsiz[1];
   stop[1]=cpos[1]+halfpsiz[1];
   sr[0] = halfpsiz[1]*(qcom1integ(spos,start,dpos)
+qcom1integ(spos,stop,dpos));
   i=1; n=1; dss=1.e5; ss=sr[0];
   while ((fabs(dss)>SILVL*fabs(ss)+ZEROSTP)&&(i<=SIN)){
        hr[i] = hr[i-1]*0.25;
        sr[i] = trapz ((double(*)(double *,double *, double *))qcom1integ,
start,stop,sr[i-1],n,spos,dpos,1);

n *= 2;
j=i; if(j>Krm) j=Krm;
polint(hr+(i-j),sr+(i-j),j+1,0.,&ss,&dss);
i++;
      }
   if(i>SIN) sincount++;
   return(ss);
}

double qcom3integ(double spos[3], double cpos[3], double dpos[3])
{  double start[3], stop[3], ss, dss;
   int i,n,j;
   double sr[SIN+1], hr[SIN+1];

   hr[0]=1.;
   ZEROSTP=ZERO*SILVL*0.01;
   start[0]=stop[0]=cpos[0];
   start[1]=stop[1]=cpos[1];
   start[2]=cpos[2]-halfpsiz[2];
   stop[2]=cpos[2]+halfpsiz[2];
   sr[0] = halfpsiz[2]*(qcom2integ(spos,start,dpos)
+qcom2integ(spos,stop,dpos));
   i=1; n=1; dss=1.e5; ss=sr[0];
   ZEROSTP=sr[0]*SILVL*0.01;
   while ((fabs(dss)>SILVL*fabs(ss)+ZEROSTP)&&(i<=SIN)&&(sincount<SINCNTMAX)){
        hr[i] = hr[i-1]*0.25;
        sr[i] = trapz ((double(*)(double *,double *, double *))qcom2integ,
start,stop,sr[i-1],n,spos,dpos,2);
n *= 2;
j=i; if(j>Krm) j=Krm;
polint(hr+(i-j),sr+(i-j),j+1,0.,&ss,&dss);
ZEROSTP=sr[i]*SILVL*0.01;
i++;
      }
   return(ss);
```

```
}

void main (int argc, char * argv[])
{
   FILE *parf;
   char *filnm,*ptr,*root,name[120],*lognm,*pknfnm,*fxifnm;
   char line[120];
   int i,j;
   double dnml[3],ssvec[3],srcpos[3],detpos[3],attpos[3],da[3];
   double distss,tmp,scatadd,newres,pdiff;
   double theta,dcxi,dceta,eta,xi;
   double sintheta,costheta,sinxi,cosxi,coseta;
   double collshift,sa0dist,sepdist;
   int limeta,lineno=1;
   int l1,l2,l3,l4,l5,l2Start=0;

   root=(char *)calloc(120,sizeof(char));
   lognm=(char *)calloc(120,sizeof(char));
   pknfnm=(char *)calloc(120,sizeof(char));
   fxifnm=(char *)calloc(120,sizeof(char));
   l2look=(float *)calloc(LookSz,sizeof(float));
   dcs0val=(float *)calloc(MaxCS0,sizeof(float));
   dcspval=(float *)calloc(MaxCSP,sizeof(float));
   if((l2look==NULL)||(dcs0val==NULL)||(dcspval==NULL))
     printf("Cannot allocate enough memory!\n");
   else /* can allocate memory */{

   if(argc<2) printf("new1genshrt <parameter file> (out_rt) (log_nm)\n");
   else /* have enough input arguments */ {
       if(!(parf=fopen(argv[1],"r"))){
   printf("Cannot read '%s'\n",argv[1]);}
       else /* have read parameter file */{
   while(((ptr=fgets(line,120,parf))!=NULL)&&(lineno<LINEMAX)){
       if(line[0]!='#'){
   switch(lineno){
   case 1:  sscanf(line,"%lf %lf %lf\n",
   &psiz[0],&psiz[1],&psiz[2]); break;
   case 2:  sscanf(line,"%lf\n",&COLLDIST); break;
   case 3:  sscanf(line,"%lf\n",&SILVL); break;
   case 4:  sscanf(line,"%d\n",&SIN); break;
   case 5:  sscanf(line,"%d\n",&SINCNTMAX); break;
   case 6:  sscanf(line,"%s\n",pknfnm); break;
   case 7:  sscanf(line,"%s\n",fxifnm); break;
   case 8:  sscanf(line,"%s\n",root); break;
   case 9:  sscanf(line,"%s\n",lognm); break;
   case 10:  sscanf(line,"%d\n",&l2Start); break;
   }
   lineno++;
       }
   }
   fclose(parf);

   printf("psiz= (%g,%g,%g)\n",psiz[0],psiz[1],psiz[2]);
   printf("CollDist= %g (cm)\n",COLLDIST);
   printf("SILVL= %g  SIN=%d  SINCNTMAX=%d \n",SILVL,SIN,SINCNTMAX);
   tag=irint(100.*(COLLDIST/psiz[0]-floor(COLLDIST/psiz[0])));
   collshift=(COLLDIST/psiz[0]-floor(COLLDIST/psiz[0]))*psiz[0];
   for(i=0;i<3;i++){psiz1[i]=1./psiz[i]; halfpsiz[i]=0.5*psiz[i];}
   detpos[0]=(0.5+centx)*psiz[0]+17.+COLLDIST;
   for(i=1;i<3;i++){ detpos[i]=(0.5+centy)*psiz[i]; }
   initdcs0val();
   initdcspval();
           getmatr(pknfnm,24001,pknpthwindat);
           getmatr(fxifnm,10001,Fofxi);
   free(pknfnm);
   free(fxifnm);

   if(argc>2) root=argv[2];
   if(argc>3) lognm=argv[3];
   printf("Generating 1st-order lookup table (from l2=%d)\n",l2Start);
   printf("   Unit size is: l1*l3*l4*l5=%d\n",LookSz);
   printf("   Writing log_file to '%s'\n",lognm);
   printf("   Writing results to '%sl2_(l2).mat%2.2d'\n",root,tag);
   bugf=fopen(lognm,"w");
   fprintf(bugf,"This is a log_file for new1genshrt.c\n");
   fprintf(bugf,"     Parameter File:  '%s'\n",argv[1]);
   fclose(bugf);
   printf("\nWorking...\n");

/* loop scatter-detector x-distance from 1 to 40 pixels */
   for(l2=l2Start;l2<56;l2++){
       if(l2>11) l2++;
       da[0]=l2*psiz[0]+collshift;
       attpos[0]=detpos[0]-da[0];
       sprintf(name,"%sl2_%d.mat%2.2d",root,l2,tag);
       for(i=0;i<LookSz;i++) *(l2look+i)=0.;

/* loop scatter-detector z-distance 0,1,sqrt(2),sqrt(5),2,sqrt(8) pixels*/
   for(l3=0;l3<6;l3++){
 switch(l3){
     case 0:  da[1]=0.; da[2]=0.; xi=0.; break;
     case 1:  da[1]=0.; da[2]=psiz[2]; xi=0; break;
     case 2:  da[1]=psiz[1]; da[2]=psiz[2]; xi=atan(psiz[1]*psiz1[2]);
      break;
     case 3:  da[1]=psiz[1]; da[2]=2.*psiz[2];
      xi=atan(psiz[1]*psiz1[2]*0.5); break;
     case 4:  da[1]=0.; da[2]=2.*psiz[2]; xi=0.; break;
     case 5:  da[1]=2.*psiz[1]; da[2]=2.*psiz[2];
```

```
      xi=atan(psiz[1]*psiz1[2]); break;
 }
 attpos[2]=detpos[2]-da[2];
 attpos[1]=detpos[1]-da[1];

/* loop eta from 0 to 180 degrees */
         if (l3==0) limeta = 4;
           else limeta = 10;
   for(l4=0;l4<limeta;l4++){
         if(l3==0){ eta=l4*0.261799388;
/*                eta=l4*15*pi/180.;*/}
         else{ eta = l4*0.34906585;
/*        eta = l4*20.*pi/180.;*/}
 ssvec[1]=sin(eta+xi);
 ssvec[2]=cos(eta+xi);

/* loop source-scatter x-distance from 0 to  pixels */
   for(l1=0;l1<41;l1++){
       if((l1==1)||(l1>12))l1++;
       sa0dist=dcs0val[l1]*psiz[0];
       srcpos[0]=attpos[0]-sa0dist;
       sa0dist=sa0dist*sa0dist;

/* loop distance source-scatter from 1 to 32 pixels */
   for(l5=0;l5<46;l5++){
       if(l5>11)l5++;
       distss=dcspval[l5]*psiz[1];
       srcpos[1]=attpos[1]-distss*ssvec[1];
       srcpos[2]=attpos[2]-distss*ssvec[2];
       sincount=0;

/* calling the polar and remainder routines and using them */
       sepdist=sqrt(distss*distss+sa0dist);
       if(sepdist>halfpsiz[0])
   newres = qcom3integ(srcpos,attpos,detpos)/(4.*pi);
         else newres = pcom1integ(srcpos,attpos,detpos)/(4.*pi);

 L2look(l1,l3,l4,l5)=newres;

     }}} /* close l5,l1,l4 loops */
 if((l3==2)||(l3==4)){
     bugf=fopen(lognm,"a");
     fprintf(bugf,"      Done loop l3=%d\n",l3);
     fflush(bugf);
     fclose(bugf);
     writemat(name,l2look,LookSz);}
     } /* close l3 loop */
 bugf=fopen(lognm,"a");
 fprintf(bugf,"Done loop l2=%d\n",l2);
 fflush(bugf);
 fclose(bugf);
 writemat(name,l2look,LookSz);
     } /* close l2 loop */
       }
   }
}
}
```